Rich Data Visualization

With the booming popularity of big data and data science, nice visualizations are getting a lot of attention. Sure, R and Python have built-in support for basic graphs and charts, but what if you want more. What if you want interaction so you can mouse-over or rotate a visualization. What if you want to explore more than a static image? Enter ***Rich Visualizations***.

And, creating them is not as hard as you might think!

html....what?

The wonderful R programming language has **htmlwidgets (http://www.htmlwidgets.org/)** which brings all the interactivity of JavaScript into your R code. That means R can now create beautiful and interactive data visualizations to display in your browser.

Below are some examples from the well-documented htmlwidgets showcase (http://www.htmlwidgets.org/showcase_dygraphs.html).

All these examples and more can be seen on the sample project at:

- **htmlwidgets and R**

Try Out Some htmlwidgets

Let's start out with a simple table. Sometimes it is beneficial to be able to sort and filter tabular data.

Data Table

Well, the DataTables R package (http://rstudio.github.io/DT/) provides exactly that in only two lines of code.

```
library('DT')
datatable(iris, options = list(pageLength = 10))
```

Time Series Line Graph

The dygraphs (http://rstudio.github.io/dygraphs/) package allows for very simple time series plotting.

```
library('dygraphs')
dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(dateWindow = c("1926-01-01", "1970-01-01"))
```

Visit Sense

Notice a few things. Very minimal code is required. Also, you can hover over the line and the exact values are dynamically displayed at the top. Finally, the date range can be expanded or contracted with the slider at the bottom.

Network Graph

A network visualization graph can be created with the help of the networkD3 (https://christophergandrud.github.io/networkD3/) package. It uses the D3 JavaScript library to create a very nice looking network graph. A network graph is useful for viewing relationships and connections in social networks. You can click on the nodes and even drag them around. Have fun with this one.

```
install.packages('networkD3')
library('networkD3')
data(MisLinks, MisNodes)
forceNetwork(Links = MisLinks, Nodes = MisNodes,
  Source =  "source", Target = "target", Value = "value",
  NodeID = "name", Group = "group", opacity = 0.4)
```

rThreeJS

ThreeJS (http://threejs.org/) is a JavaScript library to create 3D WebGL in your browser. Cool huh? Well, rThreeJS (https://github.com/bwlewis/rthreejs) brings that awesomeness to your R code.

```
library('threejs')
z <- seq(-10, 10, 0.01)
x <- cos(z*4)
y <- sin(z*4)
scatterplot3js(x,y,z, color=rainbow(length(z)))
```

Try clicking and dragging the object. You can really explore in 3D.

Conclusion

That should be enough to get you started on your rich visualization journey. Fork the sample project, **htmlwidgets and R**, (https://sense.io/ryanswanstrom/htmlwidgets-and-r) on Sense and start making your own rich visualizations. The sample project also contains extra examples of a scatterplot and flowchart. As a bonus, Sense (http://sense.io) allows your visualizations to easily be embedded on any webpage (just like you see on this blog).

Keep watching the blog for other tools you can use for creating rich data visualizations in Python and Julia.

Follow along on Facebook (https://facebook.com/SensePlatform), Twitter (https://twitter.com/SensePlatform) or RSS (http://blog.sense.io/rss/) to get the latest updates from the blog and Sense.io (http://sense.io).

Ryan Swanstrom