



rCharts

Building Data Products

Brian Caffo, Jeff Leek, Roger Peng
Johns Hopkins Bloomberg School of Public Health

rCharts



- rCharts is a way to create interactive javascript visualizations using R
- So
 - You don't have to learn complex tools, like D3
 - You simply work in R learning a minimal amount of new syntax
- rCharts was written by Ramnath Vaidyanathan (friend of the Data Science Series), who also wrote slidify, the framework we use for all of the lectures in the class
- This lecture is basically going through (<http://ramnathv.github.io/rCharts/>)

Example

```
require(rCharts)
haireye = as.data.frame(HairEyeColor)
nPlot(Freq ~ Hair, group = 'Eye', type = 'multiBarChart',
      data = subset(haireye, Sex == 'Male')
)
```

nvD3 run

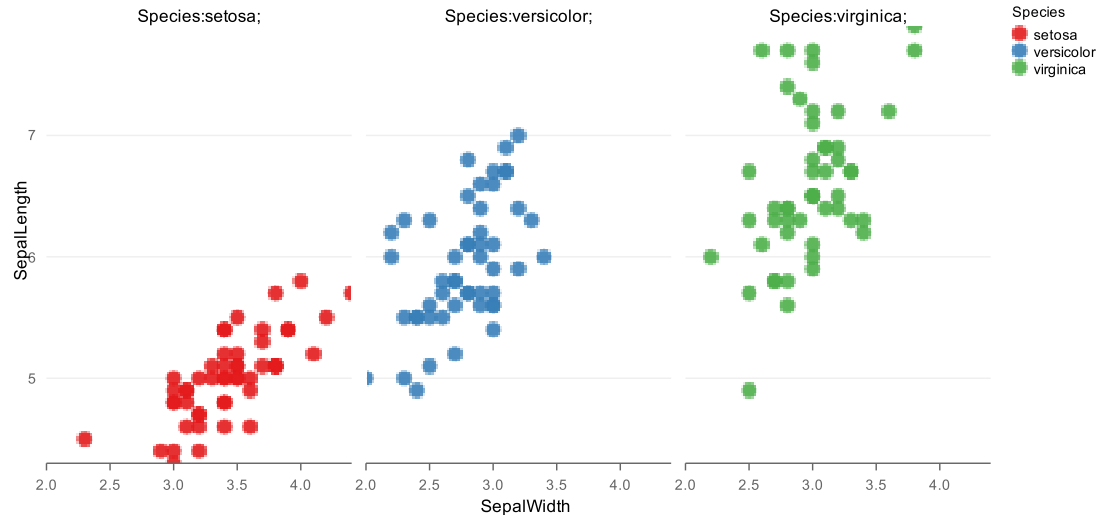
Slidify interactive

- The above was an example of embedding an rChart in a slidify document
 - In the YAML `yaml` `ext_widgets` : `{rCharts: ["libraries/nvd3"]}`
- Or, if you use more than one library
- YAML `example` `yaml` `ext_widgets` : `{rCharts: ["libraries/highcharts",libraries/nvd3", "libraries/morris"]}`

Deconstructing another example

```
## Example 1 Facetted Scatterplot
names(iris) = gsub("\\\\.", "", names(iris))
rPlot(SepalLength ~ SepalWidth | Species, data = iris, color = 'Species', type = 'point')
```

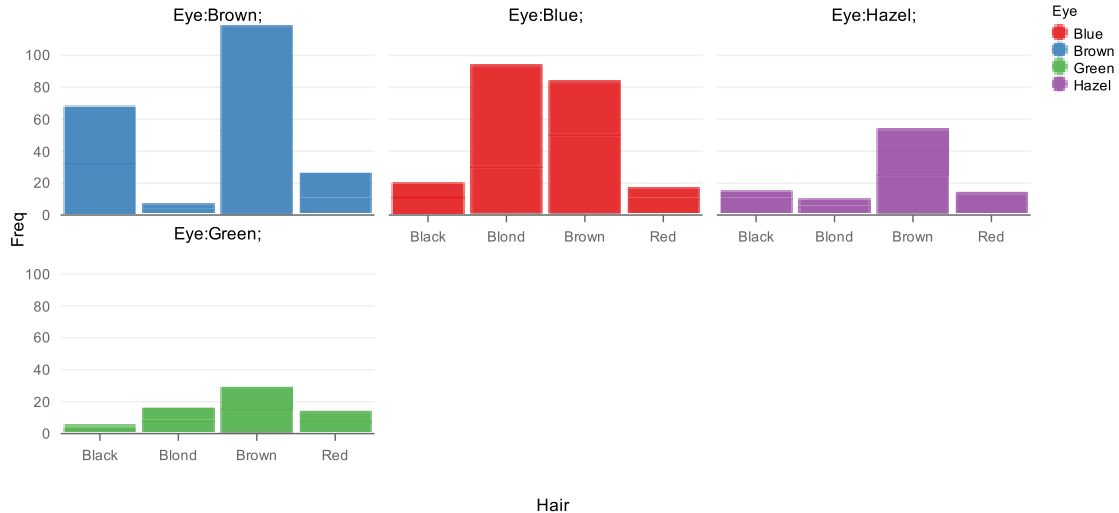
When run



Example 2 Facetted Barplot

```
hair_eye = as.data.frame(HairEyeColor)
rPlot(Freq ~ Hair | Eye, color = 'Eye', data = hair_eye, type = 'bar')
```


Example 2 Facetted Barplot, when run



How to get the js/html or publish an rChart

Now you can add whatever you'd like

```
r1 <- rPlot(mpg ~ wt | am + vs, data = mtcars, type = "point", color = "gear")
r1$print("chart1") # print out the js
r1$save('myPlot.html') #save as html file
r1$publish('myPlot', host = 'gist') # save to gist, rjson required
r1$publish('myPlot', host = 'rpubs') # save to rpubs
```

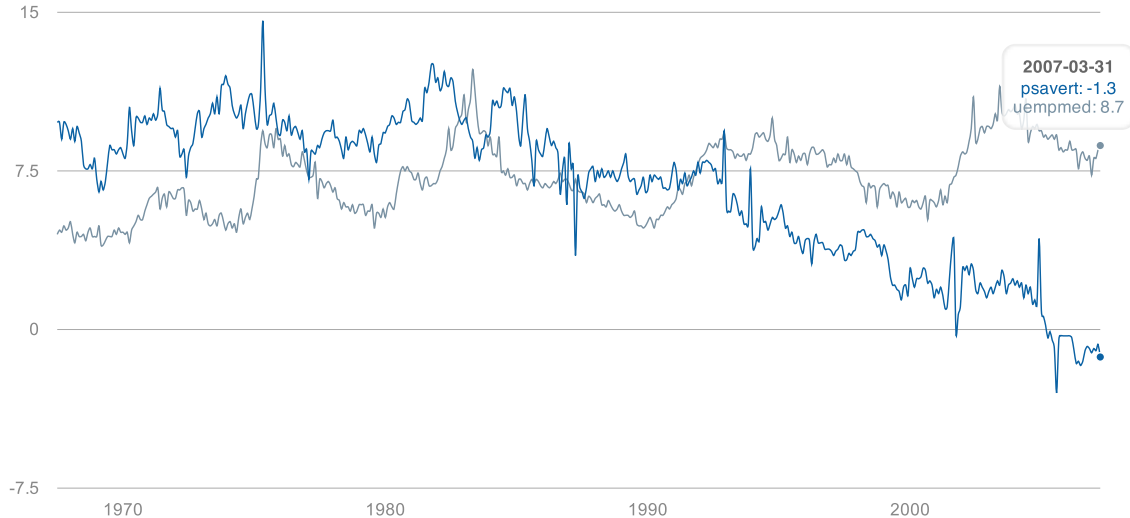
rCharts has links to several libraries

- We'll do some examples
- Note Ramnath mentions that io2012 and polychart have conflicting js
 - They seem to work for me with that theme, but I get errors if I load the polychart library
 - If debugging with io and polychart, factor that in

morris

```
data(economics, package = "ggplot2")
econ <- transform(economics, date = as.character(date))
m1 <- mPlot(x = "date", y = c("psavert", "uempmed"), type = "Line", data = econ)
m1$set(pointSize = 0, lineWidth = 1)
m1
```

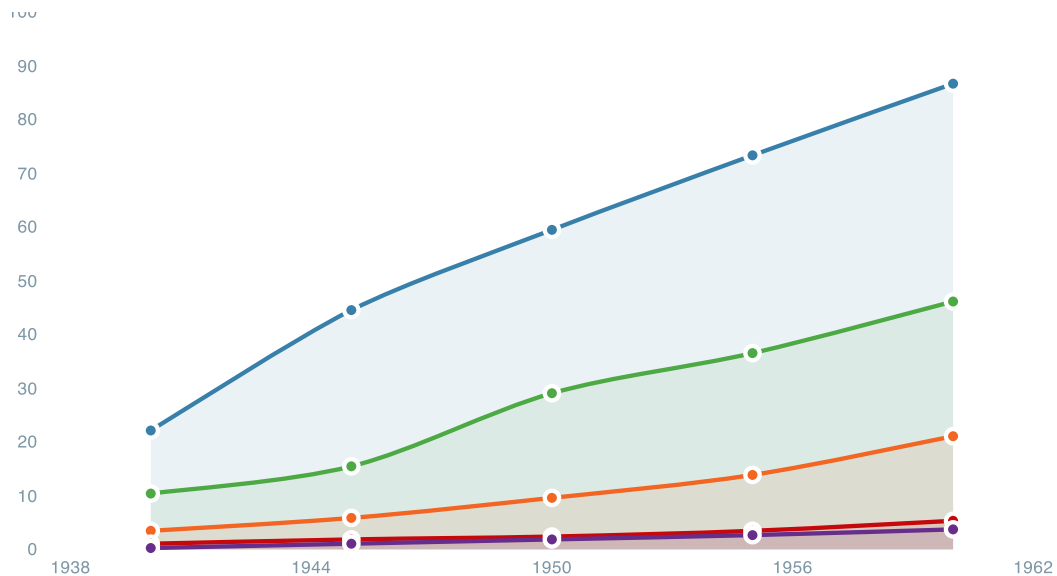
morris example run



xCharts

```
require(reshape2)
uspexp <- melt(USPersonalExpenditure)
names(uspexp)[1:2] = c("category", "year")
x1 <- xPlot(value ~ year, group = "category", data = uspexp, type = "line-dotted")
x1
```

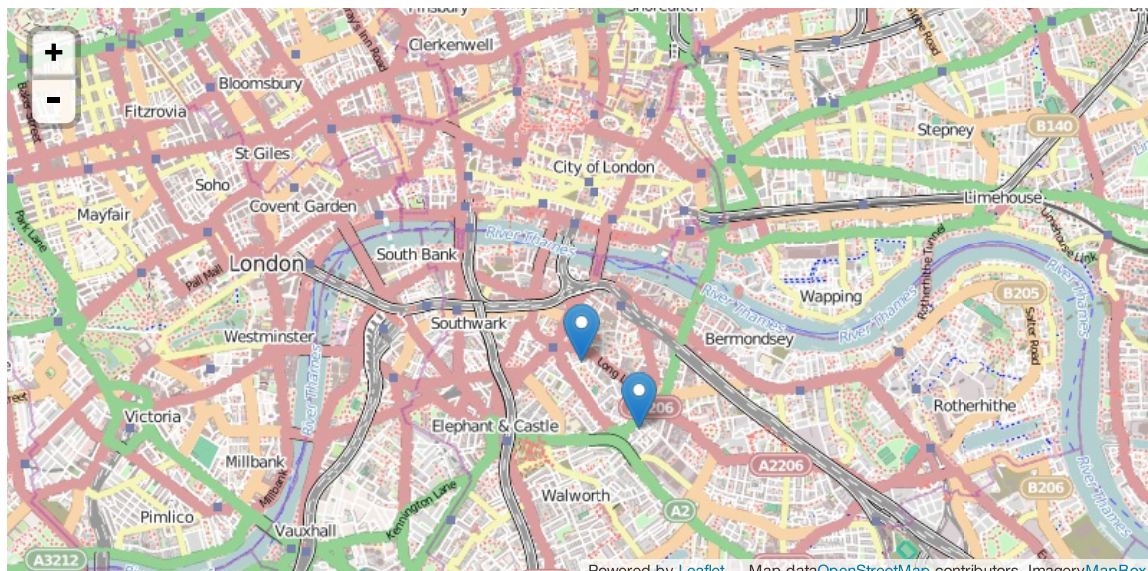
xCharts run



Leaflet

```
map3 <- Leaflet$new()
map3$setView(c(51.505, -0.09), zoom = 13)
map3$marker(c(51.5, -0.09), bindPopup = "<p> Hi. I am a popup </p>")
map3$marker(c(51.495, -0.083), bindPopup = "<p> Hi. I am another popup </p>")
map3
```

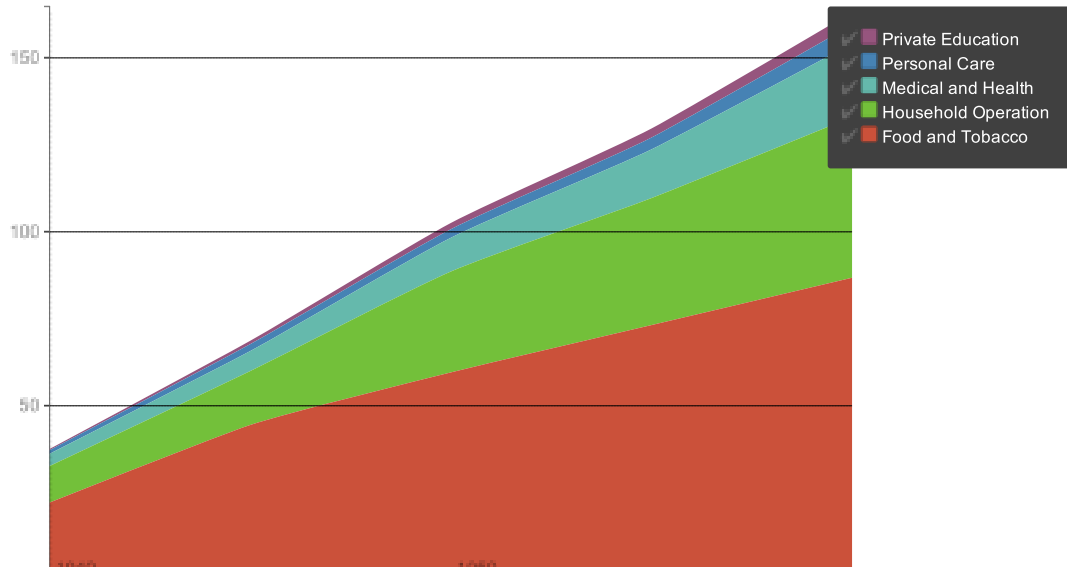

Leaflet run



Rickshaw

```
usp = reshape2::melt(USPersonalExpenditure)
# get the decades into a date Rickshaw likes
usp$Var2 <- as.numeric(as.POSIXct(paste0(usp$Var2, "-01-01")))
p4 <- Rickshaw$new()
p4$layer(value ~ Var2, group = "Var1", data = usp, type = "area", width = 560)
# add a helpful slider this easily; other features TRUE as a default
p4$set(slider = TRUE)
p4
```

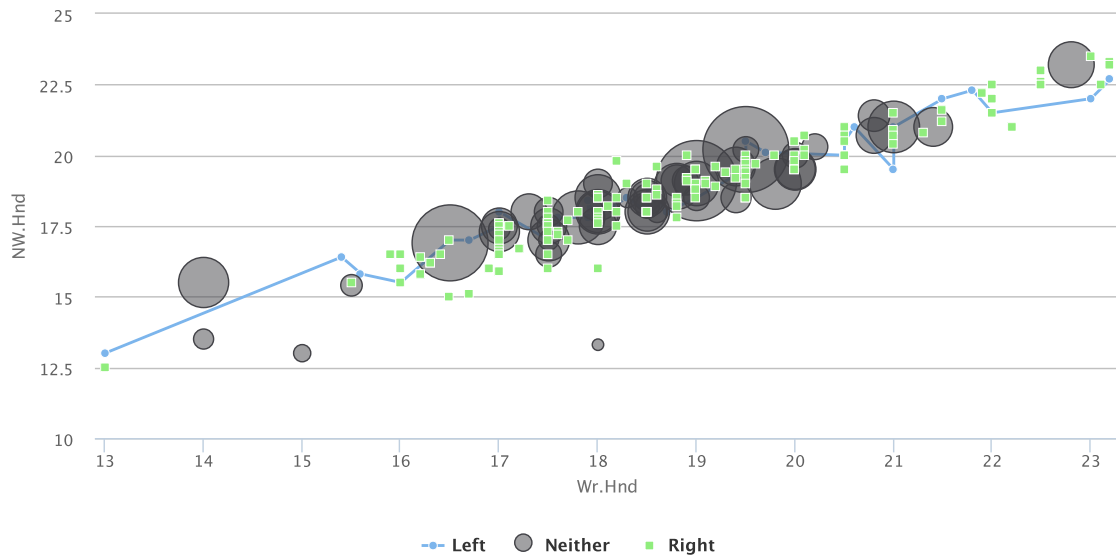
Rickshaw run



highchart

```
h1 <- hPlot(x = "Wr.Hnd", y = "NW.Hnd", data = MASS::survey, type = c("line",  
  "bubble", "scatter"), group = "Clap", size = "Age")  
h1
```

highchart run



Embedding rCharts into a shiny app

(then embedding it into slidify)

What to do now

- Go forth, and produce some eye candy