The Erdős Institute
Data Visualization Mini Course
Problem Set 2

Technical Practice

1. Creating an HTML file

Create an HTML file that contains a single `div` element that contains a `p` element with your name as well as a `p` element with your birthday. Give the `p`s `id`s that correspond to the information they contain.

2. Creating an HTML layout

Look at the layout provided in `tech_practice2_layout.pdf` and recreate it using HTML and CSS code. You may want to set a border color so that you can see if your layout follows the sketch in the PDF file.

3. Practice making a d3.js scatter plot

Using python, subset the `penguins.csv` data file in the `data` folder of the repository to remove any observations with missing data. Then, using d3.js, build a scatter plot of the penguin bill length against the penguin bill depth. After building the simple scatter plot try making the following adjustments:

- Color the points by the `species` column
- Size the points according to the penguin's body mass
- (More advanced) Add a way for the audience to toggle the size of the points according to the body mass

Be sure to include axes and axes labels.

4. Plotting a time series

In this problem you will plot a time series with d3.js. Time series using a date or datetime as their horizontal axis need a special parsing function. Remember that d3.js loads in all data as a string, this includes date data. When we had a number we would use JavaScript's `parseFloat` or `parseInt` functions. With a date or datetime we need to define a parsing function using `d3.timeParse`, https://github.com/d3/d3-time-format/blob/v4.1.0/README.md#timeParse. When using `d3.timeParse` you input a string representing the format of your date data.

Plot the closing price of the S and P 500 stock index for the year 2022 using `index_closes_2022.csv` in the `data` folder. The trading day is in the `date` column, and the closing price (as a fraction of the closing price on January 3, 2022) is in the `snp_close` column.

Be sure to include axes and axes labels. Note that you will want to use `.tickFormat()` combined with `d3.timeFormat()`, https://github.com/d3/d3-time-format, when making your horizontal axis.

5. Presidential text updates

The code in `question5_code.html` produces an unfinished Gantt chart plotting rectangles that represent the terms of all United States presidents. The chart itself is complete, but it has been left to you to complete the interactive portion of the code. When a user hovers over a rectangle we want the presidency number, president's name, president's political party, and term (the date they took office to the date they left office) to appear in the information block on the right of the chart. You will need to write `‘‘mouseover’’` and `‘‘mouseout’’` functions for the rectangles.

The data we are using is stored in `presidents_terms.csv` in the `data` folder. You will also need to use an `if else` statement to deal with the term for the incumbent president.

6. Simple animation

Write d3.js code to create an animation where a circle traces the outline of the unit square clockwise. That is, an animation where a circle moves from $(0, 0)$ to $(1, 0)$ to $(1, 1)$ to $(0, 1)$ and back to $(0, 0)$.

If you want a challenge, try to figure out how to make this animation run on an infinite loop using an update statement.

7. An NBA payroll plotter (Advanced Problem)

Using the layout you created in question 2. and the shell code in `nba_payroll_plotter.html`, create a dashboard similar to that found here https://matthew-osborne.com/mtodata/Posts/team-payroll-plotter.html using the data in `nba_payrolls.csv` in the `data` folder.

Hint: Writing a function to use for the "change" update on the `selects` is easiest. You may also want to use `document.getElementById()` to get the selected value from the `selects`.

8. Playing around with bootstrap made

Find a template you like at https://bootstrapmade.com/. Download the source code and play around with it. Make edits to personalize the webpage to your like. See if you can figure out how to include a d3.js visualization.

Annotating interactive visualizations

Just like with static (non-moving and non-interactive) charts, text annotations are of critical importance. Annotations can inform your audience how they can interact with your visualizations. Check out some examples where the authors improved their interactive visualizations by adding annotations:

- https://projects.fivethirtyeight.com/tom-brady-touchdown-passes/
- https://matthew-osborne.com/mtodata/Posts/larry-obrien-cost.html
- https://pudding.cool/2022/02/plain/

Anscombe's Quartet

While tables are a valid (and at times more effective) form of data visualization, there are times when they can conceal the true nature of the data. Plot the corresponding (x, y) pairs found in `anscombe_data.csv` in the `data` folder as scatter plots (meaning plot `y1` against `x1`, `y2` against `x2`, and so on). Then produce tables showing the mean, variance, and correlation between the x and y variables. Compare and contrast your take-aways when viewing the data as a scatter plot versus when viewing the data as summary tables.

Physical visualizations

In this mini course we have spent, and will continue to spend a lot of time on learning how we can use technological tools to quickly produce data visualizations. However, it is important to remember that data visualizations are not only produced via software. Many interesting and wonderful visualizations can be made in the physical world. Here are a few examples:

- Jaime Serra Palou's 'Café diario', https://www.domestika.org/en/projects/417262-cafe-diario. In this work Mr. Palou held onto the paper underneath his coffee mug for an entire year to visualize his daily coffee consumption
- Todd Whitehead became famous in the NBA data viz world for his physical visualizations on Twitter:
  - https://twitter.com/CrumpledJumper/status/1397672090807934978/photo/1
  - https://twitter.com/CrumpledJumper/status/1400973484604366850

You can read an interview with him here, https://nightingaledvs.com/meet-the-data-visualization-king-of-basketball-twitter-todd-whitehead/

- A somewhat popular meme that shows how the wear on a stack of weights at a gym demonstrates the distribution of the pin placement by gym goers, https://www.reddit.com/r/funny/comments/m4aaee/how_to_explain_normal_distribution_to_a_bro_at/

You could consider producing a physical data visualization for your final project!

Gestalt principles

Read the following description of the Gestalt prinicples.

The six Gestalt principles can help us understand what our audience perceives when they look at one of our data visualizations. It is important to keep them in mind when we design.

Proximity

Things that are close together tend to be perceived as belonging to a group.

Similarity

Objects that have the same color, shape, or direction are grouped by our brains.

Enclosure

Objects that are enclosed are seen as belonging to the same group.

Closure

We tend to ignore gaps and complete empty spaces of structures. Consider Figure 1 as an example.

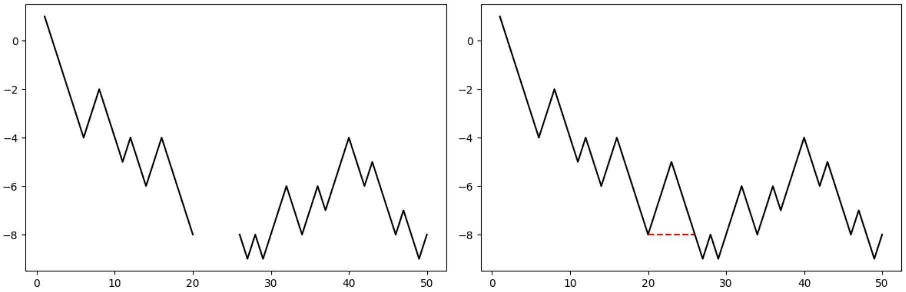

Figure 1: An illustration of the Gestalt principle of closure. If presented the image on the left our brains would tend to close the gap by drawing a straight line connecting the points (the red dotted line on the right). This could be problematic if the gap was the result of missing data that was drastically different from what our brains would complete, like the example on the right.

Continuity

When objects are aligned together or continue one another, we perceive them as a group.

Connection

Objects that are connected are seen as part of the same group.

Summary

Keeping these principles in mind when making design choices in our visualizations is important. We want to make choices that take advantage of the principles, for example, coloring or selecting markers according to a categorical variable. We want to avoid making choices in which the principles would work against our plot, for example what was demonstrated in Figure 1.

An alternative to the word cloud

A visualization that I personally dislike is the word cloud. Word clouds visualize word usage in a piece of text by sizing the words in proportion to the number of times (or frequency with which) they were used. The "cloud" portion of this comes from their somewhat random positioning in a giant mass of words. An example of a word cloud for A.A. Milne's Winnie-the-Pooh can be seen in Figure 2

A natural alternative to the word cloud is a bar chart where each bar corresponds to a word and the length displays the frequency or number of times the word is used. However, we can edit the appearance of the word cloud and make it more legible, while still keeping the spirit of using the words as the visualization. Instead of being randomly placed, the words could be ordered by their frequency. An example of such a visualization can be found in this New York Times post, `https://www.nytimes.com/2022/03/24/learning/whats-going-on-in-this-graph-march-30-2022.html`. With d3.js we can easily make such a visualization for the Winnie-the-Pooh data. Use the `winnie_the_pooh_word_counts.csv` file in the `data` folder to make such a visualization.

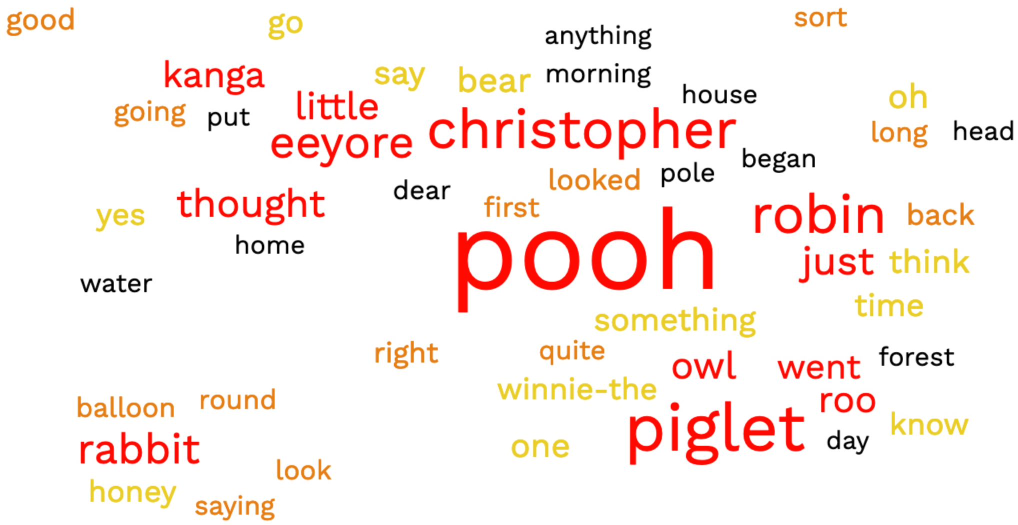

Figure 2: A word cloud of A.A. Milne's Winnie-the-Pooh as transcribed by Project Gutenberg. Created using `https://www.freewordcloudgenerator.com/`