

Assignment 2 - Data Visualization, 10 Ways

Now that you have successfully made a “visualization” of shapes and lines using d3, your next assignment is to successfully make a *actual visualization*... 10 times.

The goal of this project is to gain experience with as many data visualization libraries, languages, and tools as possible.

I have provided a small dataset about cars, `cars-sample.csv`. Each row contains a car and several variables about it, including miles-per-gallon, manufacturer, and more.

Your goal is to use 10 different tools to make the following chart:

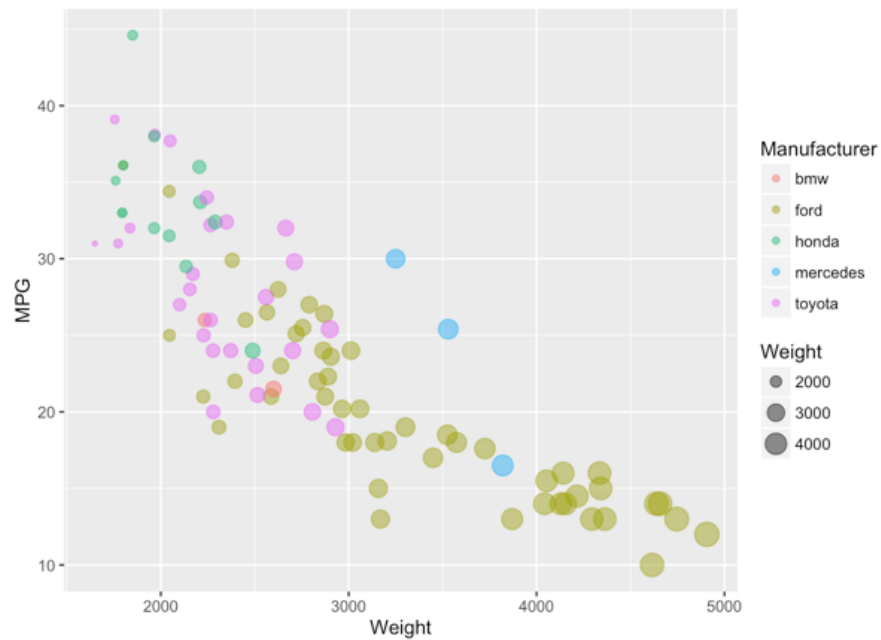


Figure 1:

These features should be preserved in your replication:

- Data positioning: it should be a downward-trending scatterplot as shown. Weight should be on the x-axis and MPG on the y-axis.
- Scales: Note the scales do not start at 0.
- Axis ticks and labels: both axes are labeled and there are tick marks at 10, 20, 30, etcetera.
- Color mapping to Manufacturer.
- Size mapping to Weight.

- Opacity of circles set to 0.5 or 50%.

Nothing else is required. This includes:

- The background grid.
- The legends.

Libraries, Tools, Languages

You are required to use 10 different tools. Of the 10 tools, you must use at least 3 languages. This could be `Python`, `R`, `Javascript`, or `Java`, `Javascript`, `Matlab` or any other combination. Dedicated tools (i.e. Excel) do not count towards the language requirement.

Otherwise, you should seek tools and libraries to fill out your 10.

Below are a few ideas. Do not limit yourself to this list! Some are poor choices (like SPSS).

- R + ggplot2
- Excel
- SPSS
- `d3` <- since the rest of the class uses this, it's a good idea to use it
- HighCharts
- nvd3
- Matplotlib
- Matlab
- Processing
- Tableau
- Java 2d
- GNUplot
- Vega/Vega-lite

You may write everything from scratch, or start with demo programs from books or the web. If you do start with code that you found, please identify the source of the code in your README and, most importantly, make non-trivial changes to the code to make it your own so you really learn what you're doing.

Tips

- If you're using `d3`, key to this assignment is knowing how to load data. You will likely use the `d3.json` or `d3.csv` functions to load the data you found. Beware that these functions are *asynchronous*, meaning it's possible to "build" an empty visualization before the data actually loads.

- *For web languages like d3* Don't forget to run a local webserver when you're debugging. See this ebook if you're stuck.

Readme Requirements

A good readme with screenshots and structured documentation is required for this project. It should be possible to scroll through your readme to get an overview of all the tools and visualizations you produced.

- Each visualization should start with a top-level heading (e.g. `# d3`)
- Each visualization should include a screenshot. Put these in an `img` folder and link through the readme (markdown command: `![caption](img/<imgname>)`).
- Write a paragraph for each visualization tool you use. What was easy? Difficult? Where could you see the tool being useful in the future? Did you have to use any hacks or data manipulation to get the right chart?

Other Requirements

0. Your code should be forked from the GitHub repo and linked using GitHub pages.
1. Place all code, Excel sheets, etcetera in a named folder. For example, `R-ggplot`, `matlab`, `mathematica`, `Excel` and so on.
2. Your writeup (readme.md in the repo) should also contain the following:
 - Description of the technical achievements you attempted with this visualization.
 - Some ideas include interaction, such as mousing over to see more detail about the point selected.
 - Description of the design achievements you attempted with this visualization.
 - Some ideas include consistent color choice, font choice, element size (e.g. the size of the circles).

GitHub Details

- Fork the GitHub Repository for Assignment 2. You now have a copy associated with your username.
- Make changes to fulfill the project requirements.
- To submit, make a Pull Request on the original repository.

Grading

Grades on a 120 point scale. 100 points will be graded for functionality: 10 points per visualization. 20 points will be based on your Technical and Design achievements.

Make sure you include the files necessary to reproduce your plots. We will choose some at random to run and test.

Note: in the past, students generally lose points for unjustified or suboptimal choices in chart design elements such as axes, size ranges, colors, and so on.