Project 0: Visualizing Anscombe's Quartet

Due 11:59pm, Friday, February 17 through GitHub
https://classroom.github.com/assignment-invitations/e212956ea5f76c1dff4c036b0e2679ee

In this project, you will work with D3 to visualize the four datasets of Anscombe's quartet. Parts 1 through 5 will be required for the assignment, whereas the bells and whistles are optional extra credit additions.

Collaboration Policy:
You may optionally work in groups on this project. If you choose to work with others, all group members must submit their own work and you should note who you worked with the complete the project at the top of your readme.txt file.

You are also welcome to use resources found online; however, the code you submit should be your own (please do not directly copy-paste code from a tutorial or online example). If you use code from any online sources, please also list those sources and what pieces of the code you used those sources to complete. Also, for each of these sources, please add your own comments as to what the code does so I can grade your project based on your own understanding of the code.

Submission:
All assignments must be submitted through GitHub. I will add you to the class GitHub using the id you provided in the course survey (if you did not provide one or do not receive an invitation to the GitHub organization by Monday, February 13, please let me know).

To submit your assignment, please create a new directory in the "Submissions" directory, named as your last name (e.g., "`szafir`"). All submissions should include a readme.txt with the answers to the free response questions and a list of any collaborators you worked with on this project.

Rubric:
Each of Parts 1 - 5 is worth 10 points. Completing the readme.txt is worth 10 points, for a total of 60 points. **If you are unable to get any pieces of the project to work, please comment out the broken pieces of code and add notes in your readme.txt as to what you were trying to do with that code and where you ran into problems.** Partial credit will be given on the basis of these comments. Partial credit will not be given for extra credit options.

Per the course policies, projects may be submitted up to 72 hours late, with a 10% point deduction per day.

Project Specifications

Part 1: Interpreting Data

Pull down the project code from GitHub. In the `submissions` directory, create a directory using your last name (e.g., `szafir`) and copy the `data` directory, `index.html`, and `project0.css` into it. Then, open up the `index.html` page in your text editor.

One of the challenges with using any dataset is figuring out how to get it from its original format into your visualization. You've been provided with four CSV (comma-separated values) files in the `data` directory of the project GitHub. Under Part One in `index.html`, use D3's `d3.csv` functionality to upload all four datasets. To do this, you should be sure to import D3 inside of your webpage's `<head>` (your choice of version) and make sure you're using a local server to work with your code (either with XAMPP or Python). Each time you use `d3.csv`, you should include a function that calls the `checkDataset` function on your data (this is included in the skeleton code).

To test if this is working, make sure your server is running and open your browser. Then go to the url `localhost/Project0/submissions/<your directory name>/index.html`. Under "Part 1", you should see a note that shows all four datasets have been uploaded correctly.

Part 2: Building Bars

Under Part Two in `index.html`, create a bar chart using the x-values (labelled `x`) for **one** of the four datasets (your choice of which one). The height of the bar should correspond to the x-value of the data, the order of the bars should correspond to either their natural order in the dataset or to the relative x-value (e.g., the position the datapoint would be in if we sorted all of the datapoints according to their x-value).

Your barchart should have both an x- and y-axis, and the data should be scaled according to the maximum x-value in the dataset. Please use the `d3.max` function to create this scale rather than hard-coding a specific value. Look at the in-class tutorial for an example of how to do this. Note that you will need to load the dataset you're using prior to building the visualization.

Part 3: Building Scatterplots

Under Part Three in `index.html`, visualize the same dataset you used in Part 2 using a scatterplot. Map `x` to the x-axis and `y` to the y-axis. As with the barchart, you should include axes and scale your visualization to fit the maximum values in the data.

Part 4: Interaction
Update your scatterplot code to be interactive. When you click on a point, change the paragraph with the id `scatterLabel` to print out the x and y values of that point. When you hover over a point, change it's color (and change it back when you mouse out!).

Part 5: Building Multiple Charts
Under Part Five, create a set of four side-by-side scatterplots, one for each data value. Add a label to each scatterplot that identifies which of the four datasets is being visualized ("Anscombe I", "Anscombe II", "Anscombe III", "Anscombe IV").

Bells & Whistles:

Each of the following additions may optionally be completed for extra credit. **Bells** are worth 2 points. **Whistles** are worth 5 points. Please note which bells and whistles you've completed in your readme.txt.

Bell: Tooltips
Create a tooltip that shows the x- and y-value of each point when you hover over a point in the Part 3 scatterplot.

Bell: Xs and Ys
For your bar chart code, create a pair of bar charts for the active dataset: one coding the x-value and one coding the y-value of the current dataset. You can only receive credit for one of this and the "Coordinated Views" whistle.

Bell: Styling your Visualization
The visualization we've created here doesn't have much pizzaz… Update the CSS stylesheet (`project0.css`) to restyle the visualization according to a central theme (e.g., 80's neons, old world parchment, borrow ideas from an Instagram filter, etc.). Add a note in your readme.txt explaining the inspiration for your styling choices.

Bell: Best Fit Lines
For each scatterplot, compute and visualize a line of fit on top of each scatterplot in part 5. You should compute this line in your code rather than manually specifying the line of fit.

Whistle: Transitions
Add a menu to your single scatterplot code that allow you to choose which of the four datasets is mapped to the scatterplot. When you change the dataset through this menu, the program should provide an animated transition between the two views.

Whistle: Replication

Replicate each of your graphs using Tableau or some other WYSIWYG visualization tool (e.g., Excel, Google Fusion Tables, etc). Make sure you replicate that visualization as precisely as possible (e.g., point color, size, x- and y-axis scales, etc.). Include a .png of your replication in your submission folder. In your readme.txt, outline how you created each visualization in the WYSIWYG system and what differences you experienced in assembling that visualization.

Whistle: Coordinated Views

Complete the “Xs and Ys” bell. Then, for each pair of charts, when you mouse over a bar in one graph, it should highlight that bar and the bar in the corresponding graph. For example, if you mouse over a bar representing an x-value, the corresponding y-value bar should also be highlighted. You can only receive credit for one of this and the “Xs and Ys” bell.