PhUSE EU Connect 2018

Paper DV05

Using plotly.js to visualize SAS data

Volker Harm, Bayer AG, Berlin, Germany

ABSTRACT

Plotly.js is an open source JavaScript charting library based on d3.js. In the first part of the paper I will describe how easy it is to build stunningly interactive visualizations by presenting some plotly.js examples. This will also exemplify the general structure of plotly.js objects. The next step is to show how to work with.csv data. Finally I will show how SAS data sets can be provided by proc json and explore how SAS Stored Processes can be used to interactively visualize these data..

INTRODUCTION

In the past years I followed several paths to find ways to interactively present SAS data on the intranet resulting in ways to create web pages using SAS tools as ods html and proc stream. Another option was to incorporate JavaScript libraries as jQuery. But the hottest thing I found is plotly.js.

PLOTLY.JS OBJECTS

GETTING TO KNOW THE PLOTLY MODEBAR

At the top of every two-dimensional Plotly plot you will find icons with the following functions:

- Download plot as a PNG

  Clicking this will instantly start a download of the plot in PNG format.

- Zoom and Pan buttons
  Clicking and holding with your mouse allows you to zoom and pan. You can toggle between modes by clicking on the zoom or pan icons.
  - Zoom

    Clicking this selects the Zoom mode. To zoom in on a region of a Plotly graph, click and hold your mouse, moving across the region. Release your mouse. To return to the original view, double-click anywhere on the plot.

  - Pan

    To pan across regions of your graph, select the Pan mode. Click and hold your mouse to explore the data. Double-click anywhere to return to the original view.

- Zoom In/Out
  You can zoom in and out by clicking on the + and - buttons. Plotly keeps axes labels and annotations the same size to preserve readability. Your axes labels will automatically optimize as you zoom in.
  - Zoom In

- Zoom Out

- Autoscale and reset axes

When a graph is made using Plotly, we guess how the graph axes should be set. This is called “autoscale”. Sometimes our guess isn’t exactly what the graph creator wants. When a graph is made, the user can change the axes, and then save their graph. You can return to this saved form by clicking on Reset Axes.

- Reset Axes

Clicking this zooms to include your Axes Range, if this has been set. If it has not been set it zooms to a setting that is optimized to include all the viewable data, the same as if Autoscale had been clicked.

- Autoscale

Clicking Autoscale zooms the plot back to a setting that is optimized to include all the viewable data regardless of the Axes Range setting.

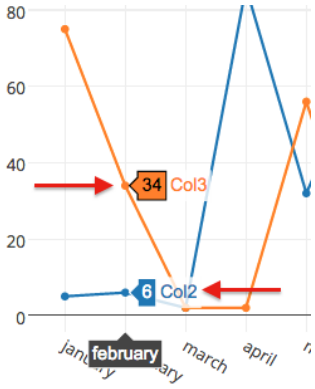
- Hover options

One of these two buttons is selected at all times.

- Show Closest Data Button

Clicking Show closest data on hover will display the data for just the one point under the cursor.

- Compare Data on Hover Button



Clicking Compare data on hover will show you the data for all points with the same x-value.

USING PLOTLY.JS WITH .CSV DATA

```
function makeplot() {

Plotly.d3.csv("https://raw.githubusercontent.com/plotly/datasets/master/2014_apple_
stock.csv", function(data){ processData(data) } );

};

function processData(allRows) {

  console.log(allRows);
  var x = [], y = [], standard_deviation = [];

  for (var i=0; i<allRows.length; i++) {
    row = allRows[i];
    x.push( row['AAPL_x'] );
    y.push( row['AAPL_y'] );
  }
  console.log( 'X',x, 'Y',y, 'SD',standard_deviation );
  makePlotly( x, y, standard_deviation );
}

function makePlotly( x, y, standard_deviation ){
  var plotDiv = document.getElementById("plot");
  var traces = [{
    x: x,
    y: y
  }];

  Plotly.newPlot('myDiv', traces,
    {title: 'Plotting CSV data from AJAX call'});
};
  makeplot();
```

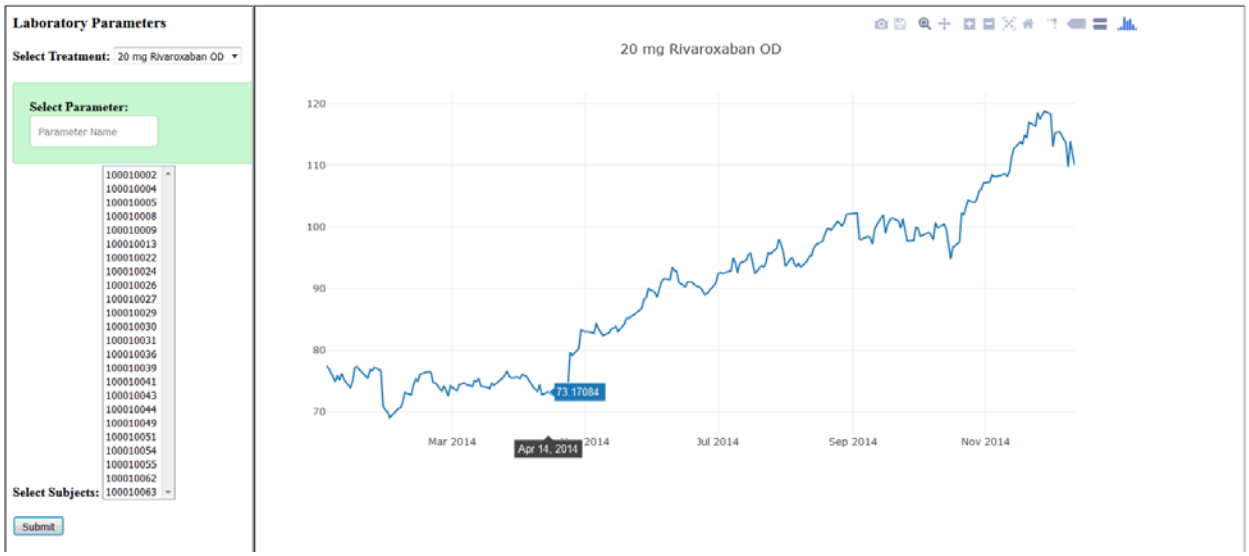
USING PROC JSON TO WRITE SAS DATA AS JAVASCRITP OBJECTS

```
filename jsontxt '\\BYARXC4\Kunde28_User$\SGJQX\Personal Data\My SEG Projects\proc
stream\Events.json';

proc json out = jsontxt pretty nosastags;
    export blogpostings /fmtdatetime;
run;
```

USING PROC STREAM TO WRITE WEB CONTENT

```
%let fref = jsontxt;
%let rc = %sysfunc (filename(fref, , temp));
%let rc =
    %sysfunc (
           dosubl(

                  proc json out = jsontxt pretty nosastags;
                       export blogpostings / fmtdatetime;
                  run;
           )
    );
&streamDelim readfile jsontxt;
```

USING SAS STORED PROCESSES TO PROVIDE INTERACTIVE VISUALIZATIONS



CONCLUSION

Although Plotly is much better integrate into other software packages asR, Python, or MatLab, plootly.js gives the basic means to also present SAS data in that modern professional way interactively in a web browser.

REFERENCES

Henderson, Don. 2014. “PROC STREAM and SAS® Server Pages: Generating Custom HTML Reports”. Proceedings of the SAS Global 2014 Conference. Available online at http://support.sas.com/resources/papers/proceedings14/1738-2014.pdf.

Hinson, Joseph. 2016. “The New STREAM Procedure as a Virtual Medical Writer”. Proceedings of the PharmaSUG 2016 Conference. Available online at https://www.pharmasug.org/proceedings/2016/AD/PharmaSUG-2016-AD17.pdf.

Weng, Anni, Zhou, Huanxue. 2018. “Graphic Visualization for Treatment Switch Pattern”. Proceedings of the PharmaSUG 2018 Conference. Available online at https://www.pharmasug.org/proceedings/2018/DV/PharmaSUG-2018-DV16.pdf.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Volker Harm
Bayer AG
Muellerstrasse 178
Berlin 13353
Work Phone: 004946811208
Email: volker.harm@bayer.com