Line and Scatter Plots in R

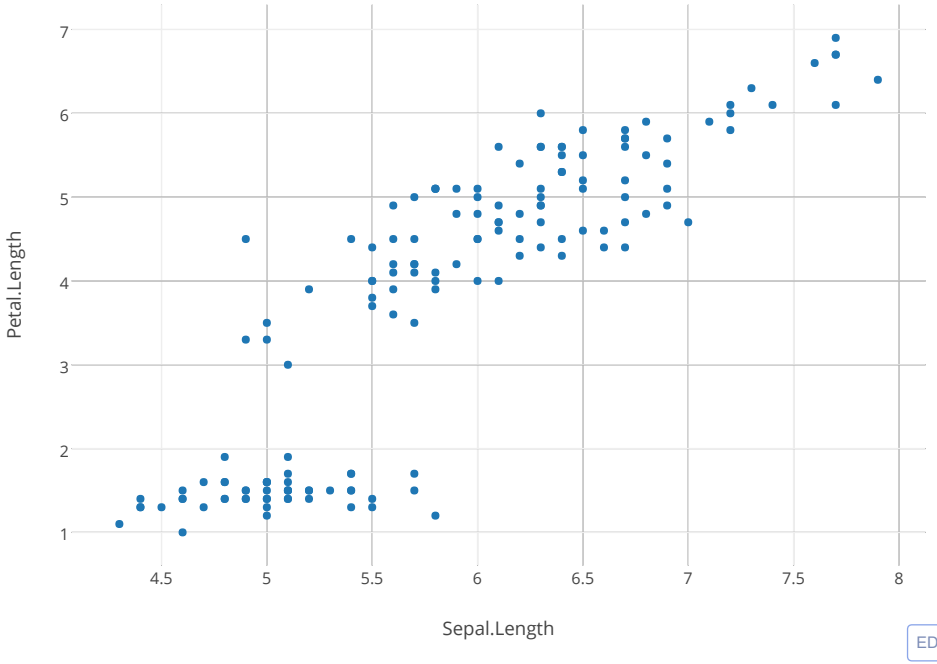
How to create line and scatter plots in R. Seven examples of basic and advanced scatter plots, time series line plots, colored charts, and density plots.

Scala | MATLAB | R | Python | Pandas | plotly.js | node.js | matplotlib

Line and Scatter Plots in R

Copy to clipboard!

```
# Simple scatterplot
library(plotly)
plot_ly(data = iris, x = Sepal.Length, y = Petal.Length, mode = "markers")
```

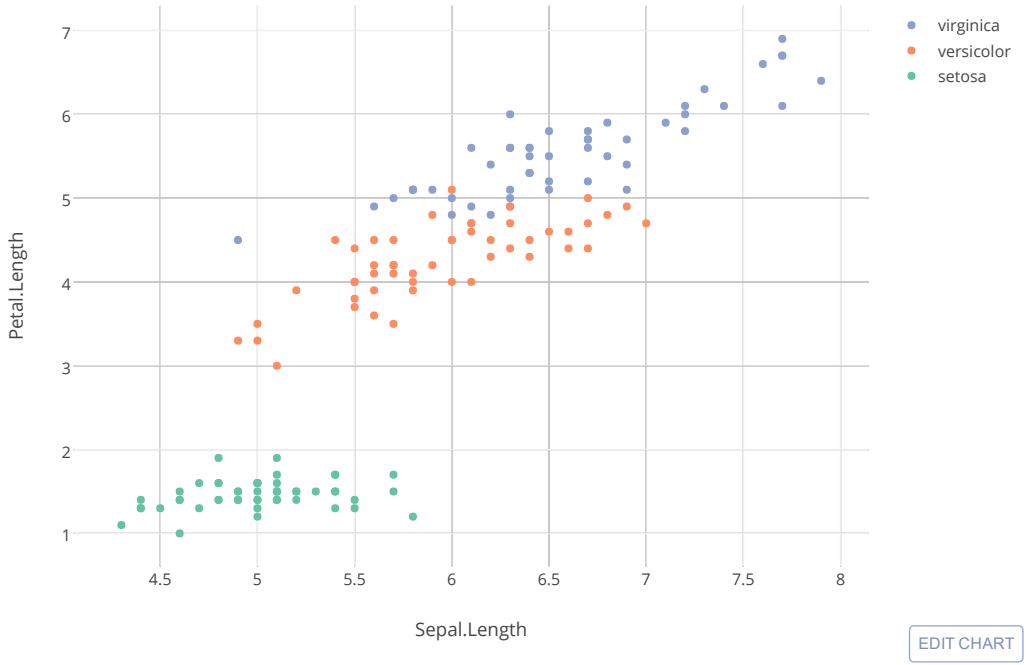


EDIT CHART

Scatter Plot with Qualitative Colorscale

Copy to clipboard!

```
plot_ly(data = iris, x = Sepal.Length, y = Petal.Length, mode = "markers",
        color = Species)
```

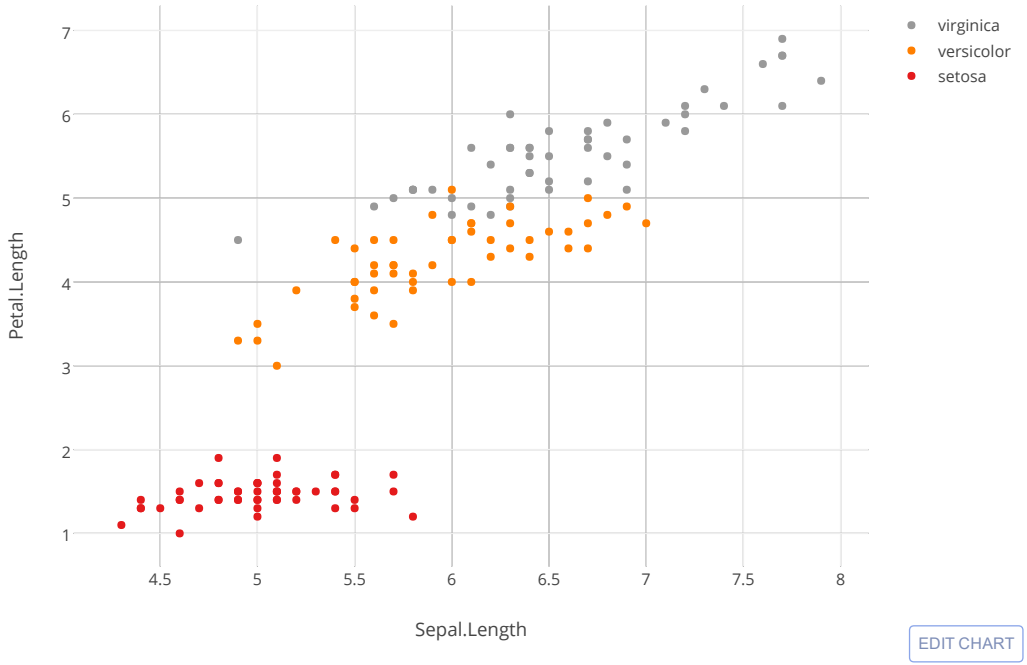


EDIT CHART

ColorBrewer Palette Names

Copy to clipboard!

```
# By default, colors will 'span the gamut'
# scales::show_col(RColorBrewer::brewer.pal("Set1"))
plot_ly(data = iris, x = Sepal.Length, y = Petal.Length, mode = "markers",
        color = Species, colors = "Set1")
```

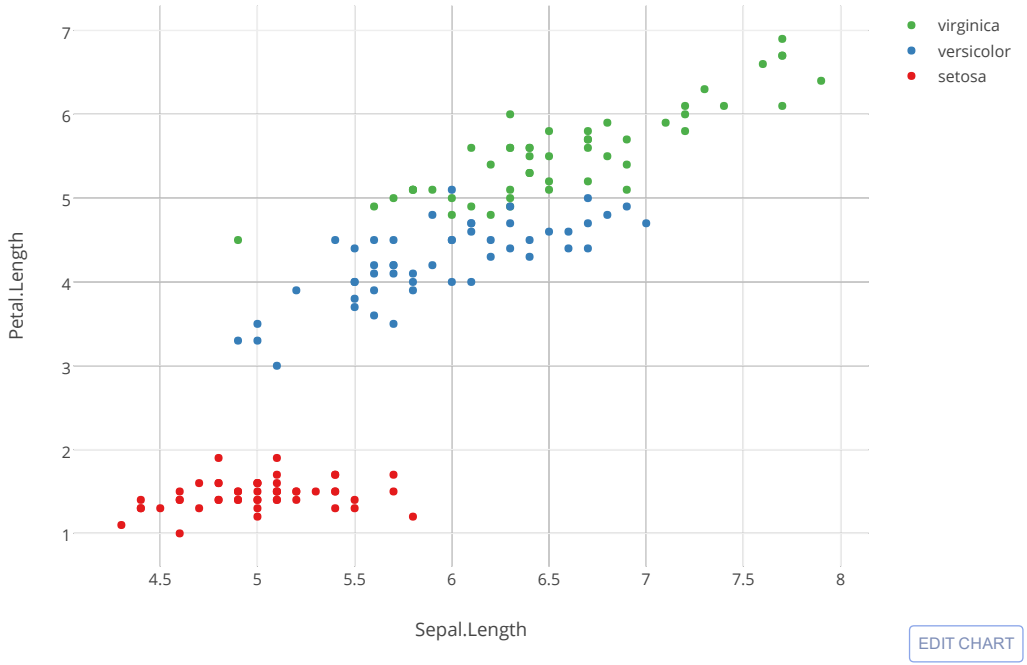


EDIT CHART

Custom Color Scales

Copy to clipboard!

```
# pass RGB or hex color codes directly to colors for finer control
pal <- RColorBrewer::brewer.pal(nlevels(iris$Species), "Set1")
plot_ly(data = iris, x = Sepal.Length, y = Petal.Length, color = Species,
        colors = pal, mode = "markers")
```

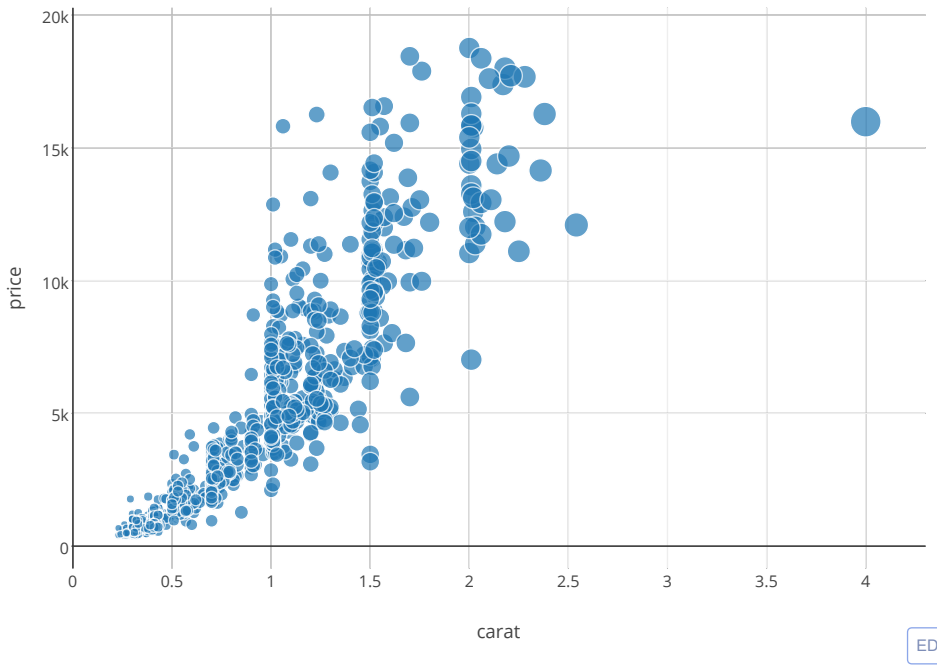


EDIT CHART

Adding Color and Size Mapping

Copy to clipboard!

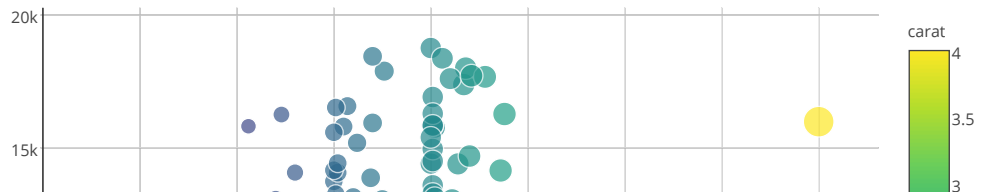
```
library(plotly)
d <- diamonds[sample(nrow(diamonds), 1000), ]
# note how size is automatically scaled and added as hover text
plot_ly(d, x = carat, y = price, size = carat, mode = "markers")
```

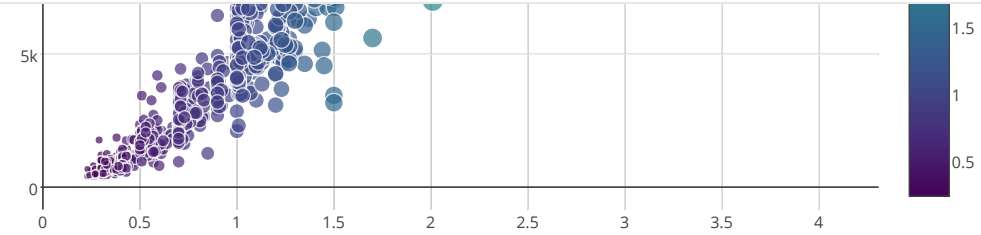


EDIT CHART

Copy to clipboard!

```
plot_ly(d, x = carat, y = price, text = paste("Clarity: ", clarity),
        mode = "markers", color = carat, size = carat, opacity = carat)
```



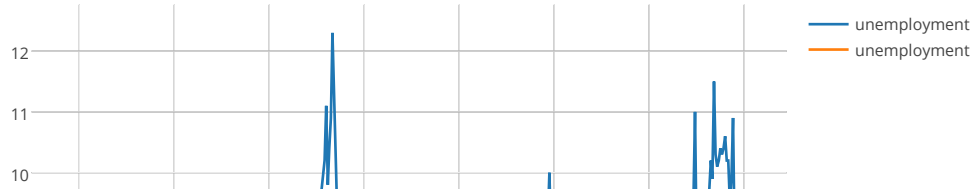


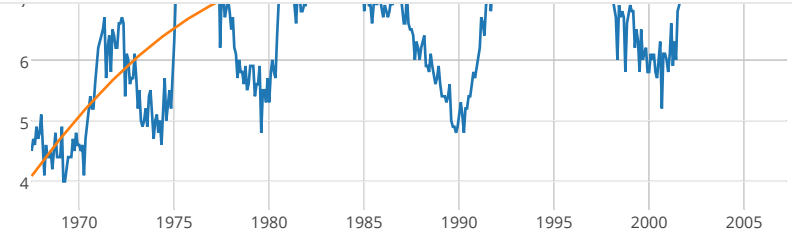
EDIT CHART

Basic Time Series Plot with Loess Smooth

Copy to clipboard!

```
p <- plot_ly(economics, x = date, y = uempmed, name = "unemployment")
p %>% add_trace(y = fitted(loess(uempmed ~ as.numeric(date))), x = date)
```



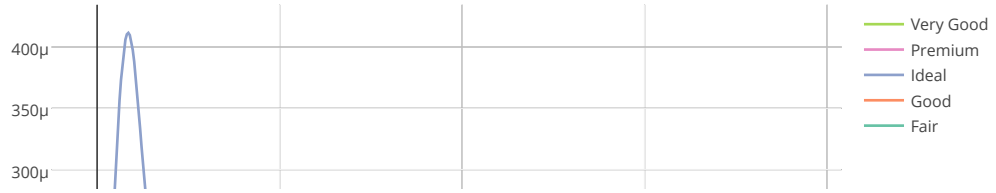


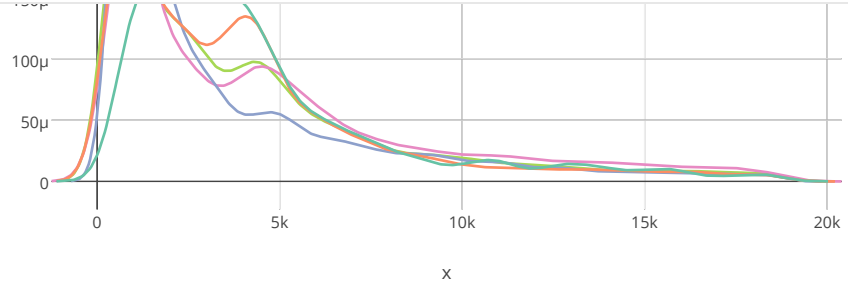
EDIT CHART

Density Plot

Copy to clipboard!

```
dens <- with(diamonds, tapply(price, INDEX = cut, density))
df <- data.frame(
  x = unlist(lapply(dens, "[[", "x")),
  y = unlist(lapply(dens, "[[", "y")),
  cut = rep(names(dens), each = length(dens[[1]]$x))
)
plot_ly(df, x = x, y = y, color = cut)
```



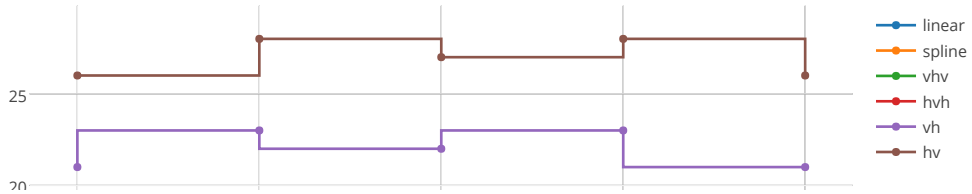


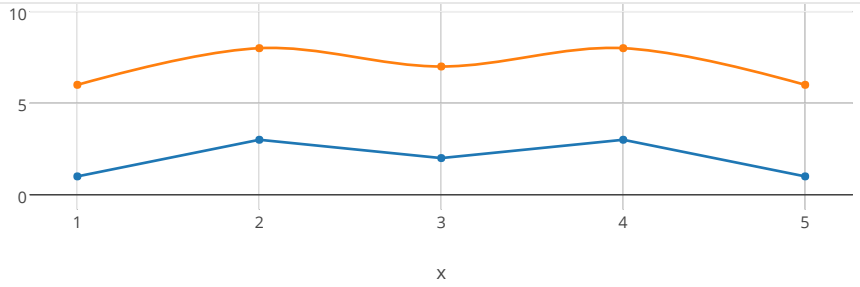
EDIT CHART

Line Interpolation Options

Copy to clipboard!

```
x <- 1:5
y <- c(1, 3, 2, 3, 1)
plot_ly(x = x, y = y, name = "linear", line = list(shape = "linear")) %>%
  add_trace(x = x, y = y + 5, name = "spline", line = list(shape = "spline")) %>%
  add_trace(x = x, y = y + 10, name = "vhv", line = list(shape = "vhv")) %>%
  add_trace(x = x, y = y + 15, name = "hvh", line = list(shape = "hvh")) %>%
  add_trace(x = x, y = y + 20, name = "vh", line = list(shape = "vh")) %>%
  add_trace(x = x, y = y + 25, name = "hv", line = list(shape = "hv"))
```





EDIT CHART

Still need help?

Contact Us

community.plot.ly | @plotlygraphs | github.com/plotly

For guaranteed 24 hour response turnarounds, upgrade to our Premium or Enterprise plans.