Major Assignment 1: Playfair

Stat 624: Data Visualization

Scott Thatcher

Fall 2022

1   Introduction

William Playfair is credited as being one of the founders of the use of graphical methods in statistics. Among other things, he is credited with creating line, area, bar, circle and pie charts to display economic data in his *Commercial and Political Atlas* (1786) and *Statistical Breviary* (1801).

Wikipedia gives an interesting read (wikipedia.org), describing him as "millwright, engineer, draftsman, accountant, inventor, silversmith, merchant, investment broker, economist, statistician, pamphleteer, translator, publicist, land speculator, convict, banker, ardent royalist, editor, blackmailer and journalist." I'd encourage you to take a look at that article!

We'll be looking at another of Playfair's books, *A Letter on our Agricultural Distresses, their Causes and Remedies: Accompanied with Tables and Copper-Plate Charts, Shewing and Comparing the Prices of Wheat, Bread, and Labour, from 1565 to 1821. Addressed to the Lords and Commons*, whose entire text can be found through a freely-available Google Books scan of a copy in The British Library.

The chart itself is reproduced in Figure 1 below, but I encourage you to read at least Appendix C of Playfair's original book, which gives an explanation of the chart and a defense of the use of data visualization that name-drops Adam Smith.

In creating this graph, Playfair certainly seems motivated by the desire to create what we might call a multivariate data visualization. There's a lot going on. More specifically, Playfair was motivated to tell the story of what we might call the purchasing power of a British laborer over time, arguing "That never at any previous period was the price of wheat so low in proportion to that of labour, as in the first thirty years of the last reign."

In successfully completing this assignment, you'll practice several skills:

- Use `ggplot` to create complex data visualizations, building on your knowledge of the syntax and structure of `ggplot` commands to build custom graph elements.

- Critique Playfair's visualization, making sure to

    – use the vocabulary we've learned to refer to the elements of design,

    – refer specifically to the principles of good data visualization we've discussed in class, and
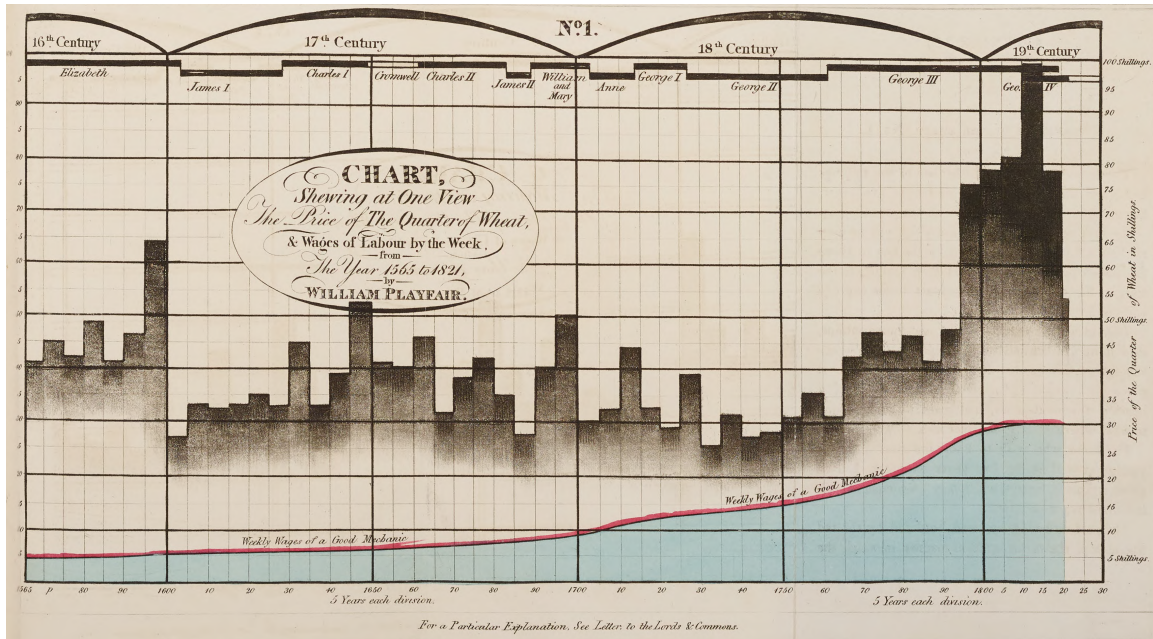
Figure 1: Playfair's "Chart, Shewing at One View the Price of the Quarter of Wheat, and Wages of Labour by the Week, the Year 1865 to 1821"

- consider Playfair's purpose in creating the visualization.
- Use `ggplot` and the principles we've learned to create a graph that improves on Playfair's data visualization.
- Write about Playfair's visualization, the conclusions that might be drawn from the visualization, and your own improvements using all the techniques of good academic writing

This assignment is definitely more open-ended that previous assignments, in terms of the approaches and the code you might use. I encourage you to talk with me if you're feeling stuck. It's also a good idea to sketch ideas on paper before trying to make a new graph in R.

2   Assignment

This homework assignment should consist of several parts:

1. A critique of Playfair's data visualization.
2. A re-creation of Playfair's graph using R and `ggplot`.
3. Creation of a new data visualization that you feel does a better job.
4. A concluding written section that explains the choices you made, and how they improve on Playfair's work.

More details of each follow in the sections below.

2.1 Part 1: Critique

This first part is a critique of Playfair's data visualization.

- In a paragraph or two, summarize aspects of this graphic that work well to display quantitative information. You may want to refer back to the general principles and vocabulary we've discussed in the past few lectures.

- What aspects of this graph are perhaps not ideal? In other words, what visual elements don't display quantitative information as well as they could? Answer in another paragraph or two. I'd encourage you to think back to the talk on principles of good data visualization for ideas and vocabulary.

- All things considered, does the graph communicate what it sets out to communicate?

2.2 Part 2: `ggplot` Re-creation

Using the data sets `playfair_wheat.csv` and `playfair_monarchs.csv`, write `ggplot` code to come as close as you can to replicating Playfair's graph. You'll likely have to do some calculation of new "helper" variables in the data frame(s) if you really want to get all the details similar to the original. One good piece of advice is to look up the various `ggplot` commands, using the `?` operator. Take note of required and optional aesthetics. Each aesthetic needs to be a column in the data frame, so you might want to think about what parameters are needed, then add them to the data frame. Preliminary work to add columns to the data frame(s) can make your `ggplot` commands much simpler.

A good submission can be made with only built-in `ggplot` geoms and commands, but if you're going for all the details, you may want to look into the `ggpattern` and `showtext` packages.

You don't have to get this recreation perfect to get most of the points. However, the goal of the assignment is to stretch your `ggplot` muscles. How close you can get (while thinking of Playfair and his engraver doing this work in a very different way)? A good subssion will get these details right and get most of the points:

- the bar graph, without worrying about shading,
- the wage curve and shading below,
- the monarchs information at the top, complete with labels,
- at least one label on the red wage line,
- axes with the same "breaks" as the original graph, without worrying about typographic details like leaving off the first two digits of some numbers,
- axes labels that contain the same information, without worrying about exact replication of text placement, and
- a title with the same words that Playfair used, but perhaps not in the same position. (Make sure it doesn't get cut off, however.)

Advice on Coding: Make sure to work in stages. Get a simple working graph first, then add pieces one at a time. This will really help the debugging. I've seen students try to include everything on the first pass, and that leads to code that is much more difficult to debug.

2.3 Part 3: New Data Visualization

I'd like you to think about how you might visualize Playfair's data to achieve the following goals:

- Make Playfair's point about purchasing power (unless he's wrong–then make the correct point!).

- Produce an accurate and honest visualization of the data.

- Think creatively (as Playfair did) to incorporate all the variables in the graph.

- Correct any problems you identified with Playfair's graph.

Then write `ggplot` code to create that new data visualization.

2.4 Part 4: Concluding Explanation

Write a few paragraphs that explain the choices you made in creating your data visualization.

- Why did you choose the particular aesthetic elements that variables were mapped to? Why were they appropriate?

- Did you make any other decisions that lead to clear communication? You might review the elements from our lectures on good design principles–for example, did you make decisions about colors, spatial organization, etc.?

- Are there any ways in which the graph could be improved, or any ways in which the graph doesn't communicate as clearly as it could? Briefly explain.

- Finally, does it appear that the data, as presented, supports Playfair's point?

3 Writing

Make sure to pay attention to thesis and topic statements, as well as overall organization. The bullet points above give a fairly good idea of one way to organize the assignment. Of course, make sure to also proofread for grammar and spelling, etc.

4 Submission

Submit both your RMarkdown file and a PDF of the final knit document.

I'd like the PDF to *not* include code this time. In other words, it should look more like a final report. This can be achieved by using options like `echo=FALSE`, `message=FALSE`, and maybe `warning=FALSE` at the beginning of code blocks (but remember to pay attention to warnings before you silence them).

Finally, also remember that you can manipulate proportions of figures (and elements within figures) by using the `fig.width` and `fig.height` options at the starts of code chunks that contain graphs.