

## 10 Visualization Software

Some of the most famous visualizations – Nightingale’s roses, the March to Moscow – were created meticulously and carefully **by hand**. The quantities being represented in the chart had to be carefully measured on the paper to ensure that the **correspondence** between the data and the spacing (shape, size, lengths, colours, angles, etc.) of the lines (markers, regions, etc.) making up the visualization was **correct**.<sup>1</sup>

As it has with so many things, the advent of **software** has clearly changed this process. Now we can create data visualizations with the push of a button, and have them **instantly update** themselves as the data itself is updated. This has been beneficial to the practice of data visualization, enabling **more practitioners** to venture into what was once rarified territory.

It bears repeating: the **democratization of data** is a good thing. But creating good visualizations is not solely a function of the software;<sup>2</sup> the **choice of tool** is not entirely irrelevant.<sup>3</sup> But data visualization software comes and goes – what was popular 5 years ago is nearly obsolete now, and what is popular now is unlikely to still be the **go-to tool** in 5 years.

In this chapter, we will briefly review some of the visualization tools frequently used by practitioners.<sup>4</sup>

### 10.1 GUI Tools

Both **GUI** (graphical user interface) tools and **command line** (programmatic) tools can create powerful and valuable data visualizations. GUI tools are arguably more **approachable** as they do not require the knowledge of any **special computer languages** in order to produce visualizations. They also allow novice data visualizers to ease their way into data visualizations without needing to worry about many of the **nuanced aspects** of data visualization right off the bat.

Some GUI tools are designed to be **relatively simple** to use.<sup>5</sup> After gaining experience, data visualizers often grow out of such tools. Some GUI tools, conversely, allow users to delve as **deeply** into the nuances of data visualization as they wish – assuming that they are willing to **click through all of the relevant menus and settings**.

10.1 GUI Tools . . . . .	199
Excel . . . . .	200
SaaS Visualization Apps . .	200
Tableau . . . . .	200
Power BI . . . . .	200
10.2 Command Line Tools . . . .	201
R . . . . .	201
Python . . . . .	201
10.3 Future Tools . . . . .	202
Virtual Reality . . . . .	202
AI-Aided Visualizations . .	202

1: Or at the very least, to ensure that the representation was **coherent**.

2: Just as taking good pictures is not only a function of the camera.

3: Good data visualizers will know the ins and outs, and the strengths and weaknesses of their preferred tools, and how to best showcase data visualization principles, and when other tools are called for.

4: Not with the intent to demonstrate how to use them *per se*, but rather as illustrations of the current state of the software. If a top-10 favourite does not appear in the list, it does not mean that it is not a good tool; more likely, we are simply too old and set in our way (or “tool agnostic”, more generously) to dive in and learn new paradigms every year. *Mea culpa*.

5: In the same way that a point and shoot cell phone camera is designed to be simple to use yet still reliably produce in-focus pictures of people and objects in typical settings.

## Excel

We could not write a chapter on data visualization software without mentioning Excel, often the first software application many analysts learn to use when working and visualizing data.

The default visualization functionality of Excel is quite basic and relatively easy to use: users select **spreadsheet data** and select from a list of **several visualization options**, leaving the rest in the hands of the software. The default visual style of Excel – choice of **colours**, **fonts**, and other **graphic design elements and styles** – leaves a lot to be desired, however.<sup>6</sup>

Under some circumstances,<sup>7</sup> Excel might do the trick. But half-serious data visualizers will quickly outgrow the basic Excel functionality and wish to move on to more dedicated visualization tools. This is not to say that Excel **power users** cannot coax some extremely impressive data visualization results out of the tool, but the time required to achieve such results is not trivial.<sup>8</sup>

## SaaS Visualization Apps

A new crop of online applications (**Software as a Service**, or SaaS) are providing learning data visualizers with new options. As an example of this type of GUI tool, consider **RawGraphs** [↗](#), a free and open source tool.

These tools frequently focus on providing quick **semi-automated** or **guided generation** of data visualizations which can then be downloaded in one of several common image formats. Some of them even have the capability to **save and work** on the same visualization over time; note that they may lack the options required to **customize** or **finesse** the final product appearance. However, they can be useful as a rapid prototyping option for data exploration and to create a first pass of a data visualization.<sup>9</sup>

## Tableau

One of the front runner data visualization applications released at the turn of the century, **Tableau** [↗](#) is a sophisticated **commercial** GUI based data visualization application, and one of the primary data visualization tools used by data practitioners and scientists. [11, 118]

## Power BI

**Power BI** [↗](#) was originally envisioned by Microsoft as a **business intelligence** tool; while visualization is one of the tool's critical **functional elements**, the broader focus means that Power BI can also serve as a desktop-friendly version of a **data pipeline**, facilitating what we refer to as **desktop data science**. In this context, the visualization component of the tool<sup>10</sup> serves as the **endpoint** of the pipeline, following the **collection**, **preparation**, and

6: An Excel chart made with the default settings can be sniffed out several kilometres away.

7: When conducting **exploratory data analysis** or making a **quick internal presentation** to team members, say.

8: See Depict Data Studio's [39+ Great Graphs You Can Make in Microsoft Excel](#) [↗](#) for some examples.

9: One caveat about SaaS tools is that users must be mindful of the **privacy** of their data. Some openly state that users of the app should not upload sensitive data, while others have different versions (e.g., **free** or **paid**) that have different data usage policies.

10: In which visualizations are embedded in a larger dashboard environment.

**analysis** of data flowing in to pipeline from other sources – these dashboards are designed to update as the data is **updated**.

Power BI dashboards can also fairly easily be **published** and made available to various stakeholder groups; its focus on visualizations in the context of dashboards, which are themselves generated to provide **organizational insight**, makes it an ideal tool for **data storytelling**.<sup>11</sup> We present a short introduction to the tool in Chapter 13 (*Power BI for Beginners*).<sup>12</sup>


## 10.2 Command Line Tools

The learning curve associated with using **command line** tools – in particular the need to learn how to use a **scripting** or **programming language** – is outweighed by the increase in **repeatability**, **automation**, **precision**, and **sophistication** that is afforded by these tools.

Both Tableau and Power BI allow users to embed scripts (e.g., R or Python) into the tool; the latter provides its own languages (M and DAX) to further automate pipeline and visualization actions. Programming languages usually provide **libraries** dedicated to data visualization (and animation) and dashboards.<sup>13</sup>



In keeping with our agnostic position regarding software, we recommend that data visualization practitioners expand their horizons by **experimenting** with various GUI and non-GUI visualization tools.<sup>14</sup>

### R

**R**  is a programming language developed to support **statistical analysis**. It has built-in **data structures** and **functionality** not typically seen in more multi-purpose languages. The base R installation comes equipped with a default graphics package that allows for the creation of **simple** (un-aesthetic) charts (see Chapter 11, *Basic Visualizations in R*); almost all packages include **default plot behaviours** for at least some of their objects. [119, 120]


Over time, even more sophisticated approaches to data visualization have been developed within the R universe. One of these, *ggplot2*, will be the focus of Chapter 12 (*ggplot2 Visualizations in R*); the package functions and syntax are based on Wilkinson's **grammar of graphics** data visualization conceptual framework (see [54, 120] and Section 5.4.)

### Python


**Python**  is a popular **multi-purpose programming language** with the data science community. As with R, a number of its modules and packages are specifically devoted to data visualization.<sup>15</sup> Recently, the grammar of graphics has also been implemented in the Python package *plotnine* .<sup>16</sup>


11: The desktop version is free, as of this writing (June 2023).

12: With the caveat that nothing becomes obsolete quicker than *How-To* manuals for software – as soon as the last instruction is laid out on the page, a new update is pushed and makes it unusable. Consult at your own risk.

13: Such as *ggplot2*, *seaborn*, *plotnine*, *gganimate*, *plotly*, *d3py*, *shiny* , *flask*, and so on.

14: Rather than focusing on a particular package or language, although we also recommend reaching expert-level status in at least one framework.

15: Special mentions go to *matplotlib* and *seaborn* .

16: Python also has a number of popular dashboard (and dashboard-like) creation packages, which allow data visualizations and dashboards to be readily attached to applications (see *flask*  and [121], for instance).

## 10.3 Future Tools

We would not be surprised if this chapter became **out-of-date** as soon as we finish writing it – there are many exciting developments on the horizon, including virtual reality and AI-aided visualizations.

### Virtual Reality

**Virtual reality** (VR) has long been touted as “the Next Big Thing” in computing circles, and has **unsuccessfully** [↗](#) tried to break into the mainstream on several occasions. We are in the midst of a renewed effort; whether this will be the moment when VR, along with its sibling **augmented reality** (AR), becomes widely adopted is not clear yet, but interesting developments are occurring in the data visualization space in connection to both VR and AR.

Although we stated fairly unequivocally that 3D visualizations **were to be avoided** in a 2D context, VR and AR may provide a more natural environment for such visualizations.<sup>17</sup> Consider, for instance, **Flow Immersive** [↗](#), which allows users to experience and interact with data presentations, or **flows**.<sup>18</sup>

17: In these framework, it may be possible to interact more directly with 3D charts.

18: Both static and animated.

### AI-Aided Visualizations

Finally, with the advent of **language-focused** AI interfaces supported by **large language models** (LLMs) and the ability to **generate** images based on text prompts, it is possible that our experience with data visualization is about to **change dramatically**.<sup>19</sup>

We suspect that there will be a synergy between changes to computer interfaces (GUIs may become outmoded, say) and changes to the type of support that these technologies can provide for analyzing data and creating associated visualizations. This may result in making the creation of data visualizations much **more accessible** to a broader segment of the population, as well as to the creation of new types of (and uses for) data visualizations.

Consequently, we encourage data practitioners to be **proactive** in tracking and testing out new tools in this field as they develop. But we would be remiss if we did not at least discuss some of the recent advances in **deep learning techniques** that support data visualization, as well as new conversational interfaces for doing data visualization and data analysis.

As mentioned above, it has become increasingly easy to use **conversational chatbots interfaces**, supported by LLM, to create data visualizations and perform data analysis. **ChatGPT** [↗](#) by OpenAI is an example of this type of technology.<sup>20</sup>

Typically, text-based AI models do not have direct access to file uploads or external storage. However, it is possible to provide the dataset to the chat interface by copying and pasting the data in a structured format. If the dataset is too large or complex to paste directly, another option is to provide a summary or a sample of the data, or work with a subset of the dataset.<sup>21</sup>

19: Our interaction with these tools suggest that we are not quite there yet... for all we know, we are running headlong into a 3rd **AI Winter** [↗](#).

20: This is a rapidly-evolving space, in which ChatGPT is only one example of available applications. But in keeping with this theme, the text that follows in this section is a collaborative effort between the authors and ChatGPT – all conversations took place in June 2023.

21: We can in fact point GPT to a data source, see Kyle Hill’s **ChatGPT Explained Completely** [↗](#).



**Patrick:** ChatGPT, I'd like to test your ability to produce charts based on data. Is this something that you can do?



**ChatGPT:** Absolutely! I can assist you in generating charts based on data. Please provide me with the data you would like to visualize, along with any specific requirements or preferences you have for the chart type.



How do I upload a dataset to make it accessible to you?



As a text-based AI model, I don't have direct access to file uploads or external storage. However, you can simply provide the dataset to me by copying and pasting the data in a structured format. If the dataset is too large or complex to paste directly, you can provide a summary or a sample of the data, or we can work with a subset of the dataset.

For example, if your dataset is in a tabular format like a CSV file, you can provide the column headers and a few rows of data. Alternatively, you can describe the structure and content of the dataset, and I can work with that information to generate charts.

Feel free to share the details or sample data, and let me know the specific type of chart you'd like me to create.



The dataset contains 5 columns: Type, Age, Rural, VKM.q, CC.q.



Thank you for providing the column headers. To proceed, please provide a few rows of data from your dataset, following the format I mentioned earlier. For example:

```
Type, Age, Rural, VKM.q, CC.q
Car, 35, No, 25000, 3
Bicycle, 28, Yes, 1000, 0
Motorcycle, 42, No, 15000, 1
...
```

Replace the placeholder values (Car, 35, etc.) with the actual data. The more rows you provide, the better I can understand the data and generate an appropriate chart for you.

22: Right on the money: daily vehicle-kilometer traveled.

23: It does: daily gas consumption, in L.

24: See Noteable's [Get your data into ChatGPT: CSV, JSON, Databases & more](#).

25: It may also be able to do basic data preparation and manipulation, such as deleting specified rows from the dataset.

26: This is manifestly false: there were 997 observations in the dataset.



PUPC 0 0 330 49  
MVAN 5 1 230 31  
[...]



I see that you have provided a dataset with several entries. Each entry has five columns: Type, Age, Rural, VKM.q, CC.q. I assume that VKM.q represents some kind of distance or mileage,<sup>22</sup> and CC.q represents a numerical value.<sup>23</sup>

Please let me know if there's anything else you'd like to do with the dataset.

Once the dataset has been uploaded,<sup>24</sup> we may ask for a specific type of chart, providing the necessary details. At this point the chatbot will attempt to generate the chart based on the **information provided**.<sup>25</sup>

If it is not possible to obtain a visualization itself, the chatbot can provide code in a requested programming language (such as R, Python) which can generate the visualization when run in the appropriate space.



How many rows are there in the dataset? Could you plot CC.q against VKM.q, using Type as the colour?



Certainly! The dataset contains 55 rows.<sup>26</sup>

To plot the column CC.q against VKM.q with the color determined by the Type column, we can use a scatter plot. Here's an example code using Python and the Matplotlib library:

```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming your dataset is stored in a DataFrame called 'df'
plt.scatter(df['CC.q'], df['VKM.q'], c=df['Type'])
plt.xlabel('CC.q')
plt.ylabel('VKM.q')
plt.title('CC.q vs VKM.q (Color by Type)')
plt.colorbar(label='Type')
plt.show()
```

This code will generate a scatter plot with the x-axis representing CC.q, the y-axis representing VKM.q, and the points colored based on the Type column. The colorbar on the side of the plot will provide a legend for the different types.

Note that you need to have the pandas and matplotlib libraries installed in your Python environment.





What if I wanted to use ggplot2 instead of Python?



If you prefer to use the R programming language and the ggplot2 package for data visualization, you can create a similar scatter plot with color encoding using:

```
# Assuming your dataset is stored in a data frame called 'df'
library(ggplot2)

ggplot(df, aes(x = CC.q, y = VKM.q, color = Type)) +
  geom_point() +
  labs(x = "CC.q", y = "VKM.q", title = "CC.q vs VKM.q (Color by Type)") +
  scale_color_manual(values = c("red", "blue", "green"), labels = c("Type 1", "Type 2", "Type 3")) +
  theme_minimal()
```

In this code, we use the `ggplot()` function to initialize the plot and specify the dataset. The `aes()` function is used to define the aesthetic mappings, where `x = CC.q` represents the x-axis variable, `y = VKM.q` represents the y-axis variable, and `color = Type` represents the variable used for color encoding.

The `geom_point()` function adds the scatter plot layer, and the `labs()` function is used to set the axis labels and the plot title. The `scale_color_manual()` function allows you to manually specify the colors and labels for the different types.

Make sure you have the ggplot2 package installed in your R environment before running this code.

It would be nice to get ChatGPT to output the chart – after all, it has access to the data and the code – but there is no reason to expect it to also have access to the software.<sup>27</sup>

Still, obtaining working code is no small feat; it is much easier to **edit a first pass** than it is to build a display **from scratch**. There is one problem, however – running the output code locally with the dataset **does not work!**<sup>28</sup>

This does not speak well for the chatbot's usefulness: perhaps the ChatGPT emperor truly has no clothes, in the final analysis.

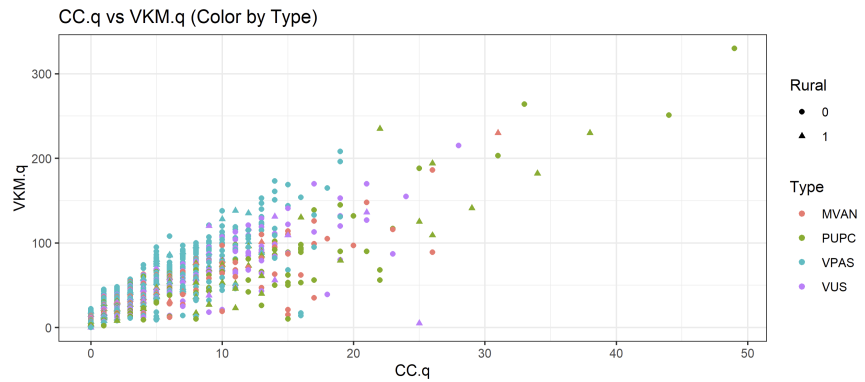
That may still prove to be the case in general, but the fault lies with us in this instance: we should have specified that `Type` was a categorical variable with **4 levels**. We could also have asked for the `Rural` field to be treated as a categorical variable and integrated into the chart as the marker's shape, and so on.

We suspect that the full measure of a chatbot's power (when it comes to data analysis and data visualization) will only ever be unlocked through clever **prompt engineering** – learning how to ask the questions so that ChatGPT and its cousins give us the needed answer. This is not unlike how learning to query search engines effectively is now seen as an **essential research skill**.

27: Or is there? This may change in the near future for the most common tools, one would presume. More on this later.

28: We receive the following error message: Error: Insufficient values in manual scale. 4 needed but only 3 provided.

We will not keep you in suspense any longer; a (modified) version of the ChatGPT suggested plot is shown in Figure 10.1 (see Chapter 12 for more information on `ggplot2`).



**Figure 10.1:** This plot is generated from a slightly modified version of the code provided by ChatGPT; the Autos dataset contains 997 observations on vehicles of 4 types (Type), with daily mileage (VKM.q), daily fuel consumption (CC.q), and where the majority of the driving is done (Rural).

It may also be possible to ask the chatbot (and by extension the engines in the backend) to perform some **data analysis** and **data modeling**, such as linear regression, say.



Could you find the linear regression model of CC.q in terms of VKM.q?



Certainly! To find the linear regression model of CC.q in terms of VKM.q, you can use the `lm()` function in R. Here's an example code snippet:

```
# Assuming your dataset is stored in a data frame called 'df'
model <- lm(CC.q ~ VKM.q, data = df)
summary(model)
```

In this code, the `lm()` function is used to fit a linear regression model. The formula `CC.q ~ VKM.q` specifies that we want to predict CC.q using VKM.q. The `data` parameter is used to specify the dataset (df in this case).

After fitting the model, you can use the `summary()` function to display a summary of the regression results. It will provide information such as the coefficients, standard errors, *t*-values, and *p*-values.

Make sure you have the necessary data loaded and the required packages (e.g., `ggplot2`) installed in your R environment before running this code.





Are you able to compute the model yourself, rather than give me the code?



I can compute the linear regression model for you. Here is the result of the linear regression model of CC.q in terms of VKM.q:

```
Call:
lm(formula = CC.q ~ VKM.q, data = df)

Coefficients:
(Intercept)      VKM.q
   -9.553032    0.001603
```

The model equation can be expressed as  $CC.q = -9.553032 + 0.001603 * VKM.q$ , where CC.q is the dependent variable and VKM.q is the independent variable.

Please note that this model assumes a linear relationship between CC.q and VKM.q and may not be suitable for all cases. Additionally, the interpretation of the coefficients should be done with caution, considering the context of your data.

Impressive, right? Apart from being **complete hogwash**, we mean. ChatGPT is a **dangerous tool** to leave in the hands of users who do not understand enough about the topic being discussed (in this case, linear regression) to recognize ridiculous answers when they encounter them: the **true** line of best fit is in fact  $CC.q = -0.1184 + 0.1221 * VKM.q$ !<sup>29</sup>

When making such requests, it is important to remember that the **quality** of answers given is not guaranteed, that results should neither be **taken at face value** nor **assumed to be correct**, at least in the current state of commonly-available technologies. Careful **validation** of the outputs is likely to prove mandatory for the foreseeable future.<sup>30</sup>

It is important to realize that **certain assumptions** may also (implicitly) be made by these analysis engines, unless otherwise specified. This time, however, ChatGPT explicitly noted it upon being asked to provide a linear regression model.<sup>31</sup>

In addition to generating visualization and data analysis code, LLMs can also try to provide **interpretations** of **well-known** data visualizations. Here's an example of a conversation where ChatGPT provides some "insight" into Minard's *March to Moscow* visualization (see Figure 1.15, Section 1.4, [3, 13]).



**Jen:** Do you know Minard's March to Moscow chart?

29: Did ChatGPT just make up the coefficients? Remember, it thought that there were only 55 observations in the dataset.

30: Although the **WolframAlpha plugin**  can provide sophisticated outcomes (in theory).

31: Can we always trust it to let us know what the implicit assumptions were?

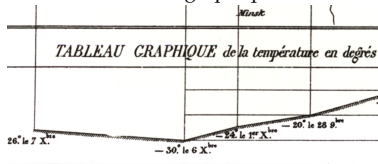
32: The chart only shows the Grande Armée's *March to Moscow* starting from and ending at Kaunas, which was part of the Russian Empire at the time but in present-day Lithuania.

33: That's not right, is it? What table does ChatGPT mean? Let's read on.

34: Nope. Lithuania, not France.

35: Same as previous sidenote.

36: There is definitely no table in the chart. Here is what may have happened. The lower section of the chart is called a "tableau graphique" in French.



This expression translates to "chart", but the first word "tableau" also translates to "table". This is not a big mistake – many English speakers might have fallen for the *faux ami* ... but would have quickly realized that there was a translation issue (we hope!).



Yes, I'm familiar with this chart. It is a famous data visualization created by Charles Joseph Minard in 1869. The chart depicts Napoleon's 1812 military campaign to Russia and showcases several variables such as the size of the army, geographical locations, temperature, and time.

The chart is remarkable because it effectively communicates multiple dimensions of data in a clear and concise manner. It shows the diminishing size of Napoleon's army as they advance towards Moscow and retreat back to France,<sup>32</sup> with the width of the line representing the troop count. It also includes additional information such as the temperature on the return journey and major events during the campaign.

The March to Moscow chart is often hailed as one of the most influential and iconic data visualizations, showcasing the power of visual storytelling and effectively conveying complex information.

I can describe its key elements and provide a textual representation. The chart consists of several interconnected elements: a map, a line representing the movement of Napoleon's army, and additional information presented in a table format.<sup>33</sup>

**Map:** The chart features a map of Europe, with Russia at the rightmost side and France at the leftmost side.<sup>34</sup> The map shows major rivers, cities, and geographical features along the campaign route.

**Line:** A thick, brown line starts in France,<sup>35</sup> representing the size of Napoleon's army at various points during the campaign. The line gradually narrows as the army advances towards Moscow, depicting the decreasing troop count due to battles, casualties, and harsh conditions.

**Table:** The chart includes a table on the bottom section, providing detailed information at specific points along the campaign route. It shows the cities or landmarks, the dates, the direction of movement, the temperature, and the remaining troop count. Notable events such as battles and river crossings are also indicated.<sup>36</sup>

Overall, the chart effectively visualizes the catastrophic losses suffered by Napoleon's army during the campaign, highlighting the devastating impact of the harsh Russian winter and strategic military engagements.

While I can't display the actual image, you can easily find it online. It is widely available and often referenced in discussions about data visualization and information design.



Do you know what the reaction to the chart was in Minard's time? I suspect that the reaction in France might have been different from that in the rest of the world.



In Charles Minard's time, the reaction to his March to Moscow chart varied depending on the audience and their perspective. While it's challenging to determine the exact public sentiment from that era, we can make some general observations based on historical context.

**In France:** It is likely that the initial reaction in France was mixed. The chart depicted the disastrous outcome of Napoleon's campaign, showcasing the immense loss of life and the failure to achieve the intended objective. This portrayal may have been seen as a blow to national pride and the reputation of Napoleon. However, it's important to note that the chart was created several decades after the events took place, so the emotional impact may have been somewhat diminished.

**Rest of the World:** Outside of France, the chart was met with considerable admiration and recognition.<sup>37</sup> Charles Minard's visualization was groundbreaking for its time, combining geographical information, statistical data, and the narrative of a historical event into a single, cohesive representation. The chart's clarity and effectiveness in conveying complex information garnered attention and praise from the international community.

Over time, Charles Minard's March to Moscow chart has become widely regarded as one of the most influential examples of data visualization.<sup>38</sup> Its enduring popularity is a testament to its ability to communicate a compelling story through data, transcending national boundaries.<sup>39</sup>

It's important to note that these observations are speculative, as there is limited information available regarding the immediate reactions to the chart in Minard's time. The interpretation and reception of the chart may have varied among individuals based on their personal views, political affiliations, and access to the visualization.<sup>40</sup>

37: Was it, though? It seems reasonable to expect that it was indeed the case, but we cannot find historical proof one way or the other. How much of this answer is due to Jen's specific prompt wording?

38: We know that this, at least, is true. But it would have been nice to see some evidence – this could be remedied through prompt engineering.

39: We appreciate the sentiment, but is that really why the chart is continuously popular?

40: An important reminder, to be sure.

41: Yes, it makes some silly (and upon reflection, easily-available) mistakes, but it has no real analytical "weapon" that it can use to tackle the questions we asked of it other than an impressive bag of **natural language processing** tricks; in that sense it is miles further afield than the chatbots from even a few years ago.

So what do you think of ChatGPT's take on this visualization classic? If the chatbot comes out of this exercise looking somewhat beleaguered, keep in mind that the freely-available bare bones version runs on GPT-3.5.<sup>41</sup>

Chatbot **upgrades** (such as **GPT-4** [↗](#) and ChatGPT Plus) and **plugins** (such as the aforementioned WolframAlpha) give users tools to augment and simplify the **data visualization and storytelling pipeline**, as we now see.



**Stephen:** Write a small blurb on the difference in temperature right now between Edmonton (Alberta) and Ottawa (Ontario), and show the difference in temperatures in a bar chart.



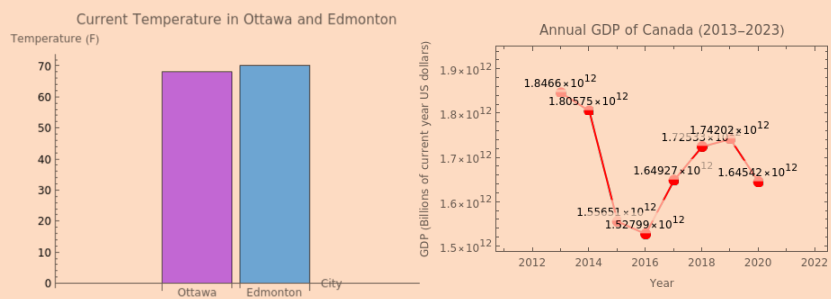
**ChatGPT + WolframAlpha:** As of the most recent data, the temperature in Ottawa, Ontario is 68F (20C), while in Edmonton, Alberta, it's slightly warmer at 70F (21C). These temperatures reflect the local climates of the two cities, with Edmonton being located more inland and having a more continental climate, while Ottawa, though also having a continental climate, is influenced by its proximity to bodies of water. Please note that these temperatures can fluctuate throughout the day and may have changed slightly since the last update.



Now, find the annual GDP of Canada from 2013 to present, and plot the data on a red line chart, adding data labels. Finally, write a 100 word overview of the Canadian economy using the data and the chart.



Canada's economy, one of the largest in the world, is a highly developed mixed economy. It is dominated by the service industry, which employs about three quarters of Canadians. However, it also has a strong natural resources sector, with logging and oil being two of the country's most important industries. The GDP of Canada has shown a general upward trend from 2013 to 2023, as illustrated in the graph below. The graph shows the annual GDP in billions of current year US dollars, with data labels indicating the GDP for each year.



Not perfect yet, for sure... but it's getting there! No doubt this chapter will look very different in the next edition. Stay tuned!