



DATA

UNDERSTANDING

DATA

ANALYSIS

DATA

SCIENCE



Volume 2: Fundamentals of Data Insight



PATRICK BOILY

Data Understanding, Data Analysis, and Data Science (Course Notes)

Volume 2: Fundamentals of Data Insight

Patrick Boily

January 2024

Quadrangle | Idlewyld Analytics and Consulting Services



This work is licensed under a [Creative Commons Attribution – NonCommercial – ShareAlike 4.0 International License](#) [↗](#).

Below is a human-readable summary of (and not a substitute for) the license. Please see [this page](#) [↗](#) for the full legal text.

You are free to:

Share – copy and redistribute the material in any medium or format

Remix – remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions – You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

This one goes out to the “Welsh” contingent: Elwyn,
Llewellyn, and Gwyneth. Your world is going to be a whole
lot different than mine was; maybe data can even help make
some of it better. But one thing’s for sure: data is not going
away any time soon – better be prepared.

Series Preface

The *first* thing to know about *Data Understanding, Data Analysis, and Data Science* (DUDADS) is that it isn't really a "book". It makes more sense to think of it as **course notes**, or as a **reference manual** and a source of examples and application.

I borrow some of its contents from authors who do a better job of explaining things than I could hope to do; I also sometimes modify their examples and code to better suit my pedagogical needs.* Major influences include [1, 2, 3, 4, 5, 6, 8] – be sure to give these masterful works the attention they deserve!

The *second* thing to know about DUDADS is that it isn't really "a" book. It makes more sense to think of it as **a bunch of books in a trench coat, masquerading as a single one**.[†] No one is expected to traverse DUDADS in one sitting, or even to tackle more than a few of its assigned chapters, sections, subsections, exercises at any given time; rather, it is intended to be read in parallel with guided lectures.

The *third* thing to know about DUDADS is that the practical examples use R and/or Python, for no particular reason other than that *some* programming language had to be used to illustrate the concepts. In the text, R code appears in blue boxes:

```
... some R code ...
```

Whereas Python code appears in green boxes:

```
... some Python code ...
```

You may look at some piece of code and think to yourself: "This isn't how I would do it" or "such-and-such a task would be easier to accomplish if we used module/package ABC or programming language XYZ". That's quite possible.

But finding the optimal tool is not the point of DUDADS. In the first place, new data science tools appear regularly, and it would be a fool's errand to try to continuously modify the book to keep up with them.[‡] In the second place, I am serious about the "understanding" part of *Data Understanding, Data Analysis, and Data Science*, and that is why I favour a **tool-agnostic** approach.

* In all cases, I have attempted to properly cite and give credit where it is due. Get in touch if you find omissions!

[†] I paid heed to this realization by splitting it into a number of volumes.

[‡] I am not saying that I won't be adding examples in different languages in the future, but let's not get ahead of ourselves.

The *fourth* thing to know about DUDADS is that it is not a place to go to in order to obtain a detailed step-by-step guide on “how to solve it”. In person, my answer to a vast array of data science related questions is, rather anti-climatically: “it depends”. Of course, it depends; on the data, on the objectives, on the cost associated with making a mistake, on the stakeholder’s appetite for uncertainty, and, perhaps more surprisingly, on the analytical and data preparation choices that are made along the way.


To some, this might smack of post-modernism: “you are saying that there is no truth, and that data analysis is pointless!” To which I respond: “analysts have agency (lots of it, it turns out), and their choices *DO* influence the results, so make sure to run multiple analyses to determine the variability of the outcomes”. That is the nature of the discipline.

The *last* thing you should probably know about DUDADS is that I have made a concerted effort to focus mainly on the **story** of (learning) data analysis and data science; sometimes, that comes at the expense of rigorous exposition.

“The early stages of education have to include a lot of lies-to-children, because early explanations have to be simple. However, we live in a complex world, and lies-to-children must **eventually be replaced** by more complex stories if they are not to become delayed-action genuine lies.” [7]

Some of the concepts and notions that I present are **incomplete** by design, but remain (I hope) true-to-their-spirit, or at least true “enough” for a first pass.[§] My position is that learning is an iterative process and that important take-aways from an early stage might need to be modified to account for new developments at a later date. But all things in good time: flexibility is a friend in your learning adventure; perfectionism, not always so.

Patrick Boily
Wakefield, January 2024
pboily@uottawa.ca

The DUDADS reference manuals are available at idlewyldanalytics.com 

- Volume 1: *Prelude to Data Understanding*
- Volume 2: *Fundamentals of Data Insight*
- Volume 3: *Spotlight on Machine Learning*
- Volume 4: *Techniques of Data Analysis*
- Volume 5: *Special Topics in Data Science and Artificial Intelligence*
- *The Practice of Data Visualization* (with S. Davies and J. Schellinck)

[§] In the parlance of the field, let me simply say that some of the details are left as an exercise for the reader (and can also be found in the numerous references).

Preface References

- [1] C.C. Aggarwal. *Data Mining: the Textbook* . Cham: Springer, 2015.
- [2] C.C. Aggarwal, ed. *Data Classification: Algorithms and Applications* . CRC Press, 2015.
- [3] C.C. Aggarwal and C.K. Reddy, eds. *Data Clustering: Algorithms and Applications* . CRC Press, 2014.
- [4] D. Dalpiaz. *R for Statistical Learning* . 2020.
- [5] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* , 2nd ed. Springer, 2008.
- [6] G. James et al. *An Introduction to Statistical Learning: With Applications in R* . Springer, 2014.
- [7] I. Stewart, J. Cohen, and T. Pratchett. *The Science Of Discworld*. Ebury Publishing, 2002.
- [8] H. Wickham and G. Grolemund. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data* . O'Reilly, Jan. 2017.

Contributors and Influences

A reference manual of this size could not have been compiled without the help of a multitude of individuals over the years, who provided contributions, influences, and/or inspiration:

Colin Daniel *Non-Technical Aspects of Quantitative and Data Work*

Ehssan Ghashim *Data Visualization and Data Exploration*

Lani Haque *Web Scraping and Automated Data Collection*

Andrew Macfie *Web Scraping and Automated Data Collection*

Aditya Maheshwari *Data Engineering and Data Management*

Maia Pelletier *Data Visualization and Data Exploration*

Bronwyn Rayfield *Non-Technical Aspects of Quantitative and Data Work*

Tristan Shaeen general editing

Jen Schellinck *Non-Technical Aspects of Quantitative and Data Work, Data Science Basics*

A hearty “thank you” to everyone, and to all others with whom I have crossed paths on this data adventure!

Learning Paths

I mostly use the material found in this volume for data workshops offered to professionals through *Idlewyld Analytics and Consulting Services*, the *Data Action Lab*, and the University of Ottawa's *Professional Development Institute*.

In particular, here is what I cover in various workshops:

- **Data Science Essentials** – Chapters 13–15, and 17;
- **Data Visualization and Dashboards** – Chapter 18 (as well as material found in *The Practice of Data Visualization*).

Chapter 16 is sometimes used as a component in special topics data analysis courses I teach at the University of Ottawa.

Contents

13 Non-Technical Aspects of Quantitative and Data Work	827
13.1 First Principles	827
13.1.1 Framework	828
13.1.2 The “Multiple I” Approach	830
13.1.3 Roles and Responsibilities	831
13.1.4 Analysis Cheatsheet	833
13.2 Project Life Cycle	834
13.2.1 Marketing	834
13.2.2 Initial Contact	836
13.2.3 Client Meetings	837
13.2.4 Assembling the Team	838
13.2.5 Team Meetings	838
13.2.6 Proposal	839
13.2.7 Contracting and IP	842
13.2.8 Project Planning	843
13.2.9 Information Gathering	845
13.2.10 Quantitative Analysis	846
13.2.11 Interpreting the Results	846
13.2.12 Reporting and Deliverables	847
13.2.13 Invoicing	848
13.2.14 Closing the File	849
13.3 Lessons Learned	849
13.3.1 About Clients	849
13.3.2 About Consultants	853
13.4 Business Development	856
13.4.1 Basics	856
13.4.2 Clients and Choices	856
13.4.3 Building Trust	857
13.4.4 Improving Trust	860
13.5 Technical Writing	862
13.5.1 Basics	862
13.5.2 Components	864
13.5.3 Traits	868
13.5.4 Examples	872
13.6 A Conversation With ...	877
13.7 Exercises	878
Chapter References	880
14 Data Science Basics	881
14.1 Introduction	881
14.1.1 What is Data?	881
14.1.2 Objects and Attributes	883
14.1.3 Data in the News	883
14.1.4 Analog vs Digital Data	884

14.2	Conceptual Frameworks	885
14.2.1	Three Modeling Strategies	886
14.2.2	Information Gathering	887
14.2.3	Cognitive Biases	892
14.3	Data Ethics	893
14.3.1	The Need for Ethics	894
14.3.2	What Is/Are Ethics?	894
14.3.3	Ethics and Data Science	895
14.3.4	Guiding Principles	896
14.3.5	The Good, the Bad, the Ugly	898
14.4	Analytics Workflows	898
14.4.1	The “Analytical” Method	898
14.4.2	Collection and Processing	901
14.4.3	Model Assessment	902
14.4.4	Automated Data Pipeline	903
14.5	Getting Insight From Data	903
14.5.1	Asking the Right Questions	905
14.5.2	Structure and Organization	909
14.5.3	Basic Analysis Techniques	919
14.5.4	Common Procedures in R	927
14.5.5	Quantitative Methods	940
14.5.6	Quantitative Fallacies	944
14.6	Exercises	946
	Chapter References	948
15	Data Preparation	951
15.1	Introduction	951
15.2	General Principles	952
15.2.1	Data Cleaning Approaches	952
15.2.2	Pros and Cons	952
15.2.3	Tools and Methods	953
15.3	Data Quality	954
15.3.1	Common Error Sources	955
15.3.2	Detecting Invalid Entries	955
15.4	Missing Values	957
15.4.1	Missing Value Mechanisms	957
15.4.2	Imputation Methods	958
15.4.3	Multiple Imputation	965
15.5	Anomalous Observations	966
15.5.1	Anomaly Detection	967
15.5.2	Outlier Tests	967
15.5.3	Visual Outlier Detection	970
15.6	Data Transformations	972
15.6.1	Common Transformations	973
15.6.2	Box-Cox Transformations	975
15.6.3	Scaling	979
15.6.4	Discretizing	979
15.6.5	Creating Variables	980
15.7	Example: Algae Blooms	980
15.7.1	Problem Description	980
15.7.2	Loading the Data	981

15.7.3	Summary & Visualization	982
15.7.4	Data Cleaning	993
15.7.5	Principal Components	997
15.8	Exercises	999
	Chapter References	1000
16	Web Scraping and Automatic Data Collection	1001
16.1	Data Analysis & Scraping	1001
16.1.1	Why Web Scraping?	1003
16.1.2	Web Data Quality	1003
16.1.3	Ethical Considerations	1005
16.1.4	Decision Process	1007
16.2	Web Technologies Basics	1007
16.2.1	Content Dissemination	1008
16.2.2	HTTP	1009
16.2.3	Web Content	1010
16.2.4	HTML/XML	1011
16.2.5	Cookies and Other Headers	1011
16.3	Scraping Toolbox	1012
16.3.1	Developer Tools	1012
16.3.2	XPath	1013
16.3.3	Regular Expressions	1023
16.3.4	BeautifulSoup	1027
16.3.5	Selenium	1033
16.3.6	APIs	1033
16.3.7	Specialized Uses	1034
16.4	Examples	1034
16.4.1	Wikipedia	1034
16.4.2	Weather Data	1041
16.4.3	CFL Play-by-Play	1049
16.4.4	Bad HTML	1056
16.4.5	Extracting Text from PDF	1057
16.4.6	YouTube Video Titles	1059
16.5	Exercises	1063
	Chapter References	1064
17	Data Engineering and Data Management	1065
17.1	Background and Context	1065
17.2	Data Engineering	1067
17.2.1	Data Pipelines	1068
17.2.2	Automatic Deployment	1073
17.2.3	Scheduled Pipelines	1075
17.2.4	Data Engineering Tools	1077
17.3	Data Management	1079
17.3.1	Databases	1079
17.3.2	Database Modeling	1082
17.3.3	Data Storage	1084
17.4	Reporting and Deployment	1086
17.4.1	Reports and Products	1086
17.4.2	Cloud vs. On-Premise	1087
	Chapter References	1088

18 Data Exploration and Data Visualization	1089
18.1 Data and Charts	1089
18.1.1 Pre-Analysis Uses	1090
18.1.2 Presenting Results	1090
18.1.3 Multivariate Elements	1091
18.1.4 Visualization Catalogue	1096
18.1.5 Accessibility	1099
18.2 Analytical Design	1099
18.2.1 Comparisons	1100
18.2.2 Mechanism/Explanation	1102
18.2.3 Multivariate Analysis	1104
18.2.4 Integration of Evidence	1106
18.2.5 Documentation	1107
18.2.6 Content First	1110
18.3 Dashboards	1111
18.3.1 Dashboard Fundamentals	1111
18.3.2 Dashboard Structure	1113
18.3.3 Dashboard Design	1114
18.3.4 Examples	1115
18.4 Exercises	1116
Chapter References	1118

List of Figures

13.1 A data science team in action	831
13.2 Avengers, assemble!	838
13.3 The communication continuum	863
13.4 The 5 components of TW; comparison with essay writing	865
13.5 Build-a-shark LEGO instructions	866
13.6 Changing a flat tire	868
13.7 Accessible description	871
14.1 A schematic diagram of systems thinking as it applies to a general problem	890
14.2 A conceptual model of the 'free software' system	890
14.3 UML diagram and ER conceptual map	891
14.4 The analytic workflow	899
14.5 Theoretical and corrupted CRISP-DM processes	900
14.6 The four analytical buckets	905
14.7 Different data cultures and terms	910
14.8 ER model diagram crow's foot relationship symbols cheat sheet	915
14.9 An implemented automated pipeline, with stages and transitions	918
14.10 AFM image of 1,5,9-trioxo-13-azatriangulene with chemical structure	919
14.11 Analysis and pattern-reveal through visualization	926
14.12 The trousers of classification	940
14.13 S&P stock price index, earnings, dividends, and interest rates (1871-2009)	942
14.14 Sales for 3 different products, measured in years, quarters, weeks	943

14.15	Illustration of Simpson's paradox	945
15.1	Data cleaning bingo card	953
15.2	Accuracy as bias, precision as standard error	954
15.3	An illustration of heaping behaviour	955
15.4	Imputation results in the grades data frame	964
15.5	Tukey's boxplot test	968
15.6	Summary visualisations for a plant dataset	970
15.7	Visualisations for a service point dataset	971
15.8	Frequency of the appendage lengths	972
15.9	Illustration of the curse of dimensionality	973
15.10	Various data transformations for a subset of the BUPA liver disease dataset	975
16.1	<i>Robots Exclusion Protocol</i> files	1005
16.2	Etiquette flow diagram for web scraping	1006
16.3	NHL's Atlantic Division standings on 20-Mar-2018	1007
16.4	NHL's Atlantic Division standings under the hood	1008
16.5	Comparison between HTML, XML, and JSON code	1009
16.6	Schematics of HTTP and AJAX requests	1010
16.7	Inspecting a website's elements using Developer Tools	1012
16.8	A simple HTML document, rendered in a browser	1013
16.9	HTML document tree for <code>fortunes.html</code>	1015
16.10	Generic node relations	1017
16.11	7-day forecast for Ottawa, ON	1042
16.12	7-day forecast for Ottawa, ON: <code>class=div-table</code>	1043
16.13	7-day forecast for Ottawa, ON: <code>class=div-column</code>	1043
16.14	7-day forecast for Ottawa, ON: <code>class=div-row1</code>	1045
16.15	7-day forecast for Ottawa, ON: <code>class=div-row2</code>	1045
16.16	CFL 2016 schedule and results	1051
16.17	CFL 2016 schedule and results: heading <code>collapsible-header</code>	1052
16.18	Play-by-play data for a CFL game	1054
16.19	Play-by-play data for a CFL game: <code>playbyplay-tab</code>	1054
16.20	Introduction to Quantitative Consulting YouTube playlist	1061
17.1	Diagram of a conceptual data pipeline	1069
17.2	Data visualization pipeline with component options	1071
17.3	An open-source data analysis pipeline	1077
17.4	A common data analysis pipeline	1078
18.1	Data visualization suggestions, by type of question	1091
18.2	Minimalist NASA CM1 scatterplot	1092
18.3	Deluxe NASA CM1 scatterplot	1092
18.4	2012 Health and Wealth of Nations (Gapminder)	1093
18.5	Choropleths of US elevation	1094
18.6	Network diagram of European language lexical distance	1095
18.7	Physical visualization of people's demographics	1096
18.8	Histogram of reported weekly work hours	1096
18.9	Classification scheme for the kyphosis dataset	1097
18.10	Estimated average project effort overlaid over product complexity, programmer capability, and product count in NASA's COCOMO dataset	1097
18.11	Trend, seasonality, shifts of a supply chain metric	1098
18.12	Diagnosis network around COPD in the Danish Medical Dataset	1098

18.13	Classification of two categories in an artificial dataset	1098
18.14	Hertzsprung-Russell diagram of stellar evolution	1099
18.15	Comparisons in the Gapminder chart: country-to-country	1101
18.16	Comparisons in the Gapminder chart: country-to-country overlap	1101
18.17	Comparisons in the Gapminder chart: country-to-average	1102
18.18	Approximate line of best fit for the Gapminder chart	1103
18.19	Close-up, bottom right quadrant	1103
18.20	Potential outliers in the Gapminder chart	1104
18.21	Potential clusters in the Gapminder chart	1105
18.22	Non-integrated Gapminder chart	1107
18.23	Legend inset for the Gapminder chart	1109
18.24	2013 Health and Wealth of Nations	1111
18.25	Exploratory dashboard for the Global Cities Index dataset	1113
18.26	Exploratory dashboard for the NHL draft class of 2015	1114
18.27	Anonymous 'ugly' dashboards	1115
18.28	'Zen' dashboard I	1119
18.29	'Zen' dashboard II	1120

List of Tables

13.1	An example of weekly time availability for students and professionals	844
13.2	A workplan example	845
15.5	Descriptive statistics for an appendage length dataset	971
16.1	Commonly-used XPath functions	1022

Non-Technical Aspects of Quantitative and Data Work

13

by **Patrick Boily**, with contributions from **Bronwyn Rayfield** and **Jen Schellinck**

With solid analytical and abstraction skills, individuals with a background in mathematics and statistics are in high demand. The gap between theory (or textbook applications) and real-world uses can prove surprisingly difficult to navigate, however, and can lead to challenges for first-time practitioners.

In this chapter, learners are introduced to various crucial non-technical aspects of quantitative and/or data work.

13.1 First Principles

The key component of data analysis and quantitative consulting is the ability to apply **quantitative methods** to business problems to obtain **actionable insight**. But it is impossible for any given individual to have expertise in **every** field of mathematics, statistics, and computer science. In our experience, the best consulting output is achieved when a small team of consultants possesses **expertise** in 2 or 3 areas, a **decent understanding** of related disciplines, and a **passing knowledge** in a variety of other domains.

This includes **keeping up with trends**, implementing **knowledge redundancies** on the team, being **conversant in non-expertise areas**, and **knowing where to find information** (online, in books, or external resources).

We present an overview of a variety of “domains” related to **quantitative analysis** in other chapters:

- survey sampling and data collection
- data processing
- data visualization
- statistical methods
- queueing models
- machine learning
- simulations
- optimization
- Bayesian data analysis
- anomaly detection and outlier analysis
- feature selection and dimensions reduction
- trend extraction and forecasting
- etc.

13.1 First Principles	827
Framework	828
The “Multiple I” Approach	830
Roles and Responsibilities	831
Analysis Cheatsheet	833
13.2 Project Life Cycle	834
Marketing	834
Initial Contact	836
Client Meetings	837
Assembling the Team	838
Team Meetings	838
Proposal	839
Contracting and IP	842
Project Planning	843
Information Gathering	845
Quantitative Analysis	846
Interpreting the Results	846
Reporting and Deliverables	847
Invoicing	848
Closing the File	849
13.3 Lessons Learned	849
About Clients	849
About Consultants	853
13.4 Business Development	856
Basics	856
Clients and Choices	856
Building Trust	857
Improving Trust	860
13.5 Technical Writing	862
Basics	862
Components	864
Traits	868
Examples	872
13.6 A Conversation With	877
13.7 Exercises	878
Chapter References	880

The domains are not free of overlaps. Large swaths of **data science** and **time series analysis** methods are quite simply **statistical** in nature, for instance, and it is not unusual to view **optimization** and **queueing** methods as sub-disciplines of **operations research**.

By design, our treatment of these topics will be **brief** and **incomplete**. Each chapter is directed at learners who have a background in quantitative methods, but not necessarily in the topic under consideration.

Our goal is to provide a quick “**reference map**” of the topic, together with a general idea of common **challenges** and **traps**, to highlight opportunities for application in a consulting context.

These chapters are not always meant to be comprehensive surveys: they often focus solely on **basics** and **talking points**. More importantly, a copious number of references are also provided.

We will complement some of these topics with write-ups of real-world consulting projects. For the time being, however, we focus on the **non-technical aspects of quantitative work**. Note that these are not just bells and whistles; analysts that neglect them will see their projects fail, no matter how cleverly their analyses were conducted.

This chapter is a companion piece to Chapter 14 (*Data Science Basics*); the latter contains a fair amount of must-read material for would-be data scientists and consultants, including:

- objects, attributes, and datasets
- modeling strategies and information gathering
- ethics in the data science context
- the “analytical” workflow
- roles and responsibilities of data analysis teams
- asking the right questions

In the rest of this section the terms **consultants**, **data scientists**, and **data analysts** are used interchangeably, as are the terms **clients** and **stakeholders**.¹

13.1.1 The Consulting/Analysis Framework

The perfect consultant/data scientist is both reliable and extremely skilled; in a pinch, it’s much better to be merely good and reliable than great but flaky. [Bronwyn Rayfield]

Consulting is the practice of providing **expertise** to an individual or organization in exchange for a **fee**.

Consultants may be hired to **supplement** existing staff (importantly, they are **NOT** hired as employees – consultants enjoy an **at-arm’s-length** relationship with their client) or to provide an **external perspective**. Consulting duties could include some of the following:

- **making recommendations** to improve products or services
- **implementing** solutions
- breathing **new life** into a failing project
- **training** employees
- **re-organizing** a company’s structure to remove inefficiencies, etc.

1: There are, to be sure, important differences: quantitative consultants do not have to be data people, and the relationship between employers/stakeholders and employees (a position held by quite a few data scientists) is of a distinct nature than that between client and consultant, but there are enough similarities for the analogy to be useful. Failing that, it could be a clever idea for data analysts and data scientists to get a sense for what motivates the consultants that might be brought in by their employers.

This seems straightforward, but there could be **complications**:

- Even though consultants are brought in by the organization, their presence is not always appreciated by employees.²
- If a consultant is brought in to implement solutions, the first question to come to mind should be: “why isn’t the company implementing the solution(s) themselves?” Is it because of a lack of resource? Are there political implications?
- The same goes for breathing new life into a failing project: why is the project failing? Is it a failure of leadership or of planning? Is the project infeasible? Are they looking for a scapegoat?
- In the training scenario, consultants need to recognize exactly how much can be done in the allotted time.³ Is the company hoping to offer the “illusion” of training? What kind of abilities the prospective trainees have? If they have the “right stuff”, why are they not training themselves? If they do not have the right skill sets and cannot be trained, what consequences might that have on success and/or reputation?

2: It is not too difficult to imagine how an outsider coming in and making recommendations to improve products and services, or to remove inefficiencies could be seen, in effect, as criticizing the current processes, let alone as potentially threatening employees’ livelihoods, causing a fair amount of friction and pushback.

3: Typically, the available time is quite short.

Consultants fall in one (or more) of the following types [6]:

Strategy Consultants focus on corporate strategy, economic policy, government policy, and so on; the projects they typically conduct for senior managers have more of an advisory nature than in implementation one;

Operations Consultants focus on improving the performance of a company’s or a department’s operations; they typically work with both strategy and technology people (in sales, marketing, production, finance, HR, logistics, etc.), on projects that run the gamut from advisory to implementation;

Human Resources Consultants focus on matters pertaining to human resources or on the workplace culture;

Management (Business) Consultants focus on variety of organizational concerns (this is a catch-all term to describe strategic, operational, and HR consultants);

Financial and Analytical Advisory Consultants focus on financial or analytical matters; for these consultants, subject matter expertise (tax law, risk analysis, statistics, etc.) is paramount;

Information Technology Consultants focus on development and application of IT, data analytics, security, and so on; they typically work on project, not on business-as-usual activities;

Specialized (Expert) Consultants are usually brought in for a specific task, which requires pointed expertise in a specific field.

For the purpose of this chapter, when we refer to **quantitative consultants** and/or **data scientists**, we usually mean someone who falls in one the last three categories, in short someone with expertise in a quantitative, analytical, technological, and/or technical field.

According to *International Management Consulting*, consultants benefit from:

- business **understanding** and external **awareness** (the so-called PESTLEE framework: political, economical, social, technological, legal, environment, ethics)
- being able to **manage** client relationships
- implementing the EDDD consulting process (engage, develop, deliver, disengage)
- being familiar with **various consulting tools and methods** specific to their area(s) of expertise, etc.

More specifically, good data scientists and quantitative consultants are expected to:

- have **business acumen**;
- learn how to **manage projects** from inception to completion, knowing that consultants are working with various people, on various projects, and that these people are also working on various projects;
- be able to slot into various **team roles**, recognize when to take the lead and when to take a backseat, when to focus on building consensus and when to focus on getting the work done;
- seek **personal** and **professional development**, which means that learning never stops;
- always display **professionalism** (externally and internally), a standard a behaviour and skills that need to be adhered to – take ownership of failures, share the credit in successes, treat colleagues, clients, and stakeholders with respect, and demand respect for teammates, clients, and stakeholders as well;
- act in accordance with their **ethical system**;
- hone their **analytical**, **predictive**, and **creative** thinking skills;
- rely on their **emotional intelligence**, as it is not sufficient to have a high IQ and recognize stated and tacit colleagues' and clients' needs, and
- **communicate effectively** with clients, stakeholders, and colleagues, to manage projects and deliver results.

13.1.2 The “Multiple I” Approach

While technical and quantitative proficiency (or expertise) is of course **necessary** to do good quantitative work, it is not **sufficient** – optimal real-world solutions may not always be the optimal academic or analytical solutions. This can be a difficult pill to swallow for individuals that have spent their entire education on purely quantitative matters.⁴

4: It certainly was for us...

The consultants' and analysts' focus must shift to the delivery of **useful analyses**, obtained *via* the **Multiple “I”** approach to data science:

- **intuition** – understanding the data and the analysis context;
- **initiative** – establishing an analysis plan;
- **innovation** – searching for new ways to obtain results, if required;
- **insurance** – trying more than one approach, even when the first approach worked;
- **interpretability** – providing explainable results;
- **insights** – providing actionable results;
- **integrity** – staying true to the analysis objectives and results;
- **independence** – developing self-learning and self-teaching skills;



Figure 13.1: A data science team in action, warts and all [Meko Deng, 2017].

- **interactions** – building strong analyses through (often multi-disciplinary) teamwork;
- **interest** – finding and reporting on interesting results;
- **intangibles** – putting a bit of yourself in the results and deliverables, and thinking “outside the box”;
- **inquisitiveness** – not simply asking the same questions repeatedly.

Data scientists and consultants should not only heed the Multiple “I”s at the delivery stage of the process – they can inform every other stage leading up to it.

13.1.3 Roles and Responsibilities

A data analyst or a data scientist (in the **singular**) is unlikely to get meaningful results – there are simply too many moving parts to any data project. Successful projects require **teams** of highly-skilled individuals who understand the **data**, the **context**, and the **challenges** faced by their teammates.⁵

Depending on the scope of the project, the team’s size could vary from a few to several dozens (or more!) – it is typically easier to manage small-ish teams (with 1-4 members, say).

Our experience as consultants and data scientists has allowed us to identify the following **quantitative/data work roles**.⁶

Project Managers / Team Leads must understand the process to the point of being able to recognize whether what is being done makes sense, and to provide realistic estimates of the time and effort required to complete tasks. Team leads act as interpreters between

5: Many newly-minted consultants and data scientists have not had enough experience with **effective team work**, and they are likely to underestimate the challenges that usually arise from such an endeavour.

6: Note that individuals can play more than one role on a team.

7: They may also need to shield the team from clients/stakeholders.

the team and the clients/stakeholders, and advocate for the team.⁷

They might not be involved with the day-to-day aspects of the projects but are responsible for the project deliverables.

Domain Experts / SMEs are, quite simply, authorities in a particular area or topic. Not “authority” in the sense that their word is law, but rather, in the sense that they have a comprehensive understanding of the context of the project, either from the client/stakeholder side, or from experience. SMEs can guide the data science team through the unexpected complications that arise from the disconnect between data science team and those “on-the-ground”, so to speak.

Data Translators have a good grasp on the data and the data dictionary, and help SMEs transmit the underlying context to the data science team.

Data Engineers / Database Specialists work with clients and stakeholders to ensure that the data sources can be used down the line by the data science team. They may participate in the analyses, but do not necessarily specialize in esoteric methods and algorithms. Most data science activities require the transfer of some client data to the analysis team. In many instances, this can be as simple as sending a .csv file as an e-mail attachment. In other instances, there are numerous security and size issues that must be tackled before the team can gain access to the data.

Data Analysts are team members who clean and process data and prepare the initial data visualizations. They have a decent understanding of quantitative methods. They typically have at most one area of expertise and can be relied upon to conduct preliminary analyses.

Data Scientists are team members who work with the processed data to build sophisticated models that provide actionable insights. They have a sound understanding of algorithms and quantitative methods, and of how they can be applied to a variety of data scenarios. They typically have 2 or 3 areas of expertise and can be counted on to catch up on new material quickly.

Computer Engineers design and build computer systems and other similar devices. They are also involved in software development, which is frequently used to deploy data science solutions.

Artificial Intelligence/Machine Learning QA/QC Specialists design testing plans for solutions that implement AI/ML models; in particular, they should help the data science team determine whether the models are able to learn.

Communication Specialists are team members who can communicate the actionable insights to managers, policy analysts, decision-makers and other stakeholders. They participate in the analyses, but do not necessarily specialize in esoteric methods and algorithms. They should keep on top of popular accounts of quantitative results. They are often data translators, as well.

Another complication: data science projects can be downright **stressful**. In an academic environment, the pace is significantly looser, but

- deadlines still exist (exams, assignments, theses),
- work can pile up (multiple courses, TAs, etc.)

In the workplace, there are two major differences:

- a data science project can only really receive 1 of 3 “grades”: **A+** (exceeded expectations), **A-** (met expectation), or **F** (did not meet expectations);
- while project quality is crucial, so is **timeliness** – missing a deadline is just as damaging as turning in uninspired or flawed work; perfect work delivered late may cost the client a sizeable amount of money.

Sound **project management** and **scheduling** can help alleviate some of the stress related to these issues. These are the purview of project managers and team leads, as is the maintenance of the quality of **team interactions**, which can make or break a project:

- ALWAYS treat colleagues/clients with respect – that includes emails, Slack conversations, watercooler conversations, meetings, progress reports, etc.;
- keep interactions **cordial** and **friendly** – you do not have to like your teammates, but you are all pulling in the same direction;
- keep the team leader/team abreast of **developments** and **hurdles** – delays may affect the project management plan in a crucial manner (plus your colleagues might be able to offer suggestions), and
- respond to requests and emails in a timely manner (within reason, of course).

13.1.4 Analysis Cheatsheet

We will end this section with a 12-point **TL;DR** (too long; did not read) snippet that summarizes the profession. These were collected (sometimes painfully) throughout the years (see [2] for more details).

1. Business solutions are not always academic solutions.
2. The data and models do not always support the stakeholder/client’s hopes, wants, and needs.
3. Timely communication is key – with the client and stakeholders, and with your team.
4. Data scientists need to be flexible (within reason), and willing and able to learn something new, quickly.
5. Not every problem calls for data science methods.
6. There are things to be learned both from good and bad experiences.
7. Manage projects and expectations.
8. Maintain a healthy work-life balance.
9. Respect the client, the project, the methods, and the team.
10. Data science is not about how smart we are; it is about how we can provide actionable insight.
11. When what the client wants cannot be done, offer alternatives.
12. “There ain’t no such thing as a free lunch.”

13.2 Project Life Cycle

Based on our experience, we think that there are twelve steps in the **consulting life cycle** (see [The Consulting Life Cycle](#) [↗](#) and [2] for details):

1. **marketing** – getting the word out;
2. **initial contact** – start discussions with prospective clients;
3. **first meeting (and meetings)** – committing to write a proposal for the client;
4. **assembling a team**;
5. **proposal and planning** – laying out what can be done for the client;
6. **contracting, insurance, IP** – if the client agrees to the proposal, this step is crucial: do not start work until this step is cleared up;
7. **information gathering** – may include data collection and cleaning, meeting with domain experts and in-house specialists to get a sense of the context;
8. **analysis** – where quantitative skills come in to play;
9. **interpretation of results** – this is what the client actually cares about;
10. **reporting, dashboarding, deployment** – there are often multiple deliverables along the way;
11. **invoicing** – required to get paid;
12. **closing the file** – conducting a post-mortem with the client and with the team and deciding what is next.

8: Every project is different, without exception. The **project life cycle** can also differ from one project to the next, or from one client to the next, but on average, most of the steps highlighted in this section will be involved in one way or another.

In this section, we dig a little deeper into each of the steps.⁸

13.2.1 Marketing

Marketing is required to let **prospective** clients know that an individual or group is **in business** (as consultants), that they possess a **specific set** of qualifications, and that they are looking for **projects** on which to work.

There are numerous marketing approaches:

- word-of-mouth
- online
- event
- newsletter
- article
- content
- niche
- reverse
- etc.

Obviously, not all these methods are applicable to every consultant and to every context. The principle underwriting marketing is simple: if prospective clients do not know that consultants **exist**, the latter cannot be found.⁹

In a broad sense, marketing is anything and everything done by a consultant to **legitimately** “get an in” with a prospective client and to convince them to hire their services. As with dating, attempts to get in “illegitimately” are usually regarded poorly and can easily backfire.¹⁰

9: Marketing is analogous to dating in this manner – **you must put yourself out there**.

10: Exactly what constitutes illegitimate behaviour is not always easy to determine, and may vary from one client to the next, but lies and misrepresentations are big no-nos.

Large consulting firms typically have **marketing teams** or departments – that is to say, individuals who are dedicated to finding clients and projects.

In smaller firms, marketing is usually done by the consultants **themselves**. Individual consultants and sole proprietors often join up with one another to avoid duplicating marketing efforts and to minimize associated costs. Keep in mind, however, that this requires a certain amount of **business compatibility** and **ideological alignment**.

Marketing avenues should not be viewed in a fixed manner – not only are they constantly changing with the advent of new technologies,¹¹ but personal preferences and the appropriateness of a given approach may change over time.

And while good marketing is necessary to consulting success (whatever form this may take), it is not a sufficient condition; there is a **marketing point of diminishing returns**, after which the results are not worth the effort.

Which avenue should a consultant select? The following questions are worth asking:

- does the avenue give a consultant an “in”?
- can it convince a client to hire the consultant’s services?
- what is its genuine cost (time, financially, energy)?
- what is its initial vs. ongoing investment?
- what are the risks associated with it?
- are there universal guidelines to the approach?

As alluded to previously, what works for one project, one client, one consultant may not work for another – **beware the tyranny of past success!**

Marketing Materials Beginning quantitative consultants could benefit from some of the following avenues:

- current and customizable project-based CVs
- client testimonials (letters of reference, etc.)
- portfolio (online, offline), including personal and pro bono projects (GitHub repository, etc.)
- active social media presence
- updated and functional website/landing page
- blog articles/white papers on variety of topics
- brochures and business cards
- attending conferences and networking activities
- adverts
- etc.

We shall provide additional details for a few of these.

Project-Based CVs contain two main sections:

- **Traditional CV** (contact info, skills, selected achievements, relevant experience, education and personal development, personal)
- **Relevant Project Experience** (list, role, project description, related reports and presentations)

11: It is recommended that consultants **stay up-to-date** on these technologies; a principled stand against a new tech may garner support in an echo chamber, but it can also mark you as **out-of-touch** with a younger and more general audience.

Other items can be added, depending on the context (publications, teaching, etc.). The traditional section should be no more than four pages (note the suggested section **order**). Projects in different domains could also be used to showcase successes and breadth of knowledge (see [3] for samples).

Social media platforms include:

- LinkedIn (consider accepting invites from people you do not know, expand your network, post regularly)
- twitter (RT articles/posts of interest, with a commentary; get to know who the experts in the fields are and follow them; post regularly)
- Facebook, Instagram, TikTok, etc.

With an online presence, there might also be a need to separate your **personal** from your **professional** online identities; it is important to avoid the common pitfalls of online use (trolling, flame wars, lousy/generational spelling, etc.), and to keep up with new tools.

Blog articles can be used for:

- content aggregation
- interacting with community
- pushing content
- showcasing communication, technical skills, and interests

12: Note that if you are going to base an article off of a project, you should make sure to obtain **client permission** first.

These are not journal articles, so **effective communication is key**.¹² Examples can be found on the *Data Action Lab* blog [8]. Remember: any (legitimate) approach is on the table if it helps get a prospective client interested in your services; consult [Marketing](#) [↗](#) and [2] for more details.

13.2.2 Initial Contact

Once a prospective client expresses an interest (no matter how faint) in working with a consultant, whether through email, a phone call, or some other approach, the consultant should:

- immediately dedicate a **project number**, an **email folder**, and a **folder in their file structure** to the potential project;
- capture and verify **email addresses** and **phone numbers** as soon as possible;
- respond to advances **promptly** (without seeming too desperate), and
- show interest in the project, **even** if the subject matter is not to your liking or if you are not an expert on the required methods

It is too early to turn down a client at this stage due to lack of interest and/or qualifications;¹³ the **goal** remains to gather some initial information about the project and set-up a meeting to discuss it in detail. Consult [Initial Contact](#) [↗](#) and [2] for more details.

13: If a project must be turned down, it is much preferable to invoke **lack of availability** or to suggest **another lead** instead

13.2.3 Client Meetings

If the client agrees to a meeting and the consultant has a meeting space, the client *may* decide to come to the consultants. If so, the consultants need to make sure that a private space is available, with:

- wi-fi connectivity and plenty of electrical outlets,
- projectors and other devices,
- water/coffee/pastries/muffins, and A/C or ventilation, etc.

If the consultant does not have a meeting space on hand, they should instead secure a shared meeting space or simply offer to go to the client.¹⁴

Furthermore, they should:

- bring a laptop computer (with battery pack or electrical cord);
- bring identification;
- go to the bathroom before you enter the client's office;
- **refrain from being late** and arrive 15 minutes early (scope the parking/busing situation), and
- dress appropriately (business, business casual?).¹⁵

(E-)business cards/marketing material should be available, to be traded **before** the first meeting starts. Eye contact and small talk (weather, sports, news events, etc.) should not be neglected – consultants and analysts are not solely being gauged as technical experts, but as **human beings** as well – it is a rare client that wants to work with a “robot”.

Remember that the “interview” process goes both ways; the consultant is also trying to determine if they want to work with the client. Before a contract has been agreed to, everything is still only **tentative**; as such, it is crucial to get a sense for client-consultant compatibility.

In general, it is preferable to let the client take the lead in describing their **situation** and **needs**. The consultant should then:

- take notes (or have an assistant take notes);¹⁶
- not let misunderstandings (acronym, details specific to their industry or company, etc.) pass by unresolved – ask for **clarification**;
- **never interrupt the client** – instead, they should wait for a natural lull in the conversation to make contributions and show that the client's needs and the underlying situation are understood;
- ensure the first meeting is not about the consultant (or at least, little of it is) – **more listening, less talking**.

Clients sometimes ask consultants to provide solutions on the spot – consultants should neither acquiesce to this nor commit to a project, a price, or a timeline at the initial meeting. If pressed, they could instead say: “I’m going to bring this information back to my team and we will evaluate the project’s feasibility. You will hear from us within *x* days/weeks.”

This stage's **objectives** are to:

- get a sense of the project's feasibility and of its suitability for the consulting team, and
- gather information about data sources and quantitative requirements, as well as the client's understanding of the same.

Consult [Meetings](#) ☞ and [2] for details.

14: That should be the first option offered, as a courtesy.

15: At the very least, consider wearing slacks/skirt, dress shirt, belt, dress shoes.



[Author unknown] These can differ based on cultural and epochal norms. Either way, you can adjust as necessary after the first meeting, but it is preferable to err on the side of “over-dressed” to start.

16: Ask for permission before recording anything.



Figure 13.2: Finding the right team members for a data science/quantitative consulting project can be a difficult task [DeviantArt artist k-3000].

13.2.4 Assembling the Team

Will consultants and analysts be working on this project alone? With a team? What roles are needed on the team? Who is available? There are pros and cons to both individual work and teamwork.

Individual Work:

- bigger share of revenues for the consultant;
- resource management easier to handle;
- no need for team meetings;
- latitude in accepting/rejecting projects;
- nobody tells anyone what to do (except for the client, perhaps);

Teamwork:

- more available resources, so project can be completed quicker (although this is only true up to a point in practice);
- more knowledge/ideas at the team's disposal;
- only one person needs to interact with client;
- managing egos and personalities can be difficult, at times;
- there is psychological strength in numbers (in theory, at least).

Our experience suggests that teams can usually achieve more, but this can come at a price: for some consultants, the level of satisfaction derived by a project might be affected by how much compromise was needed to see it through.

But the objective remains simple: assembling the team requires finding competent and pleasant people to work with. It is worth taking the time to build a team that will **work**. Consult [Assembling the Team](#) ¹⁷ and [2] for more details.

13.2.5 Team Meetings

Once a team has been assembled, it becomes crucial to plan for **team meetings**. Nobody likes those,¹⁷ but they are **crucial** to the project's success.

17: Nobody we have ever met, at least.

The designated **meeting lead** should prepare an agenda and is responsible for the team sticking to it; that step is needed because team meetings can easily become **time sucks**. Team members should take such meetings seriously; remember – **the project’s success depends on every consultant doing their work!**

Goals:

- keep the project is on track;
- exchange vital info between teammates (changes to the work plan, new discoveries by other members, etc.);
- keep team lead abreast of progress.

Consult [Meetings \(Reprise\)](#) [↗](#) and [2] for more details.

13.2.6 Proposal

If, after deliberations with the team, the project is deemed feasible (by the consultants **and** the clients), a proposal must be presented to the client. The proposal is the **foundation** of its eventual success – it is the opening salvo in the negotiation with the client.¹⁸

One of the challenges facing consultants is how to gauge the value of the services they offer – **we tend to sell ourselves short**. The proposal justifies the monetary demands to the client; it also helps limit what is known as **scope creep**.¹⁹ Consult [Proposal and Project Planning](#) [↗](#) (1:05-10:40) for more details.

A proposal should read as a **letter** to the prospective client. Its content may change depending on the specifics of each project, but the following sections should figure in the final document:

- Background
- Objective and Scope
- Methodology
- Milestones and Deliverables
- Schedule and Assumptions
- Resources and Costs
- Travel and Invoicing
- Appendices – Suggested Workplan; List of Former Relevant Projects and Clients; CVs and Bios, etc.

Let us take a more in-depth look at their contents.

Background:

- introduction
- state what consultants understand of the client’s organization (research this)

Objective and Scope:

- state what consultants understand of the client’s problem (go back to client to clarify if needed)
- delimit the tasks (“we will do such and such”, “we will not attempt to do such and such”) and keep in mind that the client may require options

18: The military imagery is intentional.

19: The client may ask for small changes, which can grow into major headache: a different font, a different chart, a different analysis, a different dashboard, a different product, etc. Opening the door to proposal modifications also opens the door to the client taking advantage of the analysis team, even if they do not intend to do so.

Methodology:

- suggest a series of steps / methods that the consultants will follow – the idea is to show the client that the team has already started thinking about their problem
- add a caveat that the data will be driving what method is used

Milestones and Deliverables:

- explicitly list the important steps and deliverables that will be produced for the client (prototype, final report, weekly progress reports, dashboard, executable code, etc.)

Schedules and Assumptions:

- provide a timeline for the milestones and deliverables, assuming that an agreement is reached by a certain date, or that the data is available by a certain date, etc.
- use **relative dates** if the client has deliverables or responsibilities for the project as well (better to err on the side of caution and **deliver on time and below cost** than the other way around!)
- establish the project authority on both the client and the consultant sides

Resources and Costs:

- list the resources that will be assigned to the project, with a short justification to reassure the client that the consultants are qualified to work on their project
- list the projected cost for each option (referring to the workplan as needed), with HST info
- reassure the client that they will not need to pay for work that is not done
- state that if more work needs to be done due to a change of scope, issues with data quality, or some other client issue,²⁰ the consulting team will wait until approval before starting the new work (**communication with the client is crucial!**)

20: Never call it a client error!

Travel and Invoicing:

- state what traveling costs will be charged to the client and that more expensive jaunts will only be undertaken with the client's approval
- state the invoicing policy – monthly / at milestones / upon completion, etc.

Suggested Workplan:

- provide a table with tasks and steps (follow the methodology), expected time expenditure, corresponding costs, and timelines
- produce a total estimate for the project (include the tax information)

Credentials and Credibility:

- add a list of previous clients for references
- add project-based CVs and short bios of team members
- add a list of other services offered by your team

The main objectives of the proposal is to let the clients know:

- what is understood of the problem at hand;
- what the consulting/data science team can do for them, and
- what they are expected to do in return.

Consult [3] for samples and [Proposal and Project Planning](#) (10:50-13:25) for more details on proposals.

The Client Dance

Assuming that the proposal is sent to the client within the agreed-upon deadline you have provided them, the next step in the process is the **client dance**.²¹

In general, the clients **do not know** (nor do they need to know):

- how busy the consultants are;
- how many projects they have on the go, and
- how many proposals are up in the air.

Conversely, **consultants do not know**:

- the project's priority for the client;
- the procurement challenges, and
- if multiple proposals were requested from different consultants.

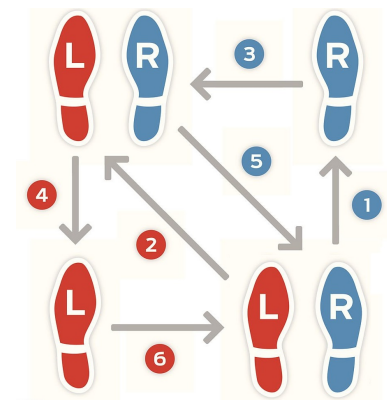
It is the **absence of this knowledge** that makes the client dance difficult to navigate. In the proposal, the client is given a deadline by which to respond to guarantee the availability of the consultant's resources:

- if they respond in time, then the next step is to **fine-tune** the proposal;
- if they respond after the deadline, then the consultants need to **reassess the situation** – perhaps the promised resources are not available anymore?;
- if they do not respond, the consultants must decide to either **poke them** or to **let the project go**.

If the client is not responsive, remember the **dating analogy** – sometimes the client dance is a negotiation tactic, but sometimes “they’re just not that into you.”

It is also important to remember that when embarking on a project with a public organization (such as the Government of Canada, for instance), the project authority on the client side is not necessarily the person who is responsible for the procurement process – that is typically conducted by a different group altogether, for whom the priorities are not necessarily in-step with the individuals the consultants have been in touch with up to that moment. All this to say that delays and complications should be expected, and are not (usually) meant as a show of disrespect.

21: Not usually as simple as this one:



[Getty Images]

22: In dating terms: will they still respect themselves in the morning?

So, from the consultants' side of things, the important questions to answer become: how desperate are they to get a project? How flexible are they? Can they afford to wait? Can they afford not to get the right terms?²²

Consult [Proposal and Project Planning](#) (from 13:25 onwards), for more details on the client dance.

23: When analysts are working for in-company stakeholders, this section can obviously be skipped.

13.2.7 Contracting and IP

After some back and forth, the consultants and the client might agree to a proposal – this is a required step in the process, but it does not constitute a **binding legal document**.²³

Consultants should **never** start work on a project until a legal contract has been signed by both parties. Some organizations insist on using their own contracts – some negotiation is possible, but they are not always willing to budge. Either way, it is crucial that consultants **REFRAIN FROM SIGNING A CONTRACT THAT THEY DO NOT UNDERSTAND OR AGREE WITH**.²⁴

24: Get legal advice from actual lawyers. For real. Please.

A **contract** sets out the roles, responsibilities, and legal obligations of both parties; it contains a number of clauses, distributed in a number of sections:²⁵

25: What constitutes an acceptable contract also depends on legal cultures – different jurisdictions may not use the same legal system (Ontario/Québec, for instance). We cannot over-emphasize how important it is to consult a lawyer for all your contracting questions.

- Identification of parties
- Acknowledgement of Processes by Parties
- Definitions
- Charges
- Term
- Conditions
- Payment
- Materials
- Confidential Information
- Warranties and Liability
- Force Majeure
- Termination
- Notices
- Conflict Between Documents
- Dispute Resolutions
- Waiver
- No Permanent Relationship
- Unenforceable Provisions
- Governing Law and Jurisdiction
- Singular and Plural
- Headings
- Amendment
- Language
- Fax and Counterparts
- Signatures

Contracts should never be prepared automatically – the clauses and sections may need to differ from one contract to another, although some of them may be used more frequently (see [3] for an example).

One aspect which is not explicitly listed above is that of **intellectual property**. Who owns the **results** of a consulting project? Most reasonable parties would conclude that it is the **client**; consequently, **non-disclosure agreements** (NDAs) are often required before a contract can be enacted.

But who owns the **methodology**? The **approach**? Can consultants re-use code for another project or publish the methods? Can an individual consultant use a method she developed as part of a team for her own work? Is it even possible for mathematical, statistical, analytical work to be patented or made **proprietary**?

This might seem like a frivolous question to ask at this stage, but consultants should take the time to reach consensus on this topic with their team, and with the client – this will save everyone a lot of heartache down the road.

The goal should be to make the eventual consulting work as simple as possible by removing the focus on anything but the quantitative analysis. Consult [Contracting and IP](#) ²⁶ and [2] for more information.

Insurance

From the client's perspective, a consulting project only has three outcomes, of which only the first two are every (ideally) in play: either

- the consultants exceed the expectations (managed *via* the proposal and open communication), A+;
- the consultants meet their expectations, A-, or
- the consultants fail to meet their expectations, F.

Given an "F", the **best case** scenario from the consultant's perspective is that the client will simply be disappointed and send future projects to other consultants; the **worst case** scenario is that they think that the consultants also failed to meet their **contractual obligations**, opening themselves to **legal action**.

Professional insurance against this (inevitability?) is a must.²⁶

13.2.8 Project Planning

While no actual work should be started before an agreement with the client is finalized, consultants should still start **planning the project** as soon as they start working on the proposal.

The goal is to ensure that consultants meet the project's (often) tight deadline without exhausting themselves in the process, so planning will help them hit the ground running!

Note that project management is often taught as separate course in business schools and there are various lists of available references (see [17], for instance) – it is even possible to get certification (as with [18], say). Such minutia is outside the scope of this section, however.

For quantitative consultants, the most important piece of advice is to prepare a timeline for **tasks/deliverables**, incorporating:

26: Note that, in Canada at least, the specifics of insurance also depend on the jurisdictions in which the client and/or the consultants operate and in which the product/service is delivered. Either way, it is important to talk to a specialist **BEFORE** things go astray.

Table 13.1: An example of weekly time availability for students and professionals; does it work with your work-life balance preferences? If not, what would?

Time Management (1 week = 168 hours)			
Students		Consultants	
Sleep	56	Sleep	56
Meals	15	Meals	15
Courses/study	40	Work	42
Work	20	Learning	10
Commute/errands	15	Commute/errands	15
Other	22	Other	30

- teammates' (and external resources') availability;
- projected delays;
- client bottlenecks;
- unexpected turns;
- holidays;
- client deadlines;
- simultaneous projects and courses;
- work-life balance, etc.,

while keeping *Hofstadter's Law* in mind:

"It always takes longer than you expect, even when taking *Hofstadter's Law* into account." [13]

Consequently, consultants should be prepared to revisit their workplan periodically, especially when preparing progress reports (internal and external) – this should not be seen as a failure, but as normal and expected **course corrections**, which occur in all project work.

Weekly Schedule

It is impossible to plan the project work without having a good sense of the team members' availability (an example is provided in Figure 13.1).

There is but a finite number of hours in a week, and each of us has responsibilities outside of work, including some necessary downtime and rest.

There is no value in creating a superhuman workplan that cannot be met – consultants must be realistic if they want to deliver on time. Failure to agree to a mutually acceptable schedule means that the project **should not go forward**.²⁷

Workplan

If the resources are available, the first project planning step is to produce a workplan that uses **high-level phases**, with item names that follow the proposal's methodology. Once these have been nailed down (with expected durations), the next step is to break down into various **task categories and sub-tasks**, with potential deliverables, timelines, and assigned team members associated to each task.

27: It is infinitely preferable to realize this **before** the contract is signed; the client is under no obligation to accommodate requests for extensions after an agreement has been reached.

Phase 1 – Data Preparation			
Tasks	Estimated Time (hrs)	Suggested Timeline	Deliverable
1.1 Importing and Hosting Data	2	Nov 9 - 13, 16 - 20	
1.2 Data Inspection (i.e. data structure, metadata info)	12		
1.3 Soundness and Data Quality	51	Nov 23 - 27, Nov 30 - Dec 4, Dec 7 - 11	
1.4 Dealing with Missing Observations	37	Dec 14 - 18	
1.5 Dealing with Anomalous Observations	37	Jan 4 - 8, 11 - 15	Deliverable 1: Cleaned Data Set (Nov 9 - Jan 15)
1.6 Data Contextualization	40	Jan 18 - 22, 25 - 29	
1.7 Data Visualization and Data Description	60	Feb 1 - 5, 8 - 12, 15 - 19	
1.8 Meetings and Reporting			
1.8.1 Creating Report Describing Findings	20	Feb 22 - 26, Feb 29 - Mar 4	
1.8.2 Meeting To Review Report	1	Mar 7 - 11	Deliverable 2: Data Visualizations (Jan 18 - Mar 18)
1.8.3 Revising Report Based on Client Feedback	10	Mar 14 - 18	Deliverable 3: Data Assessment Report (Jan 18 - Mar 18)
Phase 1 – Total Estimated Time (hrs)	270		
Phase 1 – Total Cost @80.00/hour	\$21,600.00		

Table 13.2: A workplan example.

This is as much an art as it is a science, and it can take a few sub-par projects before consultants get the hang of it. Experienced project leads can provide advice, if required, but analysts and consultants alike should be prepared to revisit and revise the workplan a number of times prior to the start of the project **AND** as the project is undertaken (see Table 13.2 for an example).

Consult [3] for samples, and [Proposal and Project Planning \(Reprise\)](#)  for more details.

13.2.9 Information Gathering

In the proposal, the consultants have demonstrated their understanding of the client's organization and of the project. But until the consultants **actually** start working on the project, that understanding may remain theoretical, at best.

Practical (and actionable) understanding is often gained through:

- field trips;
- interviews with subject matter experts (SMEs);
- readings;
- data exploration (even just trying to obtain the data can prove a pain),
- and other similar things.

The client is not a uniform entity – some of its data specialists and SMEs may even **resent** the involvement of external consultants.

This stage of the process is a chance to show the various client entities that the consultants are on their side; it is also a chance to gather valuable information that was not publicly available prior to the start of the project.

This can best be achieved by:

- asking **meaningful** questions;
- taking an **honest interest** in the SMEs experiences and expertise, and
- acknowledging their ability to **help**.

28: **Implicit** assumptions made at various stages, either by the consultant, the client, or both. Implicit assumptions are not necessarily invalid – problems arise when they are not shared by all parties (a gap which may only reliably be discovered by attempting to gather explicit information).

This is the consultants' first chance to identify **gaps in knowledge**,²⁸ which can sink a consulting project if they go undetected until they are remedied.

Much more will be said on the topic in Section 14.2.3 – its content should be thoroughly understood prior to embarking on this step of the process.

Consult [Information Gathering](#) and [2] for additional details.

13.2.10 Quantitative Analysis

If everything else has fallen correctly into place, consultants and analysts should now be itching to get going on the quantitative work.

Naturally, we assume that consultants and analysts have expertise in one or more of the following technical areas:²⁹

- data collection;
- data processing;
- data visualization;
- statistical analysis;
- data science and machine learning;
- optimization;
- queueing models;
- trend analysis and forecasting;
- simulations;
- etc.

This is where the bulk of the work comes in, and where **quantitative consultants** (as opposed to regular consultants) and **data scientists** get to shine. The quantitative consultant/data analyst's job is ... well, **to get the job done**. The time for dilly-dallying is long gone.³⁰

13.2.11 Interpreting the Results

When the analyses have been run, we obtain results. One thing to keep in mind is that clients are not actually interested in the results so much as they are interested in **insights** from analysis.

Actionable insights require results that can be interpreted and used by the client.³¹ Providing the deliverables correctly is surprisingly complicated: the only way to get this fact through to learners is through practice and experience.

The case study write-ups available at [2] focus on the **interpretation of results** and could be used as a source of inspiration – as is often the case, none of those projects required a thorough understanding of sophisticated methods.

The objective is to figure out what is useful for the client to know about the analysis outcomes and translate the analysis results accordingly.

29: See [2, 1] and the entirety of your degree(s) for more information ... as well as all the other chapters in this book and the other volumes in DUDADS.

30: And not a moment too soon, if you ask us.

31: Let it be said one last time: the best academic or theoretical solution may not be an acceptable solution in practice.

13.2.12 Reporting and Deliverables

If all has gone well, the project is now coming to an end.

Deliverables are **concrete** products provided by the consultants to the client in their search for **actionable** solutions. They constitute a type of **proof** that the work has been done.

They might include:

- deployment (code, software, apps), pseudo-code, conceptual ideas;
- literature review, case study write-up, recommendation(s), expert advice, popular account;
- **progress reports**, minutes of client meetings, notes, quality plan;
- **final report, presentation**, clean data, poster, executive summary, dashboard, user manual, white paper, technical article, etc.

Project deliverables depend on the client and on the project. Consult [Reporting and Deliverables](#) ³² and [2] for details.

Code, software, apps should be **documented** and **tested** prior to demos and delivery.³² Use programming guidelines and make sure that the code is devoid of unprofessional comments and variable/function names.

32: Code that does not work as it should when it should does not look very good on analysts and consultants.

Progress Reports

These let the client know what has been done, what is being done, what remains to be done:

- **keep to the essentials** – what is new, what is left to do, what SMEs are needed, timeline estimates, etc.;
- **frequency** should be arranged with the client (not more often than weekly, usually);
- can also be used **internally** (together with minutes and notes, for project management).

Final Report

The purpose of the project leads to the type of report. Typically, a final report (or potentially a storytelling dashboard, see [4]) contains at least the following sections (some can be lifted directly from the methodology):

- executive summary
- background
- objectives
- methodology
- results
- discussion / interpretation
- recommendations
- references

There will be instances where the story of the project is important (popular accounts, say), but in most instances consultants should strive to use **technical writing** (see Section 13.5). In either case, **say what needs to be said**, in a manner that is understandable and useful to the client.

The **executive summary** should include **recommendations** and **highlights** – it is directed at stakeholders and higher-ups who may not even be aware that the project has been undertaken.

Proof-read the report for spelling, grammar, and style; make the report **appealing** – forego fancy fonts and unusual font sizes. If you use mathematical symbols, consider using LaTeX or Markdown. Samples, as always, are available at [3]. Consult [Reporting and Deliverables](#) (from 06:30 onwards) for more details.

13.2.13 Invoicing

In a sense, this step is the most important of the process: consultants cannot get paid if they do not **invoice the client**.

The invoices should be kept **simple**; the included line items should be aligned with the deliverables and the milestones described in the proposal (and/or its amendments). Invoicing can take place

- **upfront**, such as for training sessions, say;
- at **regular intervals**, for (advisory or long-term work);
- **after** milestones and deliverables, for modular projects;
- **upon completion**, if the client is trusted, such as with a government department, or
- some **mixture** of those.

It goes without saying, the goal here is to keep money flowing to cover expenses and pay salaries.³³ Invoices should contain the consultant's contact detail, separate line items for the various deliverables, a line for the applicable taxes, payment options, and a payment deadline (30 days, 2 months, etc.), and so on.

Clients sometimes drop off the Earth without paying the invoice (an argument in favour of regular interval or milestone invoices if ever there was one); in that case, consultants need to decide for themselves when they will pursue the matter through legal means.

It is not a pleasant thought to entertain, but consultants need to be aware that this can happen in (rare) instances. It could be preferable to simply walk away without being paid, while documenting what happened (with signed contract and proof of delivery) and reporting the client to the better business bureau. In other cases, legal action could be justified.

A number of factors are at play here, so there is no one-size-fits-all approach but let us share the advice we were given when we started out as consultants: do not put all of your eggs in the same basket – only take on a project if its failure would not end your foray into consulting. It might help to know that public sector projects (such as those run by the various levels of Canadian government) are highly unlikely to turn into dine-and-dash affairs.


Consult [Invoicing](#) for more details.

33: Surprisingly, this is a step that some consultants have a challenging time doing – a possible explanation of this bizarre phenomenon can be found in the accompanying video.

13.2.14 Closing the File

The client has accepted the deliverables and has paid the invoices, and now the project is **over**. The last step in the project life cycle is the **post-mortem**, in which the team:

- analyzes the project process;
- identifies the high marks and the low points;
- plays the what-if game (how could thing have been done differently, in hindsight?);
- decides whether they would accept to take on another project with the client if the opportunity presented itself.³⁴

If they are amenable to doing so, the consulting team could also consider conducting a post-mortem with the client.³⁵ The **goal** of the (internal/external) post-mortem is not to assign blame, but to **learn lessons** (see Section 13.3) that can be applied to future projects. Consult [Closing the File](#)  for additional details.


34: There are no right or wrong answer here – remember the dating analogy: consultants have **agency**.

35: Fair warning: this process could be quite painful for the consultant/analyst's ego. Introspection is one thing when it is done with the team; being criticized by the client can prove quite unpleasant, even when it is not done with malice.

13.3 Lessons Learned

In the post-mortem, consultants and analysts synthesize what they learned about dealing with **clients** and **stakeholders**, as well as with their fellow **consultants** and **analysts**, throughout the project. The following lessons were (at times painfully) extracted from 40+ past projects.³⁶

13.3.1 About Clients and Stakeholders

More context for this section's content is provided in [Lessons Learned: About Clients](#) .

36: As ever, we use consultant and client interchangeably with analyst and stakeholder, respectively, but some of these lessons are only likely to be applicable in the consulting context.

Welcome to the Client Dance Getting a project off the ground can be exhausting (starting from when the client shows an initial interest to the time when work can start in earnest). Consultants start at a disadvantage and the client sometimes uses this as a negotiation tactic, by keeping the consultant waiting.

Solution: consultants should be polite, but they should also respect themselves, their abilities, and their work. While it will always be true that consultants need clients (to get paid), the converse is also true: clients need consultants (to get the work done).

37: Names and identifying details have been removed to preserve privacy but note the extent to which the dating analogy remains applicable.

Beware the Scope Creep and Divergent Expectations The client may start by asking for a little “something” which is not explicit in the agreement (a different font, colour scheme, etc.). These are not big demands, and the consultants may agree too do so (for several reasons).

Then the client might ask for a little something else (repeat the analysis with a slightly different dataset to reflect that new data has come in, say), and the consultant could run a few more analyses, and so on. On their own, none of these demands are “big deal”, but when all the demands

are added up, a whole new project has sprung up from these little bits, without the client having had to pay for it.

Solution: consultants should leave room in the agreement for modifications, with the caveat that the workplan may need to be revisited (as would timelines and costs).

It is entirely natural for the client to want something other than what they originally agreed to – as consultants start their quantitative work, they may expose conceptual and knowledge gaps, which could then lead the analysis into unexpected areas. Be that as it may, the agreement must be adhered to; modifications remain possible, but these should come at a cost to the client.

What Clients Want vs What They Need Clients do not always know what they want, from a quantitative perspective, and so what they want and what they need is not usually the same. This can come about because a previous consultant sold the client on an approach or buzzword, or because the client's competitors are doing something specific, and they feel they should follow suit.

Solution: it is the consultant's responsibility to offer advice on what the client needs, not necessarily what they want. This advice should be documented because the client might decide to disregard the consultant's advice and go with what they want (over what they need).

If the client comes to realize that what they wanted was not what they needed, the documentation should prevent the consultant turning into a scapegoat.

Talk is Cheap Some clients are very gregarious: they are full of promises and full of ideas when it comes to a project, they will engage consultants in the process, ... but for whatever reason they do not respond to the proposal, or they will not agree to a meeting, or they will not make the data available, etc.

Solution: consultants should not start work (in earnest) on a project until an official agreement has been reached.³⁸

Disappearing Clients Some clients pull out of the process at some stage (after the initial contact, after the proposal sent, after the project has started at invoicing, etc.).

Solution: consultants should withhold deliverables until contact has been re-established with the client.³⁹

Consultants should learn how to sniff out disappearing or flaky clients; one way to reduce the risk is by maintaining healthy and regular communication. Consultants may have legal recourse if a client disappears at invoicing time, but they should be ready for a fight.

38: Some basic prep work can still be conducted, however, but not at the expense of projects that have officially been agreed to.

39: Take the time to document attempts at reaching the client (email, phone calls, supervisors, etc.); this could come in handy at a later stage.

Helicopter Clients The opposite situation can also occur: some clients are micro-managers and want to be involved with every aspect of the project.

Solution: consultants need to learn how to sniff out helicopter clients early. The team lead should consider giving the client nominal work to do and get them out the consultants' way.⁴⁰

The team lead is responsible for sheltering the consulting team from this annoyance and may have to shoulder the brunt of the interactions with the client and may have to appeal to the agreement if it contains clauses that clearly delineate the responsibilities and roles of each party.

Desperate Clients In another common situation, clients sometimes turn to consultants as a final effort to save a project (or to shift the blame to an outsider). A desperate client is often identifiable by demands for unreasonable deadlines.

This toxic situation can quickly become unbearably stressful and taxing for the consultant team.

Solution: consultants should be clear about the scope, the objectives, and the deliverables, as often as required, and be ready to return to the proposal often to remind the client what has been agreed to. While consultants need to show flexibility to keep clients happy, some limits should not be crossed.

Dishonest Clients While the previous three lessons could be chalked up to clueless (and non-malicious) clients, the next one cannot. There is no sugar-coating it: some clients will knowingly try to take advantage of the consultants.⁴¹

Solution: at times, paying work might be hard to come by, but consultants still need to do their homework and see what there is to be found about the client from external sources before an agreement is reached – it is important to trust instincts.

Sometimes, the problems only appear after the contract has been agreed to. In that case, the priorities should be for consultants to protect their team (including themselves) by document conversations and collect a (e-)paper trail.

Consultants should avoid threatening to sue the client unless they are ready to follow through with the suit; contacting a lawyer as soon as a problem arise is necessary.

Procurement Issues The proposed project authority on the client side is not always the person who holds the purse's strings, nor do they necessarily have the final say on procurement matters – they may be interested in getting the project going, but company or departmental policies could “get in the way” and complicate the process.

Most client organizations have their own internal process to hire consultants; for (small-ish) private sector clients, the issues are likely to be minimal, but for larger private sector clients and public sector clients, there are rules in place to stamp out corruption and nepotism.

40: That way, the client feels like they are doing something, and they may stop interrupting the team with unreasonable requests.

41: We are not talking about miscommunication or honest mistakes, here – some clients have a track record of abusing consultants.

42: These vehicles require a lot of administrative set-ups on the part of the consultants, in Canada, at least [ProServices].

Procurement vehicles include sole-source contracts, standing offers, expert advisory agreement, professional services supplying, etc.⁴²

In many cases with contract value thresholds (sole-source contracts, say), clients sometimes try to squeeze a large project in under a small budget, because the only reasonable alternative would constitute contract splitting, which is disallowed (in the public service, at least, you cannot give two contracts for the same project).

Solution: consultants should avoid selling themselves short, not only because it will mark them as “easy marks”, but also for a more pragmatic reason: if the client can only offer \$25K for a project and the consultants agree to do \$50K’s worth of work, the result is a \$25K shortfall, which needs to be covered from somewhere else.

43: We’re not sure why that is the case, to be honest – if an organization does not trust its internal experts, they are not hiring the right employees, and that is entirely on them.

Speaking Truth to Power Organizations have the tendency to trust outsiders over internal experts,⁴³ so there could be instances when the client already has a pretty good idea of what the data is saying but they need a person who is external to the organization to relate it to stakeholders.

And it could also be that the client knows that whatever report will be delivered will be poorly received by the higher-ups and they do not want to suffer their wrath, so the consultant is brought in as the bearer of unwelcome news.

There is nothing wrong with this, but some consultants might not enjoy being set-up to fail (as they might see it).

Solution: it is better to have all the cards on the table so that both parties know for what purpose the consulting team has been hired.

Consulting Witches and Wizards Quantitative methods are seen as mysterious for a large swath of the population; consequently, experts in the field are sometimes viewed as witches and wizards.

As practitioners of the “magic arts”, consultants and data scientists are often saddled with expectations that can sadly not be met. Quantitative methods (coupled with sound data) can achieve many remarkable feats, but consultants do themselves (and their colleagues) a disservice by not managing expectations of their abilities early (and often).

Solution: consultants should be clear and direct about what they (and their methods) can do for the client and nip in the bud any delusions the client may harbour about the project’s outcome.

This is also in the client’s best interest and will stop them from making grandiose promises to their stakeholders – promises that simply cannot be kept.

It is always preferable to under-promise and over-deliver than *vice-versa*.

Calendars and Deadlines Time management is difficult in general, but in consulting projects, it is also difficult because clients are not always forthcoming about their own internal deadlines and calendars. They may attempt to move the project deadlines to synchronize with changes in their own deadlines.

The clients may also have deliverables for the project (getting back to the consultants in a timely manner, providing the data by a certain date, etc.).

Solution: when they hand in progress reports, consultants should remind clients of the timelines for each remaining task, as agreed to in the contract.⁴⁴

The proposal should also reflect the effect of the client not meeting their project deadlines. Having a clause that reads “task 3 will be completed 2 weeks after the data has been delivered by the client”, say, rather than “task 3 will be completed by October 10” makes it clear that if the client delivers the data on October 8, the consultants will not have to scramble to complete task 3 by October 10 – if the task takes 2 weeks to complete, the consultants should get 2 weeks to complete it.

Data Availability and Quality Invariably, the data is not as sound as the client thinks it is. And very often, due to internal politics the data will only be available way later than it was supposed to be, which can hamper the consultants’ ability to complete the project on schedule.

Solution: at no point should consultants accept the client’s word that the data is “good” and does not need to be cleaned/explored.⁴⁵ The proposal (and methodology) needs to reflect that data cleaning and data exploration are essential to the project’s success, and that consultants cannot guarantee the work unless they have access to the data.

Dealing with Adversity Even when all the stars are aligned, and the consultants did a top-notch job and the clients provided domain expertise and quality data on time, it remains possible that the analysis results will not be to the client’s liking – data does not bend to anyone’s wishes.

As it is possible that the consultants made errors along the way, the client may be in the right to ask for a re-do.⁴⁶ Where it becomes problematic is when they ask for a refund and/or put your credentials in doubt.


Solution: the proposal should reflect the nature of quantitative projects, i.e., the data/methods do not always support the client’s hopes. Consultants should not enter in a contractual agreement with clients who do not accept (and agree to) this fundamental fact.

44: There is nothing wrong with clients asking for the timeline to be revisited, and if the consultants can accommodate the new deadlines (in terms of resource availability), they should consider doing so. But the clients should not assume that a change is forthcoming just because the client’s deadlines have changed.

45: Always in a polite manner, of course.

46: Validation protocols should be in place, at any rate.

13.3.2 About Consultants and Colleagues

More context for this section’s lessons is provided in [Lessons Learned: About Consultants](#) .

Importance of the Post-Mortem It is impossible to learn any lesson if nobody knows what the lessons are. The importance of the post-mortem cannot be overstated.

Solution: post-mortems should always be conducted, even when the project was a success. It might also be a good idea to do project components post-mortem: consultants do not need to wait until the end of the project to identify what went well and what did not for a particular phase. Lessons are learned continuously.

Boom or Bust It is often the case that consultants (especially individuals and small teams) go through periods of months without a consulting project, followed by short periods where everyone wants you to work with/for them, which wreaks havoc on work-life balance.

Solution: to survive “boom” periods, consultants need stellar project management. “Bust” periods can be used for **business development** (Section 13.4), or for research and continuous learning.

Protecting Yourself Against Unreasonable Clients As discussed in the previous section, unreasonable clients will happen at some point in every consultant’s career. While this might seem to be a lesson about clients, learning to deal with them is a lesson about consultants.

A project going belly-up is not the end of the world, although it can certainly seem that way in the middle of the blow-up.

Solution: consultants should get access to lawyer and a support system before the first sign of trouble – it might be too late to do so after the fact. More importantly, consultants should be something other than consultants, at times – the benefit of physical exercise, hobbies, volunteering, personal time, and so on has been demonstrated repeatedly. Do not look down your nose at these simple solutions.

Teammates as Hurdles Teamwork is not easy. Sometimes it will feel like teammates are hindrances in the pursuit of the project’s success – they just do not understand what needs to be done or they focus on things that are considered superfluous or out-of-bound by other consultants.

Solution: whether the other team members are missing the boat or not, they (and by extension, the client too) need to be always treated with respect – there will be times when having them as part of the team will be an incredibly welcome development.

With rare exceptions, consulting teams achieve more than individual consultants (although that may be easier to do when well-defined consulting roles are assigned and adhered to).

Academia vs. Business World As quantitative consultants, technical skills are in high demand – they are experts and want to provide the best possible solution to their client. But their ways of thinking are often academic due to their training; they often favour the general over the specific.

In practice, theoretical solutions are not always actionable – they cannot always be turned into useful insights for the client.

Solution: consultants need to remember that it is not about them: “best results” in the consulting context means “most useful for the client”, not necessarily for the consultant’s publishing record – being right is important, but it is secondary to being useful.⁴⁷

47: Most consulting work is unsuitable for publication, in our experience.

Selling Yourself Short Many quantitative people sell themselves short to try to not intimidate their contemporaries or potential clients – we would strongly suggest not doing that. Consultants have skills, and their time and work are worth something (quite often, an extremely high something).

Solution: know your worth and be confident.

Being an Arrogant Blowhard But there is a slim line between confidence and cockiness – consultants being aware of their skills and technical proficiency should not translate as them being insufferable colleagues. Sherlock Holmes, Gregory House, and Sheldon Cooper might be intriguing TV characters, but consultants who try this approach in the real world will soon find themselves deserted by colleagues and clients alike, and without work.

Solution: consultants need to remember that consulting projects are not there for them to showcase how smart they are, but to give them a chance to help their clients achieve something they could not do on their own. There are tons of qualified quantitative people in the consulting ecosystem, and more are joining the fray every year – standing out for being arrogant or a blowhard is NOT a viable marketing strategy.⁴⁸

48: The dating analogy rears its head again: there are plenty of fish in the sea. Clients prefer their consultants to be friendly rather than annoying.

Keep Your Edge Nothing ages faster than a quantitative consultant and their expertise;⁴⁹ methods do not suddenly become invalid, but they can easily become *dépassé*.

49: When we were students, there were barely any business applications for machine learning, for instance.

Solution: consultants need to stay up-to-date and continue learning new methods and approaches on a regular basis, while maintaining their qualifications/certifications.

Keep your eyes peeled and any eventual shock will be lessened – things rarely go exactly as planned, but keeping track of these lessons and learning to recognize the warning signs can only be beneficial to quantitative endeavours in the long run.⁵⁰

50: You may encounter some of those in your own consulting/data analysis endeavours, but you will no doubt also encounter your very own lessons over time.

51: This section does not apply as obviously to employees as it does to consultants, but some of the concepts presented here can still be transferred to the former's context, inasmuch as employees have control over the work they do.

13.4 Business Development

There is one aspect of the job which does not usually come naturally to quantitative people: they also must be businesspeople. **Business development** (BD) is anything that helps to develop new business (for the company or for individual quantitative workers).⁵¹

Most of the material in this section comes from [15, 7, 16] and from a presentation provided by [ApexRMS](#) [↗](#); it is also covered in the following videos:

- [The Basics of Business Development](#) [↗](#)
- [Clients and Choices](#) [↗](#)
- [Building Trust](#) [↗](#)
- [Improving Trust](#) [↗](#)

13.4.1 Basics

Consultants have two types of clients: **external clients** (the usual stuff: organizations/individuals with whom they contract to offer services) and **internal “clients”**, especially in larger consulting shops, where consultants/analysts

- sell their services to project managers, and
- support project managers in delivering to their clients.

In this view, everything that a quantitative consultant does can be referred to as **providing “services” to “clients”**. BD is crucial for consultants: it allows them to be (and to remain) employed. Despite this, quantitative consultants are not usually very fond of BD, often feeling that this task is beneath them.⁵²

52: This could be a gross generalization, but we cannot find any other reasonable explanation for the reticence of quantitative folks to engage in BD.

Drumming up business is not a waste of time – pragmatically, efficient BD leads to **more time** for research, development, and quantitative analysis. A better understanding of clients and their motivations is essential to design a better BD plan, which hopefully turns into **satisfied clients**,⁵³ which hopefully turns into **repeat clients**.

53: The analytical work still must be conducted properly, however!

And when consultants do not have to worry about where their next project is coming from, they can focus their mental energy and efforts on **offering good services**.

13.4.2 Clients and Choices

Unless consultants have also been on the client side, they may not understand what drives **client choices**.

The Client Experience

From the client's perspective, a consulting project is a **risky** endeavour. There is a level of **personal risk**: they are putting their affairs in someone else's hands and are relinquishing control over the analytical process (even if they are brought in as domain experts).

There is an axis of **insecurity**: clients may wonder whether the consultant really wants to help them or is just out to help themselves, or whether the consultant will make the problem more complex than it really is (based on past experience with consultants and/or academics). Finally, the client may be **skeptical**, having been "burned" by consultants before.

They may be concerned that the consultant will not keep them informed, will be hard to reach, or will lose interest in the problem. From the client's perspective, buying professional services is not usually a pleasant experience – they would rather be buying **solutions to their problems** rather than buying a consultant's **time**.

In an employer/employee relationship, it might seem at first glance that the opposite is true: the employee's time is in fact what is purchased by the employer. But that attitude is slowly changing for quantitative workers, especially in the post-pandemic workplace.

So how do clients choose a service provider?

The Client's Choice Process

Part of the difficulty is that qualified quantitative consultants are commonplace – unless their skills are **truly** unmatched by competitors, professionals are rarely hired because of their technical capabilities.

Excellent quantitative capabilities are required to be **considered** in the first place, but it is other things that get a consultant selected – maintaining long-term business is more about **relationships** than it is about consultants' technical proficiency. Among the set of qualified candidates, clients seek the ones they can **trust**.⁵⁴

54: Crucially, this is a 2-way street: consultants also should be seeking clients they can trust.

13.4.3 Building Trust

Trust is a necessary (but not sufficient!) requirement to successful consulting projects.

The Trust "Equation"

Trust is built using several factors:

- credibility, reliability, and intimacy (all positive), and
- self-orientation (negative).

The relationship is sometimes expressed *via* the **trust "equation"**:

$$\text{Trust} = \frac{\text{Credibility} + \text{Reliability} + \text{Intimacy}}{\text{Self-Orientation}}.$$

- **Credibility** refers to the consultant's technical expertise and ability to project confidence in the latter in the client's mind;
- **reliability**, to dependability and consistency on the consultant's part (work done well and on time);
- **intimacy**, to the idea that business relationships require awareness of mutually increasing risk (clients and consultants are in this together), and
- **self-orientation**, to advisors who appear to be more interested in themselves than in the client.

Credibility

Consultants commonly achieve this component *via* qualifications and references, by presenting themselves in a professional manner, and by being accurate, precise, and complete in their work.

In general, it is not obviously clear that clients can distinguish **outstanding** work from merely **competent** work, unless they are themselves experts in the field. Most clients who leave a business relationship with a quantitative consultant do not do so because of technical incompetence, but due to **small dissatisfactions with the service**.

Even sophisticated clients will come to focus on the **quality of service** rather than the quality of work. This can sometimes be baffling to recent quantitative graduates, but it is like the notion that **how we say things** matters just as much (if not more) than **what we say**, in many contexts.

Reliability

As the number of interactions with the client increases, reliability can be demonstrated with **consistent** consultant behaviour:

on time + on spec⁵⁵ + on budget + "extra" touches.⁵⁶

55: Consultants providing what has been agreed upon.

56: As long as these originate with the consultant; when it is the client that asks for more, then there is the danger of scope creep.

Why does this prove important? Marketing is painful and not usually that effective, in the final analysis; great customer service is probably the **most effective** and **least expensive** marketing strategy.

And this does not apply only to repeat clients; current clients can serve as references for future projects – the least they should be able to say about a consultant is that they are **reliable**.

Intimacy

According to experts, "lack of intimacy" is a common **failure in building trust**. Mutual increasing risk brings clients and consultants together; **candor** and **honesty** are crucial.

These experts also claim that consultants should aim to become clients' "friends" and confidants, but we feel less confident about that: there are power dynamics at play, and the potential for abuse exists.

We will have more to say on intimacy in the next section.

Self-Orientation

Quantitative consultants should not appear to be more interested in themselves than in their clients; self-orientation is the greatest source of client distrust. They say that the client is always right, even when they are not right... within reason.

In practice, this translates to consultants never telling the clients flat-out that **they are wrong** without offering them **alternatives** or a **way out**.⁵⁷ In the consulting world, what this really means is that the clients' **true needs** should come before the consultants' "desire to create a monument to their own technical ability". Clients will smell that something is off if the consultant is just out to pad their CV.

The best way to appear interested in the client's project is to **BE** interested in the client's project, and that is best accomplished by taking on projects that are interesting to the consultants.

57: Flexibility is the consultant's ally, however: there are instances where it makes more sense for the consultant to walk away (subject to contractual obligations, of course).

Back to Business Development

People who know a lot more about BD than we do estimate that it is "5-10 times more profitable to sell new services to an existing customer than to sell a first service to a new customer, and that it is at least 5 times as expensive to get a new client than to keep an old one" [15, 7, 16].

Let us leave the numbers aside for now: the main advantage of working with old clients is that consultants do not need to build **trust** with them, and they can move on to the quantitative part of the project sooner (while maintaining the trust, obviously).

Furthermore, "the average sale is made after the 6th contact, but the average person quits after 2nd" [15, 7, 16]. While consultants need to be able to take no for an answer,⁵⁸ they also need to realize that they do not need to make a pitch on the very first contact; contacts can be used to build trust.

58: Again with the dating analogy.

The BD order of priority should be as shown below.

	Existing clients	New clients
Aware of a new need	1	3
Not aware of a new need	2	4

In general, it is easier to sell to an existing client that is not yet aware of a new need than it is to sell to a new client that is aware of their needs. The same experts also claim that "at least 70% of your business should be from past clients and their referrals; [...] very profitable firms often reach 90%".

If the working relationship with the current client is great, this is where consultants should focus their efforts **first**. The key to efficient BD is to **deliver on existing projects** and then **stay connected** with the client, as future work can usually follow with lowered effort levels.

If existing clients do not need a consultant's services anymore (or in the near future), client referrals are the next option – they may know someone in their network who could use such services. Trust must still be built in these cases, but the process has been **jump-started**.

Having said all that, there is such a thing as **client fatigue**; consulting should not become a **prison**. If consultants do not like working with a client on their project (for whatever reason), they do not have to return to them indefinitely.

Most clients are reasonable and will accept a **professionally**-handled "break-up," but some will try to pressure the consultants to return against their will.⁵⁹ Consequently, consultants might benefit from having an **exit strategy**.

59: Do we even need to name the analogy?

13.4.4 Improving Trust

Serious consultants should always be seeking to **improve** the components of the Trust Equation. **Credibility** is often more important with **new clients** – presumably, repeat clients already find consultants credible. **Reliability** and **intimacy** can be improved both with old and with new clients.

Improving Credibility

Consultants can improve credibility by:

- pointing to **publications** (peer-reviewed research papers and white papers, etc.) or to **academic honours and teaching**;
- preparing **peerless marketing materials** (current and customizable project-based CVs, client testimonials, portfolio, updated and functional website, blog articles on variety of topics, (e-)brochures, business cards, social media presence, etc.), and
- **managing client expectations**: clients are satisfied when consultants deliver more than they were expecting.

Consultants should try to strike the **right balance** – prospective clients could get scared by a portfolio on steroids. Not every detail of a consultant's history needs to be publicly available; it is easy to provide clients with more information on demand. Conversely, consultants might need to spruce up their portfolio, especially early on in their career.

Improving Reliability

Consultants can improve their reliability by:

- **doing the basics** (delivering on time and on budget and solving problems instead of generating them);
- **being available** (*via* mobile and email, being proactive with status updates even when the news is not good), and
- **being responsive** (responding to questions or comments within a 24-hour window⁶⁰).

60: Unless it has already been established that the consultant is away for a longer time period, which is allowed, of course – health and family first, always!

Consultants should also be **informing “clients”** by:

- providing **timely** budget and proactive project status updates (both internal and external – one of the biggest consulting obstacles is waiting too long to let the clients know that something is not working out);
- being **organized** (preparing for meetings and taking the lead on agenda items), and
- **managing their time** intelligently (at times, it might be preferable to turn down work – it is better for consultants to deliver an A+ to a few clients rather than a C+ or an F on many projects).

For long term clients, reliability is as important as technical competency!

Improving Intimacy

Consultants should seek opportunities to push the boundaries of the relationship, to be **candid** and offer **weaknesses** – in other words, they should avoid trying to pretend that they are **perfect**.⁶¹

61: There is no sugarcoat it: **perfectionism is a flaw**.

Connections can be made by both sides looking for **commonalities**, moving beyond the small talk, and **sharing personal experiences** (at user conferences, etc.). Advisors sometimes also suggest that consultants should think of clients as their friend.

This is an area where **judgement** must be applied – the potential for abuse increases when people make themselves vulnerable. Neither **safety**, **well-being**, nor **dignity** should be sacrificed for the sake of maintaining a relationship with a client or to stay employed. Know your rights!

Reducing Self-Orientation

Consultant self-orientation may be caused by various **fears**: the fear of not knowing, of not having the right answer, of not being intelligent enough, or simply, of being rejected by the client.

It could also stem from a little streak of **selfishness** and self-consciousness, or from a need to **appear** on top of things, or from a desire to **look smart**. For clients, self-oriented consultants seem to:

- relate the client’s stories to themselves;
- finish the client’s sentences for them;
- need to appear clever, witty, bright;
- provide only indirect answers to the client’s questions;
- be unwilling to say “I don’t know”;
- recite their qualifications at inappropriate times.

Clients understand that consultants are usually looking for future projects – it is not necessary for consultants to be the star of the show. As the saying goes: “you have two ears and one mouth. Use them in that proportion”. Consultants should be listening to the client (and letting them talk), and by demonstrate knowledge and understanding of the client’s need through **good questions**.

The best way for consultants to get what they want, which is to say, to get more paying work, is to help the client with their problems.

62: We certainly do!

Recommending that clients consider using other service providers, as needed, is a particularly effective way to reduce self-orientation (and to off-load work in “boom” periods) – it could prove useful to have a list of friendly “rivals” on hand.⁶²

It is important to realize that **not every client is going to play by the rules** – a minority of clients will try to take advantage of consultants. It might sometimes feel as though “looking out for #1” is the only reasonable approach to take.

While it remains important for consultants to protect themselves, we have found, in our experience, that focusing on reducing self-orientation is more useful overall.

13.5 Technical Writing

A consulting/data science project is only as good as how the results and recommendations are communicated, no matter how clever the analysis. We do not need to turn in reports that read like *War and Peace*, say, but the writing should not hinder the conveyance of insights; technical writing may provide an acceptable path to achieve this.

The main reference for this section is [10]; additional useful references include [5, 14, 11, 23, 12, 9]. The information provided is meant to serve as a **set of guidelines**. Bend them as needed, but remain **consistent**.

13.5.1 Basics

Technical writing (TW) is communication written for and about business and industry, focusing on products and services (and policies?), and how to manufacture, market, manage, deliver, use and/or explain them. Good TW should be **precise, clear, and accurate**.

Examples of TW may include:

- CVs and résumés
- software manuals
- company websites
- instructions that come with a device
- a job description
- a falafel recipe
- help files
- code comments
- safety protocols
- official e-mails
- use cases
- case studies
- briefing notes
- research papers
- reports

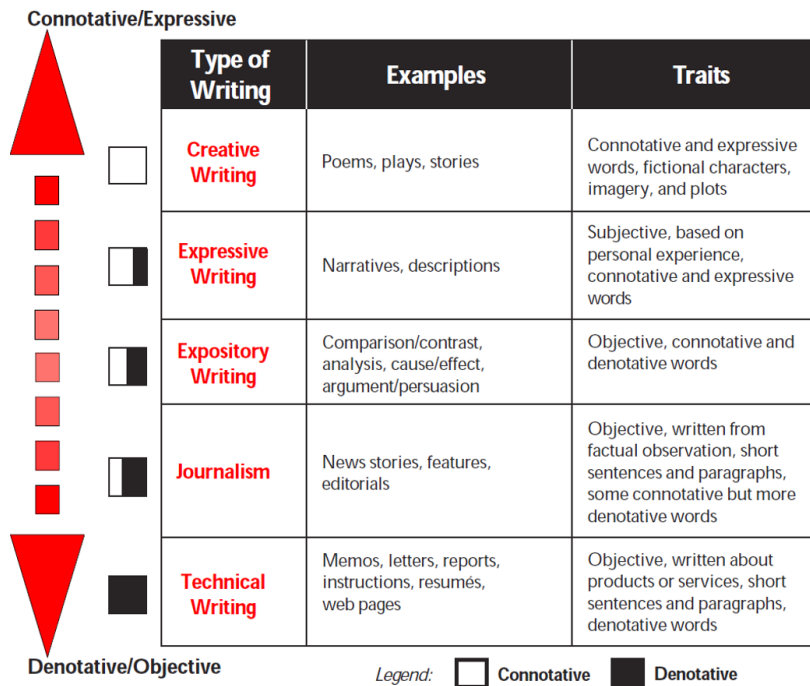


Figure 13.3: The communication continuum [10].

- storytelling dashboards
- theses
- blog articles
- etc.

TW is not prose recounting the fictional tales of characters, nor poetry which expresses deeply felt emotions through similes and metaphors; it does not narrate an occurrence/event or express an opinion; it does not report on news items; it does not focus on poetic images, nor does it describe personal experiences.

In other words, TW is neither **literature**, **journalism**, **essay writing**, nor **personal recollections**.

So what is it?

Communication Continuum

Literature is read for pleasure, essays for enlightenment, and journalism for news. TW is read to **accomplish a job**. Consequently, technical writing should be more **denotative** (provide direct definitions) than **connotative** (invoke emotional suggestions).

This **communication continuum** is illustrated in Figure 13.3.

For instance, we can compare Whitman's *When I Heard the Learn'd Astronomer* [21] (creative writing):

*When I heard the learn'd astronomer,
 When the proofs, the figures, were ranged in columns before me,
 When I was shown the charts and diagrams, to add, divide, and
 measure them,
 When I, sitting, heard the astronomer where he lectured with much
 applause in the lecture-room,*

*How soon unaccountable I became tired and sick,
Till rising and gliding out I wander'd off by myself,
In the mystical moist night-air, and from time to time,
Look'd up in perfect silence at the stars.*

to the Wikipedia definition of **astronomy** [22] (technical writing):

Astronomy (from the Greek: *αστρονομία*, literally: the science that studies the laws of the stars) is a natural science that studies celestial objects and phenomena. It uses mathematics, physics, and chemistry in order to explain their origin and evolution. Objects of interest include planets, moons, stars, nebulae, galaxies, and comets. Relevant phenomena include supernova explosions, gamma ray bursts, quasars, blazars, pulsars, and cosmic microwave background radiation. [...]

Astronomy is one of the oldest natural sciences. The early civilizations in recorded history made methodical observations of the night sky. These include the Babylonians, Greeks, Indians, Egyptians, Chinese, Maya, and many ancient Indigenous peoples of the Americas. In the past, astronomy included disciplines as diverse as astrometry, celestial navigation, observational astronomy, and the making of calendars. Nowadays, professional astronomy is often said to be the same as astrophysics.

Creative writing is “prettier” (and can be snarkier, apparently...), but technical writing conveys **precise information**. In the consulting and data science worlds, communication skills are **essential** – the best idea in the world is worthless if it cannot be communicated properly.

13.5.2 Components

TW sinks or swim based on five components (see Figure 13.4):

- **development** – preparing and presenting evidence
- **grammar** – spelling rules, syntax, conventions
- **document organization**
- **style**
- **document design** – highlighting techniques, graphs

Development

Preparing and presenting evidence should be required for all sorts of writing. TW should use examples, anecdotes, testimony, data, and research; it starts with **overall objectives**, then gets into details (items, steps, etc.), demonstrating a **logical progression** throughout.

Research often includes finding information from various sources, which should be cited when required. For quantitative writing, the **presentation of data and evidence is crucial**: TW should use paragraphs, but also charts, graphs, and tables, as necessary [5, 10].

Components	Technical Writing	Essays	Summary
Development	<ul style="list-style-type: none"> • Uses examples, anecdotes, testimony, data, research 	<ul style="list-style-type: none"> • Uses examples, anecdotes, testimony, data, research 	Same for both
Grammar	<ul style="list-style-type: none"> • It is important! 	<ul style="list-style-type: none"> • It is important! 	Same for both
Organization	<ul style="list-style-type: none"> • Provides an introduction, body, and conclusion • Uses a subject line vs. a thesis and itemization of points vs. transitional words • Uses topic sentences only when needed, dependent upon the type and length of correspondence 	<ul style="list-style-type: none"> • Provides an introduction, thesis statement, body paragraphs, transitional words, and topic sentences 	Similar in some ways, different in others
Style	<ul style="list-style-type: none"> • Uses short, denotative words; short sentences; and short paragraphs 	<ul style="list-style-type: none"> • Uses longer, connotative words; longer sentences; and longer paragraphs 	Different
Document Design	<ul style="list-style-type: none"> • Uses highlighting techniques, such as graphics, headings, subheadings, various fonts, white space, bullets, etc. 	<ul style="list-style-type: none"> • Not usually a factor 	Different

Figure 13.4: The 5 components of TW; comparison with essay writing [10].

As an example, we COULD describe how to how to put together a LEGO kit using words, but millions of kids the world over know that there is a much better approach (see Figure 13.5).

Grammar

Consulting reports and communications which do not adhere to the common spelling and syntactic rules of English⁶³ (and its conventions) might not be taken seriously by some clients.

63: Obviously, the rules differ from one language to the other.

Consultants will find the following suggestions helpful:

- always use correct grammar and spelling, no matter what language their writing in – mistakes undermine what they're trying to say;
- look it up or get help from someone who knows when in doubt;
- data scientists should use the second person and talk directly to the client and/or reader;
- avoid slang, dawg!;
- not be, like, real informal;
- explain acronyms: there are many possible meanings for most TLAs.

Compare with the following list of second person suggestions:

- always use **correct grammar** and **spelling**, no matter the language used – mistakes undermine what you are trying to say;
- when in doubt, **look it up** or get help from someone who knows;
- use the second person; talk directly to your client and/or reader;⁶⁴
- avoid slang;
- do not be informal;
- explain acronyms: there are many possible meanings for most Three Letter Acronyms (TLAs).

64: Although some writers (ahem!) think that a (semi-)consistent use of person and voice is more important.

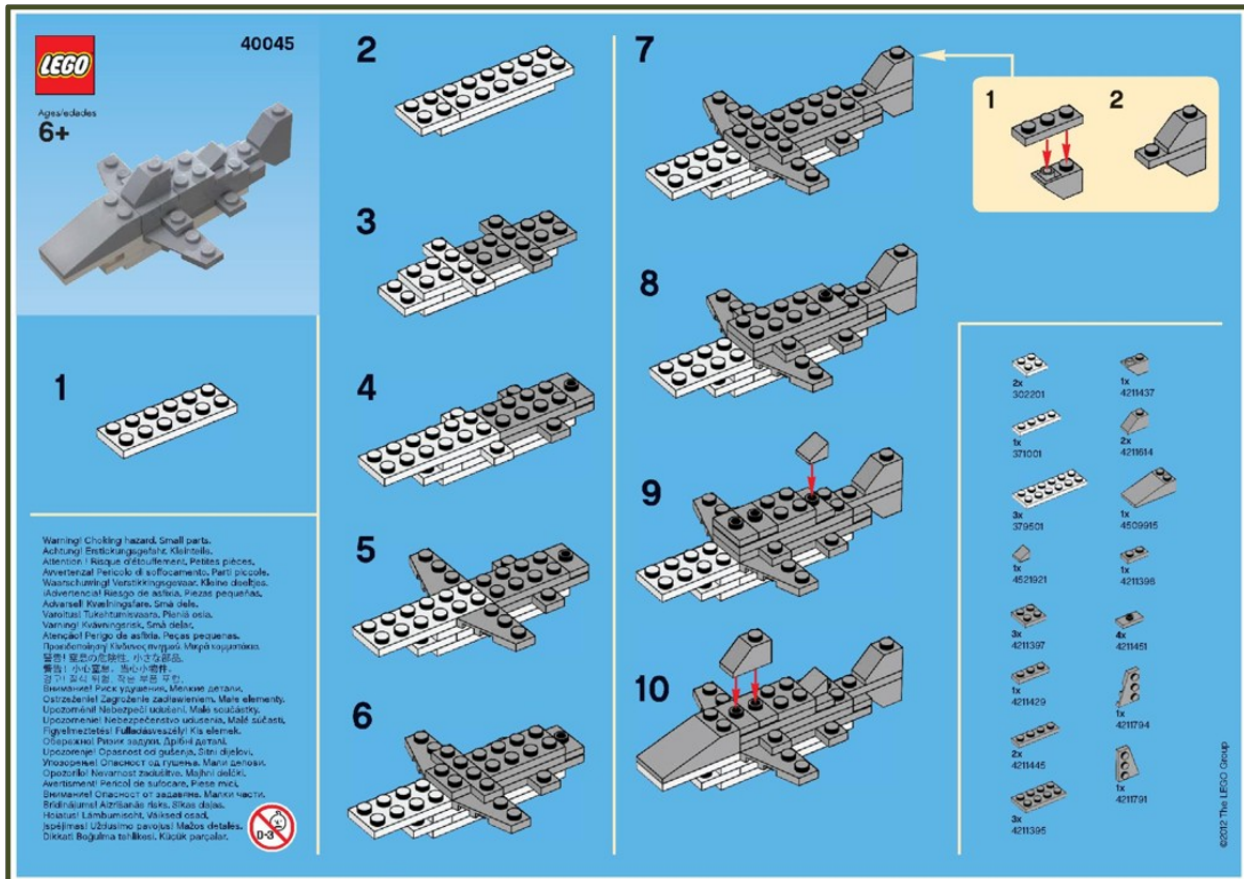


Figure 13.5: Build-a-shark LEGO instructions, with **objective** (what the assembled kit should look like), **steps** (enumerated sequence of images), and **requirements** (quantities for each piece) [LEGO].

Other suggestions/guidelines include:

- using spell-checkers wisely;
- avoiding tenses shift in the middle of a sentence;
- writing sentences with a subject and a predicate (see [19] for definitions and examples);
- avoiding run-on sentences – this makes them hard to understand;
- making the antecedents of pronouns clear (see [24] for definitions and examples);
- using correct punctuation: periods (‘.’) end sentences and commas (‘,’) separate dependent clauses;
- putting punctuation and footnotes inside quotation marks and parentheses (we are not a fan of this one, so we chose to ignore it);
- minimizing the use of semicolons (they tend to complicate TW);⁶⁵
- not using apostrophes to form a plural (“Lend me your CD’s!” is bad; “Lend me your CDs!” or “Lend me your CD” is better).

65: Ironical, we know.

Writing technical English is not easy, especially for those of us for whom English is not a first language. Most Canadian clients will recognize (and make allowances for) this reality, if the writing and grammar are consistent, but it remains to the consultant’s long-term benefit to make an effort (or to employ an editor).

Document Organization

TW does not **usually** employ

- **topic sentences** (sentence summarizing the paragraph);
- **transitions** between and within paragraphs, and
- **thesis statements** (abstracts or summaries).

In a memo or a letter, the thesis statement is usually replaced by a **subject line**. TW uses **short paragraphs** (units of text consisting of a small number of sentences expressing a single idea, with support).

Transitional words and phrases can be replaced by:

- **enumerated lists**;
- **bullet lists**, and/or
- **headings** and **subheadings**.

TW should contain **sections**, each consisting of an **introduction**, a **body**, and a **conclusion**; the most useful, general information should go first, and it should be followed with the required details.

Style

In general, TW should use

- **short, denotative words**;
- **short, simple sentences**; and
- **short paragraphs with charts** (as required).

From a stylistic perspective, the focus should be on the **audience** and on the **purpose**.⁶⁶

It is important to remember that TW readers do not necessarily have an interest in the subject matter itself. Nobody reads a microwave oven's instructions for pleasure, say – TW is simply a means to an end.

Consider the following scenario: late at night on a deserted country road in the Winter, a driver hits a pothole and realizes that one of his tires has been perforated. He has never changed a tire in his entire life. Would the instructions on the Subaru website help him to do so?

Equipment necessary to change the tire are a lift jack and a wrench. Use the jack to lift the vehicle and pick the tire up off the ground. Then use the wrench to loosen the lug nuts on the wheel. Once all the lug nuts are loose, remove them one by one and keep them in a safe place nearby. After the lug nuts are removed, the wheel and tire can be removed from the car. If a spare wheel is being put in its place, locate the spare wheel under the flooring of the trunk area, and take it out. Place the spare tire onto the lug bolts, and repeat the removal process in reverse order. Start by screwing on each lug nut, and then once all the lug nuts are screwed on, use the wrench to tighten the wheel to the disc plate. After all the lug nuts are fully tightened, disengage the jack to bring the car back to the ground. Do not exceed 50 miles per hour

66: There are parallels with fashion and gastronomy: sometimes we need to wear a fancy suit for a special holiday meal, sometimes we need a t-shirt and jeans for a poutine.

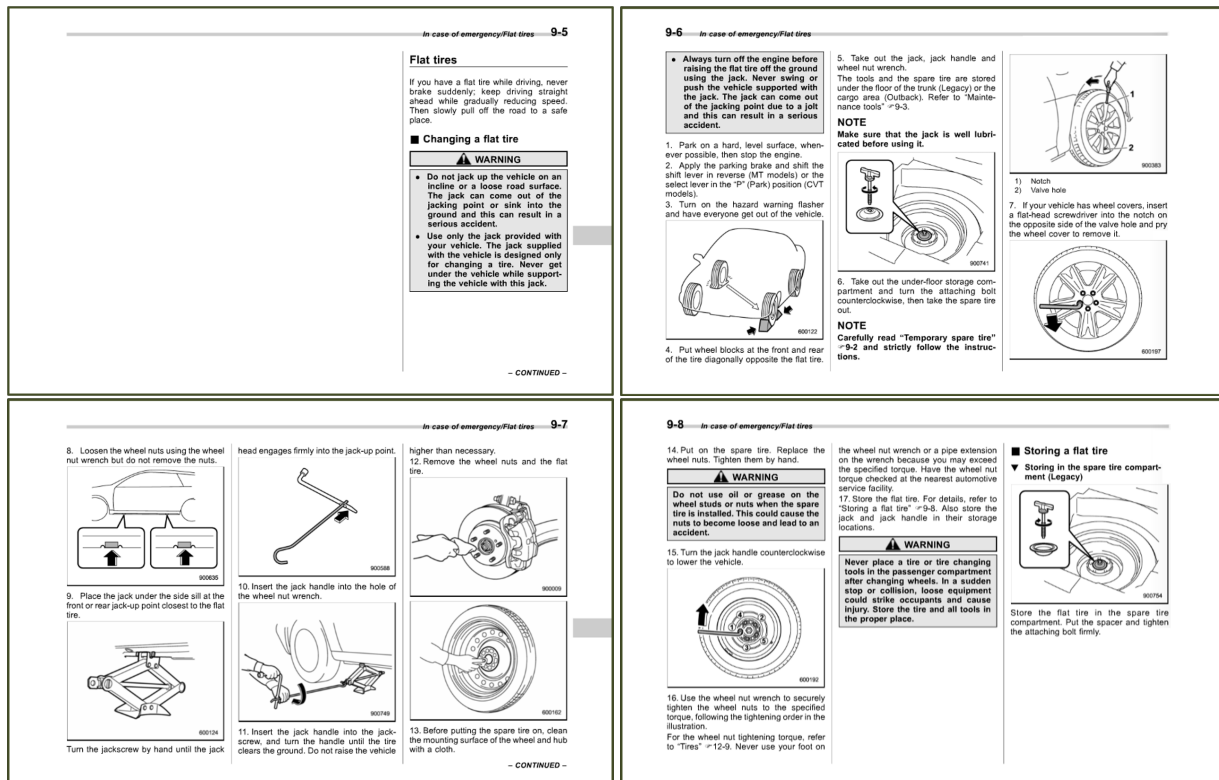


Figure 13.6: Changing a flat tire, Subaru Outback 2016 user manual [Subaru].

using a spare and changing the spare back to a standard tire as soon as possible. [Subaru.com]

Document Design

Document design refers to the **physical layout** of the correspondence. In general, TW uses **highlighting techniques**, such as graphics, lists (numbers, bullets), headings, and sectioning, but "less is more".

A small **number of different FONTS, colours, and accents** (**bold**, *italics*, underline) can help – BUT DON'T OVERDO it!

For sequential instructions, **numbered lists** are recommended. Longer documents could include a **table of contents** and an **index**, as well as lists of figures and tables; **hyperlinks** can be used for online documents. In all cases, clipart and low-resolution images should be **avoided**.

Let us revisit the flat tire example from above. Was the information provided precise? Did it get the message across? Was it understandable? What, if anything, is it missing? The user manual's tire changing instructions are shown in Figure 13.6. Which approach works best?

13.5.3 Traits

Sound TW exhibits five traits:

- **clarity** (organization);

- **conciseness** (fluency / choice);
- **accessible document design** (ideas and content);
- **audience recognition** (voice), and
- **accuracy** (writing conventions).

Let us discuss these traits one by one.

Clarity

The memo below is an example of unclear writing.

From: Manager Untel
To: New Employee Smith
Subject: Meeting

Please plan to prepare a presentation on sales. Make sure the information is detailed. Thanks.

What don't we know in this memo? What should have been included for clarity?

The **journalist's questions** (6 Ws) can help clarify communications:

- **When's** the meeting?
- **Where's** the meeting?
- **Who's** the meeting for?
- **Why** is this meeting being held?
- **What** does the manager want to be conveyed about sales?
- **How** much information is "very detailed"?
- **How** will the presentation be made?

The same memo can be made much clearer, as below.

From: Manager Untel
To: New Employee Smith
Subject: Sales Staff Meeting

Please make a presentation on improved sales techniques for our sales staff. The meeting is planned for March 28, 2017, in Room 23, from 7:00am - 6:00pm.

Our quarterly sales are down 27%. We need to help our staff accomplish the following:

1. Make new contacts.
2. Close deals more effectively.
3. Earn a 25% profit margin on all sales.

Use the new multimedia presentation system to give your talk. With your help, I know our company can get back on track.

Thanks

Clarity is the **most important criteria** for effective TW. Without it, the reader will either need to contact the writer for further clarification, or just ignore the information: the writer's and reader's time is wasted, and the message is lost.

As another example, consider a furnace maintenance safety manual. If the writing is not clear and the reader fails to understand the content, we might encounter the following consequences:

- **bad** – the furnace is damaged. The company replaces the furnace, costs accrue, and public relations have been frayed;
- **bad** – the company is sued and loses money, the writer gets fired;
- **worse** – someone gets hurt (pain, anxiety, hospital bills, etc.).

In a more general context, the 6 Ws should address the following items.

- **Who** is the audience? Are they beginners or experts?
- **What** do we want the audience to know or do?
- **When** will the work/event occur, in what order?
- **Where** will the work/event occur?
- **How** should the tasks be performed?
- **Why** is this information important?

It is preferable to **avoid imprecise words**, such as many, few, short, often, recently, thin, etc., and to use **precise words** and **terminology** instead:

“Don't block the user interface thread for more than 2 secs.”

“Use four inches of 26-gauge black wire.”

Another good suggestion is to **front-load** sentences with vital information:

“Unfortunately, your program has timed out.”

“Network connection unavailable. Call 5555 for support.”

Conciseness

Consider the 1980's referendum question:

The Government of Québec has made public its proposal to negotiate a new agreement with the rest of Canada, based on the equality of nations; this agreement would enable Québec to acquire the exclusive power to make its laws, levy its taxes and establish relations abroad – in other words, sovereignty – and at the same time to maintain with Canada an economic association including a common currency; any change in political status resulting from these negotiations will only be implemented with popular approval through another referendum; on these terms, do you give the Government of Québec the mandate to negotiate the proposed agreement between Québec and Canada?

How easy is it to understand the question? To remember what was read? How many of you even finished reading it? How does it compare to:

“Do you want Québec to be independent?”

Part Number 315564-00				Achieving Audience Recognition		
Wafer #	Quantity Received	Accepted	Rejected	Audience	Style	Example
3206-2	541		X	High Tech Peers	Abbreviations/ Acronyms OK	Please review the enclosed OP and EN .
3206-4	643	✓ <input type="checkbox"/>		Low Tech Peers	Abbreviations/ Acronyms need parenthetical definitions.	Please review the enclosed OP (Operating Procedure) and EN (Engineering Notice).
3206-5	329	✓ <input type="checkbox"/>		Lay Readers	No abbreviations/ acronyms. Explanations instead.	By following the enclosed operating procedure, you can ensure that your printer will run to our engineers' desired performance levels.
3206-6	344	✓ <input type="checkbox"/>				
3206-7	143		X			
3206-8	906		X			

Figure 13.7: Accessible description: part number 315564-000 (left); audience recognition concepts (right) [10].

Text is **concise** when it says much with few words. The idea is to keep everything short and to the point. Conciseness is important as documents must often fit in a specific physical space: a résumé being at most two pages, a car owner's manual must fit in the glove compartment, etc.

English is concise when it **avoids the passive voice**: compare “Approximately 2000 records per minute are processed by the system” (10) with “The system processes approximately 2000 records per minute” (8).⁶⁷

67: Technical writing in French is hampered by the language's definite avoidance of conciseness.

Accessible Document Design

Consider the following paragraph:

Regarding part number 315564-000, we received 541 units of wafer #3206-2. These were rejected. For the same part number, we received 643 units of wafer #3206-4. These were accepted. Three hundred and twenty-nine units of wafer #3206-5 from the same part number. These were accepted. Next, 344 of part number 315564-000's wafer #3206-6 were accepted. However, the 143 units of wafer #3206-7 (same part number) were rejected. Finally, all 906 units of wafer #3206-8 were rejected. These also were from part number 315564-00.

At a density of 8.4 words per sentence, the writing is **concise**; it is also **clear**, due to specificity of detail. But it is not intelligible.

Highlighting techniques open the text and make it inviting, while allowing for understanding and insight.

Document design refers to the physical layout of the communication (see previous sections). The document will be more **accessible** to the audience (in the sense that uninterested readers may still be able to digest it) if “walls of text” are avoided, and if **tables** are used to present information clearly (see Figure 13.7 for an accessible re-write of the passage above).

Audience Recognition

There are three kinds of TW audiences.

High-Tech Peers: readers in the same profession and at the same level as the writer (or higher);⁶⁸

Low-Tech Peers: readers who may not have the same level of expertise as the writer but who need to understand the subject;⁶⁹

Lay Readers: everybody else.⁷⁰

TW is different for each audience type. For instance:

- high-tech peers/clients can handle acronyms and abbreviations;
- low-tech peers/clients might also require parenthetical definitions;
- no acronym should be used for lay readers.

Figure 13.7 shows an accessible description of audience recognition.

Accuracy

Finally, technical writing must be **accurate**: the information it reports must be correct and representative, with no crucial information missing. Inaccuracies can create nuisances but can also be downright dangerous.

The difference between inaccuracy and **imprecision** is illustrated by the following statements: “Use 4 feet of 3/8-inch rebar” when the requirement is for 1/2-inch rebar is **innaccurate**, but “Use 4 feet of rebar” is **imprecise** as it does not specify the diameter, so the builder might not be sure.

Writers use various tricks to help with accuracy, such as:

- finishing the writing, letting it sit, then re-reading to see what might have been left out or gotten wrong;
- have someone else read it;
- reading it aloud slowly;
- reading it backwards, or upside-down,
- etc.

13.5.4 Examples

In the following pages, we provide an example of what TW could look like in a quantitative context, being a readable executive summary of the analysis of the algae blooms dataset conducted in Section 15.7 and 20.6, based on an original case study by Torgo [20].⁷¹

It is fair to say that a reasonably high number of quantitative workers got into the field to avoid writing in the first place, in the hopes that numbers and charts could carry the conversation. With an ever increasing focus on globalization, a large number of us also end up having to work in a second or even a third language, in which writing does not come naturally.⁷²

Be that as it may, being able to convey actionable insights to clients and stakeholders in a clear manner is fast becoming a **must** for quantitative consultants and data scientists alike.⁷³

68: Such as an email to a counterpart in another company.

69: Such as a summary of a software design document written for a manager.

70: Such as a list of a medication’s side-effects, written for a patient.

71: It is obviously directed at a technical audience, but it does consider the Multiple I’s as it weaves its narrative.

72: Tools like Grammarly (syntax) and DeepL (translation) can help, although they cannot be the “be all and end all” of writing endeavours... and be weary of the lure of Chat-GPT!

73: We will give you a final tip, which is often offered to beginners: **writers must first and foremost be readers**. Read voraciously, read technical papers and *comptes rendus*, read fiction and non-fiction, read in another language. There are tons of quantitative writers on the Internet; they can provide a baseline on which to build.

Workflow: Predicting Algae Blooms

PROBLEM DESCRIPTION

The ability to monitor and perform early forecasts of various river algae blooms is crucial to control the ecological harm they can cause. The dataset which is used to train the learning model consists of:

- chemical properties of various water samples of European rivers
- the quantity of seven algae in each of the samples, and
- the characteristics of the collection process for each sample.

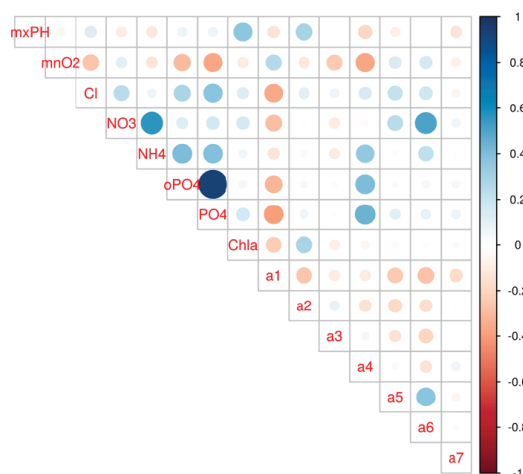
What is the data science motivation for such a model? After all, we can analyze water samples to determine if various harmful algae are present or absent. Chemical monitoring is cheap and easy to automate, whereas biological analysis of samples is expensive and slow. Another answer is that analyzing the samples for harmful content does not provide a better understanding of what drives the production of algae: it just tells us which samples contain algae.

The algae blooms dataset has 338 observations of 18 variables each: *season*, *size*, *speed*, *mxPH*, *mnO2*, *Cl*, *NO3*, *NH4*, *oPO4*, *PO4*, *Chla*, *a1*, *a2*, *a3*, *a4*, *a5*, *a6*, *a7*.

- 3 of the fields are categorical (*season*, *size*, *speed*, which refer to the data collection process)
- of the numerical fields, 8 have names that sound vaguely "chemical"
- the remaining fields refer to various algae blooms

We can get a better feel for the data frame by observing it as an array (first 6 rows):

season	size	speed	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla	a1	a2	a3	a4	a5	a6	a7
winter	small	medium	8.00	9.8	60.800	6.238	578.000	105.000	170.000	50.0	0.0	0.0	0.0	0.0	34.2	8.3	0.0
spring	small	medium	8.35	8.0	57.750	1.288	370.000	428.750	558.750	1.3	1.4	7.6	4.8	1.9	6.7	0.0	2.1
autumn	small	medium	8.10	11.4	40.020	5.330	346.667	125.667	187.057	15.6	3.3	53.6	1.9	0.0	0.0	0.0	9.7
spring	small	medium	8.07	4.8	77.364	2.302	98.182	61.182	138.700	1.4	3.1	41.0	18.9	0.0	1.4	0.0	1.4
autumn	small	medium	8.06	9.0	55.350	10.416	233.700	58.222	97.580	10.5	9.2	2.9	7.5	0.0	7.5	4.1	1.0
winter	small	high	8.25	13.1	65.750	9.248	430.000	18.250	56.667	28.4	15.1	14.6	1.4	0.0	22.5	12.6	2.9



A portrait of the relationships between the variables is provided by the correlogram on the left (for the numerical variables).

For now, we assume that the dataset has been properly explored and understood, and that any problems related to invalid data (outliers, etc.) have been solved.

PREDICTION MODELS

Our goal is to build a predictive model for the various algae blooms *a1* – *a7*. It is a supervised learning task; in order to mitigate overfitting (a consequence of the bias-variance trade-off), we set aside a test set on which the models (which will be learned on the training set) are evaluated. We use a 65%-35% split (218 – 120 randomly selected training/test observations).

GENERALIZED LINEAR MODEL

As a baseline model, we run a linear model to predict $a2$, for example, against all the predictor variables, but using only the training set as data. The results are summarized below.

```
Residuals:
    Min       1Q   Median       3Q      Max
-17.436   -5.281   -2.613    2.026   62.712

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.083e+01 1.257e+01  -2.452 0.015056 *
seasonsummer -1.166e-01 2.112e+00  -0.055 0.956035
seasonautumn  1.071e+00 2.370e+00   0.452 0.651934
seasonwinter -1.451e+00 2.000e+00  -0.726 0.468935
sizemedium   -2.628e+00 1.895e+00  -1.387 0.166896
sizelarge    -3.210e+00 2.412e+00  -1.331 0.184767
speedmedium  3.887e+00 2.485e+00   1.564 0.119325
speedhigh   -1.104e+00 2.772e+00  -0.398 0.690751
mxPH         4.859e+00 1.559e+00   3.117 0.002092 **
mnO2        -1.841e-01 3.924e-01  -0.469 0.639474
Cl           -7.432e-03 2.006e-02  -0.371 0.711351
NO3          2.132e-01 3.028e-01   0.704 0.482249
NH4         -5.979e-04 5.355e-04  -1.117 0.265510
oPO4         2.290e-03 9.876e-03   0.232 0.816875
PO4         -1.559e-03 5.936e-03  -0.263 0.793090
Chla         1.652e-01 4.614e-02   3.579 0.000432 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.74 on 202 degrees of freedom
Multiple R-squared:  0.206,    Adjusted R-squared:  0.147
F-statistic: 3.493 on 15 and 202 DF,  p-value: 2.498e-05
```

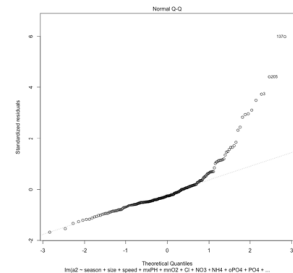
We see that the adjusted R^2 coefficient is fairly small. Furthermore, if the linear model is a good fit, the residuals should have a mean of zero and be "small", which doesn't seem to be the case (at least, relative to the range of $a2$, see 6-pt summary to the right).

The normal QQ-plot for the residuals (see figure on the right), in particular, seem to indicate that linearity of the data is probably not met, as an assumption.

On the other hand, the F-statistic seems to indicate some (linear) dependence on the predictor variables.

$a2$

Min.	: 0.000
1st Qu.	: 0.000
Median	: 2.800
Mean	: 7.207
3rd Qu.	: 10.025
Max.	: 72.600



Backward elimination stepwise selection suggests that the best linear model for $a2$ involves *speed*, *mxPH*, and *Chla*.

```
Residuals:
    Min       1Q   Median       3Q      Max
-16.195   -6.008   -2.530    2.024   63.589

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.13270 11.07921  -2.449 0.015134 *
speedmedium  4.17176  2.34330   1.780 0.076453 .
speedhigh   -0.32929  2.41899  -0.136 0.891850
mxPH         3.89794  1.35358   2.880 0.004387 **
Chla         0.15945  0.04387   3.635 0.000349 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.58 on 213 degrees of freedom
Multiple R-squared:  0.1874,    Adjusted R-squared:  0.1721
F-statistic: 12.28 on 4 and 213 DF,  p-value: 5.289e-09
```

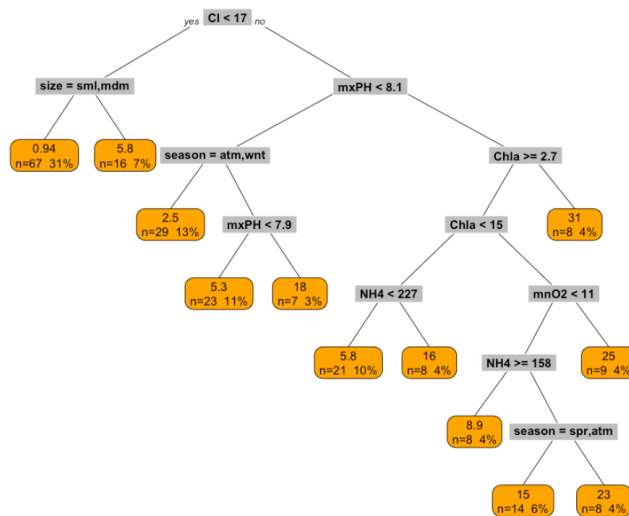
The fit is still not ideal (the value of the adjusted R^2 is quite small).

REGRESSION TREE MODEL

An alternative to regression is the use of regression trees. A recursive partition tree for $a2$ is shown below, as is a pruned tree, with the relative importance of the variables for both models:

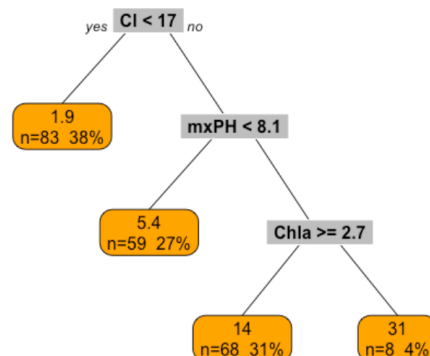
Variable importance

Chla	NH4	Cl	mxPH	oPO4	PO4	NO3	speed	mnO2	season	size
19	14	14	13	11	9	6	5	4	3	2



Variable importance

Chla	Cl	NH4	mxPH	oPO4	PO4	speed	NO3	mnO2
19	18	14	13	12	11	7	5	2



MODEL EVALUATION

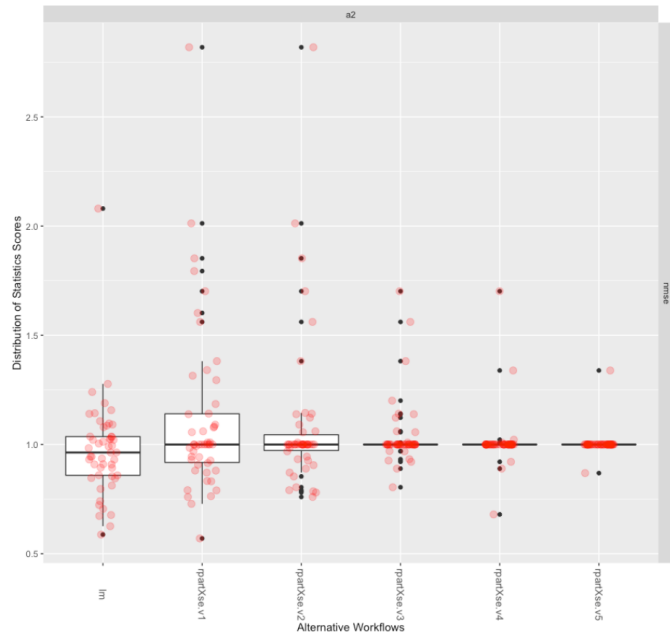
At this stage, we know that the linear model is not great for a_2 , and we have grown regression trees for a_2 but we have not yet discussed whether these models are good fits for a_2 , to say nothing of the remaining 6 algae concentrations.

Various metrics can be used to determine how the predicted values on the test set compare to the actual values: we will use the normalized mean squared error (NMSE). NMSE is unitless: values between 0 and 1 indicate that the model performs better than the baseline; values greater than 1 indicate that the model's performance is sub-par.

The test NMSE for the linear model and for a family of regression tree models (one for 5 different values of a growth/pruning parameter) is estimated using 5 repetitions of 10-fold cross-validation. For each model, the results for the 50 cross-validated models are shown in the image to the right. Summaries for the 50 models for each approach are found below.

lm	nmse	rpartXse.v1	nmse	rpartXse.v2	nmse
avg	0.9880781	avg	1.0333720	avg	1.0596868
std	0.3682616	std	0.3406970	std	0.3147441
med	0.9470239	med	1.0000000	med	1.0000000
iqr	0.2817843	iqr	0.1842643	iqr	0.0435684
min	0.4869917	min	0.6171205	min	0.5049684
max	2.5236216	max	2.4535376	max	2.4535376
rpartXse.v3	nmse	rpartXse.v4	nmse	rpartXse.v5	nmse
avg	1.028517	avg	1.012748	avg	1.001631
std	0.230181	std	0.078035	std	0.011533
med	1.000000	med	1.000000	med	1.000000
iqr	0.000000	iqr	0.000000	iqr	0.000000
min	0.528342	min	0.819828	min	1.000000
max	2.365684	max	1.413850	max	1.081548

Cross Validation Performance Estimation Results

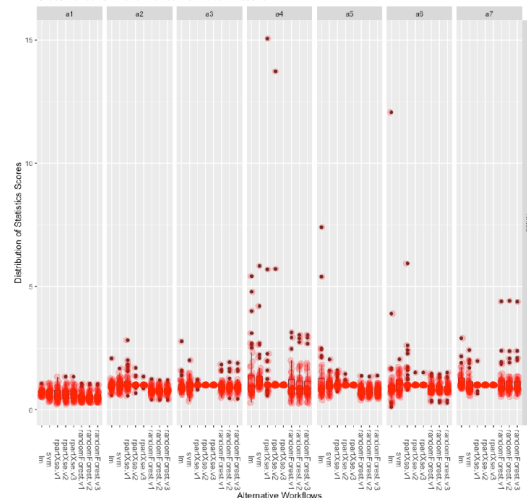


It's not necessarily clear which of the models has smaller values of NMSE overall, although it does seem that the latter versions of the regression tree models are not substantially better than the baseline model. The first regression tree model sometimes produces very small NMSE values, but that's offset by some of the larger values it also produces (similarly for the linear model). At any rate, visual evidence seems to suggest that the linear model is the best predictive model for a_2 given the training data.

This might seem disheartening at first given how poorly the linear model performed, but it is helpful to remember that there is no guarantee that a decent predictive model even exists. Furthermore, regression trees and linear models are only two of a whole collection of possible models. How do support vector regression or random forests models perform, for instance?

We repeat the task of estimating test NMSE *via* 5 replicates of 10-fold cross-validation for 8 models (linear regression, support vector regression, 3 regression trees, 3 random forests) for all target variables ($a_1 - a_7$) simultaneously. We are not looking for a single model which will optimize all learning tasks at once, but rather that we can prepare and evaluate the models for each target variable with the same bit of code. The results are shown in the figure to the right. The top performers (average value of NMSE) for each response are shown on the next page.

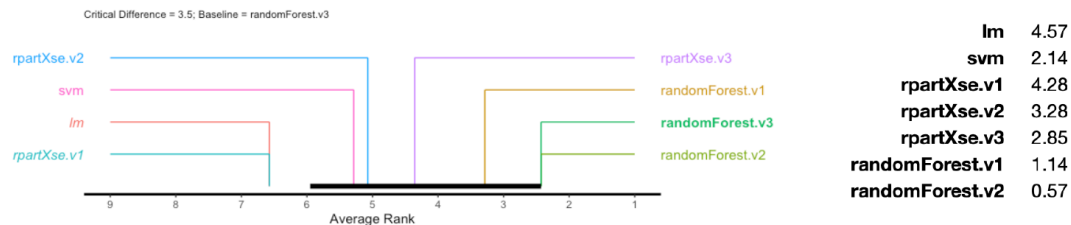
Cross Validation Performance Estimation Results



Rank.a1	model	est.nmse	Rank.a2	model	est.nmse	Rank.a3	model	est.nmse	Rank.a4	model	est.nmse
1	randomForest.v2	0.5217204	1	randomForest.v3	0.7798749	1	randomForest.v3	0.9377108	1	rpartXse.v3	1.001453
2	randomForest.v3	0.5228744	2	randomForest.v2	0.7806831	2	randomForest.v2	0.9400108	2	randomForest.v3	1.006496
3	randomForest.v1	0.5264328	3	randomForest.v1	0.7849360	3	randomForest.v1	0.9431801	3	randomForest.v1	1.006806
Rank.a5	model	est.nmse	Rank.a6	model	est.nmse	Rank.a7	model	est.nmse			
1	randomForest.v1	0.7626241	1	randomForest.v2	0.8590227	1	rpartXse.v2	1.00000			
2	randomForest.v2	0.7675794	2	randomForest.v3	0.8621478	2	rpartXse.v3	1.00000			
3	randomForest.v3	0.7681834	3	randomForest.v1	0.8663869	3	rpartXse.v1	1.00797			

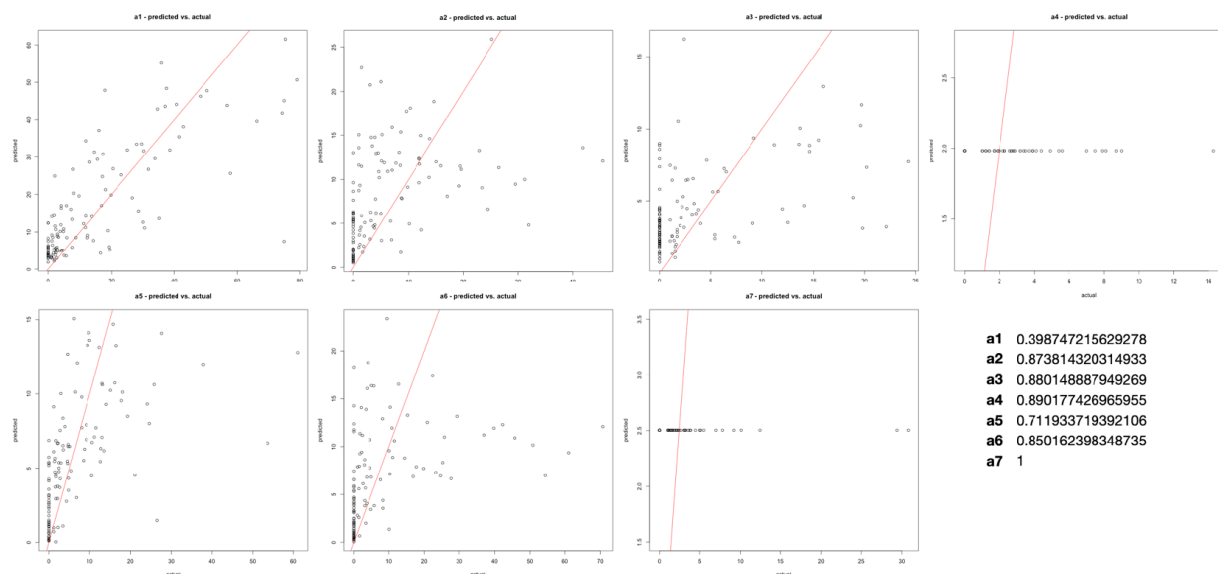
At first glance, the 3rd random forest model (the one that build predictions on 700 trees, as opposed to 200 and 500 for the other random forests models) seems to perform best, but these rankings do not report on the standard error, and so we cannot tell whether the differences between the estimated test NMSEs are statistically significant on the basis of the estimates alone.

Using the 3rd random forest model as a baseline, we compute the rank differences to the other 7 models for all target variables. The critical rank difference is 3.52. On average, the rank difference to the other models is shown in the list on the right. We can reject with 95% certainty that the performance of the baseline method is the same as that of the linear model and the first regression tree model (`rpartXse.v2`), but not that it is better than the other 5 models. The information is also displayed in the Bonferroni-Dunn CD diagram below.



MODEL PREDICTIONS

The best performer for each target response was identified from the cross-validation procedure above: for each target variable $a1 - a7$, we run the best performer on the original training data to learn a model that is used to predict the appropriate target response for observations in the original test set. Scatterplots of predicted (y-axis) vs. actual levels (x-axis) for test observations are shown below (top: $a1 - a4$, bottom: $a5 - a7$), as are the true test NMSEs.



13.6 A Conversation With ...

In 2020 and 2021, we recorded 10 conversations [↗](#) with individuals involved in quantitative consulting and data analysis, near and far.

IQC From a Student's Perspective (17:03) Smit Patel is an analyst at the Canada Border Services Agency. He is a former student in uOttawa's *Introduction to Quantitative Consulting* course. We discuss his experience as a student in the course.

Ethics in Quantitative Contexts (28:20) Julie Paquette is a professor at St-Paul University's *School of Ethics, Social Justice and Public Service*, and a Co-Director of the *Research Centre in Public Ethics and Governance*. We discuss ethics and technologies in the data context.

Multi-Tasking and Time Management (19:26) Youssouph Cissokho obtained his Ph.D. in Mathematics at the University of Ottawa in 2021. We discuss his ability to work on multiple projects simultaneously.

Consulting Experiences (26:58) Jen Schellinck is the Principal at [Sysabee](#) [↗](#) and an Adjunct Professor of Cognitive Science at Carleton University. We discuss her experience with consulting (2012–).

The Client's Point-of-View (24:17) Maryam Haghighi is the Bank of Canada's Director of Data Science. We discuss what consulting projects look like from the clients' perspective.

From Academia to the Workplace (19:37) Ying Gai is a senior analyst at the Canada Revenue Agency. We discuss her transition from academia to the “real world”.

Proposals and Budgeting (32:28) Doug Munroe is Principal and Co-Founder of [Politikos Research](#) [↗](#) (P.E.I.) and a former Professor at Quest University. We discuss writing proposals and preparing budgets.

When Projects Go “Kerplunk!” (29:09) Andrew Macfie is a former freelance software developer and consultant from Toronto. He currently works as a programmer at [Tucows](#) [↗](#). We discuss the ways in which a consulting project can go bad, and how to mitigate the risks.

Non-Technical Skills in the Workplace (20:37) Victoria Silverman is an analyst at Statistics Canada. We discuss the importance of non-technical skills in the workplace and for project work.

Setting-Up as a Consultant (22:14) Oliver Benning is a Masters Student in Computer Engineering at the University of Ottawa, and a consultant with [Strikethrough](#) [↗](#). We discuss how and why he decided to become a quantitative consultant.

13.7 Exercises

You may need to consult the videos available on [2] to answer some of the questions, a few of which are taken from [10].

1. Now that you know a little bit more about the consulting process, what are two things that appeal to you about it? What are two things that are more off-putting to you?
2. What are some ethical ideals that guide your choice of quantitative work (employment, contracts, etc.)?
3. Which of the quantitative consulting roles appeals the most to you? The least?
4. Do any of the items on the Consulting Cheat Sheet surprise you? Should anything be added to the list?
5. What does the “Tyranny of Past Success” refer to in the marketing context? How would you try to mitigate against it? Do you think this tyranny also applies in other contexts?
6. How could you address the differences between your business social media needs and personal ones?
7. When do you send a prospective client to someone else? What are the advantages of doing so?
8. What small talk would you feel comfortable doing?
9. Among the factors in favour of/against working alone or as part of a team, which ones are most compelling?
10. How long do you think team meetings should last? How frequently should they occur?
11. What might be some reasons for you to walk away from a project at the contracting stage?
12. What does work-life balance mean to you? What does your Time Management table look like? In re: Hofstadter’s Law, what do you think your factor is? believe it will, on average)?
13. Can you provide an example of a knowledge gap?
14. What are your Top 3 quantitative analysis skills? Where do you need to improve? What would you like to learn more about?
15. Why might the best academic solution not be an acceptable consulting solution?
16. What do you think are the easiest deliverables to prepare? The hardest? The most annoying?
17. Why might some consultants feel that invoicing is problematic?
18. What is the importance of the post-mortem (internal or external)?
19. What do you think are the three most important lessons about clients? About consultants? Why?
20. What is business development? Why should consultants care about business development?
21. What are client’s worries in the consulting process? What drives their choice of consultants?
22. What are the components of the trust equation? How important are they to build trust?
23. What trust equation components do you feel that you would need to improve the most? The least?
24. Write a paragraph explaining why you are taking a quantitative consulting or data analysis course. Were you precise, clear, and accurate? Is this technical writing? Does it need to be?
25. Write a rough outline (with section headers and main ideas) for a blog article on a topic of your choice. Keep in mind that the document’s organization is dependent on the target audience.
26. Revise the vague words and phrases, specifying exact information. Invent numbers and modify the rest of the sentences as needed.

- a) I have a *low* GPA.
- b) The player was *really* tall.
- c) I’ll leave *as soon as possible*.

- d) The team has a *losing* record.
- e) The phone has *lots of* memory.

27. Change the following long words to shorter words.

- a) utilize
- b) anticipate
- c) cooperate
- d) indicate

- e) initially
- f) presently
- g) prohibit
- h) inconvenience

28. Change the following phrases to one word.

- | | |
|---------------------------|-----------------------------|
| a) in the event that | f) due to the fact that |
| b) at this point in time | g) make revisions |
| c) with regard to | h) take into consideration |
| d) in the first place | i) with the exception of |
| e) is of the opinion that | j) make an adjustment to/of |

29. Revise the following long sentences, making them shorter.

- a) I will be calling you on May 31 to see if you have any questions at that time.
- b) If I can be of any assistance to you in the evaluation of this proposal, please feel free to give me a call.
- c) The company is in the process of trying to cut the cost of expenditures relating to the waste of unused office supplies.
- d) I am of the opinion that graduate students have too much work to do.
- e) In the month of July, my family will make a visit to the province of New Brunswick.
- f) It is the company's plan to take action to avoid problems with hazardous waste.
- g) On two separate occasions, the manager of personnel met with at least several different employees to ascertain whether or not they were in agreement with the company's policies regarding overtime.

30. Reformat the following text by using highlighting techniques. Consider using bullets or numbers, headings, boldface or underlining, and white space.

To make a pie chart using your word processing package's graphic components, turn on the machine. Once it has booted up, double click on the word processing icon. After the system is open, click on "graphic," scroll down to "chart," and double click. Next, click on "data chart types" and select "pie." After this, input your new data in the "data sheet." After this, click anywhere on the page to import your new pie chart. If you want to make changes, just double click again inside the pie chart; then you can revise according to your desires.

31. Make a list of 4-6 acronyms or abbreviations from an area of interest. What percentage of the audience understand your acronyms? Define / explain the terms for low-tech peers and for lay readers.
32. Describe your footwear, as accurately as possible. Without knowing the purpose of the task, how difficult is it to know how long or how specific you should be?
33. Offer a technical writing critique of this chapter (there are numerous inconsistencies).

Chapter References

- [1] P. Boily. [Techniques of Data Analysis](#) . 2020.
- [2] P. Boily. [Introduction to Quantitative Consulting \(course website\)](#) . 2021.
- [3] P. Boily. [Consulting Sample Documents](#) . 2021.
- [4] P. Boily, S. Davies, and J. Schellinck. [The Practice of Data Visualization](#) . Data Action Lab, 2023.
- [5] J. Cole. [Technical Writing Workshop](#) .
- [6] Consultancy UK. [Types of Consultants](#) .
- [7] R. Crandall. *Marketing Your Services: For People Who Hate to Sell*. McGraw-Hill, 2002.
- [8] Data Action Lab. [IQC Blog](#) . 2021.
- [9] J. Freeman. [Writing at the University of Toronto](#) .
- [10] S.M. Gerson. [A Teacher's Guide to Technical Writing](#) . Kansas Curriculum Center, Washburn University.
- [11] G.D. Gopen and J.A. Swan. 'The Science of Scientific Writing' . In: *American Scientist* 78 (Nov. 1990).
- [12] D. Hacker and N. Sommers. *The Bedford Handbook*. 9th ed. Bedford, 2013.
- [13] D. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, 1979.
- [14] M.H. Larock, J.C. Tressler, and C.E. Lewis. *Mastering Effective English*. 4th ed. Copp Clark, 1980.
- [15] D. Maister. *Managing the Professional Services Firm*. Free Press, 1993.
- [16] D. Maister, C. Green, and R. Galford. *The Trusted Advisor*. Free Press, 2001.
- [17] Project Management Institute. [PMP Reference List](#) .
- [18] Project Management Institute. [Certifications](#) .
- [19] The uOttawa Writing Centre. [The Parts of the Sentence](#) .
- [20] L. Torgo. *Data Mining with R, 2nd ed.* CRC Press, 2016.
- [21] W. Whitman. *When I Heard the Learn'd Astronomer*. 1865.
- [22] Wikipedia. [Astronomy](#) . 2021.
- [23] J.M. Williams. *Style: Ten Lessons in Clarity and Grace*. Pearson, 2004.
- [24] Your Dictionary. [What Is an Antecedent? An Explanation in Simple Terms](#) .

by Patrick Boily and Jen Schellinck

In 2012, the *Harvard Business Review* published an article calling data science the “sexiest job of the 21st century”, describing data scientists as “hybrids of data hacker, analyst, communicator, and trusted adviser” [25].

Would-be data scientists are usually introduced to the field *via* machine learning algorithms and applications. While we will discuss these topics in later chapters, we would like to start with some of the important non-technical (and semi-technical) notions that are often unfortunately swept aside in favour of diving head first into murky analysis waters.

In this chapter, we focus on some of the fundamental ideas and concepts that underlie and drive forward the discipline of data science, as well as the contexts in which these concepts are typically applied. We also highlight issues related to the ethics of practical data science. We conclude by getting a bit more concrete and considering the analytical workflow of a typical data science project, the types of roles and responsibilities that generally arise during data science projects and some basics of how to think about data, as a prelude to more technical topics.

Note: we encourage readers to take a look at Chapter 1 (*Programming Primer*) before diving into this chapter.

14.1 Introduction

The main constituent of data science is, unsurprisingly, **data**. This seems obvious, as far as statements go, but the notion of “data” is more complex than first appears.

14.1.1 What Is Data?

It is surprisingly difficult to give a clear-cut definition of **data** – we cannot even seem to agree on whether it should be used in the singular or the plural:

“the data is ” vs. “the data are ”

From a strictly linguistic point of view, a *datum* (borrowed from Latin) is “a piece of information;” **data**, then, should mean “pieces of information.” We can also think of it as a collection of “pieces of information”, and we would then use *data* to represent the whole (being potentially greater than the sum of its parts) or simply the idealized concept.

14.1 Introduction	881
What is Data?	881
Objects and Attributes . . .	883
Data in the News	883
Analog vs Digital Data . . .	884
14.2 Conceptual Frameworks . .	885
Three Modeling Strategies	886
Information Gathering . . .	887
Cognitive Biases	892
14.3 Data Ethics	893
The Need for Ethics	894
What Is/Are Ethics?	894
Ethics and Data Science . .	895
Guiding Principles	896
The Good, the Bad, the Ugly	898
14.4 Analytics Workflows	898
The “Analytical” Method .	898
Collection and Processing .	901
Model Assessment	902
Automated Data Pipeline .	903
14.5 Getting Insight From Data .	903
Asking the Right Questions	905
Structure and Organization	909
Basic Analysis Techniques .	919
Common Procedures in R .	927
Quantitative Methods . . .	940
Quantitative Fallacies . . .	944
14.6 Exercises	946
Chapter References	948

When it comes to actual data analysis, however, is the distinction really that important? Is it even clear what data is, from the definition above, and where it comes from?

Without context, does it make sense to call the following “data”?

4,529 red 25.782 Y

To paraphrase U.S. Justice Potter Stewart, while it may be hard to define what data is, “we know it when we see it.” This position may strike some of you as unsatisfying; to overcome this (sensible) objection, we will think of data simply as a collection of facts about **objects** and their **attributes**.

For instance, consider the apple and the sandwich below.



Let us say that they have the following attributes:

- *Object:* apple
 - **Shape:** spherical
 - **Colour:** red
 - **Function:** food
 - **Location:** fridge
 - **Owner:** Jen
- *Object:* sandwich
 - **Shape:** rectangle
 - **Colour:** brown
 - **Function:** food
 - **Location:** office
 - **Owner:** Pat

As long as we remember that a person or an object is not simply **the sum of its attributes**, this rough definition should not be too problematic. Note, however, that there remains some ambiguity when it comes to **measuring** (and **recording**) the attributes.

We dare say that no one has ever beheld an apple quite like the one shown above: for starters, it is a 2-dimensional representation of a 3-dimensional object. Additionally, while the overall shape of the sandwich is vaguely rectangular (as seen from above, say), it is not an exact rectangle. While no one would seriously dispute the shape attribute of the sandwich being recorded as “rectangle”, a **measurement error** has occurred.

For most analytical purposes, this error may not be significant, but it is impossible to dismiss it as such for all tasks.

More problematic might be the fact that the apple's shape attribute is given in terms of a volume, whereas the sandwich's is recorded as an area; the measurement types are **incompatible**. Similar remarks can be made about all the attributes – the function of an apple may be “food” from Jen's perspective, but from the point of view of an apple tree, that is emphatically not the case; the sandwich is definitely not uniformly “brown,” and so on.

A number of potential attributes are not even mentioned: size, weight, time, etc. Measurement errors and incomplete lists are always part of the picture, but most people would recognize that the collection of attributes does provide a reasonable **description** of the objects.

This is the **pragmatic** definition of data that we will use throughout.

14.1.2 From Objects and Attributes to Datasets

Raw data may exist in any format; we will reserve the term **dataset** to represent a collection of data that could conceivably be fed into algorithms for analytical purposes.

Often, these appear in a **table**, with rows and columns;¹ attributes are the **fields** (or columns) in such a dataset; objects are **instances** (or rows).

1: In practice, more complex **databases** are used.

Objects can then be described by their **feature vector** – the collection of attributes associated with value(s) of interest. The feature vector for a given observation is also known as its **signature**. For instance, the dataset of physical objects could contain the following items:

ID	shape	colour	function	location	owner
1	spherical	red	food	fridge	Jen
2	rectangle	brown	food	office	Pat
3	round	white	tell time	lounge	school
...

We will revisit this in Section 14.5.2 (*Structuring and Organizing Data*).

14.1.3 Data in the News

We collected a sample of headlines and article titles showcasing the growing role of **data science** (DS), **machine learning** (ML), and **artificial/augmented intelligence** (AI) in different domains of society.

While these demonstrate some of the functionality/capabilities of DS/ML/AI technologies, it is important to remain aware that new technologies are always accompanied by emerging (and not always positive) social consequences.

- “Robots are better than doctors at diagnosing some cancers, major study finds” [27]
- “Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet” [10]

- “Google AI claims 99% accuracy in metastatic breast cancer detection” [8]
- “Data scientists find connections between birth month and health” [21]
- “Scientists using GPS tracking on endangered Dhole wild dogs” [48]
- “These AI-invented paint color names are so bad they’re good” [62]
- “We tried teaching an AI to write Christmas movie plots. Hilarity ensued. Eventually.” [32]
- “Math model determines who wrote Beatles’ *In My Life*: Lennon or McCartney?” [9]
- “Scientists use Instagram data to forecast top models at New York Fashion Week” [39]
- “How big data will solve your email problem” [36]
- “Artificial intelligence better than physicists at designing quantum science experiments” [70]
- “This researcher studied 400,000 knitters and discovered what turns a hobby into a business” [75]
- “Wait, have we really wiped out 60% of animals?” [80]
- “Amazon scraps secret AI recruiting tool that showed bias against women” [24]
- “Facebook documents seized by MPs investigating privacy breach” [7]
- “Firm led by Google veterans uses A.I. to ‘nudge’ workers toward happiness” [76]
- “At Netflix, who wins when it’s Hollywood vs.the algorithm?” [60]
- “AlphaGo vanquishes world’s top Go player, marking A.I.’s superiority over human mind” [41]
- “An AI-written novella almost won a literary prize” [47]
- “Elon Musk: Artificial intelligence may spark World War III” [49]
- “A.I. hype has peaked so what’s next?” [64]
- “That Popular AI Photo App is Stealing from Human Artists – and Worse” [66]
- “Now AI can write students’ essays for them, will everyone become a cheat?” [61]

Opinions on the topic are varied – to some, DS/ML/AI provide examples of brilliant successes, while to others it is the dangerous failures that are at the forefront.

What do you think?

14.1.4 The Analog/Digital Data Dichotomy

Humans have been collecting data for a long time. In the award-winning *Against the Grain: A Deep History of the Earliest States*, J.C. Scott argues that data collection was a major enabler of the modern nation-state (he also argues that this was not necessarily beneficial to humanity at large, but this is another matter altogether) [69].

For most of the history of data collection, humans were living in what might best be called the **analogue world** – a world where our understanding was grounded in a continuous experience of **physical reality**.

Nonetheless, even in the absence of computers, our data collection activities were, arguably, the first steps taken towards a different strategy for understanding and interacting with the world. Data, by its very nature, leads us to conceptualize the world in a way that is, in some sense, **more discrete than continuous**.

By translating our experiences and observations into numbers and categories, we re-conceptualize the world into one with sharper and more definable boundaries than our raw experience might otherwise suggest. Fast-forward to the modern world and the culmination of this conceptual discretization strategy is clear to see in our adoption of the **digital computer**, which represents everything as a series of 1s and 0s.²

2: Or 'On' and 'Off', 'TRUE' and 'FALSE'.

Somewhat surprisingly, this very minimalist representational strategy has been wildly successful at **representing our physical world**, arguably beyond our most ambitious dreams, and we find ourselves now at a point where what we might call the **digital world** is taking on a reality as pervasive and important as the physical one.

Clearly, this digital world is built on top of the physical world, but very importantly, the two do not operate under the same set of rules:

- in the physical world, the default is to **forget**; in the digital world, the default is to **remember**;
- in the physical world, the default is **private**; in the digital world, the default is **public**;
- in the physical world, copying is **hard**; in the digital world, copying is **easy**.

As a result of these different rules of operation, the digital is making things that were **once hidden, visible; once veiled, transparent**. Considering data science in light of this new digital world, we might suggest that data scientists are, in essence, scientists of the **digital**, in much the same way that regular scientists are scientists of the **physical**: data scientists seek to discover the **fundamental principles of data** and understand the ways in which these fundamental principles manifest themselves in different digital phenomena.

Ultimately, however, data and the digital world are **tied to the physical world**. Consequently, what is done with data has repercussions in the physical world; and it is crucial for analysts and consultants to have a solid grasp of the fundamentals and context of data work before leaping into the tools and techniques that drive it forward.

14.2 Conceptual Frameworks for Data Work

In simple terms, we use data to represent the world. But this is not the only strategy at our disposal: we might also (and in combination) describe the world using **language**, or represent it by building **physical models**. The common thread is the more basic concept of **representation** – the idea that one object can stand in for another, and be used in its stead in order to indirectly engage with the object being represented. Humans are representational animals *par excellence*; our use of representations becomes almost undetectable to us, at times.

On some level, we do understand that “the map is not the territory”, but we do not have to make much of an effort to use the map to navigate the territory. The transition from the **representation** to the **represented** is typically quite seamless. This is arguably one of humanity’s major strengths, but in the world of data science it can also act as an Achilles’ heel, preventing analysts from working successfully with clients and project partners, and from appropriately **transferring analytical results** to the real world contexts that could benefit from them.

The best protection against these potential threats is the existence of a well thought out and explicitly described **conceptual framework**, by which we mean, in its broadest sense:

- a **specification** of which parts of the world are being represented;
- **how** they are represented;
- the **nature of the relationship** between the represented and the representing, and
- **appropriate** and **rigorous strategies** for applying the results of the analysis that is carried out in this representational framework.

It would be possible to construct such a specification from scratch, in a piecemeal fashion, for each new project, but it is worth noting that there are some overarching **modeling frameworks** that are broadly applicable to many different phenomena, which can then be moulded to fit these more specific instances.

14.2.1 Three Modeling Strategies

We suggest that there are three main not mutually exclusive **modeling strategies** that can be used to guide the specification of a phenomenon or domain:

- **mathematical** modeling;
- **computer** modeling, and
- **systems** modeling.

We start with a description of the latter as it requires, in its simplest form, no special knowledge of techniques/concepts from mathematics or computer science.

Systems Modeling

General Systems Theory was initially put forward by L. von Bertalanffy, a biologist, who felt that it should be possible to describe many **disparate** natural phenomena using a **common conceptual framework** – one which would be capable of describing many disparate phenomena, all as systems of interacting objects.

Although Bertalanffy himself presented abstracted, mathematical, descriptions of his general systems concepts, his broad strategy is relatively easily translated into a purely conceptual framework.

Within this framework, when presented with a novel domain or situation, we ask ourselves the following questions:

- which objects seem most relevant or involved in the system behaviours in which we are most interested?
- what are the properties of these objects?
- what are the behaviours (or actions) of these objects?
- what are the relationships between these objects?
- how do the relationships between objects influence their properties and behaviours?

As we find the answers to these questions about the system of interest, we start to develop a sense that we **understand the system** and its **relevant behaviours**.

By making this knowledge **explicit**, e.g. *via* diagrams and descriptions, and by sharing it amongst those with whom we are working, we can further develop a **consistent, shared understanding** of the system with which we are engaged. If this activity is carried out prior to data collection, it can ensure that the **right data** is collected.

If this activity is carried out after data collection, it can ensure that the process of **interpreting what the data represents** and how the latter should be used going forward is on solid footing.

Mathematical and Computer Modeling

The other modeling approaches come with their own general frameworks for interpreting and representing real-world phenomena and situations, separate from, but still compatible with, this systems perspective.

These disciplines have developed their own mathematical/digital (logical) worlds that are distinct from the tangible, physical world studied by chemists, biologists, and so on. These frameworks can be used to describe real-world phenomena by **drawing parallels** between the properties of objects in these different worlds and reasoning *via* these parallels.

Why these **constructed worlds** and the conceptual frameworks they provide are so effective at representing and describing the actual world, and thus allowing us to understand and manipulate it, is more of a philosophical question than a pragmatic one.

We will only note that they are **highly effective** at doing so, which provides the impetus and motivation to learn more about how these worlds operate, and how, in turn, they can provide data scientists with a means to engage with domains and systems through a powerful, rigorous and shared conceptual framework.

14.2.2 Information Gathering

The importance of achieving **contextual understanding** of a dataset cannot be over-emphasized. In the abstract we have suggested that this context can be gained by using conceptual frameworks. But more concretely, how does this understanding come about?

It can be reached through:

- **field trips**;
- interviews with **subject matter experts** (SMEs);

- readings/viewings;
- **data exploration** (even just **trying to obtain** or gain access to the data can prove a major pain),
- etc.

In general, clients or stakeholders are **not a uniform** entity – it is even conceivable that client data specialists and SMEs will **resent the involvement** of analysts (external and/or internal). Thankfully, this stage of the process provides analysts and consultants the opportunity to show that everyone is pulling in the same direction, by

- asking **meaningful** questions;
- taking an interest in the SMEs' / clients' experiences, and
- acknowledging everyone's ability to contribute.

A little tact goes a long way when it comes to information gathering.

Thinking in Systems Terms

We have already noted that a **system** is made up of **objects** with **properties** that may change over time. Within the system we perceive **actions** and **evolving properties**, leading us to think in terms of **processes**.

In order to understand how various aspects of the world interact with one another, we need to **carve out chunks** corresponding to the aspects and define their boundaries. Working with other intelligences requires this type of **shared understanding** of what is being studied. **Objects** themselves have various properties.

Natural processes generate (or destroy) objects, and may change the properties of these objects over time. We **observe**, **quantify**, and **record** particular values of these properties at particular points in time.

This process generates data points in our attempt to **capture the underlying reality** to some acceptable degree of **accuracy** and **error**, but it remains crucial for data analysts and data scientists to remember that **even the best system model only ever provides an approximation of the situation under analysis**; with some luck, experience, and foresight, these approximations might turn out to be **valid**.

Identifying Gaps in Knowledge

A **gap in knowledge** is identified when we realize that what we thought we knew about a system proves **incomplete** (or blatantly false).

This can arise as the result of a certain naïveté *vis-à-vis* the situation being modeled, but it can also be emblematic of the nature of the project under consideration: with too many moving parts and grandiose objectives, there cannot help but be knowledge gaps.³

Knowledge gaps might occur **repeatedly**, at any moment in the process:

- data **cleaning**;
- data **consolidation**;
- data **analysis**,
- even during **communication of the results** (!).

3: Note that it also happens with small, well-organized, and easily contained projects. It happens all the time, basically.

When faced with such a gap, the best approach is to be flexible: **go back**, **ask questions**, and **modify the system representation** as often as is necessary. For obvious reasons, it is preferable to catch these gaps early on in the process.

Conceptual Models

Consider the following situation: you are away on business and you forgot to hand in a very important (and urgently required) architectural drawing to your supervisor before leaving. Your office will send an intern to pick it up in your living space. How would you explain to them, by phone, how to find the document?

If the intern has previously been in your living space, if their living space is comparable to yours, or if your spouse is at home, the process may be sped up considerably, but with somebody for whom the space is new (or someone with a visual impairment, say), it is easy to see how things could get complicated.

But time is of the essence – you and the intern need to get the job done **correctly as quickly as possible**. What is your strategy?

Conceptual models are built using methodical investigation tools:

- **diagrams**;
- structured **interviews**;
- structured **descriptions**,
- etc.

Data analysts and data scientists should beware **implicit conceptual models** – they go hand-in-hand with knowledge gaps.

In our opinion, it is preferable to err on the side of “too much conceptual modeling” than the alternative (although, at some point we have to remember that every modeling exercise is wrong⁴ and that there is nothing wrong with building better models in an iterative manner, over the bones of previously-discarded simpler models).

4: “Every model is wrong; some models are useful.” *George Box*.

Roughly speaking, a **conceptual model** is a model that is not implemented as a scale-model or computer code, but one which exists only conceptually, often in the form of a diagram or verbal description of a system – boxes and arrows, mind maps, lists, definitions (see Figure 14.1, say).

Conceptual models do not necessarily attempt to capture specific behaviours, but they emphasize the **possible states** of the system: the focus is on object types, not on specific instances, with **abstraction** as the ultimate objective.

Conceptual modeling is not an exact science – it is more about making internal conceptual models **explicit** and **tangible**, and providing data analysis teams with the opportunity to **examine** and **explore** their ideas and assumptions. Attempts to formalize the concept include (see Figure 14.3):

- **Universal Modeling Language (UML)**;
- **Entity Relationship Models (ER)**, generally connected to relational databases.

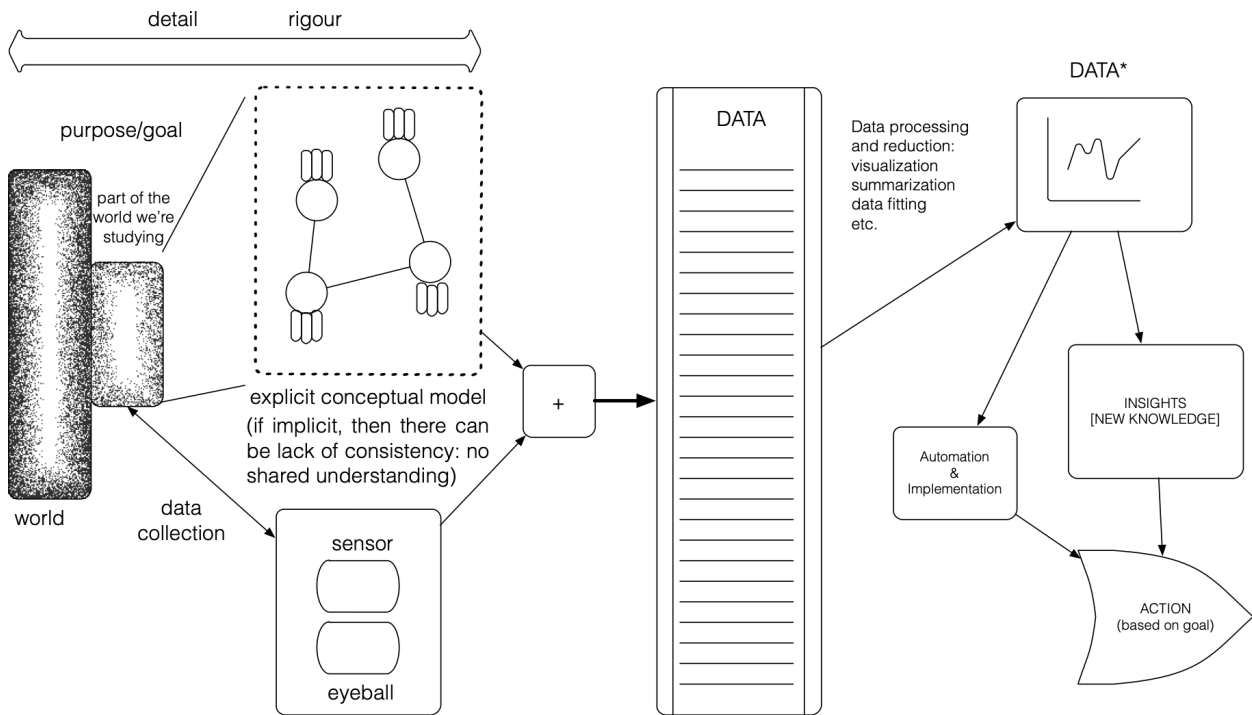


Figure 14.1: A schematic diagram of systems thinking as it applies to a general problem.

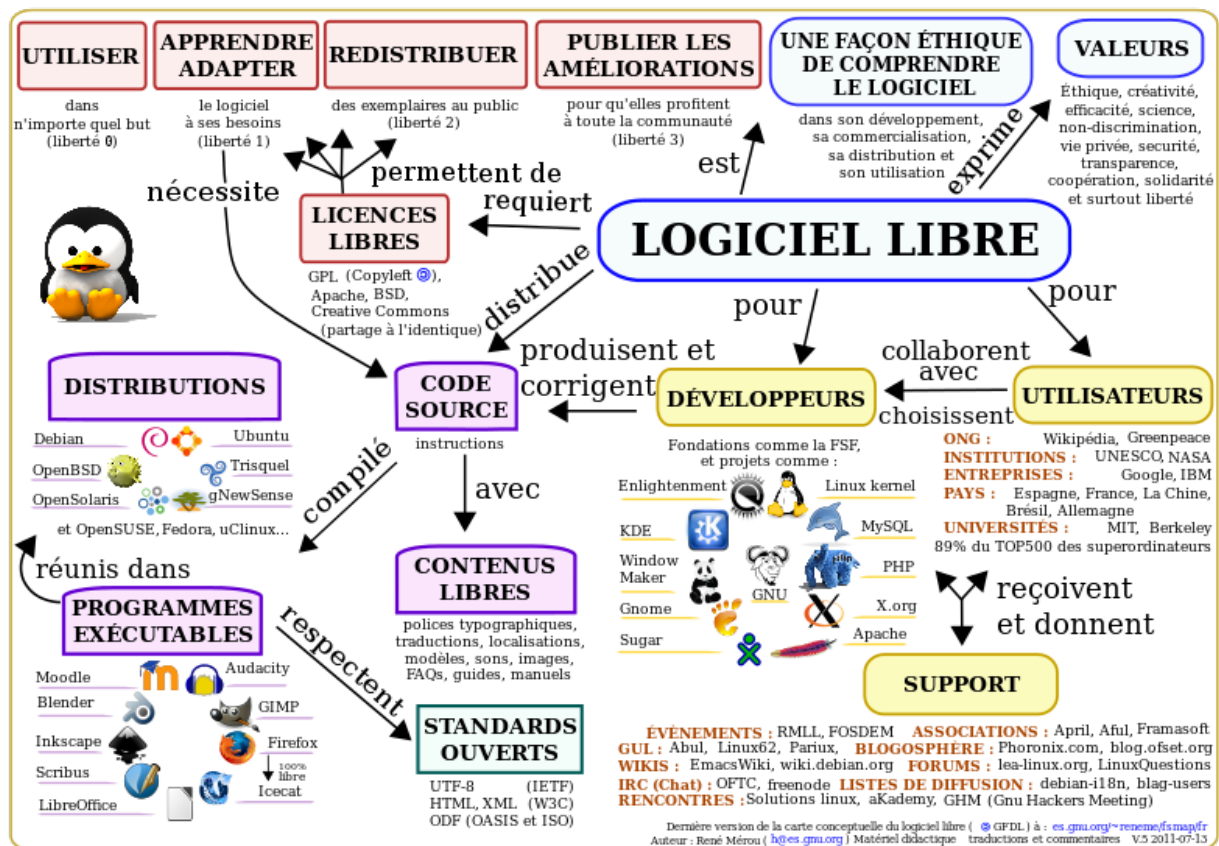


Figure 14.2: A conceptual model of the 'free software' system (in French) [51].

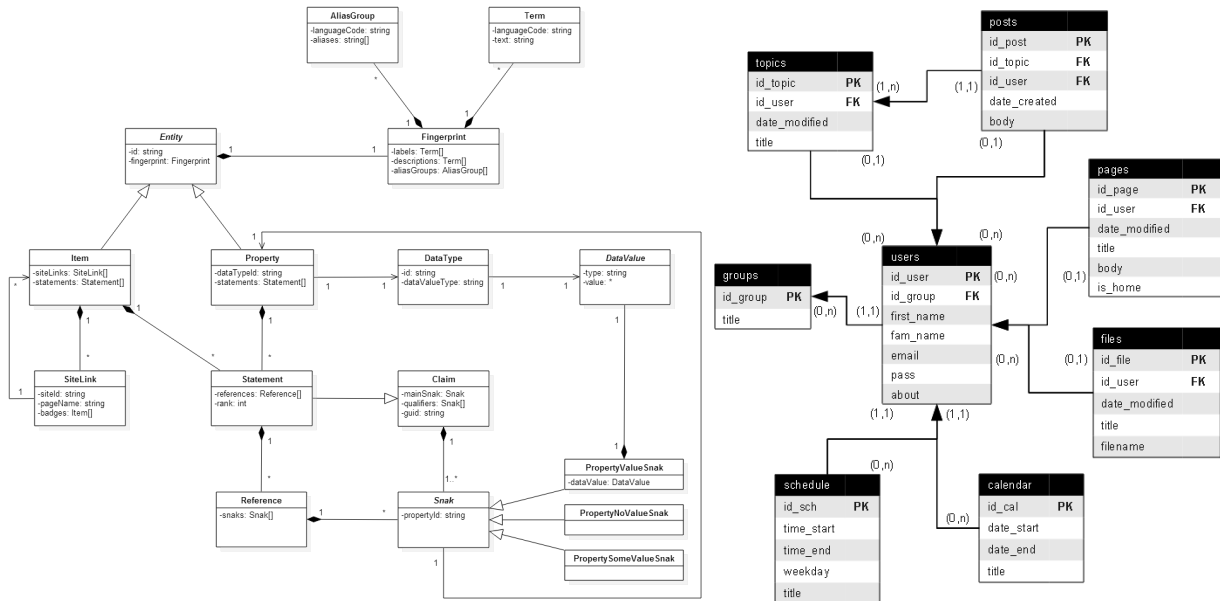
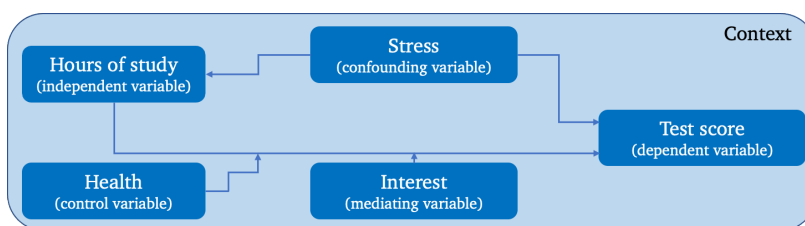


Figure 14.3: Examples of UML diagram (Wikibase Data Model, on the left [35]) and ER conceptual map (on the right [79]).

In practice, we must first select a system for the task at hand, then generate a conceptual model that encompasses:

- **relevant and key objects** (abstract or concrete);
- **properties** of these objects, and their values;
- **relationships between objects** (part-whole, is-a, object-specific, one-to-many), and
- **relationships between properties** across instances of an object type.

A simplistic example describing a supposed relationship between a **presumed cause** (hours of study) and a **presumed effect** (test score) is shown below:



Relating the Data to the System

From a pragmatic perspective, stakeholders and analysts alike need to know if the data which has been collected and analyzed will be useful to understand the system.

This question can best be answered if we understand:

- how the data is collected;
- the approximate nature of both data and system, and
- what the data represents (observations and features).

Is the **combination of system and data sufficient** to understand the aspects of the world under consideration? Once again, this is difficult to answer in practice. Contextual knowledge can help, but if the data, the system, and the world are **out of alignment**, any data insight drawn from mathematical, ontological, grammatical, or data models of the situation might ultimately prove useless.

14.2.3 Cognitive Biases

Adding to the challenge of building good conceptual models and using these to interpret the data is the fact that we are all vulnerable to a vast array of **cognitive biases**, which influence both how we construct our models and how we look for patterns in the data.

Such biases are difficult to detect in the spur of the moment, but making a conscious effort to identify them and setting up a clear and pre-defined set of thresholds and strategies for analysis will help reduce their negative impact. Here is a sample of such biases [46, 26]).⁵

5: Other biases impacting our ability to make informed decisions include: bandwagon effect, base rate fallacy, bounded rationality, category size bias, commitment bias, Dunning-Kruger effect, framing effect, hot-hand fallacy, IKEA effect, illusion of explanatory depth, illusion of validity, illusory correlations, look elsewhere effect, optimism effect, planning fallacy, pro-innovation bias, representative heuristic, response bias, selective perception, stereotyping, etc.

Anchoring Bias causes us to rely too heavily on the first piece of information we are given about a topic; in a salary negotiation, for instance, whoever makes the first offer establishes a range of reasonable possibilities in both parties' minds.

Availability Heuristic describes our tendency to use information that comes to mind quickly and easily when making decisions about the future; someone might argue that climate change is a hoax because the weather in their neck of the woods has not (yet!) changed.

Choice-Supporting Bias causes us to view our actions in a positive light, even if they are flawed; we are more likely to sweep anomalous or odd results under the carpet when they arise from our own analyses.

Clustering Illusion refers to our tendency to see patterns in random events; if a die has rolled five 3's in a row, we might conclude that the next throw is more (or less) likely to come up a 3 (gambling fallacy).

Confirmation Bias describes our tendency to notice, focus on, and give greater credence to evidence that fits with our existing beliefs; gaffes made by politicians you oppose reinforces your dislike.

Conservation Bias occurs when we favour prior evidence over new information; it might be difficult to accept that there is an association between factors *X* and *Y* if none had been found in the past.

Ostrich Effect describes how people often avoid negative information, including feedback that could help them monitor their goal progress; a professor might chose to not consult their teaching evaluations, for whatever reason.

Outcome Bias refers to our tendency to judge a decision on the outcome, rather than on why it was made; the fact that analysts gave Clinton an 80% chance of winning the 2016 U.S. Presidential Election does not mean that the forecasts were wrong.

Overconfidence causes us to take greater risks in our daily lives; experts are particularly prone to this, as they are more convinced that they are right.

Recency Bias occurs when we favour new information over prior evidence; investors tend to view today's market as the "forever" market and make poor decisions as a result.

Salience Bias describes our tendency to focus on items or information that are more noteworthy while ignoring those that do not grab our attention; you might be more worried about dying in a plane crash than in a car crash, even though the latter occurs more frequently than the former.

Survivorship Bias is a cognitive shortcut that occurs when a visible successful subgroup is mistaken as an entire group, due to the failure subgroup not being visible; when trying to get the full data picture, it helps to know what observations did not make it into the dataset.

Zero-Risk Bias relates to our preference for absolute certainty; we tend to opt for situations where we can completely eliminate risk, seeking solace in the figure of 0%, over alternatives that may actually offer greater risk reduction.

14.3 Ethics in the Data Science Context

A lapse in ethics can be a conscious choice ... but it can also be negligence. [68]

In most empirical disciplines, **ethics** are brought up fairly early in the educational process and may end up playing a crucial role in researchers' activities. At Memorial University of Newfoundland, for instance, "proposals for research in the social sciences, humanities, sciences, and engineering, including some health-related research in these areas," must receive approval from specific Ethics Review Boards.

This could apply to research and analysis involving [63]:

- living human subjects;
- human remains, cadavers, tissues, biological fluids, embryos or foetuses;
- a living individual in the public arena if they are to be interviewed and/or private papers accessed;
- secondary use of data – health records, employee records, student records, computer listings, banked tissue – if any form of identifier is involved and/or if private information pertaining to individuals is involved, and
- quality assurance studies and program evaluations which address a research question.

In our experience, data scientists and data analysts who come to the field by way of mathematics, statistics, computer science, economics, or engineering, however, are not as likely to have encountered ethical research boards or to have had **formal ethics training**.⁶

6: We are obviously not implying that these individuals have no ethical principles or are unethical; rather, that the opportunity to establish what these principles might be, in relation with their research, may never have presented itself.

Furthermore, discussions on ethical matters are often tabled, perhaps understandably, in favour of pressing technical or administrative considerations (such as algorithm selection, data cleaning strategies, contractual issues, etc.) when faced with hard deadlines.

The problem, of course, is that the current deadline is eventually replaced by another deadline, and then by a new deadline, with the end result being that the conversation may never take place. It is to address this all-too-common scenario that we take the time to discuss ethics in the **data science context**; more information is available in [56, 67].

14.3.1 The Need for Ethics

When large-scale data collection first became possible, there was to some extent a ‘Wild West’ mentality to data collection and use. To borrow from the old English law principle, whatever was not prohibited (from a technological perspective) was allowed.

Now, however, **professional codes of conduct** are being devised for data scientists [17, 74, 1], outlining responsible ways to practice data science – ways that are legitimate rather than fraudulent, and ethical rather than unethical.⁷

Although this shifts some added responsibility onto data scientists, it also provides them with protection from clients or employers who would hire them to carry out data science in questionable ways – they can refuse on the grounds that it is against their professional code of conduct.

14.3.2 What Is/Are Ethics?

Broadly speaking, ethics refers to the study and definition of right and wrong conduct. Ethics may consider what is right or wrong when it comes to actions in general, or consider how broad ethical principles are appropriately applied in more specific circumstances.

And, as noted by R.W. Paul and L. Elder, ethics is not (necessarily) the same as social convention, religious beliefs, or laws [57]; that distinction is not always fully understood. The following influential ethical theories are often used to frame the debate around ethical issues in the data science context.

- **Golden rule:** do unto others as you would have them do unto you;
- **Consequentialism:** the end justifies the means;
- **Utilitarianism:** act in order to maximize positive effect;
- **Moral Rights:** act to maintain and protect the fundamental rights and privileges of the people affected by actions;
- **Justice:** distribute benefits and harm among stakeholders in a fair, equitable, or impartial way.

In general, it is important to remember that our planet’s inhabitants subscribe to a wide variety of ethical codes, including:

Confucianism, Taoism, Buddhism, Shinto, Ubuntu, Te Ara Tika (Maori), First Nations Principles of OCAP, various aspects of Islamic ethics, etc.

7: This is not to say that ethical issues have miraculously disappeared – Volkswagen, Whole Foods Markets, General Motors, Cambridge Analytica, and Ashley Madison, to name but a few of the big data science and data analysis players, have all recently been implicated in ethical lapses [29]. More dubious examples can be found in [52, 19].

It is not too difficult to imagine contexts in which any of these (or other ethical codes, or combinations thereof) would be better-suited to the task at hand – the challenge is to remember to **inquire** and to **heed the answers**.

14.3.3 Ethics and Data Science

How might these ethical theories apply to data analysis? The (former) University of Virginia's *Centre for Big Data Ethics, Law and Policy* suggested some specific examples of data science ethics questions [16]:

- who, if anyone, owns data?
- are there limits to how data can be used?
- are there value-biases built into certain analytics?
- are there categories that should never be used in analyzing personal data?
- should data be publicly available to all researchers?

The answers may depend on a number of factors, not the least of which is the matter of who is actually providing them to you. To give you an idea of some of the complexities, let us consider as an example the first of those questions: who, if anyone, owns data?

In some sense, the **data analysts** who transform the data's potential into usable insights are only one of the links in the entire chain. Processing and analyzing the data would be impossible without raw data on which to work, so the **data collectors** have a strong ownership claim to the data.

But collecting the data can be a costly endeavour, and it is easy to imagine how the **sponsors** or **employers** (who made the process economically viable in the first place) might feel that the data and its insights are rightfully theirs to dispose of as they wish.

In some instances, the **law** may chime in as well. Indeed, one can easily list other players, but let it suffice to say that this simple question turns out to be far from easily answered, and may even change from case to case. Incidentally, this also highlights a hidden truth regarding the data analysis process: there is more to data analysis than *just* data analysis.

A similar challenge arises in regards to **open data**, where the “pro” and “anti” factions both have strong arguments (see [53, 14, 54], as well as [22] for a science-fictional treatment of the transparency vs security debate).

The answers to the above ethical questions aside, a general principle of data analysis is to **eschew the anecdotal** in favour of the **general** – from a purely analytical perspective, too narrow a focus on specific observations can end up obscuring the full picture (for a vivid illustration, see [20]).

But data points are **not** solely marks on paper or electro-magnetic bytes on the cloud. Decisions made on the basis of data science (in all manners of contexts, from security, to financial and marketing context, as well as policy) may **affect living beings in negative ways**. And it can not be ignored that outlying/marginal individuals and minority groups often suffer disproportionately at the hands of so-called evidence-based decisions [31, 42, 43].

14.3.4 Guiding Principles

Under the assumption that one is convinced of the importance of proceeding ethically, it could prove helpful to have a set of guiding principles to aid in these efforts.

In his seminal science fiction series about *positronic robots*, Isaac Asimov introduced the now-famous *Laws of Robotics*, which he believed would have to be built-in so that robots (and by extension, any tool used by human beings) could overcome humanity's *Frankenstein's* complex (the fear of mechanical beings) and help rather than hinder human social, scientific, cultural, and economic activities [5]:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the 1st Law.
3. A robot must protect its own existence as long as such protection does not conflict with the 1st and 2nd Law.

Had they been uniformly well-implemented and respected, the potential for story-telling would have been somewhat reduced; thankfully, Asimov found entertaining ways to break the Laws (and to resolve the resulting conflicts) which made the stories both enjoyable and insightful.

Interestingly enough, he realized over time that a Zeroth Law had to supersede the First in order for the increasingly complex and intelligent robots to succeed in their goals. Later on, other thinkers contributed a few others, filling in some of the holes.

Asimov's (expanded) *Laws of Robotics*:

00. A robot may not harm sentience or, through inaction, allow sentience to come to harm.
0. A robot may not harm humanity, or, through inaction, allow humanity to come to harm, as long as this action/inaction does not conflict with the 00th Law.
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm, as long as this does not conflict with the 00th or the 0th Law.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the 00th, the 0th or the 1st Law.
3. A robot must protect its own existence as long as such protection does not conflict with the 00th, the 0th, the 1st or the 2nd Law.
4. A robot must reproduce, as long as such reproduction does not interfere with the 00th, the 0th, the 1st, the 2nd or the 3rd Law.
5. A robot must know it is a robot, unless such knowledge would contradict the 00th, the 0th, the 1st, the 2nd, the 3rd or the 4th Law.

We cannot speak to the validity of these laws for **robotics** (a term coined by Asimov, by the way), but we do find the entire set satisfyingly complete.

What does this have to do with data science? Various thinkers have discussed the existence and potential merits of different sets of Laws ([71]) – wouldn't it be useful if there were *Laws of Analytics*, **moral principles that could help us conduct data science ethically**?

Best Practices

Such universal principles are unlikely to exist, but best practices have nonetheless been suggested over the years.

“Do No Harm”: Data collected from an individual **should not be used to harm the individual**. This may be difficult to track in practice, as data scientists and analysts do not always participate in the ultimate decision process.

Informed Consent: Covers a wide variety of ethical issues, chief among them being that **individuals must agree to the collection and use** of their data, and that they must have a **real understanding of what they are consenting to**, and of **possible consequences** for them and others.

The Respect of “Privacy”: This principle is dearly-held in theory, but it is hard to adhere to it religiously with robots and spiders constantly trolling the net for personal data. In the *Transparent Society*, D. Brin (somewhat) controversially suggests that privacy and total transparency are closely linked [14]:

“Transparency is also the trick to protecting privacy, if we empower citizens to notice when neighbors infringe upon it. Isn't that how you enforce your own privacy in restaurants, where people leave each other alone, because those who stare or listen risk getting caught?”

Keeping Data Public: Another aspect of data privacy, and a thornier issue – should some data be kept private? Most? All? It is fairly straightforward to imagine scenarios where adherence to the principle of public data could cause harm to individuals (for instance, revealing the source of a leak in a country where the government routinely jails members of the opposition), thereby contradicting the first principle against causing harm. It is just as easy to imagine scenarios where keeping data private would have a similar effect.

Opt-in/Opt-out: Informed consent requires the ability to **not consent**, i.e., to opt out. Non-active consent is not really consent.

Anonymize Data: Identifying fields should be removed from the dataset **prior** to processing and analysis. Remove any temptation to use personal information in an inappropriate manner from the get-go, but be aware that this is easier said than done, technically-speaking.

Let the Data Speak: It is crucial to absolutely restrain oneself from **cherry-picking** the data. Use all of it in some way or another; validate your analysis and make sure your results are repeatable.

14.3.5 The Good, the Bad, and the Ugly

Data projects could whimsically be classified as **good**, **bad** or **ugly**, either from a technical or from an ethical standpoint (or both). We have identified instances in each of these classes (of course, our own biases are showing):

- **good** projects increase knowledge, can help uncover hidden links, and so on: [23, 40, 38, 41, 75, 70, 9, 8, 6, 45, 55, 21, 10, 58, 15]
- **bad** projects can lead to bad decisions, which can in turn decrease the public's confidence and potentially harm some individuals: [60, 76, 39, 48, 20]
- **ugly** projects are, flat out, unsavoury applications, even if the initial impetus for the work was noble; either they are poorly executed from a technical perspective, or they put a lot of people at risk; these (and similar approaches/studies) should be avoided: [7, 44, 24, 43, 42, 31]

14.4 Analytics Workflows

An overriding component of the discussion so far has been the **importance of context**. And although the reader may be eager at this point to move into data analysis proper, there is one more bit of context that should be considered first – the **project context**.

We have alluded to the idea that data science is much more than merely data analysis, and this is apparent when we look at the typical steps involved in a data science project. Inevitably, data analysis pieces take place within this larger project context, as well as in the context of a larger **technical infrastructure** or **pre-existing system**.

14.4.1 The “Analytical” Method

As with the **scientific method**, there is a “step-by-step” guide to data analysis:

1. statement of objective
2. data collection
3. data clean-up
4. data analysis/analytics
5. dissemination
6. documentation

Notice that **data analysis** only makes up a small segment of the entire flow.

In practice, the process often end up being a bit of a mess, with steps taken out of sequence, steps added-in, repetitions and re-takes (see Figure 14.4).

And yet ... it tends to work on the whole, if conducted correctly.

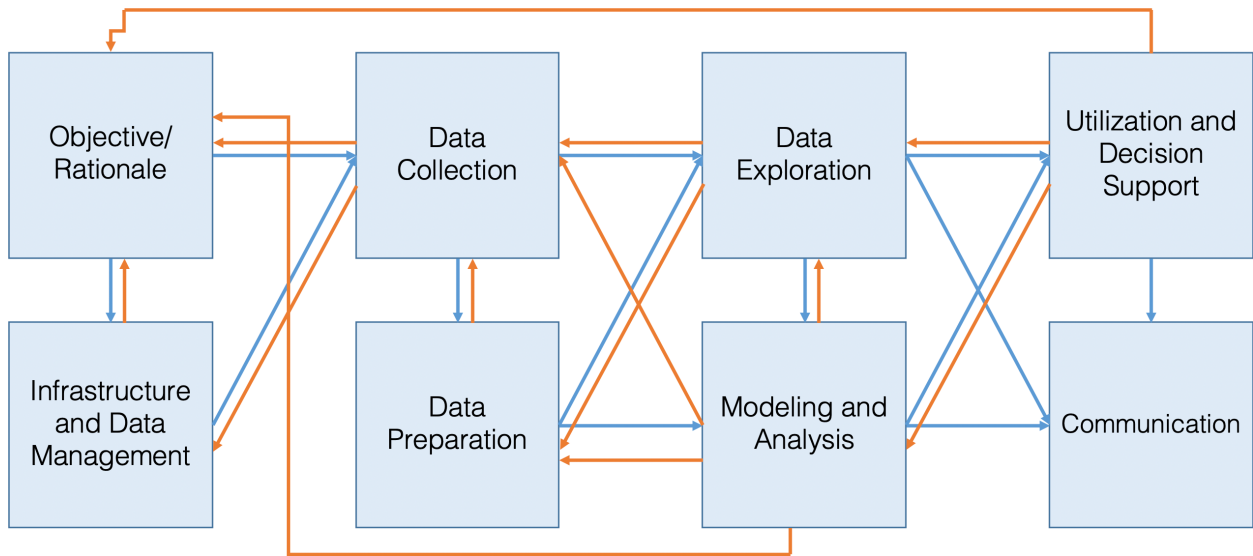
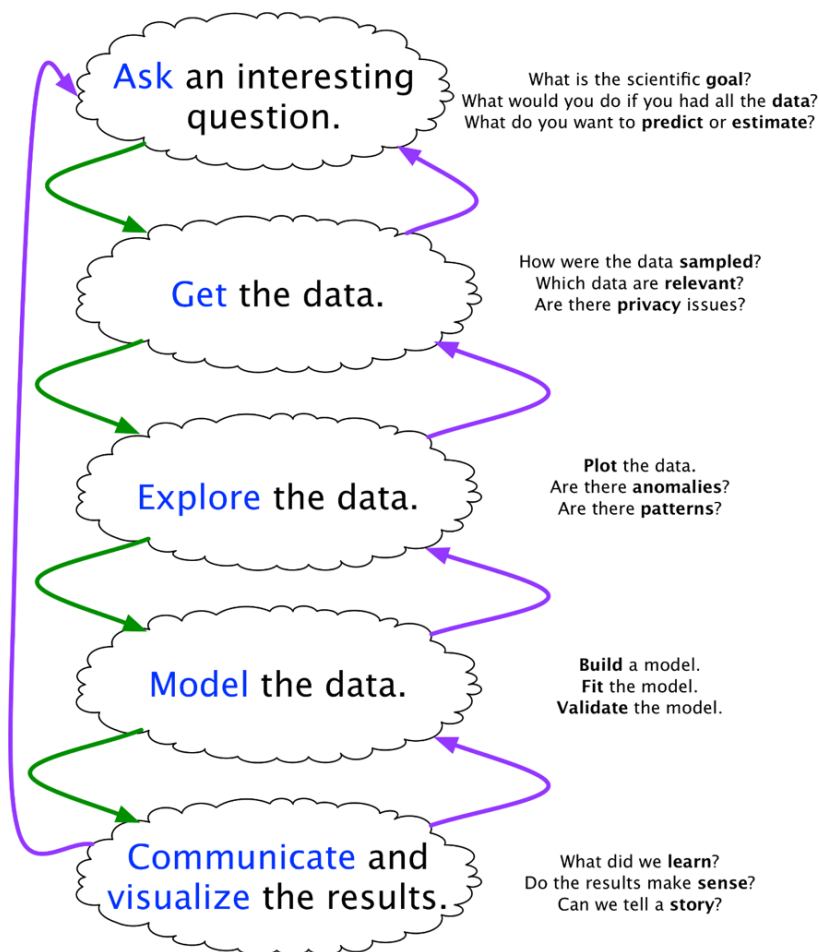


Figure 14.4: The reality of the analytic workflow – definitely not a linear process!

Blitzstein and Pfister (who teach a well-rated data science course at Harvard) provide their own workflow diagram, but the similarities are easy to spot (see below).



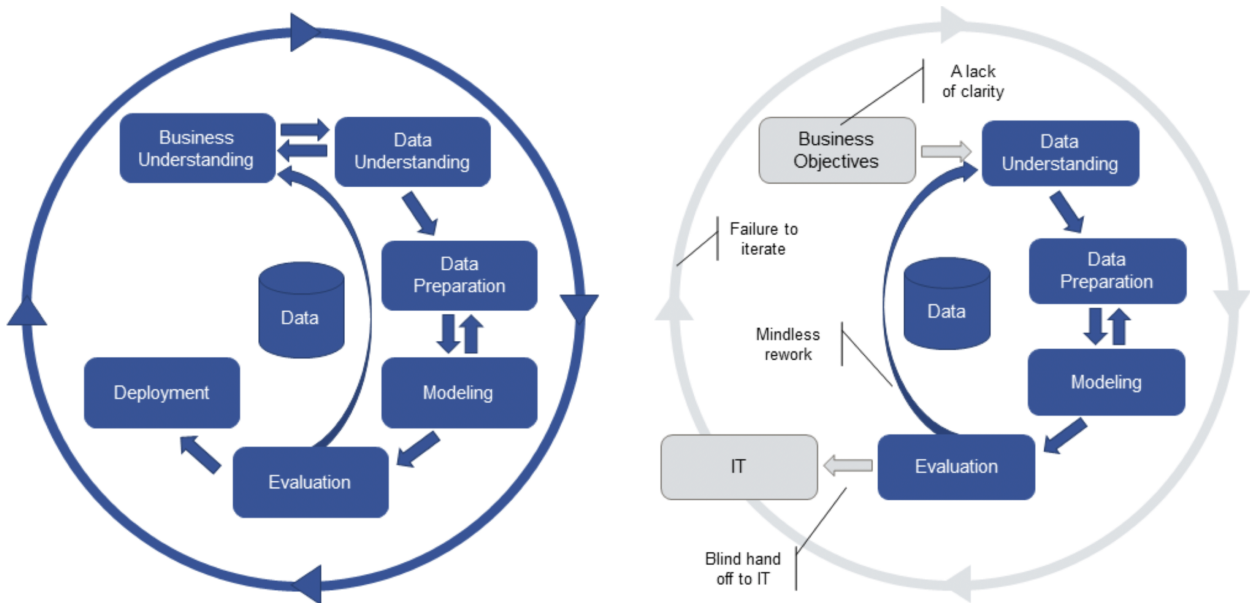


Figure 14.5: Theoretical (on the left) and corrupted (on the right) CRISP-DM processes [73].

The **Cross Industry Standard Process, Data Mining (CRISP-DM)** is another such framework, with projects consisting of 6 steps:

1. business understanding
2. data understanding
3. data preparation
4. modeling
5. evaluation
6. deployment

The process is iterative and interactive – the dependencies are highlighted in Figure 14.5. In practice, data analysis is often corrupted by:

1. lack of clarity;
2. mindless rework;
3. blind hand-off to IT, and
4. failure to iterate.

CRISP-DM has a definite old-hat flavour (as exemplified by the use of the outdated expression “data mining”), but it can be useful to check off its sub-components, if only as a **sanity check**.

Business Understanding

- understanding the business goal
- assessing the situation
- translating the goal in a data analysis objective
- developing a project plan

Data Understanding

- considering data requirements
- collecting and exploring data

Data Preparation

- selection of appropriate data
- data integration and formatting
- data cleaning and processing

Modeling

- selecting appropriate techniques
- splitting into training/testing sets
- exploring alternatives methods
- fine tuning model settings

Evaluation

- evaluation of model in a business context
- model approval

Deployment

- reporting findings
- planning the deployment
- deploying the model
- distributing and integrating the results
- developing a maintenance plan
- reviewing the project
- planning the next steps

All these approaches have a common core: data science projects are **iterative** and (often) **non-sequential**. Helping the clients and/or stakeholders recognize this central truth will make it easier for analysts and consultants to **plan the data science process** and to obtain **actionable insights** for organizations and sponsors.

The main take-away from this section, however, is that there is a great deal to consider in advance of modeling and analysis – once more, **data science is not solely about data analysis**.

14.4.2 Collection, Storage, Processing, Modeling

Data enters the **data science pipeline** by first being **collected**. There are various ways to do this:

- data may be collected in a **single pass**;
- it may be collected in **batches**, or
- it may be collected **continuously**.

The **mode of entry** may have an impact on the subsequent steps, including how frequently models, metrics, and other outputs are **updated**.

Once it is collected, data must be **stored**. Choices related to storage (and **processing**) must reflect:

- how the data is collected (mode of entry);
- how much data there is to store and process (small vs. big), and
- the type of access and processing that will be required (how fast, how much, by whom).

Unfortunately, stored data may go **stale** (both *figuratively*, as in, for example, addresses no longer accurate, names have changed, etc., and *literally*, as in the physical decay of the data and storage space); regular data audits are recommended.

The data must be **processed** before it can be analyzed. This is discussed in detail in Chapter 15 (*Data Preparation*), but the key point is that **raw data** has to be converted into a format that is **amenable to analysis**, by:

- identifying **invalid**, **unsound**, and **anomalous** entries;
- dealing with **missing values**;
- **transforming** the variables and the datasets so that they meet the requirements of the selected algorithms.

In contrast, the **analysis** step itself is almost anti-climactic – simply run the selected methods/algorithms on the processed data. The specifics of this procedure depend, of course, on the choice of method/algorithm.

8: Truth be told, choosing wisely is probably the the most **difficult** aspect of a data science project.

We will not yet get into the details of how to make that choice⁸, but data science teams should be familiar with a fair number of techniques and approaches:

- data cleaning
- descriptive statistics and correlation
- probability and inferential statistics
- regression analysis (linear and other variants)
- survey sampling
- bayesian analysis
- classification and supervised learning
- clustering and unsupervised learning
- anomaly detection and outlier analysis
- time series analysis and forecasting
- optimization
- high-dimensional data analysis
- stochastic modeling
- distributed computing
- etc.

These only represent a **small slice** of the analysis pie. It is difficult to imagine that any one analyst/data scientist could master all (or even a majority of them) at any moment, but that is one of the reasons why data science is a team activity (more on this in Section 13.1.3, *Roles and Responsibilities*).

14.4.3 Model Assessment and Life After Analysis

Before applying the findings from a model or an analysis, one must first confirm that the model is reaching valid conclusions about the system of interest.

All analytical processes are, by their very nature, **reductive** – the raw data is eventually transformed into a small(er) **numerical outcome** (or summary) by various analytical methods, which we hope is still **related** to the system of interest, see Section 14.2 (*Conceptual Frameworks for Data Work*).

9: Are the results analytically compatible with the data, say?

Data science methodologies include an **assessment** (evaluation, validation) phase. This does not solely provide an analytical sanity check;⁹ it can also be used to determine when the system and the data science process have stepped out of alignment.

Note that past successes can lead to reluctance to re-assess and re-evaluate a model (the so-called **tyranny of past success**); even if the analytical approach has been vetted and has given useful answers in the past, it may not always do so.

At what point does one determine that the current data model is **out-of-date**? At what point does one determine that the current model is no longer **useful**? How long does it take a model to react to a **conceptual shift**?¹⁰

This is another reason why regular audits are recommended – as long as the analysts remain in the picture, the only obstacle to performance evaluation might be the technical difficulty of conducting said evaluation.

When an analysis or model is ‘released into the wild’ or delivered to the client, it often takes on a life of its own. When it inevitably ceases to be **current**, there may be little that (former) analysts can do to remedy the situation.

Data analysts and scientists rarely have full (or even partial) control over **model dissemination**. Consequently, results may be misappropriated, misunderstood, shelved, or failed to be updated, all without their knowledge. Can conscientious analysts do anything to prevent this?

Unfortunately, there is no easy answer short of advocating that analysts and consultants not only focus on data analysis, but also recognize the opportunity that arises during a project to **educate clients and stakeholders** on the importance of these auxiliary concepts.

Finally, because of **analytic decay**, it is crucial not to view the last step in the analytical process as a **static dead end**, but rather as an invitation to return to the beginning of the process.

10: How long does it take Netflix to figure out that you no longer like action movies and want to watch comedies instead, say? How long does it take Facebook to recognize that you and your spouse have separated and that you do not wish to see old pictures of them in your feed?

14.4.4 Automated Data Pipelines

In the **service delivery context**, the data analysis process is typically implemented as an **automated data pipeline** to enable the analysis process to occur repeatedly and automatically.

Data pipelines usually consist of 9 components (5 **stages** and 4 **transitions**, as in Figure 14.9):

1. data collection
2. data storage
3. data preparation
4. data analysis
5. data presentation

Each of these components must be **designed** and then **implemented**. Typically, at least one pass of the data analysis process has to be done **manually** before the implementation is completed. We will return to this topic in Section 14.5.2 (*Structuring and Organizing Data*).

14.5 Getting Insight From Data

With all of the appropriate context now in mind, we can finally turn to the main attraction, **data analysis** proper. Let us start this section with a few definitions, in order to distinguish between some of the common categories of data analysis.

What is Data Analysis?

We view **finding patterns in data** as being data analysis's main goal. Alternatively, we describe the data analysis process as **using data to**:

- answer specific questions;
- help in the decision-making process;
- create models of the data;
- describe or explain the situation or system under investigation,
- etc.

While some practitioners include other analytical-like activities, such as testing (scientific) hypotheses, or carrying out calculations on data, we think of those as separate activities.

What is Data Science?

One of the challenges of working in the data science field is that nearly all quantitative work can be described as data science (often to a ridiculous extent). Our simple definition paraphrases T. Kwartler: data science is the collection of processes by which we extract **useful** and **actionable insights** from data. Robinson [65] further suggests that these insights usually come *via* **visualization** and (manual) **inferential analysis**.

The noted data scientist H. Mason thinks of the discipline as “the **working intersection** of statistics, engineering, computer science, domain expertise, and ‘hacking’ ” [78].

What is Machine Learning?

Starting in the 1940s, researchers began to take seriously the idea that machines could be taught to **learn**, **adapt** and **respond** to novel situations. A wide variety of techniques, accompanied by a great deal of theoretical underpinning, were created in an effort to achieve this goal.

Machine learning is typically used to obtain “predictions” (or “advice”), while reducing the operator’s analytical, inferential and decisional workload (although it is still present to some extent) [65].

What is Artificial/Augmented Intelligence?

The science fiction answer is that artificial intelligence is **non-human intelligence** that has been **engineered** rather than one that has evolved naturally. Practically speaking, this translates to “computers carrying out tasks that only humans can do”. A.I. attempts to remove the need for oversight, allowing for automatic “actions” to be taken by a completely unattended system.

These goals are laudable in an academic setting, but we believe that stakeholders (and humans, in general) should not seek to abdicate all of their agency in the decision-making process. As such, we follow the lead of various thinkers and suggest further splitting A.I. into **general A.I.** (who would operate independently of human intelligence) and **augmented** intelligence (which enhances human intelligence).

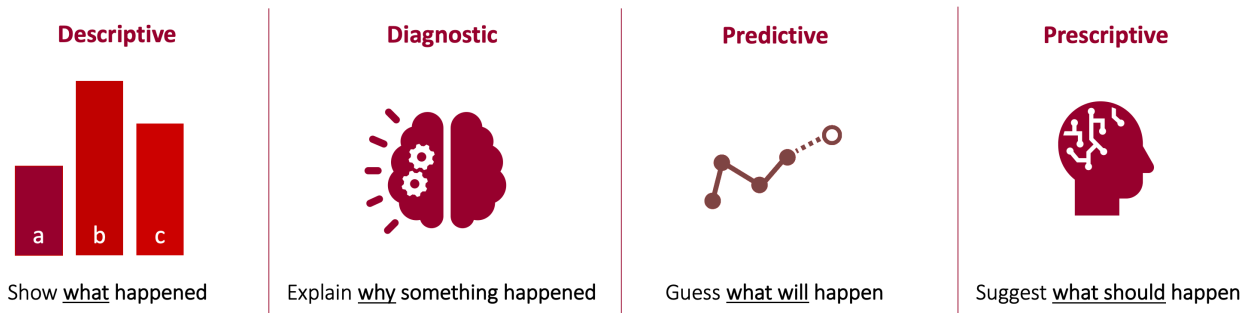


Figure 14.6: Analysis/data science buckets [Marwan Kashef].

These approaches can be further broken down into 4 **core key buckets** (see Figure 14.6), moving roughly from **low value/low difficulty** propositions (left) to **high value/high difficulty** propositions (right).

For instance, a shoe store could conduct the following analyses:

Descriptive Sales report

Diagnostic Why did the sales take a large dip?

Predictive What is the sales forecast next quarter?

Prescriptive: How should we change the product mix to reach our target sales goal?

14.5.1 Asking the Right Questions

Definitions aside, however, data analysis, data science, machine learning, and artificial intelligence are about **asking questions** and **providing answers** to these questions. We might ask various types of questions, depending on the situation.

Our position is that, from a quantitative perspective, there are only really three types of questions:

- **analytics** questions;
- **data science** questions, and
- **quantitative methods** questions.

Analytics questions could be something as simple as:

how many clicks did a specific link on my website get?

Data science questions tend to be more complex – we might ask something along the lines of:

if we know, historically, when or how often people click on links, can we predict how many people from Winnipeg will access a specific page on our website within the next three hours?

Whereas analytics-type questions are typically answered by **counting things**, data science-like questions are answered by using historical patterns to **make predictions**.

Quantitative methods questions might, in our view, be answered by making predictions but not necessarily based on historical data. We could build a model from **first principles** – the “physics” of the situation, as it were – to attempt to figure out what might happen.

For instance, if we thought there was a correlation between the temperature in Winnipeg and whether or not people click on the links in our website, then we might build a model that predicts “how many people from Winnipeg will access a page in the next week?”, say, by trying to predict the weather instead,¹¹ which is not necessarily an easy task.

Analytics models do not usually predict or explain anything – they just **report** on the data, which is itself meant to represent the situation. A data mining or a data science model tends to be **predictive**, but **not necessarily explanatory** – it shows the existence of connections, of correlations, of links, but without explaining why the connections exist.

In a quantitative method model, we may start by assuming that we know what the links are, what the connections are – which presumably means that we have an idea as to why these connections exist¹² – and then we try to **explore the consequences** of the existence of these connections and these links.

This leads to a singular realization that we share with new data scientists and analysts, potentially the single most important piece of advice they will receive in their quantitative career¹³ :

not every situation calls for analytics, data science, statistical analysis, quantitative methods, machine learning, A.I.

Take the time to identify instances where more is asked out of the data than what it can actually yield, and be prepared to warn stakeholders, as early as possible, when such a situation is encountered.

If we cannot ask the right questions of the data, of the client, of the situation, and so on, any associated project is doomed to fail from the very beginning. Without questions to answer, analysts are wasting their time, running analyses for the sake of analysis – **the finish line cannot be reached if there is no finish line**.

In order to help clients/stakeholders, data analysts and scientists need:

- questions **to answer**;
- questions that **can be answered** by the types of methods and skills at their disposal, and
- answers that will be **recognized as answers**.

“How many clicks did this link get?” is a question that is easily answerable if we have a dataset of links and clicks, but it might not be a question that the client cares to see answered. Data analysts and scientists often find themselves in a situation where they will ask the types of questions that can be answered with the **available data**, but the answers might not actually prove useful.

11: Questions can also be asked in an **unsupervised** manner, see [4, 59], among others, and Section 14.5.5 (*Quantitative Methods*), briefly.

12: Unless we’re talking about quantum physics and then all bets are off – nobody has the slightest idea why things happen the way they do, down there.

13: We are not even sure we are joking when we say this...

From a data science perspective, the right question is one that leads to **actionable insights**. And it might mean that old data is discarded and new data is collected in order to answer it. Analysts should beware: given the sometimes onerous price tag associated with data collection, it is not altogether surprising that there will sometimes be pressure from above to keep working with the available data. Stay strong – analysis on the wrong dataset is the wrong analysis!

The Wrong Questions

Wrong questions might be:

- questions that are **too broad** or **too narrow**;
- questions that **no amount of data could ever answer**,
- questions for which **data cannot reasonably be obtained**, etc.

One of the issues with “wrong” questions is that they do not necessarily “break the pipeline”:

- in the **best-case scenario**, stakeholders, clients, colleagues will still recognize the answers as irrelevant.
- in the worst-case scenario, policies will erroneously be implemented (or decisions made) on the basis of answers that have not been identified as misleading and/or useless.

Framing Questions

In general, data science questions are used to:

- **solve problems** (fix pressing issues, understand why something is or isn’t happening, etc.);
- **create meaningful change** (create new standards in the company, etc.),
- **support gut feelings** (approve or disprove blind intuition).

One thing to note is that individuals prefer to **answer a question quickly**, especially in their area of expertise. It is also **strongly** suggested that analysts avoid glancing over the data before they settle on the question(s), to avoid “begging the question”. Finally, not that just as we can be blinded by love, we can also be blinded by solutions: the right solution to the right question is not necessarily the “sexiest” solution.

The website kdnuggets.com  suggests the following roadmap to framing questions:

- Understand the problem (opportunity vs problem)
- What initial assumptions do I have about the situation?
- How will the results be used?
- What are the risks and/or benefits of answering this question?
- What stakeholder questions might arise based on the answer(s)?
- Do I have access to the data necessary for answering this question?
- How will I measure my “success” criteria?

14: Based on an example by M. Kashef.

Example: Should I buy a house? But this is a bit vague; perhaps, instead, the question could be: should I buy a single house in Scotland?¹⁴

Answer: Let's use the roadmap.

- **Understand the problem.** I've been renting for two years and feel like I'm throwing my money away. I want a chance to invest in my own space instead of someone else's.
- **What initial assumptions do I have about the situation?** It's going to be expensive but worth it – it'll be an investment that appreciates over time.
- **How will the results be used?** Either to buy a house or rent a bit longer to save more for a larger down payment.
- **What are the risks and/or benefits of answering this question?** Risk: I could put myself under immense debt and become "house poor". Benefits: I could get into the market just in time to make a fortune, and I won't have to live under the uncertainty from my landlord possibly selling his home.
- **What stakeholder questions might arise based on the answer(s)?** Would this new home be in an area that's safe for kids? Will it be close to my workplace?
- **Do I have access to the data necessary to answer this question?** Yes, through my real estate agent and online real estate brokerages, I can keep my finger on the pulse of the market.
- **How will I measure my "success" criteria?** If I manage to buy a forever home within my \$600k budget, say.

Additional Considerations

Specific questions are preferred over vague questions; questions that encourage qualification/quantification are preferred over **Yes/No questions**. Here are a few examples of questions to avoid [Health Families BC]:

- Is our revenue increasing over time? Has it increased year-over-year?
- Are most of our customers from this demographic?
- Does this project have valuable ambitions for the broader department?
- How great is our hard-working customer success team?
- How often do you triple check your work?

Consider using the following questions, instead:

- What's the distribution of our revenues over the past three months?
- Where are our top 5 high-spending cohorts from?
- What are the different benefits of pursuing this project?
- What are three good and bad traits of our customer success team?
- What kind of quality assurance testing do you carry out on your deliverables?

Question Audit Checklist [The Head Game]:

1. Did I avoid creating any yes/no questions?

2. Would anyone in my team/department understand the question irrespective of their backgrounds?
3. Does the question need more than one sentence to express?
4. Is the question 'balanced' - scope is not too broad that the question will never truly be answered, or too small that the resulting impact is minimal?
5. Is the question being skewed to what may be easier to answer for my/my team's particular skillset(s)?

14.5.2 Structuring and Organizing Data

Let us now resume the discussion that was cut short in Sections 14.1.1 (*What Is Data?*) and 14.1.2 (*From Objects and Attributes to Datasets*).

Data Sources

We cannot have insights from data without data. As with many of the points we have made, this may seem trivially obvious, but there are many aspects of **data acquisition**, **structuring**, and **organization** that have a sizable impact on what insights can be squeezed from data.

Specifically, there are a number of questions that can be considered:

- why do we collect data?
- what can we do with data?
- where does data come from?
- what does "a collection" of data look like?
- how can we describe data?
- must we distinguish between data, information, knowledge?¹⁵

15: According to the adage, "data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom." (C.Stoll, attributed).

Historically, data has had three functions:

- **record keeping** – people/societal management;
- **science** – new general knowledge, and
- **intelligence** – business, military, police, social, domestic, personal.

Traditionally, each of these functions has:

- used different **sources** of information;
- collected **different types of data**, and
- had **different data cultures** and **terminologies**.

As data science is an interdisciplinary field, it should come as no surprise that we may run into all of them on the same project (see Figure 14.7). Ultimately, data is generated from making observations about and taking measurements of the world. In the process of doing so, we are already imposing particular **conceptualizations** and **assumptions** on our raw experience.

More concretely, data comes from a variety of sources:

- records of activity;
- (scientific) observations;
- sensors and monitoring, and
- from computers themselves (more frequently, of late).

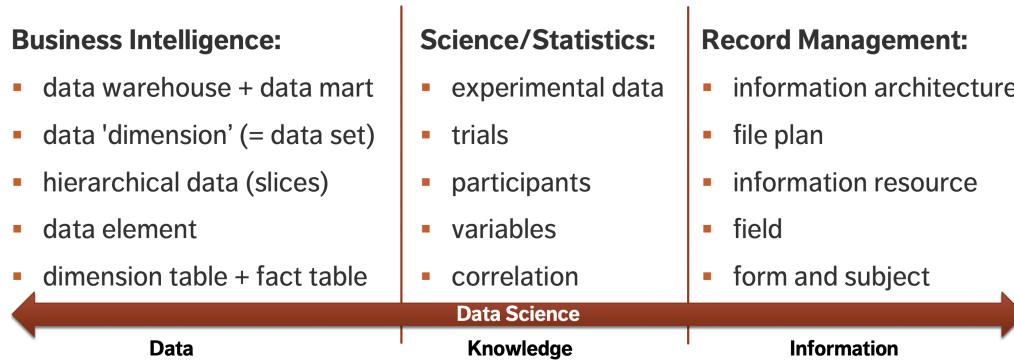


Figure 14.7: Different data cultures and terms.

As discussed in Section 14.1.4, although data may be collected and recorded by hand, it is fast becoming a **mostly digital phenomenon**.

Computer science (and information science) has its own theoretical, **fundamental** viewpoint about data and information, operating over data in a fundamental sense – 1s and 0s that represent numbers, letters, etc. Pragmatically, the resulting data is now stored on computers, and is accessible through our world-wide computer network.

While data is necessarily a representation of **something**, analysts should endeavour to remember that the data itself still has **physical properties**: it takes up physical space and requires energy with which to work. In keeping with this physical nature, data also has a shelf life – it ages over time. We use the phrases “**rotten data/decaying data**” in two senses:

- **literally**, as the data storage medium might decay, but also
- **metaphorically**, as when it no longer accurately represents the relevant objects and relationships (or even when those objects no longer exist in the same way) – compare with “analytical decay”, see Section 14.4.3.

Useful data must stay ‘fresh’ and ‘current’, and avoid going ‘stale’ – but that is both **context-** and **model-dependent!**

Before the Data

The various data disciplines share some **core concepts** and elements, which should resonate with the systems modeling framework previously discussed in Section 14.2 (*Conceptual Frameworks for Data Work*):

- all objects have **attributes**, whether concrete or abstract;
- for multiple objects, there are **relationships** between these objects and attributes, and
- all these elements evolve over time.

The **fundamental relationships** include:

- part-whole;
- is-a;
- is-a-type-of;
- cardinality (one-to-one, one-to-many, many-to-many),
- etc.,

while **object-specific relationships** include:

- ownership;
- social relationship;
- becomes;
- leads-to,
- etc.

Objects and Attributes

We can examine concretely the ways in which objects have properties, relationships and behaviours, and how these are captured and turned into data through observations and measurements, *via* the apple and sandwich example of Section 14.1.1 (*What Is Data?*).

There, we **made measurements** on an apple instance, **labeled the type of observations** we made, and **provided a value describing** the observation. We can further use these labels when observing other apple instances, and associate new values for these new apple instances.

Regarding the fundamental and object specified relationships, we might be able to see that:

- an apple is a type of fruit;
- a sandwich is part of a meal;
- this apple is owned by Jen;
- this sandwich becomes fuel, etc.

It is worth noting that while this all seems tediously obvious to adult humans, it is not so from the perspective of a toddler, or an artificial intelligence. Explicitly, “understanding” requires a basic grasp of:

- categories;
- instances;
- types of attributes;
- values of attributes, and
- which of these are important or relevant to a specific situation or in general terms.

From Attributes to Datasets

Were we to run around in an apple orchard, measuring and jotting down the height, width and colour of 83 different apples completely haphazardly on a piece of paper, the resulting data would be of limited value; although it would technically have been recorded, it would be lacking in **structure**.

We would not be able to tell which values were heights and which were widths, and which colours or which widths were associated with which heights, and *vice-versa*. **Structuring** the data using **lists**, **tables**, or even **tree structures** allows analysts to **record** and **preserve** a number of important relationships:

- those between object types and instances, property/attribute types (sometimes also called fields, features or dimensions), and values;

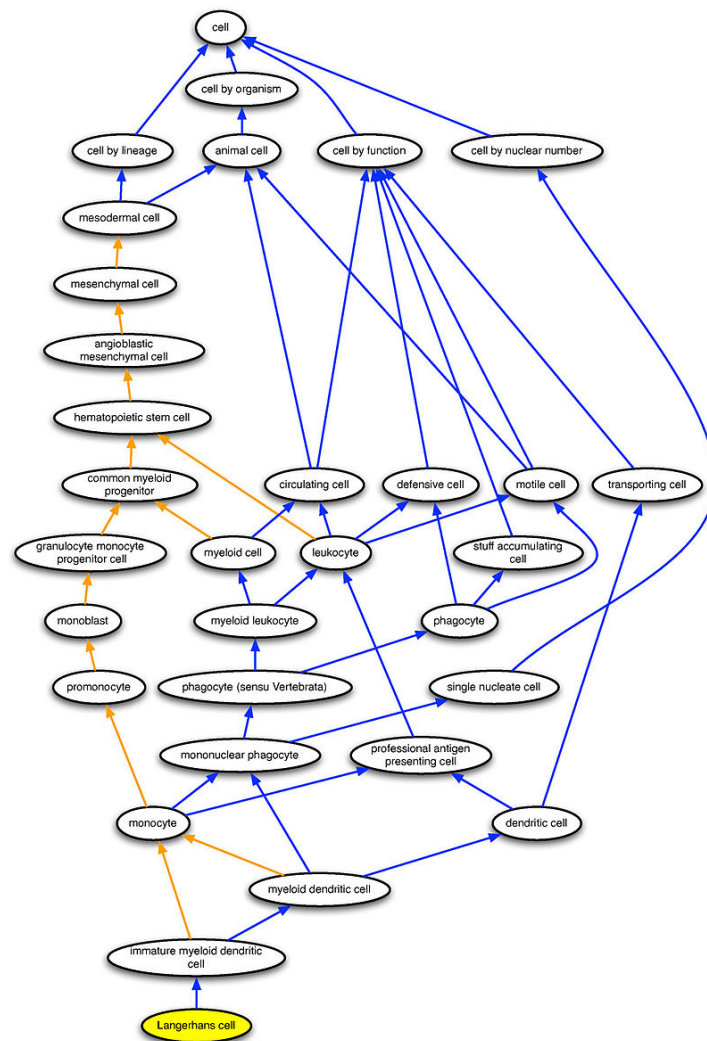
- those between one attribute value and another value (i.e., both of these values are connected to this object instance);
- those between attribute types, in the case of hierarchical data, and
- those between the objects themselves (e.g., this car is owned by this person).

Tables, also called flat files, are likely the most familiar strategy for structuring data in order to preserve and indicate relationships. In the digital age, however, we are developing increasingly sophisticated strategies to store the **structure of relationships** in the data, and finding new ways to work with these increasingly complex relationship structures.

Formally, a **data model** is an abstract (logical) description of both the **dataset structure** and the **system**, constructed in terms that can be implemented in data management software. In a sense, data models lie halfway between **conceptual models** and **database implementations**. The data proper relates to **instances**; the model to **object types**.

Ontologies provide an alternative representation of the system: simply put, they are **structured, machine-readable** collections of **facts** about a domain.¹⁶ For instance, the image below is a representation of the Langerhans cells in the *Cell Ontology* [50].

16: We could facetiously describe ontologies as “data models on steroids.” In a sense, an ontology is an attempt to get closer to the level of detail of a full conceptual model, while keeping the whole machine-readable.



Every time we move from a conceptual model to a specific type of model (a data model, a knowledge model), we lose some information. One way to preserve as much context as possible in these new models is to also provide rich **metadata** – data about the data! Metadata is crucial when it comes to successfully working with and across datasets. **Ontologies** can also play a role here, but that is a topic for another day.

Typically data is stored in a **database**. A major motivator for some of the new developments in types of databases and other data storing strategies is the increasing availability of **unstructured** and '**BLOB**' data.

- **Structured data** is labeled, organized, and discrete, with a pre-defined and constrained form. With that definition, for instance, data that is collected *via* an e-form that only uses drop-down menus is structured.
- **Unstructured data**, by comparison, is not organized, and does not appear in a specific pre-defined data structure – the classical example is text in a document. The text may have to subscribe to specific syntactic and semantic rules to be understandable, but in terms of storage (where spelling mistakes and meaning are irrelevant), it is highly unstructured since any data entry is likely to be completely different from another one in terms of length, etc.
- The acronym “BLOB” stands for **B**inary **L**arge **O**bject data, such as images, audio files, or general multi-media files. Some of these files can be structured-like (all pictures taken from a single camera, say), but they are usually quite unstructured, especially in multi-media modes.

Not every type of database is well-suited to all data types. Let us look at four currently popular database options in terms of fundamental **data** and **knowledge** modeling and structuring strategies:

- key-value pairs (e.g. JSON);
- triples (e.g. resource description framework – RDF);
- graph databases, and
- relational databases.

Key-Value Stores

In these, all data is simply stored as a giant list of keys and values, where the ‘key’ is a name or a label (possibly of an object) and the ‘value’ is a value associated with this key; **triple** stores operate on the same principle, but data is stored according to ‘subject – predicate – object’.

The following examples illustrate these concepts:

1. The *apple type – apple colour* key-value store might contain
 - Granny Smith -- green, and
 - Red Delicious -- red.
2. The *person – shoe size* key-value store might contain
 - Jen Schellinck -- women's size 7, and
 - Colin Henein -- men's size 10.
3. Other key-value stores: *word – definition*, *report name – report (document file)*, *url – webpage*.

4. Triples stores just add a *verb* to the mix: *person – is – age* might contain

- Elwyn -- is -- 20;
- Llewellyn -- is -- 9, and
- Gwynneth -- is -- 6;

while *object – is-colour – colour* might contain

- apple -- is-colour -- red, and
- apple -- is-colour -- green.

Both strategies results in a large amount of flexibility when it comes to the ‘design’ of the data storage, and not much needs to be known about the data structure prior to implementation.¹⁷

17: Additionally, missing values do not take any space in such stores.

In terms of their **implementation**, the devil is in the details; note that their extreme flexibility can also be a flaw [13], and it can be difficult to query them and find the data of interest.

Graph Databases

In **graph databases**, the emphasis is placed on the relationships between different **types of objects**, rather than between an object and the properties of that object:

- the objects are represented by **nodes**;
- the relationships between these objects are represented by **edges**;
- objects can have a relationship with other objects of the same type (such as *person is-a-sibling-of person*).

They are fast and intuitive when using relation-based data, and might in fact be the only reasonable option to use in that case as traditional databases may slow to a crawl. But they are probably too specialized for non relation-based data, and they are not yet widely supported.

Relational Databases

In **relational databases**, data is stored in a series of tables. Broadly speaking, each table represents a type of object and some properties related to this type of object; special columns in tables connect object instances across tables (the entity-relationship model diagram (ERD) of Figure 14.3 is an example of a relational database model).

For instance, a person lives in a house, which has a particular address. Sometimes that property of the house will be stored in the table that stores information about individuals; in other cases, it will make more sense to store information about the house in its own table.

The form of relational databases are driven by the **cardinality** of the relationships (one-to-one, one-to-many, or many-to-many). These concepts are illustrated in the cheat sheet found in Figure 14.8.

Relational databases are widely supported and well understood, and they work well for many types of systems and use cases. Note however, that it is difficult to modify them once they have been implemented and that, ironically, they do not really handle relationships all that well.

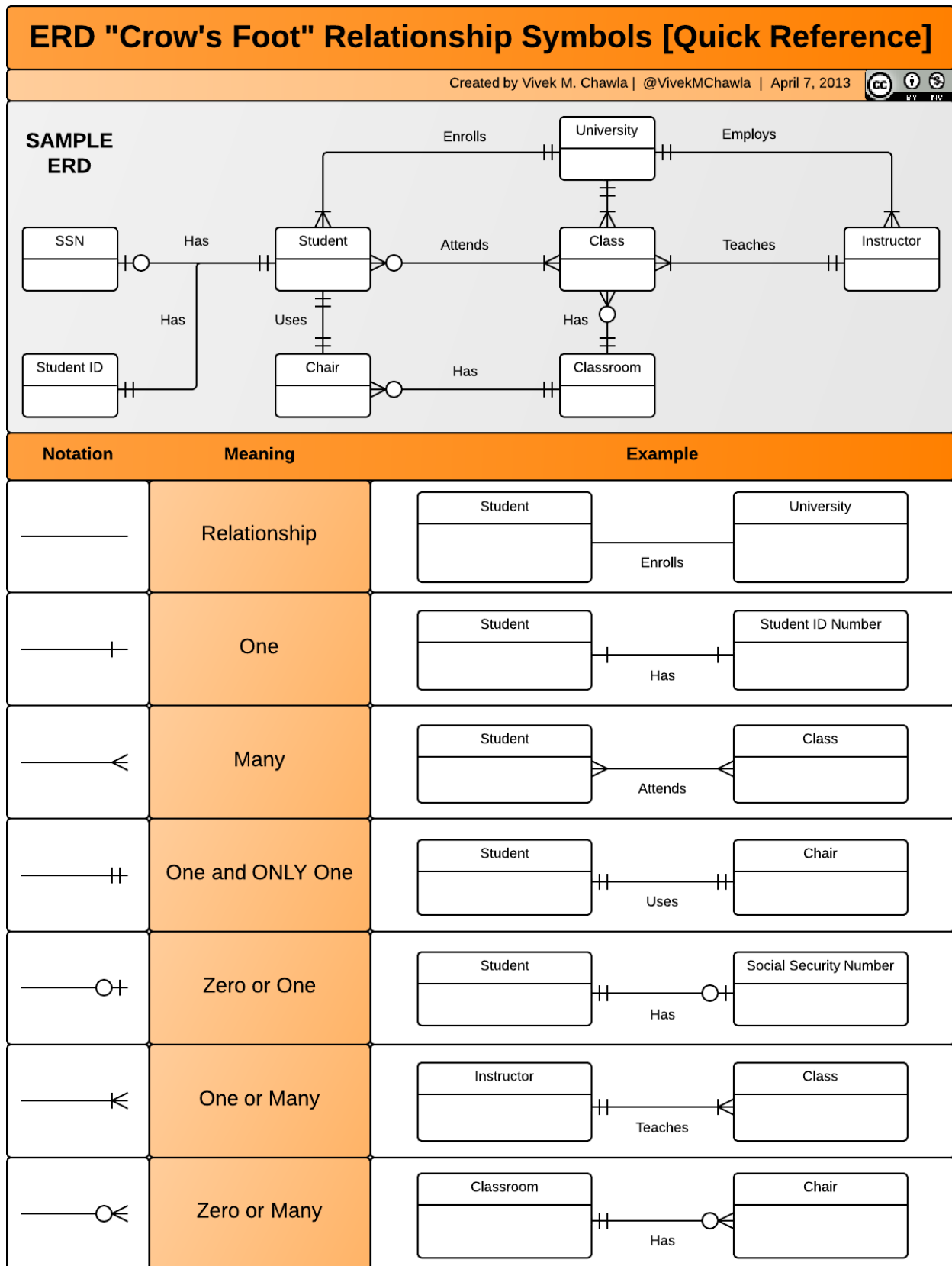


Figure 14.8: Entity-relationship model diagram (so-called) crow's foot relationship symbols cheat sheet [18].

Spreadsheets

We have said very little about keeping data in a single giant table (**spreadsheet**, **flat file**), or multiple spreadsheets (we purposely kept it out of the original list of modeling and structuring strategies).

On the positive side, spreadsheets are efficient when working with:

- **static data** (e.g., it is only collected once), or
- data about **one particular type of object** (e.g., scientific studies).

Most implementations of analytical algorithms require the data to be found in **one location** (such as an R data frame). Since the data will eventually need to be exported to a flat file anyway, why not remove the middle step and work with spreadsheets in the first place?

The problem is that it is hard to manage **data integrity** with spreadsheets over the long term when data is collected (and processed) **continuously**. Furthermore, flat files are not ideal when working with systems involving many different types of objects and their relationships, and they are not optimized for querying operations.

For small datasets or quick work, flat files are often a reasonable option; we should look for alternatives when working on **large scale projects**.

All in all, we have provided very little in the way of concrete information on the topic of databases and data stores. Be aware that, time and time again, projects have **sunk** when this aspect of the process has not been taken seriously. Simply put, serious analyses cannot be conducted properly without the **right data infrastructure**.

Implementing a Model

In order to **implement** the data/knowledge model, data engineers and database specialists need access to **data storage** and **management software**. Gaining this access might be challenging for individuals or small teams as the required software traditionally runs on **servers**.

A server allows multiple users to access the database **simultaneously**, from different client programs. The other side of the coin is that servers make it difficult to ‘play’ with the database.

User-friendly **embedded database software** (vs client-server database engines) such as SQLite can help overcome some of these obstacles. **Data management software** lets human agents interact easily with their data – in a nutshell, they are a **human–data interface**, through which

- data can be **added** to a data collection;
- subsets can be extracted from a data collection based on certain filters/criteria, and
- data can be deleted from (or edited in) a data collection.

18: “Times change, and we change with them.” *C.Huberinus*

But *tempora mutantur, nos et mutamur in illis*¹⁸ – we used to speak of:

- databases and database management systems;
- data **warehouses** (data management system designed to enable **analytics**);

- data **marts** (used to retrieve client-facing data, usually oriented to a specific business line or team);
- **Structured Query Language** (SQL, a commonly-used programming language that helps manage (and perform operations on) relational databases),

we now speak of (see [28]):

- data **lakes** (centralized repository in which to store structured/unstructured data alike);
- data **pools** (a small collection of shared data that aspires to be a data lake, someday);
- data **swamps** (unstructured, ungoverned, and out of control data lake in which data is hard to find/use and is consumed out of context, due to a lack of process, standards and governance);
- database **graveyards** (where databases go to die?),

and data might be stored in **non-traditional** data structures, such as

Popular NoSQL database software include: ArangoDB, MongoDB, Redis, Amazon DynamoDB, OrientDB, Azure CosmosDB, Aerospike, etc.

Once a logical data model is complete, we need only:

1. **instantiate** it in the chosen software;
2. **load** the data, and
3. **query** the data.

Traditional relational databases use SQL; other types of databases either use **other query languages** (AQL, semantic engines, etc.) or rely on **bespoke (tailored) computer programs** (e.g. written in R, Python, etc.).

Once a data collection has been created, it must be **managed**, so that the data remains **accurate, precise, consistent, and complete**. **Databases decay**, after all; if a data lake turns into a data swamp, it will be difficult to squeeze usefulness out of it!

Data and Information Architectures

There is no single correct structure for a given collection of data (or dataset).

Rather, consideration must be given to:

- the **type of relationships** that exist in the data/system (and are thought to be important);
- the **types of analysis** that will be carried out, and
- the **data engineering requirements** relating to the time and effort required to extract and work with the data.

The chosen structure, which stores and organizes the data, is called the **data architecture**. Designing a specific architecture for a data collection is a necessary part of the data analysis process. The data architecture is typically embedded in the larger **data pipeline infrastructure** described in Section 14.4.4 (*Automated Data Pipelines*).

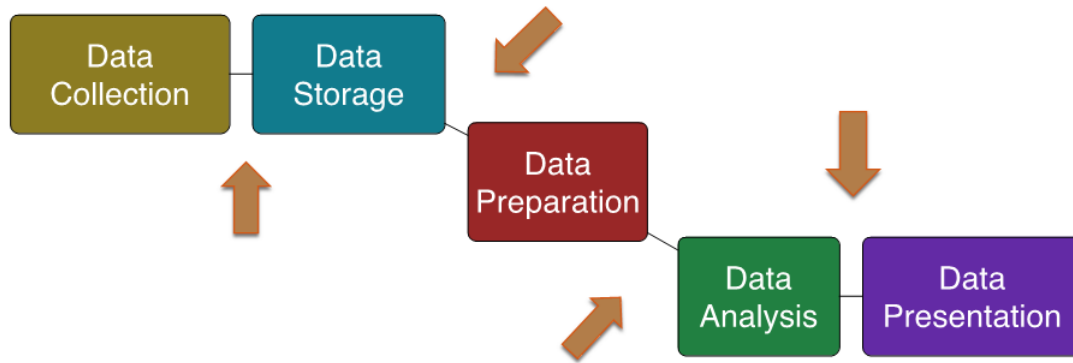


Figure 14.9: An implemented automated pipeline; note the transitions between the 5 stages.

As another example, **automated data pipelines** in the **service delivery context** are usually implemented with 9 components (5 **stages**, and 4 **transitions**, as in Figure 14.9):

1. data collection
2. data storage
3. data preparation
4. data analysis
5. data presentation

Note that **model validation** could be added as a sixth stage, to combat model “drift”.

By analogy with the human body, the **data storage** component, which houses the data and its architecture, is the “heart” of the pipeline,¹⁹ whereas the **data analysis** component is its “brain.”²⁰

Most analysts are familiar with mathematical and statistical models which are implemented in the data analysis component.

Data models, by contrast, tend to get constructed separately from the analytical models at the data storage phase. This separation can be problematic if the analytical model is not compatible with the data model.²¹

If the data comes from forms with various fields stored in a relational database, the discrepancy could create difficulties on the data preparation side of the process. Building both the analytical model and the data model off of a **common conceptual model** might help the data science team avoid such quandaries.

In essence, the task is to structure and organize both data and knowledge so that it can be:

- stored in a useful manner;
- added to easily;
- usefully and efficiently extracted from that store (the “**extract-transform-load**” (ETL) paradigm), and
- operated over by humans and computers alike (programs, bots, A.I.) with minimal external modification.

19: The engine that makes the pipeline go

20: What does that make the other components?

21: As an example, if an analyst needs a flat file (with variables represented as columns) to feed into an algorithm implemented in R, say.

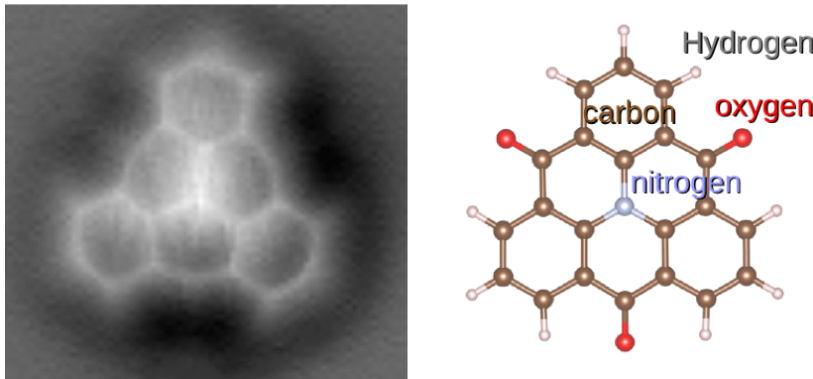


Figure 14.10: AFM image of 1,5,9-trioxo-13-azatriangulene (left) and its chemical structure model (right) [33].

14.5.3 Basic Data Analysis Techniques

Business Intelligence (BI) has evolved over the years:

1. we started to recognize that data could be used to **gain a competitive advantage** at the end of the 19th century;
2. the 1950s saw the first **business database** for decision support;
3. in the 1980s and 1990s, computers and data became increasingly available (**data warehouses, data mining**);
4. in the 2000s, the trend was to take business analytics out of the hands of data miners (and other specialists) and into the hands of **domain experts**,
5. now, **big data** and specialized techniques have arrived on the scene, as have **data visualization, dashboards, and software-as-service**.

Historically, BI has been one of the streams contributing to modern-day data science, via:

- **system of interest:** the commercial realm, specifically, the market of interest;
- **sources of data:** transaction data, financial data, sales data, organizational data;
- **goals:** provide awareness of competitors, consumers and internal activity and use this to support decision making,
- **culture and preferred techniques:** data marts, key performance indicators, consumer behaviour, slicing and dicing, business 'facts'.

But no matter the realm in which we work, the ultimate goal remains the same: **obtaining actionable insight into the system of interest**. This can be achieved in a number of ways. Traditionally, analysts hope to do so by seeking:

- **patterns** – predictable, repeating regularities;
- **structure** – the organization of elements in a system, and
- **generalization** – the creation of general or abstract concepts from specific instances (see Figure 14.10).

The underlying analytical **hope** is to find patterns or structure in the data from which **actionable insights** arise.

While finding patterns and structure can be interesting in its own right (in fact, this is the ultimate reward for many scientists), in the data science context it is how these discoveries are used that trumps all.

Variable Types

In the example of a conceptual model shown on page 891, we have identified different types of variables. In an **experimental setting**, we typically encounter:

- **control/extraneous variables** – we do our best to keep these controlled and unchanging while other variables are changed;
- **independent variables** – we control their values as we suspect they influence the dependent variables,
- **dependent variables** – we do not control their values; they are generated in some way during the experiment, and presumed dependent on the other factors.

For instance, we could be interested in the **plant height** (dependent) given the **mean number of sunlight hours** (independent), given the **region of the country** in which each test site is located (control).

Data Types

Variables need not be of the same **type**. We may encounter:

- **numerical** data – integers or numerics: 1, -7, 34.654, 0.04, etc.;
- **text** data – strings of text, which may be restricted to a certain number of characters, such as “Welcome to the park”, “AAAAA”, “345”, “45.678”, etc.;
- **categorical** data – are variables with a fixed number of values, may be numeric or represented by strings, but for which there is no specific or inherent ordering, such as (‘red’, ‘blue’, ‘green’), (‘1’, ‘2’, ‘3’), etc.,
- **ordinal** data – categorical data with an inherent ordering; unlike **integer** data, the spacing between values is not well-defined (very cold, cold, tepid, warm, super hot).

We use the following artificial dataset to illustrate some of the concepts.

Creating the artificial dataset

```
set.seed(0)      # for replicability
n.sample = 165   # num. of observations

colour=factor(c("red", "blue", "green")) # var 1: colour
p.colour=c(40,15,5)                        # parameters

year=factor(c(2012,2013))                  # var 2: year
p.year=c(60,40)                            # parameters

quarter=factor(c("Q1", "Q2", "Q3", "Q4")) # var 3: quarter
p.quarter=c(20,25,30,35)                   # parameters

signal.mean=c(14, -2, 123)                 # var 4: signal
p.signal.mean=c(5,3,1)                     # parameters
signal.sd=c(2,8,15)
p.signal.sd=c(2,3,4)
```

```

s.colour <- sample(length(colour),      # var 1: colour
                  n.sample,            # sample
                  prob=p.colour,
                  replace=TRUE)

s.year <- sample(length(year),          # var 2: year
                n.sample,              # sample
                prob=p.year,
                replace=TRUE)

s.quarter <- sample(length(quarter),   # var 3: quarter
                   n.sample,           # sample
                   prob=p.quarter,
                   replace=TRUE)

s.mean <- sample(length(signal.mean),   # var 4: signal
                 n.sample,              # sample (mean)
                 prob=p.signal.mean,
                 replace=TRUE)

s.sd <- sample(length(signal.sd),        # var 4: signal
               n.sample,                 # sample (sd)
               prob=p.signal.mean,
               replace=TRUE)

signal <- rnorm(n.sample,               # var 4: signal
               signal.mean[s.mean],     # sample
               signal.sd[s.sd])

new_data <- data.frame(colour[s.colour], # creating a
                      year[s.year],      # data frame
                      quarter[s.quarter],
                      signal)

colnames(new_data) <- c("colour",       # renaming the
                       "year",          # variables
                       "quarter",
                       "signal")

new_data |>
  dplyr::slice_head(n = 6)             # displaying the
                                       # first 6 obs

```

ID	colour	year	quarter	signal
1	blue	2013	Q2	22.998
2	red	2012	Q1	12.456
3	red	2012	Q4	9.935
4	red	2012	Q3	5.047
5	blue	2013	Q2	6.142
6	red	2012	Q4	13.498

(Do you understand what the code does?)

22: A similar approach underlies most of modern text mining, natural language processing, and categorical anomaly detection. Information usually gets lost in the process, which explains why meaningful categorical analyses tend to stay fairly simple.

We can transform categorical into numeric data by generating **frequency counts** of the different levels of the categorical variable; regular analysis techniques is then used on the new numeric variable.²²

```
table(new_data$colour)
```

colour	Freq
blue	41
green	10
red	114

Categorical data plays a special role in data analysis:

- in data science, categorical variables come with a **pre-defined** set of values;
- in experimental science, a **factor** is an independent variable with its levels being defined (it may also be viewed as a category of treatment),
- in business analytics, these are called **dimensions** (with members).

However they are labeled, these variable can be used to **subset** or **roll up/summarize** the data.

Hierarchical / Nested / Multilevel Data

When a categorical variable has multiple levels of abstraction, new categorical variables can be created from these levels. We can view these levels as new categorical variables, in a sense. The ‘new’ categorical variable has pre-defined relationships with the more detailed level.

This is commonly the case with time and space variables – we can ‘zoom’ in or out, as needed, which allows us discuss the **granularity** of the data, i.e., the ‘maximum zoom factor’ of the data. For instance, observations could be recorded hourly, and then further processed (mean value, total, etc.) at the daily level, the monthly level, the quarterly level, the yearly level, etc., as seen below.

Let us start with the number of observations by year and quarter:

```
library(tidyverse)    # to be able to use
                      # group_by() and summarise()
new_data |>
  group_by(year, quarter) |>
  summarise(n = n())
```

year	quarter	n	year	quarter	n
2012	Q1	21	2013	Q1	14
2012	Q2	17	2013	Q2	11
2012	Q3	30	2013	Q3	20
2012	Q4	37	2013	Q4	15

We can also roll it up to the number of observations by year:

```
new_data |>           # no need to load tidyverse again
  group_by(year) |>
  summarise(n = n())
```

year	n
2012	105
2013	60

Data Summaries

The **summary statistics** of variables can help analysts gain basic **univariate insights** into the dataset (and hopefully, into the system with which it is associated).

These data summaries do not typically provide the full picture and connections/links between different variables are often missed altogether. Still, they often give analysts a **reasonable sense** for the data.²³

23: At least for a **first pass**.

Common summary statistics include:

- **min** – smallest value taken by a variable;
- **max** – largest value taken by a variable;
- **median** – “middle” value taken by a variable;
- **mean** – average value taken by a variable;
- **mode** – most frequent value taken by a variable;
- **# of obs** – number of observations for a variable;
- **missing values** – # of missing observations for a variable;
- **# of invalid entries** – number of invalid entries for a variable;
- **unique values** – unique values taken by a variable;
- **quartiles, deciles, centiles;**
- **range, variance, standard deviation;**
- **skew, kurtosis,**
- **total, proportion, etc.**

We can also perform operations over subsets of the data – typically over its columns, in effect **compressing** or **‘rolling up’** multiple data values into a single **representative value**, as below, say.

We start by creating a mode function (there isn’t one in R):

Defining the mode function

```
mode.R <- function(x) {
  unique.x <- unique(x)
  unique.x[which.max(tabulate(match(x, unique.x)))]
}
```

Data scientists often have to create their own routines/functions from scratch; there is nothing wrong with borrowing from sites such as StackOverflow, but it is important to make sure that we understand what those routines do.

The data can then also be summarized using:

Summarizing the data I

```
new_data |>      # no need to load tidyverse anew
  summarise(n = n(),
            signal.mean=mean(signal),
            signal.sd=sd(signal),
            colour.mode=mode.R(colour))
```

n	signal.mean	signal.sd	colour.mode
165	20.70894	38.39866	red

Typical roll-up functions include the ‘mean’, ‘sum’, ‘count’, and ‘variance’, but these do not always give sensible outcomes: if the variable measures a proportion, say, the sum of that variable over all observations is a meaningless quantity, on its own.

We can apply the same roll-up function to many different columns, thus providing a **mapping** (list) of columns to values (as long as the computations all make sense – this might mean that all variables need to be of the same type in some cases).

We can map the mode to some dataset variables:

Summarizing the data II

```
new_data |>      # still no need to re-load the tidyverse
  summarise(year.mode=mode.R(year),
            quarter.mode=mode.R(quarter),
            colour.mode=mode.R(colour))
```

year.mode	quarter.mode	colour.mode
2012	Q4	red

Datasets can also be summarized *via* contingency and pivot tables. A **contingency table** is used to examine the relationship between two **categorical** variables – specifically the frequency of one variable relative to a second variable (this is also known as **cross-tabulation**).

Here is a contingency table, by colour and year:

Contingency table (by colour and year)

```
table(new_data$colour, new_data$year)
```

	2012	2013
blue	21	20
green	6	4
red	78	36

A contingency table, by colour and quarter:

Contingency table (by colour and quarter)

```
table(new_data$colour, new_data$quarter)
```

	Q1	Q2	Q3	Q4
blue	5	8	16	12
green	2	0	5	3
red	28	20	29	37

A contingency table, by year and quarter:

Contingency table (by year and quarter)

```
table(new_data$year, new_data$quarter)
```

	Q1	Q2	Q3	Q4
2012	21	17	30	37
2013	14	11	20	15

A **pivot table**, on the other hand, is a table generated in a software application by applying operations (e.g. sum, count, mean) to variables, possibly based on another (categorical) variable. Here is a pivot table of signal characteristics by colour:

Pivot table (signal characteristics by colour)

```
new_data |> group_by(colour) |>
  summarise(n = n(),
            signal.mean=mean(signal),
            signal.sd=sd(signal))
```

colour	n	signal.mean	signal.sd
blue	41	25.58772	40.64504
green	10	30.79947	49.71225
red	114	18.06916	36.51887

Contingency tables are special instances of pivot tables, where the roll-up function is count.

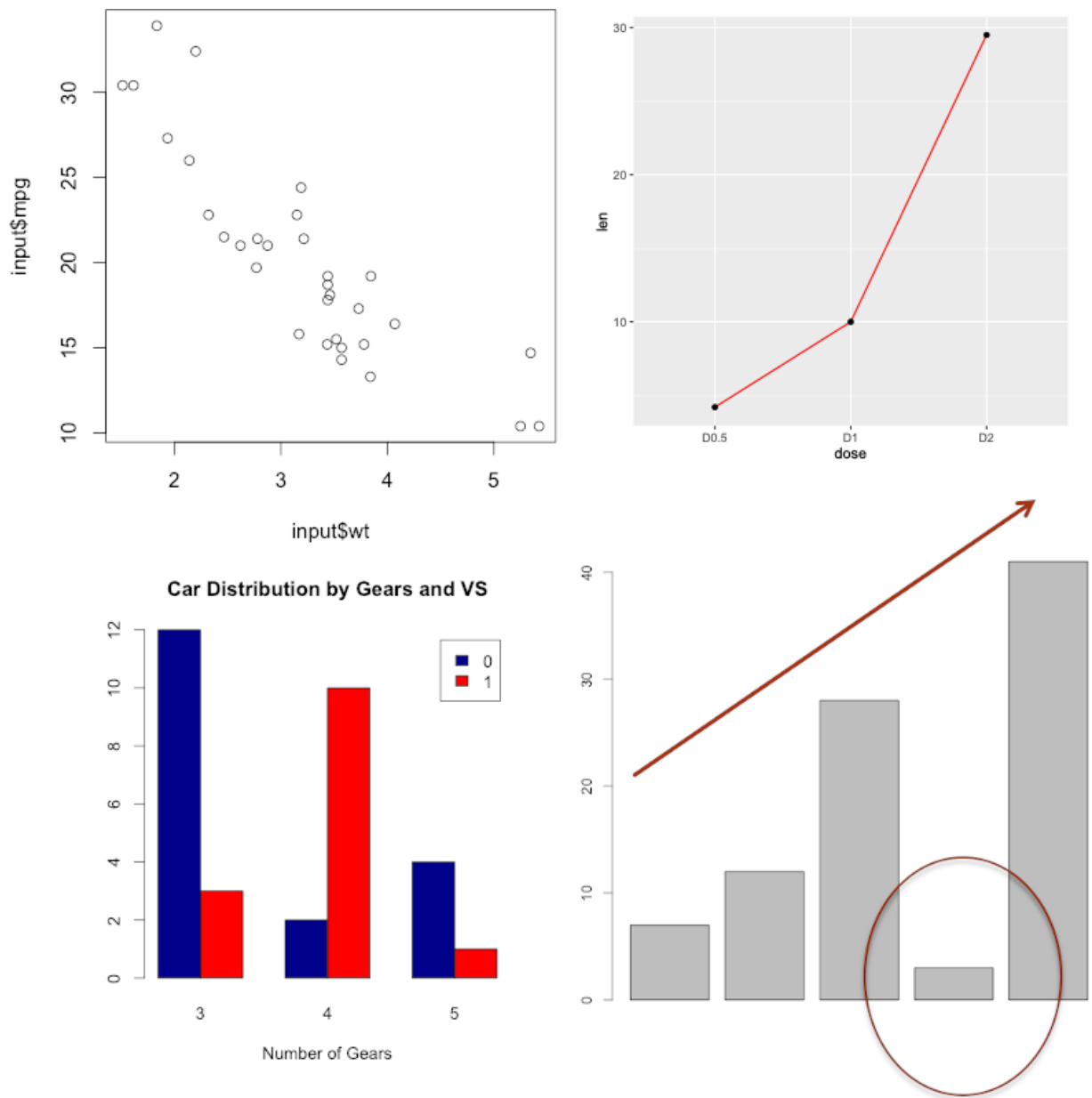


Figure 14.11: Analysis and pattern-reveal through visualization.

Analysis Through Visualization

Consider the broad definition of analysis as:

- identifying patterns or structure, and
- adding meaning to these patterns or structure by **interpreting** them in the context of the system.

There are two general options to achieve this:

1. use analytical methods of varying degrees of sophistication, and/or
2. **visualize** the data and use the brain's analytic (perceptual) power to reach meaningful conclusions about these patterns.

At this point, we will only list some simple visualization methods that are often (but not always) used to reveal patterns (see Figure 14.11):

- **scatter plots** are probably best suited for two numeric variables;
- **line charts**, for numeric variable and ordinal variable;
- **bar charts** for one categorical and one numeric, or multiple categorical/nested categorical data,
- **boxplots, histograms, bubble charts, small multiples**, etc.

An in-depth discussion of data visualization is given in Chapter 18 (*Data Visualization*); best practices and a more complete catalogue are provided in [12].

14.5.4 Common Statistical Procedures in R

The underlying goal of **statistical analysis** is to reach an **understanding of the data**. In this section, we show how some of the most common **basic** statistical concepts that can help analysts reach that goal are implemented in R; a more thorough treatment of probability and statistics notions can be found in Chapters 6 (*Probability and Applications*), 7 (*Introduction to Statistical Analysis*), and 8 (*Classical Regression Analysis*).

Once the data is properly organized and visual exploration has begun in earnest, the typical next step is to describe the distribution of each variable numerically, followed by an exploration of the relationships among selected variables.

The objective is to answer questions such as:

- What kind of mileage are cars getting these days? Specifically, what's the distribution of miles per gallon (mean, standard deviation, median, range, and so on) in a survey of automobile makes and models?
- After a new drug trial, what is the outcome (no improvement, some improvement, marked improvement) for drug versus placebo groups? Does the sex of the participants have an impact on the outcome?
- What is the correlation between income and life expectancy? Is it significantly different from zero?
- Are you more likely to receive imprisonment for a crime in different regions of Canada? Are the differences between regions statistically significant?

Basic Statistics

When it comes to calculating **descriptive statistics**, R can basically do it all. We start with functions that are included in the base installation. We will then look for extensions that are available through the use of user-contributed packages.

For illustrative purposes, we will use several of the variables from the *Motor Trend Car Road Tests* (mtcars) dataset provided in the base installation: we will focus on miles per gallon (mpg), horsepower (hp), and weight (wt):

```
myvars <- c("mpg", "hp", "wt")
head(mtcars[myvars])
```

	mpg	hp	wt
Mazda RX4	21.0	110	2.620
Mazda RX4 Wag	21.0	110	2.875
Datsun 710	22.8	93	2.320
Hornet 4 Drive	21.4	110	3.215
Hornet Sportabout	18.7	175	3.440
Valiant	18.1	105	3.460

Let us first take a look at descriptive statistics for all 32 models. In the base installation, we can use the `summary()` function.

```
summary(mtcars[myvars])
```

mpg	hp	wt
Min.: 10.40	Min.: 52.0	Min.: 1.513
1st Qu.: 15.43	1st Qu.: 96.5	1st Qu.: 2.581
Median: 19.20	Median: 123.0	Median: 3.325
Mean: 20.09	Mean: 146.7	Mean: 3.217
3rd Qu.: 22.80	3rd Qu.: 180.0	3rd Qu.: 3.610
Max.: 33.90	Max.: 335.0	Max.: 5.424

The `summary()` function provides the minimum, maximum, quartiles, and mean for numerical variables, and the respective frequencies for factors and logical vectors. In base R, the functions `apply()` or `sapply()` can be used to provide any descriptive statistics. The format in use is:

```
sapply(x, FUN, options)
```

where x is the data frame and `FUN` is an arbitrary function. Optional arguments are passed to `FUN`. Typical functions include:

- `mean()`
- `sd()`
- `var()`

- `min()`
- `max()`
- `median()`
- `length()`
- `range()`
- `quantile()`
- `fivenum()`

The next example provides several descriptive statistics using `sapply()`, including the **skew** and the **kurtosis**.

```
mystats <- function(x, na.omit=FALSE){
  if (na.omit){
    x <- x[!is.na(x)]
  }
  m <- mean(x); n <- length(x); s <- sd(x)
  skew <- sum((x-m)^3/s^3)/n; kurt <- sum((x-m)^4/s^4)/n - 3
  return(c(n=n, mean=m, stdev=s, skew=skew, kurtosis=kurt))}
```

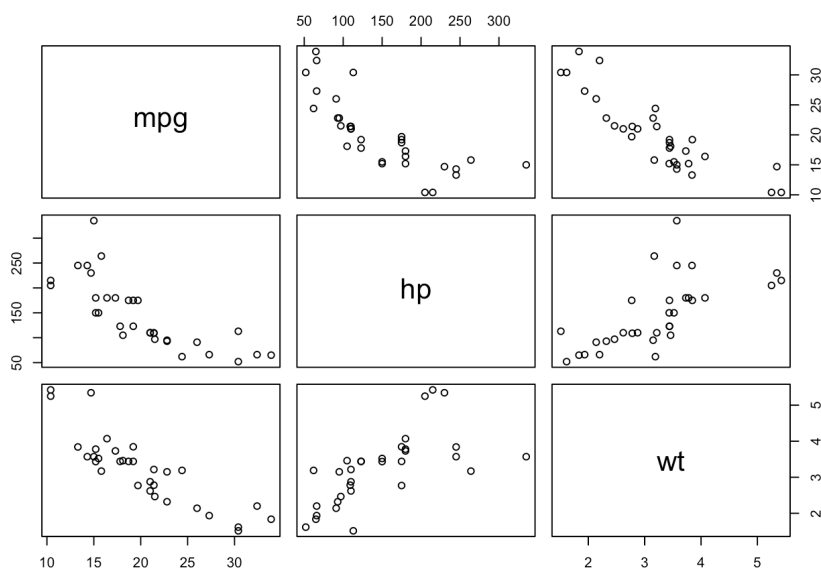
Let us apply `mystats()` to the data frame of interest.

```
sapply(mtcars[myvars], mystats)
```

	mpg	hp	wt
n	32	32	32
mean	20.090625	146.6875	3.21725
stdev	6.026948	68.5628685	0.9784574
skew	0.610655	0.7260237	0.4231465
kurtosis	-0.372766	-0.1355511	-0.0227108

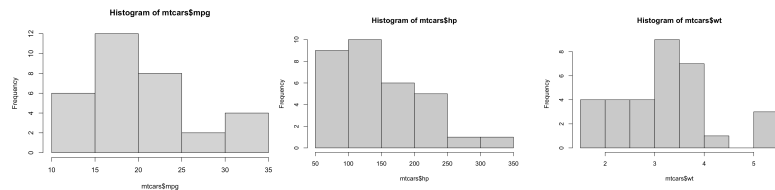
We can plot the pairwise scatterplots for the three variables.

```
plot(mtcars[myvars])
```



For cars in this sample, the mean mpg is 20.1, with a standard deviation of 6.0. The distribution is skewed to the right (+0.61) and is somewhat flatter than a normal distribution (−0.37). This is most evident if we build histograms of the data.

```
hist(mtcars$mpg)
hist(mtcars$hp)
hist(mtcars$wt)
```



To omit missing values for the computations, we would use the option `na.omit=TRUE`.

Since there are no missing observations in the dataset, we create a version of `mtcars` with some missing values, then we provide a `mystats()` summary.

Adding missing values

```
my.mtcars <- mtcars
my.mtcars[2,1] <- NA
my.mtcars[17,1] <- NA
sapply(my.mtcars[myvars], mystats, na.omit=TRUE)
```

	mpg	hp	wt
n	30	32	32
mean	20.24	146.6875	3.21725
stdev	6.1461847	68.5628685	0.9784574
skew	0.5660728	0.7260237	0.4231465
kurt	−0.4870340	−0.1355511	−0.0227108

Notice the changes in the mpg summary.

The same table can be obtained using the `dplyr` package functions instead (`skewness()` and `kurtosis()` are available in `e1071` package).

```
mpg = dplyr::summarise(mtcars, n=n(), mean=mean(mpg),
                      stdev=sd(mpg), skew=e1071::skewness(mpg),
                      kurt=e1071::kurtosis(mpg))
hp = dplyr::summarise(mtcars, n=n(), mean=mean(hp),
                     stdev=sd(hp), skew=e1071::skewness(hp),
                     kurt=e1071::kurtosis(hp))
wt = dplyr::summarise(mtcars, n=n(), mean=mean(wt),
                     stdev=sd(wt), skew=e1071::skewness(wt),
                     kurt=e1071::kurtosis(wt))
```

```
pivot = t(rbind(mpg, hp, wt))
colnames(pivot) <- c("mpg", "hp", "wt")
```

	mpg	hp	wt
n	32	32	32
mean	20.090625	146.6875	3.21725
stdev	6.026948	68.5628685	0.9784574
skew	0.610655	0.7260237	0.4231465
kurt	-0.372766	-0.1355511	-0.0227108

Hmisc and pastecs

Several packages offer functions for descriptive statistics, including `Hmisc` and `pastecs` (as do `dplyr` and `e1071`).

`Hmisc`'s `describe()` function returns the number of variables and observations, the number of missing and unique values, the mean, quantiles, and the five highest and lowest values.

```
Hmisc::describe(mtcars[myvars])
```

```
mtcars[myvars]
```

```
3 Variables      32 Observations
-----
mpg
  n missing distinct    Info    Mean      Gmd      .05      .10
  32      0       25    0.999   20.09    6.796   12.00   14.34
  .25     .50     .75     .90     .95
 15.43   19.20   22.80   30.09   31.30

lowest : 10.4 13.3 14.3 14.7 15.0, highest: 26.0 27.3 30.4 32.4 33.9
-----
hp
  n missing distinct    Info    Mean      Gmd      .05      .10
  32      0       22    0.997  146.7    77.04   63.65   66.00
  .25     .50     .75     .90     .95
 96.50  123.00  180.00  243.50  253.55

lowest : 52 62 65 66 91, highest: 215 230 245 264 335
-----
wt
  n missing distinct    Info    Mean      Gmd      .05      .10
  32      0       29    0.999   3.217    1.089   1.736   1.956
  .25     .50     .75     .90     .95
 2.581   3.325   3.610   4.048   5.293

lowest : 1.513 1.615 1.835 1.935 2.140, highest: 3.845 4.070 5.250 5.345 5.424
-----
```

The `pastecs` package includes the function `stat.desc()` that provides a wide range of descriptive statistics:

```
stat.desc(x, basic=TRUE, desc=TRUE, norm=FALSE, p=0.95)
```

where x is a data frame or a time series.

If `basic=TRUE` (the default), the number of values, null values, missing values, minimum, maximum, range, and sum are provided. If `desc=TRUE` (also the default), the median, mean, standard error of the mean, 95% confidence interval for the mean, variance, standard deviation, and coefficient of variation are also provided. Finally, if `norm=TRUE` (not the default), normal distribution statistics are returned, including skewness and kurtosis (with statistical significance) and the Shapiro–Wilk test of normality.

The p option is used to calculate the confidence interval for the mean (.95 by default).

For instance, we may obtain:

```
paste0(":", stat.desc(mtcars[myvars]))
```

	mpg	hp	wt
nbr.val	32	32	32
nbr.null	0	0	0
nbr.na	0	0	0
min	10.4	52	1.513
max	33.9	335	5.424
range	23.5	283	3.911
sum	642.9	4694	102.952
median	19.2	123	3.325
mean	20.0906250	146.6875000	3.2172500
SE.mean	1.0654240	12.1203173	0.1729685
CI.mean.0.95	2.1729465	24.7195501	0.3527715
var	36.3241028	4700.8669355	0.9573790
std.dev	6.0269481	68.5628685	0.9784574
coef.var	0.2999881	0.4674077	0.3041285

We take this opportunity to caution users against relying too heavily on a single (or even a few) specific packages.

Correlations

Correlation coefficients are used to describe relationships among quantitative variables. The sign \pm indicates the direction of the relationship (positive or inverse), and the magnitude indicates the strength of the relationship (0: no linear relationship; 1: perfect linear relationship).

In this section, we look at a variety of correlation coefficients, as well as tests of significance. We will use the `state.x77` dataset available in the base R installation. It provides data on the population, income, illiteracy rate, life expectancy, murder rate, and high school graduation rate for the 50 US states in 1977. There are also temperature and land-area measures, but we will not be using them.²⁴

24: In addition to the base installation, we will also be using the `psych` and `ggm` packages.

R can produce a variety of correlation coefficients, including:

- **Pearson's product-moment** coefficient (degree of linear relationship between two quantitative variables);
- **Spearman's rank-order** coefficient (degree of relationship between two rank-ordered variables), and
- **Kendall's tau** coefficient (nonparametric measure of rank correlation).

The `cor()` function produces all three correlation coefficients, whereas the `cov()` function provides covariances. There are many options, but a simplified format for producing correlations is

```
cor(x, use=OPT , method=METHOD)
```

where *x* is a matrix or a data frame, and *use* specifies the handling of missing data; its options are

- `all.obs` (assumes no missing data);
- `everything` (any correlation involving a case with missing values will be set to missing);
- `complete.obs` (listwise deletion), and
- `pairwise.complete.obs` (pairwise deletion).

The *method* specifies the type of correlation; its options are `pearson`, `spearman`, and `kendall`.

The default options are `use = "everything"` and `method = "pearson"`.

For the built-in dataset `state.x77`, which contains socio-demographic information about the 50 U.S. states from 1977, we find the following correlations:

Correlations in the `state.x77` data

```
states <- state.x77[,1:6]
cor(states)
```

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad
Population	1.0000000	0.2082276	0.1076224	-0.0680520	0.3436428	-0.0984897
Income	0.2082276	1.0000000	-0.4370752	0.3402553	-0.2300776	0.6199323
Illiteracy	0.1076224	-0.4370752	1.0000000	-0.5884779	0.7029752	-0.6571886
Life Exp	-0.0680520	0.3402553	-0.5884779	1.0000000	-0.7808458	0.5822162
Murder	0.3436428	-0.2300776	0.7029752	-0.7808458	1.0000000	-0.4879710
HS Grad	-0.0984897	0.6199323	-0.6571886	0.5822162	-0.4879710	1.0000000

This produces the Pearson product-moment correlation coefficients. We can see, for example, that a strong positive correlation exists between income and HS Grad rate and that a strong negative correlation exists between Illiteracy and Life Exp.

A **partial correlation** is a correlation between two quantitative variables, controlling for one or more other quantitative variables;²⁵ the `pcor()` function in the `ggm` package provides partial correlation coefficients (this

25: The use of partial correlations is common in the social sciences, but not so much in other fields.

package is not installed by default, so it must be installed before first use). The format is:

```
pcor(u, S)
```

where *u* is a vector of integers, with the

- first two entries representing the indices of the variables to be correlated, and
- remaining numbers being the indices of the conditioning variables (that is, the variables being **partialled out**),

and where *S* is the covariance matrix among the variables.

Partial correlations in the state.x77 data I

```
colnames(states)
ggm::pcor(c(1,5,2,3,6), cov(states))
```

```
"Population" "Income" "Illiteracy" "Life Exp" "Murder" "HS Grad"
0.3462724
```

In this case, 0.346 is the correlation between Population (variable 1) and the Murder rate (variable 5), controlling for the influence of Income, Illiteracy, and HS Grad (variables 2, 3, and 6 respectively).

We see that the partial correlations only change slightly if we condition against a different subset of values.

Partial correlations in the state.x77 data II

```
ggm::pcor(c(1,5,2,3), cov(states))
ggm::pcor(c(1,5,2), cov(states))
```

```
0.3621683
0.4113621
```

How do these three values compare to the direct correlation between Population and Murder?

Simple Linear Regression

In many ways, **regression analysis** is at the heart of statistics. It is a broad term for a set of methodologies used to predict a response variable (also called a dependent, criterion, or outcome variable) from one or more predictor variables (also called independent or explanatory variables).

In general, regression analysis can be used to:

- identify the explanatory variables that relate to a response;
- describe the form of the relationships involved, and
- provide an equation for predicting the response variable from the explanatory variables.

For example, an exercise physiologist might use regression analysis to develop an equation for predicting the expected number of calories a person will burn while exercising on a treadmill.

In this example, the response variable is the number of calories burned (calculated from the amount of oxygen consumed), say, and the predictor variables might include:

- duration of exercise (minutes);
- percentage of time spent at their target heart rate;
- average speed (mph);
- age (years);
- gender, and
- body mass index (BMI).

From a practical point of view, regression analysis could help answer questions such as:

- how many calories can a 30-year-old man with a BMI of 28.7 expect to burn if he walks for 45 minutes at an average speed of 4 miles per hour and stays within his target heart rate 80% of the time?
- what's the minimum number of variables needed in order to accurately predict the number of calories a person will burn when walking?

R has powerful and comprehensive features for fitting regression models – the abundance of options can be confusing. The basic function for fitting a linear model is `lm()`. The format is

```
fit <- lm(formula, data)
```

where `formula` describes the model to fit and `data` is the data frame containing the data that is used in fitting the model. The resulting object (`fit`, in this case) is a list that contains extensive information about the fitted model.

The formula is typically written as

$$Y \sim X_1 + X_2 + \dots + X_k$$

where \sim separates the response variable on the left from the predictor variables on the right, and the predictor variables are separated by $+$ symbols.

In addition to `lm()`, there are several functions that are useful when generating regression models. Each of these functions is applied to the object returned by `lm()` in order to generate additional information based on the fitted model.

As an example, the `women` dataset in R's base installation provides the heights and weights for a set of 15 women aged 30 to 39. Assume that we are interested in predicting the weight of an individual from their height.²⁶

26: An equation for predicting weight from height could help identifying individuals who are possibly overweight (or underweight), say.

Function	Action
<code>summary()</code>	Displays detailed results for the fitted model
<code>coefficients()</code>	Lists the model parameters (intercept and slopes) for the fitted model
<code>confint()</code>	Provides confidence intervals for the model parameters (95% by default)
<code>residuals()</code>	Lists the residual values in a fitted model
<code>anova()</code>	Generates an ANOVA table for a fitted model, or to compare 2+ fitted models
<code>plot()</code>	Generates diagnostic plots for evaluating the fit of a model
<code>fitted()</code>	Extracts the fitted values for the dataset
<code>predict()</code>	Uses a fitted model to predict response values for a new dataset

The linear regression on the data is obtained as follows:

Regression on the women dataset

```
fit <- lm(weight ~ height, data=women)
summary(fit)
```

Call:

```
lm(formula = weight ~ height, data = women)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-1.7333 -1.1333 -0.3833  0.7417  3.1167
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***
height       3.45000    0.09114   37.85 1.09e-14 ***
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 1.525 on 13 degrees of freedom

Multiple R-squared: 0.991, Adjusted R-squared: 0.9903

F-statistic: 1433 on 1 and 13 DF, p-value: 1.091e-14

From the output, you see that the prediction equation is

$$\widehat{\text{weight}} = -87.52 + 3.45 \times \text{height}.$$

Because a height of 0 is impossible, there is no sense in trying to give a physical interpretation to the intercept – it merely becomes an adjustment constant (in other words, 0 is **not in the domain** of the model).

From the $P(>|t|)$ column, we see that the regression coefficient (3.45) is significantly different from zero ($p < 0.001$), which indicates that there's an expected increase of 3.45 pounds of weight for every 1 inch increase in height. The multiple R-squared coefficient (0.991) indicates that the model accounts for 99.1% of the variance in weights.

The individual weights (in pound) are:

```
women$weight
```

```
115 117 120 123 126 129 132 135 139 142 146 150 154 159 164
```

and their fitted values (and residuals) are

```
fitted(fit)
residuals(fit)
```

fitted:

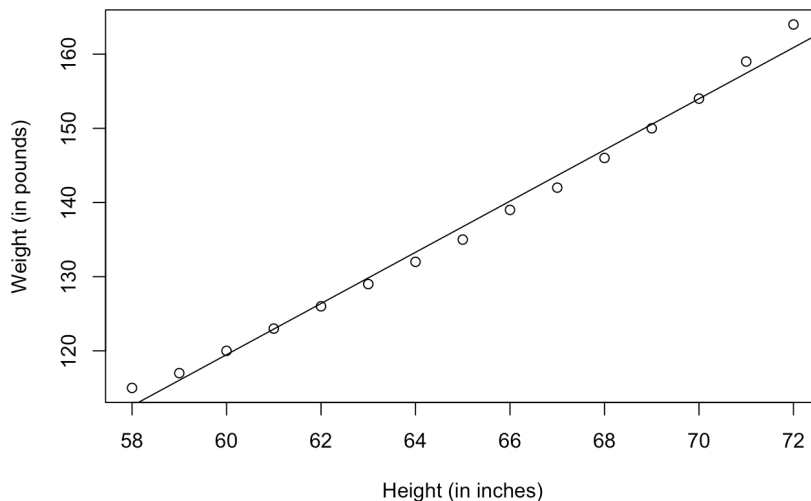
1	2	3	4	5	6	7	8
112.5833	116.0333	119.4833	122.9333	126.3833	129.8333	133.2833	136.7333
9	10	11	12	13	14	15	
140.1833	143.6333	147.0833	150.5333	153.9833	157.4333	160.8833	

residuals:

1	2	3	4	5	6	7	8
2.4167	0.9667	0.5167	0.0667	-0.3833	-0.8333	-1.2833	-1.7333
9	10	11	12	13	14	15	
-1.1833	-1.6333	-1.0833	-0.5333	0.0167	1.5667	3.1167	

We can see that the linear fit is decent (although the residual structure suggests that a quadratic fit would probably be better).

```
plot(women$height, women$weight,
     xlab="Height (in inches)", ylab="Weight (in lbs)")
abline(fit)
```



Bootstrapping

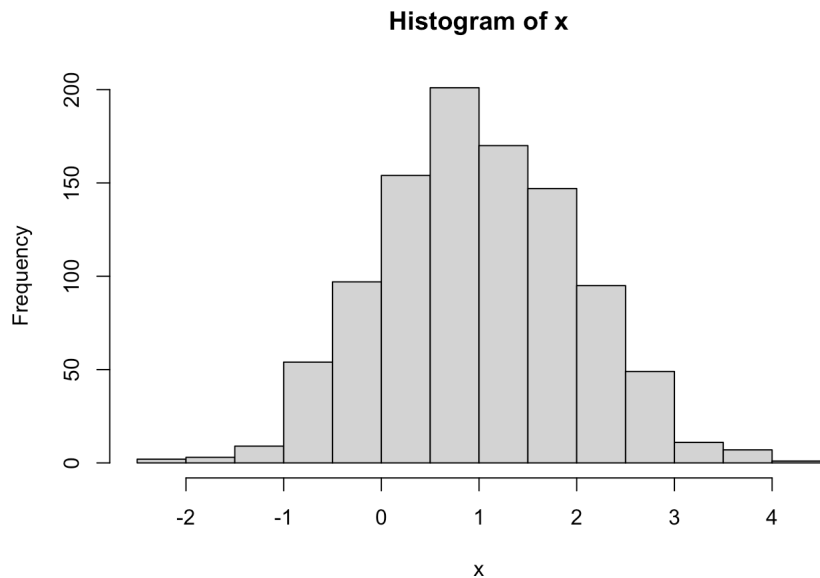
Bootstrapping is a powerful and elegant approach to estimating the sampling distribution of specific statistics. It can be implemented in many situations where asymptotic results are difficult to find or otherwise unsatisfactory.

Bootstrapping proceeds using three steps:

1. resample the dataset (with replacement) many times over (typically on the order of 10,000);
2. calculate the desired statistic from each resampled dataset,
3. use the distribution of the resampled statistics to estimate the standard error of the statistic (normal approximation method) or construct a confidence interval using quantiles of that distribution (percentile method).

There are several ways to bootstrap in R. As an example, say that we want to estimate the standard error and 95% confidence interval for the **coefficient of variation** (CV), defined as σ/μ , for a random variable X . We illustrate the procedure with 1000 generated values of $X \sim \mathcal{N}(1, 1)$.

```
set.seed(0)           # for replicability
x = rnorm(1000, mean=1, sd=1)
hist(x)
```



On this sample, the coefficient of variation is:

```
(cv=sd(x)/mean(x))
```

1.014057

We must define a function to compute the statistic of interest in R.

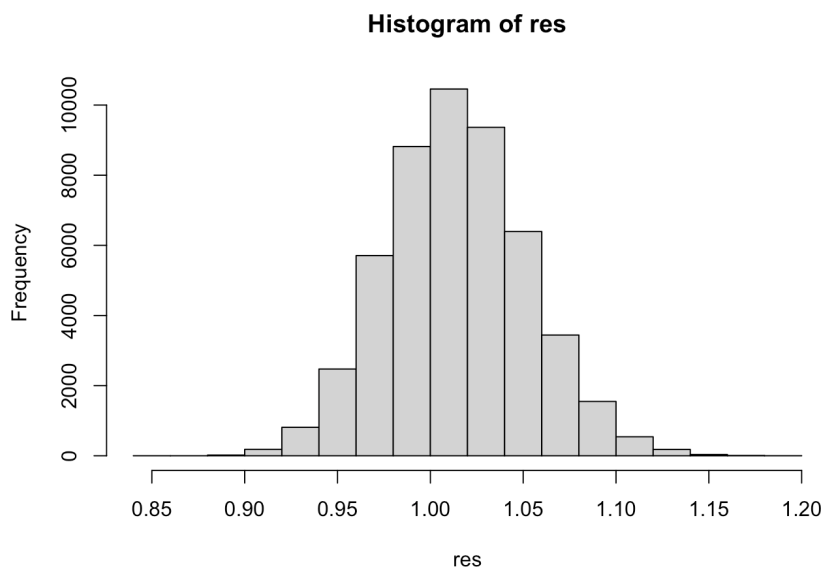
Defining the coefficient of variation in R

```
cvfun = function(x) {
  return(sd(x)/mean(x))
}
```

The `replicate()` function is the base R tool for repeating function calls. We nest a call to `cvfun()` and a call to sample the data with replacement using the `sample()` function (with 50000 replicates).

Bootstrap distribution of CV(x)

```
res = replicate(50000, cvfun(sample(x, replace=TRUE)))
hist(res)
```



We can also compute quantiles, as below:

95% Confidence Interval for CV(x)

```
quantile(res, c(.025, .975))
```

```
      2.5%      97.5%
0.9432266 1.0917185
```

This seems reasonable, as we would expect the CVs to be centered around 1, given that $\mu = \sigma = 1$ (in this example).

The percentile interval is easy to calculate from the observed bootstrapped statistics. If the distribution of the bootstrap samples is approximately normally distributed, a t -interval could be created by calculating the standard deviation of the bootstrap samples and finding the appropriate multiplier for the confidence interval. Plotting the bootstrap sample estimates is helpful to determine the form of the bootstrap distribution.

The framework can also be extended to include **non-linear** models, **correlated variables**, **probability estimation**, and/or **multivariate** models; any book on statistical analysis contains at least one chapter or two on the topic (see [37, 11], for instance).

We will not pursue the topic further except to say that regression analysis and bootstrapping are two of the arrows that every data scientist should have in their quiver.

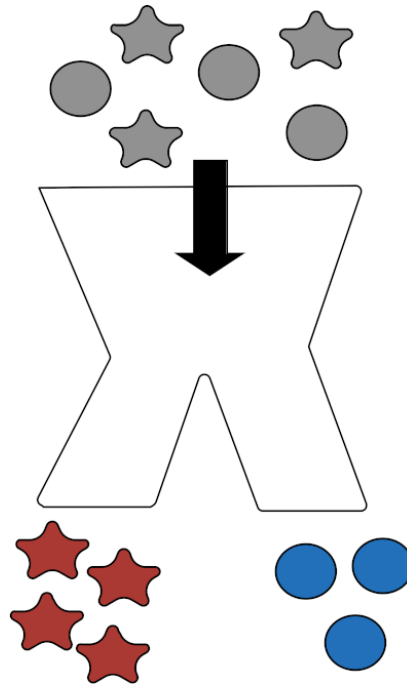


Figure 14.12: The trousers of classification.

14.5.5 Quantitative Methods

We provided a list of quantitative methods in Section 14.4.2 (*Data Collection, Storage, Processing, and Modeling*); we finish this section by expanding on a few of them.

Classification and Supervised Learning Tasks

Classification is one of the cornerstones of machine learning. Instead of trying to predict the numerical value of a response variable (as in regression), a **classifier** uses **historical data**²⁷ to identify general patterns that could lead to observations belonging to one of several **pre-defined categories**.

For instance, if a car insurance company only has resources to investigate up to 20% of all filed claims, it could be useful for them to predict:

- whether a claim is likely to be fraudulent;
- whether a customer is likely to commit fraud in the near future;
- whether an application for a policy is likely to result in a fraudulent claim,
- the amount by which a claim will be reduced if it is fraudulent, etc.

Analysts and machine learning practitioners use a variety of different techniques to carry this process out (see Figure 14.12 for an illustration, and Chapters 19 (*Introduction to Machine Learning*) and 21 (*Focus on Classification*), as well as [34, 3, 2] for more details), but the general steps always remain the same:

1. use **training data** to teach the classifier;
2. test/validate the classifier using **hold-out data**,
3. if it passes the test, use the classifier to classify **novel instances**.

27: This training data usually consists of a **randomly** selected subset of the **labeled** (response) data.

Some classifiers (such as deep learning neural nets) are ‘**black boxes**’: they might be very good at classification, but they are not **explainable**. In some instances, that is an acceptable side effect of the process, in others, it might not be – if an individual is refused refugee status, say, they might rightly want to know **why**.

Unsupervised Learning Techniques

The hope of artificial intelligence is that intelligent behaviours will eventually be able to be **automated**. For the time being, however, that is still very much a work in progress.²⁸

Classification, for instance, is the prototypical supervised task: can we learn from historical/training examples? It seems like a decent approach to learning: evidence should drive the process.

But there are limitations to such an approach: it is difficult to make a **conceptual leap** solely on the basis of training data [if our experience in learning is anything to go by. . .], if only because the training data might not be representative of the system, or because the learner target task is **too narrow**.

In **unsupervised** learning, we learn without examples, based solely on what is found in the data. There is no specific question to answer (in the classification sense), other than “what can we learn from the data?”

Typical unsupervised learning tasks include:

- **clustering** (finding novel categories);
- **association rules mining**,
- **recommender systems**, etc.

For instance, an online bookstore might want to make recommendations to customers concerning additional items to browse (and hopefully purchase) based on their buying patterns in prior transactions, the similarity between books, and the similarity between **customer segments**:

- but what are those patterns?
- how do we measure similarity?
- what are the customer segments?
- can any of that information be used to create promotional bundles?
- etc.

The lack of a specific target makes unsupervised learning much more **difficult** than supervised learning, as does the challenges of **validating the results**.²⁹

More general information and details on clustering can be found in Chapters 19 (*Introduction to Machine Learning*) and 22 (*Focus on Clustering*), as well as in [4, 2, 77].

28: One of the challenges in that process is that not every intelligent behaviour arises from a supervised process.

29: This contributes to the proliferation of clustering algorithms and cluster quality metrics.

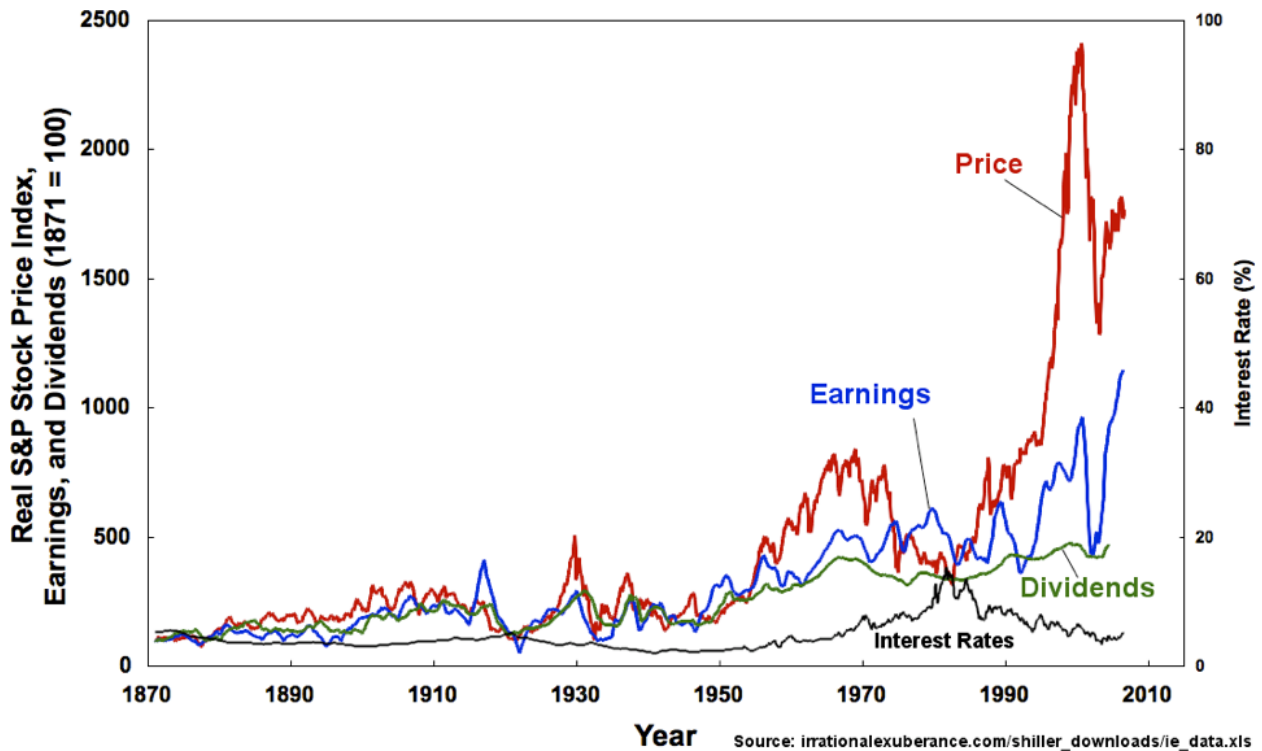


Figure 14.13: Real S&P stock price index (red), earnings (blue), and dividends (green), together with interest rates (black), from 1871 to 2009 [R.J. Shiller].

Other Machine Learning Tasks

These scratch but a **minuscule** part of the machine learning ecosystem. Other common tasks include [59]:

- profiling and behaviour description;
- link prediction;
- data reduction,
- influence/causal modeling, etc.

to say nothing of more sophisticated learning frameworks (semi-supervised learning, reinforcement learning [72], deep learning [30], etc.).

Time Series Analysis and Process Monitoring

Processes are often subject to **variability**:

- variability due the **cumulative effect** of many small, essentially unavoidable causes, such as regular variations in the weather or in the quality of materials (a process that only operates with such **common causes** is said to be **in (statistical) control**),
- variability due to **special causes**, such as improperly adjusted machines, poorly trained operators, defective materials, etc. (the variability is typically much larger for special causes, and such processes are said to be **out of (statistical) control**).

The aim of **statistical process monitoring** (SPM) is to identify occurrence of special causes. This is often done *via* **time series analysis**.

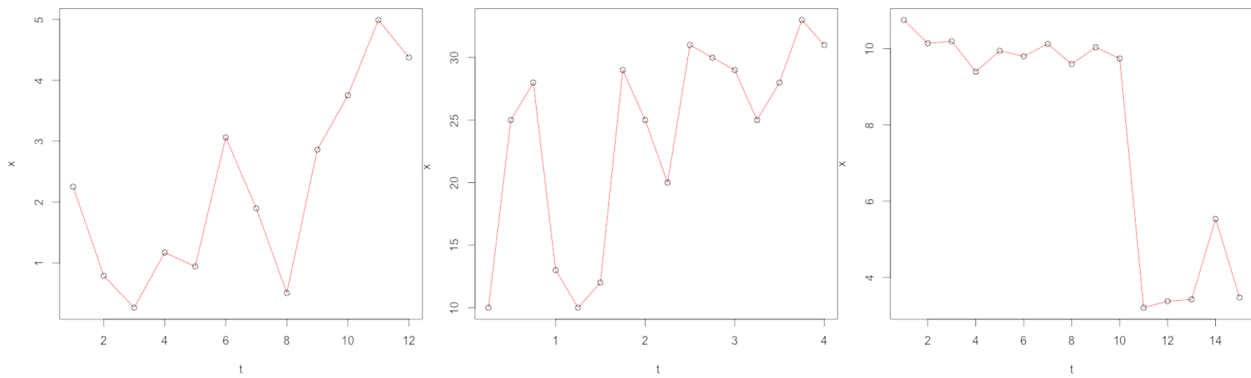


Figure 14.14: Sales (in \$10,000's) for 3 different products – years (left), quarters (middle, but labeled in years), weeks (right).

Consider n observations $\{x_1, \dots, x_n\}$ arising from some collection of processes. In practice, the index i is often a **time index** or a **location index**, i.e., the x_i are observed in **sequence** or in **regions**.³⁰ The processes that generate the observations could change from one time/location to the next due to:

- **external factors** (war, pandemic, election results, etc.), or
- **internal factors** (policy changes, modification of manufacturing process, etc.).

In such case, the mean and standard deviation alone might not provide a useful summary of the situation.

To get a sense of what is going on with the data (and the associated system), it could prove preferable to **plot the data in the order that it has been collected** (or according to geographical regions, or both). The horizontal coordinate would then represent:

- the **time of collection** t (order, day, week, quarter, year, etc.), or
- the **location** i (country, province, city, branch, etc.).

The vertical coordinate represents the observations of interest x_t or x_i (see Figure 14.13 for an example). In process monitoring terms, we may be able to identify potential special causes by identifying **trend breaks**, **cycles discontinuities**, or **level shifts** in time series.

For instance, consider the three time series of Figure 14.14.

- in the first example (left), there are occasional drops in sales from one year to the next, but the **upward trend** is clear – we see the importance of considering the full time series; if only the last two points are presented to stockholders, they might conclude that action is needed, whereas the whole series paints a rosier outlook;
- in the second case (middle), there is a **cyclic effect** with increases from Q1 to Q2 and from Q2 to Q3, but decreases from Q3 to Q4 and from Q4 to Q1. Overall, we also see an upward trend – the presence of regular patterns is a positive development,
- finally, in the last example (right), something clearly happened after the tenth week, causing a **trend level shift**. Whether it is due to internal or external factors depends on the context, which we do not have at our disposal, but some action certainly seems to be needed.

30: In the first situation, the observations form a **time series**.

We might also be interested in using historical data to **forecast** the future behaviour of the variable. These are the familiar analysis goals:

- **finding patterns** in the data, and
- **creating a (mathematical) model** that captures the essence of these patterns.

Time series patterns can be quite complex and must be **broken down** into multiple component models (trend, seasonal, irregular, etc.).³¹

31: Typically, this can be achieved with fancy analysis methods, but it is not a simple topic, in general (some details can be found in Chapter 9, *Time Series and Forecasting*) – software libraries can help.

Anomaly Detection

The special points from process monitoring are anomalous in the sense that something unexpected happens there, something that changes the nature of the data pre- and post-break.

In a more general context, **anomalous observations** are those that are **atypical** or **unlikely**. From an analytical perspective, anomaly detection can be approached using supervised, unsupervised, or conventional statistical methods.

The discipline is rich and vibrant (and the search for anomalies can end up being an arms race against the “bad guys”), but it is definitely one for which analysts should heed contextual understanding – blind analysis leads to blind alleys!³²

32: A more thorough treatment is provided in Chapter 26 (*Anomaly Detection and Outlier Analysis*).

14.5.6 Quantitative Fallacies

Quantitative fallacies and **misinterpretations** are a consequence of our notoriously poor skills at quantitative interpretation, which manifest themselves through incorrect reasoning or the misuse of statistics (by design or by accident).

- **Correlation is not causation:** causality is one kind of correlation but correlation is not necessarily causal – it’s conceivable that a man who purchases diapers also decides to buy beer, but the purchase of diapers does not cause the purchase of beer. The statement is sometimes extended to imply that while correlation is not causation, it can be highly suggestive.
- **Extreme patterns can mislead:** rare patterns need to be considered separately from the rest of the data. They are either invalid patterns and need to be removed altogether, or interesting and they could reveal that the story has more depth. The presence of extreme patterns or cases in the modeling process could introduce biases to the model and the final model is less likely to be a good fit for the data. For instance, a severe snowstorm hit Ottawa on Feb 16, 2016, causing a large number of road collisions – if we want to predict the number of road collisions on an average day in Ottawa, keeping this day in the model may skew the results.
- **Small effects can be significant:** a statistically significant result does not need to be large, it just needs to be unlikely to be due to chance alone. The terminology is partly to blame for the confusion: in the statistical context, **significance** is not the same as **importance**.

- **Leaving a study's range:** a fallacy can occur when an assumption is replaced by a seemingly similar one which turns out not to be interchangeable. For instance, when a snow storm drops 20cm of snow in Ottawa, traffic may be delayed slightly, but it's business as usual for most citizens; we might expect a similar reaction in Winnipeg, but the same 20cm would paralyze Beijing and block sewers. The effects of a snow storm may not be transferable.
- **There is a human component to any analytical activity:** it is impossible to avoid human bias altogether when analyzing data. The ultimate choice of explanatory parameters or of the final model (to name but these two) can never be done with complete and total detachment.
- **Odd results sometimes happen:** the patterns in subgroups of the data may not align with patterns in the entire dataset, thanks to Simpson's paradox (see Figure 14.15).

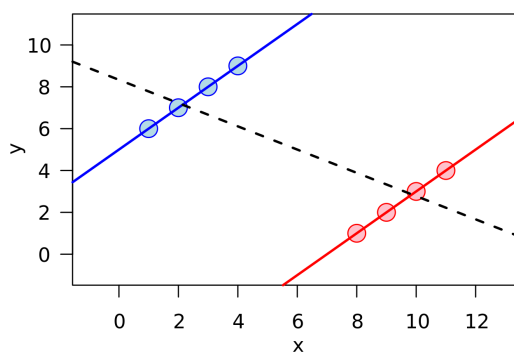




Figure 14.15: Simpson's paradox: the slope of the line of best fit in each of the two subgroups (blue and red) is positive, while the slope of the line of best fit for the entire dataset (black) is negative [author unknown].

- **Keeping the base rate in mind:** the base rate fallacy occurs when the underlying characteristics of a subgroup are ignored. As an example, the likelihood that an individual will die of lung cancer depends on whether he or she is a smoker; if this information is not known, the prediction will also depend on the likelihood of the individual being a smoker. This fallacy is best avoided through the application of Bayesian analysis.
- **Randomness plays a role:** if a situation has occurred more frequently in the past, it is possible that it will be more likely to happen again in the future, but it is also possible that it happened more frequently in the past by chance alone. Statistical analysis will help to separate the Gambler's fallacy from the presence of a signal in the data.
- **Misinterpretation of p -values:** the p -value reveals the probability of observing a result given that the null hypothesis is true, rather than the probability that the null hypothesis is true. As an example, suppose a Department wants to find out whether a reported increase in efficiency is due to the implementation of a new policy; the null hypothesis would be that the new policy has no effect on efficiency. Using available data, the model produces a p -value of 0.05; we cannot conclude that there is a 95% probability that the null hypothesis is false and that the policy change had an effect on efficiency. We can only conclude that there is a 5% probability that our model would show an effect even if none was present.

There is a lot more to say on the topic of data analysis – we will delve into various topics in detail in subsequent chapters.

14.6 Exercises

1. Are the following examples of good questions? Are they vague or specific? What are the ranges of answers we could expect? How would you improve them?
 - a) How does rain affect goal percentage at a soccer match?
 - b) Did the Toronto Maple Leafs beat the Edmonton Oilers?
 - c) Did you like watching the Tokyo Olympics?
 - d) What types of recovery drinks do hockey players drink?
 - e) How many medals will Canada win at the Paris 2024 Olympics?
 - f) Should we fund the Canadian Basketball team more than the Canadian Hockey team?
2. Write a paper discussing some of the ethical issues surrounding the use of artificial intelligence (A.I.), data science (D.S.), and/or machine learning (M.L.) algorithms in the public sector, the private sector, or in academia.
 - a) Establish a list of the 3 most important ethical principles that the use of such algorithms should abide by. Explain why you have selected each of these principles.
 - b) Describe (at least) 2 instances of the use of A.I./D.S./M.L. in the public sector, the private sector, or in academia, when the ethical principles you have chosen were violated. Discuss how the failure to abide by your selected ethical principles have caused (or could cause) harm to individuals, organizations, countries, etc.
 - c) Suggest how the projects discussed above could have been modified so that their use of A.I./D.S./M.L. algorithms would abide by your selected ethical principles.
3. Provide additional data summaries and some simple visual summaries of the artificial dataset of pages 920-921.
4. Select a data project of interest to you (either personally or professionally) and provide a first planning draft for it, touching on the topics discussed in this module and in Chapter 13 (*Non-Technical Aspects of Quantitative and Data Work*). The following questions can help guide your proposal:
 - a) What are some questions associated with the project?
 - b) What is the conceptual model of the underlying situation?
 - c) What kind of dataset(s) exist that could help you answer these questions?
 - d) Are there data or analytical limitations?
 - e) Do you need to collect new data to handle such questions?
 - f) How is the data stored/accessed? What are the infrastructure requirements?
 - g) What do deliverables look like?
 - h) How would successes be quantified/qualified?
 - i) What are your timelines and availability?
 - j) What skillsets are required to work on this project?
 - k) Would you work on this alone or as part of a team?
 - l) How costly would it be to initiate and complete this project?
 - m) What does the data analysis pipeline look like?
 - n) What software and analytical methods will be used?
5. The file `cities.txt` [contains](#) population information about a country's cities. A city is classified as "small" if its population is below 75K, as "medium" if it falls between 75K and 1M, and as "large" otherwise.
 - a) Locate and load the file into the workspace of your choice. How many cities are there? How many in each group?
 - b) Display summary population statistics for the cities, both overall and by group.
6. Find examples of recent "Data in the News" stories. Were they successes or failures? What social consequences could emerge from the technologies described in the stories?
7. In what format is your organization's data available? Are you able to access it easily? Is it updated regularly? Are there data dictionaries? Have you read them?

8. Consider the following situation: you are away on business and you forgot to hand in a very important (and urgently required) architectural drawing to your supervisor before leaving. Your office will send an intern to pick it up in your living space. How would you explain to them, by phone, how to find the document? If the intern has previously been in your living space, if their living space is comparable to yours, or if your spouse is at home, the process may be sped up considerably, but with somebody for whom the space is new (or someone with a visual impairment, say), it is easy to see how things could get complicated. Time is of the essence – you and the intern need to get the job done correctly as quickly as possible. What is your strategy?
9. Translate the cognitive biases to analytical contexts. What cognitive biases are you, your team, and your organization most susceptible to? Least?
10. Research the recent data ethics scandals involving Volkswagen, Amazon, Whole Foods Markets, Cambridge Analytica, Ashley Madison, General Motors, or any other organization. What transpired? Who was affected? What were the consequences to the general public, the organization, the data community? How could it have been avoided?
11. Establish a statement of ethics for your data work. Are there areas that you are unwilling to work on?
12. The remaining exercises use the [Gapminder Tools](#)  (there is also an [offline version](#) ).
 - a) Take some time to explore the tool. In the online version, the default starting point is a bubble chart of 2020 life expectancy vs. income, per country (with bubble size associated with total population). In the offline version, select the “Bubbles” option.
 - b) Can you identify the available variable categories and some of the variables? [You may need to dig around a bit.]
 - c) Why do you think that Gapminder has selected Life Expectancy and Income as the default plotting variables?
 - d) Replace Life Expectancy by Babies per woman. Observe and discuss the changes from the default plot.
 - e) Formulate a few questions that could be answered with the default data.
 - f) Formulate a few questions that could be answered using some of the other variables.
 - g) At what point in the data science workflow do you think that visualizations of this nature could be useful?
 - h) Do these visualizations provide a sound understanding of the system under investigation (the geopolitical Earth)?
 - i) What do you think the data sources are for the underlying dataset? [You may need to dig around the internet to answer this question].
 - j) Are all variables and measurements equally trustworthy? How could you figure this out?
 - k) Is the underlying dataset structured or unstructured?
 - l) Provide a potential data model for the dataset.
 - m) What are the types of the 4 default variables (Life Expectancy, Income, Population, World Regions)?
 - n) Play around with the charts for a bit. Can you find pairs of variables that are positively correlated? Negatively correlated? Uncorrelated?
 - o) Among those variables that are correlated, do any seem to you to exhibit a dependent-independent relationship? How could you identify such pairs?
 - p) Can you provide an eyeball estimate of the mean, the median, and the range of various numerical variables?
 - q) Can you provide an eyeball estimate of the mode of the categorical variables?
 - r) Can you identify epochal moments (special temporal points) in the data where a shift occurs, say?
 - s) Is the tool and its underlying dataset useable? What factors does your answer depend on?

Chapter References

- [1] [ACM Code of Ethics and Professional Conduct](#) . Association for Computing Machinery. Accessed: June 18, 2017.
- [2] C.C. Aggarwal. [Data Mining: the Textbook](#) . Cham: Springer, 2015.
- [3] C.C. Aggarwal, ed. [Data Classification: Algorithms and Applications](#) . CRC Press, 2015.
- [4] C.C. Aggarwal and C.K. Reddy, eds. [Data Clustering: Algorithms and Applications](#) . CRC Press, 2014.
- [5] I. Asimov. *Foundation Series*. Gnome Press, Spectra, Doubleday.
- [6] F.R. Bach and M.I. Jordan. ‘Learning Spectral Clustering, With Application To Speech Separation’. In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 1963–2001.
- [7] [Facebook documents seized by MPs investigating privacy breach](#) . BBC News. Nov. 2018.
- [8] BeauHD. ‘[Google AI Claims 99 Percent Accuracy In Metastatic Breast Cancer Detection](#)’ . In: *Slashdot.com* (Oct. 2018).
- [9] E. Betuel. ‘[Math Model Determines Who Wrote Beatles’ In My Life: Lennon or McCartney?](#)’ . In: *Inverse* (July 2018).
- [10] N. Bien et al. ‘Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet’. In: *PLOS Medicine* 15.11 (2018), pp. 1–19. doi: [10.1371/journal.pmed.1002699](#).
- [11] P. Boily. [MAT2377 - Probability and Statistics for Engineers](#) . Course website.
- [12] P. Boily, S. Davies, and J. Schellinck. [The Practice of Data Visualization](#) . Data Action Lab, 2023.
- [13] boot4life. [What JSON structure to use for key-value pairs](#) . StackOverflow. June 2016.
- [14] D. Brin. [The Transparent Society: Will Technology Force Us to Choose Between Privacy and Freedom?](#) . Perseus, 1998.
- [15] S.E. Brossette et al. ‘Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance’. In: *Journal of the American Medical Informatics Association* 5.4 (July 1998), pp. 373–381. doi: [10.1136/jamia.1998.0050373](#).
- [16] [Centre for Big Data Ethics, Law, and Policy](#) . Data Science Institute, University of Virginia. Accessed: June 18, 2017.
- [17] [Code of Ethics/Conducts](#) . Certified Analytics Professional. Accessed: June 17, 2017.
- [18] V.M Chawla. [ERD "Crow's Foot" Relationship Symbols Cheat Sheet](#) . 2013.
- [19] M. Chen. ‘[Is ‘Big Data’ Actually Reinforcing Social Inequalities?](#)’ . In: *The Nation* (Sept. 2013).
- [20] N. Cohn. ‘[How One 19-Year-Old Illinois Man is Distorting National Polling Averages](#)’ . In: *The Upshot* (June 2016).
- [21] Columbia University Irving Medical Center. ‘[Data Scientists Find Connections Between Birth Month and Health](#)’ . In: *NewsWire.com* (June 2015).
- [22] J.S.A. Corey. *The Expanse*. Orbit Books.
- [23] J. Cranshaw et al. ‘[The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City](#)’ . In: *ICWSM*. Ed. by John G. Breslin et al. The AAAI Press, 2012.
- [24] J. Dastin. ‘[Amazon scraps secret AI recruiting tool that showed bias against women](#)’ . In: *Reuters* (Oct. 2018).
- [25] T.H. Davenport and D.J. Patil. ‘[Data Scientist: the Sexiest Job of the 21st Century](#)’ . In: *Harvard Business Review* (Oct. 2012).
- [26] [Cognitive Biases](#) . The Decision Lab. Accessed: Sep 3, 2021.
- [27] L. Donnelly. ‘[Robots are better than doctors at diagnosing some cancers, major study finds](#)’ . In: *The Telegraph* (May 2018).
- [28] N. Feldman. [Data Lake or Data Swamp?](#) . July 2015.

- [29] K. Fung. 'The Ethics Conversation We're Not Having About Data [↗](#)'. In: *Harvard Business Review* (Nov. 2015).
- [30] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press Cambridge, 2016.
- [31] A. Gumbus and F. Grodzinsky. 'Era of Big Data: danger of discrimination [↗](#)'. In: *ACM SIGCAS Computers and Society* 45.3 (2015), pp. 118–125.
- [32] K. Hao. 'We tried teaching an AI to write Christmas movie plots. Hilarity ensued. Eventually [↗](#)'. In: *MIT Technology Review* (Dec. 2018).
- [33] P. Hapala et al. 'Mapping the electrostatic force field of single molecules from high-resolution scanning probe images'. In: *Nature Communications* 7.11560 (2016).
- [34] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction [↗](#)*, 2nd ed. Springer, 2008.
- [35] Henning (WMDE). *UML diagram of the Wikibase Data Model [↗](#)*. Wikimedia.
- [36] J. Hiner. 'How big data will solve your email problem [↗](#)'. In: *ZDNet* (Oct. 2013).
- [37] R.V. Hogg and E.A. Tanis. *Probability and Statistical Inference*. 7th. Pearson/Prentice Hall, 2006.
- [38] K.-W. Hsu et al. 'Data Mining Based Tax Audit Selection: A Case Study of a Pilot Project at the Minnesota Department of Revenue'. In: *Real World Data Mining Applications*. Cham: Springer International Publishing, 2015, pp. 221–245. doi: [10.1007/978-3-319-07812-0_12](#).
- [39] Indiana University. 'Scientists use Instagram data to forecast top models at New York Fashion Week [↗](#)'. In: *Science Daily* (Sept. 2015).
- [40] A.B. Jensen et al. 'Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients'. English. In: *Nature Communications* 5 (2014). doi: [10.1038/ncomms5022](#).
- [41] M. Jing. 'AlphaGo vanquishes world's top Go player, marking A.I.'s superiority over human mind [↗](#)'. In: *South China Morning Post* (May 2017).
- [42] I. Johnston. 'AI robots learning racism, sexism and other prejudices from humans, study finds [↗](#)'. In: *The Independent* (Apr. 2017).
- [43] M. Judge. 'Facial-Recognition Technology Affects African-Americans More Often [↗](#)'. In: *The Root* (May 2016).
- [44] M. Kosinski and Y. Wang. 'Deep Neural Networks Are More Accurate Than Humans at Detecting Sexual Orientation From Facial Images'. In: *Journal of Personality and Social Psychology* 114.2 (Feb. 2018), pp. 246–257.
- [45] H. T. Kung and D. Vlah. 'A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks'. In: *Int. J. Sen. Netw.* 8.3/4 (Oct. 2010), pp. 202–208. doi: [10.1504/IJSNET.2010.036195](#).
- [46] S. Lee and D. Baer. '20 Cognitive Biases That Screw Up Your Decisions [↗](#)'. In: *Business Insider* (Dec. 2015).
- [47] D. Lewis. 'An AI-Written Novella Almost Won a Literary Prize [↗](#)'. In: *Smithsonian Magazine* (Mar. 2016).
- [48] *Scientists Using GPS Tracking on Endangered Dhole Wild Dogs [↗](#)*. Live View GPS. Oct. 2018.
- [49] E. Mack. 'Elon Musk: Artificial intelligence may spark World War III [↗](#)'. In: *CNET* (Sept. 2017).
- [50] A.M. Masci et al. 'An improved ontological representation of dendritic cells as a paradigm for all cell types [↗](#)'. In: *BMC Bioinformatics* (2009).
- [51] R. Mérou. *Conceptual map of Free Software [↗](#)*. Wikimedia. 2010.
- [52] C. O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy [↗](#)*. Crown, 2016.
- [53] *Open Data [↗](#)*. Wikipedia. Accessed: June 19, 2017.
- [54] *Open Up Guide: Using Open Data to Combat Corruption [↗](#)*. Open Data Charter. Accessed: June 20, 2017.

- [55] V. U. Panchami and N. Radhika. 'A novel approach for predicting the length of hospital stay with DBSCAN and supervised classification algorithms'. In: *ICADIWT. IEEE*, 2014, pp. 207–212.
- [56] *A Conversation with ...: Ethics in Quantitative Contexts*. Paquette, J. and Boily, P.
- [57] R.W. Paul and L. Elder. *Understanding the Foundations of Ethical Reasoning*. 2nd. Foundation for Critical Thinking, 2006.
- [58] C. Plant et al. 'Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease'. In: *NeuroImage* 50.1 (2010), pp. 162–174.
- [59] F. Provost and T. Fawcett. *Data Science for Business*. O'Reilly, 2015.
- [60] S. Ramachandran and J. Flint. 'At Netflix, who wins when it's Hollywood vs. the algorithm?'. In: *Wall Street Journal* (Nov. 2018).
- [61] R. Reich. 'Now AI can write students' essays for them, will everyone become a cheat?'. In: *The Guardian* (Nov. 2022).
- [62] S. Reichman. 'These AI-invented paint color names are so bad, they're good'. In: *Curbed* (May 2017).
- [63] *Research Integrity & Ethics*. Memorial University of Newfoundland.
- [64] T. Rikert. 'A.I. hype has peaked so what's next?'. In: *TechCrunch* (Sept. 2017).
- [65] D. Robinson. 'What's the difference between data science, machine learning, and artificial intelligence?'. In: *Variance Explained* (Jan. 2018).
- [66] S. Rogers. 'That Popular AI Photo App is Stealing from Human Artists – and Worse'. In: *Yahoo! News* (Dec. 2022).
- [67] J. Schellinck and P. Boily. 'Data, Automation, and Ethics'. In: *Data Science Report Series* (2020).
- [68] R. Schutt and C. O'Neill. *Doing Data Science: Straight Talk From the Front Line*. O'Reilly, 2013.
- [69] J.C. Scott. *Against the Grain: A Deep History of the Earliest States*. eng. New Haven: Yale University Press, 2017.
- [70] B. Smith. 'Artificial intelligence better than physicists at designing quantum science experiments'. In: *ABC Science* (Oct. 2018).
- [71] I. Stewart. 'The Fourth Law of Humanics'. In: *Nature* 535 (2016).
- [72] R. Sutton and G. Barto. *Reinforcement Learning: an Introduction*. MIT Press, 2018.
- [73] J. Taylor. 'Four Problems in Using CRISP-DM and How To Fix Them'. In: *KDnuggets.com* (2017).
- [74] *Development of National Statistical Systems*. United Nations, Statistics Division. Accessed: June 17, 2017.
- [75] A. Van Dam. 'This researcher studied 400,000 knitters and discovered what turns a hobby into a business'. In: *Washington Post* (Nov. 2018).
- [76] D. Wakabayashi. 'Firm led by Google veterans uses A.I. to 'nudge' workers toward happiness'. In: *New York Times* (Dec. 2018).
- [77] Wikipedia. *Cluster Analysis Algorithms*.
- [78] D. Woods. 'bitly's Hilary Mason on "What is A Data Scientist?"'. In: *Forbes* (Mar. 2012).
- [79] Woottoo. *Entity - Relationship Model*. Wikimedia.
- [80] E. Yong. 'Wait, have we really wiped out 60% of animals?'. In: *The Atlantic* (Oct. 2018).

by **Patrick Boily**, with contributions from **Jen Schellinck**

Once raw data has been collected and stored in a database or a dataset, the focus should shift to data cleaning and processing.

This requires testing for soundness and fixing errors, designing and implementing strategies to deal with missing values and outlying/influential observations, as well as low-level exploratory data analysis and visualization to determine what data transformations and dimension reduction approaches will be needed before embarking on a more sophisticated path.

In this chapter, we establish the essential elements of data cleaning and data processing.

15.1 Introduction

Martin K: Data is messy, Alison.

Alison M: Even after it's been cleaned?

Martin K: Especially after it's been cleaned.

(P. Boily, J. Schellinck, *The Great Balancing Act* [unpublished]).

Data cleaning and data processing are essential aspects of quantitative analysis projects; analysts and consultants should be prepared to spend up to 80% of their time on data preparation, keeping in mind that:

- processing should **NEVER** be done on the original dataset – make copies along the way;
- **ALL** cleaning steps need to be documented;
- if **too much** of the data requires cleaning up, the data collection procedure might need to be **revisited**, and
- records should only be discarded as a **last resort**.

Another thing to keep in mind is that cleaning and processing may need to take place more than once depending on the type of data collection (one pass, batch, continuously), and that it is essentially impossible to determine if all data issues have been found and fixed.

Note: in this chapter, we are assuming that the datasets of interest contain only numerical and/or categorical observations. Additional steps must be taken when dealing with unstructured data, such as text or images (we'll have more to say on this topic later).

15.1 Introduction	951
15.2 General Principles	952
Data Cleaning Approaches	952
Pros and Cons	952
Tools and Methods	953
15.3 Data Quality	954
Common Error Sources	955
Detecting Invalid Entries	955
15.4 Missing Values	957
Missing Value Mechanisms	957
Imputation Methods	958
Multiple Imputation	965
15.5 Anomalous Observations	966
Anomaly Detection	967
Outlier Tests	967
Visual Outlier Detection	970
15.6 Data Transformations	972
Common Transformations	973
Box-Cox Transformations	975
Scaling	979
Discretizing	979
Creating Variables	980
15.7 Example: Algae Blooms	980
Problem Description	980
Loading the Data	981
Summary & Visualization	982
Data Cleaning	993
Principal Components	997
15.8 Exercises	999
Chapter References	1000

15.2 General Principles

Dilbert: I didn't have any accurate numbers, so I just made up this one. Studies have shown that accurate numbers aren't any more useful than the ones you make up.

Pointy-Haired Boss: How many studies showed that?

Dilbert: [*beat*] Eighty-seven.

(S. Adams, *Dilbert* [comic](#), 8 May 2008)

15.2.1 Data Cleaning Approaches

We recognize two main **philosophical** approaches to data cleaning and validation:

- **methodical**, and
- **narrative**.

The **methodical** approach consists in running through a **checklist** of potential issues and flagging those that apply to the data.

The **narrative** approach, on the other hand, consists in **exploring** the dataset while searching for unlikely or irregular patterns.

Which approach the consultant/analyst opts to follow depends on a number of factors, not the least of which is the client's needs and views on the matter – it is important to discuss this point with the clients.

15.2.2 Pros and Cons

The methodical approach focuses on **syntax**; the checklist is typically **context-independent**, which means that it (or one of its subsets) can be reused from one project to another – this makes data analysis pipelines **easy to implement** and **automate**. In the same vein, this approach allows for common errors to be **easily identified**.

On the flip side, the checklist may be quite extensive and the entire process may prove **time-consuming**, but the biggest disadvantage of the methodical approach is that it makes it difficult to identify **new types of errors**.

In contrast, the narrative approach focuses on **semantics**; even false starts may simultaneously produce **data understanding** prior to an eventual switch to a more mechanical approach.

It is easy, however, to miss important (and perhaps obvious) sources of errors as well as invalid observations when the datasets have a **large number of features**.

There is an additional downside: **domain expertise**, coupled with the narrative approach, may bias the process by neglecting “uninteresting” areas of the dataset – it takes a special person to spend time on potentially barren lands when they know that greener pastures are available just over yonder.

random missing values	outliers	values outside of expected range - numeric	factors incorrectly/consistently coded	date/time values in multiple formats
impossible numeric values	leading or trailing white space	badly formatted date/time values	non-random missing values	logical inconsistencies across fields
characters in numeric field	values outside of expected range - date/time	DCB!	inconsistent or no distinction between null, 0, not available, not applicable, missing	possible factors missing
multiple symbols used for missing values	???	fields incorrectly separated in row	blank fields	logical inconsistencies within field
entire blank rows	character encoding issues	duplicate value in unique field	non-factor values in factor	numeric values in character field

Figure 15.1: Data cleaning bingo card [J. Schellinck].

15.2.3 Tools and Methods

A non-exhaustive list of common data issues can be found in the *Data Cleaning Bingo Card* (see Figure 15.1). Other methods include:

- **visualizations** – which may help easily identify observations that need to be further examined;
- **data summaries** – # of missing observations; 5-pt summary, mean, standard deviation, skew, kurtosis, for numerical variables; distribution tables for categorical variables;
- ***n*-way tables** – counts for joint distributions of categorical variables;
- **small multiples** – tables/visualizations indexed along categorical variables, and
- **preliminary data analyses** – which may provide “huh, that’s odd...” realizations.

It is important to note that there is nothing wrong with running a number of analyses to flush out data issues, but remember to label your initial forays as **preliminary** analyses.¹

Data scientists, dataanalysts, and quantitative consultants alike need to be comfortable with **both** approaches.

As an analogy, the narrative approach is akin to working out a crossword puzzle with a pen and accepting to put down potentially erroneous answers once in a while to try to open up the grid.²

The methodical approach, on the other hand, is similar to working out the puzzle with a pencil and a dictionary, only putting down answers when their correctness is guaranteed.³

1: From the client or stakeholder’s perspective, repeated analyses may create a sense of unease and distrust, even if they form a crucial part of the analytical process.

2: What artificial intelligence researchers call the “exploration” approach.

3: The “exploitation” approach of artificial intelligence.

More puzzles get solved when using the first approach, but missteps tend to be spectacular. Not as many puzzles get solved the second way, but the trade-off is that it leads to fewer mistakes.

15.3 Data Quality

Calvin's Dad: OK Calvin. Let's check over your math homework.

Calvin: Let's not and say we did.

Calvin's Dad: Your teacher says you need to spend more time on it. Have a seat.

Calvin: More time?! I already spent 10 whole minutes on it! 10 minutes shot! Wasted! Down the drain!

Calvin's Dad: You've written here $8 + 4 = 7$. Now you know that's not right.

Calvin: So I was off a little bit. Sue me.

Calvin's Dad: You can't **add** things and come with **less** than you started with!

Calvin: I can do that! It's a free country! I've got my rights!

(B. Watterson, *Calvin and Hobbes*, 15-09-1990.)

The quality of the data has an important effect on the quality of the results: as the saying goes: "garbage in, garbage out."

Data is said to be **sound** when it has as few issues as possible with:

- **validity** – are observations sensible, given data type, range, mandatory response, uniqueness, value, regular expressions, etc. (e.g. a value that is expected to be text value is a number, a value that is expected to be positive is negative, etc.);
- **completeness** – are there missing observations (more on this in a subsequent section)?;
- **accuracy and precision** – are there measurement and/or data entry errors (e.g., an individual has 3 children but only 2 are recorded, etc., see Figure 15.2, linking accuracy to bias and precision to the standard error)?;
- **consistency** – are there conflicting observations (e.g., an individual has no children, but the age of one kid is recorded, etc.)?; and
- **uniformity** – are units used uniformly throughout (e.g., an individual is 6ft tall, whereas another one is 145cm tall)?

Finding an issue with data quality after the analyses are completed is a sure-fire way of losing the stakeholder's or client's trust – check early and often!

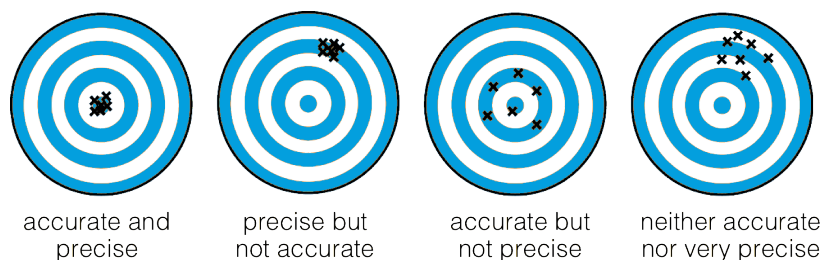


Figure 15.2: Accuracy as bias, precision as standard error [author unknown].

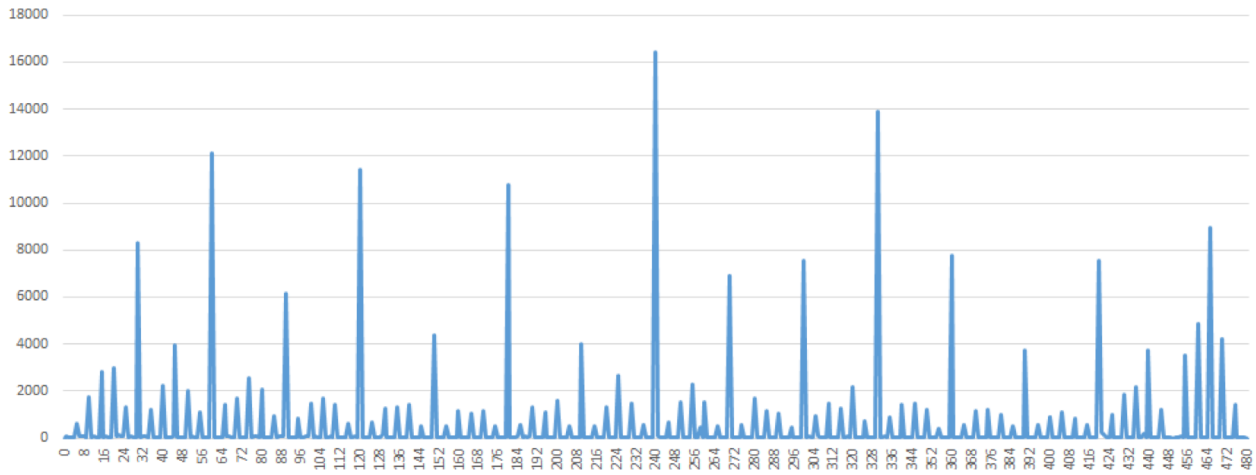


Figure 15.3: An illustration of heaping behaviour: self-reported time spent working in a day [personal file]. The entries for 7, 7.5, and 8 hours are omitted. Note the rounding off at various multiples of 5 minutes.

15.3.1 Common Error Sources

If the analysts have some control over the data collection and initial processing, regular data validation tests are easier to set-up.

When analysts are dealing with **legacy**, **inherited**, or **combined** datasets, however, it can be difficult to recognize errors that arise from:

- missing data being given a code;
- NA/blank entries being given a code;
- data entry errors;
- coding errors;
- measurement errors;
- duplicate entries;
- heaping (see Figure 15.3 for an example),
- etc.

15.3.2 Detecting Invalid Entries

Potentially invalid entries can be detected with the help of a number of methods:

- **univariate descriptive statistics** – *z*-score, count, range, mean, median, standard deviation, etc.;
- **multivariate descriptive statistics** – *n*-way tables and logic checks;
- **data visualization** – scatterplot, histogram, joint histogram, etc. (see Chapter 18, *Data Visualization and Data Exploration*, and [2] for more information on the topic),
- and so on.

It is important to point out that univariate tests do not always tell the **whole** story (and may in fact obscure important details).

Example: consider, for instance, an artificial medical dataset consisting of 38 patients' records, containing, among others, fields for the **sex** and the **pregnancy status** of the patients.

A summary of the data of interest is provided in the **frequency counts** (1-way tables) of the table below:

Sex	Male	19
	Female	17
	(blank)	2
	Total	38

Pregnant	Yes	7
	No	27
	99	1
	(blank)	3
	Total	38

Analysts can quickly notice that some values are missing (in green) and that an entry has been miscoded as 99 (in yellow). Using only these univariate summaries, however, it is impossible to decide what to do with these invalid entries.

The 2-way frequency counts shed some light on the situation, and uncover other potential issues with the data and/or the data collection protocol.

		Pregnant				Total
		Yes	No	99	(blank)	
Sex	Male	1	17	1	0	19
	Female	6	9	0	2	17
	(blank)	0	1	0	1	2
Total		7	27	1	3	38

One of the green entries is actually blank along the two variables; depending on the other information, this entry could be a candidate for **imputation** or outright **deletion** (more on these concepts in the next section).

Three other observations are missing a value along exactly one variable, but the information provided by the other variables may be complete enough to warrant imputation. Of course, if more information is available about the patients, the analyst may be able to determine why the values were missing in the first place (however privacy concerns at the collection stage might muddy the waters).

The mis-coded information on the pregnancy status (99, in yellow) is linked to a male client, and as such re-coding it as 'No' is likely to be a reasonable decision.⁴

A similar reasoning process should make the analyst question the validity of the entry shaded in red – it might very well be correct, but it is important to at least inquire about this data point, as the answer could lead to an eventual re-framing of the definitions and questions used at the collection stage.

In general, there is no universal or one-size-fits-all approach – a lot depends on the **nature of the data**. As always, domain expertise can provide valuable help and suggest fruitful exploration avenues.

4: Although this may **not necessarily be the correct decision**... data measurements are rarely as clear cut as we may think upon only a first reflection.

15.4 Missing Values

Obviously, the best way to treat missing data is not to have any (T. Orchard, M. Woodbury, [8]).

Why does it matter that some values may be **missing**?

As a start, missing values can potentially introduce **bias** into the analysis, which is rarely (if at all) a good thing, but, more pragmatically, they may interfere with the functioning of most analytical methods, which cannot easily accommodate missing observations without breaking down.⁵

Consequently, when faced with missing observations, analysts have two options: they can either **discard** the missing observation (which is not typically recommended, unless the data is missing completely randomly), or they can **create a replacement value** for the missing observation (the **imputation** strategy has drawbacks since we can never be certain that the replacement value is the true value, but is often the best available option; information in this section is taken partly from [5, 9, 12, 10]).

Blank fields come in 4 flavours:

- **nonresponse** – an observation was expected but none was entered;
- **data entry issues** – an observation was recorded but was not entered in the dataset;
- **invalid entries** – an observation was recorded but was considered invalid and has been removed, and
- **expected blanks** – a field has been left blank, but expectedly so.

Too many missing values of the first three types can be indicative of **issues with the data collection process**, while too many missing values of the fourth type can be indicative of **poor questionnaire design** (see Section 10.2 for a brief discussion on these topics).

Either way, missing values cannot simply be **ignored**: either the

- corresponding record is removed from the dataset (not recommended without justification, as doing so may cause a loss of auxiliary information and may bias the analysis results), or
- missing values must be **imputed** (that is to say, a reasonable replacement value must be found).⁶

15.4.1 Missing Value Mechanisms

The relevance of an imputation method is dependent on the underlying **missing value mechanism**. Indeed, values may be:

- **missing completely at random** (MCAR) – the item absence is independent of its value or of the unit's auxiliary variables (e.g., an electrical surge randomly deletes an observation in the dataset);
- **missing at random** (MAR) – the item absence is not completely random, and could, in theory, be accounted by the unit's complete auxiliary information, if available (e.g., if women are less likely to tell you their age than men for societal reasons, but not because of the age values themselves), and

5: As an example, the normal equations $\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^T \mathbf{Y}$ of linear regression could not be solved if some of the data is missing as $\mathbf{X}^T \mathbf{X}$ is not defined in that case.

6: A non-negligible proportion of stakeholders, who would be the first to tell you that they understand nothing about data analysis in the first place, balk at this notion. It helps to remember that we do not generally conduct data analysis to fully understand any particular unit/observation, but rather to get a sense for overall **patterns** in the data (and how we could use these patterns to make predictions for the future, say). We suspect that this is directly linked to the fear that our “personhood” will be erased in favour of numerical summaries, that we are who the data says we are instead of ... well, who we **really** are. Measurement errors happen.

- **not missing at random** (NMAR) – the reason for nonresponse is related to the item value itself (e.g., if illicit drug users are less likely to admit to drug use than teetotallers).

The analyst's main challenge in that regard is that the missing mechanism cannot typically be determined with **any degree of certainty**.

15.4.2 Imputation Methods

There are numerous statistical **imputation** methods. They each have their strengths and weaknesses; analysts should take care to select a method which is appropriate for the situation at hand.⁷

7: Imputation methods work best under MCAR or MAR, but keep in mind that they all tend to produce **biased estimates** nonetheless... the hope is the bias is small, and that the benefits of obtaining an estimate in the first place overcomes the presence of bias.

- In **list-wise deletion**, all units with at least one missing value are removed from the dataset. This straightforward imputation strategy assumes MCAR, but it can introduce bias if MCAR does not hold, and it leads to a reduction in the sample size and an increase in standard errors.
- In **mean** or **most frequent imputation**, the missing values are substituted by the average or most frequent value in the unit's subpopulation group (stratum). This commonly-used approach also assumes MCAR, but it can create distortions in the underlying distributions (such as a spike at the mean) and create spurious relationships among variables.
- In **regression** or **correlation imputation**, the missing values are substituted using a regression on the other variables. This model assumes MAR and trains the regression on units with complete information, in order to take full advantage of the auxiliary information when it is available. However, it artificially reduces data variability and produces over-estimates of correlations.
- In **stochastic regression imputation**, the regression estimates are augmented with random error terms added. Just as in regression estimation, the model assumes MAR; an added benefit is that it tends to produce estimates that "look" more realistic than regression imputation, but it comes with an increased risk of type I error (false positives) due to small standard errors.
- **Last observation carried forward** (LOCF) and its cousin **next observation carried backward** (NOCB) are useful for longitudinal data; a missing value can simply be substituted by the previous or next value. LOCF and NOCB can be used when the values do not vary greatly from one observation to the next, and when values are MCAR. Their main drawback is that they may be too "generous" for studies that are trying to determine the effect of a treatment over time, say.
- Finally, in **k-nearest-neighbour imputation**, a missing entry in a MAR scenario is substituted by the average (or median, or mode) value from the subgroup of the *k* most similar complete respondents. This requires a notion of **similarity** between units (which is not always easy to define reasonably). The choice of *k* is somewhat arbitrary and can affect the imputation, potentially distorting the data structure when it is too large.

What does imputation look like in practice? Consider the following scenario (which is, somewhat embarrassingly, based on a true story).

Example: after marking the final exams of the 211 students who did not drop her course in *Advanced Retroencabulation* at State University, Dr. Helga Vanderwhede creates a data frame grades of final exam grades and mid term-grades.

Setting up the grades data frame

```
MT = c(
80,73,83,60,49,96,87,87,60,53,66,83,32,80,66,90,72,55,76,
46,48,69,45,48,77,52,59,97,76,89,73,73,48,59,55,76,87,55,
80,90,83,66,80,97,80,55,94,73,49,32,76,57,42,94,80,90,90,
62,85,87,97,50,73,77,66,35,66,76,90,73,80,70,73,94,59,52,
81,90,55,73,76,90,46,66,76,69,76,80,42,66,83,80,46,55,80,
76,94,69,57,55,66,46,87,83,49,82,93,47,59,68,65,66,69,76,
38,99,61,46,73,90,66,100,83,48,97,69,62,80,66,55,28,83,59,
48,61,87,72,46,94,48,59,69,97,83,80,66,76,25,55,69,76,38,
21,87,52,90,62,73,73,89,25,94,27,66,66,76,90,83,52,52,83,
66,48,62,80,35,59,72,97,69,62,90,48,83,55,58,66,100,82,78,
62,73,55,84,83,66,49,76,73,54,55,87,50,73,54,52,62,36,87,
80,80
)

FE = c(
41,54,93,49,92,85,37,92,61,42,74,84,61,21,75,49,36,62,92,
85,50,90,52,63,64,85,66,51,41,75,4,46,38,71,42,18,76,42,
94,53,77,65,95,3,74,0,97,62,74,61,80,47,39,92,59,37,59,71,
20,67,69,88,53,52,81,41,81,48,67,65,92,75,68,55,67,51,83,
71,58,37,65,66,51,43,83,34,55,59,20,62,22,70,64,59,73,74,
73,53,44,36,62,45,80,85,41,80,84,44,73,72,60,65,78,60,34,
91,40,41,54,91,49,92,85,37,92,61,42,74,84,61,21,75,49,36,
62,92,85,50,92,52,63,64,85,66,51,41,75,4,46,38,71,42,18,
76,42,92,53,77,65,92,3,74,0,52,62,74,61,80,47,39,92,59,37,
59,71,20,67,69,88,53,52,81,41,81,48,67,65,94,75,68,55,67,
51,83,71,58,37,65,66,51,43,83,34,55,59,20,62,22,70,64,59
)

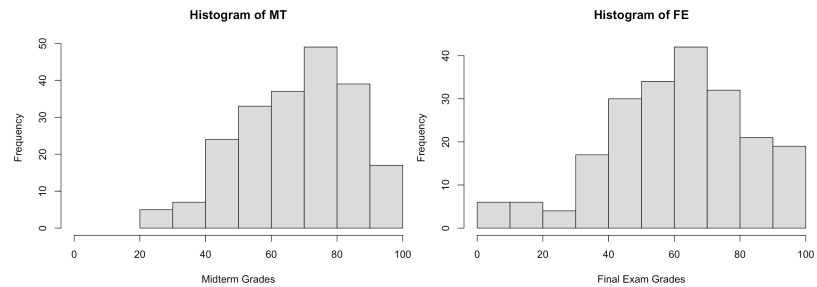
grades=data.frame(MT,FE)
summary(grades)
```

MT	FE
Min.: 21.0	Min.: 0.00
1st Qu.: 55.00	1st Qu.: 56.50
Median: 70.00	Median: 62.00
Mean: 68.74	Mean: 60.09
3rd Qu.: 82.50	3rd Qu.: 75.00
Max.: 100.00	Max.: 97.00

She plots the final exam grades (y) against the mid-term exam grades (x), as seen below.

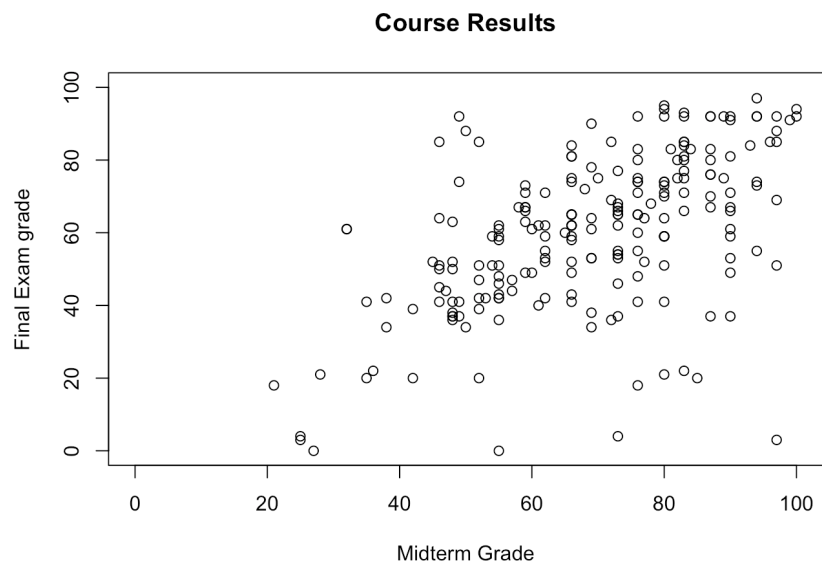
Plotting the grades data frame I

```
hist(MT, xlim=c(0,100), xlab=c("Midterm Grades"))
hist(FE, xlim=c(0,100), xlab=c("Final Exam Grades"))
```



Plotting the grades data frame II

```
plot(grades, xlim=c(0,100), ylim=c(0,100),
     xlab=c("Midterm Grade"), ylab=c("Final Exam grade"),
     main=c("Course Results"))
```



Looking at the data, she sees that final exam grades are **weakly correlated** with mid-term exam grades: students who performed well on the mid-term tended to perform well on the final, and students who performed poorly on the mid-term tended to perform poorly on the final (as is usually the case), but the link is not that strong.

Correlation between mid-term and final grades

```
cor(grades$MT, grades$FE)
```

```
[1] 0.5481776
```

She also sees that there is a **fair amount of variability** in the data: the noise is not very tight around the (eye-balled) line of best fit. The linear regression model is:

Line of best fit

```
model <- lm(FE ~ MT, data=grades)
summary(model)

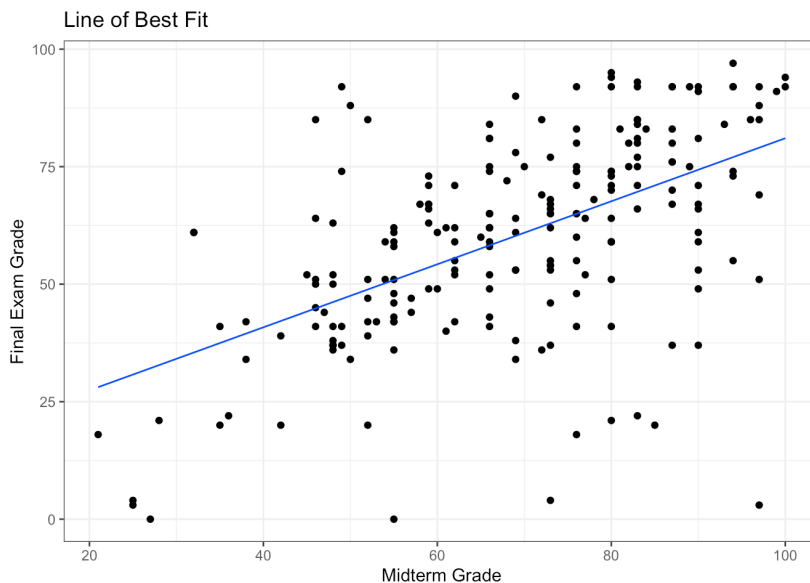
library(ggplot2)
ggplot(model) + geom_point(aes(x=MT, y=FE)) +
  geom_line(aes(x=MT, y=.fitted), color="blue" ) +
  theme_bw() +
  xlab(c("Midterm Grade")) +
  ylab(c("Final Exam Grade")) +
  ggtitle(c("Line of Best Fit"))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	14.00968	5.01523	2.793	0.0057 **
MT	0.67036	0.07075	9.475	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.81 on 209 degrees of freedom
 Multiple R-squared: 0.3005, Adjusted R-squared: 0.2972
 F-statistic: 89.78 on 1 and 209 DF, p-value: < 2.2e-16



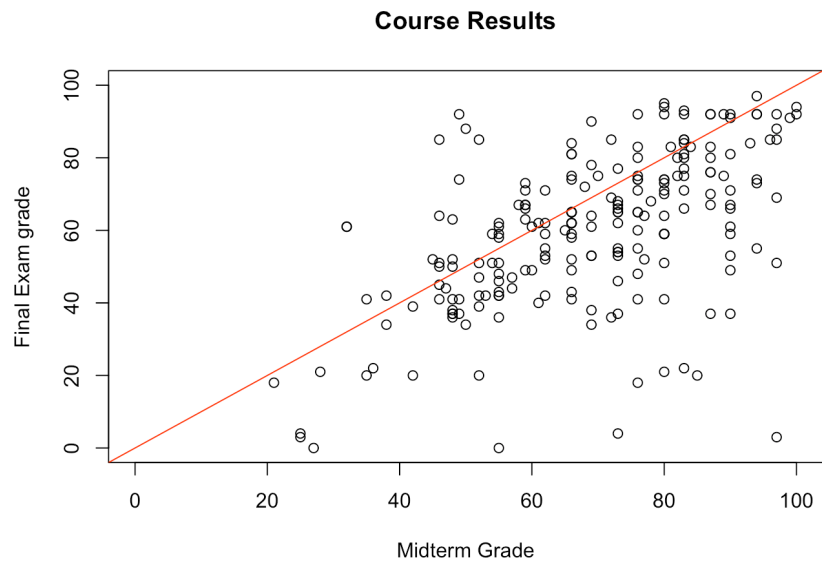
Furthermore, she realizes that the final exam was harder than the students expected (as the slope of the line of best fit is smaller than 1, as only 29% of observations lie above the line $MT=FE$) – she suspects that they simply did not prepare for the exam seriously,⁸ as most of them could not match their mid-term exam performance.

8: And not that she made the exam too difficult, no matter what her ratings on RateMyProfessor.com suggest.

```
sum(grades$MT <= grades$FE)/nrow(grades)
```

[1] 0.2890995

```
plot(grades, xlim=c(0,100), ylim=c(0,100),
     xlab=c("Midterm Grade"), ylab=c("Final Exam grade"),
     main=c("Course Results"))
abline(a=0, b=1, col="red")
```



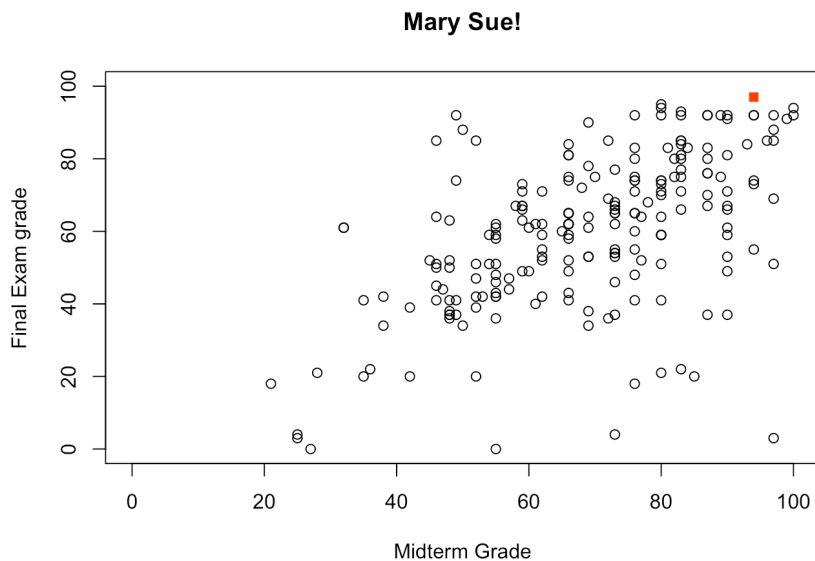
As Dr. Vanderwhede comes to terms with her disappointment, she takes a deeper look at the numbers, at some point sorting the dataset according to the mid-term exam grades.

```
s.grades <- grades[order(-grades$MT),]
head(s.grades,16)
```

student	MT	FE
122	100	92
188	100	94
116	99	91
28	97	51
44	97	3
61	97	69
125	97	92
143	97	85
179	97	88
6	96	85
*47	94	97
54	94	92
74	94	55
97	94	73
139	94	92
162	94	74

It looks like good old Mary Sue (row number 47) performed better on the final than on the mid-term (where her performance was already superlative), scoring the highest grade. What a great student she is!⁹

```
plot(s.grades[,c("MT","FE")], xlim=c(0,100), ylim=c(0,100),
     col=ifelse(row.names(s.grades)=="47",'red','black'),
     pch=ifelse(row.names(s.grades)=="47",22,1), bg='red',
     xlab=c("Midterm Grade"), ylab=c("Final Exam grade"),
     main=c("Mary Sue!"))
```



9: And such a fantastic person – in spite of her superior intellect, she is adored by all of her classmates, thanks to her sunny disposition and willingness to help at all times. If only all students were like Mary Sue...

She continues to toy with the spreadsheet until the phone rings. After a long and exhausting conversation with Dean Bitterman about teaching loads and State University's reputation, Dr. Vanderwhede returns to the spreadsheet and notices in horror that she has accidentally deleted the final exam grades of all students with a mid-term grade greater than 93.

```
s.grades$FE.NA <- ifelse(s.grades$MT>93,NA,s.grades$FE)
```

What is she to do? Anyone with a modicum of technical savvy would advise her to either undo her changes or to close the file without saving the changes,¹⁰ but in full panic mode, the only solution that comes to her mind is to impute the missing values.

10: Or to simply re-enter the final grades by comparing with the physical papers...

She knows that the missing final grades are MAR (and not MCAR since she remembers sorting the data along the MT values); she produces the imputations shown in Figure 15.4.

List-wise deletion

```
plot(s.grades[,c("MT","FE.NA")], xlim=c(0,100),
     ylim=c(0,100), xlab=c("Midterm Grade"),
     ylab=c("Final Exam grade"),
     main=c("List-wise Deletion"))
```

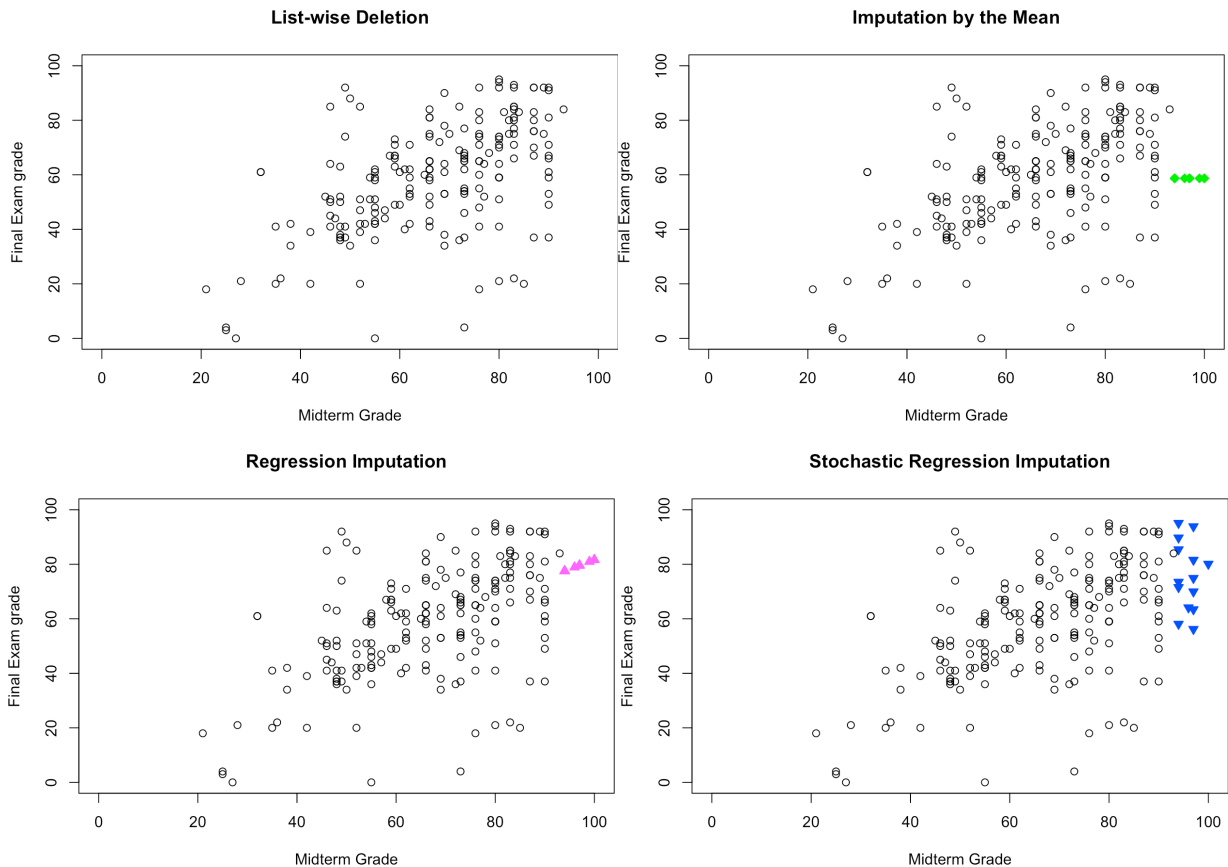


Figure 15.4: Imputation results in the grades data frame: listwise deletion (top left), mean imputation (top right), regression imputation (bottom left), stochastic regression imputation (bottom right).

Mean imputation

```
s.mean <- mean(s.grades$FE.NA, na.rm = TRUE)
s.grades$FE.NA.mean <- ifelse(s.grades$MT>93,s.mean,
                              s.grades$FE)
plot(s.grades[,c("MT","FE.NA.mean")], xlim=c(0,100),
     ylim=c(0,100), pch=ifelse(s.grades$MT>93,23,1),
     col=ifelse(s.grades$MT>93,'green','black'),
     bg='green', xlab=c("Midterm Grade"),
     ylab=c("Final Exam grade"),
     main=c("Imputation by the Mean"))
```

Regression imputation

```
model.2 <- lm(FE.NA ~ MT, data=s.grades)
s.grades$FE.NA.reg <- ifelse(s.grades$MT>93,
                             model.2[[1]][1]+model.2[[1]][2]*s.grades$MT,
                             s.grades$FE)
plot(s.grades[,c("MT","FE.NA.reg")], xlim=c(0,100),
     ylim=c(0,100), pch=ifelse(s.grades$MT>93,24,1),
     col=ifelse(s.grades$MT>93,'magenta','black'),
     bg='magenta', xlab=c("Midterm Grade"),
     ylab=c("Final Exam grade"),
     main=c("Regression Imputation"))
```

Stochastic regression imputation

```

model.3 <- lm(FE.NA ~ MT, data=s.grades)
s.grades$FE.NA.sreg <- ifelse(s.grades$MT>93,
  model.3[[1]][1]+model.3[[1]][2]*s.grades$MT +
  rnorm(nrow(s.grades),0,summary(model.3)$sigma),
  s.grades$FE)
plot(s.grades[,c("MT","FE.NA.sreg")], xlim=c(0,100),
  ylim=c(0,100), pch=ifelse(s.grades$MT>93,25,1),
  col=ifelse(s.grades$MT>93,'blue','black'),
  bg='blue', xlab=c("Midterm Grade"),
  ylab=c("Final Exam grade"),
  main=c("Stochastic Regression Imputation"))

```

She remembers what the data looked like originally, and concludes that the best imputation method is the stochastic regression model.

This conclusion only applies to this specific example, however. In general, that might not be the case due to various *No Free Lunch* results.¹¹

The main take-away from this example is that various imputation strategies lead to different outcomes, and perhaps more importantly, that even though the imputed data might “look” like the true data, we have no way to measure its **departure from reality** – any single imputed value is likely to be completely off.

Mathematically, this might not be problematic, as the average departure is likely to be relatively small, but in a business context or a personal one, this might create gigantic problems – how is Mary Sue likely to feel about Dr.Vanderwhede’s solution to her conundrum?

```

s.grades[row.names(s.grades) == "47",
  c("MT", "FE", "FE.NA.reg")]

```

student	MT	FE	FE.NA.reg
*47	94	97	77.54035

And how would Dean Bitterman react were he to find out about the imputation scenario from irate students? The solution has to be compatible with the ultimate data science objective: from Dr. Vanderwhede’s perspective, perhaps the only thing that matters is capturing the **essence** of the students’ performance, but from the student’s perspective, the objective is emphatically different.¹²

Even though such questions are not quantitative in nature, their answer will impact any actionable solution.

15.4.3 Multiple Imputation

Another drawback of imputation is that it tends to increase the noise in the data, because the imputed data is treated as the *actual* data.

11: “There ain’t no such thing as a free lunch” – there is no guarantee that a method that works best for a dataset works even reasonably well for another.

12: Analysts cannot simply hide their heads in the sand on this topic: if the data science objectives are incompatible with the units’ well-being, it is the objectives that need to change – we cannot ask the entities represented by those units to “get over it”.

In **multiple imputation**, the impact of that noise can be reduced by consolidating the analysis outcome from multiple imputed datasets. Once an imputation strategy has been selected on the basis of the (assumed) missing value mechanism,

1. the imputation process is repeated m times to produce m versions of the dataset (assuming a stochastic procedure – if the imputed dataset is always the same, this procedure is worthless);
2. each of these datasets is analyzed, yielding m outcomes, and
3. the m outcomes are pooled into a single result for which the mean, variance, and confidence intervals are known.

On the plus side, multiple imputation is **easy to implement**, **flexible**, as it can be used in a most situations (MCAR, MAR, even NMAR in certain cases), and it accounts for **uncertainty** in the imputed values.

However, m may need to be quite **large** when the values are missing in large quantities from many of the dataset's features, which can substantially slow down the analyses.

There may also be additional technical challenges when the output of the analyses is not a single value but some more complicated object. A generalization of multiple imputation was used by Transport Canada to predict the Blood Alcohol Level (BAC) content level in fatal traffic collisions that involved pedestrians [1].

15.5 Anomalous Observations

The most exciting phrase to hear [...], the one that heralds the most discoveries, is not "Eureka!" but "That's funny..." [I. Asimov (attributed)].

Outlying observations are data points which are **atypical** in comparison to the unit's remaining features (*within-unit*), or in comparison to the measurements for other units (*between-units*), or as part of a collective subset of observations. Outliers are thus observations which are **dissimilar to other cases** or which contradict **known dependencies/rules**.¹³

13: Outlying observations may be anomalous along any of the individual variables, or in combination.

Note that observations could be anomalous in one context, but not in another. Consider, for instance, an adult male who is 6 feet tall. Such a man would fall in the 86th percentile among Canadian males [6], which, while on the tall side, is not unusual; in Bolivia, however, the same man would land in the 99.9th percentile [6], which would mark him as extremely tall and quite dissimilar to the rest of the population.¹⁴

14: Anomaly detection points towards interesting questions for analysts and subject matter experts: in this case, why is there such a large discrepancy in the two groups?

A common mistake that analysts make when dealing with outlying observations is to remove them from the dataset without carefully studying whether they are **influential data points**, that is, observations whose absence leads to **markedly different** analysis results.

When influential observations are identified, remedial measures (such as data transformation strategies) may need to be applied to minimize any undue effect. Outliers may be influential, and influential data points may be outliers, but the conditions are neither necessary nor sufficient.

15.5.1 Anomaly Detection

By definition, anomalies are **infrequent** and typically surrounded by **uncertainty** due to their relatively low numbers, which makes it difficult to differentiate them from banal **noise** or **data collection errors**.

Furthermore, the boundary between normal and deviant observations is usually **fuzzy**; with the advent of e-shops, for instance, a purchase which is recorded at 3AM local time does not necessarily raise a red flag anymore.

When anomalies are actually associated to **malicious activities**, they are more than often **disguised** in order to blend in with normal observations, which obviously complicates the detection process.

Numerous methods exist to identify anomalous observations; **none of them are foolproof** and judgement must be used. Methods that employ graphical aids (such as box-plots, scatterplots, scatterplot matrices, and 2D tours) to identify outliers are particularly easy to implement, but a low-dimensional setting is usually required for ease of interpretation.

Analytical detection methods also exist (using Cooke's or Mahalanobis' distances, for instance), but in general some additional level of analysis must be performed, especially when trying to identify influential points (*cf.* **leverage**, Chapter 8, *Classical Regression Analysis*).

With small datasets, anomaly detection can be conducted on a case-by-case basis, but with large datasets, the temptation to use **automated detection/removal** is strong – care must be exercised before the analyst decides to go down that route.¹⁵

In the early stages of anomaly detection, **simple data analyses** (such as descriptive statistics, 1- and 2-way tables, and traditional visualizations) may be performed to help identify anomalous observations, or to obtain insights about the data, which could eventually lead to modifications of the analysis plan.

15: This stems partly from the fact that once the “anomalous” observations have been removed from the dataset, previously “regular” observations can become anomalous in turn in the smaller dataset; it is not clear when that runaway train will stop.

15.5.2 Outlier Tests

How are outliers *actually* detected? Most methods come in one of two flavours: **supervised** and **unsupervised** (we will discuss those in detail in later sections).

Supervised methods use a historical record of **labeled** (that is to say, previously identified) anomalous observations to build a **predictive classification or regression model** which estimates the probability that a unit is anomalous; domain expertise is required to gate the data.

Since anomalies are typically **infrequent**, these models often also have to accommodate the **rare occurrence problem**.¹⁶

Unsupervised methods, on the other hand, use no previously labeled information or data, and try to determine if an observation is an outlying one solely by comparing its behaviour to that of the other observations. The following traditional methods and tests of outlier detection fall into this category:¹⁷

16: Supervised models are built to minimize a cost function; in default settings, it is often the case that the mis-classification cost is assumed to be symmetrical, which can lead to technically correct but useless solutions. For instance, the vast majority (99.999+%) of air passengers emphatically do not bring weapons with them on flights; a model that predicts that no passenger is attempting to smuggle a weapon on board a flight would be 99.999+% accurate, but it would miss the point completely.

17: Note that **normality** of the underlying data is an assumption for most tests; how robust these tests are against departures from this assumption depends on the situation.

- Perhaps the most commonly-used test is **Tukey's boxplot test**; for normally distributed data, regular observations typically lie between the **inner fences**

$$Q_1 - 1.5(Q_3 - Q_1) \quad \text{and} \quad Q_3 + 1.5(Q_3 - Q_1).$$

Suspected outliers lie between the inner fences and their respective **outer fences**

$$Q_1 - 3(Q_3 - Q_1) \quad \text{and} \quad Q_3 + 3(Q_3 - Q_1).$$

Points beyond the outer fences are identified as **outliers** (Q_1 and Q_3 represent the data's 1st and 3rd quartile; see Figure 15.5).

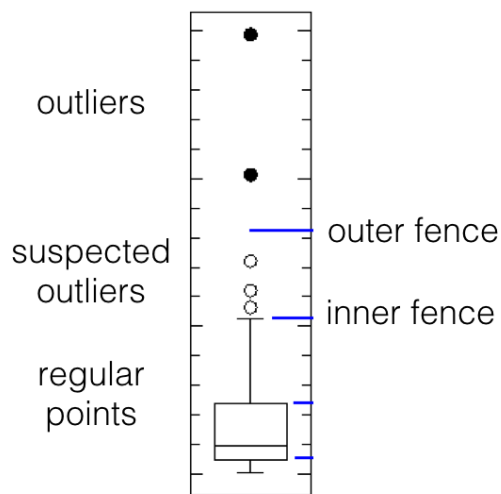
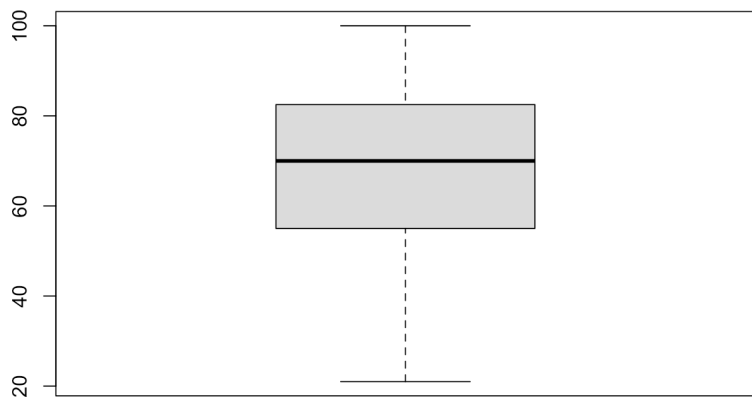


Figure 15.5: Tukey's boxplot test; suspected outliers are marked by white disks, outliers by black disks [author unknown].

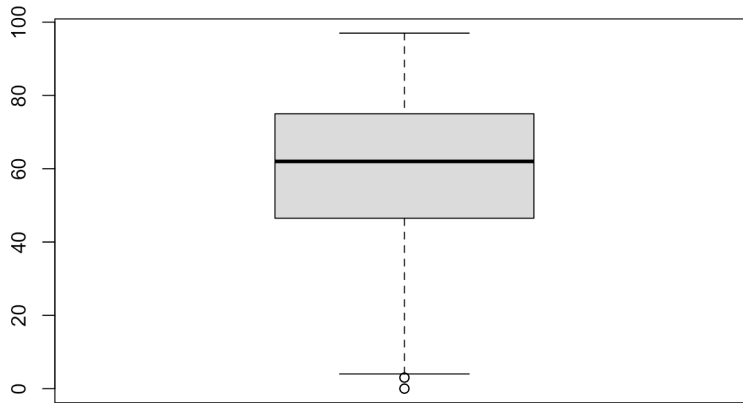
As an example, let's find the outliers for the midterm and final exam grades in Dr. Vanderwhede's *Advanced Retroencabulation* course. There are no boxplot anomalies for midterm grades:

```
boxplot(grades$MT)
```



but there are 4 boxplot anomalies for final exam grades:

```
boxplot(grades$FE)
boxplot.stats(grades$FE)$out
```



```
[1] 3 0 3 0
```

The corresponding observations can be found as follows:

```
out <- boxplot.stats(grades$FE)$out
out_ind <- which(grades$FE %in% c(out))
grades[out_ind,]
```

student	MT	FE	student	MT	FE
44	97	3	161	25	3
46	55	0	163	27	0

- The **Grubbs test** is another univariate test, which takes into consideration the number of observations in the dataset. Let x_i be the value of feature X for the i^{th} unit, $1 \leq i \leq N$, let (\bar{x}, s_x) be the mean and standard deviation of feature X , let α be the desired significance level, and let $T(\alpha, N)$ be the critical value of the Student t -distribution at significance $\alpha/2N$. Then, the i^{th} unit is an **outlier along feature X** if

$$|x_i - \bar{x}| \geq \frac{s_x(N-1)}{\sqrt{N}} \sqrt{\frac{T^2(\alpha, N)}{N-2+T^2(\alpha, N)}}.$$

- Other common tests include:
 - the **Mahalanobis distance**, which is linked to the leverage of an observation (a measure of influence), can also be used to find multi-dimensional outliers, when all relationships are linear (or nearly linear);
 - the **Tietjen-Moore test**, which is used to find a specific number of outliers;
 - the **generalized extreme studentized deviate** test, if the number of outliers is unknown;
 - the **chi-square** test, when outliers affect the goodness-of-fit, as well as
 - DBSCAN and other **clustering-based** outlier detection methods.

We will have a lot more to say on the topic in Chapter 26 (*Anomaly Detection and Outlier Analysis*).

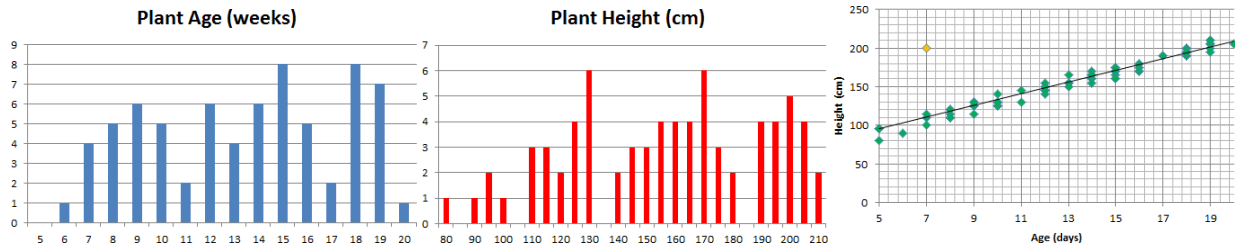


Figure 15.6: Summary visualisations for an artificial plant dataset: age distribution (left), height distribution (middle), height vs. age, with linear trend (right).

15.5.3 Visual Outlier Detection

The following three (simple) examples illustrate the principles underlying visual outlier and anomaly detection.

Example: On a specific day, the height of several plants are measured. The records also show each plant's age (the number of weeks since the seed has been planted).

Histograms of the data are shown in Figure 15.6 (age on the left, height in the middle).

Very little can be said about the data at that stage: the age of the plants (controlled by the nursery staff) seems to be somewhat haphazard, as does the response variable (height). A scatter plot of the data (rightmost chart in Figure 15.6), however, reveals that growth is strongly correlated with age during the early period of a plant's life for the observations in the dataset; points clutter around a linear trend. One point (in yellow) is easily identified as an **outlier**.

There are (at least) two possibilities: either that measurement was botched or mis-entered in the database (representing an invalid entry), or that one specimen has experienced unusual growth (outlier). Either way, the analyst has to investigate further.

Example: a government department has 11 service points in a jurisdiction. Service statistics are recorded: the monthly average arrival rates per teller and average service rates per teller are available for each service point.

A scatter plot of the service rate per teller (y axis) against the arrival rate per teller (x axis), with linear regression trend, is shown in the leftmost chart in Figure 15.7. The trend inches upwards with increasing x values.

A similar chart, but with the left-most point removed from consideration, is shown in the middle chart of Figure 15.7. The trend still slopes upward, but the fit is significantly improved, suggesting that the removed observation is unduly **influential** (or anomalous) – a better understanding of the relationship between arrivals and services is afforded if it is set aside.

Any attempt to fit that data point into the model must take this information into consideration. Note, however, that influential observations depend on the analysis that is ultimately being conducted – a point may be influential for one analysis, but not for another.

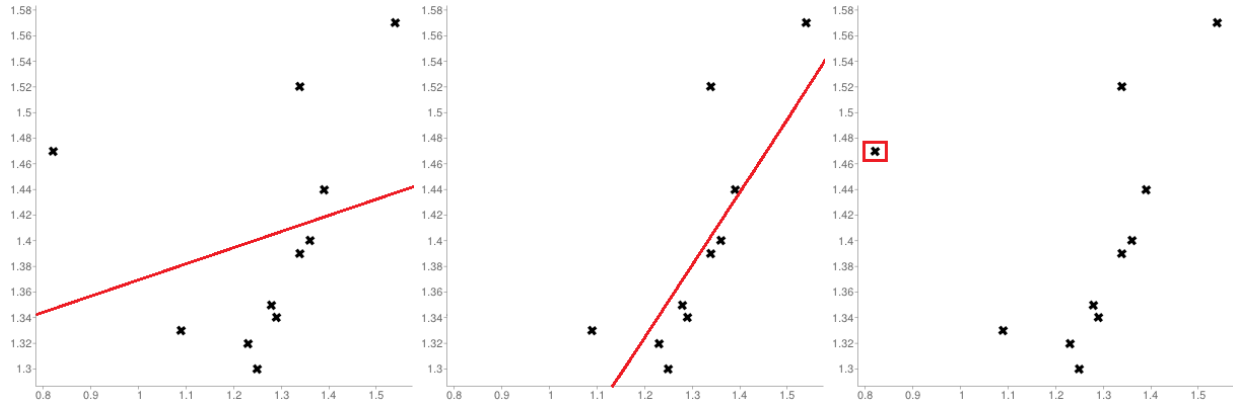


Figure 15.7: Visualisations for an (artificial) service point dataset: trend for 11 service points (left), trend for 10 service points (middle), influential observations (right).

Example: measurements of the length of the appendage of a certain species of insect have been made on 71 individuals. Descriptive statistics have been computed; the results are shown in Table 15.5.

<i>Appendage length (mm)</i>	
Mean	10.35
Standard Deviation	16.98
Kurtosis	16.78
Skewness	4.07
Minimum	0
First Quartile	0
Median	8.77
Third Quartile	10.58
Maximum	88
Range	88
Interquartile Range	10.58
Mode	0
Count	71

Table 15.5: Descriptive statistics for an (artificial) appendage length dataset.

Analysts who are well-versed in statistical methods might recognize the tell-tale signs that the distribution of appendage lengths is likely to be asymmetrical and to have a “fat” tail.¹⁸

The mode, minimum, and first quartile values belong to individuals without appendages, so there appears to be at least two sub-groups in the population (perhaps split along the lines of juveniles/adults, or males/females).

The maximum value has already been seen to be quite large compared to the rest of the observations, which at first suggests that it might belong to an **outlier**.

The histogram of the measurements, however, shows that there are 3 individuals with very long appendages (see the chart in Figure 15.8): it now becomes plausible for these anomalous entries to belong to individuals from a different species altogether who were **erroneously added** to the dataset. This does not, of course, constitute a proof of such an error, but it raises the possibility, which is often the best that an analyst can do in the absence of subject matter expertise.

18: Since the skewness is non-negligible, and due to the kurtosis being commensurate with the mean and the standard deviation, the range being so much larger than the interquartile range, and the maximum value being so much larger than the third quartile.

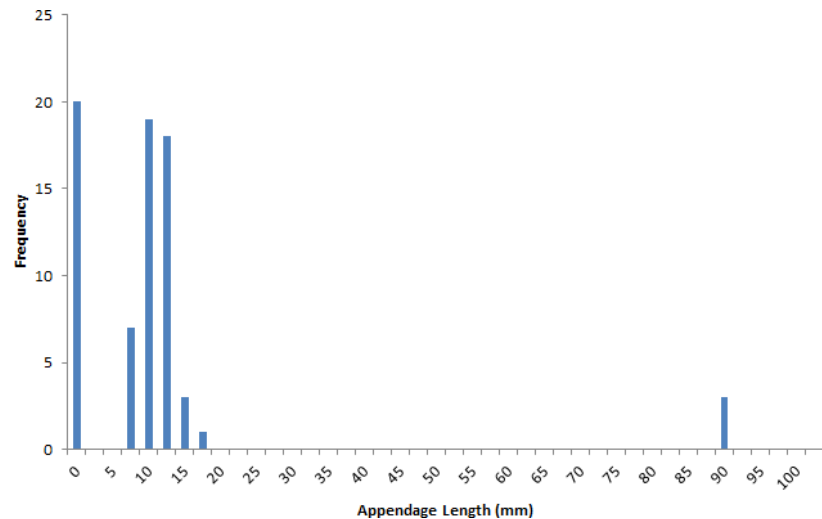


Figure 15.8: Frequency chart of the appendage lengths in the artificial dataset.

15.6 Data Transformations

History is the transformation of tumultuous conquerors into silent footnotes. [P. Eldridge]

This **crucial** step is often neglected or omitted altogether. Various transformation methods are available, depending on the analysts' needs and data types, including:

- **standardization** and **unit conversion**, which put the dataset's variables on an equal footing – a requirement for basic comparison tasks and more complicated problems of clustering and similarity matching;
- **normalization**, which attempts to force a variable into a normal distribution – an assumption which must be met in order to use number of traditional analysis methods, such as ANOVA or regression analysis, and
- **smoothing methods**, which help remove unwanted noise from the data, but at a price – perhaps removing natural variance in the data.

Another type of data transformation is pre-occupied with the concept of **dimensionality reduction**. There are many advantages to working with low-dimensional data:

- **visualization methods** of all kinds are available to extract and present insights out of such data;
- high-dimensional datasets are subject to the so-called **curse of dimensionality**, which asserts (among other things) that multi-dimensional spaces are vast, and when the number of features in a model increases, the number of observations required to maintain predictive power also increases, but at a **substantially higher rate** (see Figure 15.9),
- another consequence of the curse is that in high-dimension sets, all observations are roughly **dissimilar** to one another – observations tend to be nearer the dataset's boundaries than they are to one another.

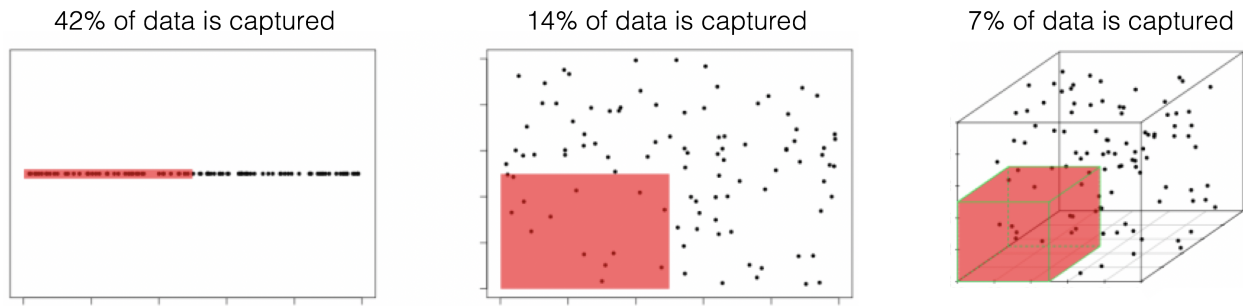


Figure 15.9: Illustration of the curse of dimensionality; $N = 100$ observations are uniformly distributed on the unit hypercube $[0, 1]^d$, $d = 1, 2, 3$. The red regions represent the smaller hypercubes $[0, 0.5]^d$, $d = 1, 2, 3$. The percentage of captured datapoints is seen to decrease with an increase in d [7].

Dimension reduction techniques such as:

- **principal component analysis, independent component analysis, and factor analysis** for numerical data, or
- **multiple correspondence analysis** for categorical data

project multi-dimensional datasets onto low-dimensional but high information spaces;¹⁹ feature selection techniques, including the popular family of **regularization methods** (see Chapter 20, *Regression and Value Estimation*) select an **optimal subset of variables** with which to accomplish tasks, according to some appropriate, context-dependent criterion.

19: The so-called **Manifold Hypothesis**.

Some information is necessarily lost in the process, but in many instances the drain can be kept under control and the gains made by working with smaller datasets can offset the losses of completeness. We will have more to say on the topic in Chapter 23 (Feature Selection and Dimension Reduction).

15.6.1 Common Transformations

Models often require that certain data assumptions be met. For instance, ordinary least square regression assumes:

- that the response variable is a **linear combination** of the predictors;
- **constant** error variance;
- **uncorrelated residuals**, which may or may not be statistically independent,
- etc.

In reality, it is rare that raw data meets all these requirements, but that does not necessarily mean that we need to abandon the model – an **invertible** sequence of data transformations may produce a derived data set which *does* meet the requirements, allowing the consultant to draw conclusions about the original data.

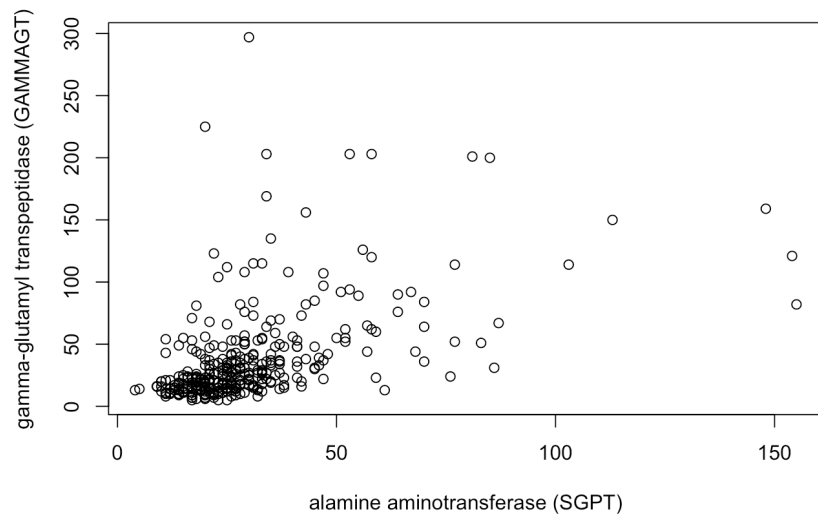
In the regression context, invertibility is guaranteed by **monotonic** transformations: identity, logarithmic, square root, inverse (all members of the power transformations), exponential, etc.

These transformations are illustrated below on a subset of the BUPA **liver disease dataset** [4].

Subset of the BUPA liver disease dataset

```
library(kernldwd)
data(BUPA)
plot(BUPA$X[,3],BUPA$X[,5],
     main="Scatterplot of a subset of the BUPA dataset",
     xlab="alamine aminotransferase (SGPT)",
     ylab="gamma-glutamyl transpeptidase (GAMMAGT)")
```

Scatterplot of a subset of the BUPA dataset



In Figure 15.10, we show the effect of various transformations on $X = \text{SGPT}$ and $Y = \text{GAMMAGT}$.

There are rules of thumb and best practices to transform data, but analysts should not discount the importance of exploring the data visually before making a choice.

Transformations on the **predictors** X may be used to achieve the **linearity assumption**, but they usually come at a price – Pearson correlations are not preserved by such transformations, for instance.²⁰

Transformations on the target Y can help with **non-normality** of residuals and **non-constant variance** of error terms.

Note that transformations can be applied **both** to the target variable or the predictors: as an example, if the linear relationship between two variables X and Y is expressed as $Y = a + bX$, then a unit increase in X is associated with an average of b units in Y .

But a better fit might be provided by either of

$$\log Y = a + bX, \quad Y = a + b \log X, \quad \text{or} \quad \log Y = a + b \log X,$$

for which:

- a unit increase in X is associated with an average $b\%$ increase in Y ;
- a 1% increase in X is associated with an average $0.01b$ unit increase in Y , and
- a 1% increase in X is associated with a $b\%$ increase in Y , respectively.

20: Spearman correlations are preserved (in magnitude) by monotonous transformations, however.

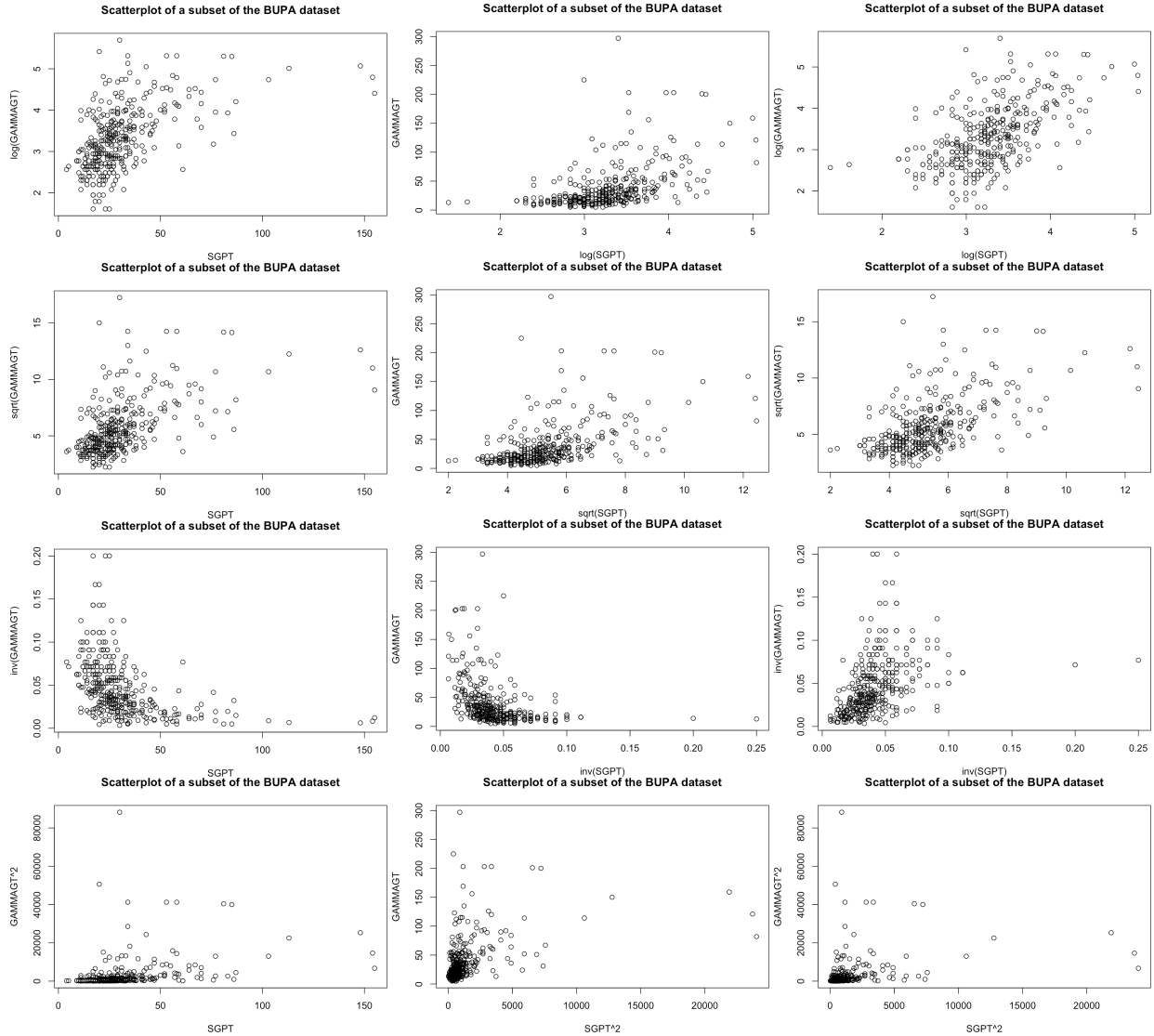


Figure 15.10: Various data transformations for a subset of the BUPA liver disease dataset [4], involving the logarithm, the square root, the inverse, and the square of both variables (see the axes for the specific transformation).

15.6.2 Box-Cox Transformations

There is a useful framework that provides an optimal transformation, in a certain sense. Consider the task of predicting the target Y with the help of the predictors $X_j, j = 1, \dots, p$. The usual model takes the form

$$y_i = \sum_{j=1}^p \beta_j X_{j,i} + \varepsilon_i, \quad i = 1, \dots, n.$$

If the residuals are skewed, or their variance is not constant, or the trend itself does not appear to be linear, a power transformation on the response might be indicated, but if so, which one? The **Box-Cox transformation** $y_i \mapsto y'_i(\lambda)$, $y_i > 0$ is defined by

$$y'_i(\lambda) = \begin{cases} (y_1 \dots y_n)^{1/n} \ln y_i, & \text{if } \lambda = 0 \\ \frac{y_i^\lambda - 1}{\lambda} (y_1 \dots y_n)^{\frac{1-\lambda}{n}}, & \text{if } \lambda \neq 0 \end{cases}$$

The **suggested** choice of λ is the value that maximizes the log-likelihood

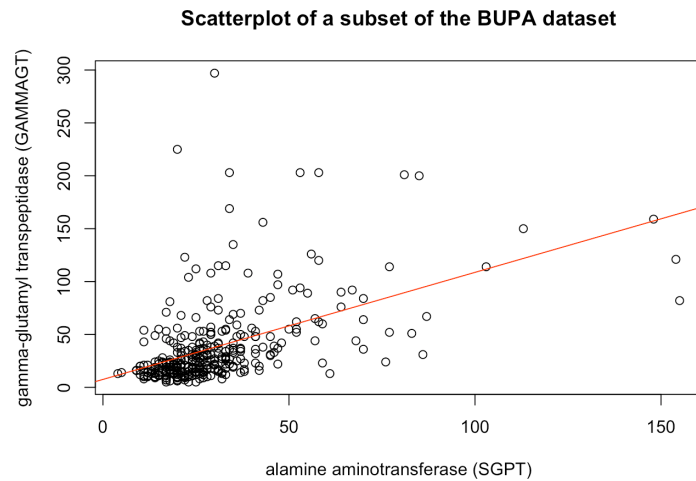
$$\mathcal{L} = -\frac{n}{2} \log \left(\frac{2\pi\hat{\sigma}^2}{(y_1 \dots y_n)^{2(\lambda-1)/n}} + 1 \right).$$

The following code shows the effect of the Box-Cox transformation on the linear fit of Y (GAMMAGT) against X (SGPT) in the BUPA dataset.²¹

21: Assume that `library(kernwd)` and `data(BUPA)` have already been called.

Linear fit in the BUPA liver disease dataset

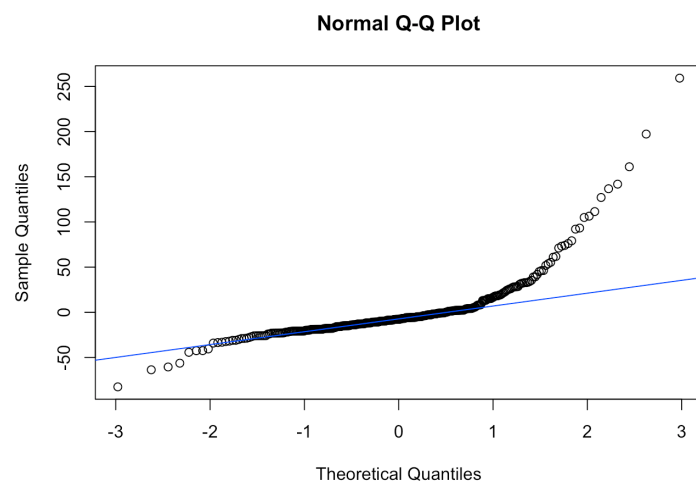
```
model <- lm(BUPA$X[,5] ~ BUPA$X[,3])
plot(BUPA$X[,3], BUPA$X[,5],
     main="Scatterplot of a subset of the BUPA dataset",
     xlab="alamine aminotransferase (SGPT)",
     ylab="gamma-glutamyl transpeptidase (GAMMAGT)")
abline(a=model[[1]][1], b=model[[1]][2], col="red")
```



The fit looks decent, but the qq -plot of the residuals makes it clear that the normality assumption of the linear regression model is not met.

QQ plot of the untransformed BUPA model

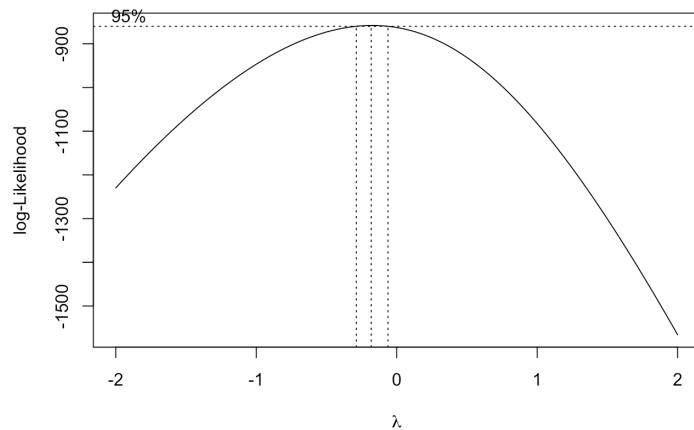
```
qqnorm(model$residuals)
qqline(model$residuals, col="blue")
```



We find the Box-Cox transformation on Y as follows:

Linear fit in the Box-Cox transformed BUPA model

```
library(MASS)
box.cox <- boxcox(BUPA$X[,5] ~ BUPA$X[,3])
(lambda <- box.cox$x[which.max(box.cox$y)])
```



```
[1] -0.1818182
```

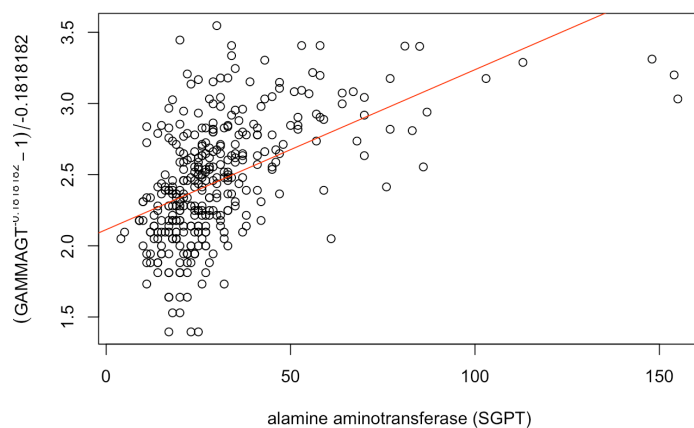
The linear model on the Box-Cox transformed response is given as follows.

Linear fit in the Box-Cox transformed BUPA model

```
box.cox.Y <- (BUPA$X[,5]^lambda-1)/lambda
bc.model <- lm(box.cox.Y ~ BUPA$X[,3])

plot(BUPA$X[,3],box.cox.Y,
     main="Scatterplot of a subset of the BUPA dataset",
     xlab="alamine aminotransferase (SGPT)",
     ylab="Box-Cox response")
abline(a=bc.model[[1]][1], b=bc.model[[1]][2], col="red")
```

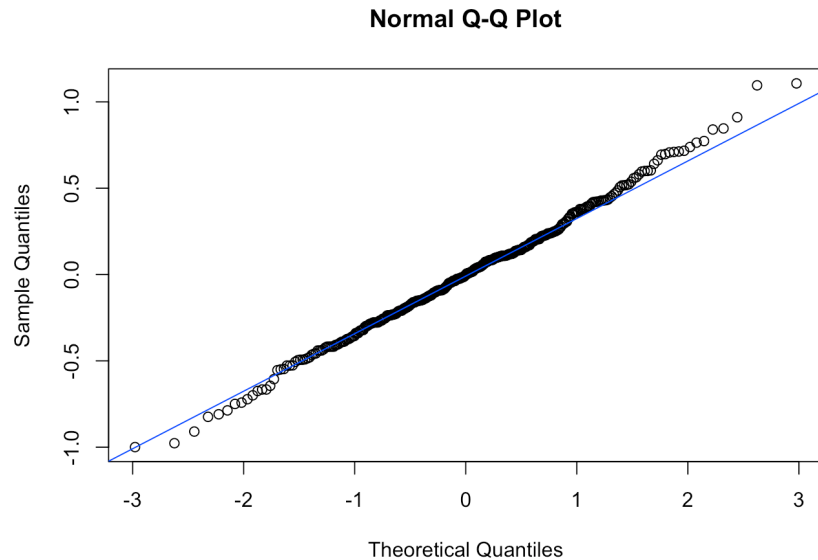
Scatterplot of a subset of the BUPA dataset



That the model on the Box-Cox data is better is evidenced by the qq -plot.

QQ plot of the transformed BUPA model

```
qqnorm(bc.model$residuals)
qqline(bc.model$residuals, col="blue")
```



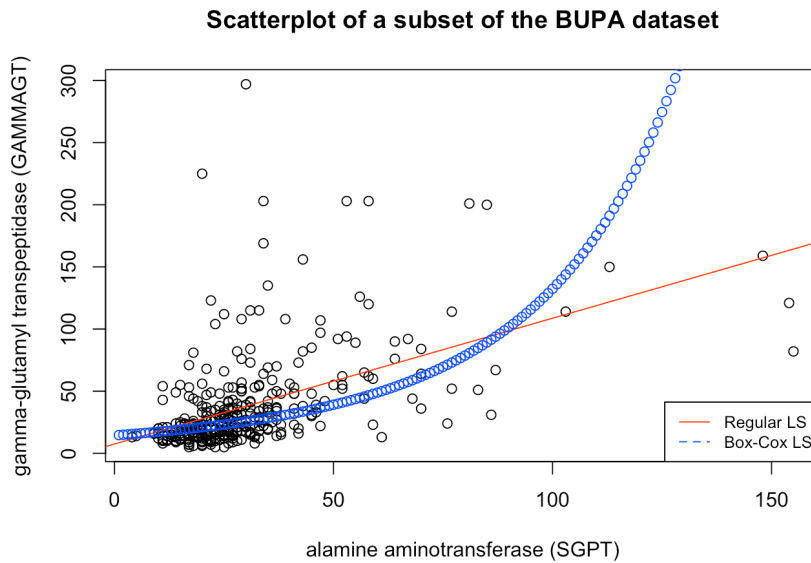
There might be theoretical rationales which favour a particular choice of λ – these are not to be ignored. It is also important to produce a residual analysis, as the best Box-Cox choice does not necessarily meet all the least squares assumptions.

Finally, it is important to remember that the resulting parameters have the least squares property **only with respect to the transformed data points** (in other words, the inverse transformation has to be applied to the results before we can make interpretations about the original data).

In the BUPA example, the corresponding curve in the untransformed space is shown below.

Linear Box-Cox model in the untransformed BUPA data

```
plot(BUPA$X[,3],BUPA$X[,5],
     main="Scatterplot of a subset of the BUPA dataset",
     xlab="alamine aminotransferase (SGPT)",
     ylab="gamma-glutamyl transpeptidase (GAMMAGT)")
df <- data.frame(order(BUPA$X[,3]),
                  (lambda * (bc.model[[1]][[1]][1] +
                             bc.model[[1]][[2]][1] * order(BUPA$X[,3]))
                  + 1)^(1/lambda))
abline(a=model[[1]][1], b=model[[1]][2], col="red")
points(df, col='blue', pch=1)
legend("bottomright", legend=c("Regular LS", "Box-Cox LS"),
      col=c("red", "blue"), lty=1:2, cex=0.8)
```



15.6.3 Scaling

Numeric variables may have different scales (weights and heights, for instance). Since the variance of a large-range variable is typically greater than that of a small-range variable, leaving the data **unscaled** may introduce biases, especially when using unsupervised methods.²²

22: See Chapter 19, *Machine Learning* 101.

It could also be the case that it is the relative positions (or rankings) which is of importance, in which case it could become important to look at relative distances between levels:

- **standardisation** creates a variable with mean 0 and std deviation 1:

$$Y_i = \frac{X_i - \bar{X}}{s_X},$$

- **normalization** creates a variable in the range [0, 1]:

$$Y_i = \frac{X_i - \min\{X_k\}}{\max\{X_k\} - \min\{X_k\}}.$$

There are other options; different schemes can lead to different outputs.

15.6.4 Discretizing

In order to reduce computational complexity, a numeric variable may need to be replaced with an **ordinal** variable (*height* values could be replaced by the qualitative “short”, “average”, and “tall”, for instance).²³

It is far from obvious how to determine the bins’ limits – **domain expertise** can help, but it could introduce unconscious bias to the analyses. In the absence of such expertise, limits can be set so that either the bins each:

- contain (roughly) the same **number of observations**;
- have the same **width**, or
- the performance of some modeling tool is maximized.

Again, various choices may lead to different outputs.

23: Of course, what these terms represent depend on the context; Canadian short and Bolivian tall may be fairly commensurate, to revisit the example at the start of the preceding section.

15.6.5 Creating Variables

Finally, it is possible that new variables may need to be introduced (in contrast with dimensionality reduction). These new variables may arise:

- as **functional relationships** of some subset of available features (introducing powers of a feature, or principal components, say);
- because the modeling tool may require **independence of observations** or **independence of features** (in order to remove multicollinearity, for instance), or
- to simplify the analysis by looking at **aggregated summaries** (often used in text analysis).

There is no limit to the number of new variables that can be added to a dataset – but consultants should strive for **relevant additions**.

15.7 Example: Algae Blooms

This example is based on a Case Study by L. Torgo [11]. It provides a concrete illustration of the data preparation process on a realistic dataset: `algae_blooms.csv`, which is also available at the [UCI Machine Learning Repository](#) [↗](#).

The ultimate problem is to predict the occurrence of harmful algae in water samples. Torgo also uses it to highlight various aspects of **data exploration**, **data cleaning**, and **R syntax**.²⁴

Readers who would prefer to try this example on their own are invited to skip this section and head to the first exercise of Section 15.8.

15.7.1 Problem Description

The ability to monitor and perform early forecasts of various river algae blooms is crucial to control the ecological harm they can cause.

The dataset which is used to train the learning model consists of:

- chemical properties of various water samples of European rivers
- the quantity of seven algae in each of the samples, and
- the characteristics of the collection process for each sample.

What is the data science motivation for such a model? After all, we **can** analyze water samples to determine if various harmful algae are present or absent.

The answer is simple: chemical monitoring is **cheap** and **easy to automate**, whereas biological analysis of samples is **expensive** and **slow**.

Another answer is that analyzing the samples for harmful content does not provide a better understanding of algae **drivers**: it just tells us which samples contain algae.

24: We will continue this work in Section 20.6.

15.7.2 Loading the Data

Before we can take a look at the data and begin the process in earnest, we need to load it in the R workspace. If the dataset was downloaded from the UCIML repository and stored in the CSV file `algae_blooms.csv`, we can run the following:

```
algae_blooms <- read.csv("algae_blooms.csv", sep=",",
                        stringsAsFactors = TRUE, header=TRUE)
```

It is also available in Torgo's `DMwR` package. As we will use some of its functions in this example, we will show how to install and load it. Unfortunately, it could not be installed directly from CRAN with the `install.packages()` function, as of January 2023.

Instead, we suggest doing the following:

1. Download the package source `DMwR_0.4.1.tar.gz` from the [DMwR CRAN archive page](#) (additional information about the package is also available there) and save it locally to some path (in this example, the file was saved to the folder `docs/code/`).
2. Install the following dependencies directly from CRAN:

```
install.packages(c("xts", "quantmod", "ROCR"))
```

The dependencies list might be different, based on the packages already installed locally; any eventual error message in the next step will inform you of the exact dependencies to install.

3. Install `DMwR` from the package source:

```
install.packages("docs/code/DMwR_0.4.1.tar.gz",
                 repos=NULL, type="source")
```

4. Load the package and prepare the data:

```
library(DMwR)
algae_blooms <- as.data.frame(rbind(DMwR:algae,
                                    DMwR:algae.sols))
```

Either way, we can get a sense for the data frame's structure by calling the `str` function.

```
str(algae_blooms)
```

```
'data.frame':  340 obs. of  18 variables:
 $ season: Factor w/ 4 levels "winter" "spring" "autumn" "spring" ...
 $ size  : Factor w/ 3 levels "small" "small" "small" "small" ...
 $ speed : Factor w/ 3 levels "medium" "medium" "medium" "medium" ...
 $ mxPH  : num  8 8.35 8.1 8.07 8.06 8.25 8.15 8.05 8.7 7.93 ...
 $ mnO2  : num  9.8 8 11.4 4.8 9 13.1 10.3 10.6 3.4 9.9 ...
 $ Cl    : num  60.8 57.8 40 77.4 55.4 ...
 $ NO3   : num  6.24 1.29 5.33 2.3 10.42 ...
```

```

$ NH4 : num  578 370 346.7 98.2 233.7 ...
$ oP04 : num  105 428.8 125.7 61.2 58.2 ...
$ P04 : num  170 558.8 187.1 138.7 97.6 ...
$ Chla : num  50 1.3 15.6 1.4 10.5 ...
$ a1 : num  0 1.4 3.3 3.1 9.2 15.1 2.4 18.2 25.4 17 ...
$ a2 : num  0 7.6 53.6 41 2.9 14.6 1.2 1.6 5.4 0 ...
$ a3 : num  0 4.8 1.9 18.9 7.5 1.4 3.2 0 2.5 0 ...
$ a4 : num  0 1.9 0 0 0 0 3.9 0 0 2.9 ...
$ a5 : num  34.2 6.7 0 1.4 7.5 22.5 5.8 5.5 0 0 ...
$ a6 : num  8.3 0 0 0 4.1 12.6 6.8 8.7 0 0 ...
$ a7 : num  0 2.1 9.7 1.4 1 2.9 0 0 0 1.7 ...

```

Notes:

- 3 of the fields are categorical (season, size, speed, which refer to the data collection process);
- of the numerical fields, 8 have vaguely “chemical” names;
- presumably, the remaining fields refer to the various algae blooms.

We can get a better feel for the data frame by observing it in its natural habitat, so to speak, using the `head()` or `tail()` functions.

```
head(algae_blooms,4)
```

```

season  size    speed mxPH  ...  Chla    a1    ...  a7
winter  small   medium  8.00  ...  50.0    0.0    ...  0.0
spring  small   medium  8.35  ...   1.3    1.4    ...  2.1
autumn  small   medium  8.10  ...  15.6    3.3    ...  9.7
spring  small   medium  8.07  ... 138.7    1.4    ...  1.4

```

15.7.3 Summary and Visualization

As it happens, we are not given an awful lot of information about the dataset’s **domain**.²⁵ **Data exploration**, in the form of summaries and visualization, can help provide a handle on the problem at hand.²⁶

A call to the summary function (on the next page) provides frequency counts for categorical variables, and 6-pt summaries for numerical variables. As a bonus, the number of missing values is also tabulated.²⁷

Notes:

- The *chemical* variables all have missing values, ranging from only 2 to 7, 16, and 23.
- The observations seem fairly uniformly distributed in terms of the seasons, but large rivers and low speed rivers are not represented as often as their counterparts.
- All numerical values are non-negative, which makes sense in the context of concentrations
- We do not know what the range of the chemical values *should* take in a real-world context, but some of the maximum values seem ... unrealistic (NH4!!, oPO4, a7, etc.)
- Does anything else jump at you?

25: We remain woefully ill-prepared to deal with matters of a chemical nature, to our eternal shame.

26: **IMPORTANT NOTE:** we may have given you the impression that exploration is *only really necessary* when domain expertise escapes us. Domain expertise can help analysts frame the problem and the analysis results in the appropriate manner, but it often also gives them a false sense of security. Errors can creep anywhere – data exploration at an early stage may save you a lot of embarrassing back-tracking at a later stage.

27: The default setting only lists a limited number of categorical levels – the summary documentation will explain how to increase the number of levels that are displayed.


```
summary(algae_blooms)
```

```
season      autumn spring summer winter
           80      84      86      90

size        large medium  small
           83     136    121

speed       medium   high    low
           140     142     58

           mxPH  mnO2   Cl   NO3    NH4   oP04   P04   Chla
Min.:    5.6   1.5   0.2   0.0    5.0    1.0    1.0    0.2
Q1  :    7.8   7.9  10.9   1.1   37.8   13.0   40.0    2.1
Med.:    8.0   9.7  32.4   2.3  107.3   37.2  101.5    5.1
Mean:    7.9   9.1  42.5   3.1  471.7   73.0  136.7   12.7
Q3   :    8.3  10.8  57.7   4.1  244.9   88.1  200.2   17.2
Max.:    9.7  13.4 391.5  45.6 24064.0 1435.0 1690.0  110.4
NA's:    2     2   16     2     2     2     7    23

           a1     a2     a3     a4     a5     a6     a7
Min.:    0.0    0.0    0.0    0.0    0.0    0.0    0.0
Q1  :    1.5    0.0    0.0    0.0    0.0    0.0    0.0
Med.:    7.1    2.8    1.4    0.0    2.2    0.0    0.0
Mean:   16.7    7.2    3.9    1.8    5.5    6.4    2.2
Q3   :   25.1   10.1    4.6    2.3    8.0    7.0    2.2
Max.:   89.8   72.6   42.8   44.6   61.1   77.6   31.6
```

Of course, these summaries each apply to a single variable (1-way tables).
Can we find anything else using n -way tables?²⁸

28: On categorical variables, by necessity.

2-way tables

```
table(algae_blooms$speed,algae_blooms$size)
table(algae_blooms$speed,algae_blooms$season)
table(algae_blooms$season,algae_blooms$size)
```

```
      large medium small
high    13     56    73
low     32     24     2
medium  38     56    46
```

```
      autumn spring summer winter
high    32     34     38     38
low     16     13     12     17
medium  32     37     36     35
```

```
      large medium small
autumn  19     33     28
spring  21     34     29
summer  19     36     31
winter  24     33     33
```

3-way tables

```
table(algae_blooms$season, algae_blooms$size,
      algae_blooms$speed)
```

```
, , = high
      large medium small
autumn    3      13    16
spring    3      14    17
summer    3      16    19
winter    4      13    21
```

```
, , = low
      large medium small
autumn    9       6     1
spring    7       6     0
summer    6       6     0
winter   10       6     1
```

```
, , = medium
      large medium small
autumn    7      14    11
spring   11      14    12
summer   10      14    12
winter   10      14    11
```

The 6-pt summary provides some information about the underlying distribution, but not much on the parametric front. A more traditional summary can be displayed using the `psych` library's `describe()` function.

```
psych::describe(algae_blooms)
```

(the output is shown at the top of the next page)

Notes:

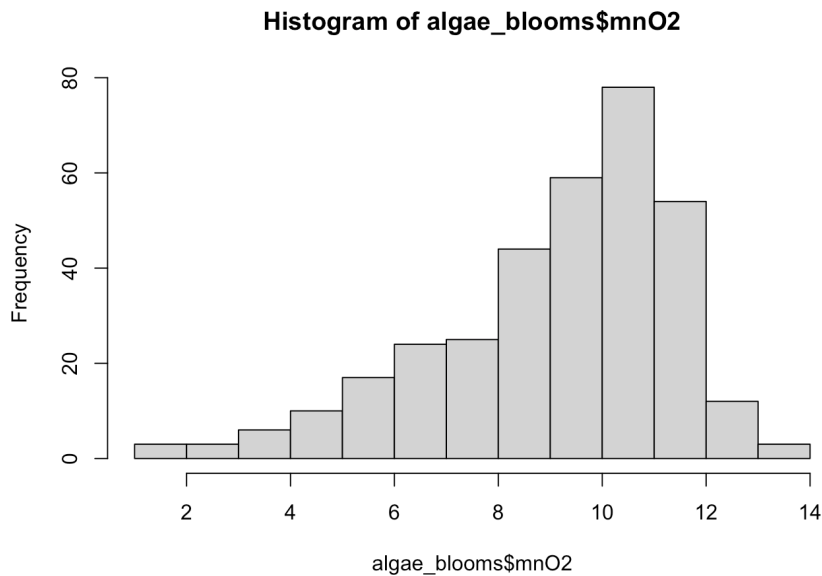
- the categorical variables are marked by an asterisk *; the levels are coded with an integer, and treated as numerical variables for the purpose of the analysis, so the results for these fields are meaningless
- the `trimmed` variable refers to the *trimmed mean*, the mean obtained when a certain percentage of the observations are removed from both end of the spectrum (what percentage, exactly?)
- the `mad` variable refers to the *median absolute deviation (from the median)*

We personally find such a table hard to read and really grasp once there are more than a few variables in the dataset. **Visualization** comes in handy in such cases.

Basic histograms can be constructed with the `hist()` function.

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
season*	1	340	2.547059	1.1186895	3.0000	2.558823	1.482600	1.000	4.000	3.000	-0.0544729	-1.3652641	0.0606695
size*	2	340	2.111765	0.7676208	2.0000	2.139706	1.482600	1.000	3.000	2.000	-0.1915031	-1.2876572	0.0416301
speed*	3	340	1.994118	0.9120437	2.0000	1.992647	1.482600	1.000	3.000	2.000	0.0115387	-1.8012592	0.0494625
mxPH	4	338	7.997293	0.5783188	8.0450	8.035864	0.467019	5.600	9.700	4.100	-0.8402202	1.9865897	0.0314564
mnO2	5	338	9.156716	2.3130799	9.7000	9.388198	1.927380	1.500	13.400	11.900	-0.9392605	0.5497464	0.1258150
Cl	6	324	42.517246	44.4906037	32.4700	34.997591	33.478591	0.222	391.500	391.278	2.8485675	14.0485533	2.4717002
NO3	7	338	3.120784	3.2851622	2.3555	2.699504	2.181646	0.000	45.650	45.650	6.7874378	81.1290663	0.1786893
NH4	8	338	471.734411	1739.0774580	107.3570	156.746838	119.949753	5.000	24064.000	24059.000	9.1184171	105.4980942	94.5933434
oPO4	9	338	73.091882	114.1420517	37.2430	51.792802	43.460936	1.000	1435.000	1434.000	5.9906708	60.0469965	6.2085091
PO4	10	333	136.685699	149.4773125	101.4550	115.867652	114.999352	1.000	1690.000	1689.000	4.2262442	35.1788385	8.1913063
Chla	11	317	12.796196	18.0813363	5.1110	8.782608	6.094969	0.200	110.456	110.256	2.6186753	7.8575379	1.0155490
a1	12	340	16.701765	20.9987076	7.1000	12.593750	10.526460	0.000	89.800	89.800	1.5040883	1.4996118	1.1388148
a2	13	340	7.200882	10.7549412	2.8000	4.854412	4.151280	0.000	72.600	72.600	2.3280170	6.7216331	0.5832686
a3	14	340	3.904412	6.4205247	1.4000	2.358456	2.075640	0.000	42.800	42.800	2.4150758	6.7202844	0.3482018
a4	15	340	1.810000	3.8292948	0.0000	1.036765	0.000000	0.000	44.600	44.600	5.9516431	52.8944003	0.2076727
a5	16	340	5.515588	8.4186630	2.2000	3.692279	3.261720	0.000	61.100	61.100	2.7562479	10.4260075	0.4565661
a6	17	340	6.411471	12.3978237	0.0000	3.279779	0.000000	0.000	77.600	77.600	2.8805737	9.2050442	0.6723664
a7	18	340	2.206471	4.9472217	0.0000	1.040074	0.000000	0.000	31.600	31.600	4.0903606	18.5615144	0.2683008

```
hist(algae_blooms$mnO2)
```



Based on this histogram, we can conclude that the underlying distribution of `mnO2` has a **negative skew**, say, which is confirmed by the table above.

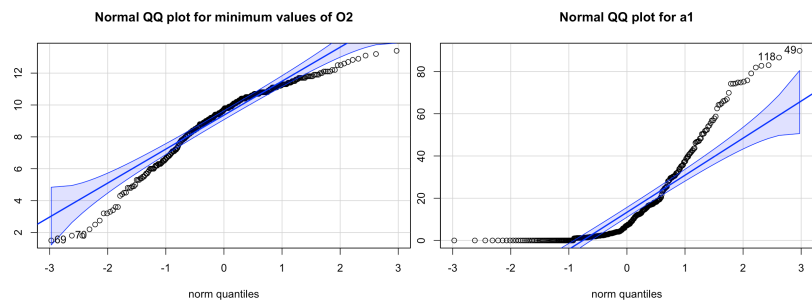
The variable `mnO2` clearly does not follow a normal distribution (it never takes on negative values, and the distribution is skewed negatively, as indications); but we see that viewing it as normal would be a much better approximation than viewing the distribution of `a1` as normal.

```
hist(algae_blooms$a1)
```



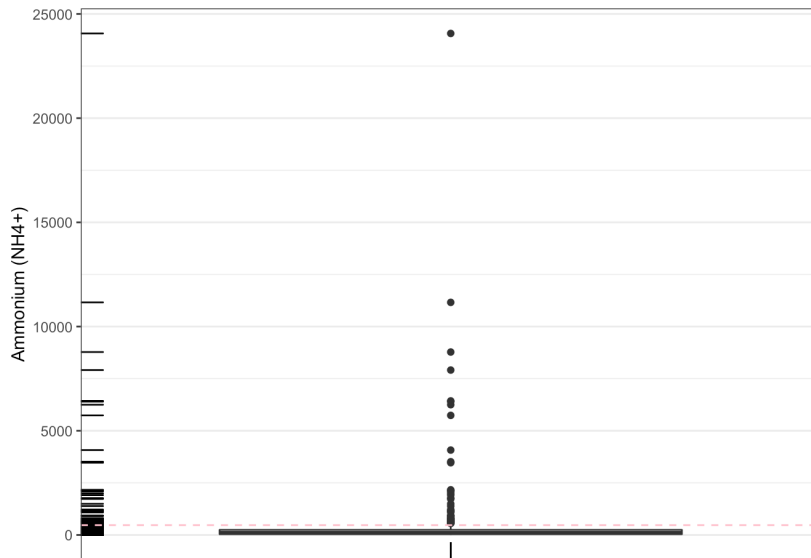
qq-plots, another traditional statistical plot, can be produced with the *car* library's `qqPlot()` function. Again, we can see that the normal distribution is not a good fit for `mnO2` (left), but the fit is even worse for `a1` (right).

```
car::qqPlot(algae_blooms$mnO2, ylab="",
  main='Normal QQ plot for minimum values of O2')
car::qqPlot(algae_blooms$a1, ylab="",
  main='Normal QQ plot for a1')
```



We can also take a look at some of the odd values for `NH4` using *ggplot2* [2, 3, 14, 13].

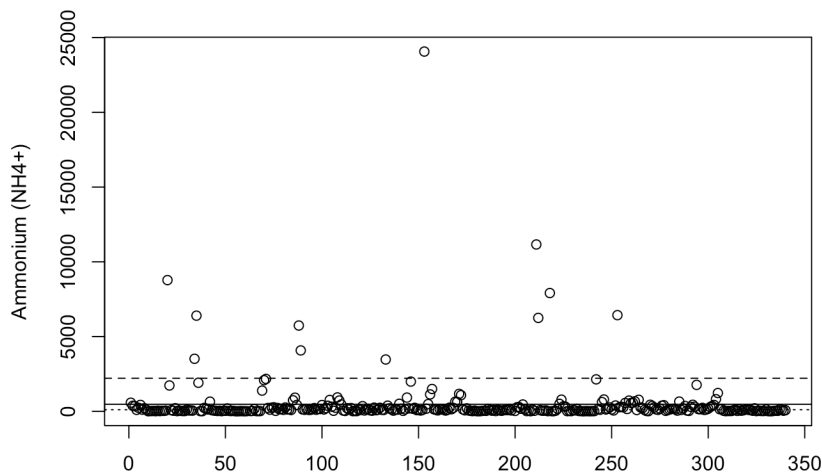
```
library(ggplot2)
ggplot(algae_blooms, aes(x=factor(0), y=NH4)) +
  geom_boxplot() + geom_rug() +
  geom_hline(aes(yintercept=mean(algae_blooms$NH4,
    na.rm=TRUE)), linetype=2, colour="pink") +
  ylab("Ammonium (NH4+)") + xlab("") +
  scale_x_discrete(breaks=NULL)
```



We see that there are a string of values falling way above the boxplot. If the underlying distribution was normal, say, these would definitely be considered outliers.

Let us investigate further.

```
plot(algae_blooms$NH4, xlab="", ylab="Ammonium (NH4+)")
abline(h=mean(algae_blooms$NH4, na.rm=TRUE), lty=1)
abline(h=mean(algae_blooms$NH4, na.rm=TRUE) +
       sd(algae_blooms$NH4, na.rm=TRUE), lty=2)
abline(h=median(algae_blooms$NH4, na.rm=TRUE), lty=3)
```



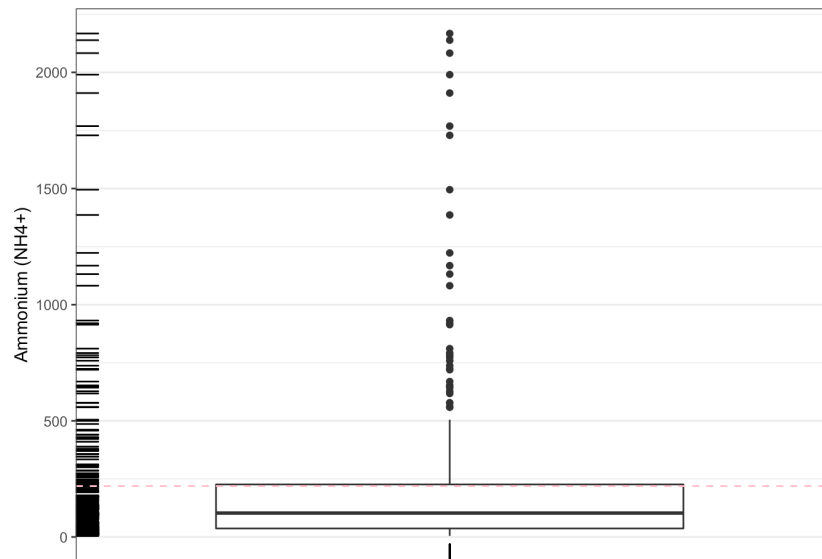
We can also look at the data and see which observations have values of NH4 below 3000 (roughly all values below the long dashed line above).

```
nrow(algae_blooms[-which(algae_blooms$NH4>3000),])
```

```
[1] 329
```

What does the boxplot above look like without the suspected outliers?

```
ggplot(algae_blooms[-which(algae_blooms$NH4>3000),],
  aes(x=factor(0),y=NH4)) +
  geom_boxplot() + geom_rug() +
  geom_hline(aes(yintercept=mean(algae_blooms[
    -which(algae_blooms$NH4>3000),8], na.rm=TRUE)),
    linetype=2, colour="pink") +
  ylab("Ammonium (NH4+)") + xlab("") +
  scale_x_discrete(breaks=NULL)
```

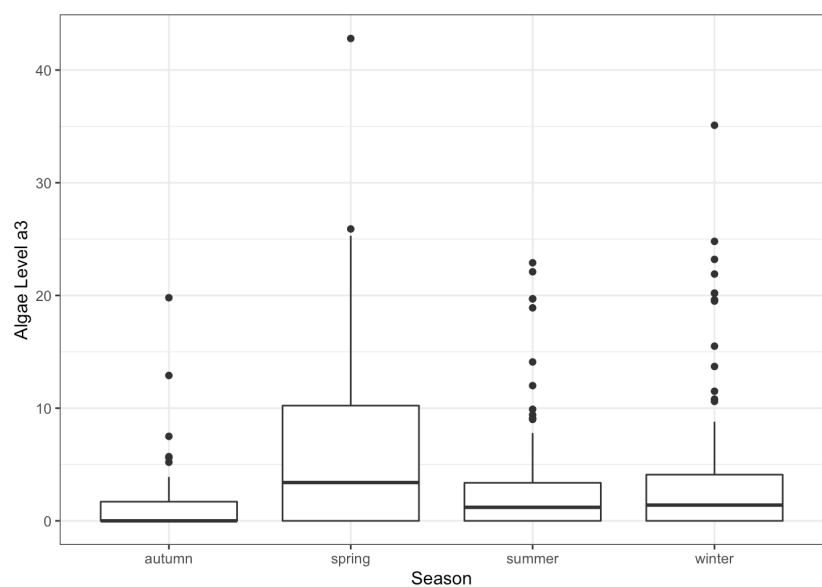


29: The box structure has expanded, and there still seems to be some very high values. Perhaps that is to be expected? How would we find out?

It is a bit better, to be sure.²⁹

Now, let us take a look at some of the algae levels.

```
ggplot(algae_blooms,aes(x=season,y=a3)) +
  geom_boxplot() + xlab("Season") + ylab("Algae Level a3")
```



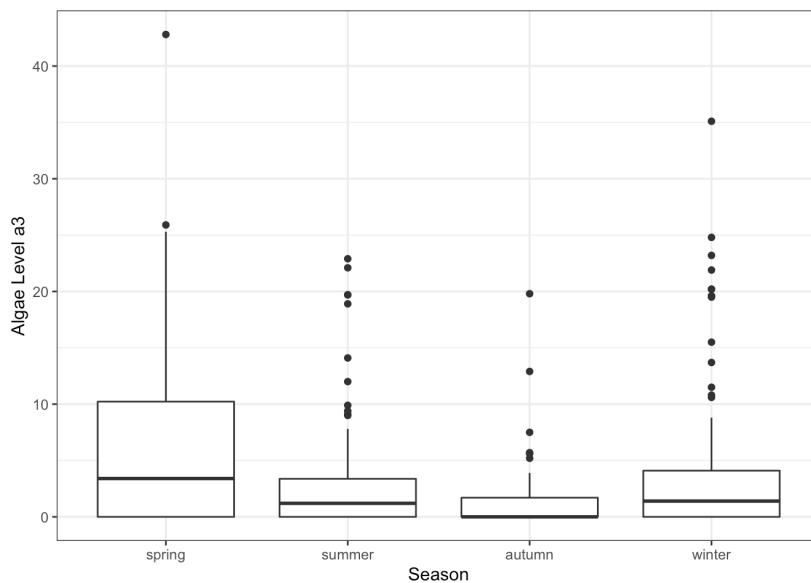
What does that tell us? It is hard to get a good handle on the situation because the season are out of sequential order.

We can re-arrange the factors, but it requires a bit of fancy footwork using the `forcats`' library `fct_relevel()` function, and `dplyr`'s `mutate()`.

```
library(forcats) # for fct_relevel
library(dplyr)   # for mutate

algae_blooms = mutate(algae_blooms,
  size=fct_relevel(size,c("small","medium","large")),
  speed=fct_relevel(speed,c("low","medium","high")),
  season=fct_relevel(season,c("spring","summer","autumn",
    "winter")))
)
```

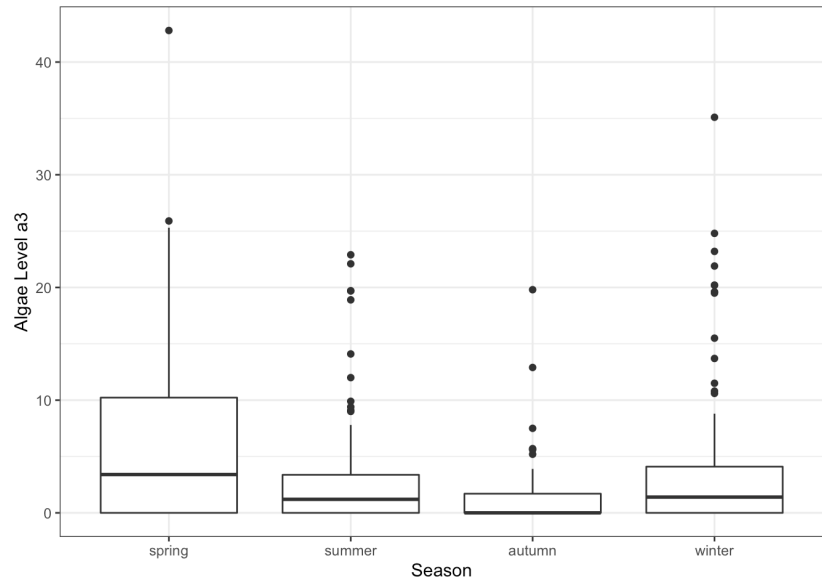
```
ggplot(algae_blooms,aes(x=season,y=a3)) +
  geom_boxplot() +
  xlab("Season") +
  ylab("Algae Level a3")
```



We only have 1 year's worth of data, so it might be too early to tell, but it certainly seems as though the a3 levels decrease from spring to winter.

Violin plots are cousins to the boxplots. Can we get a bit more insight on the a3 trend?

```
ggplot(algae_blooms,aes(x=season,y=a3)) +
  geom_violin() +
  geom_jitter() +
  xlab("Season") +
  ylab("Algae Level a3")
```



This plot certainly seems to suggest that a3 levels are cyclic, with a peak in the spring and low levels in the fall.

Let us return to NH4 for a second to see if we can spot a link with the season (as we did for a3). We only keep the observations for which the NH4 value is greater than 3000, and we bin them with respect to the **quartiles**.

30: Remember that `library(dplyr)` has been called on the previous page.

First, filter the `algae_blooms` dataset to remove the 2 observations with missing values.³⁰

```
f.NH4.data <- filter(algae_blooms,!is.na(NH4))
nrow(f.NH4.data)
```

```
[1] 338
```

Next we remove the 11 observations for which `NH4 > 3000` (again, based on the mean + sd “argument”)

```
f.NH4.data <- filter(algae_blooms,!is.na(NH4)) |>
  filter(NH4<3000)
nrow(f.NH4.data)
```

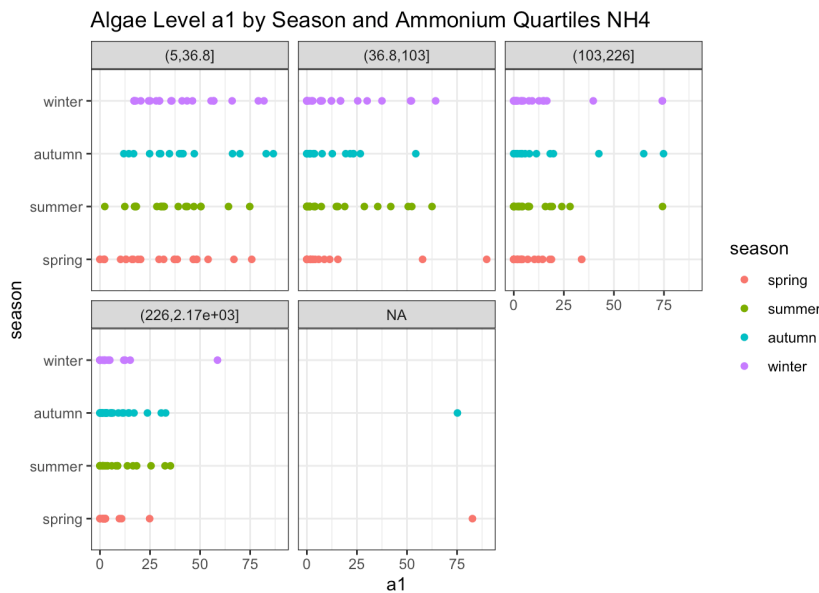
```
[1] 327
```

We create a variable indicating in which quartile the NH4 value falls.

```
f.NH4.data <- filter(algae_blooms,!is.na(NH4)) |>
  filter(NH4<3000) |>
  mutate(q.NH4=cut(NH4,
    quantile(NH4,c(0,0.25,0.5,0.75,1))))
```

We can now use the new variable `q.NH4` to make multi-variate comparisons, say between `a1`, `NH4`, and `season`.


```
ggplot(f.NH4.data,aes(x=a1,y=season,color=season)) +
  geom_point() +
  facet_wrap(~q.NH4) +
  ggtitle("Algae Level a1 by Season and
    Ammonium Quartiles NH4")
```



That seems decidedly odd ... why are we seeing missing values here?
Have we not just removed the NAs? Let us delve in a bit deeper.

```
f.NH4.data[which(is.na(f.NH4.data$q.NH4)),]
table(f.NH4.data$q.NH4, useNA="ifany")
```

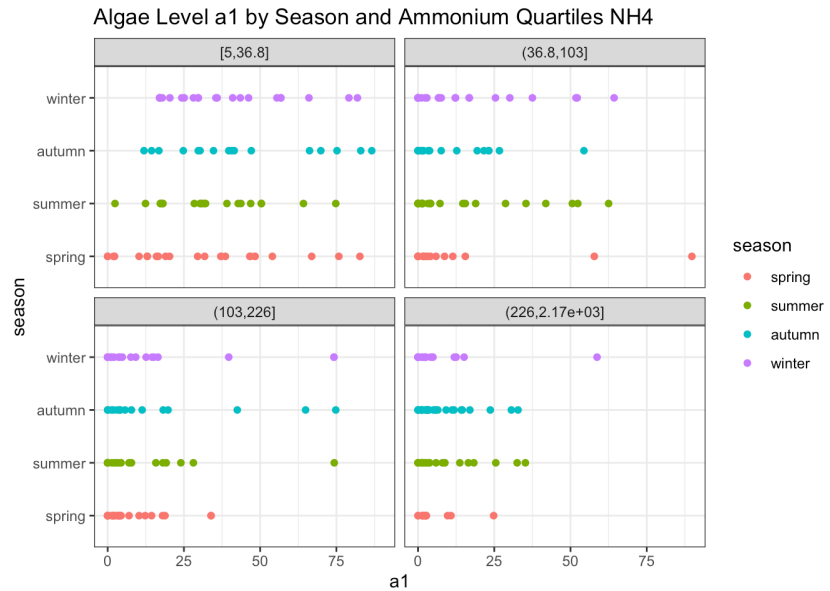
	season	size	speed	mxPH	mnO2	CI	NO3	NH4	oPO4	PO4	Chla	a1	a2	a3	a4	a5	a6	a7	q.NH4
53	spring	small	medium	5.6	11.8	NA	2.22	5	1	1	NA	82.7	0	0	0	0	0	0	NA
223	autumn	small	high	5.9	11.9	NA	1.88	5	1	2	NA	75.2	0	0	0	0	0	0	NA

```
Var1      Freq
(5,36.8]   80
(36.8,103] 82
(103,226]  81
(226,2.17e+03] 82
NA          2
```

The quartiles do not include their lower bound; we can remedy the situation by including an additional parameter in the `mutate()` call.

```
f.NH4.data <- filter(algae_blooms, !is.na(NH4)) |>
  filter(NH4<3000) |> mutate(q.NH4=cut(NH4,
    quantile(NH4,c(0,0.25,0.5,0.75,1)),
    include.lowest=TRUE))
```

```
ggplot(f.NH4.data, aes(x=a1, y=season, color=season)) +
  geom_point() + facet_wrap(~q.NH4) +
  ggtitle("Algae Level a1 by Season and
  Ammonium Quartiles NH4")
```

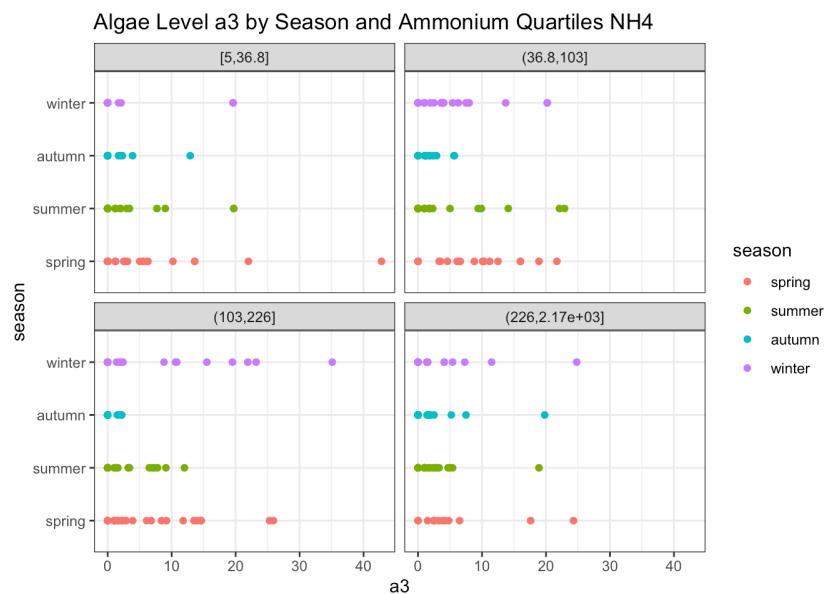


31: Although we would need more work to conclude this with any degree of certainty.

The NAs have disappeared from the graph. In any case, it seems as though the a1 levels are inversely correlated with the NH4 levels but that the season does not have much of an effect.³¹

We can create a similar graph for a3 instead of a1.

```
ggplot(f.NH4.data, aes(x=a3, y=season, color=season)) +
  geom_point() + facet_wrap(~q.NH4) +
  ggtitle("Algae Level a3 by Season and
  Ammonium Quartiles NH4")
```



15.7.4 Data Cleaning

We found some potential anomalies in the data when we explored the dataset,³² now let us take some time to **clean the data** to some extent.

Anomalies come in various flavours; we have already explored some potential **outlying behaviour**, now we handle **missing values**.³³ The function `complete.cases()` lists the observations for which every field is present (note that it says nothing about the **validity** of the case).

32: Although we are electing to keep them in the dataset for the time being as we lack the domain expertise to make a reasonable decision on that front.

33: Again, assume that `library(dplyr)` has already been loaded.

```
table(complete.cases(algae_blooms))
```

```
Var1    Freq
FALSE     34
TRUE     306
```

The vast majority of observations do not have missing cases, but a few still do. Is there anything special about them? Are the values missing completely at random?

```
nrow(filter(algae_blooms, !complete.cases(algae_blooms)))
summary(filter(algae_blooms, !complete.cases(algae_blooms)))
```

```
[1] 34
```

```
season      spring summer autumn winter
           7         6         8         13
```

```
size        small medium large
           26         1         7
```

```
speed        low medium  high
           4         13         17
```

```
      mxPH  mn02  Cl  N03  NH4  oP04  P04  Chla
Min.:  5.6   5.7  0.2  0.2   5.0   1.0   1.0   0.3
Q1  :  6.6   9.2  4.5  0.8  10.0   1.0   6.0   1.7
Med.:  7.2  10.8  9.0  1.4  11.8   3.6  10.8   4.0
Mean:  7.3  10.1 19.3  2.1  62.0  25.6  34.5  13.9
Q3  :  8.0  11.3 25.2  2.5  46.3  20.2  19.2  12.2
Max.:  9.7  12.6 71.0 11.0 500.0 295.6 380.0 68.0
NA's:  2     2  16   2     2     2     7  23
```

```
      a1  a2  a3  a4  a5  a6  a7
Min.:  0.0  0.0  0.0  0.0  0.0  0.0  0.0
Q1  : 16.8  0.0  0.0  0.0  0.0  0.0  0.0
Med.: 30.3  0.0  0.0  0.0  0.0  0.0  0.0
Mean: 36.0  4.5  1.4  2.3  1.5  1.1  1.7
Q3  : 54.4  3.4  1.1  1.9  0.9  0.0  1.6
Max.: 83.0 36.5 14.6 28.8 21.1 14.5 28.0
```

34: By which we mean that low-speed rivers do not seem to have a systematic missing value problem.

35: In a real-life setting, we should *definitely* verify that this assumption is valid.

Right off the bat, missing cases seem to be over-represented in small rivers and under-represented in low-speed rivers. But upon further investigation (that is, comparing with the original dataset), we suspect that the under-representation of low-speed rivers is not problematic as it falls in-line with the numbers in the larger dataset.³⁴

Let us assume for now (in the interest of efficiency) that the fact that small rivers have a lot of missing cases (mostly Cl and Chla) is also not a problem.³⁵ The bulk of the missing values seem to come from either Cl, Chla, or P04. There is also a consistent 2 missing values across the board, but we cannot use the summary output to determine if they arise from the same two observations.

Which observations have missing NH4 values, say?

```
algae_blooms[which(is.na(algae_blooms$NH4)),]
```

	season	size	speed	mxPH	mnO2	Cl	NO3	NH4	oPO4	PO4	Chla	a1	a2	a3	a4	a5	a6	a7
62	summer	small	medium	6.4	NA	NA	NA	NA	NA	14	NA	19.4	0.0	0.0	2	0	3.9	1.7
199	winter	large	medium	8.0	7.6	NA	NA	NA	NA	NA	NA	0.0	12.5	3.7	1	0	0.0	4.9

While these observations also have missing values in other fields, they do have some non-missing fields as well. But they are both missing 6 of the predictor variables. How useful could they be in training a predictive model?³⁶ We can easily write a function that will compute how many missing cases there are for each observations.

```
table(apply(algae_blooms[,1:11],1,
            function(x) sum(is.na(x)))) # 1:rows, 2: columns
which(apply(algae_blooms[,1:11],1,
            function(x) sum(is.na(x))>2))
```

```
Var1 Freq
0      306
1       20
2       12
6         2

[1] 62 199
```

Most observations have no missing cases, which is great news. There are a few with 1 or 2, but observations 62 and 199 are **wild cards**, with 6 missing predictors (out of 11). Based on the small number of such wild cards, we elect to remove them from the analysis.

IMPORTANT NOTES

- If we decide to remove observations for any reason whatsoever, we need to document the process that lead us to eliminate them, and make that process available to other analysts or to the audience.

- Why do we remove the observations with 6 missing cases, but not the ones with 2 missing cases? Had there been observations with 4 missing cases, what should we have done? What factors could influence this decision?

This dataset still contains observations with missing cases, however.

```
algae_blooms.sna = algae_blooms[-which(apply(
  algae_blooms[,1:11],1, function(x) sum(is.na(x))>2),)
nrow(algae_blooms.sna)]
```

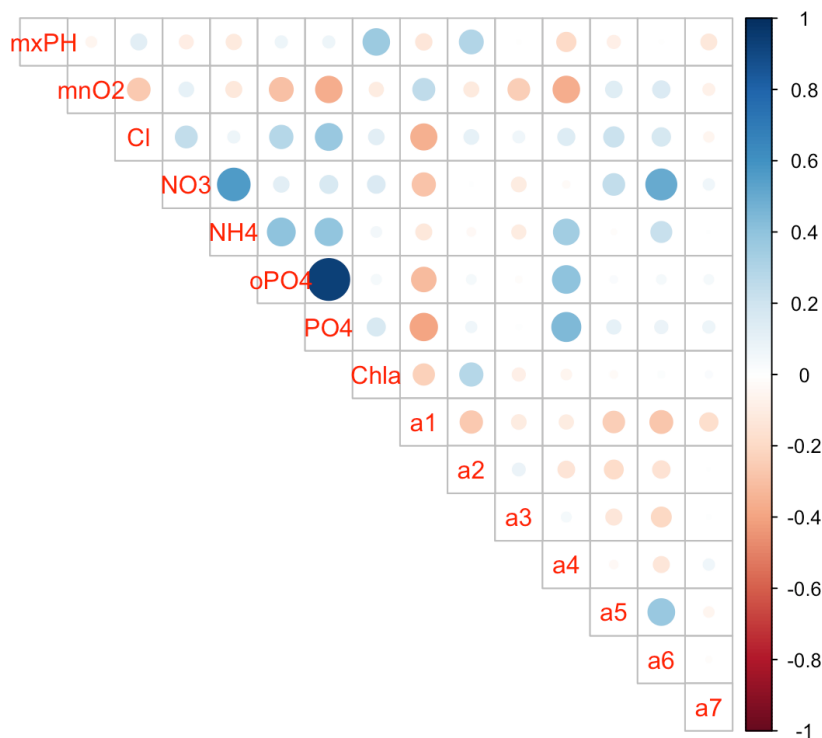
```
[1] 338
```

What can we do with the other observations for which values are missing?

One possibility is to use the set of complete observations to compute a **correlation matrix**, and to see if any numerical field is strongly correlated with another field. That way, if there is a missing value in the first field, the second could be used to impute it.

IMPORTANT NOTE: this approach only works for variables that are linearly correlated to a single other variable. Non-linear correlations and multi-variate associations will not be uncovered.

```
library(corrplot)
corrplot(cor(algae_blooms.sna[,4:18], use="complete.obs"),
  type="upper", tl.pos="d")
```



The correlation between P04 (which has missing cases) and oP04 (which does not, anymore) is clear. What is the nature of the relation? We use the set of complete cases to find it.

```
algae_blooms.nona <- algae_blooms.sna[-which(apply(algae_blooms.sna,1,
  function(x) sum(is.na(x))>0),)]
nrow(algae_blooms.nona)
```

```
[1] 306
```

```
P04.oP04.model = lm(P04 ~ oP04, data=algae_blooms.nona)
P04.oP04.model
```

Coefficients:

(Intercept)	oP04
51.811	1.203

The regression function is $P04 = 51.811 + 1.203 \cdot oP04$ (we are not particularly interested in the fit statistics at this point).

```
Intercept = P04.oP04.model$coefficients[[1]]
Slope = P04.oP04.model$coefficients[[2]]
```

What are the observations for which P04 is missing?

```
which(is.na(algae_blooms.sna$P04)==TRUE)
```

```
[1] 28 221 291 326 331 335
```

We can use the regression function to impute the missing P04 values.

```
algae_blooms.sna2 <- algae_blooms.sna
algae_blooms.sna2$P04 <- ifelse(is.na(algae_blooms.sna2$P04),
  max(Intercept + Slope*algae_blooms.sna2$oP04,0),
  algae_blooms.sna2$P04)
```

We can clearly see that no values of P04 are missing anymore.

```
which(is.na(algae_blooms.sna2$P04)==TRUE)
```

```
integer(0)
```

That takes care of the missing values with strong linear correlation to another field. Where do we stand now?

```
summary(algae_blooms.sna2)
```

We suppress the output in the interest of legibility, but there are still some missing values. And we have exhausted the correlation trick. What else can we do?

There are many ways to tackle the problem, but we will use *k*NN imputation.³⁷ The principle is simple:

1. using some similarity/distance metric (typically based on the Euclidean distance between points), identify the *k* nearest (complete) neighbours of each observation with a missing case;
2. compute the mean value of the missing case in the *k*-group of complete observations, and use *that* value as the imputed value.

37: More details on this topic are available in Chapter 21.

IMPORTANT NOTES

- As we have seen when we were discussing, we often suggest **scaling** the data when dealing with distance metrics. We elected not to scale the data explicitly here. How much of an effect can that have?
- We are going to be using DMwRs implementation of `knnImputation()` (below). How would you go about determining if the routine scales the data internally?

```
algae_blooms.sna2 <- DMwR::knnImputation(algae_blooms.sna2,
                                         k=10)
```

Sure enough, there are no further observations with incomplete cases.

```
table(apply(algae_blooms.sna2,1,
            function(x) sum(is.na(x))))
```

```
0
338
```

15.7.5 Principal Components

Principal components analysis (PCA) is typically used on the (numeric) predictor variables. There are methods that can be used to combine numeric and categorical variables, but for the purposes of this example, we will simply ignore the categorical fields.³⁸

38: We revisit this concept in Chapter 23.

```
pca.algae = algae_blooms.sna2[,4:11]
head(pca.algae)
```

mxPH	mnO2	CI	NO3	NH4	oPO4	PO4	Chla
8.00	9.8	60.800	6.238	578.000	105.000	170.000	50.0
8.35	8.0	57.750	1.288	370.000	428.750	558.750	1.3
8.10	11.4	40.020	5.330	346.667	125.667	187.057	15.6
8.07	4.8	77.364	2.302	98.182	61.182	138.700	1.4
8.06	9.0	55.350	10.416	233.700	58.222	97.580	10.5
8.25	13.1	65.750	9.248	430.000	18.250	56.667	28.4

We can scale the data frame using the `scale()` function in R:

```
head(scale(pca.algae))
```

mxPH	mnO2	CI	NO3	NH4	oPO4	PO4	Chla
-0.0019358	0.2748339	0.4423909	0.9488773	0.0611046	0.2795474	0.0145250	2.1362239
0.6101140	-0.5036258	0.3731037	-0.5578976	-0.0584991	3.1159254	1.4939011	-0.6253276
0.1729355	0.9667981	-0.0296707	0.6724831	-0.0719160	0.4606113	0.0794349	0.1855592
0.1204741	-1.8875541	0.8186771	-0.2492370	-0.2147992	-0.1043426	-0.1045862	-0.6196571
0.1029870	-0.0711482	0.3185826	2.2206563	-0.1368740	-0.1302752	-0.2610671	-0.1036382
0.4352426	1.7020100	0.5548406	1.8651183	-0.0239980	-0.4804704	-0.4167602	0.9113879

Notice the different values in the dataset. The principal components are obtained *via* the `princomp()` function.

```
pca.1 = princomp(scale(pca.algae))
summary(pca.1)
```

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
Standard deviation	1.5334	1.2022	1.0900	0.9329	0.8750	0.8081	0.6549	0.5224
Proportion of Variance	0.2947	0.1812	0.1489	0.1091	0.0959	0.0818	0.0537	0.0342
Cumulative Proportion	0.2947	0.4760	0.6249	0.7341	0.8301	0.9119	0.9657	1.0000

If we can live with 75% of the variability in the numerical component of the predictors being explained by principal components, then we can reduce the dataset dimension by 4.

```
reduced.algae = data.frame(algae_blooms.sna2[,1:3],
                           pca.1$scores[,1:4], algae_blooms.sna2[12:17])
head(reduced.algae, 3)
```

	season	size	speed	Comp.1	Comp.2	Comp.3	Comp.4	a1	a2	a3	a4	a5	a6
1	winter	small	medium	1.0634865	0.2877982	1.6268674	0.26280655	0.0	0.0	0.0	0.0	34.2	8.3
2	spring	small	medium	2.1808329	0.5908937	-2.1258588	1.07795406	1.4	7.6	4.8	1.9	6.7	0.0
3	autumn	small	medium	0.2097857	-0.4229638	0.6216107	0.52245855	3.3	53.6	1.9	0.0	0.0	0.0

Whether this reduction proves useful or not will ultimately depend on what we would like to accomplish with the data; we will study this dataset again with a particular objective in mind in Section 20.6.

15.8 Exercises

1. The ability to monitor and perform early forecasts of various river algae blooms is crucial to control the ecological harm they can cause. The `algae_blooms.csv` dataset consists of: chemical properties of various water samples of European rivers; the quantity of seven (biological) algae in each of the samples, and other characteristics of the collection process for each sample.
 - a) Identify questions that could be tackled with this dataset.
 - b) Determine the structure of the dataset, and provide a summary of its features.
 - c) What can you say about the dataset, in terms of missing values and of the ranges of its values?
 - d) Do 2-way and 3-way tables (for the categorical variables) provide you with additional insights about the dataset?
 - e) Provide some simple (univariate and multivariate) visualizations of `season`, `mn02`, `NH4+`, `a1`, `a3`, and at least one other variable of your choice.
 - f) Does your analysis above suggest that there are anomalies in the dataset? Take action as needed.
 - g) Identify observations (cases) with only 1 missing value, 2 missing values, and so on. Are there strategies that would allow you to handle some of the cases (hint: what is the relationship between `P04` and `oP04`, for instance)? Are there observations that should be removed from the dataset?
 - h) Produce a clean dataset to be used in analysis, with justification.
2. Consider the datasets [GlobalCitiesPBI.csv](#), [2016collisionsfinal.csv](#), [polls_us_election-2016.csv](#), [HR_2016_Census_simple.xlsx](#), and [UniversalBank.csv](#). For each one:
 - a) Create a “data dictionary” to explain the different fields and variables. Can you find a source for these datasets online?
 - b) Develop a list of questions you would like answered about the datasets.
 - c) Investigate individual variables (through simple charts, univariate statistics, etc.).
 - d) Repeat the process with bivariate investigations (though simple charts, joint distributions, variable interactions, etc.).
 - e) Do you trust the dataset, or not? Support your answer. If you do not trust the dataset, flag potential invalid entries, anomalous observations, missing values, or outliers. How should these entries be treated?
 - f) Does any of your analysis suggest that some of the variables should be transformed? Do any of the questions you developed in step 2 support such transformations? If so, transform the data appropriately.
3. Repeat the last question with any dataset of your liking.
4. The remaining exercises use the [Gapminder Tools](#) (there is also an [offline version](#)).
 - a) Explore the dataset with the Gapminder Tools in its default configuration. Do you think that there could be problems with the reported values? For instance, select Sweden and the United States from the checkbox menu on the right and follow their path from 1799 to 2018/2020. From what point onwards are the values sensible? What do you think is happening at the start of the time series?
 - b) Follow Eritrea for the same duration. Look up the country’s independence date from Ethiopia. What do you think the measurements prior to that date represent?
 - c) Follow Austria for the same duration. Look up the historical timeline of the country’s boundaries (Austria-Hungary, Anschluss, modern borders, etc.). What does that imply for the measurements?
 - d) Follow Finland for the same duration. What happens in 1809? Does that tell you anything about the way data is coded in the dataset?
 - e) De-select all countries and let the simulation run from 1799 to 2018/2020. Can you identify instances where a large subset of observations behaves in unexpected manners? If so, do you think that this is due to data cleaning/data processing issues?
 - f) Continue exploring the dataset. You may change which variables are displayed or work with some of the other visualization methods. Overall, do you think that the dataset is sound? Would you use it to run analyses? What are some of its strengths and weaknesses?

Chapter References

- [1] P. Boily. ‘An Imputation Algorithm of Blood Alcohol Content Levels for Drivers and Pedestrians in Fatal Collisions’ . In: *Data Science Report Series* (2007).
- [2] P. Boily, S. Davies, and J. Schellinck. *The Practice of Data Visualization* . Data Action Lab, 2023.
- [3] W. Chang. *R Graphics Cookbook*. O’Reilly, 2013.
- [4] Dheeru Dua and Casey Graff. *Liver Disorders dataset at the UCI Machine Learning Repository* . 2017.
- [5] S. Hagiwara. *Nonresponse Error in Survey Sampling: Comparison of Different Imputation Methods*. Honours Thesis. 2012.
- [6] *Height Percentile Calculator, by Age and Country* .
- [7] *Interactive visualization to teach about the curse of dimensionality* .
- [8] T. Orchard and M. Woodbury. *A Missing Information Principle: Theory and Applications*. University of California Press, 1972.
- [9] T. Raghunathan et al. ‘A Multivariate Technique for Multiply Imputing Missing Values Using a Sequence of Regression Models’. In: *Survey Methodology* 27.1 (2001), pp. 85–95.
- [10] D.B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987.
- [11] L. Torgo. *Data Mining with R, 2nd ed.* CRC Press, 2016.
- [12] S. van Buuren. *Flexible Imputation of Missing Data*. CRC Press, 2012.
- [13] H. Wickham. ‘A Layered Grammar of Graphics’. In: *Journal of Computational and Graphical Statistics* 19 (2009), pp. 3–28.
- [14] H. Wickham, D. Navarro, and T. Lin Pedersen. *ggplot2: Elegant Graphics for Data Analysis*. Springer, 2021.

Web Scraping and Automatic Data Collection

16

by **Patrick Boily**, with contributions from **Andrew Macfie**

Data analysis tools and techniques work in conjunction with collected data. The type of data that needs to be collected to carry out such analyses, as well as the priority placed on the collection of quality data relative to other demands, dictate the choice of data collection strategies. The manner in which the resulting outputs of these analyses are used for decision support will, in turn, influence appropriate data presentation strategies and system functionality.

We have already discussed how data can be processed and transformed to make it more suitable for analysis (see Section 15), and how questionnaire design and probabilistic sampling can be used to obtain representative datasets (see Section 10); in this chapter we explore the technical aspects of automated data collection and web scraping, as well as the many ways in which this activity can go awry.*

16.1 Data Analysis and Web Scraping

Although analysts should always endeavour to work with **representative** and **unbiased data**, there will be times when the available data is flawed and not easily repaired.

We have a professional responsibility to explore the data, looking for potential fatal flaws **prior** to the start of the analysis and to inform clients and stakeholders of any findings that could halt, skew, or simply hinder the analytical process or its applicability to the situation at hand.¹

We might also be called upon to provide suggestions to evaluate or fix the **data collection system**. The following items could help with that.

Data validity: the system must collect the data in such a way that data validity is ensured during initial collection. In particular, data must be collected in a way that ensures sufficient accuracy and precision of the data, relative to its intended use.

Data granularity, scale of data: the system must collect the data at a level of granularity appropriate for future analysis.

Data coverage: the system must collect data that comprehensively, rather than only partially or unevenly, represents the objects of interest; the system must collect and store the required data over a sufficient amount of time, and at the required intervals, to support data analyses that require data spanning a certain duration;

* Some of the material is modified, in part, from [6, 5].

16.1 Data Analysis & Scraping .	1001
Why Web Scraping?	1003
Web Data Quality	1003
Ethical Considerations . .	1005
Decision Process	1007
16.2 Web Technologies Basics .	1007
Content Dissemination . .	1008
HTTP	1009
Web Content	1010
HTML/XML	1011
Cookies and Other Headers	1011
16.3 Scraping Toolbox	1012
Developer Tools	1012
XPath	1013
Regular Expressions . . .	1023
BeautifulSoup	1027
Selenium	1033
APIs	1033
Specialized Uses	1034
16.4 Examples	1034
Wikipedia	1034
Weather Data	1041
CFL Play-by-Play	1049
Bad HTML	1056
Extracting Text from PDF	1057
YouTube Video Titles . . .	1059
16.5 Exercises	1063
Chapter References	1064

1: It is **EXTREMELY IMPORTANT** that these flaws not simply be swept under the carpet; they need to be addressed, and the analysis outcomes that result must be presented or reported on with an appropriate *caveat*.

Data storage: the system must have the functionality to store the types and amount of data required for a particular analysis.

Data accessibility: the system must provide access to the data relevant for a particular analysis, in a format that is appropriate for this analysis.

Computational/analytic functionality: the system must have the ability to carry out the computations required by relevant data analysis techniques.

Reporting, dashboard, visualization: the system must be able to present the results of the data analysis in a meaningful, usable and responsive fashion.

A number of different overarching strategies for data collection can be employed. Each of these different strategies will be more or less appropriate under certain data collection circumstances, and will result in different system **functional requirements**.

This is partly why analysts must take the time to **understand their systems** before embarking on data analysis (see Chapter 14, *Data Science Basics* for details).

World Wide Web

It has been said that the “streets of the Web are paved with data just waiting to be collected” [6], but you might be surprised to discover how much of that data is “trash” [Boily].

The way we **share**, **collect**, and **publish** data has changed over the past few years due to the ubiquity of the *World Wide Web*. **Private businesses**, **governments**, and **individual users** are posting and sharing all kinds of data and information. At every moment, new channels generate vast amounts of data.

There was a time in the recent past where both scarcity and inaccessibility of data was a problem for researchers and decision-makers. That is **emphatically** not the case anymore.

But **data abundance** carries its own set of problems, in the form of:

- tangled masses of data, and
- traditional data collection methods and classical data analysis techniques not being up to the task anymore.²

2: Which is not to say that the results they would give would be incorrect; it's rather their lack of efficiency that comes into play.

The growth and increasing popularity and power of **open source software**, such as R and Python, for which the source code can be inspected, modified, and enhanced by anyone, makes program-based automated data collection quite appealing.

One note of warning, however: time marches on and packages become **obsolete** in the blink of an eye. If the analyst is unable (or unwilling) to **maintain their extraction/analysis code** and to **monitor the sites** from which the data is extracted, the choice of software will not make much of a difference.

16.1.1 The What and Why of Web Scraping

So why bother with **automated data collection**? Common considerations include:

- the sparsity of financial resources;
- the lack of time or desire to collect data manually;
- the desire to work with up-to-date, high-quality data-rich sources, and
- the need to document the analytical process from beginning (data collection) to end (publication).

Manual collection, on the other hand, tends to be cumbersome and prone to error; non-reproducible processes are also subject to heightened risks of “death by boredom”, whereas program-based solutions are typically more reliable, reproducible, time-efficient, and produce datasets of higher quality (this assumes, of course, that coherently presented data exists in the first place).

Automated Data Checklist

That being said, **web scraping** is not always recommended. As a starting point, it is possible that no online and freely available source of data meets the analysis’ needs, in which case an approach based on survey sampling is preferable, in all likelihood.

If most of the answers to the following questions are positive, however, then an automated approach may be the right choice:

- is there a need to repeat the task from time to time?³
- is there a need for others to replicate the data collection process?
- are online sources of data frequently used?
- is the task non-trivial in terms of scope and complexity?
- if the task can be done manually, are the financial resources required to let others do the work lacking?
- is the will to automate the process by means of programming there?

3: E.g., to update a database, say.

The objective is simple: automatic data collection should yield a collection of unstructured or unsorted datasets, at a reasonable cost.

16.1.2 Web Data Quality

Data quality issues are inescapable. It is not rare for stakeholders or clients to have spent thousands of dollars on data collection (automatic or manual) and to respond to the news that the data is flawed or otherwise unusable with: “well, it’s the best data we have, so find a way to use it.”

These issues can be side-stepped to some extent if consultants get involved in the project during or prior to the data collection stage, asking questions such as:

- what **type of data** is best-suited to answer the client’s question(s)?
- is the available data of **sufficiently high quality** to answer the client’s question(s)?
- is the available information **systematically flawed**?

Web data can be **first-hand** information (a tweet or a news article), or **second-hand** (copied from an offline source or scraped from some online location, which may make it difficult to retrace).

Cross-referencing is a standard practice when dealing with secondary data. Data quality also depends on its **use(s)** and **purpose(s)**. For example, a sample of tweets collected on a random day could be used to analyse the use of a hashtags or the gender-specific use of words, but that dataset might not prove as useful if it had been collected on the day of the 2016 U.S. Presidential Election to predict the election outcomes.⁴

4: Due to **collection bias**.

Example Say a client is interested in using a standard telephone survey to find out what people think of a new potato peeler. Such an approach has a number of pitfalls:

- **unrepresentative sample** – the selected sample might not represent the intended population;
- **systematic non-response** – people who do not like phone surveys might be less (or more) likely to dislike the new potato peeler;
- **coverage error** – people without a landline cannot be reached, say, and
- **measurement error** – are the survey questions providing suitable info for the problem at hand?

Traditional solutions to these require the use of **survey sampling**, **questionnaire design**, **omnibus surveys**, reward systems, audits, etc.⁵

5: See Chapter 10 for a discussion of the first two items.

These solutions can be **costly**, **time-consuming**, and **ineffective**. **Proxies** – indicators that are strongly related to the product's popularity without measuring it directly, could be used instead.

If **popularity** is defined as large groups of people preferring a potato peeler over another one, then sales statistics on a commercial website may provide a proxy for popularity. Rankings on '[Amazon.ca](https://www.amazon.ca)' [↗](#) (or a similar website) could, in fact, paint a more comprehensive portrait of the potato peeler market than would a traditional survey.

It could suffice, then, to build a scraper that is compatible with Amazon's **application program interface** (API) to gather the appropriate data. Of course, there are potential issues with this approach as well:

- **representativeness of the listed products** – are all potato peelers listed? If not, is it because that website does not sell them or is there some other reason?
- **representativeness of the customers** – are there specific groups buying/not-buying online products? Are there specific groups buying from specific sites? Are there specific groups leaving/not-leaving reviews?
- **truthfulness of customers and reliability of reviews** – how can we distinguish between paid (fake) reviews and real reviews?

Web scraping is usually well-suited for collecting data on products (such as the aforementioned potato-peeler), but there are numerous questions for which it is substantially more difficult to imagine where data could be found online: what data could be collected online to measure the popularity of a government policy, say?

16.1.3 Ethical Considerations

So is all the data on the Internet ACTUALLY “freely” available?

A **spider** is a program that grazes or crawls the web rapidly, looking for information. It jumps from one page to another, grabbing the entire page content. **Scraping**, on the other hand, is defined as taking specific information from specific websites: how are these different?

“Scraping inherently involves **copying**, and therefore one of the most obvious claims against scrapers is copyright infringement.” [6]

What can be done to minimize the risk? Analysts should:

- work as **transparently** as possible;
- **document** data sources at all time;
- **give credit** to those who originally collected/published the data;
- keep in mind that if someone else collected the data, **permission is probably required** to reproduce it, and, more importantly,
- **not do anything illegal**.

A number of cases have shown that the courts have not yet found their footing in this matter – see *eBay vs. Bidder’s Edge*, *Associated Press vs. Meltwater*, *Facebook vs. Pete Warden*, *United States vs. Aaron Swartz*, for instance [5].

There are legal issues that we are not qualified to discuss, but in general, it seems as though larger companies/organisations usually emerge victorious from such battles.

Part of the difficulty is that it is not clear which scraping actions are illegal and which are legal, but there are rough guidelines: re-publishing content for commercial purposes is considered more problematic than downloading pages for research/analysis, say.

A site’s `robots.txt` (Robots Exclusion Protocol) file tells scrapers what information on the site may be harvested with the publisher’s consent – analysts must heed that file (see Figure 16.1 for examples of such files).



```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used: http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html
User-agent: *
Crawl-delay: 10
# Directories
Disallow: /includes/
Disallow: /misc/
Disallow: /modules/
Disallow: /profiles/
Disallow: /scripts/
Disallow: /themes/
# Files
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
Disallow: /INSTALL.sqlite.txt
Disallow: /install.php
Disallow: /INSTALL.txt
Disallow: /LICENSE.txt
Disallow: /MAINTAINERS.txt
Disallow: /update.php
Disallow: /UPGRADE.txt
Disallow: /xmlrpc.php
User-agent: Twitterbot
Allow: /
User-agent: *
Disallow: /esi/
Disallow: /webview
Disallow: /vuwweb
Disallow: /news/sponsored
Disallow: /search
Disallow: /19849159/
theweathernetwork.com/robots.txt
User-agent: *
Disallow:
Crawl-delay: 10
cfl.ca/robots.txt
```

Figure 16.1: The Robots Exclusion Protocol file for cqads.carleton.ca, theweathernetwork.com, cfl.ca (as of Dec 2022).

Perhaps more importantly, **be friendly!** Not everything that can be scraped needs to be scraped. Scraping programs should

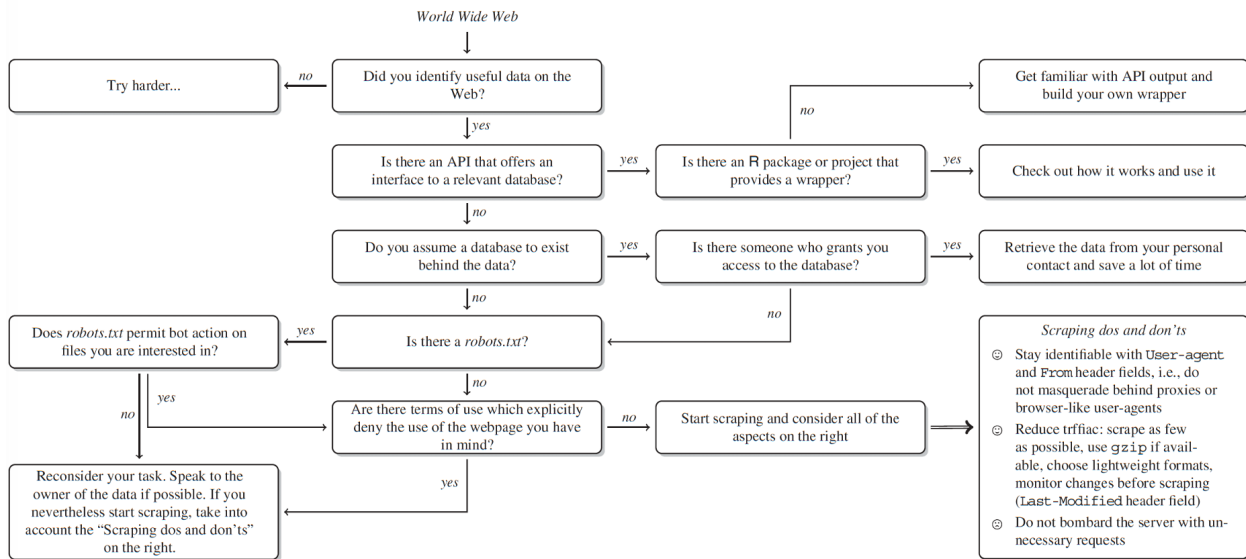


Figure 16.2: Etiquette flow diagram for web scraping. [6]

1. behave “nicely”;
2. provide useful data, and
3. be efficient, in that order.

Any data accessed by HTTP forms is stored in some sort of database. When in doubt, contact the data provider to see if they will grant access to the databases or files.

The larger the amount of data required, the better it is for both parties to communicate before starting to harvest data – for “small” amounts of data, that may be less important, but small *for someone* does not necessarily mean small *for all*.

Finally, note the importance of following the **Scraping Do’s and Don’t’s**:

1. **stay identifiable**;
2. **reduce traffic** – accept compressed files, check that a file has been changed before accessing it again, retrieve only parts of a file;
3. **do not bother server with multiple requests** – many requests per second can bring smaller server downs, webmasters may block a scraper if it is too greedy (a few requests per second is fine), and
4. **write efficient and polite scrapers** – there is no reason to scrape pages daily or to repeat the same task over and over, select specific resources and leave the rest untouched.

6: Really quickly and really often, in fact.

The **design of webpages** tends to change quickly and often.⁶ A broken scraper will still consume bandwidth, however, without payoff; scraper **maintenance** is paramount to successful data collection.

This is all put together in an etiquette flow diagram (or perhaps that should be “ethiquette”?) provided by [6] (see Figure 16.2).

16.1.4 Automated Data Collection Decision Process

Let us end this section by providing a short summary of the **automated data collection decision process** [6, 5], from the point of view of analysts or quantitative consultants:

1. **Know exactly what kind of information the client needs**, either **specific** (e.g. GDP of all OECD countries for last 10 years, sales of top 10 tea brands in 2017, etc.) or **vague** (people's opinion on tea brand X, etc.)
2. **Find out if there are any web data sources that could provide direct or indirect information on the client's problem**. That is easier to achieve for specific facts (a tea store's webpage will provide information about teas that are currently in demand) than it is for vague facts (where would one find opinions on a collection of tea brands?). Tweets and social media platforms may contain opinion trends; commercial platforms can provide information on product satisfaction.
3. **Develop a theory of the data generation process when looking into potential data sources**. When was the data generated? When was it uploaded to the Web? Who uploaded the data? Are there any potential areas that are not covered, consistent, or accurate? How often is the data updated?
4. **Balance the advantages and disadvantages of potential data sources**. Validate the quality of data used – are there other independent sources that provide similar information against which to crosscheck? Can original source of secondary data be identified?
5. **Make a data collection decision**. Choose the data sources that seem most suitable, and document reasons for this decision. Collect data from several sources to validate the final choice.

16.2 Web Technologies Basics

Online data can be found in **text**, **tables**, **lists**, **links**, and other structures, but the way data is presented in browsers is not necessarily how it is stored in HTML/XML.

For instance, consider the NHL's Atlantic Division standings on 20-Mar-2018 below.

Atlantic	GP	W	L	OT	PTS	ROW	GF	GA	DIFF	HOME	AWAY	S/O	L10	STRK	Last Game	Next Game
Tampa Bay	72	49	19	4	102	43	260	202	+58	26-8-2	23-11-2	6-2	7-2-1	W1	Mar 18: TBL 3 - EDM 1	Mar 20 vs TOR
Boston	71	45	17	9	99	42	239	184	+55	25-7-5	20-10-4	3-2	7-2-1	OT1	Mar 19: BOS 4 - CBJ 5	Mar 21 @ STL
Toronto	72	43	22	7	93	36	243	204	+39	25-8-2	18-14-5	7-2	6-2-2	W4	Mar 17: TOR 4 - MTL 0	Mar 20 @ TBL
Florida	70	36	27	7	79	33	212	216	-4	22-11-3	14-16-4	3-3	7-2-1	W1	Mar 19: FLA 2 - MTL 0	Mar 20 @ OTT
Montréal	73	26	35	12	64	24	182	232	-50	17-12-8	9-23-4	2-6	2-6-2	L3	Mar 19: MTL 0 - FLA 2	Mar 21 @ PIT
Ottawa	71	26	34	11	63	24	197	244	-47	15-14-6	11-20-5	2-7	5-4-1	L1	Mar 17: OTT 1 - CBJ 2	Mar 20 vs FLA
Detroit	72	26	35	11	63	22	184	224	-40	13-14-8	13-21-3	4-1	0-9-1	L6	Mar 18: DET 1 - COL 5	Mar 20 vs PHI
Buffalo	72	23	37	12	58	22	172	236	-64	11-21-5	12-16-7	1-2	5-4-1	L1	Mar 19: BUF 0 - NSH 4	Mar 21 vs ARI

Figure 16.3: NHL's Atlantic Division standings on 20-Mar-2018 [nhl.com ↗]

This table is human-readable: most people familiar with professional competitions can recognize what it “means”, even if they know very little about hockey or the National Hockey League.

In a **web browser**, this is not how the information is found, however (see Figure 16.4).

```

<th class="no-sort col-0 th--fixed" data-index="0">
  <span>
    <span class="col--title--name">Atlantic</span> = $0
    <span class="col--title--name__abbrev">ATL</span>
  </span>
</th>
<th class="sortable col-1 th--fixed" data-index="1"></th>
<th class="sortable col-2 th--fixed" data-index="2"></th>
<th class="sortable col-3 th--fixed" data-index="3"></th>
<th class="sortable col-4 th--fixed" data-index="4"></th>
<th class="sortable col-5 th--fixed desc th--selected" data-index="5"></th>
<th class="sortable col-6 th--fixed" data-index="6"></th>
<th class="sortable col-7 th--fixed" data-index="7"></th>
<th class="sortable col-8 th--fixed" data-index="8"></th>
<th class="sortable col-9 th--fixed" data-index="9"></th>
<th class="no-sort col-10 th--fixed" data-index="10"></th>
<th class="no-sort col-11 th--fixed" data-index="11"></th>
<th class="no-sort col-12 th--fixed" data-index="12"></th>
<th class="no-sort col-13 th--fixed" data-index="13"></th>
<th class="sortable col-14 th--fixed" data-index="14"></th>
<th class="no-sort col-15 th--fixed" data-index="15"></th>
<th class="no-sort col-16 th--fixed" data-index="16"></th>
</tr>
</thead>
<tbody>
<tr data-index="0" class>
  <td class="col-0 row-0 td--fixed" data-col="0" data-row="0">
    <span>
      <a href="/lightning">
        <svg class="logo logo--light-theme team--logo"></svg>
        <span class="team--name">x-Tampa Bay</span>
        <span class="team--name__abbrev">x-TBL</span>
      </a>
    </span>
  </td>
  <td class="col-1 row-0 td--fixed" data-col="1" data-row="0">
    <span>72</span>
  </td>
  <td class="col-2 row-0 td--fixed" data-col="2" data-row="0">
    <span>49</span>
  </td>
  <td class="col-3 row-0 td--fixed" data-col="3" data-row="0">
    <span>19</span>
  </td>
  <td class="col-4 row-0 td--fixed" data-col="4" data-row="0">
    <span>4</span>
  </td>
  <td class="col-5 row-0 td--fixed td--highlighted" data-col="5" data-row="0">
    <span>102</span>
  </td>
  <td class="col-6 row-0 td--fixed" data-col="6" data-row="0">
    <span>43</span>
  </td>
  <td class="col-7 row-0 td--fixed" data-col="7" data-row="0">
    <span>260</span>
  </td>
  <td class="col-8 row-0 td--fixed" data-col="8" data-row="0"></td>

```

Figure 16.4: NHL’s Atlantic Division standings on 20-Mar-2018 (under the hood) [nhl.com ↗]

Furthermore, when web pages are **dynamic**, there is a “cost” associated with automated collection. Consequently, a basic knowledge of the web and web-related techs and documents is crucial. [Information can readily be found online and in [5, 6].]

There are three areas of importance for data collection on the web:

- technologies for **content dissemination** (HTTP, HTML/XML, JSON, plain text, etc.);
- technologies for **information extraction** (R, Python, XPath, JSON parsers, BeautifulSoup, Selenium, regexps, etc.), and
- technologies for **data storage** (R, Python, SQL, binary formats, plain text formats, etc.).

16.2.1 Content Dissemination

The information that web scrapers look for on webpages appears in one of the following formats:

HTML – Hypertext Markup Language is used to display information on the web; it is not a dedicated data storage format, but it typically contains the information of interest; HTML is interpreted and transformed into “pretty” output by browsers (using CSS);

HTML: predefined tags	XML: self-defined tags	JSON	XML
HTML 1 <pre><h1>title</h1> <p>paragraph</p> <p>paragraph</p></pre>	XML 1 <pre><headline>title</headline> <paragraph>paragraph</paragraph> <paragraph>paragraph</paragraph></pre>	<pre>{ "person": { "xmlns": "urn:ns:person", "firstName": { "\$t": "John" }, "lastName": { "\$t": "Smith" }, "contactInfo": { "default": "true", "type": "home", "xmlns": "urn:ns:contactinfo", "phone": [{ "type": "voice", "\$t": "203-555-1212" }, { "type": "fax", "\$t": "203-555-1213" }], "email": { "xmlns": "", "\$t": "jsmith@example.com" } }, "photo": { "xmlns": "", "\$t": "http://example.com/jsmith/profile.png" } } }</pre>	<pre><person xmlns="urn:ns:person"> <firstName>John</firstName> <lastName>Smith</lastName> <contactInfo xmlns="urn:ns:contactinfo" type="home" default="true" > <phone type="voice">203-555-1212</phone> <phone type="fax">203-555-1213</phone> <email xmlns="">jsmith@example.com</email> </contactInfo> <photo xmlns="">http://example.com /jsmith/profile.png</photo> </person></pre>
HTML 2 <pre><h1>title</h1> <p>paragraph</p> <p>paragraph</p></pre>	XML 2 <pre><chief>title</chief> <worker>paragraph</worker> <worker>paragraph</worker></pre>		

Figure 16.5: Comparison between HTML and XML (left, e-cartouches.ch), and between JSON and XML (right, activevos.com).

XML – Extensible Markup Language is a popular format for exchanging data over the web; its main purpose is to store data; XML is data wrapped in user-defined tags and as such is more flexible for storing data than HTML is;

JSON – JavaScript Object Notation is another data storage and exchange format; it is compatible with many programming languages and software; it is easier to parse than HTML or XML, and there is no need to use a specific query language (high level R is usually sufficient);

AJAX – Asynchronous JavaScript and XML is a group of technologies that enables websites to request data in the background of the browser session and update its visual appearance in a dynamic fashion, while allowing navigation to proceed when waiting for server reply (this can be a nuisance for web scrapers).

16.2.2 Hyper Text Transfer Protocol

Hypertext Transfer Protocol (HTTP) is a message language used between web browsers and web servers; Hypertext Transfer Protocol **Secure (HTTPS)** combines HTTP with SSL (encryption) and TLS (authentication) protocols.

In a nutshell, when we type in a URL in a browser to access a web page, the browser sends an HTTP **request** to the underlying **server**.

A request is made up of a verb, a path, a list of headers, and possibly some parameters. Common **verbs** include: GET (click on a link) and PUT (fill-out a form and submit).

For instance, if we type `http://www.yahoo.com/search` into the browser, a GET request is sent by the browser to the `yahoo.com` server, together with the path `/search`.

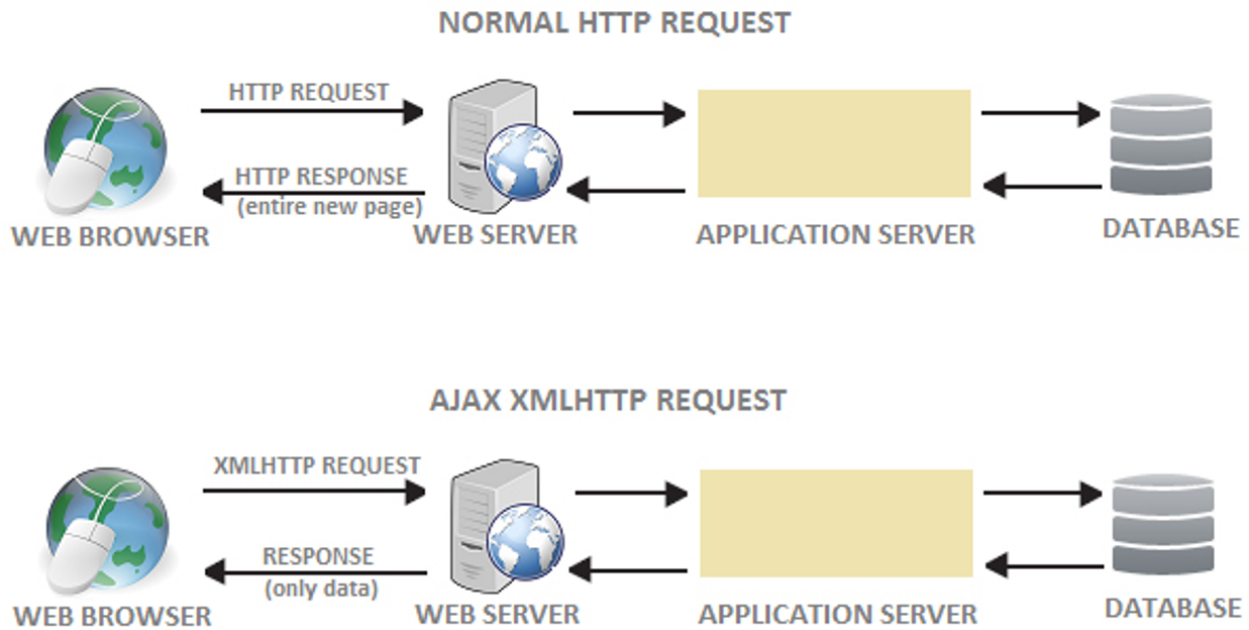


Figure 16.6: Schematics of HTTP (top) and AJAX (bottom) requests; a new HTTP request refreshes the entire page, a new AJAX request only refreshes the data javalazy.blogspot.ca.

The web server then sends a **response** to the browser, containing a code (404, 200, etc.), as well as headers and content.

The 200 response, say, means that the request was successful: the browser reads the content and uses it (and CSS files) to display the page.

16.2.3 Web Content

Webpage content itself comes into three main types:

- **Hypertext Markup Language** and variants (HTML/XML) is used for web content and code;
- **Cascading Style Sheets** (CSS) is used to define the webpage style, and
- **JavaScript** (JS) is used to provide webpage interactivity.

HTML is, in some sense, the most **fundamental** (the other two are optional); HTML is a **document language**, like \LaTeX or markdown (on which this book is based). A fresh HTTP/HTTPS request for a page usually returns an HTML file, which may contain references to additional server files (CSS, JavaScript, images, etc.) – the browser makes additional requests for these when the webpage is rendered.

Understanding the **tree structure** of HTML documents goes a long way towards helping analysts make full use of the **scraping toolbox** (see Section 16.3).

CSS defines the colour schemes, the fonts, spacing, and so on. It operates basically as a PowerPoint template would. In the absence of a CSS file, the browser uses a default style to render the webpage.

JS, on the other hand, is a programming language. After the browser parses and displays the HTML file, it executes any JS files referenced

in the HTML. JS can be used to manipulate most things on the page (delete/add/change content, change CSS, fetch more files from server, go to new page, etc.), and it can set up actions that run as a result of page events (clicking a button, typing in a text box, etc.)

16.2.4 HTML/XML

HTML syntax is fairly straightforward. HTML is a document language based on **tags**. Tags either come in **pairs**:

- `<title>...</title>` (self-explanatory),
- `...` (bold face text), etc.,

or as stand alone **singletons**:

- `
` (linebreak),
- `<hr>` (horizontal rule), etc.

Paired tags are **nested**:

- `...` is acceptable, whereas
- `...` is not.

An HTML file is a **tree of tags**, also known as **elements** or **nodes**.

Tags consist of a **name/type** (mandatory) and **attributes** (optional): the tag `<p lang="en">...</p>`, for instance, is of type `p` (paragraph), and it has a single attribute: `lang="en"`.

Plain text is allowed inside tags: `Hello World!`.

Beyond this, the only other thing left to learn is the set of possible tags, and the set of possible attributes. The list is extensive; information can be found at [2].

Two attributes are particularly important for web scraping:

- `id` uniquely identifies an element: `...`, `<p id="saleInfo">...</p>`, etc.;
- `class` can contain multiple values, separated by spaces and is not unique, but it identifies a set of elements: `<h1 class="lightBackground oddPage">...</h1>`, etc.

16.2.5 Cookies and Other Headers

We discuss briefly three common headers:

- a **cookie** is a string that is sent and received with HTTP/HTTPS; it allows servers to keep track of user sessions. Upon logging on to a website, users receive a cookie. If the cookie is included in future requests, the user (and its preferences and choices) is recognized by the server; otherwise, the website acts as though the user has logged out.
- **user agent** contain the name and the version of the user's browser.
- **referrer** sends the page URL from which the request was initiated; if the user is on Page A and clicks a link to Page B, the server for Page B will see that the user came from Page A.

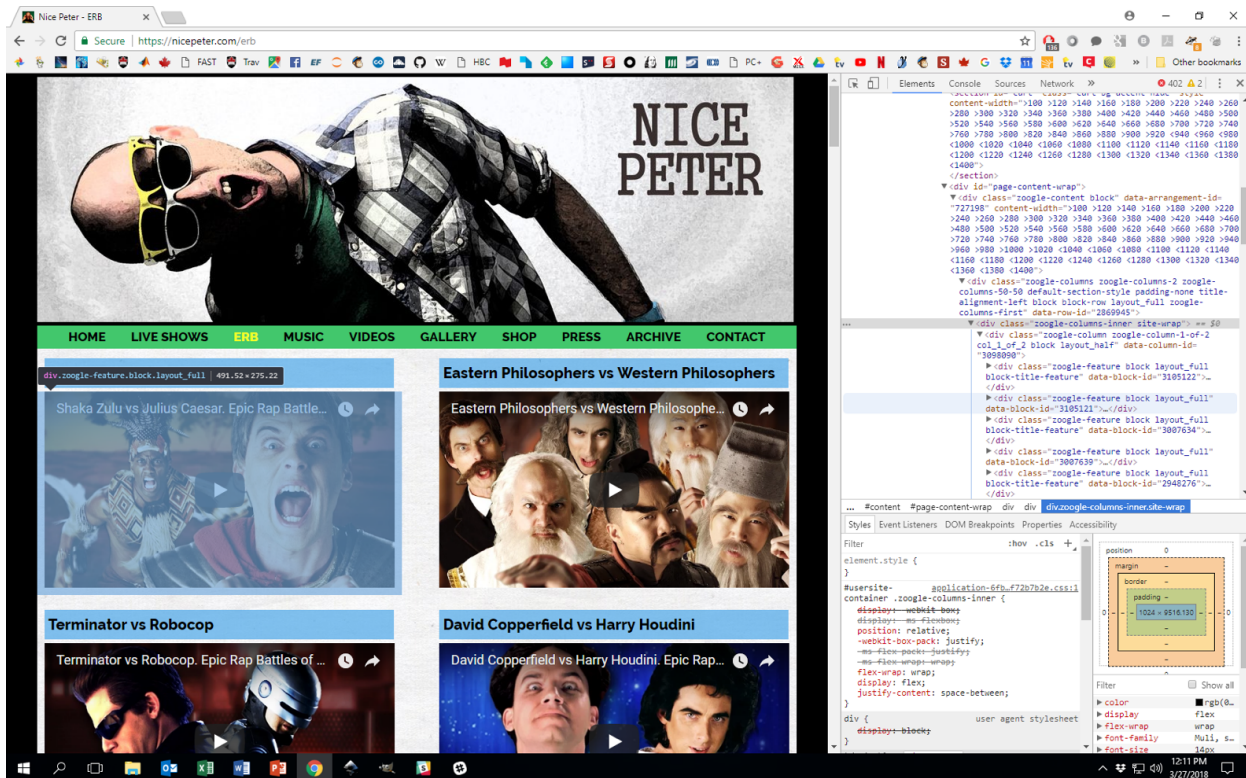


Figure 16.7: Inspecting Nice Peter’s website’s elements using Developer Tools in Chrome.

16.3 Scraping Toolbox

From experience, we know that a number of tools can facilitate the automated data extraction process, including:

- Developer Tools,
- XPath,
- regular expressions,
- BeautifulSoup, and
- Selenium.

We will briefly introduce each of them in this section.

16.3.1 Developer Tools

Developer Tools allow us to see the correspondence between the HTML code for a page and the rendered version seen in the browser, as illustrated in Figure 16.7.

Unlike “View Source”, Developer Tools show the *dynamic* version of the HTML content.⁷ Inspecting a page’s various elements and discovering where they reside in the HTML file is **crucial** to efficient web scraping:

- **Firefox** – right click page → Inspect Element
- **Safari** – Safari → Preferences → Advanced → Show Develop Menu in Menu Bar, then Develop → Show Web Inspector
- **Chrome** – right click page → Inspect

7: That is, the HTML is shown with any changes made by JavaScript since the page was first received.

16.3.2 XPath

XPath is a query (domain-specific) language which is used to select specific pieces of information from marked-up documents.⁸ Before this can be done, the information stored in a marked-up document needs to be converted (or **parsed**) into a format suitable for processing and statistical analysis; this is implemented in the R package XML, for instance.

The process is simple; it involves

1. **specifying** the data of interest;
2. **locating it** in a specific document, and
3. tailoring a query to the document to **extract** the desired info.

HTML/XML tags have **attributes** and **values**. HTML files must be parsed before they can be queried by XPath. XPath queries require both a **path** and a **document** to search; paths consist of hierarchical addressing mechanism (succession of nodes, separated by forward slashes ("/")), while a query takes the form `xpathSApply(doc, path)`.⁹

We will illustrate Xpath's functionality with the following webpage:

Laws of the *Internet*

Osmo Antero Wiio

Communication usually fails, except by accident.

Source: Wiio lait - ja vähän muidenkin

Melvin Kranzberg

Technology is neither good nor bad; nor is it neutral.
(Kranzberg's 1st Law)

Source: [Technology and Culture](#), 27 (3): 544-560.

Theodore Sturgeon

90% of everything is crap.
(Sturgeon's Revelation)

Source: "Books: On Hand". Venture Science Fiction. Vol. 2, no. 2, p. 66.

Others:

- ☐ The 1% Rule: "Only 1% of the users of a website actively create new content, while the other 99% of the participants only lurk."
- ☐ D!@kwad Theory: "Normal Person + Anonymity + Audience = Total D!@kwad"
- ☐ Godwin's Law: "As an online discussion grows longer, the probability of a comparison involving Nazis or Hitler approaches one."
- ☐ Poe's Law: "Without a clear indicator of the author's intent, parodies of extreme views will be mistaken by some readers or viewers as sincere expressions of the parodied views."
- ☐ Skitt's Law: "Any post correcting an error in another post will contain at least one error itself."
- ☐ Law of Exclamation: "The more exclamation points used in an email (or other posting), the more likely it is a complete lie."
- ☐ Cunningham's Law: "The best way to get the right answer on the Internet is not to ask a question, it's to post the wrong answer."
- ☐ The Wiki Rule: "There's a wiki for that."
- ☐ Danth's Law: "If you have to insist that you've won an Internet argument, you've probably lost badly."
- ☐ Law of the Echo Chamber: "If you feel comfortable enough to post an opinion of any importance on any given Internet site, you are most likely delivering that opinion to people who already agree with you."
- ☐ Munroe's Law: "You will never change anyone's opinion on anything by making a post on the Internet. This will not stop you from trying."

¹⁵ *Fundamental Laws of the Internet*, by Matthew Jones

The underlying HTML code is in the file `laws.html`; we parse the document using XML's `htmlParse()`.

```
parsed_doc <- XML::htmlParse(file = "Data/laws.html")
print(parsed_doc)
```

8: Such as HTML, XML, or variants such as SVG, RSS.

9: `xpathSApply(parsed_doc, "/html/body/div/p/i")`, for instance, would find all `<i>` tags under a `<p>` tag, itself under a `<div>` tag in the body of the html file of `parsed_doc`. A substantially heftier treatment can be found in [6].

Figure 16.8: A simple HTML document, rendered in a browser, based on [4].

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head><title>Laws of the Internet</title></head>
<body>
<h1>Laws of the <i>Internet</i>
</h1>
<div id="wiio" lang="english" date="1978">
  <h2>Osmo Antero Wiio</h2>
  <p><i>Communication usually fails, except by accident.</i></p>
  <p><b>Source: </b>Wiion lait - ja vähän muidenkin</p>
</div>

<div lang="english" date="1986">
  <h2>Melvin Kranzberg</h2>
  <p><i>Technology is neither good nor bad; nor is it neutral.</i> <br><emph>(Kranzberg's 1st Law)</emph></p>
  <p><b>Source: </b><a href="https://www.jstor.org/stable/3105385">Technology and Culture. 27 (3): 544-560.</a></p>
</div>

<div lang="english" date="1958">
  <h2>Theodore Sturgeon</h2>
  <p><i>90% of everything is crap.</i> <br><emph>(Sturgeon's Revelation)</emph></p>
  <p><b>Source: </b>"Books: On Hand". Venture Science Fiction. Vol. 2, no. 2. p. 66.</p>
</div>

<div id="other">
<h2>Others:</h2>
<ul>
<li>The 1% Rule: "Only 1% of the users of a website actively create new content, while the other 99% of the participants only lurk."</li>
<li>D!@kwad Theory: "Normal Person + Anonymity + Audience = Total D!@kwad"</li>
<li>Godwin's Law: "As an online discussion grows longer, the probability of a comparison involving Nazis or Hitler approaches one."</li>
<li>Poe's Law: "Without a clear indicator of the author's intent, parodies of extreme views will be mistaken by some readers or viewers as sincere expressions of the parodied views."</li>
<li>Skitt's Law: "Any post correcting an error in another post will contain at least one error itself."</li>
<li>Law of Exclamation: "The more exclamation points used in an email (or other posting), the more likely it is a complete lie."</li>
<li>Cunningham's Law: "The best way to get the right answer on the Internet is not to ask a question, it's to post the wrong answer."</li>
<li>The Wiki Rule: "There's a wiki for that."</li>
<li>Danth's Law: "If you have to insist that you've won an Internet argument, you've probably lost badly."</li>
<li>Law of the Echo Chamber: "If you feel comfortable enough to post an opinion of any importance on any given Internet site, you are most likely delivering that opinion to people who already agree with you."</li>
<li>Munroe's Law: "You will never change anyone's opinion on anything by making a post on the Internet. This will not stop you from trying."</li>
</ul>
</div>

<address>
<a href="https://exceptionnotfound.net/15-fundamental-laws-of-the-internet/"><i>15 Fundamental Laws of the Internet</i></a>, by Matthew Jones<a></a>
</address>

</body>
</html>

```

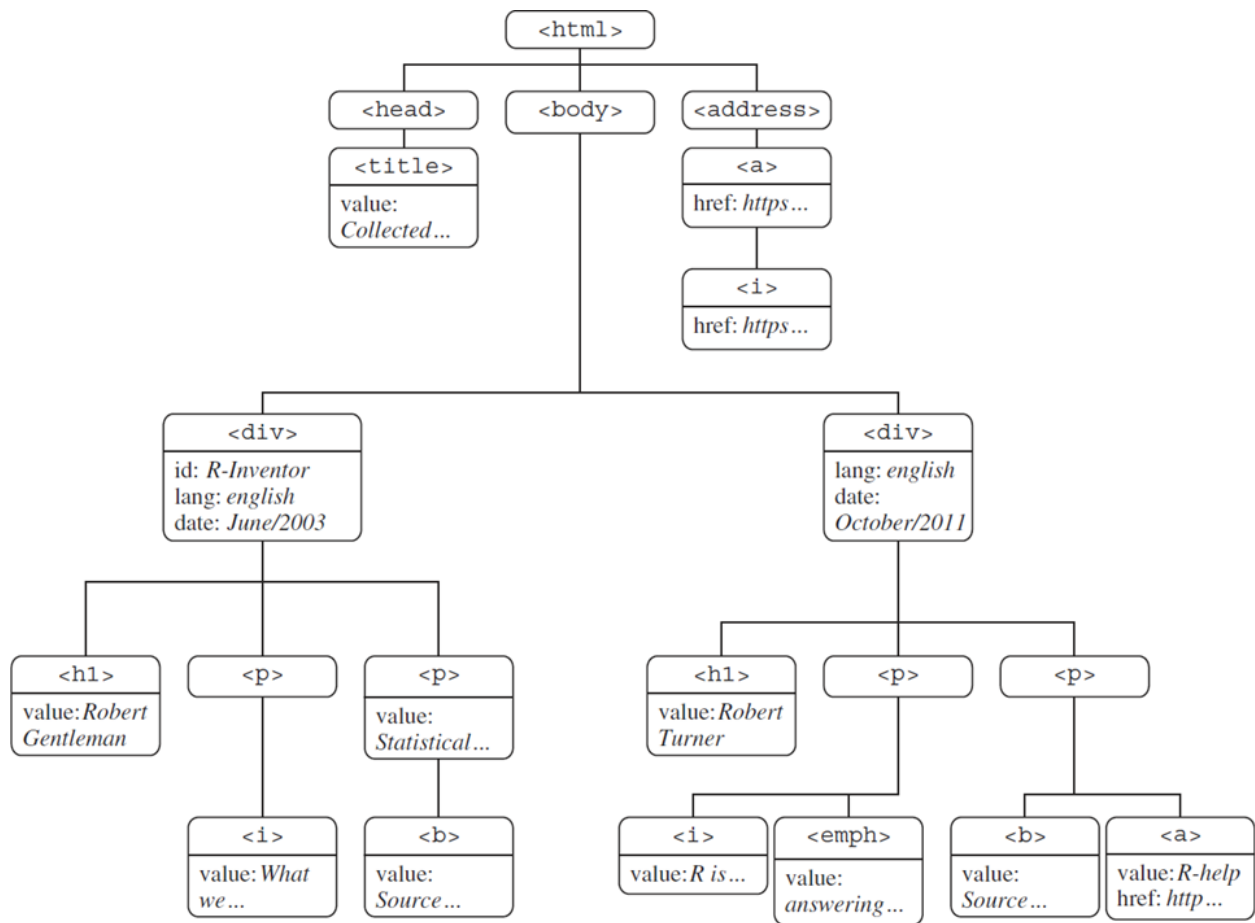



Figure 16.9: The HTML document tree for the R built-in fortunes.html file [6].

Basic Structural Queries

XPath queries are called using `xpathApply()`, which requires a parsed document `doc` and a query path `path`.

It is much easier to determine the required query paths if we have some idea of the structure of the underlying **HTML document tree**.¹⁰

10: See Figure 16.9 for an example.

Absolute paths are represented by single forward slashes [/]; **relative paths** by double forward slashes [//]. The next three calls will all return the same output.

```
XML::xpathApply(doc = parsed_doc, path = "/html/body/div/p/i")
XML::xpathApply(parsed_doc, "//body//p/i")
XML::xpathApply(parsed_doc, "//p/i")
```

```
[[1]]
```

```
<i>Communication usually fails, except by accident.</i>
```

```
[[2]]
```

```
<i>Technology is neither good nor bad; nor is it neutral.</i>
```

```
[[3]]
```

```
<i>90% of everything is crap.</i>
```

Wildcards are represented by an asterisk [*]: the code below once again has the same output as the one above.

```
XML::xpathApply(parsed_doc, "/html/body/div/*/i")
```

Going up one level in the parsed tree is represented by a double dot [..].

```
XML::xpathApply(parsed_doc, "//title/..")
```

```
[[1]]
<head>
  <title>Laws of the Internet</title>
</head>
```

The **disjunction** (OR) of two paths is represented by the operator [|].

```
XML::xpathApply(parsed_doc, "//address | //title")
```

```
[[1]]
<title>Laws of the Internet</title>

[[2]]
<address>
<a href="https://exceptionnotfound.net/15-fundamental-laws-of-the-internet/">
  <i>15 Fundamental Laws of the Internet</i></a>, by Matthew Jones<a/>
</address>
```

We can also **concatenate** multiple queries (which, in this case, would produce the same output as the immediate call above).

```
twoQueries <- c(address = "//address", title = "//title")
XML::xpathApply(parsed_doc, twoQueries)
```

Note, however, that absolute (or even relative) paths cannot always succinctly select nodes in large or complicated files.

Node Relations

A query's path can also exploit a node's relation to other nodes. By analogy with a **family tree**, a node's placement in the parsed tree often mimics the relations in extended families.

Relations are denoted according to `node1/relation::node2`. For instance:

- `"//a/ancestor::div"` returns all `<div>` nodes that are an ancestor to an `<a>` node;
- `"//a/ancestor::div//i"` returns all `<i>` nodes contained in a `<div>` node that is an ancestor to an `<a>` node, etc.¹¹

11: See Figure 16.10 for a complete list of node relations.

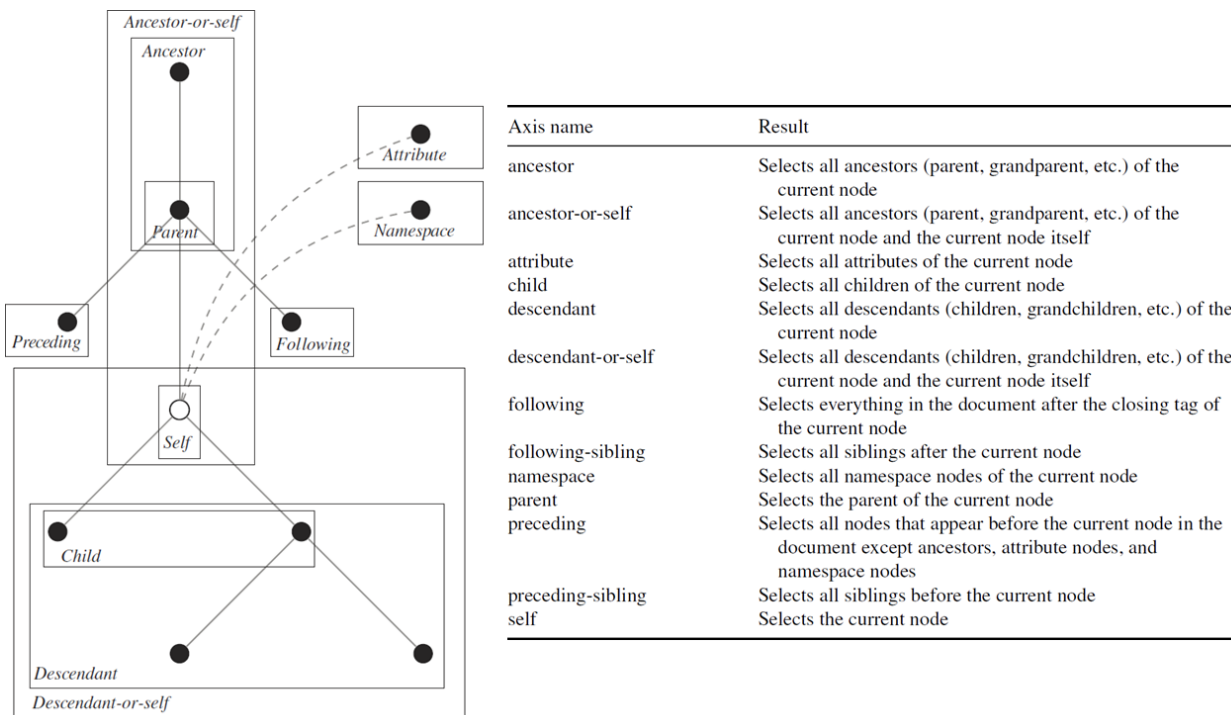


Figure 16.10: Generic node relations [6].

The following XPath query looks for <a> tags in the document, and produces their **ancestors** <div> tag.¹²

```
XML::xpathSApply(parsed_doc, "//a/ancestor::div")
```

12: There is only one of each in this example, but that is an accident of the file with which we are working; there could be more in general.

```
[[1]]
<div lang="english" date="1986">
  <h2>Melvin Kranzberg</h2>
  <p><i>Technology is neither good nor bad; nor is it neutral.</i>
    <br/><emph>(Kranzberg's 1st Law)</emph></p>
  <p><b>Source: </b><a href="https://www.jstor.org/stable/3105385">Technology and Culture.
    27 (3): 544-560.</a></p>
</div>
```

The following XPath query looks for <a> tags in the document, and produces all <i> tags of their ancestors <div> tag (there is only one in this example).

```
XML::xpathSApply(parsed_doc, "//a/ancestor::div//i")
```

```
[[1]]
<i>Technology is neither good nor bad; nor is it neutral.</i>
```

The following XPath query looks for <p> tags in the document, and produces the <h2> tags of all their preceding-sibling nodes (there are three in this example).

```
XML::xpathSApply(parsed_doc, "//p/preceding-sibling::h2")
```

```
[[1]]
<h2>Osmo Antero Wiio</h2>
```

```
[[2]]
<h2>Melvin Kranzberg</h2>
```

```
[[3]]
<h2>Theodore Sturgeon</h2>
```

What do you think this query will do?

```
XML::xpathSApply(parsed_doc, "//title/parent::*")
```

XPath Predicates

A **predicate** is a function that applies to a node's **name**, **value**, or **attributes** and that returns a logical TRUE or FALSE.

Predicates modify the path input of an XPath query: the query selects the nodes for which the relation holds.

Predicates are denoted by square brackets, placed after a node.

For instance:

- `"//p[position()=1]"` returns the first `<p>` node relative to its parent node;
- `"//p[last()]"` returns the last `<p>` node relative to its parent node, and
- `"//div[count(./@*)>2]"` returns all `<div>` nodes with 2+ attributes.

This XPath query finds the first `<p>` node in each `<div>` node.

```
XML::xpathSApply(parsed_doc, "//div/p[position()=1]")
```

```
[[1]]
<p>
  <i>Communication usually fails, except by accident.</i>
</p>
```

```
[[2]]
<p><i>Technology is neither good nor bad; nor is it neutral.</i>
<br/><emph>(Kranzberg's 1st Law)</emph></p>
```

```
[[3]]
<p><i>90% of everything is crap.</i>
<br/><emph>(Sturgeon's Revelation)</emph></p>
```

This XPath query finds the last <p> node in each <div> node.

```
XML::xpathSApply(parsed_doc, "//div/p[last()]")
```

```
[[1]]
<p><b>Source: </b>Wiion lait - ja vähän muidenkin</p>

[[2]]
<p>
  <b>Source: </b>
  <a href="https://www.jstor.org/stable/3105385">Technology and Culture. 27 (3): 544-560.</a>
</p>

[[3]]
<p><b>Source: </b>"Books: On Hand". Venture Science Fiction. Vol. 2, no. 2. p. 66.</p>
```

This next XPath query finds the second last <p> node in each <div> node.

```
XML::xpathSApply(parsed_doc, "//div/p[last()-1]")
```

```
[[1]]
<p>
  <i>Communication usually fails, except by accident.</i>
</p>

[[2]]
<p><i>Technology is neither good nor bad; nor is it neutral.</i>
<br/><emph>(Kranzberg's 1st Law)</emph></p>

[[3]]
<p><i>90% of everything is crap.</i> <br/><emph>(Sturgeon's Revelation)</emph></p>
```

This XPath query finds the <div> nodes that have at least one <a> node among their children.

```
XML::xpathSApply(parsed_doc, "//div[count(./a)>0]")
```

```
[[1]]
<div lang="english" date="1986">
  <h2>Melvin Kranzberg</h2>
  <p><i>Technology is neither good nor bad; nor is it neutral.</i>
  <br/><emph>(Kranzberg's 1st Law)</emph></p>
  <p><b>Source: </b><a href="https://www.jstor.org/stable/3105385">Technology and Culture.
  27 (3): 544-560.</a></p>
</div>
```

A number of commonly-used XPath functions are shown in Table 16.1.

For instance, the following XPath query finds the <div> nodes that have more than 2 attributes.

```
XML::xpathSApply(parsed_doc, "//div[count(./@*)>2]")
```

```
[[1]]
<div id="wiio" lang="english" date="1978">
  <h2>Osmo Antero Wiio</h2>
  <p><i>Communication usually fails, except by accident.</i></p>
  <p><b>Source: </b>Wiion lait - ja vähän muidenkin</p>
</div>
```

This XPath query finds the nodes for which the text component has more than 50 characters.

```
XML::xpathSApply(parsed_doc, "//*[string-length(text())>50]")
```

```
[[1]]
<i>Technology is neither good nor bad; nor is it neutral.</i>
```

```
[[2]]
<p><b>Source: </b>"Books: On Hand". Venture Science Fiction. Vol. 2, no. 2. p. 66.</p>
```

```
[[3]]
<li>The 1% Rule: "Only 1% of the users of a website actively create new content, while ...
```

```
[[4]]
<li>D!@kwad Theory: "Normal Person + Anonymity + Audience = Total D!@kwad"</li>
```

```
[[5]]
<li>Godwin's Law: "As an online discussion grows longer, the probability of a comparison ...
```

```
[[6]]
<li>Poe's Law: "Without a clear indicator of the author's intent, parodies of extreme views ...
```

```
[[7]]
<li>Skitt's Law: "Any post correcting an error in another post will contain at least ...
```

```
[[8]]
<li>Law of Exclamation: "The more exclamation points used in an email (or other posting), the ...
```

```
[[9]]
<li>Cunningham's Law: "The best way to get the right answer on the Internet is not to ask ...
```

```
[[10]]
<li>Danth's Law: "If you have to insist that you've won an Internet argument, you've ...
```

```
[[11]]
<li>Law of the Echo Chamber: "If you feel comfortable enough to post an opinion of ...
```

```
[[12]]
<li>Munroe's Law: "You will never change anyone's opinion on anything by making a post ...
```

This XPath query finds all <div> nodes with 2 or fewer attributes.

```
XML::xpathSApply(parsed_doc, "//div[not(count(./@*)>2)]")
```

```
[[1]]
<div lang="english" date="1986">
  <h2>Melvin Kranzberg</h2>
  <p><i>Technology is neither good nor bad; nor is it neutral.</i> <br/><emph>(Kranzberg's ...
  <p><b>Source: </b><a href="https://www.jstor.org/stable/3105385">Technology and Culture...
</div>

[[2]]
<div lang="english" date="1958">
  <h2>Theodore Sturgeon</h2>
  <p><i>90% of everything is crap.</i> <br/><emph>(Sturgeon's Revelation)</emph></p>
  <p><b>Source: </b>"Books: On Hand". Venture Science Fiction. Vol. 2, no. 2. p. 66.</p>
</div>

[[3]]
<div id="other">
<h2>Others:</h2>
<ul><li>The 1% Rule: "Only 1% of the users of a website actively create new content...
<li>D!@kwad Theory: "Normal Person + Anonymity + Audience = Total D!@kwad"</li>
<li>Godwin's Law: "As an online discussion grows longer, the probability of a comparison ...
<li>Poe's Law: "Without a clear indicator of the author's intent, parodies of extreme ...
<li>Skitt's Law: "Any post correcting an error in another post will contain at least ...
<li>Law of Exclamation: "The more exclamation points used in an email (or other posting), the ...
<li>Cunningham's Law: "The best way to get the right answer on the Internet is not to ...
<li>The Wiki Rule: "There's a wiki for that."</li>
<li>Danth's Law: "If you have to insist that you've won an Internet argument, you've ...
<li>Law of the Echo Chamber: "If you feel comfortable enough to post an opinion of any ...
<li>Munroe's Law: "You will never change anyone's opinion on anything by making a post ...
</ul></div>
```

Can you predict what the following queries do? What they will return?

```
XML::xpathSApply(parsed_doc, "//div[@date='1958']")
XML::xpathSApply(parsed_doc, "//*[contains(text(), '%')]" )
XML::xpathSApply(parsed_doc, "//div[starts-with(./@id, 'wii0')]" )
```

Extracting Node Elements

XPath queries can also extract specific elements, using the `fun` option (`xmlValue`, `xmlAttrs`, `xmlGetAttr`, `xmlName`, `xmlChildren`, `xmlSize`).

For instance, `xmlValue` returns a node's value:

```
XML::xpathSApply(parsed_doc, "//title", fun = XML::xmlValue)
```

```
[1] "Laws of the Internet"
```

Function	Description	Example
<code>name(<node>)</code>	Returns the name of <node> or the first node in a node set	<code>//*[name()='title'];</code> Returns: <title>
<code>text(<node>)</code>	Returns the value of <node> or the first node in a node set	<code>//*[text()='The book homepage'];</code> Returns: <i> with value <i>The book homepage</i>
<code>@attribute</code>	Returns the value of a node's attribute or of the first node in a node set	<code>//div[@id='R_Inventor'];</code> Returns: <div> with attribute <i>id</i> value <i>R_Inventor</i>
<code>string-length(str1)</code>	Returns the length of str1. If there is no string argument, it returns the length of the string value of the current node	<code>//h1[string-length()>11];</code> Returns: <h1> with value <i>Robert Gentleman</i>
<code>translate(str1, str2, str3)</code>	Converts str1 by replacing the characters in str2 with the characters in str3	<code>//div[translate(./@date, '2003', '2005')='June/2005'];</code> Returns: first <div> node with date attribute value <i>June/2003</i>
<code>contains(str1, str2)</code>	Returns TRUE if str1 contains str2, otherwise FALSE	<code>//div[contains(@id, 'Inventor')];</code> Returns: first <div> node with id attribute value <i>R_Inventor</i>
<code>starts-with(str1, str2)</code>	Returns TRUE if str1 starts with str2, otherwise FALSE	<code>//i[starts-with(text(), 'The')];</code> Returns: <i> with value <i>The book homepage</i>
<code>substring-before(str1, str2)</code>	Returns the start of str1 before str2 occurs in it	<code>//div[substring-before(@date, '/')='June'];</code> Returns: <div> with date attribute value <i>June/2003</i>
<code>substring-after(str1, str2)</code>	Returns the remainder of str1 after str2 occurs in it	<code>//div[substring-after(@date, '/')=2003];</code> Returns: <div> with date attribute value <i>June/2003</i>
<code>not(arg)</code>	Returns TRUE if the boolean value is FALSE, and FALSE if the boolean value is TRUE	<code>//div[not(contains(@id, 'Inventor'))];</code> Returns: the <div> node that does not contain the string <i>Inventor</i> in its id attribute value
<code>local-name(<node>)</code>	Returns the name of the current <node> or the first node in a node set—without the namespace prefix	<code>//*[local-name()='address'];</code> Returns: <address>
<code>count(<node>)</code>	Returns the count of a nodeset <node>	<code>//div[count(./a)=0];</code> Result: The second <div> with one <a> child
<code>position(<node>)</code>	Returns the index position of <node> that is currently being processed	<code>//div/p[position()=1];</code> Result: The first <p> node in each <div> node
<code>last()</code>	Returns the number of items in the processed node list <node>	<code>//div/p[last()];</code> Result: The last <p> node in each <div> node

Table 16.1: Commonly-used XPath functions [6].

On the other hand, `xmlAttrs` returns a node's attributes. In the first call, the first component returns 4 nodes; the second component returns each of these nodes' attributes.

```
XML::xpathSApply(parsed_doc, "//div", XML::xmlAttrs)
```

```
[[1]]
      id      lang      date
"wiio" "english" "1978"

[[2]]
      lang      date
"english" "1986"

[[3]]
      lang      date
"english" "1958"

[[4]]
      id
"other"
```

Finally, `xmlGetAttr` can be used to return a specific attribute:

```
XML::xpathSApply(parsed_doc, "//div", XML::xmlGetAttr, "lang")
```

```
[[1]]      [[2]]      [[3]]      [[4]]
[1] "english" [1] "english" [1] "english" NULL
```


16.3.3 Regular Expressions

Regular expressions can be used to achieve the main web scraping objective, which is to extract relevant information from reams of data. Among this mostly unstructured data lurk **systematic elements**, which can be used to help the automation process, especially if quantitative methods are eventually going to be applied to the scraped data.

Systematic structures include numbers, names (countries, etc.), addresses (mailing, e-mailing, URLs, etc.), specific character strings, etc. Regular expressions (**regexprs**) are abstract sequences of strings that match concrete recurring patterns in text; they allow for the systematic extraction of the information components from plain text, HTML, and XML.

The examples in this section are based on [3].

Initializing the Environment

The Python module for regular expressions is `re`.

```
import re
```

Let us take a quick look at some basics, through the `re` method `match()`. We can try to match a pattern from the beginning of a string, as below:

```
re.match('super', 'supercalifragilisticexpialidocious')
```

```
<re.Match object; span=(0, 5), match='super'>
```

No such match occurs in the following chunk of code, however.

```
re.match('super', 'Supercalifragilisticexpialidocious')
```

The regular expression pattern (more on this in a moment) for “word” is `\w+`. The following bit of code would match the first word in a string:

```
w_regex = '\w+'
re.match(w_regex, 'Hello World!')
```

```
<re.Match object; span=(0, 5), match='Hello'>
```

Common Regular Expression Patterns

A **regular expression pattern** is a short form used to indicate a type of (sub)string:

- `\w+`: word
- `\d`: digit
- `\s`: space
- `.`: wildcard

- + or *: greedy match
- \W: not word
- \D: not digit
- \S: not space
- [a-z]: lower case group
- [A-Z]: upper case group

There are a few `re` functions which, combined with regexps, can make it easier to extract information from large, unstructured text documents:

- `split()`: splits a string on a regexp;
- `findall()`: finds all substrings matching a regexp in a string;
- `search()`: searches for a regexp in a string, and
- `match()`: matches an entire string based on a regexp

Each of these functions takes two arguments: a **regexp** (first) and a **string** (second). For instance, we can split a string on the spaces (and remove them):

```
re.split('\s+', 'Can you do the split?')
```

```
['Can', 'you', 'do', 'the', 'split?']
```

The `\` in the regexp above is crucial. The following code splits the sentence on the `s` (and removes them):

```
re.split('s+', 'Can you do the split?')
```

```
['Can you do the ', 'plit?']
```

We can also split on single spaces and remove them:

```
re.split('\s', 'Can you do the split?')
```

```
['Can', '', 'you', 'do', 'the', 'split?']
```

Alternatively, we can also split on the words and remove them:

```
re.split('\w+', 'Can you do the split?')
```

```
['', ' ', ' ', ' ', ' ', ' ', ' ', '?']
```

Or better yet, split on the non-words and remove them:

```
re.split('\W+', 'Can you do the split?')
```

```
['Can', 'you', 'do', 'the', 'split', '']
```

Let us take some time to study a silly sentence, saved as a string.

```
test_string = 'Oh they built the built the ship Titanic.
    It was a mistake. It cost more than 1.5 million dollars.
    Never again!'
test_string
```

'Oh they built the built the ship Titanic. It was a mistake. It cost more than 1.5 million dollars. Never again!'

In English, only three characters can end a sentence: ., ?, !.¹³ We create a regexp group (more on those in a moment) as follows:¹⁴

```
sent_ends = r"[.?!]"
```

We could then split the string into its constituent sentences:

```
print(re.split(sent_ends, test_string))
```

```
['Oh they built the built the ship Titanic', ' It was a mistake',
 ' It cost more than 1', '5 million dollars', ' Never again', '']
```

If we wanted to know how many such sentences there were, we simply use the `len()` function:

```
print(len(re.split(sent_ends, test_string)))
```

6

The regexp range consisting of words with an uppercase initial letter is easy to build:

```
cap_words = r"[A-Z]\w+" # Upper case characters
```

We can find all such words (and how many there are in the string) through:

```
print(re.findall(cap_words, test_string))
print(len(re.findall(cap_words, test_string)))
```

```
['Oh', 'Titanic', 'It', 'It', 'Never']
```

5

The regexp for spaces is:

```
spaces = r"\s+" # spaces
```

We can then split the string on spaces, and count the number of **tokens** (see Chapter 27, *Text Analysis and Text Mining*):

13: Apparently, nobody's heard of the interrobang...

14: In Python, regular expression patterns must be prefixed with an `r` to differentiate between the **raw string** and the **string's interpretation**.

```
print(re.split(spaces, test_string))
print(len(re.split(spaces, test_string)))
```

```
['Oh', 'they', 'built', 'the', 'built', 'the', 'ship', 'Titanic.',
 'It', 'was', 'a', 'mistake.', 'It', 'cost', 'more', 'than', '1.5',
 'million', 'dollars.', 'Never', 'again!']
21
```

The regexp for numbers (contiguous strings of digits) is:

```
numbers = r"\d+"
```

We can find all the numeric characters using:

```
print(re.findall(numbers, test_string))
print(len(re.findall(numbers, test_string)))
```

```
['1', '5']
2
```

The main difference between `search()` and `match()` is that `match()` tries to match from the beginning of a string, whereas `search()` looks for a match anywhere in the string.

Regular Expressions Groups '()' and Ranges '[']' With OR '|'

We can create more complicated regexps using **groups**, **ranges**, and/or “or” statements:

- `[a-zA-Z]+`: an unlimited number of lower and upper case English/French (unaccented) letters;
- `[0-9]`: the digits from 0 to 9;
- `[a-zA-Z'\.\-]+`: any combination of lower and upper case English/French (unaccented) letters, ' ., and -;
- `(a-z)`: the characters a, -, and z;
- `(\s+|,)`: any number of spaces, or a comma;
- `(\d+|\w+)`: words or numerics

For instance, consider the following text string and regexps groups:

```
text = 'On the 1st day of xmas, my boat sank.'
numbers_or_words = r"(\d+|\w+)"
spaces_or_commas = r"(\s+|,)"
```

This next chunk of code does exactly what one would expect:

```
print(re.findall(numbers_or_words,text))
```

```
['On', 'the', '1', 'st', 'day', 'of', 'xmas', 'my', 'boat', 'sank']
```

What about this one?

```
print(re.findall(spaces_or_commas,text))
```

```
[' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
```

Now, consider a different string:

```
text = "will something happen after the semi-colon; I don't think so"
```

What might happen in each of the following cases?

```
print(re.match(r"[a-z -]+",text))
print(re.match(r"[a-z ]+",text))
print(re.match(r"[a-z]+",text))
print(re.match(r"(a-z-)+",text))
```

16.3.4 BeautifulSoup

Simple web requests require some networking code to fetch a page and return the HTML contents.

Browsers do a lot of work to intelligently parse improper HTML syntax,¹⁵ so that something like

```
<a href="data-action-lab.com"> <b>link text<a> </b>
```

say, would be correctly interpreted as

```
<a href="data-action-lab.com"><b>link text</b></a>.
```

BeautifulSoup (BS) is a Python library that helps extract data out of HTML and XML files; it parses HTML files, even if they are broken. But BS does not simply convert bad HTML to good X/HTML; it allows a user to fully inspect the (proper) HTML structure it produces, in a programmatical fashion.¹⁶

Typical HTML elements to be extracted/read come in various formats, such as:

- text
- tables
- form field values
- images
- videos
- etc.

15: Only up to a certain point, of course.

16: The R equivalent is `rvest`; we will not describe how to use it, but you are **strongly encouraged** to read up on this versatile tool and to use it in the Exercises.

When BS has finished its work on an HTML file, the resulting *soup* is an API for **traversing**, **searching**, and **reading** the document's elements. In essence, it provides **idiomatic** ways of navigating, searching, and modifying the parse tree of the HTML file, which can save a fair amount of time.

For instance, `soup.find_all('a')` would find and output all `<a ...> ... ` tag pairs (with attributes and content) in the soup, whereas the following chunk of code would output the URLs found in the same tag pairs.

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

The *BeautifulSoup* documentation is quite explicit and provides numerous examples [1]. We use the lyrics to [Meet the Elements](#), a song by *They Might Be Giants*, to illustrate BeautifulSoup's functionality.

```
html_doc = """
<html>
<head><title>Meet the Elements</title> <meta name="author" content="They Might Be Giants"></head>
<body><p class="title"><b>Meet the Elements</b></p>

<p class="author"><i>They Might Be Giants</i></p>

<div class="lyrics"><p class="verse" id="verse1">
<a href="https://en.wikipedia.org/wiki/Iron" class="element" id="link1">Iron</a> is a metal, you
    see it every day<br>
<a href="https://en.wikipedia.org/wiki/Oxygen" class="element" id="link2">Oxygen</a>, eventually,
    will make it rust away<br>
<a href="https://en.wikipedia.org/wiki/Carbon" class="element" id="link3">Carbon</a> in its
    ordinary form is coal<br>
    Crush it together, and diamonds are born</p>

<p class="chorus" id="chorus1">
    Come on, come on, and meet the elements <br>
    May I introduce you to our friends, the elements? <br>
    Like a box of paints that are mixed to make every shade <br>
    They either combine to make a chemical compound or stand alone as they are</p>

<p class="verse" id="verse2">
<a href="https://en.wikipedia.org/wiki/Neon" class="element" id="link4">Neon</a>'s a gas that
    lights up the sign for a pizza place <br>
    The coins that you pay with are <a href="https://en.wikipedia.org/wiki/Copper" class="element"
        id="link5">copper</a>, <a href="https://en.wikipedia.org/wiki/Nickel" class="element"
        id="link6">nickel</a>, and <a href="https://en.wikipedia.org/wiki/Zinc" class="element"
        id="link7">zinc</a> <br>
<a href="https://en.wikipedia.org/wiki/Silicon" class="element" id="link8">Silicon</a> and oxygen
    make concrete bricks and glass <br>
    Now add some <a href="https://en.wikipedia.org/wiki/Gold" class="element" id="link9">gold</a> and
    <a href="https://en.wikipedia.org/wiki/Silver" class="element" id="link10">silver</a> for some
    pizza place </p>

<p class="chorus" id="chorus2">
    Come on, come on, and meet the elements <br>
    I think you should check out the ones they call the elements <br>
```

```

Like a box of paints that are mixed to make every shade <br>
They either combine to make a chemical compound or stand alone as they are <br>
Team up with other elements making compounds when they combine <br>
Or make up a simple element formed out of atoms of the one kind </p>

<p class="verse" id="verse3">
Balloons are full of <a href="https://en.wikipedia.org/wiki/Helium" class="element"
    id="link11">helium</a>, and so is every star <br>
Stars are mostly <a href="https://en.wikipedia.org/wiki/Hydrogen" class="element"
    id="link12">hydrogen</a>, which may someday fill your car <br>
Hey, who let in all these elephants? <br>
Did you know that elephants are made of elements? <br>
Elephants are mostly made of four elements <br>
And every living thing is mostly made of four elements <br>
Plants, bugs, birds, fish, bacteria and men <br>
Are mostly carbon, hydrogen, <a href="https://en.wikipedia.org/wiki/Nitrogen" class="element"
    id="link13">nitrogen</a>, and oxygen</p>

<p class="chorus" id="chorus3">
Come on, come on, and meet the elements <br>
You and I are complicated, but we're made of elements <br>
Like a box of paints that are mixed to make every shade <br>
They either combine to make a chemical compound or stand alone as they are <br>
Team up with other elements making compounds when they combine <br>
Or make up a simple element formed out of atoms of the one kind <br>
Come on come on and meet the elements <br>
Check out the ones they call the elements <br>
Like a box of paints that are mixed to make every shade <br>
They either combine to make a chemical compound or stand alone as they are</p>

</div>
"""

```

Note that the HTML file contains neither a `</body>` nor a `</html>` tag. We import the BeautifulSoup module, and parse the file into a soup using the `html.parser`.

```

from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
print(soup.prettify())

```

```

<html>
<head>
  <title>
    Meet the Elements
  </title>
  <meta content="They Might Be Giants" name="author"/>
</head>
<body>
  <p class="title">
    <b>
      Meet the Elements
    </b>
  </p>

```

```

<p class="author">
  <i>
    They Might Be Giants
  </i>
</p>
<div class="lyrics">
  <p class="verse" id="verse1">
    <a class="element" href="https://en.wikipedia.org/wiki/Iron" id="link1">
      Iron
    </a>
    is a metal, you see it every day
  <br/>

  ...

<p class="chorus" id="chorus3">
  Come on, come on, and meet the elements
  <br/>
  You and I are complicated, but we're made of elements
  <br/>
  Like a box of paints that are mixed to make every shade
  <br/>
  They either combine to make a chemical compound or stand alone as they are
  <br/>
  Team up with other elements making compounds when they combine
  <br/>
  Or make up a simple element formed out of atoms of the one kind
  <br/>
  Come on come on and meet the elements
  <br/>
  Check out the ones they call the elements
  <br/>
  Like a box of paints that are mixed to make every shade
  <br/>
  They either combine to make a chemical compound or stand alone as they are
</p>
</div>
</body>
</html>

```

The parser has “fixed” the file by appending the missing tags; it also indents the tags to make it easier to spot the document’s hierarchic (tree) structure.

BeautifulSoup Functionality

Is the functionality of BS clear from the following examples?

```
print(soup.title)
```

```
<title>Meet the Elements</title>
```



```
print(soup.title.name)
```

```
title
```

```
print(soup.title.string)
```

```
Meet the Elements
```

```
print(soup.title.parent.name)
```

```
head
```

```
print(soup.p)
```

```
<p class="title"><b>Meet the Elements</b></p>
```

```
soup.p['class']
```

```
['title']
```

```
print(soup.a)
```

```
<a class="element" href="https://en.wikipedia.org/wiki/Iron" id="link1">Iron</a>
```

```
soup.find_all('a')
```

```
[<a class="element" href="https://en.wikipedia.org/wiki/Iron" id="link1">Iron</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Oxygen" id="link2">Oxygen</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Carbon" id="link3">Carbon</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Neon" id="link4">Neon</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Copper" id="link5">copper</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Nickel" id="link6">nickel</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Zinc" id="link7">zinc</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Silicon" id="link8">Silicon</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Gold" id="link9">gold</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Silver" id="link10">silver</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Helium" id="link11">helium</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Hydrogen" id="link12">hydrogen</a>,
 <a class="element" href="https://en.wikipedia.org/wiki/Nitrogen" id="link13">nitrogen</a>]
```

```
print(soup.find(id="link5"))
```

```
<a class="element" href="https://en.wikipedia.org/wiki/Copper"
id="link5">copper</a>
```

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

```
https://en.wikipedia.org/wiki/Iron
https://en.wikipedia.org/wiki/Oxygen
https://en.wikipedia.org/wiki/Carbon
https://en.wikipedia.org/wiki/Neon
https://en.wikipedia.org/wiki/Copper
https://en.wikipedia.org/wiki/Nickel
https://en.wikipedia.org/wiki/Zinc
https://en.wikipedia.org/wiki/Silicon
https://en.wikipedia.org/wiki/Gold
https://en.wikipedia.org/wiki/Silver
https://en.wikipedia.org/wiki/Helium
https://en.wikipedia.org/wiki/Hydrogen
https://en.wikipedia.org/wiki/Nitrogen
```

```
print(soup.get_text())
```

Meet the Elements

Meet the Elements

They Might Be Giants

Iron is a metal, you see it every day

Oxygen, eventually, will make it rust away

Carbon in its ordinary form is coal

Crush it together, and diamonds are born

Come on, come on, and meet the elements

May I introduce you to our friends, the elements?

Like a box of paints that are mixed to make every shade

They either combine to make a chemical compound or stand alone as they are

Neon's a gas that lights up the sign for a pizza place

The coins that you pay with are copper, nickel, and zinc

Silicon and oxygen make concrete bricks and glass

Now add some gold and silver for some pizza place class

Come on, come on, and meet the elements

I think you should check out the ones they call the elements

Like a box of paints that are mixed to make every shade

They either combine to make a chemical compound or stand alone as they are

Team up with other elements making compounds when they combine

Or make up a simple element formed out of atoms of the one kind

Balloons are full of helium, and so is every star

Stars are mostly hydrogen, which may someday fill your car

Hey, who let in all these elephants?

Did you know that elephants are made of elements?

Elephants are mostly made of four elements
 And every living thing is mostly made of four elements
 Plants, bugs, birds, fish, bacteria and men
 Are mostly carbon, hydrogen, nitrogen, and oxygen
 Come on, come on, and meet the elements
 You and I are complicated, but we're made of elements
 Like a box of paints that are mixed to make every shade
 They either combine to make a chemical compound or stand alone as they are
 Team up with other elements making compounds when they combine
 Or make up a simple element formed out of atoms of the one kind
 Come on come on and meet the elements
 Check out the ones they call the elements
 Like a box of paints that are mixed to make every shade
 They either combine to make a chemical compound or stand alone as they are

16.3.5 Selenium

Selenium is a Python tool used to automate web browser interactions. It is used primarily for testing purposes, but it has data extraction uses as well. Mainly, it allows the user to open a browser and to act as a human being would:

- clicking buttons;
- entering information in forms;
- searching for specific information on a page, etc.

Selenium requires a driver to interface with the chosen browser. Firefox, for example, uses `geckodriver`.¹⁷

Selenium automatically controls a complete browser, including **rendering** the web documents and **running JavaScript**. This is useful for pages with a lot of dynamic content that is not in the base HTML. Selenium can program actions like “click on this button”, or “type this text”, to provide access to the dynamic HTML of the current state of the page, not unlike what happens in *Developer Tools* (but now the process can be fully automated). More information can be found in [9, 7, 8].

17: Here are the driver URL for supported browsers:

- [Chrome](#)
- [Edge](#)
- [Firefox](#)
- [Safari](#)

16.3.6 APIs

An **application programming interface** (API) is a website's way of giving programs access to their data, without the need for scraping. APIs provide **structured access to structured data**: not every bit of information will necessarily be made available to analysts.

For example, a finance site might offer an API with financial aggregate data, the *New York Times* might offer an API for news articles from a specific time period, Twitter might offer an API to collect tweets by users or hashtags, etc. In all cases, however, the data will be available in a **pre-defined, structured** format (often JSON).


In the examples we consider in Section 16.4, the APIs we consider have R/Python libraries that encapsulate all required networking and encoding. This means that users only need to read the library documentation to get a sense for what needs to be done to get the data.¹⁸

18: A full list of R API libraries can be found [here](#).

16.3.7 Specialized Uses and Applications


Although we will not be discussing them in these notes, it could prove useful for web scrapers to learn how to handle:

HTML Forms Sometimes we do not just want to receive data from the server, we also want to **send** data, such as a **username/password** combination to log in to a site. Other input types include: check boxes, radio buttons, hidden inputs, etc. Real users accomplish this by filling out forms and submitting them to the server. When this happens the browser looks at the form HTML and sends a request with the user inputs as *parameters*. The server can use those parameters to send back different data.

Encoding What if we wanted to write `
` as **text** in an HTML file? If we just type it in as-is, it would be interpreted as an HTML tag, not as text. The solution is to use HTML **encoding**. In order to type `
`, we have to encode it in a special form of text that the browser understands. An HTML decoder/encoder can be found [here](#) .

Combination HTML forms can specify a method for GET as well as for PUT. In that case the parameters are appended to the URL after a “?”, like so:


```
http://search.yahoo.com/search/?p=data+analysis&lang=en.
```

In that example, the parameter names are `p` and `lang`. The parameter value `data+analysis` actually represents the string “data analysis”, but spaces get encoded in URLs. Other characters (such as “/”) often are as well; use the urlencoder.org  to get the correct strings.

16.4 Examples

In this section, we provide web scraping examples (in R and Python) that highlight some of the notions we discussed in the chapter.¹⁹

16.4.1 Wikipedia

This example is inspired by a task found in [6]. We analyze the list of largest cities on the planet, found on [Wikipedia](#) .

²⁰

Preamble

We will be using the following R libraries:

- `stringr`, `stringi`, and `strex`, for string manipulation;
- `XML`, for reading and creating XML documents;
- `maps`, to display maps, and
- `rvest`, which provides a wrapper for HTTP requests in R.

19: These examples all worked as of Dec 2022; but it is possible that the websites that are being scraped have changed their structure or been deleted, or that the tools used have been updated/upgraded/made obsolete in the intervening time.

20: Wikipedia is a commonly-used source of data on various topics (in a first pass, at the very least), but it should probably not be your ONLY source of information.

Loading and Parsing the Data

We read the material from the Wikipedia website using `rvest`'s `read_html()` command, and we store it to the object `html`.

```
html <- rvest::read_html("https://en.Wikipedia.org/wiki/List_of_largest_cities")
```

A call to the object shows the entire structure of the page under the hood.

```
html
```

```
{html_document}
<html class="client-nojs" lang="en" dir="ltr">
[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...
[2] <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject ...
```

Now that we have the information from the webpage, we parse it to create a string of words.

```
cities_parsed <- XML::htmlParse(html, encoding="UTF-8")
```

Note that this new output contains the same information as the original object `html`, but that if it was displayed, it would be so in a format resembling what a human programmer would expect to see (at least, to some extent). We opt not to display it due to its excessive length.

Now that the information from the webpage is parsed, we create tables to hold the words, using XML's `readHTMLTable()`.

```
tables <- XML::readHTMLTable(cities_parsed, stringsAsFactors = FALSE)
```

Essentially, `readHTMLTable()` hunts for `<table>...</table>` tag pairs in the file; it finds 4 here. We get some structural information by calling `str` on the resulting object `tables`.

```
str(tables)
```

List of 4

```
$ NULL:'data.frame': 5 obs. of 1 variable:
..$ V1: chr [1:5] "Ekistics" "" "List of largest cities\nList of cities proper by ...
$ NULL:'data.frame': 84 obs. of 13 variables:
..$ V1 : chr [1:84] "City[a]" "Definition" " " "Tokyo" ...
..$ V2 : chr [1:84] "Country" "Population" "" "Japan" ...
..$ V3 : chr [1:84] "UN 2018 population estimates[b]" "Area.mw-parser-output ...
..$ V4 : chr [1:84] "City proper[c]" "Density(/km2)" "" "Metropolis prefecture" ...
..$ V5 : chr [1:84] "Urban area[8]" "Population" "" "13,515,271" ...
..$ V6 : chr [1:84] "Metropolitan area[d]" "Area(km2)" "" "2,191" ...
..$ V7 : chr [1:84] NA "Density(/km2)" "" "6,169[13]" ...
..$ V8 : chr [1:84] NA "Population" "" "39,105,000" ...
```

```

..$ V9 : chr [1:84] NA "Area(km2)" "" "8,231" ...
..$ V10: chr [1:84] NA "Density(/km2)" "" "4,751[e]" ...
..$ V11: chr [1:84] NA NA "" "37,274,000" ...
..$ V12: chr [1:84] NA NA "" "13,452" ...
..$ V13: chr [1:84] NA NA "" "2,771[14]" ...
$ NULL: 'data.frame': 7 obs. of 2 variables:
..$ V1: chr [1:7] "v\nt\ne\n\nWorld's largest cities" "City proper" "Metropolitan area" ...
..$ V2: chr [1:7] NA "Capitals\nAfrica\n\nAmericas (North\n\nLatin\nCentral\n\nSouth)\n\nAsia ...
$ NULL: 'data.frame': 9 obs. of 2 variables:
..$ V1: chr [1:9] "v\nt\ne\n\nCities" "Urban geography" "Urban government" "Urban economics" ...
..$ V2: chr [1:9] NA "Urban area\n\nCity centre\nDowntown\nSuburb\nExurb\nCore city\nTwin ..."

```

Data Processing and Data Cleaning

We extract the table containing the information of interest.

```
cities_table <- tables[[2]]
```

The column headers are not as we might want them:

```
colnames(cities_table)
```

```

[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7"
[13] "V8" "V9" "V10" "V11" "V12" "V13"

```

Compare with the second row of `cities_table`:

```
cities_table[2,]
```

```

      V1      V2      V3
2 Definition Population Area.mw-parser-output .nobold{font-weight:normal}(km2)
      V4      V5      V6      V7      V8      V9
2 Density(/km2) Population Area(km2) Density(/km2) Population Area(km2)
      V10 V11 V12 V13
2 Density(/km2) <NA> <NA> <NA>

```

This is still not ideal: the first and second rows of the table contain variable information, and the data itself starts with row 3. We need to manually input the column names, and delete the non-data rows.

```

colnames(cities_table) <- c("city", "country", "un.2018.pop", "city.def", "city.pop", "city.area",
  "city.den", "metro.pop", "metro.area", "metro.den", "urban.pop", "urban.area", "urban.den")
cities_table <- data.frame(cities_table[4:nrow(cities_table),])

```

We only select a sample of the columns of the table:

- `city [1];`
- `country [2];`
- `urban.pop [11];`
- `urban.area [12],` and
- `urban.den [13].`

```
cities_table <- cities_table[,c(1,2,11,12,13)]
```

It is never a bad idea to **validate** our work as we build the scraper: are we getting what we would expect along the way? Let us take a look at the structure of the data and compare the first 6 entries of the table to the information we can see on the Wikipedia page.

```
str(cities_table)
```

```
'data.frame':  81 obs. of  5 variables:
 $ city      : chr  "Tokyo" "Delhi" "Shanghai" "São Paulo" ...
 $ country   : chr  " Japan" " India" " China" " Brazil" ...
 $ urban.pop : chr  "37,274,000" "29,000,000" "--" "21,734,682" ...
 $ urban.area: chr  "13,452" "3,483" "--" "7,947" ...
 $ urban.den : chr  "2,771[14]" "8,326[16]" "--" "2,735[20]" ...
```

```
head(cities_table)
```

	city	country	urban.pop	urban.area	urban.den
4	Tokyo	Japan	37,274,000	13,452	2,771[14]
5	Delhi	India	29,000,000	3,483	8,326[16]
6	Shanghai	China	—	—	—
7	São Paulo	Brazil	21,734,682	7,947	2,735[20]
8	Mexico City	Mexico	21,804,515	7,866	2,772[22]
9	Cairo	Egypt	—	—	—

We see that all variables appear as **character strings**, and that there are oddities with some of the numerical values (square brackets, missing values, comma separators, etc.).

We obtain the numerical values using `stringr::str_extract()` and `regexps()`, or `strex::str_extract_numbers()` and `str_first_number()`.

The urban populations are all above 5M, and they are all displayed using comma separators, thus they all have values that look like `ddd,ddd,ddd`, where $d \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

We extract only the portion of the strings that follow this pattern from the population column using `str_extract()`,²¹ removing the commas after the fact using `gsub()`, and coercing the outcome to a numerical format using `as.numeric()`.

21: Which does not retain the footnote markers.

```
cities_table$urban.pop <- as.numeric(
  gsub(",", "",
    stringr::str_extract(
      cities_table$urban.pop,
      stringr::regex("\\d+,\\d+,\\d+")
    )
  )
)
```

22: Otherwise, the output would be a list.

The area column contains no footnote, so we can directly extract the comma-separated values (using `str_extract_numbers()`) and coerce to a vector using `as.numeric()`.²²

```
cities_table$urban.area <- as.numeric(
  strex::str_extract_numbers(
    cities_table$urban.area,
    commas=TRUE
  )
)
```

23: Both characters and numerics.

Finally, we extract the first number that appears in each density value, removing the footnotes,²³ using `str_first_number()`; the result is then coerced to a numeric vector using `as.numeric()`.

```
cities_table$urban.den <-
  as.numeric(strex::str_first_number(
    cities_table$urban.den,
    commas=TRUE
  )
)
```

The first six entries are shown below.

```
rownames(cities_table) = NULL
head(cities_table)
```

city	country	urban.pop	urban.area	urban.den
Tokyo	Japan	37274000	13452	2771
Delhi	India	29000000	3483	8326
Shanghai	China	NA	NA	NA
São Paulo	Brazil	21734682	7947	2735
Mexico City	Mexico	21804515	7866	2772
Cairo	Egypt	NA	NA	NA

We can download [latitude and longitude details](#)  for $\approx 41\text{K}$ cities.

```
world_cities = read.csv("worldcities.csv",
  stringsAsFactors = TRUE, nrow=200)
str(world_cities)
```

```
'data.frame':  200 obs. of  3 variables:
 $ city_ascii: Factor w/ 198 levels "Abidjan","Ahmedabad", ...
 $ lat      : num  35.69 -6.21 28.66 18.97 14.6 ...
 $ lng      : num  139.7 106.8 77.2 72.8 121 ...
```

We extract a 5-digit code for each city, in the hope of being able to match them in both datasets.

We remove accents using `stringi`'s `stri_trans_general()`, which will convert every character to its nearest equivalent in the Latin ASCII character list.


```

world_cities$code = stringi::stri_trans_general(
  tolower(
    substr(
      world_cities$city_ascii,1,5)
    ),
  "Latin-ASCII"
)

cities_table$code = stringi::stri_trans_general(
  tolower(
    substr(cities_table$city,1,5)
    ),
  "Latin-ASCII"
)

```

We merge the data frames:

```
(complete = merge(cities_table, world_cities, all.x=TRUE))
```

code	city	country	urban.pop	urban.area	urban.den	city_ascii	lat	lng
ahmed	Ahmedabad	India	6300000	NA	NA	Ahmedabad	23.0300	72.5800
alexa	Alexandria	Egypt	NA	NA	NA	Alexandria	31.2000	29.9167
atlan	Atlanta	United States	5949951	21690	274	Atlanta	33.7627	-84.4224
baghd	Baghdad	Iraq	NA	NA	NA	Baghdad	33.3500	44.4167
banga	Bangalore	India	NA	NA	NA	Bangalore	12.9699	77.5980
...
seoul	Seoul	South Korea	25514000	11704	2180	Seoul	37.5600	126.9900
shang	Shanghai	China	NA	NA	NA	Shangrao	28.4419	117.9633
shang	Shanghai	China	NA	NA	NA	Shanghai	31.1667	121.4667
shang	Shanghai	China	NA	NA	NA	Shangqiu	34.4259	115.6467
...
suzho	Suzhou	China	NA	NA	NA	Suzhou	31.3040	120.6164
suzho	Suzhou	China	NA	NA	NA	Suzhou	33.6333	116.9683
...
toron	Toronto	Canada	5928040	5906	1004	Toronto	43.7417	-79.3733
washi	Washington	United States	6263245	17009	368	Washington	38.9047	-77.0163
wuhan	Wuhan	China	NA	NA	NA	Wuhan	30.5872	114.2881
xi'an	Xi'an	China	NA	NA	NA	Xi'an	34.2667	108.9000
yango	Yangon	Myanmar	NA	NA	NA	NA	NA	NA

There are still some issues with the data:

- Suzhou shows up twice, with two different sets of coordinates, but the appropriate coordinates are found online to be (31.299999, 120.599998);
- Neither Yangon nor Fukuoka appear in the `world_cities` dataset, but their coordinates are found online to be (16.871311, 96.199379) and (33.583332, 130.399994), respectively;
- Shanghai has been associated to three cities: Shanghai, Shangrao, and Shangqiu, each with its own coordinates. As neither Shangrao nor Shangqiu appears in the original list, they may be removed with impunity,

24: Were they really, though? The problem arises because variables V11, V12, and V13 were poor choices in the first place. We will ask you to revisit this in the exercises.

- there is no population data for Foshan, but the Wikipedia page informs us that Foshan is included in the Guangzhou urban area, so we will remove the former from the dataset, and
- there are missing population, area, and density values for a number of cities, but these were missing in the original dataset, so we will leave it be for now.²⁴

```
# remove duplicate entries
complete = complete[-c(23,68,70,76),c(7,3,8,9,4,5,6)]
rownames(complete) = NULL

# add Fukuoka coordinates
complete[23,3] = 33.5833
complete[23,4] = 130.3999

# add Yangon coordinates
complete[80,3] = 16.8713
complete[80,4] = 96.1994

# add new factor levels for missing city names
complete$city_ascii = factor(complete$city_ascii,
  levels=c(levels(complete$city_ascii),"Yangon","Fukuoka"))
complete[23,1] = "Fukuoka"
complete[80,1] = "Yangon"

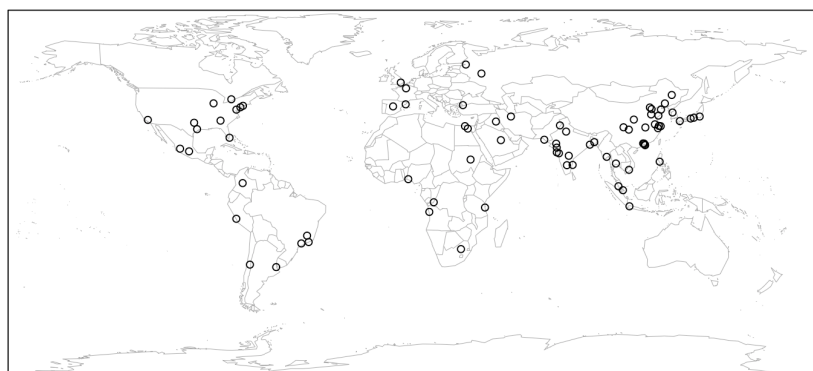
# rename city_ascii to city
colnames(complete)[1] <- "city"
```

Visualization

All the work we have done has brought the data in a format that is amenable to analysis. As an illustration, we plot the cities on a map of the world. We can display a Mercator projection by using `maps`'s `map()`.

```
par(oma=c(0,0,0,0)); par(mar=c(0,0,0,0))
maps::map("world", col = "darkgrey", lwd = .5, mar = c(0.1,0.1,0.1,0.1))
points(complete$lng, complete$lat, col = "black", cex = .8)
title("Locations of the 80 most populous urban areas", line=1)
box()
```

Locations of the 80 most populous urban areas



Loading and Parsing the Data

25: This version of the code requires that it be run before 3PM EST; the various webpages have a different format in the evenings, unfortunately; see exercises.

We get a handle on the website structure by studying the page for a single location, say [Ottawa, Ontario](#) ²⁵

```
ottawaURL = "https://weather.gc.ca/city/pages/on-118_metric_e.html"
```

The page looks something like the image in Figure 16.11.

Ottawa (Kanata - Orléans), ON

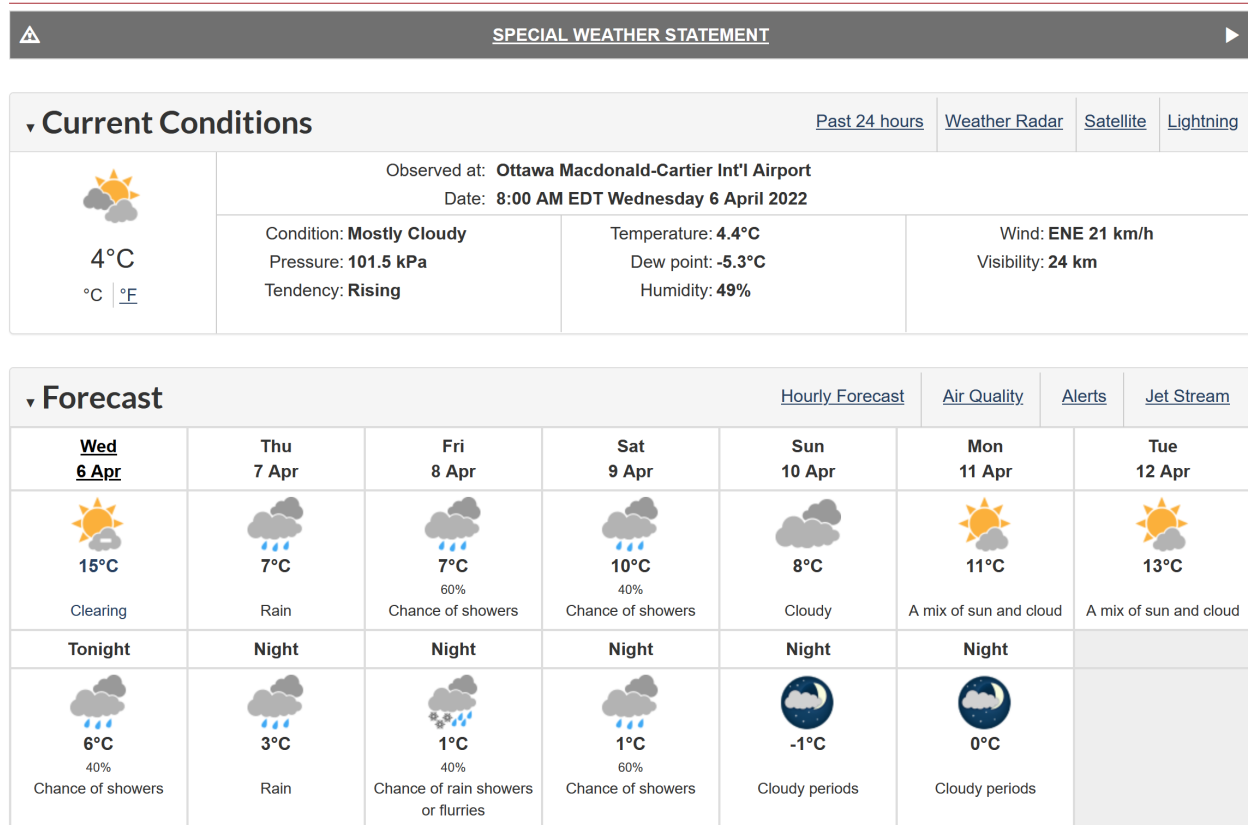


Figure 16.11: 7-day forecast for Ottawa, ON, on Wednesday April 6, 2022. [[weather.gc.ca](#)]

We download the HTML and load it into *BeautifulSoup*, using `html.parser`.²⁶

26: Other parsers can also be used, depending on the type of files with which we work.

```
ottawaHTML = urlopen(ottawaURL)
ottawaBS = BeautifulSoup(ottawaHTML, 'html.parser')
```

The soup (parsed content) is now available in `ottawaBS`. The data of interest is in there, we just need to pick it out of the document.

If we open developer tools pane in our browser, we can examine the specific HTML elements that contain the numbers we want. The table with the 7 day forecast appears to correspond to `div` element with `class=div-table` (see Figure 16.12); the weather information is contained in 7 columns, each of which is a `div` element with `class=div-column` (see Figure 16.13).

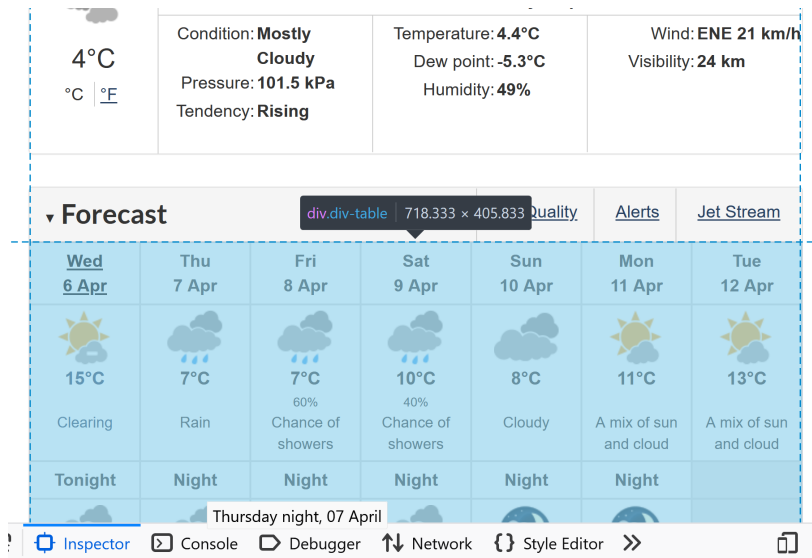


Figure 16.12: 7-day forecast for Ottawa, ON, on Wednesday April 6, 2022; the 'div' element with 'class=div-table' is highlighted in the Firefox Inspector.

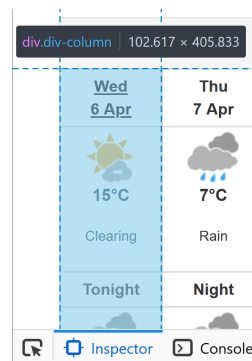


Figure 16.13: 7-day forecast for Ottawa, ON, on Wednesday April 6, 2022; the 'div' element with 'class=div-column' is highlighted in the Firefox Inspector.

We can find it in the soup `ottawaBS` as follows:

```
sevenDaysBS = ottawaBS.find_all('div', attrs={"class" : "div-column"})
```

We display the HTML for the first of those columns below.

```
print(sevenDaysBS[0].prettify())
```

```
<div class="div-column">
  <div class="div-row div-row1 div-row-head">
    <a href="/forecast/hourly/on-118_metric_e.html">
      <strong title="Friday">
        Fri
      </strong>
    <br/>
    7
    <abbr title="October">
      Oct
    </abbr>
  </a>
</div>
```

```

<a class="linkdate" href="/forecast/hourly/on-118_metric_e.html">
<div class="div-row div-row2 div-row-data">
  
  <p class="mrqn-bttm-0">
    <span class="high wxo-metric-hide" title="max">
      10°
      <abbr title="Celsius">
        C
      </abbr>
    <span class="abnTrend">
      *
    </span>
  </span>
  <span class="high wxo-imperial-hide wxo-city-hidden" title="max">
    50°
    <abbr title="Fahrenheit">
      F
    </abbr>
    <span class="abnTrend">
      *
    </span>
  </span>
</p>
<p class="mrqn-bttm-0 pop text-center" title="Chance of Precipitation">
  <small>
    60%
  </small>
</p>
<p class="mrqn-bttm-0">
  Chance of showers
</p>
</div>
</a>
<div class="div-row div-row3 div-row-head">
  Tonight
</div>
<div class="div-row div-row4 div-row-data">
  
  <p class="mrqn-bttm-0">
    <span class="low wxo-metric-hide" title="min">
      0°
      <abbr title="Celsius">
        C
      </abbr>
    </span>
    <span class="low wxo-imperial-hide wxo-city-hidden" title="min">
      32°
      <abbr title="Fahrenheit">
        F
      </abbr>
    </span>
  </p>
  <p class="mrqn-bttm-0 pop text-center">
  </p>
  <p class="mrqn-bttm-0">
    Mainly cloudy
  </p>
</div>

```

In each of the columns, the first row contains the date (see Figure 16.14), and the second the maximum forecast temperature during the day (see Figure 16.14).²⁷

27: Only if the page is accessed before 3PM, however.

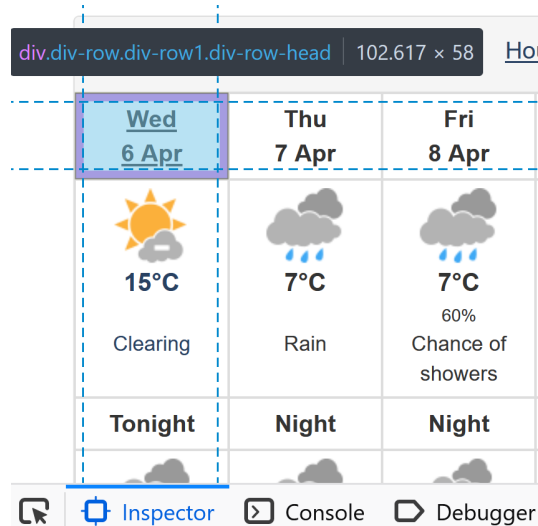


Figure 16.14: 7-day forecast for Ottawa, ON, on Wednesday April 6, 2022; the 'div' element with 'class=div-row1' is highlighted in the Firefox Inspector.

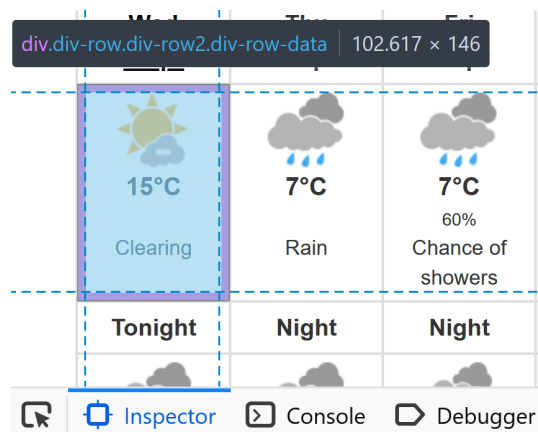


Figure 16.15: 7-day forecast for Ottawa, ON, on Wednesday April 6, 2022; the 'div' element with 'class=div-row2' is highlighted in the Firefox Inspector.

We can extract the strings in each of the first two cells of the first column using the `.strings` method, as below:

```
# date
list(sevenDaysBS[0].find(class_="div-row div-row1 div-row-head").strings)
# temp
list(sevenDaysBS[0].find(class_="high wxo-metric-hide").strings)
```

```
['Fri', '7\xa0', 'Oct']
['10°', 'C', '*']
```

The lists contains the information of interest, together with additional characters; for both variables, we join the list elements into a single string and remove the odd characters (°C, \xa0,*), using the `.replace()` method.

```
' '.join(list(sevenDaysBS[0].find(class_="div-row div-row1 div-row-head").strings))
    .replace("\xa0", "").replace("*", "")
'''.join(list(sevenDaysBS[0].find(class_="high wxo-metric-hide").strings))
    .replace("°C", "").replace("*", "")
```

```
'Fri 7 Oct'
'10'
```

28: Additional cleaning is required (see the various `.replace()` calls).

Based on this work, we now write functions that extract a 7-day forecast, the corresponding dates, the city name, and the province code given a URL of the right format.²⁸

```
def sevenDayForecast(url):
    html = urlopen(url)
    htmlBS = BeautifulSoup(html, 'html.parser')
    sevenDaysBS = htmlBS.find_all('div', attrs={"class" : "div-column"})
    temp_degree = []
    for day in sevenDaysBS:
        temp_de = int(
            ''.join(list(day.find(class_="high wxo-metric-hide").strings)).replace("°C", "")
            .replace("*", "")
        )
        temp_degree.append(temp_de)
    return temp_degree

def sevenDayForecastDates(url):
    html = urlopen(url)
    htmlBS = BeautifulSoup(html, 'html.parser')
    sevenDaysBS = htmlBS.find_all('div', attrs={"class" : "div-column"})
    temp_date = []
    for day in sevenDaysBS:
        temp_da = ' '.join(list(day.find(class_="div-row div-row1 div-row-head").strings))
            .replace("\xa0", "").replace("\n ", "").replace(" \n", "").replace("*", "")
        temp_date.append(temp_da)
    return temp_date

def cityName(url):
    html = urlopen(url)
    htmlBS = BeautifulSoup(html, 'html.parser')
    nameBS = htmlBS.find('h1', attrs={"property" : "name"})
    city_name = list(nameBS.strings)[0].replace(" \n", "").replace(", ", "")
    return city_name

def provinceCode(url):
    html = urlopen(url)
    htmlBS = BeautifulSoup(html, 'html.parser')
    nameBS = htmlBS.find('h1', attrs={"property" : "name"})
    province_code = list(nameBS.strings)[1]
    return province_code
```

29: Again, a reminder that this will only work if the code is run before 3PM EST, as the format of the webpage changes after that time.

We validated the functions on the Ottawa URL, on Oct 7, 2022.²⁹


```
sevenDayForecast(ottawaURL)
sevenDayForecastDates(ottawaURL)
cityName(ottawaURL)
provinceCode(ottawaURL)
```

```
[10, 11, 13, 12, 14, 17, 15]
['Fri 7 Oct', 'Sat 8 Oct', 'Sun 9 Oct', 'Mon 10 Oct', 'Tue 11 Oct', 'Wed 12 Oct', 'Thu 13 Oct']
'Ottawa (Kanata - Orléans)'
'ON'
```

Data Processing

We now prepare the data for analysis. We select the 20 Canadian cities that appear on the website's [main page](#) ³⁰.

For each of these cities, we extract the 7-day forecast, and display “today’s” temperature, “tomorrow’s” prediction, the weekly change 1 week from “today”, and the mean prediction over the 7-day forecast.

This could be done manually by feeding the URL to each of the 4 functions defined above (in the example for Ottawa), but we will use *BeautifulSoup* to scrape the information automatically (and cleanly).³⁰ We start by finding the URL for each of the cities on the main page.

```
wURL = "https://weather.gc.ca/canada_e.html"
wHTML = urlopen(wURL)
wBS = BeautifulSoup(wHTML, 'html.parser')
tableBS = wBS.find('table', attrs={"class" : "table
    table-hover table-striped table-condensed"})
citiesBS = tableBS.find_all('a', href=True)

citiesFURLs = []
for a in citiesBS:
    temp = a['href']
    citiesFURLs.append(temp)

citiesURLs = ["https://weather.gc.ca" + citiesFURLs[index]
    for index in range(len(citiesFURLs))]
```

30: We include only minimal comments in what follows; it may prove helpful to visit the corresponding web pages to clarify the context and make sense of the code outputs.

Next we build a dictionary containing the desired data, for each city:³¹

```
today_date = sevenDayForecastDates(citiesURLs[0])[0]

row_dict = []
for row in range(len(citiesURLs)):
    d = dict()
    tmp = sevenDayForecast(citiesURLs[row])
    d['city'] = cityName(citiesURLs[row])
    d['province'] = provinceCode(citiesURLs[row])
    d['date'] = today_date
    d['today'] = tmp[0]
```

31: Date of scraping: Oct 7, 2022.

```
d['tomorrow'] = tmp[1]
d['1 week change'] = np.subtract(tmp[6], tmp[0])
d['weekly mean'] = np.mean(tmp)
row_dict.append(d)
```

Finally, we convert the dictionary into a pandas data frame:

```
wDF = pandas.DataFrame(row_dict)
wDF
```

	city	province	...	1 week change	weekly mean
0	Calgary	AB	...	6	20.000000
1	Charlottetown	PE	...	-3	14.142857
2	Edmonton	AB	...	0	20.285714
3	Fredericton	NB	...	-4	15.571429
4	Halifax	NS	...	-2	15.285714
5	Iqaluit	NU	...	2	-1.142857
6	Montréal	QC	...	0	14.571429
7	Ottawa (Kanata - Orléans)	ON	...	5	13.142857
8	Prince George	BC	...	-2	16.142857
9	Québec	QC	...	-4	12.714286
10	Regina	SK	...	3	18.428571
11	Saskatoon	SK	...	0	19.000000
12	St. John's	NL	...	-2	12.714286
13	Thunder Bay	ON	...	2	11.857143
14	Toronto	ON	...	4	14.857143
15	Vancouver	BC	...	-2	18.571429
16	Victoria	BC	...	-2	20.285714
17	Whitehorse	YT	...	-13	10.571429
18	Winnipeg	MB	...	3	14.285714
19	Yellowknife	NT	...	-9	7.571429

[20 rows x 7 columns]

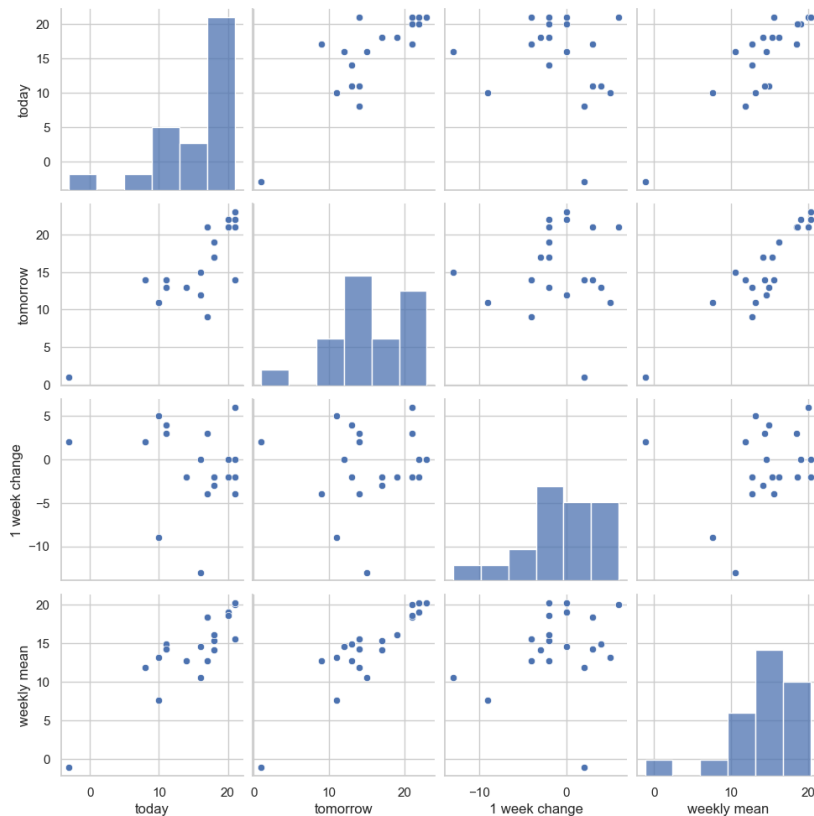
Visualization

As a last exercise, we provide a basic visualization for the collected dataset.

```
import seaborn as sns
sns.set(style='whitegrid')
```

We use seaborn's `pairplot()` to produce the scatterplot matrix of the data, and matplotlib's `plt()` to display it.

```
cols = ['today', 'tomorrow', '1 week change',
        'weekly mean']
sns.pairplot(data=wDF[cols], size=2.5)
```



Perhaps unsurprisingly, there does not seem to be much insight available in the dataset. If there is an association between tomorrow's prediction and the prediction one week from now, we require more information to explore it; data collected on a daily basis, perhaps?³²

That is an important point to keep in mind: the process is sometimes long and complicated, but that **does not always translate into insight at the end of the day**.³³

32: In which case, it would be useful to save the data; how would this be accomplished?

33: Unless the absence of an apparent link is insight. . . which it could very well be, in certain cases.

16.4.3 CFL Play-by-Play

In this example, we obtain **structured play-by-play** data for past CFL.³⁴ games. We could use this information to ask questions such as:

- how often do teams convert on 3rd and X?
- do teams come back from 7+ pt deficits in the 4th quarter?
- etc.

34: Canadian Football League.

Preamble

Before you start, make sure that *BeautifulSoup*, *Selenium*, *Pandas*, *Firefox*, and *Geckodriver* are installed in your Python environment. You can use the code below to install the Python modules.

- `pip3 install beautifulsoup4`
- `pip3 install pandas`
- `pip3 install selenium`

You can get download information for Firefox and Geckodriver here:

- [Firefox ↗](#)
- [Geckodriver ↗](#)

Of course, other browsers have their own installation information. We will use the following Python modules for pulling data out of HTML and XML files (BeautifulSoup), for dealing with potentially dynamic websites (Selenium), to open URLs (urllib.request), and other regular tasks.

```
from bs4 import BeautifulSoup
from pyvirtualdisplay import Display
from selenium import webdriver
from urllib.request import urlopen
import csv
import pandas
import time
import warnings; warnings.filterwarnings('ignore')
```

Game Schedule

Let us start by getting a list of all games in a season; we will switch to processing data on a game-by-game basis at a later stage. All games in a season (2016, say) are listed at a single URL in the following format.

```
year = 2016
scheduleURL = 'https://www.cfl.ca/schedule/?season={}'
              .format(year)
```

This produces the following URL:

```
scheduleURL
```

```
'https://www.cfl.ca/schedule/?season=2016'
```

Now we open the schedule page and parse it with *BeautifulSoup*:

```
scheduleHTML = urlopen(scheduleURL)
scheduleBS = BeautifulSoup(scheduleHTML, 'html.parser')
```

We could display the HTML code with:

```
scheduleBS
```

Warning: the HTML file contains a lot of information, so the display has been suppressed. For completeness' sake, when rendered in a browser, the page looks like the image in Figure 16.16.

TEAM SITES

FRANÇAIS

FOLLOW

NEWSLETTER

SEARCH

CFL

NEWS

VIDEO

SCHEDULE

STANDINGS

STATS

PLAYERS

TICKETS

SHOP

FORUMS

...

SEASON
2016


WEEK
PRESEASON WEEK 1

TIMEZONE
EDT

PRESEASON WEEK 1

▼


WED JUN 8
FINAL

 MTL

13

@

36


WPG 

GAMETRACKER

>

▼


SAT JUN 11
FINAL

 HAM

16

@

25


TOR 

GAMETRACKER

>

▼


SAT JUN 11
FINAL

 BC

28

@

16


SSK 

GAMETRACKER

>

▼


SAT JUN 11
FINAL

 EDM

23

@

13


CGY 

GAMETRACKER

>

▼


MON JUN 13
FINAL

 WPG

14

@

18

OTT 


GAMETRACKER

>

PRESEASON WEEK 2

▼


FRI JUN 17
FINAL

 OTT

25

@

42

HAM 

GAMETRACKER

>

Figure 16.16: Extract of the 2016 CFL schedule and results cfl.ca.

35: A 25-16 victory by the Toronto Argonauts over the Hamilton Tiger-Cats.

36: We might need to try right-clicking over a few locations as there sub-elements in the game box.

37: See Figure 16.17.

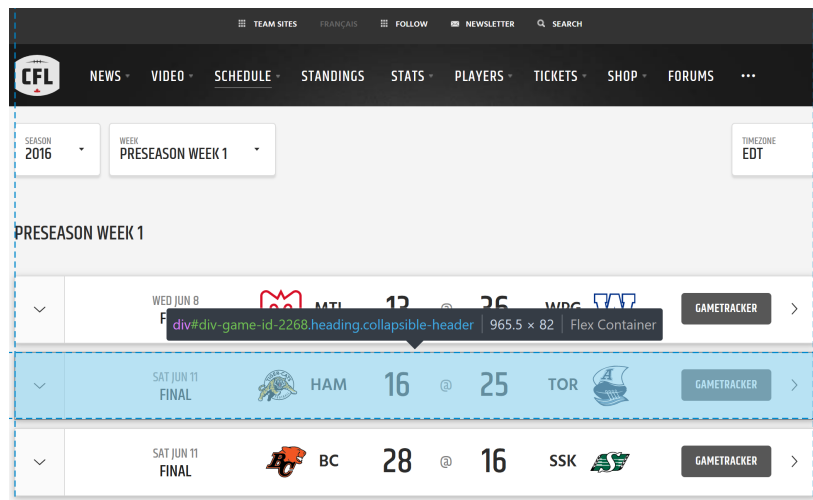
Figure 16.17: CFL 2016 schedule and results; the 'div' element with 'heading collapsible-header' is highlighted in the Firefox Inspector.

We could sift through the HTML to try to find what each piece of code corresponds to on , but that is not the most efficient approach to use.

Instead, we use the Developer Tools to get a better idea. In the example below, let's say we are interested in the second pre-season game.³⁵

Right-click on the box containing the game information, and select "Inspect Element (Q)" from the menu that appears. In the Developer Tool, you will be taken to the section of HTML code corresponding to the element you selected.³⁶

Each game is represented by a row. According to developer tools, these rows are div elements with the class heading collapsible-header.³⁷



Scrolling down on the schedule page, it appears as though every game is presented in the same format, so it is worth a shot to ask *Beautiful Soup* to find all rows that contain the class heading collapsible-header.³⁸

38: Note the _ after class, and the single quotes.

```
scheduleRows = scheduleBS.findAll(class_='heading
collapsible-header')
```

Here's a better view of a single row, with some parts omitted:

```
<div id="div-game-id-2268" class="heading collapsible-header">
  <div class="controls">
    ...
  </div>
  <div class="sponsored">
    ...
  </div>
  <div class="date-time">
    ...
  </div>
  <div class="matchup">
    ...
  </div>

  <div class="action">
    <a
      href="javascript:void(0);"
      data-url="https://www.cfl.ca/games/2268/hamilton-tiger-cats-vs-toronto-argonauts/"
      class="gametracker">
      <span class="btn">Gametracker</span>
    </a>
  </div>
</div>
```

We want the URL that the “GAMETRACKER” button links to – this is the game page that contains the play-by-play info. The link is found in the data-url attribute, rather than in the href attribute. We can get the link for the 2nd pre-season game by querying `scheduleRows[1]`.³⁹

39: Recall that list indexing starts with 0 in Python.

```
row = scheduleRows[1]
button = row.find(class_='gametracker')
button['data-url']
```

```
'https://www.cfl.ca/games/2268/hamilton-tiger-cats-vs-toronto-argonauts/'
```

These are all the steps we need to get the list of game page URLs for an entire season.

We might also want to store each of these game pages in a Python array. This can be done as follows.

```
urls = []

for row in scheduleRows:
    button = row.find(class_='gametracker')
    url = button['data-url']
    urls.append(url)

# uncomment to display the URLs
# print(urls)

df = pandas.DataFrame(urls)
df.to_csv(path_or_buf='Data/CFL_Schedule_2016.csv',
          header=False)
```

Incidentally, how many games were played in total in 2016, including the pre-season and the playoffs?

That is easy to answer:

```
len(urls)
```

95

Scraping Game Data

Here is a URL for one particular game.

```
gameURL = 'https://www.cfl.ca/games/2391/ottawa-redblacks-vs-toronto-argonauts'
```

The screenshot of Figure 16.18 shows the page as it is rendered in the browser **after** clicking the “PLAY BY PLAY” button.

FINAL MON JUL 24

OTT 0 9 0 15
TOR 3 9 5 10

ATTENDANCE: 15,801

MATCHUP RECAP BOX SCORE **PLAY BY PLAY** VIDEOS GAME NOTES

TYPE OF PLAY ALL

1ST QUARTER

PLAY	TEAM	DN	YDS	LOS	TYPE	DETAILS	AWAY	HOME
1	OTT	-	-	O35	Kickoff	(15:00) B. MAHER Kickoff (60 yds), Returned by M. JACKSON from T15 (23 yds), Special Teams Tackle: J. BOLDUC	0	0
2	TOR	1	10	T38	Pass	(14:53) R. RAY Completed Pass to A. COOMBS, caught at T34 (26 yds, 26 YAC), Pushed Out of Bounds by K. JOHNSON	0	0
3	TOR	1	10	O46	Pass	(14:29) R. RAY Incomplete Pass intended for S. GREEN at O39	0	0
4	TOR	2	10	O46	Pass	(13:59) R. RAY Completed Pass to A. EDWARDS, caught at O41 (6 yds, 1 YAC), Tackle: K. BASS	0	0
5	TOR	3	4	O40	Field Goal	(13:18) L. HAJRULLAHU Field Goal Attempt (47 yds), Good	0	3
6	OTT	1	10	O35	Pass	(13:08) T. HARRIS Incomplete Pass intended for P. LAVOIE at O40, Pass dropped.	0	3

Figure 16.18: Play-by-play data for the July 24, 2016 CFL game between the Ottawa Redblacks and the Toronto Argonauts cfl.ca.

The page actually only loads the play-by-play data once the “PLAY BY PLAY” button is pressed. If we download the HTML before pressing the button, the data just **isn’t there**.

```
gameBS = BeautifulSoup(urlopen(gameURL))
gameBS.text.count('Kickoff')
```

0

The page does contain JavaScript code that tells the browser to fetch more data when the button is clicked and add it to the page. The most straightforward way to get this data is to run a browser but control it automatically. All we need is a way to identify the button to press.

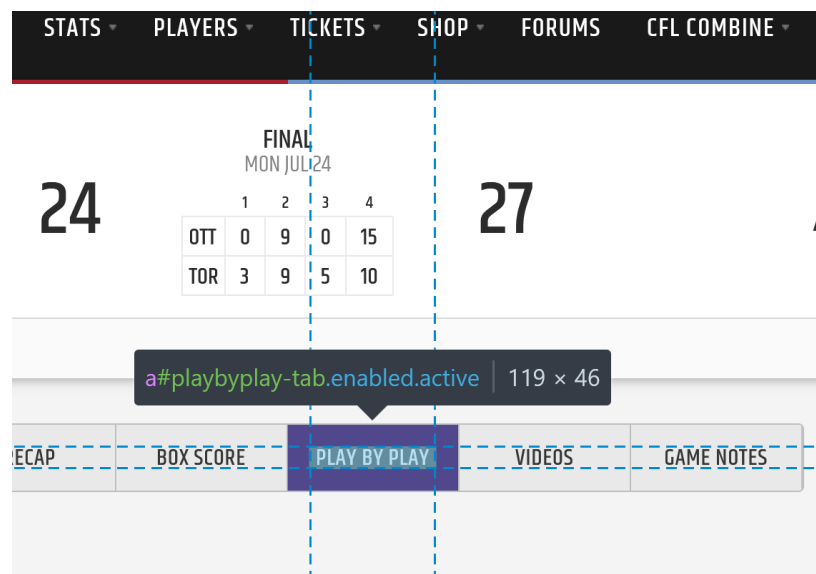


Figure 16.19: Play-by-play data for the July 24, 2016 CFL game between the Ottawa Redblacks and the Toronto Argonauts; the ‘div’ element with ‘playbyplay-tab’ is highlighted in the Firefox Inspector.

Luckily the button has a (unique) id (see Figure 16.19), so we can use that. We define an XPATH string for that id.


```
pbp_btn_xpath = '//*[@id="playbyplay-tab"]'
```

For browser automation, we use Firefox with Selenium – it is important to ensure that geckodriver is installed (or whatever the appropriate driver is for the browser in use).

In the next block, we run code for the driver object (in this case, Selenium controlling Firefox), telling it to load the page, click the button, and then get the HTML. Depending on the system, the variable `executable_path` will vary.

```
display = Display(visible=0, size=(1440, 1080))
display.start()
driver = webdriver.Firefox(executable_path='/usr/local/bin/geckodriver')

# Open the page
driver.get(gameURL)

# Wait for loading
time.sleep(5)
# less about robots.txt but more about content "physically" being there

# Click button to get play-by-play data
playbyplay_btn = driver.find_element_by_xpath(pbp_btn_xpath)
playbyplay_btn.click()

# Wait again for loading
time.sleep(5)

# Take HTML and save in BS object
soup = BeautifulSoup(driver.page_source)
driver.close()
```

The URL of the loaded play-by-play page can be loaded and parsed into a soup.

```
pbpURL = 'https://www.cfl.ca/games/2391/ottawa-redblacks-vs-toronto-argonauts#playbyplay'
pbpHTML = urlopen(pbpURL)
soup = BeautifulSoup(pbpHTML, 'html.parser')
```

Now that we have the HTML of the loaded page, we can extract data as usual with Beautiful Soup, such as finding the home team and the away teams, and so on.

```
# away, home
[soup.find(class_='js-data-team_2_location').text,
soup.find(class_='js-data-team_1_location').text]
```

```
['Ottawa', 'Toronto']
```

16.4.4 Bad HTML

When we write a R/Python program with incorrect syntax, we get an error and our program does not work. If we write an HTML page with incorrect syntax, there's a good chance that browsers will be able to make sense of it anyway – browsers **try to guess** ways to correct each error.

We can check whether a webpage on the internet uses correct syntax or not by entering the URL at validator.w3.org.

If what we see in our browsers is a fixed-up version of the HTML, then when we parse HTML with Python we'd like to be able to get a similarly fixed-up version. We look at some simple examples of how *Beautiful Soup* handles bad HTML.

```
from bs4 import BeautifulSoup
```

First we pass *Beautiful Soup* a proper (incomplete) HTML document:

```
goodBS = BeautifulSoup(
    '<html><head><title>blah</title></head><body></body></html>', 'html.parser')
```

As expected, we can operate with the parsed document, such as finding elements and getting their data.

```
goodBS.find('title').text
```

```
'blah'
```

Now what if we omit the closing `</title>` tag? We print the corrected version that *BeautifulSoup* builds.

```
badBS = BeautifulSoup(
    '<html><head><title>blah</head><body></body></html>',
    'html.parser')
print(badBS)
```

```
<html><head><title>blah</title></head><body></body></html>
```

You see that the closing tag has been returned. Similar behaviour is seen in the following examples where tags are misplaced or omitted. Note that although `` (list item) tags are supposed to be put inside a list tag such as `` (unordered list) or `` (ordered list), *Beautiful Soup* doesn't add those tags.

```
badBS = BeautifulSoup(
    '<html><head></head><body><li><em>hi</body></em></html>',
    'html.parser')
print(badBS.prettify())
```



```
<html>
<html>
  <head>
</head>
  <body>
    <li>
      <em>
        hi
      </em>
    </li>
  </body>
</html>
```

```
badBS = BeautifulSoup(
  '<html><head></head><body><li><em>hi<li></body></em>
  </html>', 'html.parser')
print(badBS.prettify())
```

```
<html>
<head>
</head>
<body>
  <li>
    <em>
      hi
    <li>
    </li>
    </em>
  </li>
</body>
</html>
```

In general, if the browser can do a good enough job to render an HTML page as intended, we can trust *BeautifulSoup* to fix things up **logically**. But when we automate the data collection process, we do not usually visit each page before it is scraped; there might be surprises in store!

16.4.5 Extracting Text from a PDF File

[Apache Tika](#)  can be used to convert PDF files to TXT files, but a few R libraries can also do so.⁴⁰ We use the `pdftools` library to extract text from the [DAL Data Visualization Learning Map](#) .

40: And are potentially easier to use, depending on the document's structure.

```
library(pdftools)
DAL <- pdf_text("https://www.data-action-lab.com/wp-content/uploads/2020/01/
  Learning-Map-Data-Visualization-ACF0.pdf")
length(DAL)
N <- 1:length(DAL)
```



```
DAL.clean[6]
```

```
[1] "DAL instructors have consulted for DATA EXPLORATION (and taught to participants from) a
AND DATA VISUALIZATION variety of groups, a selection of which is shown below: $ Canada Revenue
Agency $ Canada School of Public Service's Digital Academy $ Canadian Air Transport Security
Authority $ Canadian Coast Guard $ Canadian Food Inspection Agency $ Canadian Institute for
Health Information $ The Children's Hospital of Eastern Ontario $ Communications Research
Centre Canada $ Department of National Defence $ Environment and Climate Change Canada
$ Fisheries and Ocean Canada $ Health Canada $ Immigration, Refugees and Citizenship
Canada $ Indigenous and Northern Affairs Canada $ Natural Resources Canada $ Nuclear Waste
Management Organization $ Office of the Privacy Commissioner of Canada $ Privy Council Office
$ Public Services and Procurement Canada $ Royal Canadian Mounted Police $ Transport Canada
$ Treasury Board Secretariat Consult our Data Training Catalogues for a list of practical data
analysis and data leadership courses. Visit data-action-lab.com or contact info@data-action-lab.com
for more information. DATA ACTION LAB | info@data-action-lab.com"
```

Not too shabby, eh? It's almost readable, even!

16.4.6 YouTube Video Titles

In this example, we will see how to use the YouTube API to scrape the titles of YouTube videos.

```
from apiclient.discovery import build
from apiclient.errors import HttpError
from oauth2client.tools import argparser

# some of these will only be useful for the exercises
import pandas
from functional import seq
import codecs
import glob
import html
import os
import re
import unicode
import urllib
import urllib.request
import warnings; warnings.filterwarnings('ignore')
```

Authentication

The task is to build or add to a corpus of text by fetching video transcripts from YouTube. To use the YouTube API, we need to authenticate ourselves. Create a config.json file in the main directory, whose only content looks like this:

```
{ "DEVELOPER_KEY": "your_key_here" }
```

Instructions on how to obtain a key are provided [here](#) ↗

Once that done, we create an object `youtube`, through which we can access YouTube API methods (more information is available at [Wikipedia](#) ↗, [YouTube I](#) ↗, [YouTube II](#) ↗).

```
config = seq.json('config.json').dict()

DEVELOPER_KEY = config['DEVELOPER_KEY']
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"

youtube = build(YOUTUBE_API_SERVICE_NAME,
                YOUTUBE_API_VERSION,
                developerKey=DEVELOPER_KEY)
```

YouTube API

We could hand-pick videos to read, but we will take a shortcut by getting all the transcripts in a whole playlist of videos.

The first task is taking a playlist ID and using the API to get the video IDs of each entry in the playlist. The API for getting entries in a playlist is **paginated**. This means that we have to make one request for the first chunk of entries, then make another request to get some more entries, and so on until we have got the entire playlist.

It's designed this way so that we don't download more than we need; for example if we were building an infinite scrolling menu, we wouldn't want to load everything up front.

41: In this example, 10 videos at a time.

After we have obtained the first chunk,⁴¹ we need to tell the API where to start the next chunk. This is done using a **page token**.

We take the `nextPageToken` of the response we get, and pass it to the API for the next request, until the API returns no `nextPageToken` value.

```
def fetch_playlist_videos(playlistId):
    """
    get all videos in a playlist.
    Returns: list of dictionaries representing
            playlistItem resources,
    see https://developers.google.com/youtube/v3/docs/
        /playlistItems#resource-representation
    for the structure of this resource
    """

    # API method: https://developers.google.com/youtube/
    # v3/docs/playlistItems/list
    res = youtube.playlistItems().list(
        part="snippet",
        playlistId=playlistId,
        maxResults="10").execute()
```

```

nextPageToken = res.get('nextPageToken')
while ('nextPageToken' in res):
    nextPage = youtube.playlistItems().list(
        part="snippet",
        playlistId=playlistId,
        maxResults="10",
        pageToken=nextPageToken).execute()
    res['items'] = res['items'] + nextPage['items']

    if 'nextPageToken' not in nextPage:
        res.pop('nextPageToken', None)
    else:
        nextPageToken = nextPage['nextPageToken']

return res['items']

```

Playlist Extraction

The playlist entries come in the form of `playlistItem` **resource dictionaries**.⁴² In the Python API, the object is a **nested dictionary**. We want to get to the video ID, which we will do for all playlist items.

First, take the time to explore the following YouTube playlist: [Introduction to Quantitative Consulting](#) ↗.

42: A data format defined in the API documentation that contains fields for all the information associated with an item in a playlist.

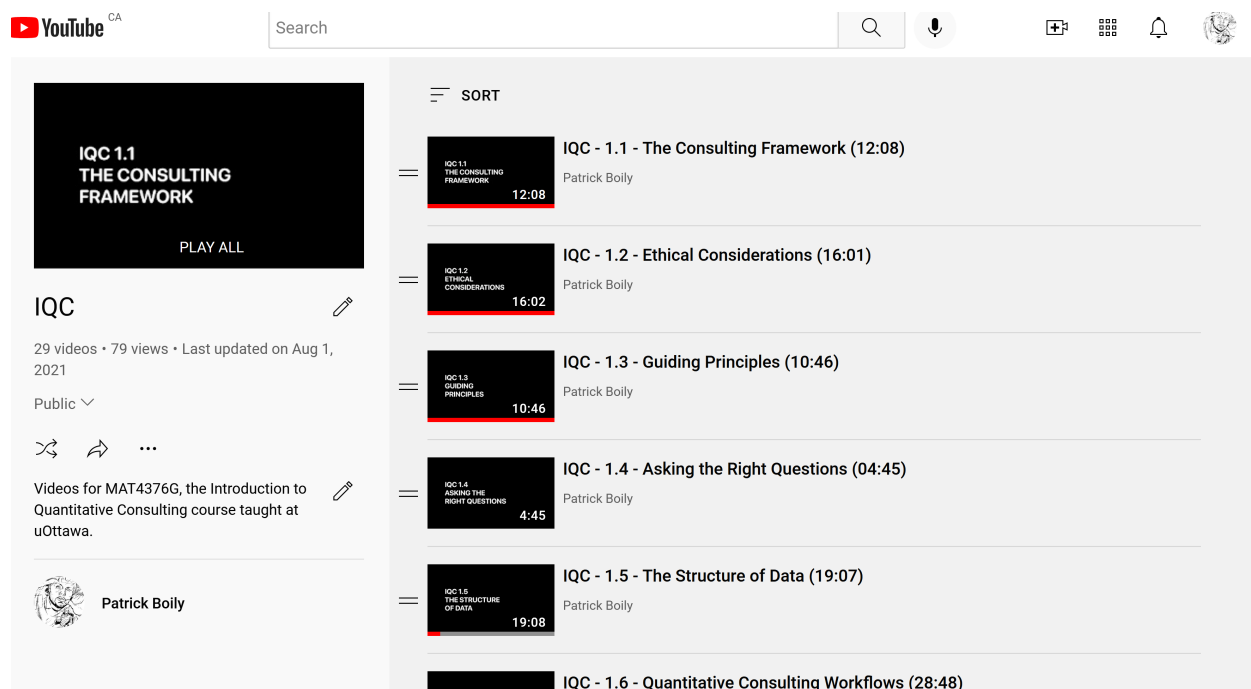


Figure 16.20: Introduction to Quantitative Consulting YouTube playlist.

Next, we build the list of videos.

```

# some playlists with English transcripts available
IQCPlaylist = ['PLbVTnkp2K536WxfoqSvoY08aJ3sLBg9mI']

```

```
videos = []
for playlistID in IQCPlaylist:
    videos += fetch_playlist_videos(playlistID)
```

We can explore the list by looking at the 3rd video in the playlist, say.

```
print(videos[2])
```

```
{'kind': 'youtube#playlistItem',
 'etag': 'LYF00si6rdPvtzVAqAY-TobYZnE',
 'id': 'UExiVlRua3AySzUzNld4Zm9xU3ZvWTA4YUozc0xCZzltSS4xMkVVGQjNCMUM1N0RFNEUx',
 'snippet': {'publishedAt': '2020-06-20T21:18:23Z',
 'channelId': 'UCIi6fq-A7sTT4iBDQUKekyg',
 'title': 'IQC - 1.3 - Guiding Principles (10:46)',
 'description': '1.3.1 Best Practices\n1.3.2 The Good, the Bad, and the Ugly',
 'thumbnails':
  {'default':
   {'url': 'https://i.ytimg.com/vi/eodNQzJFJpg/default.jpg', 'width': 120, 'height': 90},
   'medium':
   {'url': 'https://i.ytimg.com/vi/eodNQzJFJpg/mqdefault.jpg', 'width': 320, 'height': 180},
   'high':
   {'url': 'https://i.ytimg.com/vi/eodNQzJFJpg/hqdefault.jpg', 'width': 480, 'height': 360},
   'standard':
   {'url': 'https://i.ytimg.com/vi/eodNQzJFJpg/sddefault.jpg', 'width': 640, 'height': 480},
   'maxres':
   {'url': 'https://i.ytimg.com/vi/eodNQzJFJpg/maxresdefault.jpg', 'width': 1280, 'height': 720}},
 'channelTitle': 'Patrick Boily',
 'playlistId': 'PLbVTnkp2K536WxfoqSvoY08aJ3sLBg9mI',
 'position': 2,
 'resourceId': {'kind': 'youtube#video', 'videoId': 'eodNQzJFJpg'},
 'videoOwnerChannelTitle': 'Patrick Boily',
 'videoOwnerChannelId': 'UCIi6fq-A7sTT4iBDQUKekyg'}}
```

We get list of all video IDs and their titles as follows.

```
videoIDs = [ video['snippet']['resourceId']['videoId'] for video in videos ]
videotitles = [ video['snippet']['title'] for video in videos ]
print(videoIDs)
```

```
['-dZImvCSPKI', '0vBXkgiJIP8', 'eodNQzJFJpg', 'IiQJ1G4QJWg', 'ycBovk3EtfQ', 'RErsLHdKFSM',
 '5eu_FoJu7uo', 'HUzosM19QCs', 'n4Z3SgEJ4bg', 'P-jkx_XdJlw', 'LFS6RbpzLSw', '0hrtH6sGbtA',
 'bI04JmGVf_k', 'LUU_UKk2YyQ', 'dgtapT4n484', '1cRmNcT1pvo', 'Ga6VEPk_HfY', 'Q2o8bIV6328',
 '-ZLuiE0j8Ts', 'WqTH30vPKxQ', '_9eUuc_-z9s', 'ITGBju0wY4w', 'cQvCq1_Eoms', 'erP8Xc0h00U',
 'mb7p4B2spP0', 'KG4SBzXccEk', 'A9Wh4L7ZJr0', 'CL_cVCZ5l7Q', 'yxP4Nz09rSE']
```

We put this information into a dictionary:

```
yt_dict = []
for row in range(len(videoIDs)):
    d = dict()
    d['youtubeURL'] = 'https://youtu.be/{}'.format(videoIDs[row])
    d['title'] = videotitles[row]
    yt_dict.append(d)
```


It is now child's play to convert the dictionary to a Pandas dataframe:

```
ytDF = pandas.DataFrame(yt_dict)
ytDF
```

	youtubeURL	title
0	https://youtu.be/-dZImvCSPKI	IQC - 1.1 - The Consulting Framework (12:08)
1	https://youtu.be/0vBXkgiJIP8	IQC - 1.2 - Ethical Considerations (16:01)
2	https://youtu.be/eodNQzJFJpg	IQC - 1.3 - Guiding Principles (10:46)
3	https://youtu.be/IiQJ1G4QJWg	IQC - 1.4 - Asking the Right Questions (04:45)
4	https://youtu.be/ycBovk3EtfQ	IQC - 1.5 - The Structure of Data (19:07)
5	https://youtu.be/RErsLHdKFSM	IQC - 1.6 - Quantitative Consulting Workflows ...
6	https://youtu.be/5eu_FoJu7uo	IQC - 1.7 - Roles and Responsibilities (14:24)
...		
21	https://youtu.be/ITGBju0wY4w	IQC - 2.11 - Invoicing (07:25)
22	https://youtu.be/cQvCq1_Eoms	IQC - 2.12 - Closing the File (03:43)
23	https://youtu.be/erP8Xc0h00U	IQC - 3.1 - Lessons Learned: About Clients (19:...
24	https://youtu.be/mb7p4B2spP0	IQC - 3.2 - Lessons Learned: About Consultants...
25	https://youtu.be/KG4SBzXccEk	IQC - 4.1 - The Basics of Business Development...
26	https://youtu.be/A9Wh4L7ZJr0	IQC - 4.2 - Clients and Choices (04:00)
27	https://youtu.be/CL_cVCZ5L7Q	IQC - 4.3 - Building Trust (10:33)
28	https://youtu.be/yxP4Nz09rSE	IQC - 4.4 - Improving Trust (09:04)

16.5 Exercises

In these exercises, use R's *rvest*, Python's *Beautiful Soup*, or any other tool (whether we discussed it or not in the main text) that will allow you to complete the task. You may need to look up various tutorials and examples, and consult documentation, Stack Overflow, and so on.

- Complete the unanswered questions in Sections 28.3.2 (XPath) and 28.3.3 (regex).
- Recreate the web scraping example of Section 28.4.1, this time selecting (or creating) variables that will provide population and area values for all entries in the table (not necessarily variables V11, V12, V13). What changes? What stays the same? Why is that the case?
- Web data is available from a variety of sources, in a variety of formats and languages. Your job is to build a collection of 5 text corpora, each one consisting of documents written in a different language (English, French, Spanish, Italian, and Other). The text documents will be collected from the [New Zealand Government's press releases](#), from Wikipedia, from twitter, from a PDF document, and from other sources. Your final dataset will consist of all of the observations (text) placed in rows, each row associated with a specific language code ("Eng", "Fra", "Esp", "Ita", "Oth").
 - English: the text of all Canadian government press releases published in 2020.
 - French: the text from the (French) Wikipedia entries of all French actresses whose last name starts with "L".
 - Spanish: 700 tweets (total) from @realmadrid, @PaulinaRubio, @Armada_esp + 2 other tweeters of your choice.
 - Italian: the text from Giovannino Guareschi's *Tutto don Camillo* (I racconti del Mondo piccolo) – Volume 1 di 5 (PDF), 1 page per row.
 - Other: 500 other text documents, in other languages that use a Latin-based alphabet.
- Use [Zomato](#) to find which Canadian city has the best sushi restaurants.

5. Build a scraper that automatically collects a multiple-day forecast for all Canadian cities in the database (not only those found on the [landing page](#)), independently of the time at which the scraping takes place.
6. Consider the `parsed_doc` object from the XPath section. What do you think the following blocks of code do?

```
lowerCaseFun <- function(x) {
  x <- tolower(xmlValue(x))
  return(x)
}

XML::xpathSApply(parsed_doc, "//div//i",
  fun = lowerCaseFun)
```

```
dateFun <- function(x) {
  require(stringr)
  date <- xmlGetAttr(node = x, name = "date")
  year <- str_extract(date, "[0-9]{4}")
  return(year)
}

XML::xpathSApply(parsed_doc, "//div", dateFun)
```

7. In the CFL example, the play-by-play data is in separate tables for each quarter. Write a routine that grabs the information and produces a Pandas dataframe for each quarter, with the following headers: ID, away, details, down, home, quarter, time, type, and yard.
8. Modify the YouTube example in order to extract the videos' captions. Clean them using *BeautifulSoup*.
9. Use *twitterR* (or other packages) to build a data frame of tweets related to the *Marvel Cinematic Universe*. Do your tweets mostly originate from Android or iPhone devices? Plot the frequency of tweets against time. Do the same for retweets. Do any patterns emerge?
10. Collect **all Canadian government press releases** for the 2021 calendar year. Identify the date, emanating Department(s), and the number of characters in each release. Are there Departments who release news more frequently than others? Are there Departments whose releases are typically longer than average? What other insights can you draw from your data frame? Repeat this process with [French-language press releases](#).
11. Produce a data frame listing all new products available at [David's Tea](#), the page number where the product was listed, and its price. Remember the scraping do's and don't's!

Chapter References

- [1] [Beautiful Soup Documentation](#).
- [2] [HTML Cheat Sheet](#).
- [3] K. Jarmul. *Natural Language Processing Fundamentals in Python*.
- [4] M. Jones. [15 Fundamental Laws of the Internet](#).
- [5] R. Mitchell. *Web Scraping with Python: Collecting Data From the Modern Web*. 2nd. O'Reilly Media, 2018.
- [6] S. Munzert et al. *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. 2nd. Wiley Publishing, 2015.
- [7] [Selenium Documentation](#).
- [8] [The Selenium Browser Automation Project](#).
- [9] R. Taracha. [Introduction to Web Scraping Using Selenium](#). 2017.

by Aditya Maheshwari

In this chapter, we briefly explain some of the basic concepts that help data scientists go beyond theoretical/small scale projects (mostly used for experiments/local research/conceptual solutions) and introduce the concepts and frameworks that allow data scientists in conjunction with data teams to building data science products that process and deliver results at scale. We will discuss this in the context of exploring the role of data engineering in data projects and providing an overview of some of the types of data pipeline infrastructure commonly involved in these projects.

In the current data ecosystem, most data scientists are still not required to understand the inner workings of data engineering and data management; however, as modeling tools become increasingly automated, and as machine learning solutions move from conceptual to practical, most data project requirements become engineering focused.

We only provide a cursory look at the topic in this chapter; in-depth information is available at [1, 3, 7, 8, 5], while shorter overviews can be found at [6, 2]. Learners interested in database design should also consult [11].

17.1 Background and Context

In the 2010s, the field of data science gained prominence, with an emphasis on creating algorithms to decipher patterns from the vast data generated by digital platforms and technologies with continuous monitoring capabilities.

This marked a notable shift from traditional data analysis methods that primarily focused on smaller datasets in a scientific context. In classic statistical learning, the primary mode of data collection was through surveys, as detailed in Chapter 10, *Survey Sampling Methods*. It also included methods not directly connected to user activities, such as post-interaction surveys, evident with the post-visit survey of the *Canada Revenue Agency's My Account* service.¹

These methodologies posed theoretical challenges, especially with handling modest sample sizes. However, the objective was clear: under a set of assumptions, can we determine any correlation between variables (features) and actions (outcomes)?

Historically, research often relied on fragmented or rudimentary systems. These were mostly adequate for routine, automated, or substantial tasks

17.1 Background and Context .	1065
17.2 Data Engineering	1067
Data Pipelines	1068
Automatic Deployment .	1073
Scheduled Pipelines . . .	1075
Data Engineering Tools . .	1077
17.3 Data Management	1079
Databases	1079
Database Modeling	1082
Data Storage	1084
17.4 Reporting and Deployment	1086
Reports and Products . . .	1086
Cloud vs. On-Premise . .	1087
Chapter References	1088

1: Such approaches have their merits but can feel detached from real-time user actions.

2: Relying solely on such systems not only constitutes poor practice but may also risk task failures from technical glitches.

using real-time datasets. However, this isolated approach wasn't always ideal due to the risk of technical issues.²

Today, with digital platforms' proliferation, the volume of accessible data is unparalleled. These platforms can record every user interaction. Consider a **cross-sectional dataset** that captures phrases spoken by a user at home through devices like Amazon's *Alexa*, several Google searches over days, frequent product views across websites, and the ensuing transaction records.

Instead of selective data gathering, every interaction is catalogued. Beyond the ethical concerns surrounding such comprehensive use of personal data (as highlighted in Section 14.3, *Ethics in the Data Science Context*), there are technical challenges, such as processing massive data and deriving meaningful insights from it.

Data inquiries now predominantly fall into **reporting** ("what occurred?"), **real-time analytics** ("what's transpiring now?"), and **predictive modelling** ("what might unfold?"), as opposed to **causal inference** ("why did it happen?").

A significant challenge for data scientists today is to format these vast data repositories to be algorithm-friendly. As a result, a key focus of modern **data engineering**, as discussed in subsequent sections, pertains to the processing of this ever-growing data influx.

Once data is appropriately organized, data scientists deploy machine learning techniques to develop **proofs-of-concept**. Subsequently, AI/ML engineers transform these into **deployable models** as part of **data pipelines**, encapsulating the broader domain of data engineering.

Though data and AI/ML engineering have been around for a while, the advent of **cloud computing** places a heightened emphasis on their importance, sometimes overshadowing data science in specific sectors.

Organizations with **low data maturity** often lean on software like Excel to craft makeshift solutions for standard data pipeline tasks.³ Such makeshift systems might suffice for their immediate needs but are insufficient when dealing with expansive datasets.

In contrast, entities with **enhanced data maturity** use a mix of SQL warehouse queries and R/Python scripts. They aggregate data using the entire population for reporting and then sample to build proofs-of-concept on local systems. However, even these methods don't exploit the full potential offered by contemporary tools and **data stacks**.

At its core, data engineering aims at collecting, storing, and analyzing data **at scale**.⁴

Investing in data engineering components is invaluable for such extensive operations, a topic explored further below.

In smaller enterprises, roles in data engineering and data science may overlap, especially if the company's needs tilt more towards data engineering. Conversely, many larger companies maintain **dedicated** data engineers, responsible for managing **data pipelines** and overseeing **data warehouses**.

3: We could easily join those criticizing these rudimentary methods, and while we generally concur, we aren't suggesting Excel should NEVER be used. It has its place, albeit limited.

4: Scalability refers to a system's capability to handle a growing workload efficiently or its potential to expand to accommodate that growth.

17.2 Data Engineering

Data engineering is best understood as a subset of computer engineering that emphasizes designing, constructing, and maintaining systems specifically geared towards data handling – from collection and ingestion to analysis and presentation. Given its roots in computer engineering, a grasp of certain computer engineering fundamentals can be beneficial when diving into data engineering.

At their core, computers comprise:

- **memory**, which is visualized as labeled containers holding binary data (ones and zeros), and
- **circuits**, systems that process memory content as input to produce outputs, which are then stored back in memory.

A computer processor's array of circuits constitutes its **instruction set**. When executing a computer program, these instructions are followed in a specific sequence.

In this frame of reference, data represents distinct **patterns** in memory. These patterns can be:

- **duplicated** into other memory locations;
- **relocated** through copying followed by erasure of the original, and/or
- **altered** using the patterns as inputs for a series of instructions, resulting in new patterns stored back in memory.

Notably, computer programs also manifest as data in memory. They're loaded into the processor, translated into basic hard-coded instructions, and then actualize the program's directives.

Software engineering spotlights the software side of computers. As defined by IEEE: "The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software" [4]. Essentially, computer engineering's objective is to create programs that manipulate binary patterns via suitable instruction sets.

Data Engineering and IT How does **information technology** (IT) relate to this? Generally, IT revolves around technology that manipulates data and information, extending beyond just computer systems to encompass communication systems and even television. Thus, computer and data engineering can be perceived as IT subdomains.

However, colloquially, IT often denotes the use of **pre-existing software and hardware for data and information management**. IT professionals amalgamate these technologies to formulate comprehensive systems with specific information processing capabilities.

Data engineers, in this landscape, traverse the domains of software engineers and IT professionals. They might design **bespoke software applications** and **tailored architectures** for unique data types, but typically within the scope of **using established applications to construct data pipeline infrastructures**.

In the modern age, awash with data, data engineering's relevance has exploded, with applications spanning nearly every sector. Organizations, flush with vast data troves, are investing in the right talent and tools to refine this raw data, prepping it for data scientists and analysts.

Data Team Roles Within a data team, data engineers enable streamlined data collection from varied sources. Subsequently, database analysts manage this data, priming it for analytical tasks and inclusion in data projects. Let's delve deeper into these roles (for a comparison, refer to Section 13.1.3, *Roles and Responsibilities*).

5: From sources like paper tax forms manually fed into databases, or real-time data from online tax platforms that's streamed into databases.

Data engineers obtain data.⁵ They organize, disperse, and store this data in data lakes and warehouses. Moreover, they craft tools and data models, aiding data scientists in data querying.

Data scientists use the data curated by data engineers to extract insights, build prototype predictive models, evaluate and enhance outcomes, and construct data models. Typically, they employ languages like Python or R and work within analytical notebooks such as RMarkdown or Jupyter. These notebooks interface with clusters, converting queries into commands for big data platforms (like Apache Spark).

ML engineers implement and deploy data models, acting as a bridge between data engineers and scientists. They take prototype ideas and upscale them, establishing feedback mechanisms to allow data scientists to monitor aggregate performance and rectify any issues in their prototype solutions.

See [10] for another perspective on these roles, in particular as they relate to the job market.

17.2.1 Data Pipelines

The work of data engineers largely revolves around data pipelines, which conduct a series of routine data manipulation tasks in an automated manner. Intriguingly, their work shares similarities with chemical systems and process engineers, but with a focus on data transformation rather than chemical transformation.

Here are typical tasks for data pipelines:

1. **acquiring** datasets that match business requirements;
2. **developing** algorithms to convert data into actionable insights;
3. **building, testing, and maintaining** database pipeline structures;
4. **collaborating** with management to grasp company goals;
5. **creating** new data validation techniques and data analysis tools;
6. **ensuring** adherence to data governance and security policies.

A functional data pipeline consists of **interfaces** and **mechanisms** that aid in information flow. Data engineers establish and manage this **data infrastructure**, using it to ready data for analysis by data analysts and scientists. It also delivers the outcomes of this data transformation and analysis to the end users.

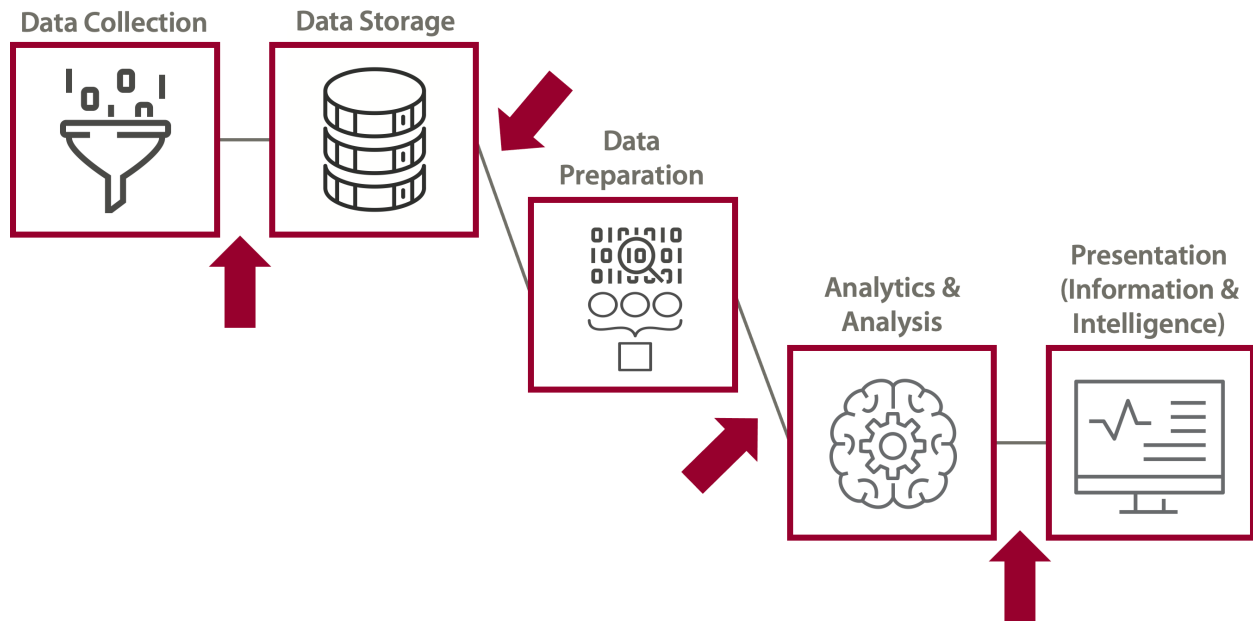


Figure 17.1: Illustration of a conceptual data pipeline highlighting the components and transitions.

Data can originate from myriad sources in diverse formats and sizes. Converting this vast amount of raw data into a usable format for data scientists is termed **building a data pipeline**.

Generally, pipelines encompass the following stages.

1. **ingestion**: collecting data from various sources;
2. **processing**: cleansing and transferring the data to a suitable data storage;
3. **storage**: placing the data in a reachable location and crafting a data model; and
4. **access**: facilitating access to cleaned data for analysis and display.

While the number and sequence of steps can differ across frameworks, they must remain consistent within a particular program.

A primary challenge for data engineers is crafting a pipeline that can **operate in near-real-time when prompted**, providing users with **current information** swiftly.⁶

6: Considering the dataset size.

Data engineers typically start by designing a prototype of a functional pipeline. Once tested, a more resilient pipeline is devised, which then undergoes **deployment** and **production**.

Related tasks include:

- checking data quality;
- enhancing query performance;
- forming a continuous integration and delivery environment;
- aggregating and storing data from various sources following a specific data model; and
- implementing machine learning and data science methods on distributed systems.

Consider this scenario: The *Canada Revenue Agency* (CRA) aims to determine the number of individuals in a region not submitting tax returns

7: After submitting, the benefits received exceed the taxes paid.

and, as a result, forgoing net positive benefits.⁷ They also want to ascertain who among them possibly missed the deadline due to unawareness rather than oversight.

Potential pipeline processes include:

- gathering and storing data from third-party reports indicating tax filing numbers in the region;
- assessing historical tax filing data for that region;
- using predictive modeling to assess known non-filers' characteristics to predict unintentional filing oversights;
- presenting results via a dashboard.

Data Pipeline Connections In our framework, the links between pipeline components facilitate:

- transition from **collection** methods to an effective **storage** area;
- movement from **storage** to **preparation** where data undergoes transformation;
- transfer of **transformed** data to **analysis** or **modeling** phases; and
- use of **modeling** outcomes for **presentation**.

Typical challenges are:

- transferring data into a **data lake** can be time-consuming, especially with repeated data ingestion tasks;
- data platforms are evolving, leading to a cycle of building, maintaining, then **rebuilding** and **continuous maintenance**;
- the growing need for real-time data means **low latency** pipelines⁸ become essential, making *Service Level Agreements* (SLAs) harder to establish.⁹

8: Those with minor delays.

9: Data pipeline SLAs detail client-service provider agreements integrated into client pipelines. Such SLAs necessitate regular performance checks and tuning.

Without careful planning, these challenges can quickly escalate.

Data Pipeline Operations A data pipeline is essentially an **automated sequence of data operations**. This can range from simple tasks, like moving data from one place to another, to more intricate processes that aggregate, analyze, and present data.

Common elements for each step include:

- **data sources** – applications, mobile apps, microservices, and more;
- **data integration** – ETL, stream data integration, and the like;
- **data storage** – MDM, data lakes, warehouses, etc.;
- **data analysis** – machine learning, AI, predictive analytics, etc.;
- **delivery and presentation** – dashboards, reports, notifications, and more.

Furthermore, pipelines allow breaking down a large task into manageable steps, optimizing each phase.¹⁰ For example, it might be beneficial to use a specific language or framework for a pipeline segment. With a monolithic script, all processes, from data collection to presentation, would need to use the same tool, which might not be optimal.

A more efficient strategy, adopted by most **data pipeline tools**, is to choose the best framework for each component.

10: This enhances efficiency, scalability, and reusability.

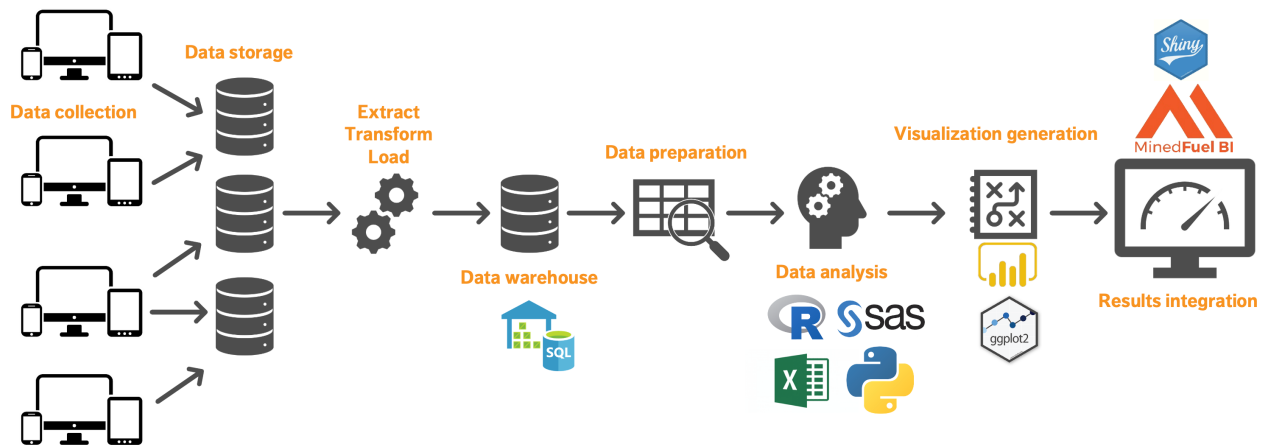


Figure 17.2: A depiction of a data visualization pipeline showcasing different component choices.

ETL Framework In the realm of data pipeline design, one cannot overlook the significance of the **ETL framework**, an acronym for **Extracting**, **Transforming**, and **Loading** data. While ETL has its origins in the earlier domains of **data marts** and **warehouses**, its principles remain indispensable in shaping modern-day data pipelines.

All data processes start with the **extraction** of data from a given source or temporary holding. When data comes from multiple sources, a subsequent **loading** phase often follows the extraction. This step ensures that multiple systems and processes uniformly handle the data from the same extraction point. In scenarios where data from multiple sources “converge”, centralizing this data before its **transformation** proves beneficial.¹¹

Once data from all pertinent sources is assembled, the onus falls on the data engineer to strategize the most effective way to merge these datasets. This strategy often encompasses the creation of data pipeline components that facilitate a seamless flow of data from the source systems to a format amenable for querying by business intelligence tools.

A pivotal role of data engineers lies in assuring the reliability and correctness of these pipelines. Often, this involves reconciling data or even deploying supplementary pipelines to corroborate against the original data sources. They are also tasked with ensuring a consistent and updated data flow, an endeavour often supported by various monitoring solutions and **site reliability engineering** (SRE) methodologies.

User-centric products typically source data from a diverse range of origins, potentially spanning multiple systems or third-party **integrations**. It’s imperative that such data aligns with end-user specifications and maintains its **integrity**. However, challenges arise when data sourcing relies on inter-dependent systems. Each execution of the pipeline mandates fresh queries to these systems, potentially extending the pipeline’s runtime. Even with such intricacies, the ETL framework can often render subsequent data pipelines more efficient.

Timely access to updated information is a frequent demand from business units. In navigating this, data engineers must holistically evaluate pipeline performance. This evaluation must account for the frequency of new data

11: Depending on specific requirements, post-transformation, this data might either be transferred to a different location like a **data warehouse** or undergo transformations right at the source before a final load.

inflows, the duration of transformation phases, and the time required to update the final data storage location.

Data Architecture Optimal results in data management often hinge on a shared understanding of the organizational structure of the data and its flow mechanisms. Serving as a blueprint for these aspects is the **data architecture**, which ideally encompasses the following notions.

- **Storage layout:** an overview of the methodology and locations of data storage. This should factor in the standards pertaining to file paths, file type specifics for file/object storage, as well as naming conventions for databases, schemas, views, and tables in the context of database storage.
- **Data landscape:** a depiction of how data is categorized within a specific repository.
- **Data abstractions:** a comprehensive elucidation of the components of any data abstractions present on the platform, complete with illustrative diagrams.¹²
- **Data access:** an outline detailing the authorization mechanisms of the data repository, capturing nuances like user-role mapping, the structure of role hierarchies, and privilege allocations to these roles.

12: These serve as foundational blueprints for crafting various elements on the data platform.

Additionally, a well-structured data architecture clearly outlines how data travels between different repositories within the platform. Such insights answer pivotal questions about permissible sources and destinations for data and the tools enlisted for these transfers.

13: This pertains to a comprehensive data management strategy that empowers an organization to uphold superior data quality throughout its entire lifecycle. It encompasses controls that align with business goals. Primary areas of focus include data availability, usability, consistency, integrity, security, and regulatory compliance. It mandates protocols for upholding data quality across an organization, holding entities accountable for discrepancies and ensuring a universal data access paradigm. [12]

Data Governance and Self-Serviceability **Data governance**¹³ is a non-negotiable aspect; while data platforms are envisioned to catalyze innovation by democratizing data access, discernment is required in determining access levels, especially with sensitive datasets.

Specific datasets, especially from realms like sales and HR, often house sensitive information necessitating restricted access. Further, data sets comprising customer or health-related data are often tethered to **compliance protocols**, dictating access and usage parameters.

For users aiming to access a dataset on the platform, there should be robust mechanisms in place to petition for access. Concurrently, there should be workflows to review these petitions, ensuring that data access remains judicious and controlled. This ethos of **self-serviceability** is instrumental in liberating data access on any platform.

Further deepening the canvas of self-serviceability is the capability for users to request the instantiation of new structures or objects within the data repository. As evolving use cases emerge, the creation of new "workspaces" allows data engineering teams to devise new data transformations and datasets. This is but a glimpse into the myriad scenarios where self-serviceability amplifies the autonomy and dynamism of the data platform.

17.2.2 Automatic Deployment and Operations

Automating data pipelines can range from simple tasks like directing data between locations to intricate operations such as automated aggregation, transformation, and redistribution of data from various sources.

It's become increasingly viable to automate the ingestion of petabytes of ever-evolving data. This enables pipelines to efficiently provide data suitable for analytics, data science, and machine learning.

Notable **automated operations** include:

- **on-the-fly data processing** from files or real-time sources like Kafka, DBMS, NoSQL, and others;
- **automatic detection of schema** (or column) changes across different data formats;
- real-time tracking of incoming data;
- implementing **auto-backup measures** to prevent data loss.

As referenced earlier, ETL provides essential **decision points** for data pipeline creation. With contemporary tools, data engineers can minimize development duration, concentrating on business logic and data quality checks using SQL, Python, R, and similar. Achievable actions include:

- **intent-focused declarative development**, clarifying the problem, and simplifying the solution;
- automatic generation of **detailed data lineage** and handling table dependencies **within the pipeline**;
- auto-checking for **missing components**, **syntax anomalies**, and ensuring **data pipeline recovery**.

To enhance **data reliability**, we can:

- establish **data quality and integrity measures** within the pipeline;
- address **data discrepancies** using **pre-set policies** such as alerts, quarantine, or dropping faulty data;
- use **metrics** that continuously monitor, log, and report on data quality throughout the pipeline.

Strategies for Automated Pipeline Deployment Traditional software deployment often involved:

- initiating a build;
- manually transferring the build to a production server, and
- an ad-hoc “test” to verify application functionality.

This approach is neither scalable nor efficient, with manual steps increasing risk. The ultimate aim of automated pipeline design and deployment is to prototype and validate scalable components **before their full deployment**, while supporting an ongoing development cycle. Agile methodologies align well here.¹⁴

14: See [What Is Agile? And When to Use It](#)  for an overview.

15: This not only validates the logic but ensures the code operates as anticipated.

16: Ensuring seamless interplay between systems is the primary objective of this test layer.

17: "Blue-green deployment" is a strategy used for releasing applications by having two separate environments – a "blue" one and a "green" one:

- the **blue environment** is the currently running production environment, serving all the user traffic;
- the **green environment** is a clone of the blue environment, to which updates or new versions are deployed.

Once the green environment is ready and fully tested, the traffic is switched from the blue environment to the green environment, making the transition seamless to users. This approach is popular because it allows deployments with no downtime/service interruption, and because if something goes wrong in the green environment after the switch, traffic can be quickly rerouted back to the blue environment, ensuring high availability and minimizing disruptions. Note that there may be issues with data synchronization, especially for databases. Additionally, since two environments are running concurrently (at least during deployment), the infrastructure costs can double (albeit temporarily).

18: RTO represents how long the system or application can be down before there's a significant impact on the organization; RPO represents how much data an organization can afford to lose in the event of an incident.

19: Defining and managing servers, databases, networks, and other infrastructure components through code, rather than manually.

Effective Testing Practices Deploying in a live environment without exhaustive testing can expose end-users to unresolved bugs or issues. Optimal **code promotion** practices involve automated verification processes checking code functionality across varied scenarios:

- **unit tests** examine code segments, verifying if, given specific inputs, they yield expected outputs without depending on external code;¹⁵
- **integration testing** verifies that multiple code segments cohesively function and produce anticipated results.¹⁶

Combining both testing methods with modern strategies like **blue-green deployments** drastically reduces potential disruptions when introducing new code.¹⁷

Disaster Recovery Protocols Before advancing changes to a system, it's imperative to pass them through rigorous testing. Furthermore, a contingency plan for **system failure** is vital. Systems should be robust against catastrophic failures. Typical metrics in data engineering for disaster recovery include **Recovery Time Objective (RTO)** and **Recovery Point Objective (RPO)**.¹⁸

During disaster recovery, it's essential to gauge the impact on consumers and system downtime. Data engineers are tasked with ensuring that data pipelines and storage solutions comply with acceptable recovery benchmarks.

Guidelines for Effective Pipeline Development With the influx of data into platforms, adopting **development best practices** is crucial to guarantee reliability, especially with the dynamic nature of data platforms. Standard practices encompass:

- leveraging **Source Code Management (SCM)** utilities;
- using **Continuous Integration (CI)/Continuous Delivery (CD)**;
- designing using **diverse deployment settings**;
- prioritizing testing and data quality;
- embracing **Infrastructure as Code (IaC)**;¹⁹
- using **database change control**;
- formulating effective **rollback strategies**; and
- constant monitoring and alert mechanisms.

The guiding principles revolve around **automation**, **testing**, and **monitoring**:

- streamlining the construction and validation of digital components;
- forming deployment pipelines to distribute these components;
- evaluating deployments and advancing components through different stages; and
- incorporating data quality assessments in pipelines, and raising flags on inconsistencies.

When tests detect issues, automated rollback procedures should initiate. Given the constantly shifting landscape of data governance standards, tooling, best practices, security measures, and business requirements, deployments must be both **automated** and **verifiable**.

17.2.3 Scheduled Pipelines and Workflows

There are three primary **data pipeline architectures** that cater to the automated scheduling of tasks and workflows:

- **batch data pipelines** transfer vast data amounts at designated intervals;²⁰ ;
- **streaming data pipelines** transfer data from its origin to its destination immediately upon being generated,²¹ and
- **change capture data pipelines**, whose role is to renew large datasets and uphold data uniformity across platforms.²²

Blueprint for Efficient Pipelines Conceptually, constructing an **efficient** data pipeline is a systematic six-phase endeavour:

1. **cataloging and overseeing data** entails facilitating enterprise-wide access to trustworthy and compliant data;
2. **proficient data ingestion** involves drawing data from myriad sources – on-premises databases, SaaS applications, IoT devices, streaming apps – and channeling it into a cloud-centric data lake;
3. **data fusion** cleans, enriches, and remodels the data – the creation of specific zones, such as landing areas, enrichment hubs, and enterprise territories, is integral here;
4. **implementation of data quality protocols** ensures data purity and organizational distribution, bolstering DataOps;²³
5. **data refinement** prepares the sanitized data to be migrated to a cloud data warehouse, thus allowing for self-driven analytics and data science scenarios, and
6. **real-time data processing** ensures that insights are gleaned from real-time data sources, like Kafka, and subsequently channeled to a cloud data warehouse for analytical use.

To support ML/AI and process big data at reasonable service level objectives, an efficient pipeline should also:

- **seamlessly deploy and process** any data on any cloud ecosystem, such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and Snowflake for both batch and real-time processing;
- **efficiently ingest data** from any source,²⁴ into any target, such as cloud data warehouses and data lakes;
- **detect schema drift** in relational data base management systems (RDBMS) schema [11] in the source database or a modification to a table,²⁵ and **automatically replicate** the target changes in real time for data synchronization and real-time analytics use cases;
- **provide a simple wizard-based interface** with no hand coding for a unified experience;
- **incorporate automation and intelligence capabilities** such as auto-tuning, auto-provisioning, and auto-scaling to design time and run-time, and
- **deploy in a fully managed advanced server-less environment** for improving productivity and operational efficiency.

20: Prevalent in scenarios where tables require daily or weekly updates for reporting or dashboard functionality.

21: They often fill data lakes, serve data warehouse integration, or disseminate data for real-time uses, such as stock price updates or on-the-spot fraud detection.

22: Pivotal when datasets are shared among multiple systems.

23: DataOps, starting as best practices, has evolved into a comprehensive data analytics methodology. It addresses the entire data life cycle, fostering collaboration between data analytics groups and IT functionalities. [13]

24: Such as legacy on-premises systems, databases, change data capture (CDC) sources, applications, or IoT sources.

25: Such as adding a column or modifying a column size.

Assessing Pipeline Performance and SLO A pivotal performance metric is the pipeline's alignment with business prerequisites. **Service level objectives** (SLOs) offer concrete performance benchmarks against set standards.

For instance, a system could have the following SLO framework:

- **data timeliness** – 90% of product advice should stem from user online activity within the last three minutes;
- **data accuracy** – fewer than 0.5% of monthly client bills should have inaccuracies;
- **data isolation and resource allocation** – within a workday, priority payments should be processed within 10 minutes of submission, with standard ones being settled by the subsequent business day.

Data freshness relates to data's relevance in regards to its age. Typical SLOs for data freshness encompass:

- **$x\%$ of data processed within y time units [sec, min, days]** – this is commonly used for batch pipelines that process bounded data sources; the metrics are the input and output data sizes at key processing steps relative to the elapsed pipeline run-time; we may choose a step that reads an input dataset and another step that processes each item of the input;
- **oldest data shouldn't exceed y time units [sec, min, days]** – this is commonly used for streaming pipelines that process data from unbounded sources; the metrics indicate how long the pipeline takes to process data, such as the age of the oldest unprocessed item,²⁶ or the age of the most recently processed item;
- **pipeline task completion within y time units [sec, min, days]** – this sets a deadline for successful completion and is commonly used for batch pipelines that process data from bounded data sources; it requires the total pipeline-elapsed time and job-completion status, in addition to other signals that indicate the success of the job.²⁷

26: That is, how long an unprocessed item has been in the queue.

27: For example, the percentage of processed elements that result in errors.

28: One challenge is that reference data for validating correctness might not always be available. Therefore, there might be a need to generate reference data using automated tools, or even manually.

Data correctness refers to data being free of errors. We can determine data correctness through different means. One method is to check whether the data is consistent by using a set of validation rules, such as rules that use regular expressions (regexps). Another method is to have a domain expert verify that the data is correct, perhaps by checking it against reference data.²⁸ These reference datasets can then be stored and used for different pipeline tests.

With reference datasets, we can verify data correctness in the following contexts:

- **unit and integration tests**, which are automated through continuous integration;
- **end-to-end pipeline tests**, which can be executed in a pre-production environment after the pipeline has successfully passed unit and integration tests, and is automated *via* continuous delivery, and/or
- **pipelines running in production**, when using monitoring to observe metrics related to data correctness.

For running pipelines, defining a data correctness target usually involves measuring correctness over a period of time, such as:

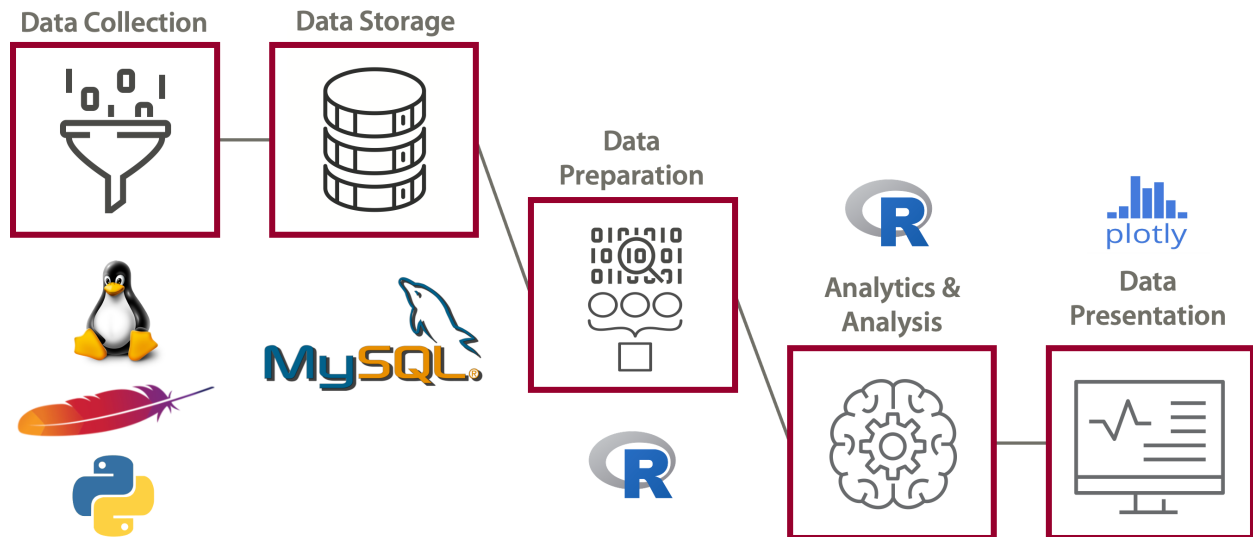


Figure 17.3: An open-source data analysis pipeline.

- **on a per-job basis, fewer than $x\%$ of input items contain data errors** – this SLO can be used to measure data correctness for batch pipelines;²⁹
- **over an y -minute moving window, fewer than $x\%$ of input items contain data errors** – this SLO can be used to measure data correctness for streaming pipelines.³⁰

To measure these SLO, we can use metrics over a suitable period of time to accumulate the number of errors by type, such as the data being incorrect due to a malformed schema, or the data being outside a valid range.

29: As an example, consider: “For each daily batch job to process electricity meter readings, fewer than 3% of readings contain data entry errors”.

30: As an example, consider: “Fewer than 2% of electricity meter readings over the last hour contain negative values.”

17.2.4 Data Engineering Tools

While it is unlikely that any one data engineer could achieve mastery over all possible data engineering tools, it would be beneficial for data teams to have competencies in a fair number of the following:³¹

- **analytical databases** (Big Query, Redshift, Synapse, etc.)
- **ETL** (Spark, Databricks, DataFlow, DataPrep, etc.)
- **scalable compute engines** (GKE, AKS, EC2, DataProc, etc.)
- **process orchestration** (AirFlow / Cloud Composer, Bat, Azure Data Factory, etc.)
- **platform deployment and scaling** (Terraform, custom tools, etc.)
- **visualization tools** (Power BI, Tableau, Google Data Studio, D3.js, ggplot2, etc.)
- **programming** (tidyverse, numpy, pandas, matplotlib, scikit-learn, scipy, Spark, Scala, Java, SQL, T-SQL, H-SQL, PL/SQL, etc.)

31: The content of this section is highly time-sensitive and is liable to have changed completely within 1-2 years from publication. That’s life in the fast data engineering lane for you!

Here are some currently popular pipeline tools [2].

1. **Luigi** (Spotify) builds long-running pipelines (thousands of tasks stretching across days or weeks); it is a Python module available on an open-source license under Apache. It addresses the “plumbing” issues typically associated with long-running batch processes,

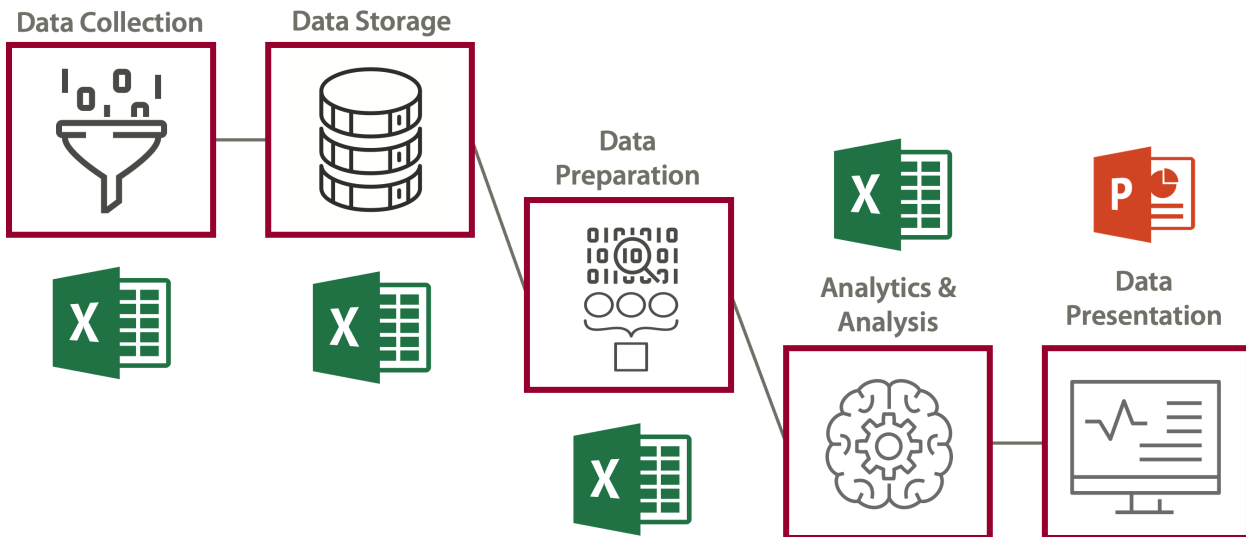


Figure 17.4: An unfortunately still far-too-common data analysis pipeline.

32: Luigi uses 3 steps to build pipelines: `requires()` defines the dependencies between the tasks, `output()` defines the target of the task, and `run()` defines the computation performed by each task. Luigi tasks are intricately connected with the data that feeds into them, making it difficult to create, modify, and test a single task, but relatively easy to string tasks together.

33: Airflow defines workflows as Directed Acyclic Graphs (DAG), and tasks are instantiated dynamically. Airflow is built around: **hooks** (high-level interfaces for connections to external platforms), **operators** (predefined tasks that become DAG nodes), **executors** (run jobs remotely, handle message queuing, and decide which worker will execute each task), and **schedulers** (trigger scheduled workflows and submit tasks to the executors).

where many tasks need to be chained together (Hadoop jobs, dumping data to/from databases, running machine learning algorithms, etc.).³²

2. **Airflow** (AirBnB) is used to build, monitor, and retrofit data pipelines. It is a very general system, capable of handling flows for a variety of tools and highly complex pipelines; it is good tool for pipeline orchestration and monitoring. It connects well with other systems (databases, Spark, Kubernetes, etc.).³³
3. **scikit-learn pipelines**: scikit-learn pipelines are not used to orchestrate big tasks from different services; rather they help make code cleaner and easier to reproduce/re-use. They are found in scikit-learn, a popular Python data science module. The pipelines allow users to concatenate a series of **transformers**, followed by a final **estimator**; this is useful for model training and data processing, for instance. With scikit-learn pipelines, data science workflows are easy to read and understand, which also makes it easier to spot issues such as **data leakage** (unplanned or unauthorized release of data). The pipelines only work with scikit-learn transformers and estimators, however, and they must all be run within the same run-time, which makes it impossible to run different pipeline parts on different worker nodes while keeping a single control point.
4. **Pandas (Python) or Tidyverse (R) Pipes**: pandas and the tidyverse are popular data analysis and manipulation libraries. When data analysis becomes very sophisticated, the underlying code tends to become messier. Pandas and tidyverse pipes keep the code clean by allowing users to concatenate multiple tasks using a simple framework, similar to scikit-learn pipelines. These pipes have one criterion, the “data frame in, data frame out” principle: every step consists of a function with a **data frame** and other parameters as arguments, and a data frame as output. Users can add as many steps as needed to the pipe, as long as the criterion is satisfied.

17.3 Data Management

As covered in the previous section, a major element of data engineering (that is, developing data pipelines) involves moving data from storage to storage as it is processed. In this sense, data storage can be viewed as the metaphorical heart of the data pipeline, while the machine learning model components of the pipeline could be thought of as the brains. In this section we will focus on this metaphorical heart, and consider management of the brains (analytics and machine learning models) in Section 17.4, *Reporting and Deployment*.

Computers have advanced significantly in their ability to store large amounts of data. In this section, we will cover **databases**, **data modeling**, and **data storage**. Readers are invited to refer to [11] (and Section 14.5, *Getting Insight From Data*) for more details.

17.3.1 Databases

Historically, computers relied on a **file-based system** (i.e., they manipulate data files). File-based systems face a number of shortcomings:

1. **data redundancy**: files and applications are created by different programmers from various departments over long periods of time. This can lead to redundancy, a situation that occurs in a database when a field needs to be updated in more than one table, inconsistencies in data format, the same info being stored in multiple files, and conflicting copies;
2. **data isolation**: it can prove difficult for new applications to retrieve the appropriate data, which might be stored in various files;
3. **data integrity**: maintenance may be required to ensure that data in a database are correct and consistent;
4. **security**: it can be difficult to enforce access constraints (if needed) when application requirements are added to the system in an *ad-hoc* manner, and
5. **concurrency**: if multiple users access the same file at the same time, there can be issues with file locking.

Spreadsheets were originally designed for a single user, which is reflected in their characteristics. They are adequate for single users or for small teams of users who have no need for complicated data manipulations.

Databases, on the other hand, hold massive amounts of information, and allow multiple concurrent users to quickly and securely access and query data using highly complex logic and language. They only need to be defined once before being accessed by various users.

Databases They consist of a representation of some aspect of the real world, in the form of a collection of **data elements** representing **real world information**.³⁴ They are:

- logical, coherent, and internally consistent;
- designed, built, and populated with data for a specific purpose;
- made up of data items, which are stored in fields,
- populated with tables, which are combinations of fields.

34: Most databases use a **structured query language** (SQL) for writing and querying data. SQL statements include: create/-drop/alter table; select, insert, update, delete; where, like, order by, group by, count, having; join.

A **database management system** (DBMS) is a collection of programs that enables users to create and maintain databases and control all access to them. The primary goal of a DBMS is to provide an environment for users to retrieve and store information in a convenient and efficient manner.

Data management is “simply” care-taking for the data so that it works for its users and remains useful for tasks. Managing information using a database allows data scientists to become strategic users of the data at their disposal. The processing power in a database can be used to manipulate the data it houses, namely: sort, match, link, aggregate, filter, compute contents, etc. Because of the versatility of databases, we find them powering all sorts of projects.

Database Benefits While databases might be overkill for small datasets, they have many benefits (especially for larger projects):

1. **self-describing nature of a database system:** a database contains the data and the metadata, which describes and defines relationships between tables in the database. This separation of data and information about the data makes a database system entirely different from the traditional file-based system in which the data definition is part of the application program;
2. **insulation between program and data** (also called program-data independence): in a file-based system, the structure of data files is defined in the application programs, so if a user wants to change the structure of a file, all programs that access it need to be changed as well. In a database system, the data structure is stored in the system catalogue and not in the programs. Therefore, one change (such as adding a new variable) is all that is needed to change the structure of a file;
3. **support for multiple views:** a database supports multiple views, or subsets, of the database. Each view contains data that is only of interest to the group of users subscribed to the particular view;
4. **sharing of data and multi-users:** many users can access data at the same time, through features called concurrency control strategies. The design of model multi-user database systems is a great improvement from those in the past which restricted usage to one user at a time,
5. **control of redundancy:** ideally, each data item is only found in one location, but redundancy can sometimes improve query performance (even though it should be kept to a minimum wherever possible).

Types of Databases Databases come in various flavours:

- the most common (as of 2022) are **relational databases**, in which data items are organized as a set of tables with columns and rows;
- data in **object-oriented databases** is represented in the form of objects, as in object-oriented programming (OOP);³⁵
- in **distributed databases**, two or more files are located in different sites – such databases may be stored on multiple computers located in the same physical location, or scattered over different networks, etc.;

35: We discuss OOP briefly in Chapter 1, but there is a lot more to be said on the topic.

- **data warehouses** are central repository for data, designed specifically for fast query and analysis;
- **NoSQL warehouses** are non-relational databases that allow for unstructured and semi-structured data to be stored and manipulated (in contrast with relational databases which define how all the data inserted into the database must be composed) – NoSQL has grown popular as web apps have become more common and more complex,
- **graph databases** store data in terms of entities and relationships between entities – for instance, online transaction processing (OLTP) databases are speedy analytic databases designed for large numbers of transactions performed by multiple users.

Database Challenges Today's large enterprise databases often support very complex queries and are expected to deliver nearly instant responses to those queries. As a result, database administrators are constantly called upon to employ a wide variety of methods to help improve performance and overcome some common database challenges.

- **Absorbing significant increases in data volume:** the explosion of data coming in from sensors, connected machines, and dozens of other sources keeps database administrators scrambling to manage and organize their companies' data efficiently;
- **ensuring data security:** data breaches are happening at an ever-increasing rate, and hackers are getting more and more inventive – it is more important than ever to ensure that data is secure ... yet also easily accessible to users;
- **keeping up with demand:** in today's fast-moving business environment, companies need real-time access to their data to support timely decision-making and to take advantage of new opportunities;
- **managing and maintaining the database and infrastructure:** database administrators must continually watch the database for problems and perform preventative maintenance, as well as apply software upgrades and patches; as databases become more complex and data volumes grow, companies are faced with the expense of hiring additional talent to monitor and tune their databases;
- **removing limits on scalability:** some claim that businesses need to grow if they are going to survive, and so must their data management; but it is nearly impossible for database administrators to predict how much capacity the company will need, particularly with on-premises databases,
- **ensuring data residency, data sovereignty, or latency requirements:** some organizations have use cases that are better suited to run on-premises; in those cases, engineered systems that are pre-configured and pre-optimized for running the database are ideal.

Addressing all of these challenges can be time-consuming and can prevent database administrators from performing more strategic functions.

17.3.2 Database Modeling

Database modeling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems. This includes both data elements and structures/relationships between them.

1. Requirements are put into a **conceptual model** (tech independent specifications), which describes the semantics of a domain and the scope of the model. For example, a model of the interest area of an organization or industry. This consists of entity classes, representing the kinds of things of significance in the domain, and relationship assertions about associations between pairs of entity classes. A conceptual schema specifies the kinds of facts or propositions that can be expressed using the model. In that sense, it defines the allowed expressions in an artificial 'language' with a scope that is limited by the scope of the model.
2. The structure of the database data is put into a **logical model**, which describes the model semantics, as represented by a particular data manipulation technology. This consists of descriptions of tables and columns, object-oriented classes, and XML tags, among other things. The implementation of a single conceptual model may require multiple logical models. The logical models are then incorporated into a physical data model that organizes data into tables, which accounts for access, performance, and storage details.
3. The **physical data model** describes the physical means by which data is stored, including partitions, CPUs, tablespaces, and the like.

A database model, then, is a specification describing how a database is structured and used.

- The **flat (table) model** may not strictly qualify as a data model; it consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be roughly similar values, and all members of a row are assumed to be related to one another.
- The **network model** organizes data using two fundamental constructs: the records and the sets. Records contain fields, and sets define one-to-many relationships between records: one owner, many members. The network data model is an abstraction of the design concept used in the implementation of databases.
- The **hierarchical model** is similar to the network model except that links in the hierarchical model form a tree structure, while the network model allows arbitrary graphs.
- The **relational model** is a database model based on first-order predicate logic. Its core idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values. The power of the relational data model lies in its mathematical foundations and its simple user-level paradigm.
- The **object-relational model** is similar to a relational database model, but objects, classes and inheritance are directly supported in database schemas and in the query language.

- **Object-role modeling** is an approach that has been defined as “attribute free” and “fact-based”. The result is a verifiably correct system, from which other common artifacts, such as ERD, UML, and semantic models may be derived. Associations between data objects are described during the database design procedure, leading to inevitable database normalization.³⁶
- The **star schema** is the simplest of the data warehouse schemas; it consists of a few “fact tables” (possibly only one, justifying the name) referencing any number of “dimension tables”. The star schema is considered an important special case of the snowflake schema.

Data modeling can also be phrased as a high-level abstract design phase used to describe:

- the data contained in the database;
- the relationships between data items, and
- constraints on data.

The data items, relationships and constraints are all expressed using concepts provided by the high-level data model. Because these concepts do not include the implementation details, the result of the data modeling process is a semi-formal representation of the database structure. Database design includes logical design which is the definition of a database in a data model of a specific DBMS, and physical design which defines the internal database storage structure, file organization, and indexing techniques.

Database Design In database design, the first step is to identify **business rules** [11]. The design is then created and implemented using a DBMS.

- In an **external model**, the user’s view of a database (multiple different external views) is closely related to the real world as perceived by each user.
- **Conceptual models** provide flexible data-structuring capabilities; they offer a “community view” of the entire database (logical structure). This contains the data stored in the database and it shows relationships including: constraints, semantic information (e.g., business rules), security and integrity information, etc.³⁷
- **Internal models** are relational, network, and/or hierarchical data models. They consider the database as a collection of fixed-size records, closer to the physical level or the file structure. Internal models offer a representation of the database as seen by the DBMS and require the database designer to match the conceptual model’s characteristics and constraints to those of the selected implementation model; this may involve mapping entities in the conceptual model to tables in the relational model, say.
- **Physical models** are physical representations of the database, its lowest level of abstraction. The focus is on how to deal with runtime, storage utilization and compression, file organization and access, and data encryption. The physical level is managed by the operating system; it provides concepts that describe how the data is stored in computer memory, in detail.

36: “Database normalization is a technique for creating database tables with suitable columns and keys by decomposing a large table into smaller logical units. The process also considers the demands of the environment in which the database resides. Normalization is an iterative process. Commonly, normalizing a database occurs through a series of tests. Each subsequent step decomposes tables into more manageable information, making the overall database logical and easier to work with.” [9]

37: Conceptual models consider that a database is a collection of entities (objects) of various kinds, but they avoid detailed descriptions of the main data objects, in effect being independent of the eventual database implementation model.

Schemas We have already mentioned **schemas**, which are database descriptions represented by an **entity relationship diagram** (ERD, see *Structuring and Organizing Data* in Section 14.5). The most popular data models today are relational data models, although hierarchical and network data models are also often used on mainframe platforms. Relational data models describe the world as “a collection of inter-related relations (or tables)” [11].

Fundamental concepts include:

1. **relations** (table or file), which are subset of the Cartesian product of a list of domains characterized by a name;³⁸
2. **tables** and **columns** house the basic data components, into which content can be broken down;³⁹
3. a column’s **domain**, the range of values found in the column, and
4. **records**, which contain related fields, and **degree**, which refers to the number of attributes.⁴⁰

38: Within each table, the row represents a group of related data values; the row is known as a record or a tuple. Table columns are known as fields or attributes. Attributes are used to define a record, and a record contains a set of attributes.

39: Columns are combined into tables. Tables must have distinct names, no duplicate rows, and atomic entries (values that cannot be divided) in its columns.

40: Records and fields form the basis of all databases. A simple table provides the clearest picture of how records and fields work together in a database storage project.

17.3.3 Data Storage

Data storage refers to the collection and retention of digital information: the bits and bytes behind applications, network protocols, documents, media, address books, user preferences, and so on.

For computers, short term information is handled on random-access memory (RAM), and long-term information is held on storage volumes. Computers also distribute data by type. Markup languages have become popular formats for digital file storage: UML, XML, JSON, CSV, etc.

Data storage basically boils down to:

- the different ways to store different files;
- how to store them in the right kind of structures based on data type, and
- how those structures link together in a database.

It is data engineers and database analysts (**data managers**) that are responsible for storing collected and transformed data in various locations depending on the business requirements. Each combination of tool and location may store and access the data in different ways; the limitations, benefits, and use cases for each location and set of data must be taken into account as part of good data management.

For instance, let us assume a company is ingesting a million records a day from a particular data source. If the data is stored on a disk, we cannot simply append the daily updates to a singular file!⁴¹) Any report or question needing a particular piece of information found on the disk would never be produced/answered.

Instead, the company’s data engineers would:

- know that the data needs to be **partitioned** across different files and directories within the file system to separate the data;
- evaluate the data and how it is loaded and consumed to determine the appropriate way to **split it**,
- determine how to **update** specific pieces of data as changes are applied to the data source.

41: This would be akin to looking for a needle in the world’s largest haystack!

At a more meta level, there are other factors to consider, such as:

- is the data **key-value based** (see *Structuring and Organizing Data*, in Section 14.5)?
- are there **complex relationships** in the data?
- does the data need to be **processed** or **joined** with other datasets?
- and so on.

Data Warehousing **Data warehousing** is the term used to refer to the storage process of structured data. Data storage is transforming rapidly, since files can be compressed to take up less memory space, and computers can hold more files locally and in RAM.

Cloud-based data warehousing solutions like *Snowflake*, *AWS Redshift*, *Azure Synapse*, and *Google BigQuery* allow for pay-per-use data warehouses too, giving seemingly infinite storage.⁴²

For **on-premise data warehousing solutions**, the investment is all upfront. The customers pay for the data warehousing solution, but do not get to see any return on investment while the hardware is set up, configured, and operationalized.⁴³ Initially, then, businesses are left with a severely under-utilized piece of hardware, making such a move a high-risk leap of faith for anyone but the biggest players.

At some point in the warehouse lifetime, enough use cases exist to **eat the available hardware computer power or storage**. When this occurs, either more hardware must be acquired (at another large hit to the budget) or existing use cases that can be scaled back (and to what extent) must be identified to create the required “space”. Purchasing more hardware in this stage is not as much of a leap of faith as the initial commitment was, but will once again leave the organization with an under-utilized data platform as new use cases are prioritized and solutions built for them.

In comparison, **cloud-based data warehouse solutions** use a pay-per-use cost model, where there is an opportunity to prove the value of a use case using an iterative approach. The initial step is to implement a use case solution with very light requirements to help gauge cost estimates and to understand how valuable that solution might be. Future iterations can expand on the solution, modifying the complexity of data transformation or how data flows through it, and even remove it to focus on another use case, if appropriate.

At no point is there a need to consider purchasing and installing additional hardware, as new warehouses or clusters can be created on-demand. Using a cloud-based data warehouse allows costs to scale according to the number of use cases and their complexity. However, this requires a level of expertise⁴⁴ and a lack of control over any changes to prices or policies that go with cloud tools.

It is also important to consider who has access to what pieces of information that are stored (**data governance**). In practice, rules and regulations define who should have access to particular pieces of information within your organization. For a shipping company, as an example, we may need to separate the data that suppliers and customers can see at any given time, or ensure that different suppliers cannot see information about other suppliers.

42: This was written in August 2022; that list is liable to have changed quite a lot since then.

43: It might take months, with millions of dollars already invested, just to be able to start to implement a solution for the first use case.

44: Potentially different than the level of expertise required for on-premise warehousing, if not necessarily more sophisticated.

45: This might include adding additional data points to the collected data or storing data separately on disk.

46: Unfortunately, data governance is not achieved by using a specific tool or set of tools. Tools exist to support some of the aspects of data governance, but they only enhance existing data governance practices. Part of the challenge is that data governance is very much a “**people and process**” oriented discipline, intending to make data secure, usable, available, and of high quality.

47: Data pipelines do not need to contain machine learning models; they may instead focus, for example, on business intelligence functionality. But we will assume that they do.

48: For example, MLOps processes monitor models for drift in the context of the automated data stream, monitor models for performance relative to volume of data, and iteratively and automatically train and improve models over time based on the feedback received from this monitoring.

49: In modern data science contexts, MLOps may also refer to the entire data science process, from ingestion of the data to a live application that runs in a business environment and makes an impact at the business level. In this respect, MLOps overlaps with DataOps and DevOps.

50: CI/CD components refer to the training/re-training loop of a model, and do not extend to the full reporting and deployment pipeline. Even the concept of a CI/CD pipeline is often used to refer only to the training loop and do not extend to include the **entire operational pipeline**.

This requires **data classification, tagging, and access constraints**. When gathering data from various systems, a data engineer is responsible for applying the classification and tagging rules upon collection.⁴⁵

Then, when the data is aggregated or transformed, the end result must include this same information. When setting up access constraints to the data, the data engineer also has to enforce the required policies.

As more organizations are obtaining additional data from ever-growing new sources, they are faced with new problems:

- securing the data;
- ensuring regulatory compliance, and
- general management of the data.

These are also problems that **data governance** exists to solve.⁴⁶

17.4 Reporting and Deployment

Currently, the two main applications of data science in industry are **reporting and deployment** of machine learning models. In the context of data engineering, these machine learning models are **embedded** in the data pipeline.⁴⁷ As noted in a previous section, these machine learning models can be viewed metaphorically as the **brains** of the pipeline.

In an implemented context, managing machine learning models is known as **MLOps**. The traditional AI training cycle often involves a single pass of the following steps:

1. preparing the training data;
2. training the model,
3. evaluating the model.

These are still present in MLOps, with an increased focus on **ongoing monitoring/management** of the models embedded in the pipeline.⁴⁸

This iterative or interactive approach often includes **automated machine learning** (AutoML) capabilities; what happens outside the scope of the trained model is not included in this traditional definition.⁴⁹

17.4.1 Reports and Products

In the **research-first** approach to data science, which still dominates a lot of industry applications, machine learning models are used to generate **static or interactive reports** for business analysts; data science is handled as a **silo**, running batch predictions on historical data and returning the results for someone else to incorporate manually into applications.

In those conditions, there is little demand for **resiliency, scale, real-time access, or continuous integration and deployment (CI/CD)**; the results are of limited value, in and of themselves, and are used more as proof-of-concept.

Most data science solutions and platforms today still start with a research workflow but fail to move past the proof-of-concept stage.⁵⁰

Generally, we start an AI project with the **development of a model**:

1. data scientists **receive data**, which may be extracted manually from many sources;
2. the data is then **joined** and **cleaned** in an interactive way (using notebooks, perhaps), and
3. **training** and **experiments** are conducted while tracking results.

The model is **generated** and **tested/validated** until the results “look good”,⁵¹ at which point different teams take the results and attempt to integrate them into real-world applications.⁵² In most cases, eventually, the original data science product is set aside and re-implemented in a **robust** and **scalable** way which fits production, but which may not be what the data scientist originally intended.

A **production pipeline** starts with automated data collection and preparation, continues with automated training and evaluation pipelines, and incorporates real-time application pipelines, data quality and model monitoring, feedback loops, etc.

As applications that demand real-time recommendations, prevent fraud, predict failures, and make decisions continue to be in demand, engineering efforts are required to make them feasible. Business needs have forced data science components to be **robust**, **performant**, **highly scalable**, and **aligned with agile software and DevOps practices**.⁵³

Instead of this siloed, complex, and manual process, we should start by designing the ML elements of the pipeline using a **modular strategy**, where the different parts of the ML component provide a continuous, automated, and far simpler way to move from research and development to scalable production pipelines, without the need to refactor code, add glue logic, and spend significant efforts on data and ML engineering.⁵⁴

17.4.2 Cloud and On-Premise Architecture

Organizations have to make decisions on how much of their data architecture to **build in-house**, and how much to build with **off-the-shelf** tools. Additionally, there are compromises and benefits to building infrastructure **on the cloud** (renting external resources) with potential to publish results for anyone in the world to see and build on, and building solutions on premise which depend heavily on local capacity and hardware.*

Developers must write new code for every data source, and may need to rewrite it if a vendor changes its API, or if the organization adopts a different data warehouse destination. Data engineers must also address **speed** and **scalability**: for time-sensitive analysis or business intelligence applications, ensuring **low latency** can be crucial to providing data that drives decisions.

51: That is, they meet a certain performance threshold.

52: Modern tools (such as Flask) allow data scientists to **serialize a model** into a file and then simply call the file to make predictions. However, the full process of monitoring, creating feedback loops, then retraining and updating the model still requires an underlying architecture.

53: It is all too often the case that **operationalizing** machine learning (in the sense of considering all business requirements, such as federated data sources, need for scale, critical implications of real-time data ingestion or transformation, online feature engineering, handling upgrades, monitoring, etc.) comes as an afterthought, making it all the more difficult to create real business value with AI.

54: ML production-ready pipelines have four key components:

1. **feature store**: collects, prepares, catalogues, and serves data features for development (offline) and real-time (online) usage;
2. **machine learning CI/CD pipeline**: automatically trains, tests, optimizes, and deploys or updates models using a snapshot of the production data (generated by the feature store) and code from the source control (Git);
3. **real-time/event-driven application pipeline**: includes the API handling, data preparation/enrichment, model serving, ensembles, driving and measuring actions, etc., and
4. **real-time data and model monitoring**: monitors data, models, and production components, and provides a feedback loop for exploring production data, identifying drift, alerting on anomalies or data quality issues, triggering re-training jobs, measuring business impact, etc.

* Many companies, such as Spotify, build their own pipelines from scratch to analyze data and understand user preferences, and map customers to music preferences, say. The main challenges to developing in-house pipelines are that different data sources provide **different application program interfaces** (API, see Section 16.3.6) and involve different kinds of technologies.

55: Such pipelines may be all that is required in certain cases, such as establishing a proof-of-concept for business processes that require less frequent and manual decision-making. For example, a retailer can use them to make decisions about the order of recommendation of certain items in an online store, but may miss on recommending a product to an individual on a certain short-term buying spree in real-time.

56: Even without larger, more specialized tools, simple desktop tools such as *Tableau*, *Looker*, or *Microsoft's Power BI* can still be used to run queries and reports, and with a modern real-time pipeline the results will be current and immediately actionable.

57: This might perhaps be the only thing worth remembering from this chapter, especially since technology changes so quickly.

Data solutions need to be able to dynamically access more resources as data volume grows. Therefore, in-house pipelines can be expensive to **build and maintain**.

On-premise amateur-ish data pipelines ingest data in **pre-scheduled batches** (e.g., twice every hour or every night, say), and are not ideal for any real-time analytics solutions.⁵⁵

ETL tools that work with in-house data warehouses do as much preparation work as possible, including transformation, prior to loading data into data warehouses. Cloud data warehouses like *Amazon Redshift*, *Google BigQuery*, *Azure SQL Data Warehouse*, and *Snowflake* can scale up and down in seconds or minutes, so developers can replicate raw data from disparate sources and define transformations in SQL and run them in the data warehouse after loading or at the time of query.

Just as there are cloud-native data warehouses, there also are **ETL services built for the cloud**. Organizations can set up a cloud-first platform for moving data in minutes, and data engineers can rely on the solution to monitor and handle unusual scenarios and failure points.⁵⁶

Overall, cloud tools are becoming more and more popular to host data pipelines and by extension data science solutions.

Data engineering and data management are not always the most interesting aspects of the discipline for data analysts, but the long and the short of it is that it is impossible to conduct meaningful data science without the right data or without the right tools.⁵⁷ But tools do not only refer to the analytical tools; becoming familiar with the entire **data ecosystem** will pay off in the end.

Chapter References

- [1] *Introduction to Data Engineering* [↗](#).
- [2] Anouk Dutrée. *Data pipelines: what, why and which ones* [↗](#). 2021.
- [3] *What is Data Engineering? Everything You Need to Know in 2022* [↗](#). phData, 2022.
- [4] *Systems and software engineering - Vocabulary, ISO/IEC/IEEE std 24765:2010(E)* [↗](#). 2010.
- [5] M. Kleppmann. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly, 2017.
- [6] Henryk Konsek. *Automating Data Pipelines: Types, Use Cases, Best Practices* [↗](#).
- [7] J. Kunigk et al. *Architecting Modern Data Platforms: A Guide to Enterprise Hadoop at Scale*. O'Reilly, 2018.
- [8] T. Malaska and J. Seidman. *Foundations for Architecting Data Solutions: Managing Successful Data Projects*. O'Reilly, 2018.
- [9] *What is Database Normalization?* [↗](#).
- [10] E. Uz. *Analysis of the data job market using "Ask HN: Who is hiring?" posts* [↗](#). Aug. 2023.
- [11] Adrienne Watt. *Database Design* [↗](#). BCCampus, 2014.
- [12] *Data Governance* [↗](#).
- [13] *DataOps* [↗](#).

by **Patrick Boily**, with contributions from **Ehssan Ghashim** and **Maia Pelletier**

Why do we display evidence in a report, in a newspaper article, or online? What is the fundamental goal of our charts and graphs? Representing data properly, in a manner that allows an audience to gain insight about the underlying situation, is, without a doubt, one of the most important skill a data scientist and/or quantitative consultant must possess.

In this chapter, we introduce some commonly-used charts, discuss the fundamental principles of analytical designs, and give a brief overview of dashboards. A **substantially** more thorough treatment (including storytelling with data, the grammar of graphics and its implementation in R and `ggplot2`, and the basics of Power BI) is available in [1].*

Interested readers should also consider consulting the following indispensable data visualization resources: [2, 3, 7, 13, 16, 18, 19, 20, 21, 22], among others.

18.1 Data and Charts

As data scientist Damian Mingle once put it, modern data analysis is a different beast:

“Discovery is no longer limited by the collection and processing of data, but rather management, analysis, and visualization. [15]”

What can be done with the data, once it has been collected/processed?

Two suggestions come to mind:

- **analysis** is the process by which we extract actionable insights from the data (this process is discussed in later subsections), while
- **visualization** is the process of presenting data and analysis outputs in a visual format; visualization of data *prior* to analysis can help simplify the analytical process; **post-analysis**, it allows for the results to be communicated to various stakeholders.

In this section, we focus on important visualization concepts and methods; we shall provide examples of data displays to illustrate the various possibilities that might be produced by the data presentation component of a data analysis system.

* In this chapter, we preview (roughly) Sections 1.3, 1.5, and 2.3 of [1].

18.1 Data and Charts	1089
Pre-Analysis Uses	1090
Presenting Results	1090
Multivariate Elements	1091
Visualization Catalogue	1096
Accessibility	1099
18.2 Analytical Design	1099
Comparisons	1100
Mechanism/Explanation	1102
Multivariate Analysis	1104
Integration of Evidence	1106
Documentation	1107
Content First	1110
18.3 Dashboards	1111
Dashboard Fundamentals	1111
Dashboard Structure	1113
Dashboard Design	1114
Examples	1115
18.4 Exercises	1116
Chapter References	1118

18.1.1 Pre-Analysis Uses

Even before the analytical stage is reached, data visualization can be used to set the stage for analysis by:

- detecting **invalid entries** and **outliers**;
- shaping the **data transformations** (binning, standardization, dimension reduction, etc.);
- getting a **sense for the data** (data analysis as an art form, exploratory analysis), and
- identifying **hidden data structures** (clustering, associations, patterns which may inform the next stage of analysis, etc.).

18.1.2 Presenting Results

The crucial element of data presentations is that they need to help **convey the insight** (or the message); they should be clear, engaging, and (more importantly) readable. Our ability to think of questions (and to answer them) is in some sense **limited by what we can visualize**.

There is always a risk that if certain types of visualization techniques dominate in evidence presentations, the kinds of questions that are particularly well-suited to providing data for these techniques will come to dominate the landscape, which will then affect data collection techniques, data availability, future interest, and so forth.

Generating Ideas and Insights In *Beautiful Evidence* [18], E. Tufte explains that evidence is presented to assist our thinking processes. He further suggests that there is a symmetry to visual displays of evidence – that visualization consumers should be seeking exactly (and explicitly) what the visualization producers should be providing, namely:

- meaningful comparisons;
- causal networks and underlying structure;
- multivariate links;
- integrated and relevant data, and
- a primary focus on content.

We will discuss this further in Section 18.2.

Selecting a Chart Type The choice of visualization methods are strongly dependent on the analysis objective, that is, on the **questions that need to be answered**. Presentation methods should not be selected randomly (or simply from a list of easily-produced templates) [1].

In Figure 18.1, F. Ruys suggests various types of visual displays that can be used, depending on the objective:

- who is involved?
- where is the situation taking place?
- when is it happening?
- what is it about?
- how/why does it work?
- how much?

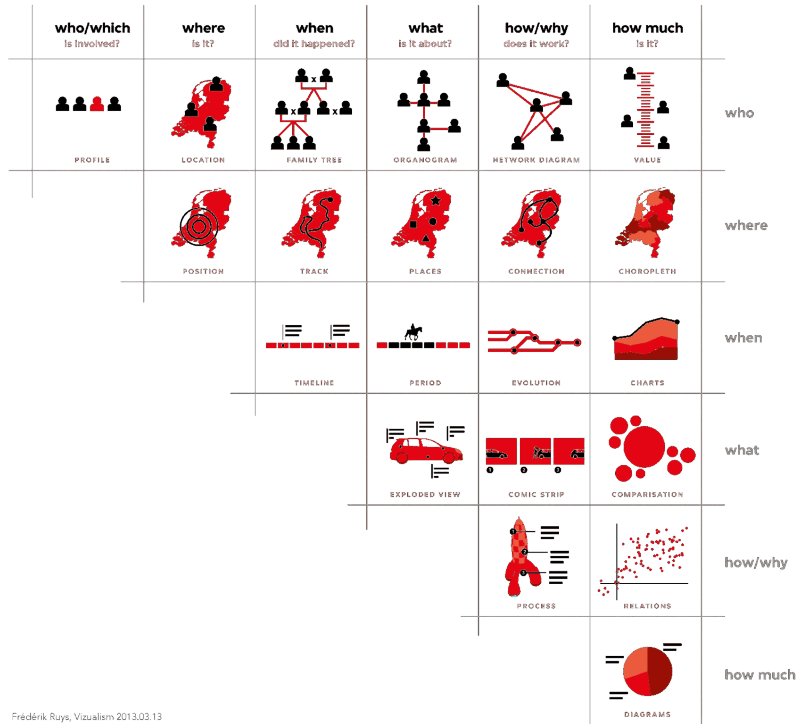


Figure 18.1: Data visualization suggestions, by type of question [F.Ruys, Vizualism.nl].

A general dashboard should at least be able to produce the following types of display:

- **charts** – comparison and relation (scatterplots, bubble charts, parallel coordinate charts, decision trees, cluster plots, trend plots);
- **choropleth maps** (heat maps, classification maps);
- **network diagrams** and connection maps (association rule networks, phrase nets),
- **univariate diagrams** (word clouds, box plots, histograms).

18.1.3 Multivariate Elements in Charts

It is not unusual to see modern datasets with 5, 10, 100, or even 1000+ variables. **High-dimensionality** brings a host of problems;¹ from a data visualization perspective, the challenge is that at most two attributes can be represented by position in the plane. How can we then represent other crucial elements on a flat computer screen or a piece of paper?

Potential solutions include using a **3-dimensional physical display** (such as one produced by 3D printing), or, more reasonably, one of the following **visual elements**:

- | | |
|-------------------------------|------------------------|
| ▪ marker size, | ▪ line orientation, |
| ▪ marker colour, | ▪ marker shape, and/or |
| ▪ colour intensity and value, | ▪ motion/movie. |
| ▪ marker texture, | |

These elements do not always mix well – there can be “too much of a good thing”, so to speak.

1: Such as the **curse of dimensionality**, which is remedied by feature selection and dimension reduction (see Chapter 23).

2: Efficient design is as much art as it is science: while we do want to highlight **multivariate** relationships, we do need to keep things “parsable”. Less is more, as long as “less” is enough.

In practice, human brains can reasonably be hoped to integrate 4 or 5 design elements in a chart (including 2 reserved for position), together with a motion component; the use of additional design elements tends to complicate matters and confuse the reader more than anything.²

In Figure 18.2, we provide a two-dimensional scatterplot display of a subset of the [NASA CM1 dataset](#).

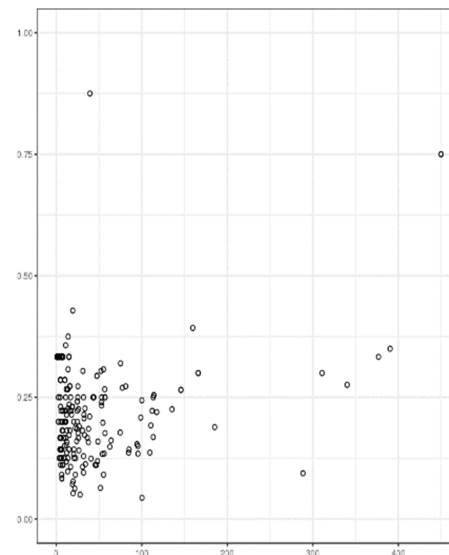


Figure 18.2: Scatterplot of a subset of the NASA CM1 dataset (minimalist).

3: Who could blame anyone for skipping over this chart entirely on their way to more entertaining fare? Is there **anything** of even remote interest that can be said about the chart and the underlying data?

We went out of our way to find a display that is as **uninformative** as it is dull – you will agree that we succeeded in our attempt.³ The addition of data-aligned design elements does not only make the chart more appealing – it also allows for **insight discovery** (see Figure 18.3).

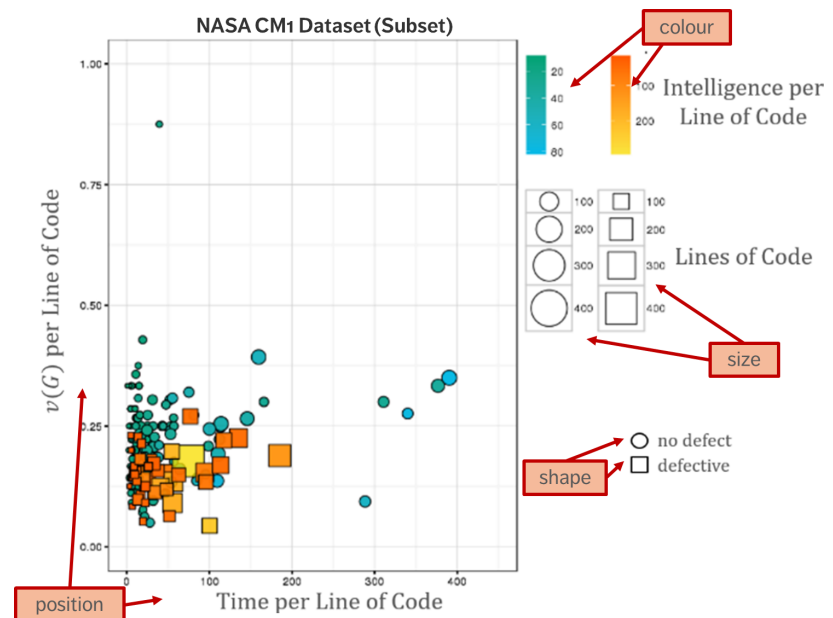


Figure 18.3: Bubble chart of a subset of the NASA CM1 dataset (with 5 variables).

4: True, this might not come as much of a surprise, but even such simple insights as these remain unreachable from Figure 18.2 alone.

For instance, we can see that defective components tend to contain more lines of code than non-defective components, *on average*, and that components for which more time was spent per line of code tend to be non-defective.⁴

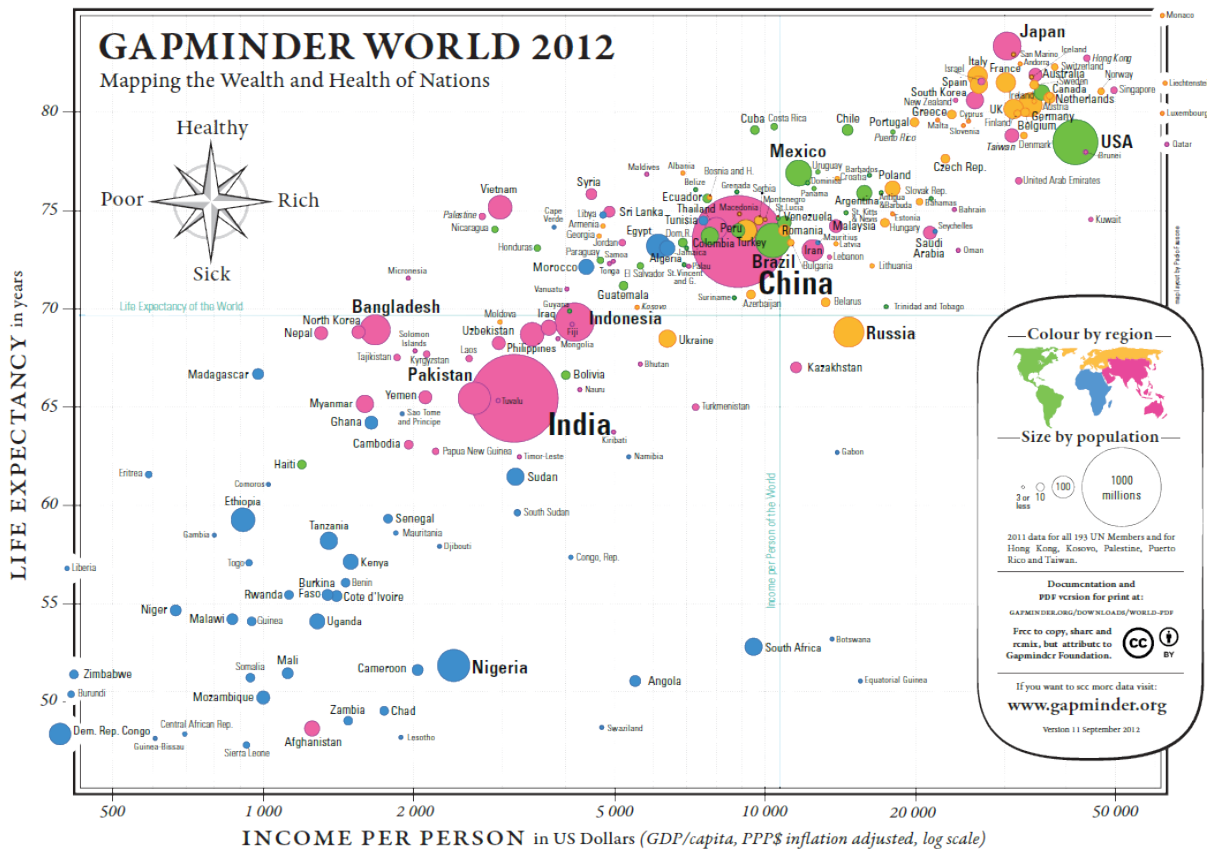


Figure 18.4: Gapminder's Health and Wealth of Nation (2012) [17].

Example: Bubble Chart Health and Wealth of Nations (Figure 18.4).

■ Data:

- 2012 life expectancy in years
- 2012 inflation adjusted GDP/capita in USD
- 2012 population for 193 UN members and 5 other countries

■ Some Questions and Comparisons:

- Can we predict life expectancy given a GDP/capita?⁵
- Are there outlier countries?⁶
- Are countries with a smaller population healthier?⁷
- Is continental membership an indicator of health and wealth levels?⁸
- How do countries compare against world values for life expectancy and GDP per capita?⁹

■ Multivariate Elements: position for health and wealth, bubble size for population, colour for continental membership, and labels to identify the nations.

■ Comments:

- Are life expectancy and GDP/capita appropriate proxies for health and wealth?
- A fifth element could also be added to a screen display: the passage of time. In this case, how do we deal with countries coming into existence (and ceasing to exist as political entities)?

5: The trend is linear: $\text{Expectancy} \approx 6.8 \times \ln \text{GDP/capita} + 10.6$

6: Botswana, South Africa, and Vietnam, among others, at a glance.

7: Bubble size seems uncorrelated with the axes' variates.

8: There seems to be a clear divide between Western Nations (and Japan), most of Asia, and Africa.

9: The vast majority of countries fall in three of the quadrants. There are very few wealthy countries with low life expectancy. China sits near the world values, which is expected for life expectancy, but more surprising when it comes to GDP/capita – compare with India.

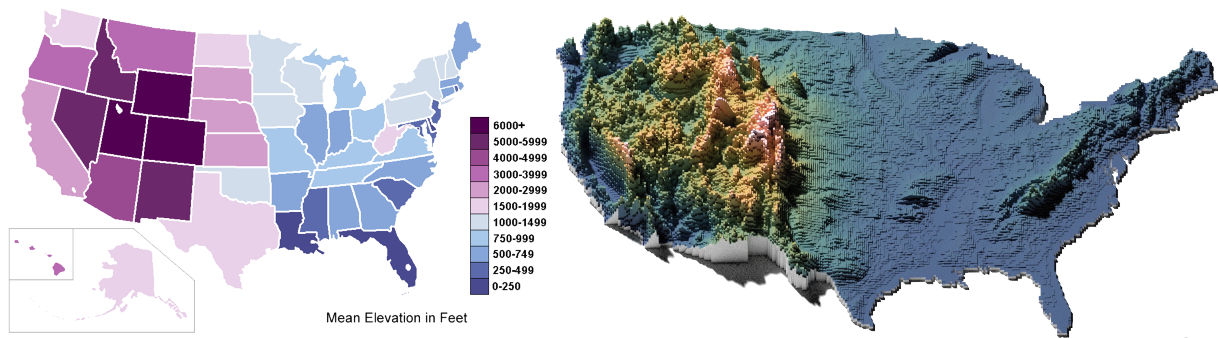


Figure 18.5: Mean elevation by U.S. state, in feet (left, source unknown); high resolution elevation map (right, by twitter user @cstats).

Example: Choropleth Map Mean Elevation by U.S. State (Figure 18.5).

- **Data:** 50 observations, ranging from sea level (0-250) to (6000+)
- **Some Questions and Comparisons:**
 - Can the mean elevation of the U.S. states tell us something about the global topography of the U.S.? ¹⁰
 - Are there any states that do not “belong” in their local neighbourhood, elevation-wise? ¹¹
- **Multivariate Elements:** geographical location (position) and elevation (purple-blue colour gradient as the marker for mean elevation in the chart on the left; height and colour gradient in the chart on the right).
- **Comments:**
 - Is the ‘mean’ the right measurement to use for this map? ¹²
 - Are there ways to include other variables in this chart? ¹³
 - What is going on with the scale in the legend? ¹⁴

10: Western states have higher mean elevation, probably due to the presence of the Rockies; Eastern coastal states are more likely to suffer from rising water levels, for instance.

11: West Virginia and Oklahoma seem to have the “wrong” shade – is that an artefact of the colour gradient and scale?

12: It depends on the author’s purpose.

13: Population density with texture, for instance

14: No idea,,,

Example: Network Diagram Lexical Distances (Figure 18.6).

- **Data:**
 - speakers and language groups for 43 European languages
 - lexical distances between languages
- **Some Questions and Comparisons:**
 - Are there languages that are lexically closer to languages in other lexical groups than to languages in their own groups? ¹⁵
 - Which language has the most links to other languages? ¹⁶
 - Are there languages that are lexically close to multiple languages in other groups? ¹⁷
 - Is there a correlation between the number of speakers and the number of languages in a language group? ¹⁸
 - Does the bubble size refer only to European speakers? ¹⁹
- **Multivariate Elements:**
 - colour and cluster for language group
 - line style for lexical distance
 - bubble size for number of speakers

15: French is lexically closer to English than it is to Romanian, say.

16: English has 10 links.

17: Greek is lexically close to 5 groups

18: Language groups with more speakers tend to have more languages.

19: Portuguese seems to have as many speakers as French? Worldwide, that may be the case, but in Europe that is definitely not so.

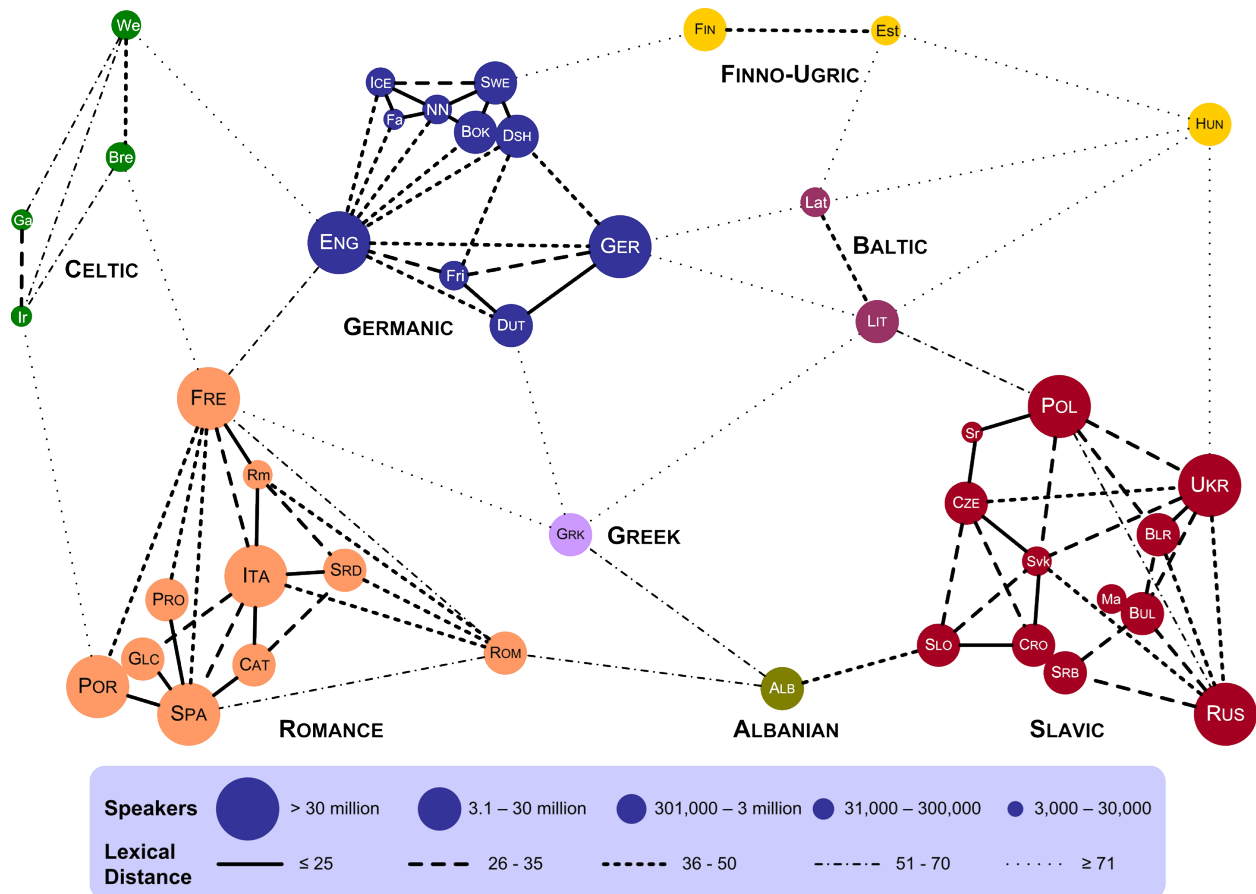


Figure 18.6: Lexical distances of European languages [6].

■ **Comments:**

- How is lexical distance computed? Is it actually a “distance” in the mathematical sense?
- Some language pairs are not joined by links – does this mean that their lexical distance is too large to be rendered?
- Are the actual geometrical distances meaningful? For instance, Estonian is closer to French in the chart than it is to Portuguese ... is it also lexically closer?²⁰

20: Full disclosure, we are not sure if “lexically” is a word... but you know what we mean.

Example: Parallel Coordinates Data Strings (Figure 18.7).

■ **Data:**

- demographic information for a conference’s participants
- preference for using a spoon (red string) or a fork (white string) for breakfast

■ **Some Questions and Comparisons:**

- Where is the conference most likely to be taking place?²¹
- Were there more men or women at the conference?²²
- Can we see a correlation between age and height, or height and weight?²³

21: Most strings go through Europe (conference taking place in Barcelona, in fact).

22: Seems about 50-50, but note the one string bypassing the gender axis completely – evidence of sub-optimal questionnaire design?

23: There is likely to be one, but the inclusion of “status” and “left/right handedness” between the pairs of interest makes it impossible to tell.

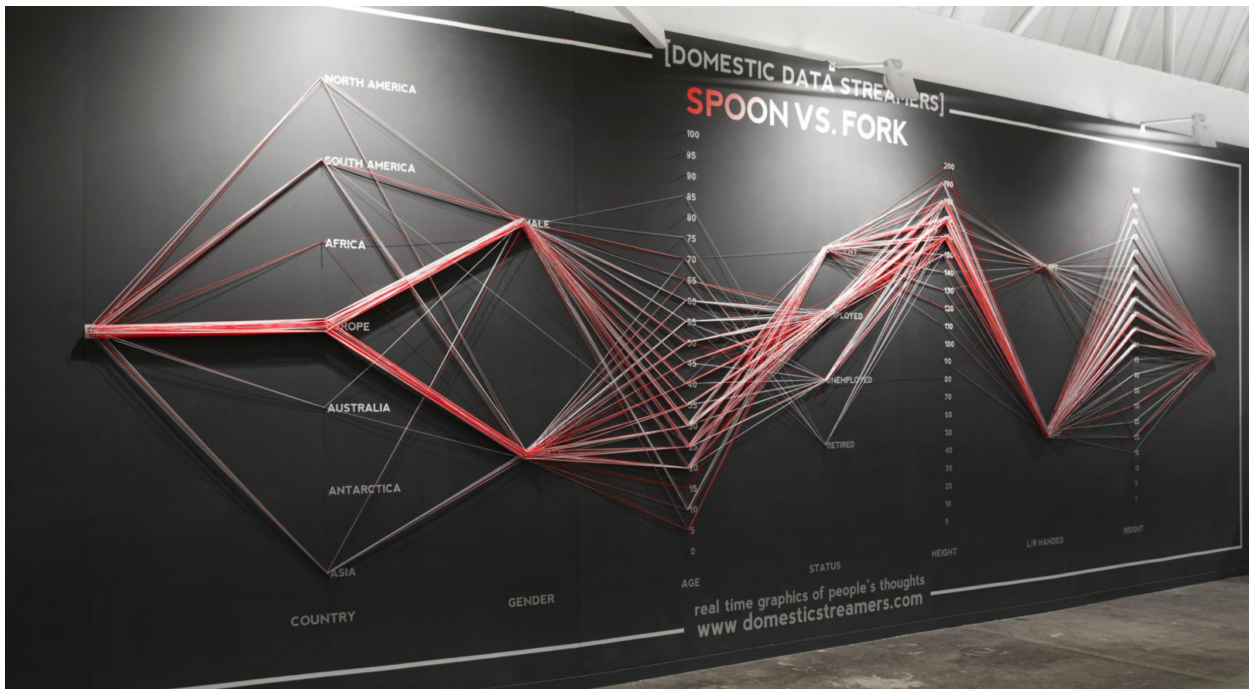


Figure 18.7: Physical visualization of people's demographics [Domestic Data Streamers [↗](#)].

■ **Multivariate Elements:**

- seven demographic measurements (country (?), gender, age, status, height, left/right handed, weight)
- colour to represent the preference (spoon vs. fork)

■ **Comments:**

- There does not seem to be a link between the colour of the string and the measurements – it seems as though the spoon vs. fork question is a red herring (or at least a way to get attendees to participate in the hands-on data collection exercise)
- Are the selected scales reasonable? Why include heights below 60cms?

18.1.4 Visualization Catalogue

Here are some examples of other types of visualizations; more comprehensive catalogues can be found in [1, 2, 3, 21, 13], among others.

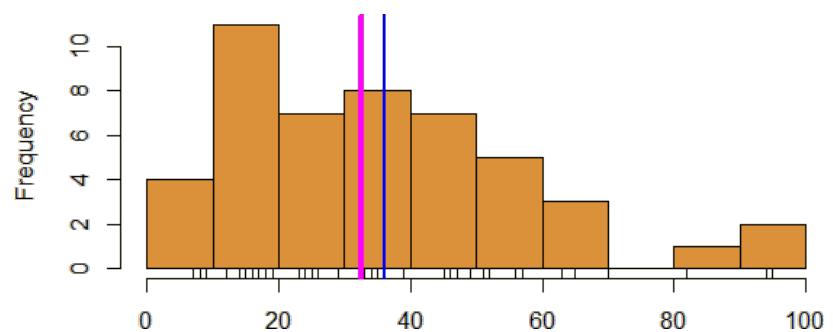


Figure 18.8: Histogram: reported weekly work hours (personal file).

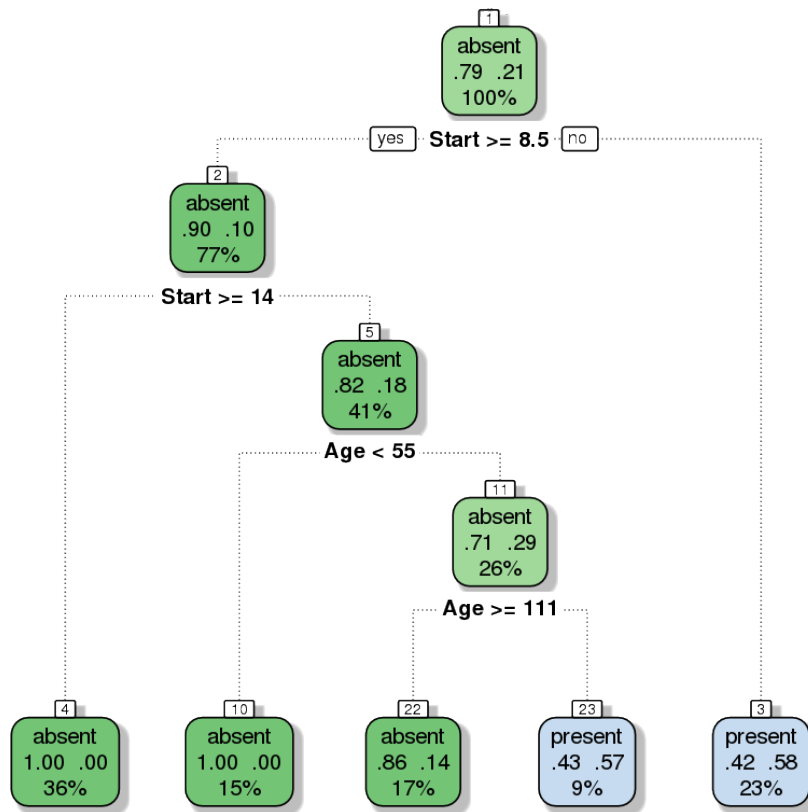


Figure 18.9: Decision Tree: classification scheme for the kyphosis dataset (personal file).

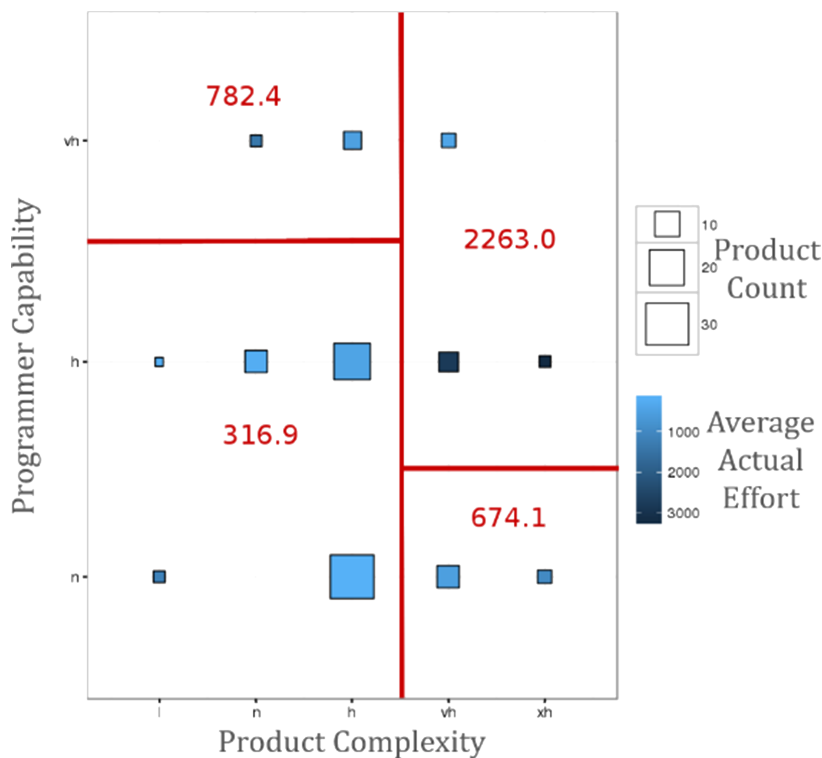


Figure 18.10: Decision tree bubble chart: estimated average project effort (in red) overlaid over product complexity, programmer capability, and product count in NASA's COCOMO dataset (personal file).

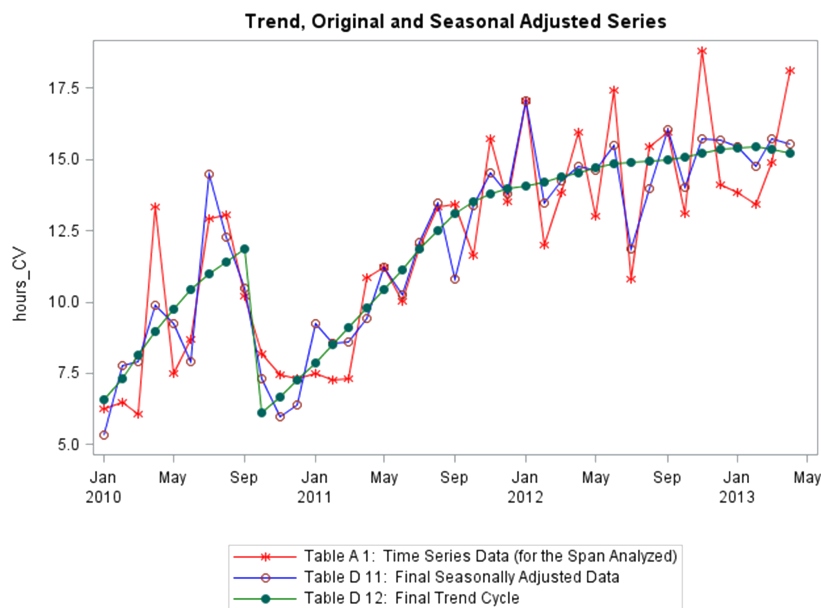


Figure 18.11: Time series: trend, seasonality, shifts of a supply chain metric (personal file).

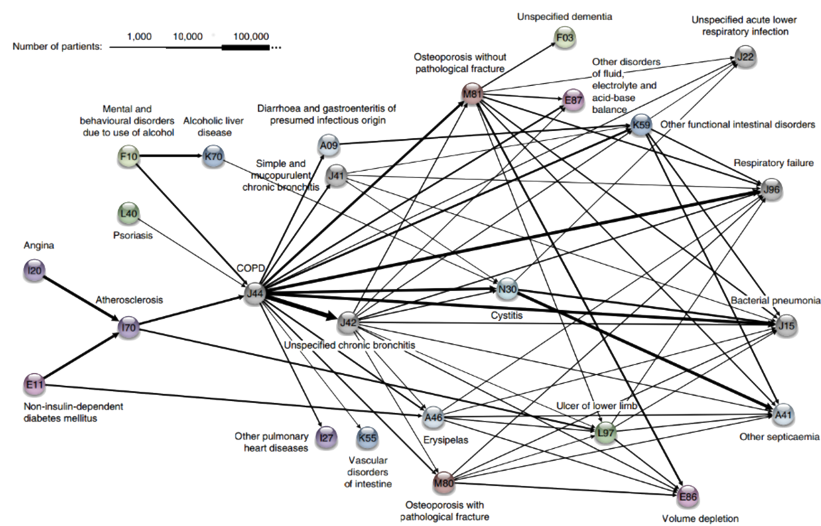


Figure 18.12: Association rules network: diagnosis network around COPD in the Danish Medical Dataset [Jensen].

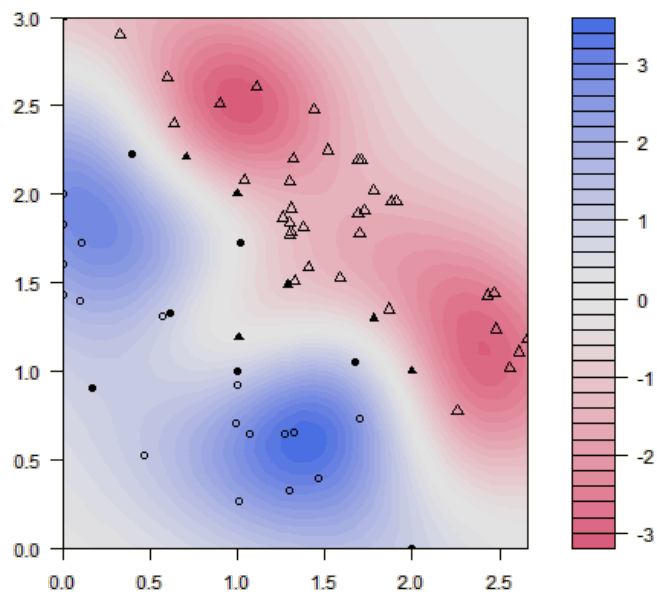


Figure 18.13: Classification scatterplot: artificial dataset (personal file).

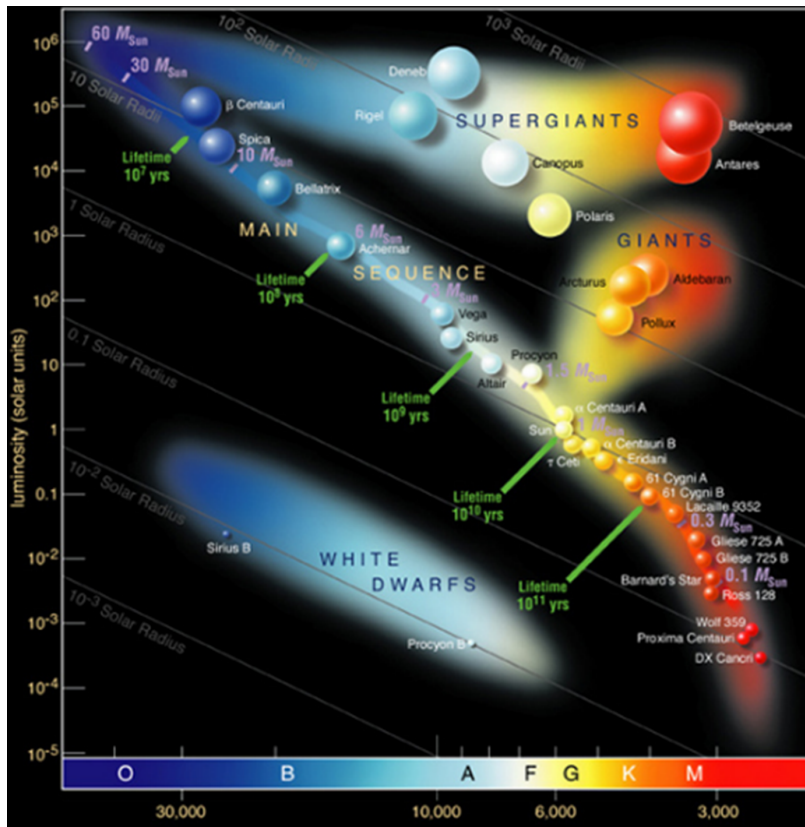


Figure 18.14: Classification bubble chart: Hertzsprung-Russell diagram of stellar evolution (European Southern Observatory).

18.1.5 A Word About Accessibility

While visual displays can help provide analysts with insight, some work remains to be done in regard to visual impairment – short of describing the features/emerging structures in a visualization, graphs can at best succeed in conveying relevant information to a subset of the population.

The onus remains on the analyst to not only produce **clear** and **meaningful** visualizations (through a clever use of **contrast**, say), but also to describe them and their features in a fashion that allows all to “see” the insights. One drawback is that in order for this description to be done properly, the analyst needs to have seen all the insights, which is not always possible. Examples of “data physicalizations” can be found in [5].²⁴

24: There is a lot more to say on the topic (see [1], for instance).

18.2 Principles of Analytical Design

In his 2006 offering *Beautiful Evidence*, E. Tufte highlights what he calls the **Fundamental Principles of Analytical Design** [18]. Tufte suggests that we present evidence to assist our thinking processes [18, p.137].

In this regard, his principles seem universal – a strong argument can be made that they are dependent neither on technology nor on culture.

Reasoning (and communicating our thoughts) is intertwined with our lives in a causal and dynamic multivariate Universe (the 4 dimensions of space-time making up only a small subset of available variates); whatever cognitive skills allow us to live and evolve can also be brought to bear on

the presentation of evidence. Tufte also highlights a particular symmetry to visual displays of evidence, being that **consumers of charts should be seeking exactly what producers of charts should be providing** (more on exactly what that is in a little bit).

Physical science displays tend to be less descriptive and verbal, and more visual and quantitative; up to now, these trends have tended to be reversed when dealing with evidence displays about human behaviour.

In spite of this, Tufte argues that his principles of analytical design can also be applied to social science and medicine. To demonstrate the universality of his principles, he describes in detail how they are applied to [Minard](#)'s celebrated *March to Moscow*.²⁵

25: His lengthy analysis of the image is well worth the read [18, pp.122-139] – it will not be repeated here (although other aspects of the chart are discussed in [1]).

Rather, we will illustrate the principles with the help of the Gapminder's Foundation 2012 *Health and Wealth of Nations* data visualization (see Figure 18.4), a bubble chart plotting 2012 life expectancy, adjusted income per person in USD (log-scaled), population, and continental membership for 193 UN members and 5 other countries (a high-resolution version of the image is available on the [Gapminder website](#).)

Tufte identifies 6 basic properties of superior analytical charts:

- their ability to conduct **meaningful comparisons**;
- their ability to identify **underlying structures** and **potential causal avenues**;
- their incorporation of **multivariate links**;
- their ease of **integration** and use of **relevant data**;
- their **honest and complete documentation**, and
- their primary focus on content

18.2.1 Comparisons

“Show comparisons, contrasts, differences.” [18, p.127]

Comparisons come in varied flavours: for instance, one could compare a:

- unit at a given time against the same unit at a later time;
- unit's component against another of its components;
- unit against another unit,
- or any number of combinations of these flavours.

Tufte further explains that

[...] the fundamental analytical act in statistical reasoning is to answer the question “Compared with what?” Whether we are evaluating changes over space or time, searching big data bases, adjusting and controlling for variables, designing experiments, specifying multiple regressions, or doing just about any kind of evidence-based reasoning, **the essential point is to make intelligent and appropriate comparisons** [*emphasis added*]. Thus, visual displays [...] should show comparisons. [18, p.127]

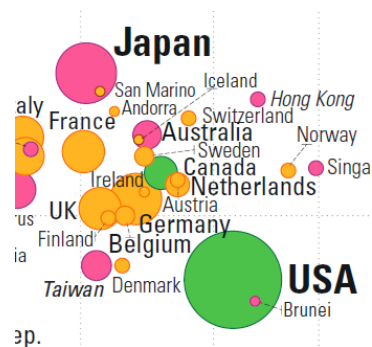
Not every comparison will be insightful, but avoiding comparisons altogether is equivalent to producing a useless display, built from a single datum.

Health and Wealth of Nations First, note that each bubble represents a different country, and that the location of each bubble's centre is a precise point corresponding to the country's life expectancy and its GDP per capita. The size of the bubble correlates with the country's population and its colour is linked to continental membership.

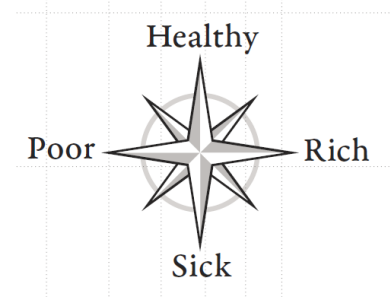
The chart's compass²⁶ provides a handy comparison tool:

- a bubble further to the right (resp. the left) represents a “wealthier” (resp. “poorer”) country;
- a bubble further above (resp. below) represents a “healthier” (resp. “sicker”) country.

A comparison between Japan, Germany and the USA shows that Japan is healthier than Germany, which is itself healthier than the USA (as determined by life expectancy) while the USA is wealthier than Germany, which is itself wealthier than Japan (as determined by GDP per capita, see Figure 18.15).²⁷



26: Top left:



27: Health and wealth are such complicated concepts that they cannot simply be represented by a single measure (and perhaps not even by multiple measures). Nevertheless, we use them as proxies in this example.

Figure 18.15: Comparisons in the Gapminder chart: country-to-country.

It is possible for two countries to have roughly the same health and the same wealth: consider Indonesia and Fiji, or India and Tuvalu, for instance (see Figure 18.16).

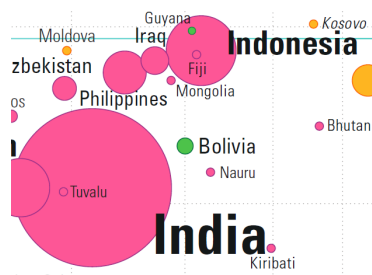


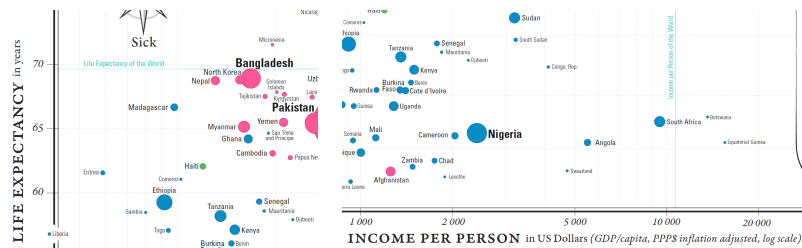
Figure 18.16: Comparisons in the Gapminder chart: country-to-country overlap.

In each pair, the centres of both bubbles (nearly) overlap: any difference in the data must be found in the bubbles' area or in their colour.

Countries can also be compared against **world values** for life expectancy and GDP per capita (a shade under 70 years and in the neighbourhood of \$11K, respectively). The world's mean life expectancy and income per person are traced in light blue (see Figure 18.17).²⁸

28: We see for instance, that Nepal's life expectancy is just a hair below the life expectancy on the planet, and that Botswana's GDP per capita is above the global income per person.

Figure 18.17: Comparisons in the Gapminder chart: country-to-world-life-expectancy (left) and country-to-world-income-per-person (right).



Wealthier, healthier, poorer, and sicker are relative terms, but we can also use them to classify the world's nations with respect to these mean values, “wealthier” meaning “wealthier than the average country”, and so on.

18.2.2 Mechanism, Structure, Explanation

“Show causality, mechanism, explanation, systematic structure.” [18, p.128]

In essence, this is the core principle behind data visualization: the display needs to explain *something*, it needs to provide (potential) links between cause and effect. As Tufte points out,

[...] often the reason that we examine evidence is to understand causality, mechanism, dynamics, process, or systematic structure [emphasis added] [...] Reasoning about reforms and making decisions also demands causal logic. To produce the desired effects, we need to know and govern the causes; thus “policy-thinking is and must be causality-thinking”. [18, p.128], [4]

Note also that

simply collecting data may provoke thoughts about cause and effect: measurements are inherently comparative, and comparisons promptly lead to reasoning about various sources of differences and variability. [18, p.128]

Finally, if the visualization can be removed without diminishing the narrative, then that chart should in all probability be excluded from the final product, no matter how pretty and modern it looks, or how costly it was to produce.

Health and Wealth of Nations At a glance, the relation between life expectancy and the logarithm of income per person seems to be increasing more or less linearly. Without access to the data, the exact parameter values cannot be estimated analytically, but an approximate line-of-best-fit has been added to the chart in Figure 18.18.

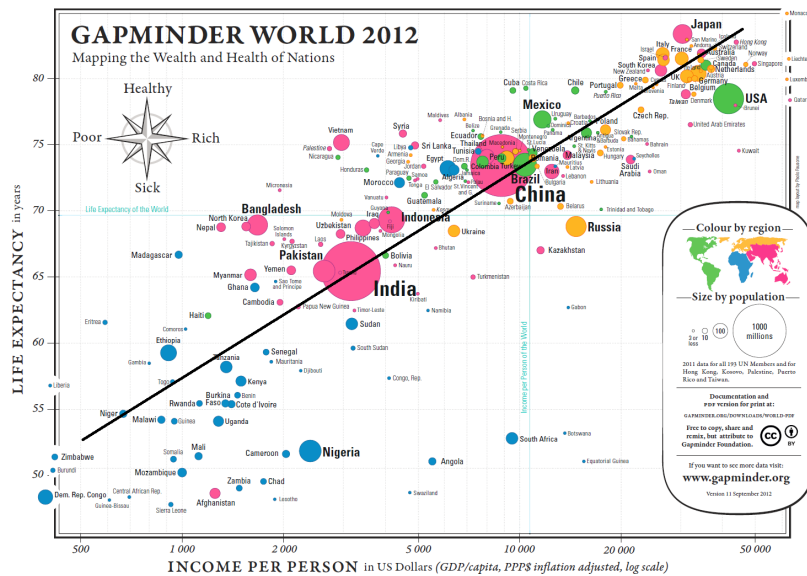


Figure 18.18: Approximate line of best fit for the Gapminder chart.

Using the points (10K, 73.5) and (50K, 84.5) yields a line with equation

$$\text{Life Expectancy} \approx 6.83 \times \ln(\text{Income Per Person}) + 57.76.$$

The exact form of the relationship and the numerical values of the parameters are of little significance at this stage – the key insight is that wealthier countries appear to be healthier, generally, and *vice-versa*.²⁹

The chart also highlights an interesting feature in the data, namely that the four quadrants created by separating the data along the Earth's average life expectancy and GDP per capita do not all host similar patterns.

Naïvely, it might have been expected that each of the quadrants would contain about 25% of the world's countries (although the large population of China and India muddle the picture somewhat). However, one quadrant is substantially under-represented in the visualization. Should it come as a surprise that there are so few “wealthier” yet “sicker” countries? (see Figure 18.19).

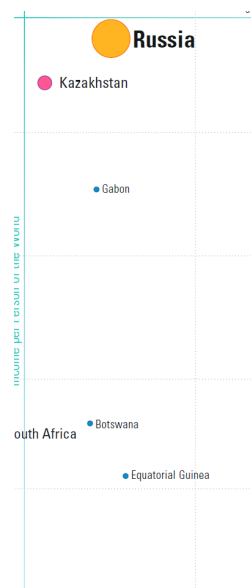


Figure 18.19: Close-up, bottom right quadrant.

It could even be argued that Russia and Kazakhstan are in fact too near the “separators” to really be considered clear-cut members of the quadrant, so that the overwhelming majority of the planet’s countries are found in one of only three quadrants.

In the same vein, when we consider the data visualization as a whole, there seems to be one group of outliers below the main trend, to the right, and to a lesser extent, one group above the main trend, to the left (see Figure 18.20).

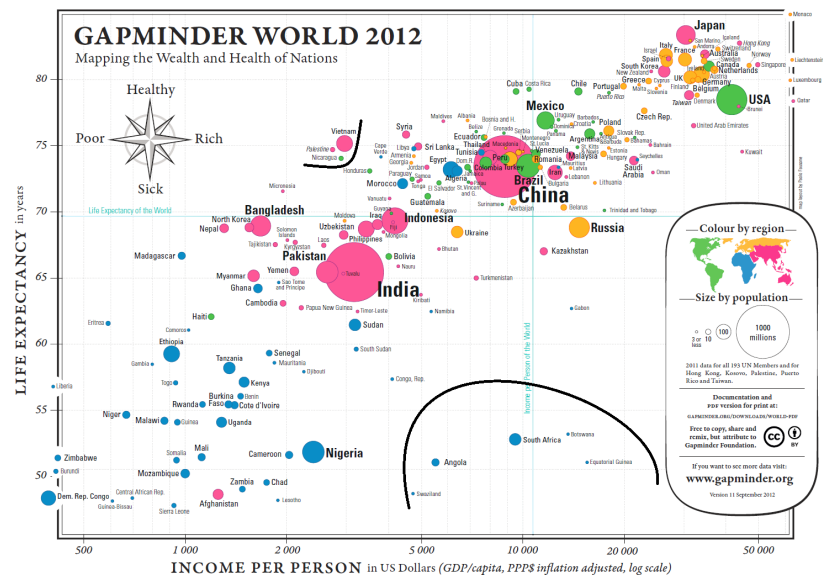


Figure 18.20: Potential outliers in the Gap-minder chart.

30: Could income disparity between a poorer majority and a wealthier minority push the bubble to the right, while lower life expectancy of the majority drives the overall life expectancy downward?

31: We discuss the topic further in the exercises.

These cry out for an explanation: South Africa, for instance, has a relatively high GDP per capita but a low life expectancy.³⁰ This brings up a crucial point about data visualization: it seems virtually certain that the racial politics of *apartheid* played a major role in the position of the South African outlier, but the chart emphatically **DOES NOT** provide a proof of that assertion.³¹

Charts suggest, but “proofs” come from deeper **domain-specific analyses**.

18.2.3 Multivariate Analysis

“Show multivariate data; that is, show more than 1 or 2 variables.” [18, p.130]

In an age where data collection is becoming easier by the minute, this seems like a no-brainer: why waste time on uninformative univariate plots? Indeed,

nearly all the interesting worlds (physical, biological, imaginary, human) we seek to understand are inevitably multivariate in nature. [18, p.129]

Furthermore, as Tufte suggest,

the analysis of cause and effect, initially bivariate, quickly becomes multivariate through such necessary elaborations as the conditions under which the causal relation holds, interaction effects, multiple causes, multiple effects, causal sequences, sources of bias, spurious correlation, sources of measurement error, competing variables, and whether the alleged cause is merely a proxy or a marker variable (see for instance, [11]). [18, p.129]

While we should not dismiss low-dimensional evidence simply because it is low-dimensional, Tufte cautions that

reasoning about evidence should not be stuck in 2 dimensions, for the world we seek to understand is profoundly multivariate [*emphasis added*]. [18, p.130]

Analysts may question the ultimate validity of this principle: after all, doesn't *Occam's Razor* warn us that "it is futile to do with more things that which can be done with fewer"? This would seem to be a fairly strong admonition to not reject low-dimensional visualizations out of hand. This interpretation depends, of course, on what it means to "do with fewer": are we attempting to "do with fewer", or to "do with fewer"?

If it is the former, then we can produce simple charts to represent the data (which quickly balloons into a multivariate meta-display), but any significant link between 3 and more variables is unlikely to be shown, which drastically reduces the explanatory power of the charts.

If it is the latter, the difficulty evaporates: we simply retain as many features as are necessary to maintain the desired explanatory power.

Health and Wealth of Nations Only 4 variables are represented in the display, which we could argue just barely qualifies the data as multivariate. The population size seems uncorrelated with both of the axes' variates, unlike continental membership: there is a clear divide between the West, most of Asia, and Africa (see Figure 18.21).

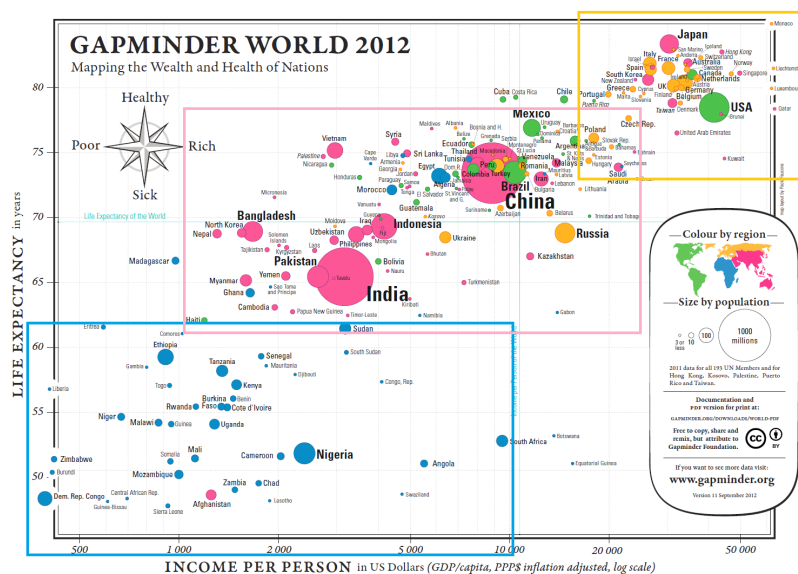


Figure 18.21: Potential outliers in the Gapminder chart.

This “clustering” of the world’s nations certainly fits with common wisdom about the state of the planet, which provides some level of validation for the display, but it not (by far) the **only way** to cluster the observations.

Another multivariate interpretation is afforded by Figure 18.18: countries in the upper right corner (whose location is given by 2 variables) are coloured differently (according to a third variable) than those in the lower left corner.

Other variables could also be considered or added, notably the year, allowing for bubble movement: one would expect that life expectancy and GDP per capita have both been increasing over time. The Gapminder Foundation’s [online tool](#) [↗](#) can build charts with other variates, leading to interesting inferences and suggestions.

18.2.4 Integration of Evidence

“Completely integrate words, numbers, images, diagrams.”
[18, p.131]

Data does not live in a vacuum. Tufte’s approach is clear:

the evidence doesn’t care what it is – whether word, number, image. **In reasoning about substantive problems, what matters entirely is the evidence, not particular modes of evidence** [*emphasis added*]. [18, p.130]

The main argument is that evidence from data is better understood when it is presented with context and accompanying meta-data. Indeed,

words, numbers, pictures, diagrams, graphics, charts, tables belong together [*emphasis added*]. Excellent maps, which are the heart and soul of good practices in analytical graphics, routinely integrate words, numbers, line-art, grids, measurement scales. [18, p.131]

Finally, Tufte makes the point that we should think of data visualizations and data tables as elements that provide vital evidence, and as such they should be integrated in the body of the text:

tables of data might be thought of as paragraphs of numbers, tightly integrated with the text for convenience of reading rather than segregated at the back of a report. [...] Perhaps the number of data points may stand alone for a while, so we can get a clean look at the data, although techniques of layering and separation may simultaneously allow a clean look as well as bringing other information into the scene. [18, p.131]³²

32: There is a flip side to this, of course, and it is that charts and displays should be annotated with as much text as is required to make their **context clear**.

When authors and researchers select a single specific method or mode of information during the inquiries, the focus switches from “can we explain what is happening?” to “can the method we selected explain what is happening?”

There is an art to **method selection**, and experience can often suggest relevant methods, but remember that “when all one has is a hammer,

everything looks like a nail”: the goal should be to use the necessary evidence to shed light on “what is happening”. If that goal is met, what modes of evidence were used is irrelevant.

Health and Wealth of Nations The various details attached to the chart (such as country names, font sizes, axes scale, grid, and world landmarks) provide substantial benefits when it comes to consuming the display. They become lost in the background, with the consequence of being taken for granted, but their presence is still crucial.

Case in point, compare the display obtained from the same data, but without integration of evidence in Figure 18.22 – who could reasonably guess what it is about without context?

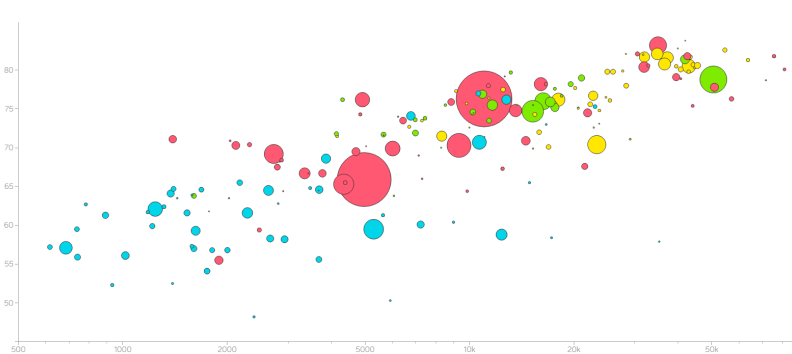


Figure 18.22: Non-integrated Gapminder chart.

18.2.5 Documentation

“Thoroughly describe the evidence. Provide a detailed title, indicate the authors and sponsors, document the data sources, show complete measurement scales, point out relevant issues.” [18, p.133]

We cannot always tell at a glance whether a pretty graphic speaks the truth or presents a relevant piece of information. Documented charts may provide a hint, as

the credibility of an evidence presentation depends significantly on the quality and integrity of the authors and their data sources. Documentation is an essential mechanism of quality control for displays of evidence. **Thus authors must be named, sponsors revealed, their interests and agenda unveiled, sources described, scales labeled, details enumerated** [*emphasis added*]. [18, p.132]

Depending on the context, questions and items to address could include:

- What is the title/subject of the visualization?
- Who did the analysis? Who created the visualization? (if distinct from the analyst(s))
- When was the visualization published? Which version of the visualization is rendered here?

- Where did the underlying data come from? Who sponsored the display?
- What assumptions were made during data processing and clean-up?
- What colour schemes, legends, scales are in use in the chart?

It is not obvious whether all this information can fit inside a single chart in some cases. But, keeping in mind the principle of integration of evidence, charts should not be presented in isolation in the first place, and some of the relevant information can be provided in the text, on a webpage, or in an accompanying document.

This is especially important when it comes to discussing the methodological assumptions used for data collection, processing, and analysis. An honest assessment may require sizable amounts of text, and it may not be reasonable to include that information with the display:³³

33: In that case, a link to the accompanying documentation should be provided.

publicly attributed authorship indicates to readers that someone is taking responsibility for the analysis; conversely, the absence of names signals an evasion of responsibility. [...] **People do things, not agencies, bureaus, departments, divisions** [*emphasis added*]. [18, pp.132-133]

Health and Wealth of Nations The Gapminder map might just be one of the best-documented charts in the data visualization ecosystem. Let us see if we can answer the questions suggested above.

- **What is the title/subject of the visualization?** The health and wealth of nations in 2012, using the latest available data (2011).
- **Who did the analysis? Who sponsored the display? Who created the visualization?** The analysis was conducted by the Gapminder Foundation; the map layout was created by Paulo Fausone. No data regarding the sponsors is found on the chart or in the documentation. It seems plausible that there were none.
- **When was the visualization published? Which version is rendered here?** The 11th version of the chart was published in September 2012.
- **Where did the underlying data come from? What assumptions were made during data processing and clean-up?** Typically, the work that goes into preparing the data is swept under the carpet in favour of the visualization itself; there are no explicit source of data on this chart, for instance. However, there is a URL in the legend box that leads to [detailed information](#). For most countries, life expectancy data was collected from:
 - the Human Mortality database;
 - the UN Population Division World Population Prospects;
 - files from historian James C. Riley;
 - the Human Life Table database;
 - data from diverse national statistical agencies;
 - the CIA World Fact book;
 - the World Bank, and
 - the South Sudan National Bureau of Statistics.

Benchmark 2005 GDP data was derived via regression analysis from International Comparison Program data for 144 countries, and extended to other jurisdictions using another regression against data from:

- the UN Statistical Division;
- Maddison Online;
- the CIA World Fact book, and
- estimates from the World Bank.

The 2012 values were then derived from the 2005 benchmarks using long-term growth rates estimate from:

- Maddison Online;
- Barro & Ursua;
- the UN Statistical Division;
- the Penn World Table (mark 6.2);
- the IMF’s World Economic Outlook database;
- the World Development Indicators;
- Eurostat, and
- national statistical offices or some other specific publications.

Population estimates were collated from:

- the UN Population Division World Population Prospects;
- Maddison Online;
- Mitchell’s International Historical Statistics;
- the UN Statistical Division;
- the US Census Bureau;
- national sources, and
- undocumented sources and “guesstimates”.

Exact figures for countries with a population below 3 million inhabitants were not needed as this marked the lower end of the chart resolution.

- **What colour schemes, legends, scales are in use in the chart?** The *Legend Inset* is fairly comprehensive (see Figure 18.23).

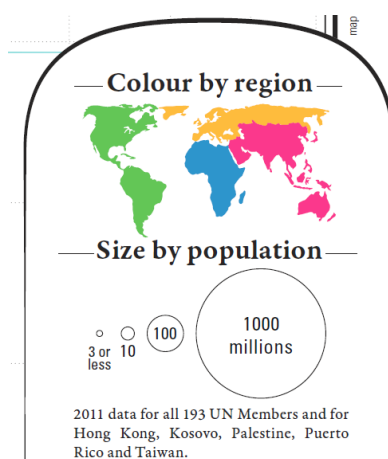


Figure 18.23: Legend inset for the Gap-minder chart.

Perhaps the last item of note is that the scale of the axes differs: life expectancy is measured linearly, but GDP per capita is measured on a logarithmic scale.

18.2.6 Content First and Foremost

“Analytical presentations ultimately stand or fall depending on the quality, relevance, and integrity of their content.” [18, p.136]


Any amount of time and money can be spent on graphic designers and focus groups, but

the most effective way to improve a presentation **is to get better content** [*emphasis added*] [...] design devices and gimmicks cannot salvage failed content. [...] The first questions in constructing analytical displays are not “How can this presentation use the color purple?” Not “How large must the logotype be?” Not “How can the presentation use the Interactive Virtual Cyberspace Protocol Display Technology?” Not decoration, not production technology. The first question is “**What are the content-reasoning tasks that this display is supposed to help with?**” [*emphasis added*] [18, p.136]

The main objective is to produce a compelling narrative, which may not necessarily be the one that was initially expected to emerge from a solid analysis of sound data. Simply speaking, the visual display should assist in explaining the situation at hand and in answering the original questions that were asked of the data.

Health and Wealth of Nations How would we answer the following questions:

- Do we observe similar patterns every year?
- Does the shape of the relationship between life expectancy and log-GDP per capita vary continuously over time?
- Do countries ever migrate large distances in the display over short periods?
- Do exceptional events affect all countries similarly?
- What are the effects of secession or annexation?

The 2012 Health and Wealth of Nations data represent a single datum in the general space of data visualizations; in this context, getting better content means getting data for other years, as well as for 2012 (such as in Figure 18.24 and the [Gapminder Tools](#) .)

Are the same countries still outlying observations? Is the relationship between life expectancy and GPD per capita still linear? Are the country groupings the same? Is the bottom right quadrant still empty?

Is that story **more or less** similar? Is it **significantly** different?

It should be noted that Tufte’s views are not shared by every practitioner. We mentioned it in Section 18.1.2, but it bears repeating: the name of the game is **conveying insights** – any visual that helps with that is on the table. It is true that if a chart satisfies Tufte’s principles, the chance that it will convey insights is good. But it is not necessarily a **requirement**.

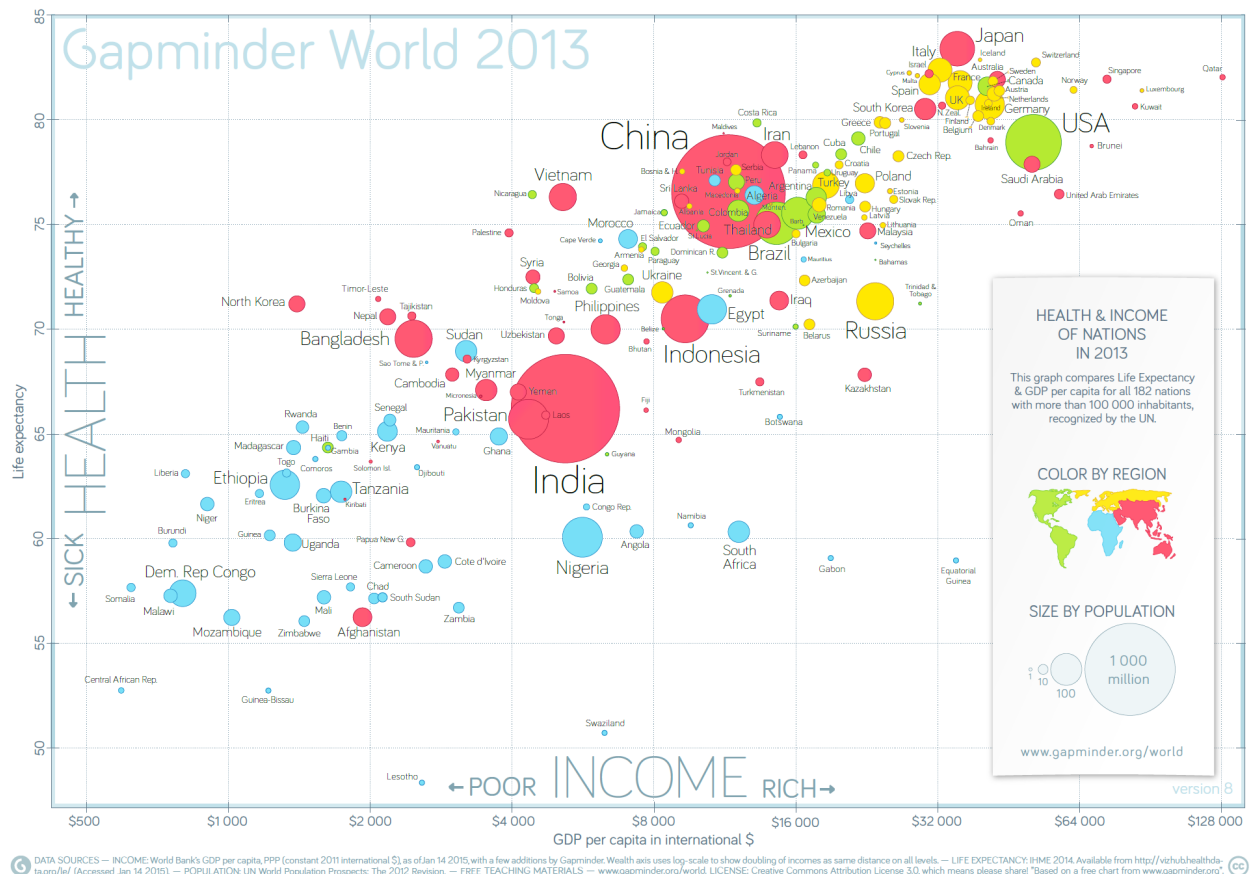


Figure 18.24: Life expectancy and income per capita in 2013, by nation (Gapminder Foundation www.gapminder.org).

18.3 Introduction to Dashboards

Dashboards are a helpful way to **communicate** and **report** data. They are versatile in that they support multiple types of reporting. Dashboards are predominantly used in business intelligence contexts, but they are being used more frequently to communicate data and visualize analysis for non-business services also. Popular dashboarding platforms include Tableau, and Power BI, although there are other options, such as Excel, R + Shiny, Geckoboard, Matillion, JavaScript, etc.

These technologies aim to make creating data reports as simple and user-friendly as possible. They are intuitive and powerful; creating a dashboard with these programs is quite easy, and there are tons of how-to guides available online [9, 10, 20].

In spite of their ease of use, however, dashboards suffer from the same limitations as other forms of data communication, to wit: how can results be **conveyed effectively** and how can an **insightful data story** be relayed to the desired audience? Putting together a “good” dashboard is more complicated than simply learning to use a dashboarding application.

18.3.1 Dashboard Fundamentals

Effective dashboarding requires that the designers answer questions about the planned-for display:

- who is the target audience?
- what value does the dashboard bring?
- what type of dashboard is being created?

Answering these questions can guide and inform the visualization choices that go into creating dashboards.

Selecting the **target audience** helps inform data decisions that meet the needs and abilities of the audience. When thinking of an audience, consider their **role** (what decisions do they make?), their **workflow** (will they use the dashboard on a daily basis or only once?), and **data expertise level** (what is their level of data understanding?).

When creating a dashboard, it's important to understand (and keep in mind) why one is needed in the first place – does it find **value** in:

- helping managers make decisions?
- educating people?
- setting goals/expectations?
- evaluating and communicating progress?

Dashboards can be used to communicate numerous concepts, but not all of them can necessarily be displayed in the same space and at the same time so it becomes important to know where to direct the focus to meet individual dashboard goals. Dashboard decisions should also be informed by the **scope**, the **time horizon**, the required **level of detail**, and the dashboard's **point-of-view**. In general,

- the **scope** of the dashboard could be either broad or specific – an example of a broad scope would be displaying information about an entire organization, whereas a specific scope could focus on a specific product or process;
- the **time horizon** is important for data decisions – it could be either historical, real-time, snapshot, or predictive:
 - **historical** dashboards look at past data to evaluate previous trends;
 - **real-time** dashboards refresh and monitor activity as it happens;
 - **snapshot** dashboards show data from a single time point, and
 - **predictive** dashboards use analytical results and trend-tracking to predict future performances;
- the **level of detail** in a dashboard can either be high level or drill-able – **high level** dashboards provide only the most critical numbers and data; **drill-able** dashboards provide the ability to “drill down” into the data in order to gain more context.
- the dashboard **point of view** can be prescriptive or exploratory – a **prescriptive** dashboard prescribes a solution to an identified problem by using the data as proof; an **exploratory** dashboard uses data to explore the data and find possible issues to be tackled.

The foundation of good dashboards comes down to deciding what information is most important to the audience in the context of interest; such dashboards should have a **core theme** based on either a **problem to solve** or a **data story to tell**, while removing extraneous information from the process.

18.3.2 Dashboard Structure

The dashboard structure is informed by four main considerations:

- **form** – format in which the dashboard is delivered;
- **layout** – physical look of the dashboard;
- **design principles** – fundamental objectives to guide design,
- **functionality** – capabilities of the dashboard.

Dashboards can be presented on paper, in a slide deck, in an online application, over email (messaging), on a large screen, on a mobile phone screen, etc.

Selecting a **format** that suits the dashboard needs is a necessity; various formats might need to be tried before arriving at a final format decision.

The structure of the dashboard itself is important because visuals that tell similar stories (or different aspects of the same story) should be kept close together, as **physical proximity of interacting components** is expected from the viewers and consumers. Poor structural choices can lead to important dashboard elements being undervalued.

The dashboard shown in Figure 18.25 provides an example of **group visuals** that tell similar stories.³⁴

34: The corresponding Power BI file can be found on the [Data Action Lab](#) website).

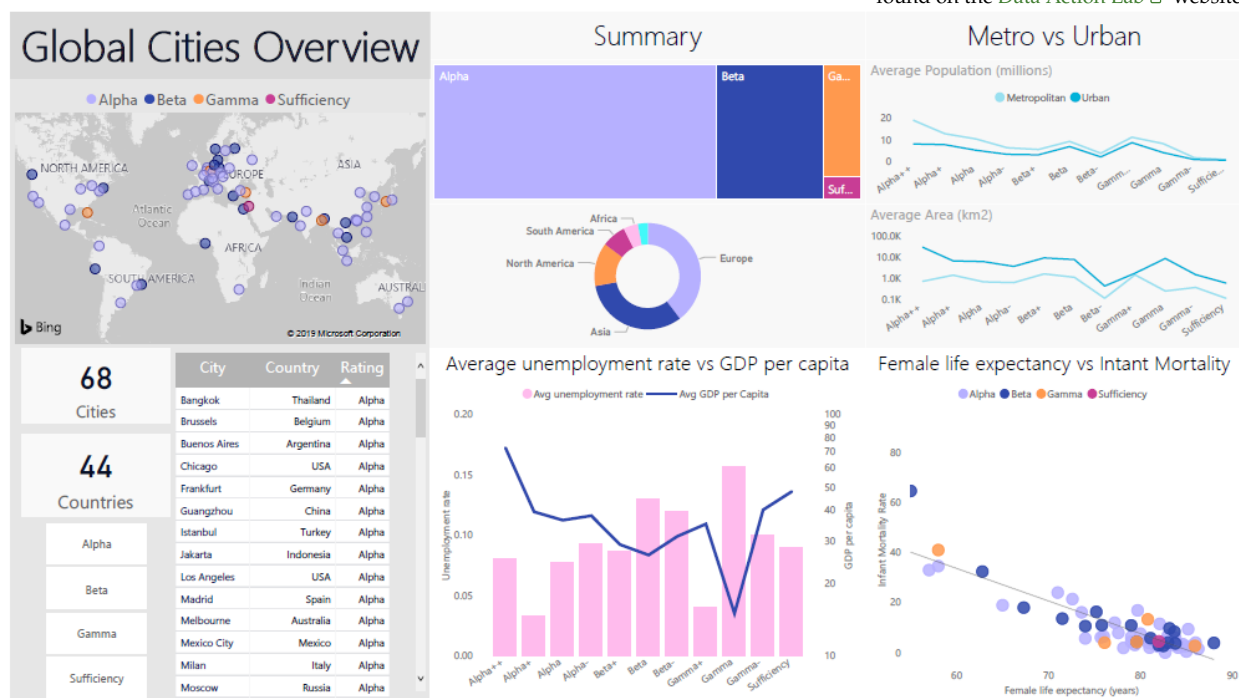


Figure 18.25: An exploratory dashboard showing metrics about various cities ranked on the Global Cities Index. The dashboard goal is to allow a general audience to **compare and contrast** the various globally ranked cities – statistics that contribute to a ‘higher’ ranking immediately pop out. Viewers can also very easily make comparisons between high- and low-ranking cities. The background is kept neutral with a fair amount of blank space in order to keep the dashboard open and easy to read. The colours complement each other (*via* the use of a colour theme picker in Power BI) and are clearly indicative of ratings rather than comparative statistics (personal file).

Knowing which visual displays to use with the “right” data helps dashboards achieve structural integrity:

- **distributions** can be displayed with **bar charts** and **scatter plots**;
- **compositions** with **pie charts**, **bar charts**, and **tree maps**;

- **comparisons** use **bubble charts** and **bullet plots**, and
- **trends** are presented with **line charts** and **area plots**.

An interesting feature of dashboard structure is that it can be used to guide **viewer attention**; critical dashboard elements can be highlighted with the help of visual cues such as use of **icons**, **colours**, and **fonts**. Using **filters** is a good way to allow dashboard viewers of a dashboard to customize the dashboard scope (to some extent) and to investigate specific data categories more closely.

The dashboard shown in Figure 18.26 provides an example of a dashboard that makes use of an interactive filter to analyze data from specific categories.

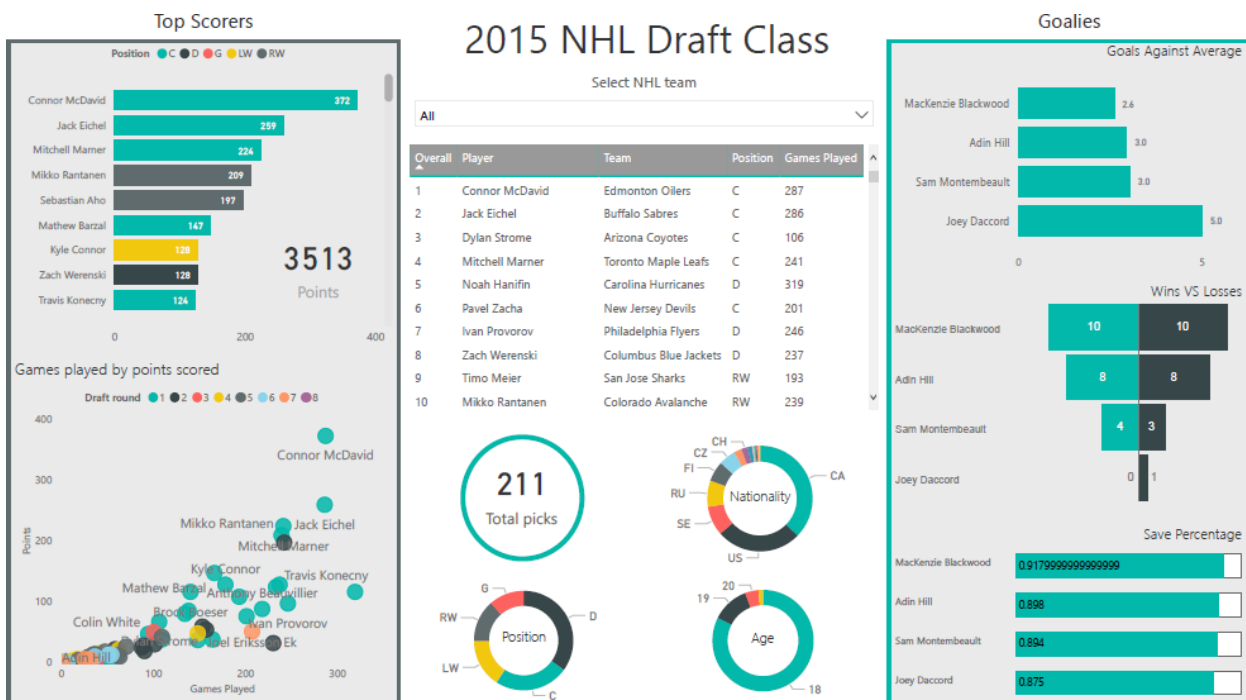


Figure 18.26: An exploratory dashboard showing information about the National Hockey League draft class of 2015. The dashboard displays professional statistics (as of August 2019) of hockey players drafted into the NHL in 2015, as well as their overall draft position. This dashboard allows **casual hockey fans** to **evaluate the performance** of players drafted in 2015. It provides demographic information to give context to possible market deficiencies during this draft year (i.e., defence players were drafted more frequently than any other position). This dashboard is designed to be interactive; the filter tool at the top allows dashboard viewers to drill-down on specific teams (personal file).

18.3.3 Dashboard Design

An understanding of design improves dashboards; **dissonant** designs typically make for poor data communication. Design principles are discussed in [19, 18, 16, 2, 14]. For dashboards, the crucial principles relate to the use of **grids**, **white space**, **colour**, and **visuals**. When laying out a dashboard, **gridding** helps direct viewer attention and makes the space easier to parse.³⁵

In order to help viewers avoid becoming overwhelmed by clutter or information overload, consider leaving enough **blank space** around and within the various charts.³⁶ In general, clutter shuts down the communication process – Figure 18.27 provides two impressive breakdown examples.

35: The various visuals are **aligned** in a grid format to lay the data out in a clean, readable manner in Figure 18.25.

36: While the dashboard of Figure 18.26 displays a lot of information, there is a lot of blank/white space between the various visuals, which provides viewers with space to breathe.



Figure 18.27: Anonymous ‘ugly’ dashboards [12, 8]; how about these data communication breakdowns?

Colour provides meaning to data visualizations – bright colours, for instance, should be used as alarms as they draw the viewer’s attention. Colour themes create cohesiveness, improving the overall readability of a dashboard.³⁷

That being said, dashboards that are **elegant** (as well as **truthful** and **functional**) will deliver a bigger bang for their buck [2, 3]. In the same vein, keep in mind that all dashboards are by necessity **incomplete**.³⁸

Finally, designers and viewers alike must remember that a dashboard can **only be as good as the data it uses**; a dashboard with badly processed or unrepresentative data, or which is showing the results of poor analyses, cannot be an effective communication tool, independently of design.

37: There are no perfect dashboards – no collection of charts will ever suit everyone who encounters it.

38: A good dashboard may still lead to dead ends, but it should allow its users to ask: “Why? What is the root cause of the problem?”

18.3.4 Examples

Dashboards are used in varied contexts, such as:

- interactive displays that allows people to explore motor insurance claims by city, province, driver age, etc.;
- a PDF file showing key audit metrics that gets e-mailed to a Department’s DG on a weekly basis;
- a wall-mounted screen that shows call centre statistics in real-time;
- a mobile app that allows hospital administrators to review wait times on an hourly- and daily-basis for the current year and the previous year; etc.

The Ugly While the previous dashboards all have some strong elements, it is a harder to be generous for the two examples provided in Figure 18.27.³⁹ The first of these is simply “un-glanceable” and the overuse of colour makes it unpleasant to look at; the second one features 3D visualizations (rarely a good idea), distracting borders and background, lack of filtered data, insufficient labels and context, among others.

39: Is it easy to figure out, at a glance, who their audience is meant to be? What are their strengths (do they have any)? What are their limitations? How could they be improved? What can they be used for?

The Good Good dashboards, on the other hand, simply breathe. The number of charts on each page is small, boxes are eschewed, simple colour schemes are preferred, and the canvas is quiet (see Figures 18.28 and 18.29, for instance). We discuss topics relating to data storytelling and design principles in [1].

Golden Rules In a (deleted) blog article, N. Smith posted his Golden Rules:

- **consider the audience** (who are you trying to inform? does the DG really need to know that the servers are operating at 88% capacity?);
- **select the right type of dashboard** (operational, strategic/executive, analytical);
- **group data logically, use space wisely** (split functional areas: product, sales/marketing, finance, people, etc.);
- **make the data relevant to the audience** (scope and reach of data, different dashboards for different departments, etc.);
- **avoid cluttering the dashboard** (present the most important metrics only), and
- **refresh your data at the right frequency** (real-time, daily, weekly, monthly, etc.).

With dashboards, as with data analysis and data visualization in general, there is no substitute for **practice**: the best way to become a proficient builder of dashboards is to . . . well, to go out and build dashboards, try things out, and, frequently, to stumble and learn from the mistakes.

18.4 Exercises

You may wish to consult [1] (in particular, Chapters 11-13) for instructions on how to use R, ggplot2, and/or Power BI for this chapter's exercises.

1. Find examples of data presentations that you consider to be particularly insightful and/or powerful. Discuss their strengths and weaknesses.
2. Find examples of data presentations that you consider to be particularly misleading and/or useless. Discuss their strengths and weaknesses.
3. How do you think new technologies (e.g. virtual or augmented reality, 3D-printing, wearable computing) will influence data presentations?
4. Consider the following datasets:
 - [GlobalCitiesPBI.csv](#)
 - [2016collisionsfinal.csv](#)
 - [polls_us_election_2016.csv](#) , and
 - [HR_2016_Census_simple.xlsx](#) .
 - a) Create a data dictionary for each dataset. Establish a list of variables that you think are crucial to a good understanding of the dataset. Justify your choices.
 - b) Create (at least) 5 bivariate/univariate visualizations that can help you understand each dataset.

- c) Produce (at least) 3 “definitive” visualizations for each dataset. Use the principles discussed in class (including documentation, legends, annotations, Multiple I’s, etc.). Emphasis should be placed on content AND on presentation (suggestions: consider creating a reasonably high number of charts using a random selection of a random number of variables in order to minimize the odds of missing out on useful information).
5. Repeat the previous question with any dataset of your liking.
 6. Identify a scenario for which a dashboard could prove useful. Determine specific questions that the dashboard could help answer or insights that it could provide. Identify data sources and data elements that could be fed into your dashboard. Design a display (with pen and paper) with mock charts. What are the strengths and limitations of your dashboard? Is it functional? Elegant?
 7. The remaining exercises use the [Gapminder Tools](#) (there is also an [offline version](#)).
 - a) At what point in the data science workflow do you think that visualizations of this nature could be useful?
 - b) What are the ways in which observations could be anomalous? Have you found any such anomalies? Do you have explanations for them? In particular, consider the case of South Africa in 2012, which appears to be a clear outlier. Follow the path of the South African bubble from 1975 to 2020, in relation to the general pattern. Does the apartheid/income inequity explanation suggested in the text still make sense?
 - c) Pick 2+ “definitive” visualizations (methods, variables, etc.) other than the default configuration. What are some important insights?
 - d) How would you describe the insights of step 3 without resorting to visual vocabulary?
 - e) Can you think of ways in which the data of interest to you in your day-to-day activities could benefit from the same treatment? What situations could you explore in such a scenario? How would that help your team better understand the system under consideration?
 8. Consider the following Australian population figures, by state (in 1000s):
 - a) Graph the New South Wales (NSW) population with all defaults using `plot()`. Redo the graph by adding a title, a line to connect the points, and some colour.
 - b) Compare the population of New South Wales (NSW) and the Australian Capital Territory (ACT) by using the functions `plot()` and `lines()`, then add a legend to appropriately display your graph.
 - c) Use a bar chart to graph the population of Queensland (QLD), add an appropriate title to your graph, and display the years from 1917 to 2017 on the appropriate bars.
 - d) Create a light blue histogram for the population of South Australia (SA) over the years. Does this chart make sense?

Year	NSW	Vic.	Qld	SA	WA	Tas.	NT	ACT	Aust.
1917	1904	1409	683	440	306	193	5	3	4941
1927	2402	1727	873	565	392	211	4	8	6182
1937	2693	1853	993	589	457	233	6	11	6836
1947	2985	2055	1106	646	502	257	11	17	7579
1957	3625	2656	1413	873	688	326	21	38	9640
1967	4295	3274	1700	1110	879	375	62	103	11799
1977	5002	3837	2130	1286	1204	415	104	214	14192
1987	5617	4210	2675	1393	1496	449	158	265	16264
1997	6274	4605	3401	1480	1798	474	187	310	18532
2007	6889	5205	4182	1585	2106	493	215	340	21017
2017	7861	6324	4928	1723	2580	521	246	410	24599

Chapter References

- [1] P. Boily, S. Davies, and J. Schellinck. *The Practice of Data Visualization* [↗](#). Data Action Lab, 2023.
- [2] A. Cairo. *The Functional Art*. New Riders, 2013.
- [3] A. Cairo. *The Truthful Art*. New Riders, 2016.
- [4] Robert A. Dahl. 'Cause and Effect in the Study of Politics'. In: *Cause and Effect*. Ed. by Daniel Lerner. New York: Free Press, 1965, pp. 75–98.
- [5] P. Dragicevic and Y. Jansen. *List of Physical Visualizations and Related Artifacts* [↗](#).
- [6] T. Elms. *Lexical Distance of European Languages* [↗](#). Etymologikon, 2008.
- [7] Stephanie Evergreen. *Effective Data Visualization: the Right Chart for the Right Data*. Second edition. Thousand Oaks, California: SAGE Publications, Inc.
- [8] Geckoboard.com. 'Two Terrible Dashboard Examples [↗](#)'. In: ().
- [9] Z. Gemignani and C. Gemignani. *Data Fluency: Empowering Your Organization with Effective Data Communication*. Wiley, 2014.
- [10] Z. Gemignani and C. Gemignani. *A Guide to Creating Dashboards People Love to Use* [↗](#). (ebook).
- [11] A.B. Hill. 'The environment and disease: association or causation?' In: *Proc R Soc Med* 58.5 (1965), pp. 295–300.
- [12] Matillion.com. 'Poor Use of Dashboard Software [↗](#)'. In: ().
- [13] I. Meirelles. *Design for Information*. Rockport, 2013.
- [14] I. Meirelles. *Design for Information : an Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations*. Rockport, 2013.
- [15] Damian Mingle. <https://twitter.com/DamianMingle/status/655534652833288192> [↗](#).
- [16] C. Nussbaumer Knaflc. *Storytelling with Data*. Wiley, 2015.
- [17] H. Rosling. *The Health and Wealth of Nations* [↗](#). Gapminder Foundation, 2012.
- [18] E. Tufte. *Beautiful Evidence*. Graphics Press, 2008.
- [19] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.
- [20] S. Wexler, J. Shaffer, and A. Cotgreave. *The Big Book of Dashboards*. Wiley, 2017.
- [21] N. Yau. *FlowingData* [↗](#).
- [22] N. Yau. *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley, 2011.

Course Metrics

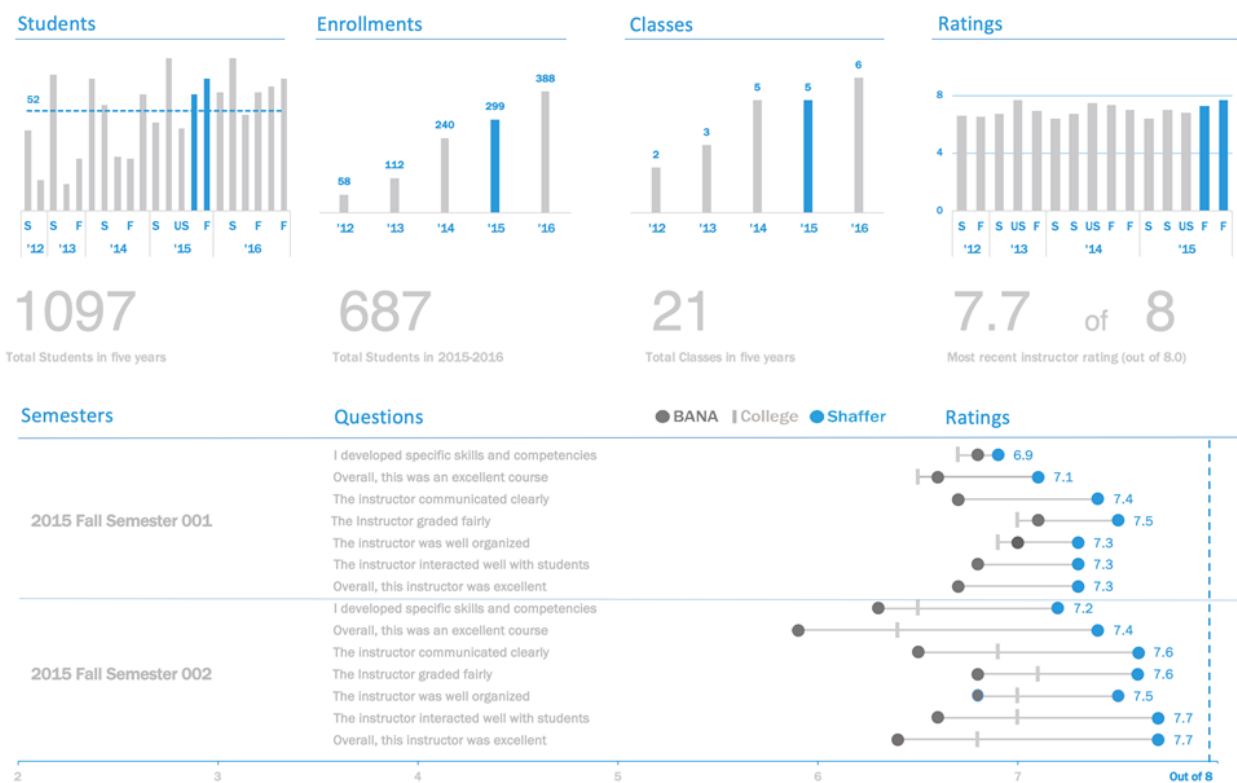


Figure 18.28: 'Zen' dashboard: course evaluations at the University of Cincinnati [1].

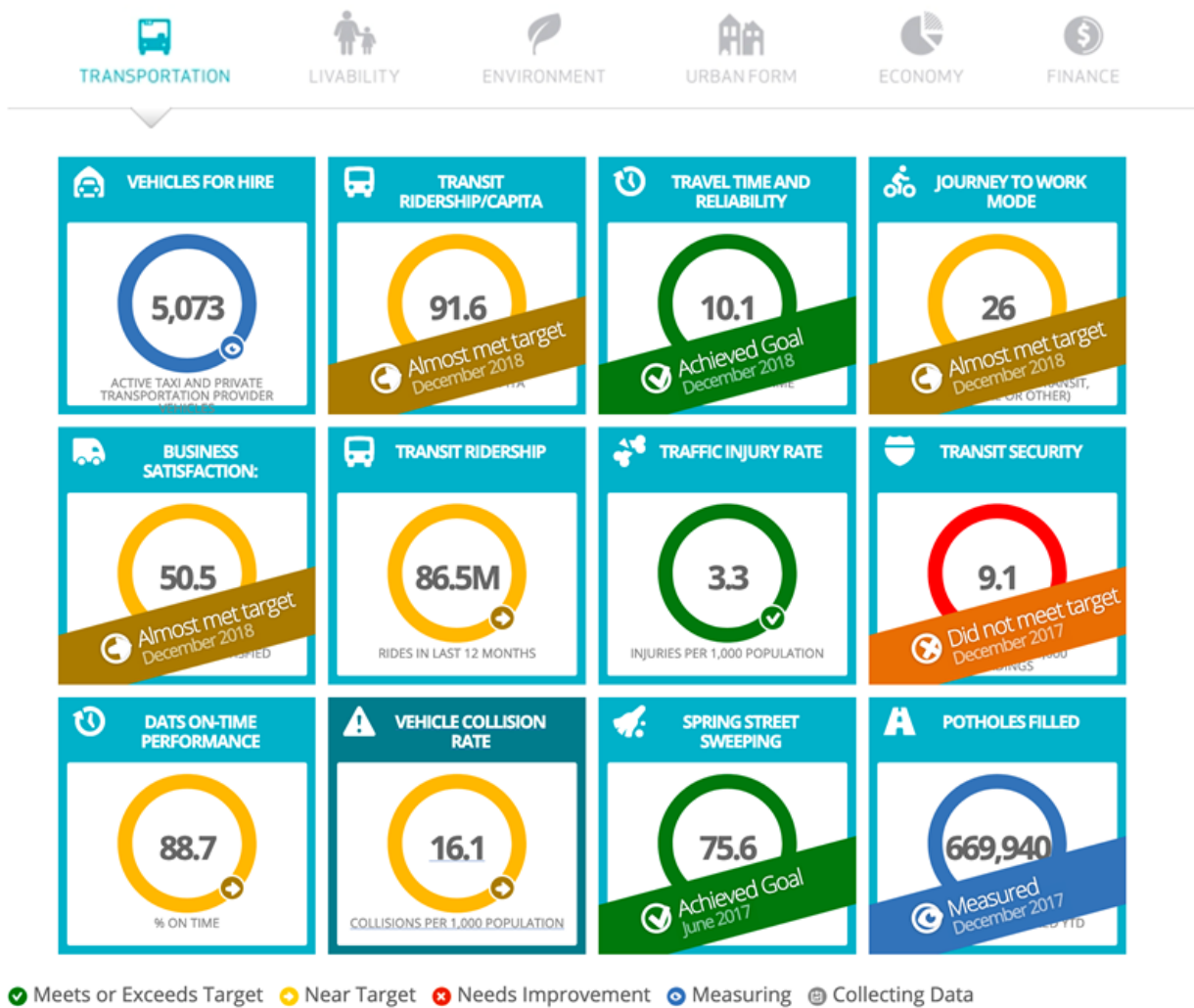


Figure 18.29: 'Zen' dashboard: former dashboard of the City of Edmonton.

Data Understanding, Data Analysis, and Data Science

Volume 2 – Fundamentals of Data Insight

Build on your data science foundation with **Fundamentals of Data Insight**, the second volume in the **Data Understanding, Data Analysis, and Data Science** series. This volume shifts the focus from technical execution to the broader context in which data work occurs, including collaboration, communication, ethics, and decision-making.

After exploring these non-technical dimensions, readers move into essential practical skills, such as data preparation, web scraping, automated data collection, data engineering, and data management. The volume concludes with techniques for data exploration and visualization, incorporating perspectives from contributors with varied experience.

These chapters emphasize thoughtful interpretation over mechanical procedure and highlight the importance of understanding both data and context. Whether used as a companion to lectures or for self-guided study, **Fundamentals of Data Insight** encourages a reflective and well-rounded approach to working with data.

About the Author

Patrick Boily is an Assistant Professor in the Department of Mathematics and Statistics at the University of Ottawa. He earned his Ph.D. in Mathematics in 2006 and is the author of seven textbooks on mathematics, statistics, and data science, available at idlewyldanalytics.com.

Since 1999, he has taught more than 75 courses at the University of Ottawa, the Université du Québec en Outaouais, and Carleton University. From 2008 to 2012, he served as a federal public servant, contributing to several projects including the award-winning Canadian Vehicle Use Study. From 2012 to 2019, he launched and managed Carleton University's Centre for Quantitative Analysis and Decision Support (CQADS), and he is a founding member of the Data Action Lab, which offers workshops, short courses, and consulting services in data analysis.

Patrick's academic work focuses on the application of mathematics and statistics to evidence-based decision support. He has provided consulting services to a wide range of public and non-profit organizations, including United Way, the Public Health Agency of Canada, the Canadian Air Transport Security Authority, and the Department of National Defence. His areas of expertise include operations research, data science and predictive analytics, stochastic modelling, and simulation.

Patrick is an avid hockey player, cross-country skier, cyclist, mountain biker, and swimmer; he enjoys crosswords, playing the guitar, and watching British murder mysteries. He lives with his family in Wakefield, Quebec.

