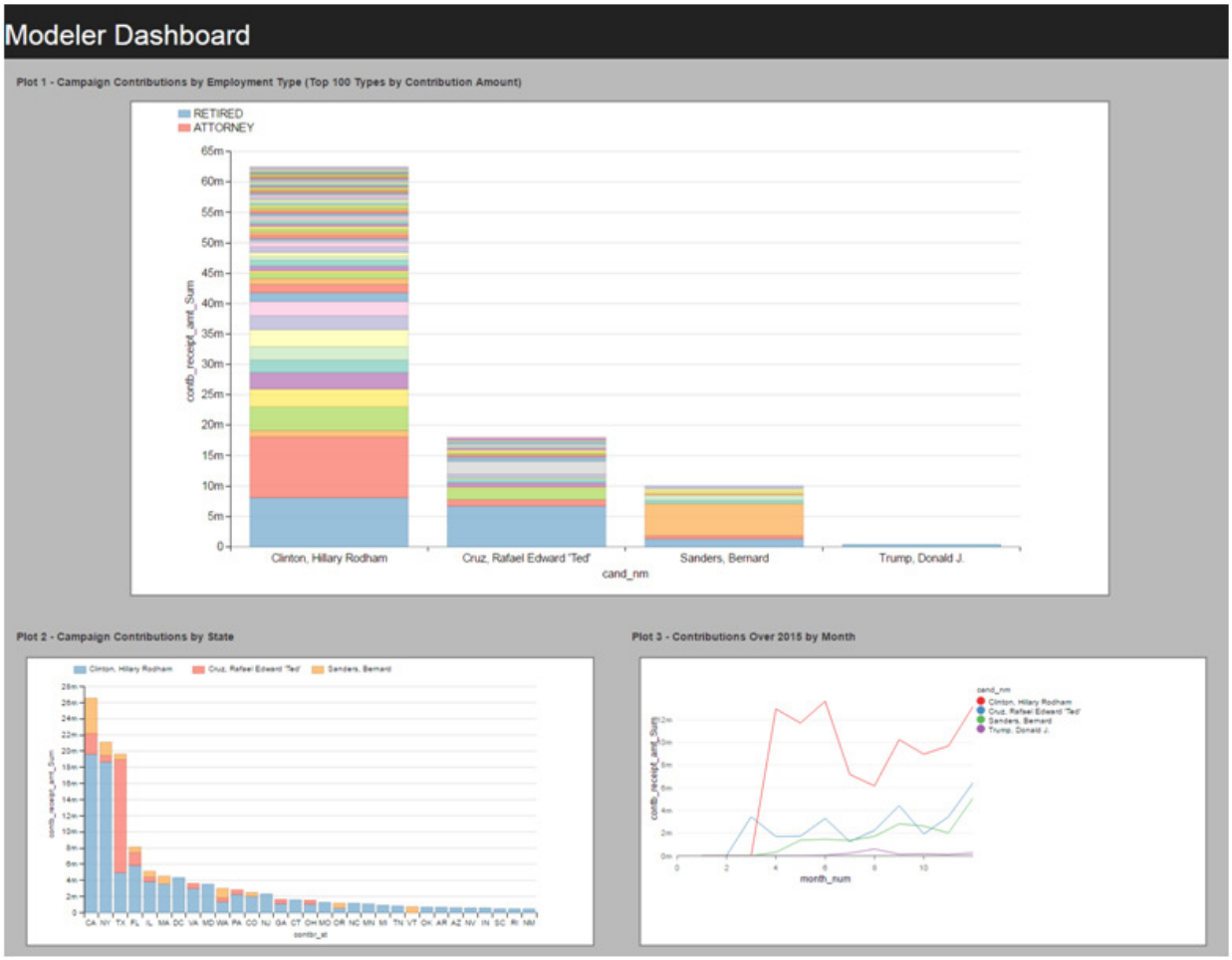
Step by Step Example – Building a Dashboard with rCharts

For this example, I will demonstrate how to use rCharts visualizations created in SPSS Modeler to create a dashboard that can be viewed in a web browser. This dashboard will be composed of two bar plots and one line plot and will contain data for 2015 campaign contributions for the U.S. Presidential election. First, I'll show a screen shot of what we will be building:

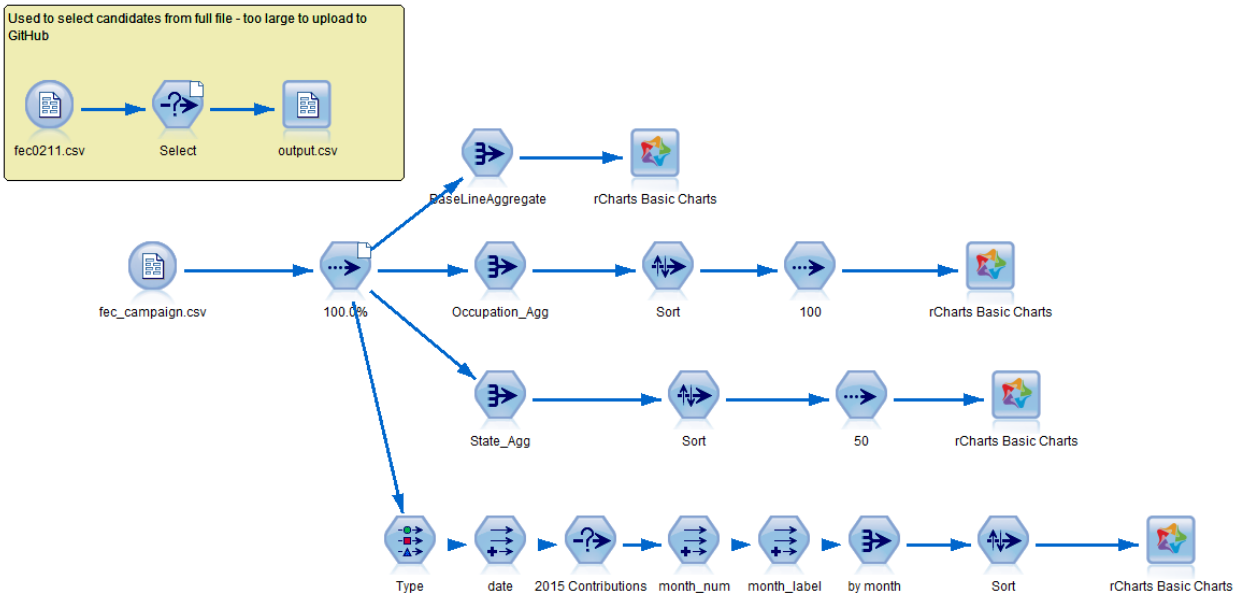


1. Get the data. I have included the CSV for this demo in the example folder for this extension's repository*. If you would like a more up to date version of the data, go to http://www.fec.gov/disclosurep/PDownload.do and select the data you are interested in.

   *GitHub has a 100mb limit on data files so a file subset was uploaded which only had the four candidates of interest. The stream still has the select node for reference but is not needed using the provided dataset.

2. Get the stream- You can either build the stream from scratch or download the copy that is in the example directory. I'll provide a quick overview with what we are doing in this stream

   a. After loading the data, we are 'Selecting' only the candidates of interest, I chose Bernie Sanders, Hillary Clinton, Donald Trump, and Ted Cruz. Note - this is only for the original data, the set on GitHub already has this processed.
   b. The top split goes to a sample node (used for testing), where we then split to aggregate for three different fields which allow us to create 3 different plots.
      i. Baseline aggregate just shows total contributions for candidate – it was not used for dashboard
      ii. State_Agg shows each state's contributions for each candidate
      iii. Occupation_Agg shows how employee groups contribute to each candidate
   c. The other main split was for the time series plot. This section does some date clean up and restricts contributions to only occur in 2015.



d. The end of each stream uses the rCharts Basic Charts node to create a plot. I have included the screenshots below for how each dialog should be configured to create the desired plot. For each plot we will put the contb_receipt_amt_Sum on the Y-axis and will adjust the X, Category, and Chart Type.

Note: For all plots we will use CDN and only this first plot will be 500x1000 (since it is in the top position), the other two plots will be 300x600.

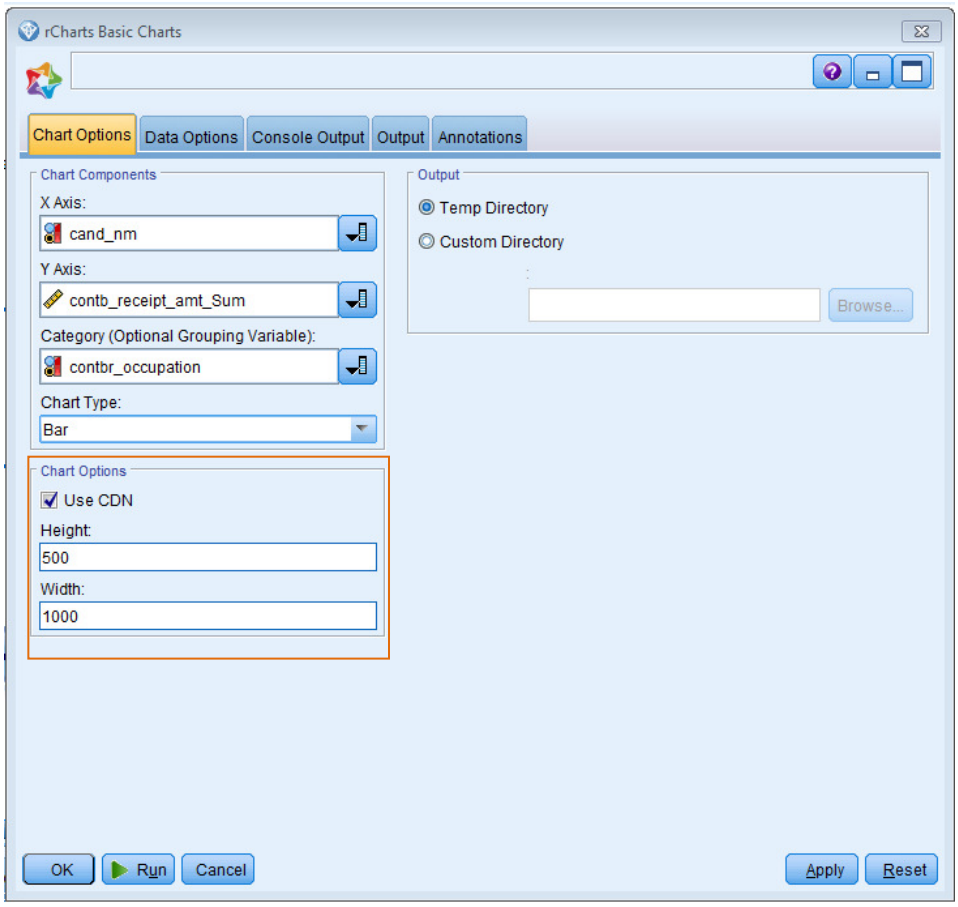


Figure 1 - Campaign Contributions by Employment Type (Top 100 Types by Contribution Amount)

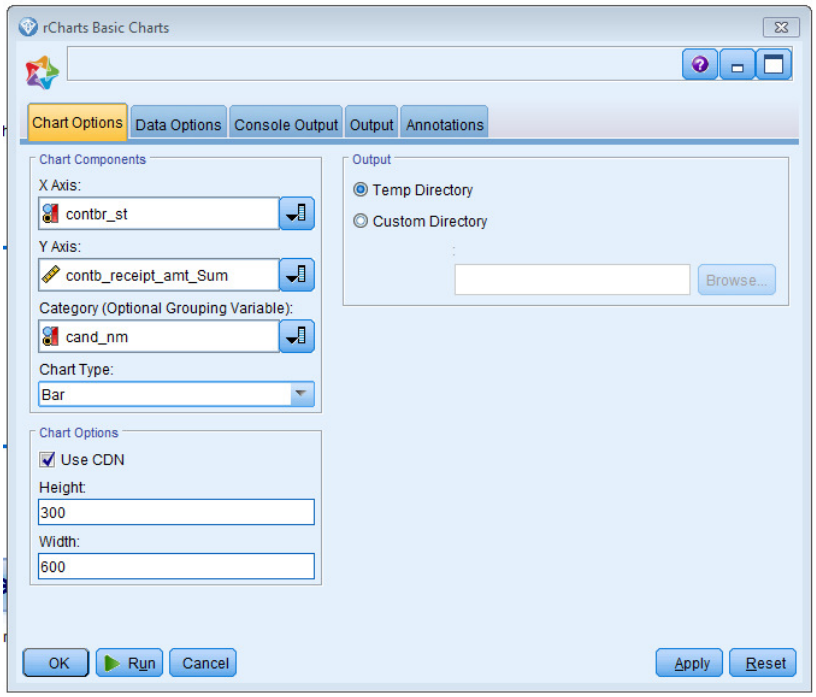


Figure 2- Campaign Contributions by State

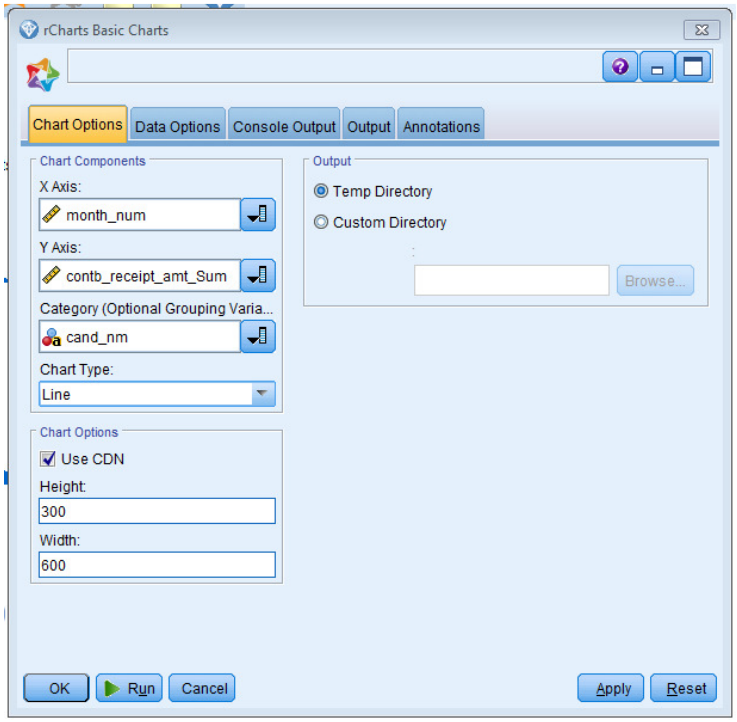
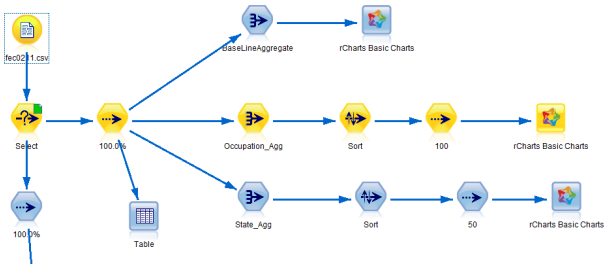
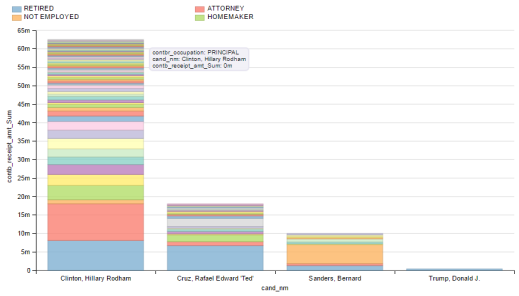


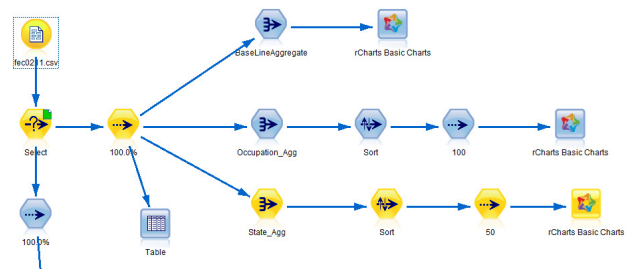
Figure 3- Contributions Over 2015 by Month

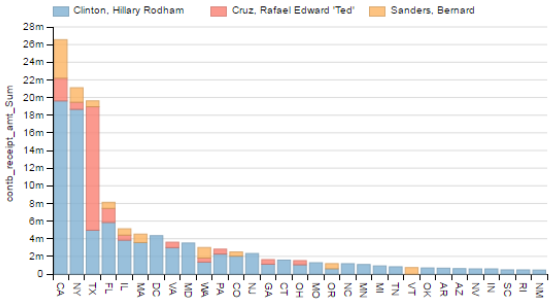
3. Let's run our streams to create our plots. First, run the streams as instructed below – this will open a browser with the plot results in a new tab. In the next section I will review taking the plot from the web page and putting it into our dashboard. For these steps, just run the streams and keep the plots open in your web browser.
    a. Plot 1:



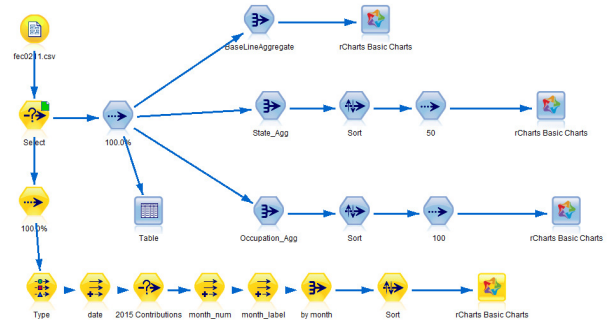


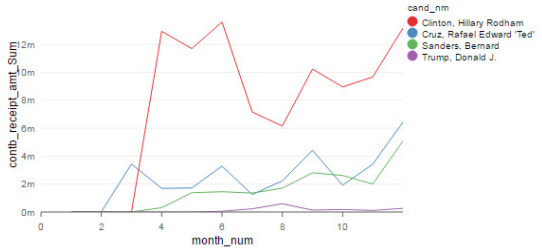
    b. Plot 2:





    c. Plot 3:



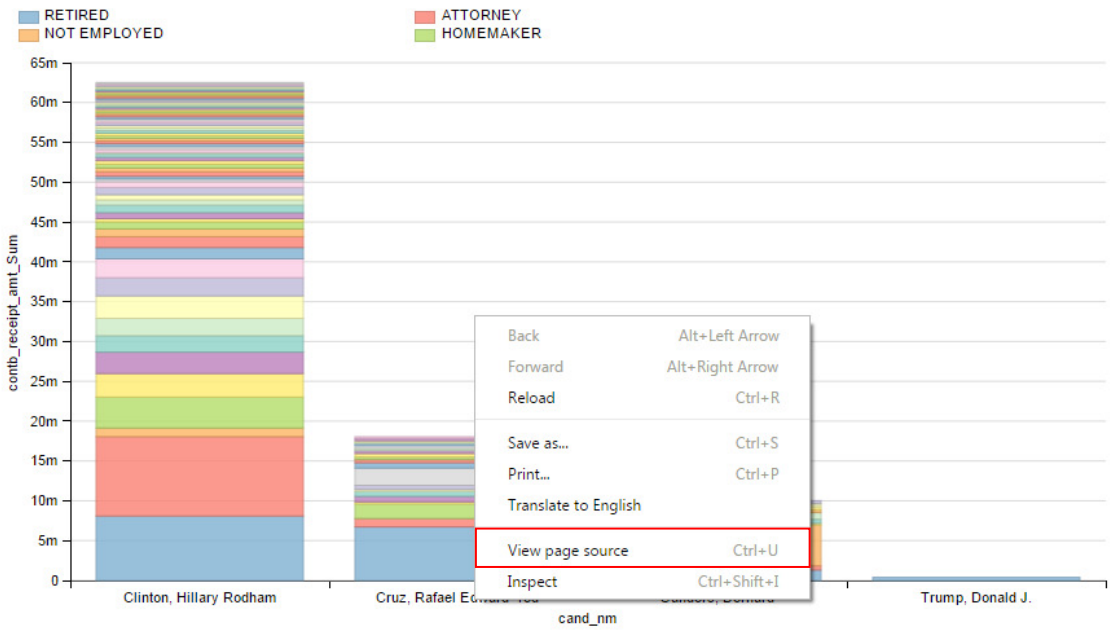


4. Populate Dashboard with plots
    a. In the example directory there is a document – “dashboard-empty.html”. Open this document in a text editor (If you only have something like Notepad, try out Notepad ++, it will make this a lot easier)
        i. If you open this document in a web browser you’ll see it looks a lot like the first screen shot in this guide except it has no plots – we’ll be adding those now.
        ii. At the end of this guide, I’ll make a couple notes about this html but for now let’s keep moving
    b. In this document I included the following text where we need to paste our plot code:

        <!-- Enter Plot x here -->
        Plot x Placeholder

```
        <div class="chart-wrapper">
            <div class="chart-stage">
<!-- Enter Plot 1 here -->
Plot 1 Placeholder

            </div>
        </div>
```

    c. Do a Ctrl-F to find this text or scroll down. Now we are in the right place, let’s go back to our browser and look at Plot 1 we created in the prior step.
    d. When viewing this plot, right click anywhere and click “View Source”



e. We want the entire contents of the body of this page. Toward the top of the page you will see something like this:

```
<body >

    <div id = 'chart29544a1bf0a' class = 'rChart dimple'></div>
```

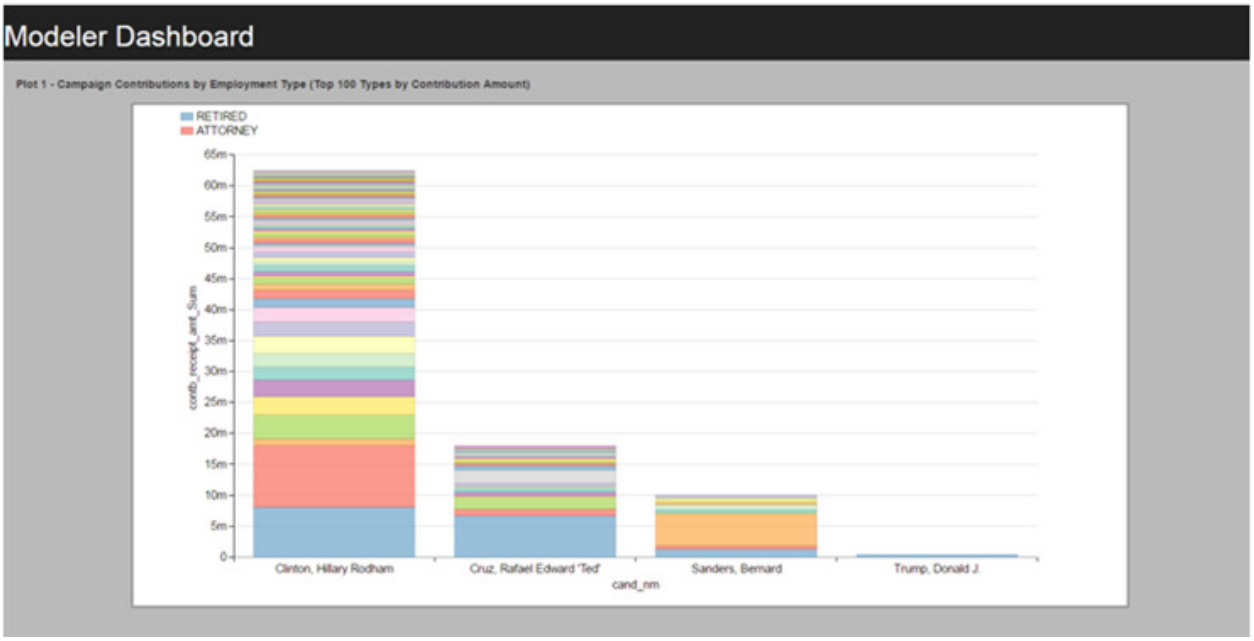
f. Select everything inside the body tag (not including the body tag) – so you would start selecting ‘ ‘ and scroll to the bottom of the page to stop at ‘<script></script> ‘ , if it looks like the text below:

```
    <script></script>

</body>
```

g. With that selected, copy, then go back to your text editor and replace the placeholder text (‘Plot 1 Placeholder’) with the text you copied. It will look something like this (Code was pasted after :

```
<div class="row">
  <div class="col-sm-1"></div>
  <div class="col-sm-10">
  <div class="chart-wrapper">
        <div class="chart-stage">

       <div id = 'chart1ee82b686686' class = 'rChart dimple'></div>
       <script type="text/javascript">
     var opts = {
    "dom": "chart1ee82b686686",
    "width":    1000,
    "height":    500,
    "xAxis": {
     "type": "addCategoryAxis",
    "showPercent": false
    },
    "yAxis": {
     "type": "addMeasureAxis",
    "showPercent": false
    },
    "zAxis": [],
    "colorAxis": [],
    "defaultColors": [],
    "layers": [],
    "legend": {
     "x":     60,
```

i. If you save this file now and open the file in a web browser then you will see the first plot in the top position like the screen shot below



h. Now repeat the steps of step 4 for the remaining 2 plots. If you have issues finding the right place to paste the plot code, refer to the dashboards.html file to see what the complete html looks like.
i. Now save your file and launch in a web browser and you should see the screen shot shown on page 1. I tested this page on Chrome.

HTML Template Notes:

- In the head of the html I pasted links to the CDN containing the JavaScript libraries necessary to create our visualizations. If you create your own template you will need to include this link for your visualization to work
- I also included a short style tag in the template add some CSS for styling. Everything is relatively straight forward but note that .tooltip { opacity: 1 !Important; } is required to see the text in the tooltip (hover) text.
- The template is using Bootstrap to help with the alignment of content. This was done for ease and the CSS and JS for Bootstrap are linked from a CDN as well.