Interactive web visualizations for R (Special Types)

CHEAT SHEET

Map(Leaflet)

Installation:
install.packages("leaflet")

Basemaps
m <- leaflet() %>% setView(lng , lat , zoom) %>% addTiles()

Fit the View
fitBounds(lng1, lat1, lng2, lat2)

Popups
addPopups(lng , lat , content, options)

Markers
addMarkers(lng, lat, popup,label)

Circles
addCircles(lng, lat, weight, radius, ...)

Rectangles
addRectangles(lng1, lat1, lng2, lat2, fillColor, ...)

Polygons
addPolygons(color, weight, smoothFactor, opacity, fillOpacity, fillColor,highlightOptions, …)

Color
Pal<- colorNumeric(palette,domain,alpha,reverse)

Legend
addLegend("bottomright", pal , values, title, labFormat,opacity)

Time Series Charting (Dygraphs)

Installation:
install.packages("dygraphs")

Basic Use
plot<-dygraph(nhtemp, main = “Main Title”, ylab)

Set the default time range
plot%>%dyRangeSelector(dateWindow))

Candlestick
plot%>%dyCandlestick()

Time Zones
plot%>%dyOptions(labelsUTC = TRUE)

Diagram(DiagrammeR)

Installation:
install.packages('DiagrammeR')

Basic:

```
grViz(“
      digraph{ #statement of the graph here}
)
```

'graph' statement
graph [overlap = true, fontsize = 10]

‘node’ statememt
node [shape = circle, fixedsize = true, width = 0.9]
node_name1; node_name2……

'edge' statements
node_name1->node_name2
node_name2->node_name3
……

3D ScatterPlot(threejs)

Installation:
install.packages(“threejs”)

Basic:

```
z <- seq(-10, 10, 0.1)
x <- cos(z)
y <- sin(z)
scatterplot3js(x, y, z, color=rainbow(length(z)))
```

Interactive ScatterPlot (metricsgraphics)

Installation:
devtools::install_github("hrbrmstr/metricsgraphics")

Basic:
Plot<-dataframe %>%mjs_plot(x,y)

Connect with line
Plot%>% mjs_line()

Add Baseline
Plot%>%mjs_add_baseline(150, “text”)

Interactive Heatmap(heatmaply)

Installation: install.packages(“heatmaply”)

Basic:

```
p <- heatmaply(mat,
      dendrogram = "none", xlab = "", ylab = "",
      main = "", scale = "column",
      margins = c(60,100,40,20), grid_color = "white",
      grid_width = 0.00001, hide_colorbar = TRUE,
      branches_lwd = 0.1,
      label_names = c("", "", ""),
      fontsize_row = 5, fontsize_col = 5,
      labCol = colnames(mat),
      labRow = rownames(mat),
      heatmap_layers =
theme(axis.line=element_blank()) )
```

Word Cloud

Installation:
devtools::install_github("lchiffon/wordcloud2")

Basic:

```
library(wordcloud2)
wordcloud2(data, size = 1,
shape = 'star‘ ,
 minRotation = -pi/6,
 maxRotation = -pi/6,
fontFamily = "",
 color = "random-light",
)
```

WebGL scenes(rglwidget)

Installation:
install.packages(“rglwidget”)

Basic:

```
theta <- seq(0, 6*pi, len=100)
xyz <- cbind(sin(theta), cos(theta), theta)
lineid <- plot3d(xyz, type="l", alpha = 1:0,
          lwd = 5, col = "blue")["data"]
```

Or… You can directly make your existed ggplot to be interactive

Installation:
install.packages(“ggiraph”)

Basic:
Instead of using geom_point, use geom_point_interactive, instead of using geom_sf, use geom_sf_interactive… Provide at least one of the aesthetics tooltip, data_id and onclick to create interactive elements.

Interactivity is added to ggplot geometries, legends and theme elements, via the following aesthetics:

tooltip:
tooltips to be displayed when mouse is over elements.

onclick:
JavaScript function to be executed when elements are clicked.

data_id:
id to be associated with elements (used for hover and click actions)

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

Interactive web visualizations for R (General Types)

CHEAT SHEET

plotly

To make the general Interactive web visualizations , we can also use plotly. Plotly is an R package for creating interactive web-based graphs via the open source JavaScript graphing library plotly.js. It can be used for multiple purposes.

How to use?

1.Install
install.packages('plotly')

2. Sign up& Configure
plot.ly/r/getting-started

3. Make the plot

4. Plot the figure
p or print(p)

```
library(plotly)
plot<-plot _ly ( x ,y,
 type = 'scatter' ,
 mode = 'markers' ,
 size=~z, color=~z,
 marker = list (
 color = c( 'red',
'blue' , 'green' )))
```

x,y should be vector

Chart Type

Use line/marker, etc. to represent each observation.

Size, Color can be connected to other variables in the dataset

Chart	Type=?
2D	
Line Plots	"scatter" with mode='lines"
Scatter Plots/3D	"scatter" with mode="markers"
Bar charts	"bar" with mode="markers"
Heatmap	"heatmap"
Box Plot	"box"
2D Histogram	'histogram2d'
3D	
3D surface plot	"surface"
3D Line Plot	'scatter3d' with mode = 'lines'
3D scatter Plot	'scatter3d' with mode = 'markers'

Layout

Legends

```
plot%>$layout ( legend = list( x = 0.5 , y = 1 , bgcolor = '#F3F3F3' ))
```

Axis

Number of ticks on the axis

```
axis_template <- list ( showgrid = F , zeroline = F , nticks = 20 , showline = T , title = 'Axis Title' , mirror = 'all' )
plot%>%layout ( xaxis = axis _template , yaxis = axis _template )
```

Subplots

```
fig <- subplot(fig1, fig2) %>%
layout(title)
```

2 subplots!

Easily Convert the ggplot to plotly:

```
ggplotly(scatterPlot)
```

Put your ggplot here!