D3 Assignment

Information Visualization 2016/17

Assume you were contacted by the Royal Dutch Meteorological Institute (KNMI). They have a dataset of daily temperature measurements on the Schiphol Airport in the last five years (2011 - 2015) and they want to visualize the average monthly temperatures of each year in a bar chart. They have provided the dataset (meteo.csv), your task is to provide this visualization.

A complete solution contains the following elements (use the example solution in figure 1 as a reference for some of them):

1. Written completely in D3
2. Loads and uses the weather data from the provided CSV file
3. Each year represented by 12 bars for each month with each bar's size corresponding to the average temperature in said month
4. Month labels below the bars
5. Labels displaying the temperature at the top of each bar
6. The size of each bar is computed dynamically using D3 scale
7. Dynamically computed y-axis on the left-hand side
8. A label showing the current year
9. The user can navigate through years by pressing the left arrow (previous year) or right arrow (next year); pressing an arrow toggles update of the year label and recomputing of the bars and axes

The assignment is handed in through Blackboard, the deadline is 23 Feb 9:00, i.e. the assignment should be handed in before the third practical session. Please hand in a ZIP file containing your complete solution.

IMPORTANT: Don't be afraid to ask questions, also through e-mail (g.strezoski@uva.nl, d.arya@uva.nl). Also, if your assignment does not include all the above-mentioned elements by the deadline, hand in what you have anyway. The purpose of the assignment is to develop your D3 skills, for which you receive feedback at latest in the initial stage of the project. Do not defer the hand-in until the assignment is good enough.

The assignment (and also the following text) is divided in six stages. Those are for your convenience, so that you can progress step-by-step.

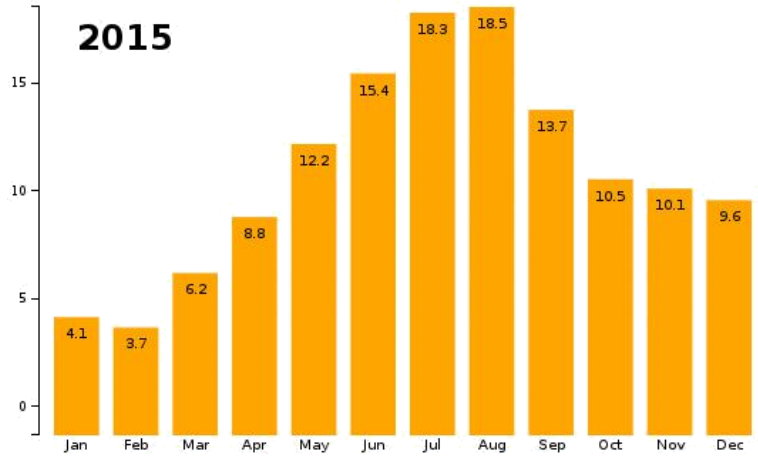

Figure 1: Example solution of the assignment.

1 Setting up

Output: A basic HTML page with a D3-generated text.

This phase is pretty straightforward: put all the building blocks in place, so that you can start working with D3. Use [DashD3] chapters 4 (Basic Building Blocks), 5 (D3.js First Steps) and 6 (Adding a DOM element) to guide you along the way. Alternatively, you can use [MurrayD3] chapters 2 (Fundamentals), 3 (Setup) and 4 (Adding elements).

2 Show the data

Output: The basic HTML page now prints out a toy dataset in text.

In this phase, you do your first data binding. This time, to keep it simple, we just output data to screen as text. Since the final data will have 12 bars for each year, it might be useful to define the data as an array of 12 elements, e.g. like this:

```
var data = [5,10,15,20,25,30,30,25,20,15,10,5];
```

Again, use one (or both) the tutorials to guide you: chapters 7 (Adding an SVG Element, the chain syntax part), 8 (Binding data to DOM Elements) and 9 (Using Data Bound to DOM Elements); or chapters 5 (Chaining methods), 6 (Binding data) and 7 (Using your data) of [MurrayD3].

3 Bar chart skeleton

Output: The HTML page now shows a bar (i.e. SVG rectangle) for each data point, whose height is determined by the data point's value

Now, we move to utilizing graphics for our visualization. Since we're making a bar chart, the more straightforward description is in [MurrayD3], chapters 10 (An SVG primer), 11 (Drawing SVG) and 13 (Making a bar chart). [DashD3], however, provides a more verbose and expansive description on general SVG, which will be also useful for you: in this phase, those are especially chapters 10 (Creating SVG Elements Based on Data), 11 (Using the SVG Coordinate Space) and 14 (SVG Basic Shapes and D3.js).

Use the data set from previous phase and construct just the plain bars: do not worry about the aesthetics at this point. The output of this phase should be a very plain bar chart (just the bars) and your comprehension of how to bind the data to and SVG element and how to place this data-bound SVG element on the SVG canvas.

4 Scales and axes

Output: A nice bar chart, resembling the one in figure 1, meeting requirements 3-8

Now that we understand how the SVG works, let's make the bar chart more presentable now. Utilize all the remaining chapters in the two tutorials [DashD3] [MurrayD3] to add dynamic scaling, axes and text labels. Introduce padding between the bars and make it nice. This phase takes a lot of trial and error, but completing it is worth it: you will have made the raw data into an indeed presentable visualization!

5 Load and transform data

Output: The visualization retains the nice style from the previous phase, but now displays real data from meteo.csv

This is a programming phase where you will utilize your JavaScript knowledge. You received the meteo.csv file, which is not directly usable in D3. Before using it, you need to load it into your script. The method for loading your CSV file (as well as other external files) is explained in [MurrayBook], chapter 5 (Data), section Binding Data. In a nutshell, what you need to do is this:

```
d3.csv("meteo.csv", function(data)
  { //ALL code using the data here
});
```

It is really important to place ALL the computations involving the data (that means the visualizations too) into the function passed to the d3.csv function. You can always construct custom functions outside of d3.csv, but make sure that you pass all the necessary variables to them.

Loading external data sources into your webpage, may be frowned upon by some of the browsers. We strongly recommend to use Firefox, there, your D3 code with external data should work fine. If you want to use Chrome, you may need to start it from console using this command:

"chrome.exe --disable-web-security--allow-file-access-from-files"
(Windows)

"open -a Google Chrome --args --disable-web-security--allow-file-access-from-files"
(Mac)

Should work both in Chromium and Chrome. Chrome might complain that your security will suffer. For development purposes, you can ignore that. If there are still problems with ring up your D3 in Chrome, don't spend too much time on that - the safe bet is to use Firefox.

A look into the meteo.csv file shows that there are four columns: year, month, day and temperature. Each row corresponds to a temperature on a particular day. Hence, you need to compute the averages yourself and place them in an array, since that is the data format D3 expects. Moreover, the temperatures are listed in tenths of degrees Celsius, so you need to convert the temperatures to whole degrees Celsius too. Once you are done with this, you just replace the toy data dataset used in previous phases with the real data from 2014. Your visualization should show the same values as the example solution in figure 1.

6 Interaction and transitions

Output: Complete assignment, the user can now navigate through years using the arrow keys

This is a pure trial-and-error and experimental phase. The most useful resource for interactions and transitions are the working interactive examples [D3Ex]. An especially relevant one is the Population Pyramid, which achieves the same thing we are trying to achieve interaction-wise. If you have completed the previous phases, you should be able to read and adapt the code. Make sure that both your bars and axes adapt to the data from each year. For value recomputations between years, try to use exit() to get rid of the old data.

Optional: Try to make the transition between years animated by adapting the code from examples.

Resources

Dashing D3 [DashD3]: *https://www.dashingd3js.com*

Scott Murray's D3 tutorial [MurrayD3]: *http://alignedleft.com/tutorials/ d3/*

Scott Murray's Interactive Data Visualization for the Web [MurrayBook]: *http://chimera.labs.oreilly.com/books/1230000000345/index.html*

D3 examples with code [D3Ex]: *https://github.com/mbostock/d3/wiki/ Gallery*

D3 official documentation wiki [D3Doc]: *https://github.com/mbostock/d3/wiki*

Mike Bostock's blog (author of D3): *http://bost.ocks.org/mike/*