

Make Your Data Dance with Plotly

Spring Shi

Introduction to Plotly

- Plotly is a data visualization package that allows you to build interactive graphs.
- Example of plotly Graph:
 - <https://plot.ly/r/line-and-scatter/>
 - <https://plot.ly/r/choropleth-maps/>
- Plotly has both a graphic UI and API libraries for R, Python, Julia etc.
- Plotly allows you to convert your ggplot or matplotlib graph easily.



How to build a plotly graph

- `library(plotly)`
- `set.seed(100)`
- `d <- diamonds[sample(nrow(diamonds), 1000),]`
- `plot_ly(d, x = ~carat, y = ~price, color = ~carat,`
- `size = ~carat, text = ~paste("Clarity: ", clarity))`
- `library(plotly)`
- `trace_0 <- rnorm(100, mean = 5)`
- `trace_1 <- rnorm(100, mean = 0)`
- `trace_2 <- rnorm(100, mean = -5)`
- `x <- c(1:100)`
- `data <- data.frame(x, trace_0, trace_1, trace_2)`
- `plot_ly(data, x = ~x, y = ~trace_0, name = 'trace 0', type = 'scatter', mode = 'lines') %>%`
- `add_trace(y = ~trace_1, name = 'trace 1', mode = 'lines+markers') %>%`
- `add_trace(y = ~trace_2, name = 'trace 2', mode = 'markers')`

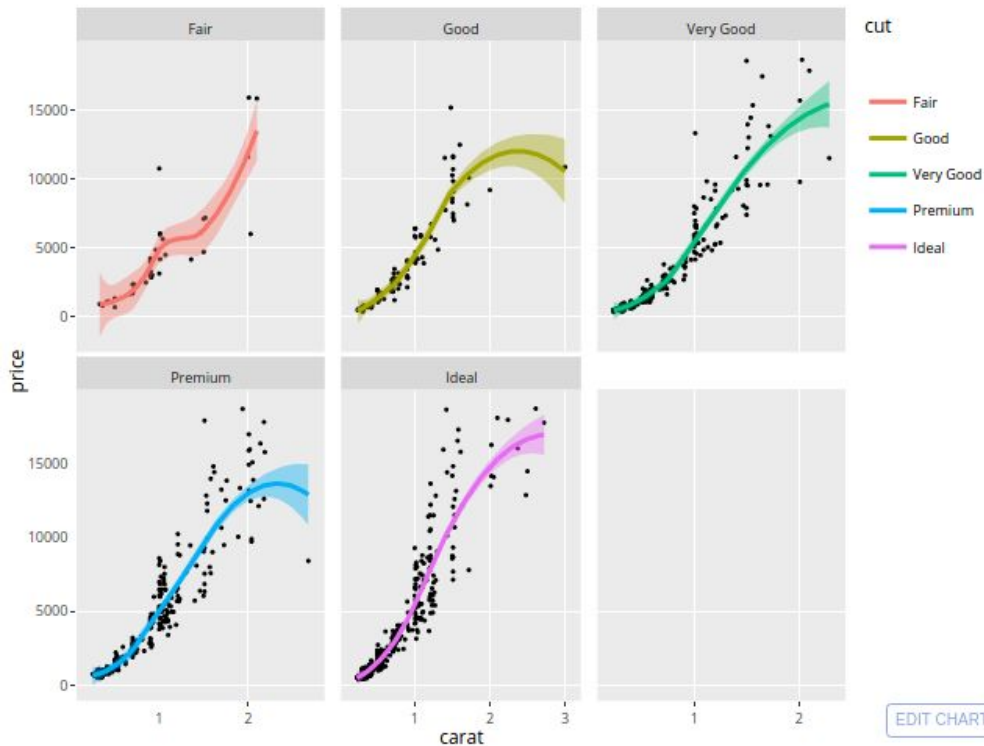
<https://plot.ly/r/>

<https://plot.ly/r/line-and-scatter/>

Plotly and ggplot

- `library(plotly)`
- `set.seed(100)`
- `d <- diamonds[sample(nrow(diamonds), 1000),]`
- `p <- ggplot(data = d, aes(x = carat, y = price)) +`
- `geom_point(aes(text = paste("Clarity:", clarity)), size = .5) +`
- `geom_smooth(aes(colour = cut, fill = cut)) +`
- `facet_wrap(~ cut)`
-
- `p <- ggplotly(p)`

<https://plot.ly/ggplot2/>



EDIT CHART

Highly compatible between R and Python

- It's a package available in both Python and R.
- You can access raw code of graph in multiple languages in plotly platform.



<https://plot.ly/~pspa/1586>

Good application examples

➤ Graph

- <https://plot.ly/r/>

➤ Shiny

- <https://plot.ly/r/shinyapp-map-click/>

➤ Deck/Report

- <http://andrewbtran.github.io/JRN-418/class3/plotly.html#/1>