Assignment 2

Goals

Start working with Data-Driven Documents (D3).

Instructions

You may complete the assignment in a single HTML file or use multiple files (e.g. one for CSS, one for HTML, and one for JavaScript). You must use D3 for this assignment. All visualization should be done using D3 calls. You may use other libraries (e.g. lodash) as long as they are not used to construct or modify SVGs, but you must credit them in the HTML file you turn in. Extensive [documentation for D3](https://github.com/mbostock/d3/wiki) is available, and Vadim Ogievetsky's [example-based introduction](http://vadim.ogievetsky.com/IntroD3) that we went through in class is also a useful reference. In addition, Scott Murray has written a great book named [Interactive Data Visualization for the Web](http://chimera.labs.oreilly.com/books/1230000000345). A screenshot of [an example solution](http://www.cis.umassd.edu/%7Edkoop/dsc530-2016sp/a2/solution.png) is available; please note that your solution may be different but must adhere to the requirements below.

Due Date

The assignment is due at 11:59pm on Friday, March 4.

Submission

You should submit any Files required for this assignment on Canvas In a .zip file. You may complete the assignment in a single HTML file or use multiple files (e.g. one for HTML, one for CSS, and one for JavaScript). Note that the files should be linked to the main HTML document accordingly. The filename of the main HTML document should be a2.html .

Details

This assignment will use data relating to campaign finance in the 2016 US presidential election. As of February 2016 there were eight candidates from the two major parties (Republican and Democratic) remaining. There is a file of summary data of the donations received both by the campaigns and several political action committees (PACs) associated with the candidates. The reported totals are from the [Federal Election Commission](http://www.fec.gov/) (FEC) as reported in mid- and end-year reports by the candidates or committees for 2015 and 2016. Information about the unofficial associations between candidates and PACs is taken from the [OpenSecrets](https://www.opensecrets.org/pres16/) site maintained by the Center for Responsive Politics.

The data is available in a single JavaScript file, campaign\_finance.js on Canvas. It contains two arrays, cands and pacs. cands contains candidate names, party affiliations, and both mid-year (q2) and end-year (q4) 2015 totals. pacs contains the PAC name, associated candidate name, and both mid-year (q2) and end-year (q4) 2015 totals. It also contains the createInitialScatterplot function that is used in Part 3. Both files also contain an additional field (final) for totals at the end of 2016.

There is a template a2.html on Canvas for the assignment. This is needed because Part 3 requires you to have a particular setup. Please use a2.html as your starting point.

0. Info

Like Assignment 1, start by creating an HTML web page with the title "Assignment 2". It should contain the following text:

* Your name
* The course title ("Data Visualization (COM3571)"), and
* The assignment title ("Assignment 2")

If you used any additional JavaScript libraries, please append a note indicating their usage to the text above (*e.g.* "I used the jQuery (http://jquery.com/library to write callback functions.") Include links to the libraries used. You do not need to adhere to any particular style for this text, but I would suggest using headings to separate the sections of the assignment.

1. Bar Chart: 30 points

You should write code to use the given candidate data ( cands ) to produce a bar chart that shows the receipts for the end of the year 2015 ( q4 ) for all eight candidates. The visualization must:

* Faithfully represent the amount of money each candidate raised
* Have bars oriented vertically.
* Red color bars for Republican candidates and blue bars for Democratic candidates
* Have a y-axis with ticks and numeric labels
* Have labels for each bar that indicate which candidate the bar is associated with

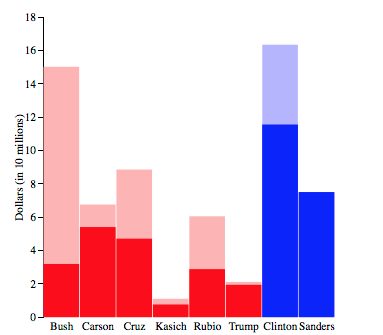
You may choose to display the amounts in millions of dollars to reduce the size of the labels.

Hints:

* Make sure you leave enough space for the labels. The [D3 margin convention](http://bl.ocks.org/mbostock/3019563) simplifies much of the math involved in adding margins to a visualization.
* Check the possible settings for [text-anchor](https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute%20/text-anchor).
* Use d3.scale and d3.svg.axis to create an axis and map values to bar heights.
* Use CSS to style the axes to make them look nicer. Use the inspector to see what elements are involved in the axes D3 generates.
* Remember you can use .orient() to change how the tick marks and labels are drawn for an axis.
* JavaScript's [String.split](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split) method may help you obtain a candidate’s last name.
* Use a translation followed by a rotation to obtain a rotated label.

2. Stacked Bar Chart: 30 points

Create a new visualization that adds the total PAC receipts associated with each candidate. The new visualization should make clear which donations are to the campaign and which are to the PACs, and the total height of the bar should be the sum of all donations. Again, show this for the end of the year totals (q4). Your visualization should adhere to the same requirements as Part 1 plus you must use stacked bars. Use a different color (e.g. a lighter shade) for PAC money.



Example Solution for Part 2

**Hints:**

There are multiple approaches for building a stacked bar chart. You will need to associate a candidate with his PAC money. Right now, the pacs array has an entry for each PAC not each candidate. An associative JavaScript array may be very useful here.

3. Scatterplot: 30 points

Start from the provided template. You must initialize the SVG you use in this part of the assignment by the provided createInitialScatterplot method. That method puts four red circles and four blue circles on the SVG canvas. You should use these circles, potentially adding, updating, and removing them, to create a scatterplot comparing mid- and end-year totals for the candidates (do not include PAC money). Add two axes that indicate amounts. Again, red circles indicate Republican candidates and blue circles indicate Democratic candidates.

The createInitialScatterplot method is repeated here for your reference; note that it is already defined in campaign\_finance.js, which you should import. The code is provided so you can identify defined classes and the provided margins, width, and height.

function createInitialScatterplot() {

var margin = {top: 20, right: 20, bottom: 40, left: 40};

var width = 400 - margin.left - margin.right;

var height = 400 - margin.top - margin.bottom;

// create an SVG using the D3 Margins Convention

// http://bl.ocks.org/mbostock/3019563

var svg = d3.select("#scatter").append("svg")

.attr("width", width + margin.left + margin.right)

.attr("height", height + margin.top + margin.bottom)

.append("g")

.attr("transform", "translate(" + margin.left + "," + margin.top + ")" );

svg.selectAll("circle.republican")

.data(d3.range(0,4))

.enter().append("circle")

.attr("cx", function(d,i) { return 4 \* i \* i + 60\*i + 30; })

.attr("cy", function(d,i) { return (15-i) \* 15 + 15; })

.attr("r", 5)

.attr("class", "republican")

.attr("fill", "red")

svg.selectAll("circle.democratic")

.data(d3.range(0,4))

.enter().append("circle")

.attr("cx", function(d,i) { return (16-3\*i) \* (16-3\*i) + 20\*i; })

.attr("cy", function(d,i) { return i \* 30 + 2 \* (70-i); })

.attr("r", 5)

.attr("class", "democratic")

.attr("fill", "blue")

}

**Hints:**

This requires understanding how D3 divides the enter, update, and exit selections. Think about which elements fall into each group. The red and blue circles have class attributes set to make it easy to select each group. To access the canvas, use the #scatter svg g selector (note the g at the end). This is because createInitialScatterplot uses the [D3 margin convention](http://bl.ocks.org/mbostock/3019563).

4. Transitions: 10 points (you can get an OK grade without this)

For some extra pizazz, add transitions to the scatterplot above. Added circles should fade in, removed circles should fade out, and moved circles should be animated from their start to end positions. There is a video in Canvas that shows how such transitions might look.