# Table of Contents

Plot.ly Charts using PluginR and Tiki ................................................................. 2

*Installing Plotly* ................................................................. 2

*Overlaid Histograms* ............................................................. 3

*Log-normal Boxplot* ............................................................... 4

*HeatMaps* ........................................................................ 5

*More types of charts...* ............................................................... 6

*Collaborating and Sharing* ............................................................... 7
Plot.ly Charts using PluginR and Tiki

See
- https://plot.ly/
- https://plot.ly/api/r/

The Plotly R graphing library allows you to create and share interactive, publication-quality plots in your browser. Plotly is also built for working together, and makes it easy to post graphs and data publicly with a URL or privately to collaborators.

Page contents
- Installing Plotly
- Overlaid Histograms
- Log-normal Boxplot
- HeatMaps
- More types of charts...
- Collaborating and Sharing

Installing Plotly

```r
# Install de required packages if you don't have them yet
if(!require(devtools)){
  install.packages("devtools",
  repos="http://ftp.heanet.ie/mirrors/cran.r-project.org/"
} 
require("devtools")
if(!require(RCurl)){ install.packages("RCurl",
  repos="http://ftp.heanet.ie/mirrors/cran.r-project.org/"
} 
if(!require(bitops)){ install.packages("bitops",
  repos="http://ftp.heanet.ie/mirrors/cran.r-project.org/"
} 
if(!require(RJSONIO)){ install.packages("RJSONIO",
  repos="http://ftp.heanet.ie/mirrors/cran.r-project.org/"
}

# Next, install plotly (a big thanks to Hadley, who suggested the GitHub route):
if(!require(plotly)){ devtools::install_github("plotly/R-api") }
```
# Then sign-up like this (adapt username and email to your choice) or at https://plot.ly/:

```r
require(plotly)
response = signup (username = 'yournewusername', email= 'youremail@example.com')
```

# You’ll get as output in the R console:

## Thanks for signing up to plotly!

## Your username is: yournewusername

## Your temporary password is: yourtemppassword. You use this to log into your plotly account at https://plot.ly/plot.

## Your API key is: API_Key. You use this to access your plotly account through the API.

## To get started, initialize a plotly object with your username and api_key, e.g.

```r
p <- plotly(username="yournewusername", key="API_Key")
```

## Then, make a graph!

```r
res <- p$plotly(c(1,2,3), c(4,2,1))
```

And we’re up and running! You can change and access your password and key in your homepage.

---

**Overlaid Histograms**

```r
x0 = rnorm(500)
x1 = rnorm(500)+1
data0 = list(x=x0,
    type='histogramx',
    opacity=0.8)
```
data1 = list(x=x1,  
              type='histogramx',  
              opacity=0.8)  
layout = list(barmode='overlay')

response = p$plotly(data0, data1, kwargs=list(layout=layout))

# The script makes a graph. Use the RStudio viewer or add  
"browseURL(response$url)" to your script  
# to avoid copy and paste routines of your URL and open the graph directly.
#browseURL(response$url)

# In Tiki, you can plot the graph in a wiki page by means of an iframe to the response$url  
#  
#cat(response$url)  
#  
# But since Tiki doesn't allow the word "url" in scripts, we will use the  
# index of the value in the response list: cat(unlist(response[1]))  
cat(unlist(response[1]))  
# This will produce something like:  
#  
#"https://plot.ly/~yournewusername/0/"  
#  
# Then you just need to include that url in an iframe as usual in Tiki  
{RR}

{iframe name=myPlotlyChart width=800 height=600 align=middle frameborder=0  
marginheight=0 marginwidth=0 scrolling=auto  
src="https://plot.ly/~yournewusername/0/"}

**Log-normal Boxplot**

{RR(echo="1", cacheby="pagename", wikisyntax="0")}
# 2. Log-normal Boxplot  
# ------------------------
require(plotly)  
# ------------------------

p <- plotly(username='USERNAME', key='API_KEY')

x <- c(seq(0,0,length=1000),seq(1,1,length=1000),seq(2,2,length=1000))
y <- c(rlnorm(1000,0,1),rlnorm(1000,0,2),rlnorm(1000,0,3))
s <- list(
  type = 'box',
  jitter = 0.5
)
layout <- list(
  title = 'Fun with the Lognormal distribution',
  yaxis = list(
    type = 'log'
  )
)

response <- p$plotly(x,y, kwargs = list(layout = layout, style=s))

#browseURL(response$url)
# Again, in Tiki, you can plot the graph in a wiki page by means of an
# iframe to the response$url = unlist(response[1])
{RR}

{iframe name=myPlotlyChart width=800 height=600 align=middle frameborder=0 marginheight=0 marginwidth=0 scrolling=auto src="https://plot.ly/~ueb/14/"}

HeatMaps

{RR(echo="0", cacheby="pagename", wikisyntax="0")}

# Days of the Week Heatmap Demo
# Questions? Email feedback@plot.ly
# For more docs, see plot.ly/api

require(RColorBrewer)
require(plotly)

py <- plotly(username='yournewusername', key='API_Key')

x <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
y <- c('Morning', 'Afternoon', 'Evening')
z <- list(
  c(1., 20., 30, 50, 1),
  c(20., 1., 60, 80, 30),
  c(30., 60., 1., -10, 20)
)

# Data for the heatmap
heatmap_data <- matrix(c(1, 20, 30, 50, 1,
                         20, 1, 60, 80, 30,
                         30, 60, 1, -10, 20), nrow=3, byrow=TRUE)

# Create a heatmap using plotly package
heatmap <- plot_ly(py, z = heatmap_data, x = x, y = y, type = 'heatmap')

# Update the heatmap with different colors
heatmap <- heatmap.update(zcolorscale = 'Viridis')

# Show the heatmap
print(heatmap)
```r
# Color brewer YlOrBr colorscale http://colorbrewer2.org
# scl=\[[0,"rgb(128, 0, 38)"],[0.125,"rgb(189, 0, 38)"],[0.25,"rgb(227, 26, 28)"],
#       [0.375,"rgb(252, 78, 42)"],[0.5,"rgb(253, 141, 60)"],[0.625,"rgb(254, 178, 76)"],
#       [0.75,"rgb(254, 217, 118)",[0.875,"rgb(255, 237, 160)"],[1,"rgb(255, 255, 204)\]]

scl <- brewer.pal(9,'YlOrBr')
data <- list(
  x = x,
  y = y,
  z = z,
  scl= list(
    c(0,"rgb(128, 0, 38)"),
    c(0.125,"rgb(189, 0, 38)"),
    c(0.25,"rgb(227, 26, 28)"),
    c(0.375,"rgb(252, 78, 42)"),
    c(0.5,"rgb(253, 141, 60)"),
    c(0.625,"rgb(254, 178, 76)"),
    c(0.75,"rgb(254, 217, 118)"),
    c(0.875,"rgb(255, 237, 160)"),
    c(1,"rgb(255, 255, 204)")
  ),
  type = 'heatmap'
)
response <- py$plotly(data)

# url and filename
#unlist(response[1])
#filename <- response$filename

{RR}
{iframe name=myPlotlyChart width=800 height=600 align=middle frameborder=0 marginheight=0 marginwidth=0 scrolling=auto src="https://plot.ly/~ueb/15/"}
```
More types of charts...

For more examples of chart types possible with Plot.ly and R, see:
- https://plot.ly/api/r/

Collaborating and Sharing

# Collaborating and Sharing: You’re in Control
# ---------------------------------------------
#
# Keep in mind that:
#
# (1) You control if graphs are public or private, and who you share with
# (like Google Docs)
# (2) Public sharing in Plotly is free (like GitHub).
# To share privately, press “Share” in our GUI or share with your script.
# Users you share with get an email and can edit and comment on graphs.
# That means no more emailing data, graphs, screenshots, and spreadsheets
# around: you can do it all in Plotly.
# You can also save and apply custom themes to new data to avoid re-making
# the same graphs with new data.
# Just upload and apply your theme.