

Plotly for R is now entirely open source, free, and self-hosted

Learn more about why we've open sourced

Star 646

Upgrade to Plotly Pro Today!



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



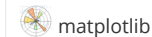
Pandas



plotly.js



node.js

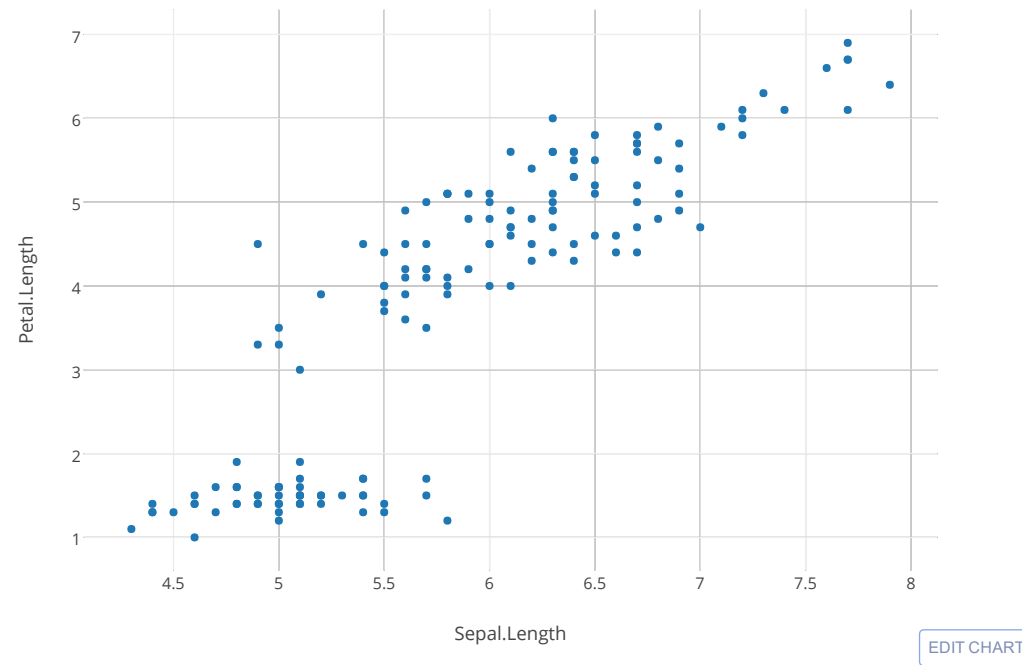


matplotlib

Line and Scatter Plots in R [🔗](#)

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

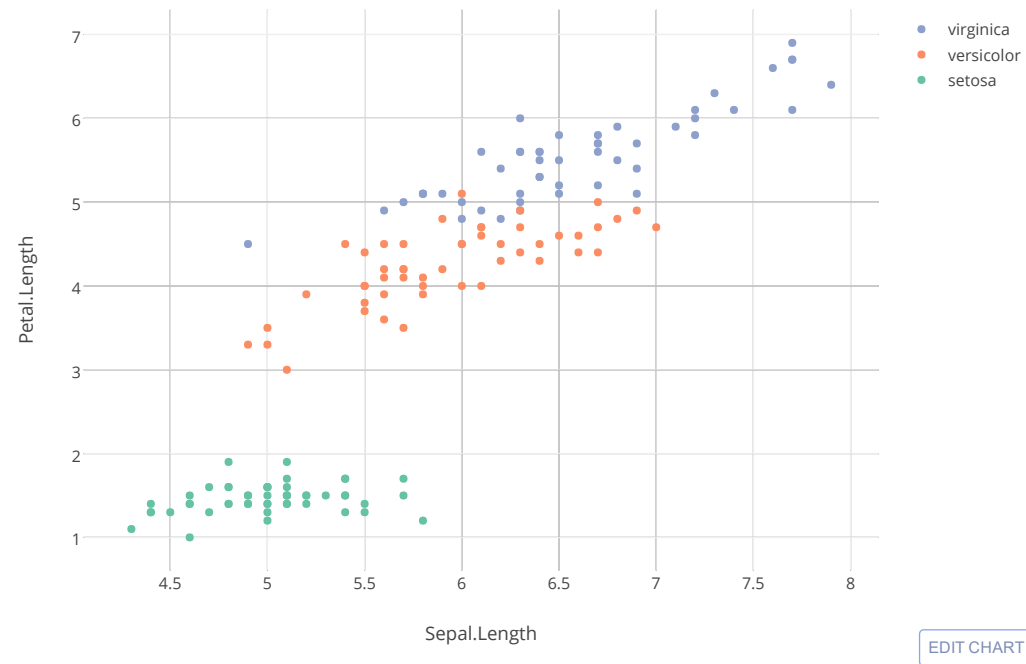
Copy to clipboard!



Scatter Plot with Qualitative Colorscale [↗](#)

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

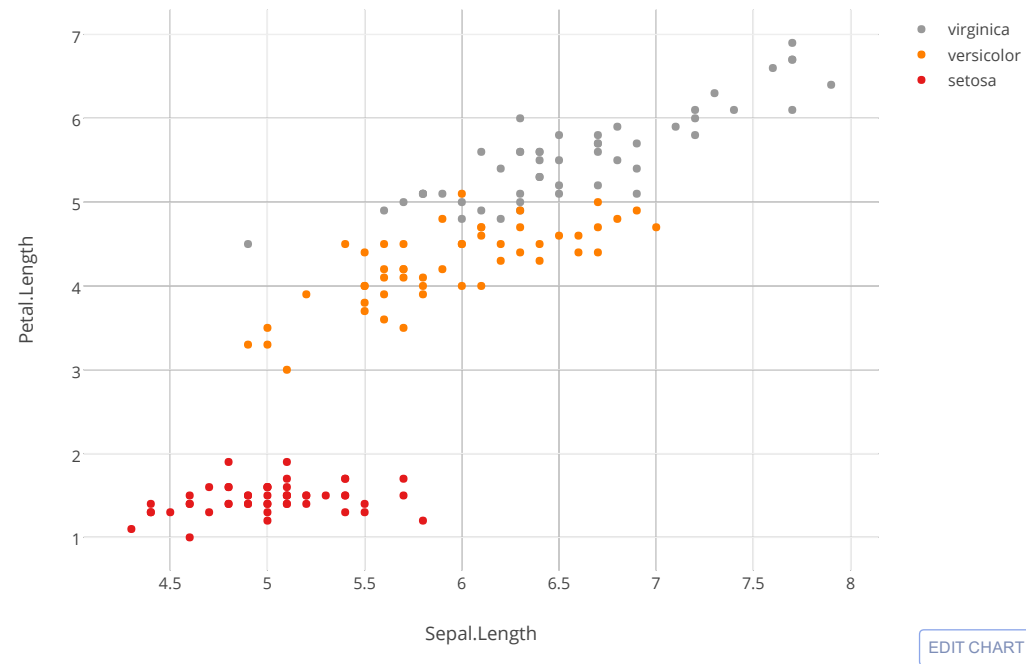
[Copy to clipboard!](#)



ColorBrewer Palette Names [↗](#)

```
# 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")
```

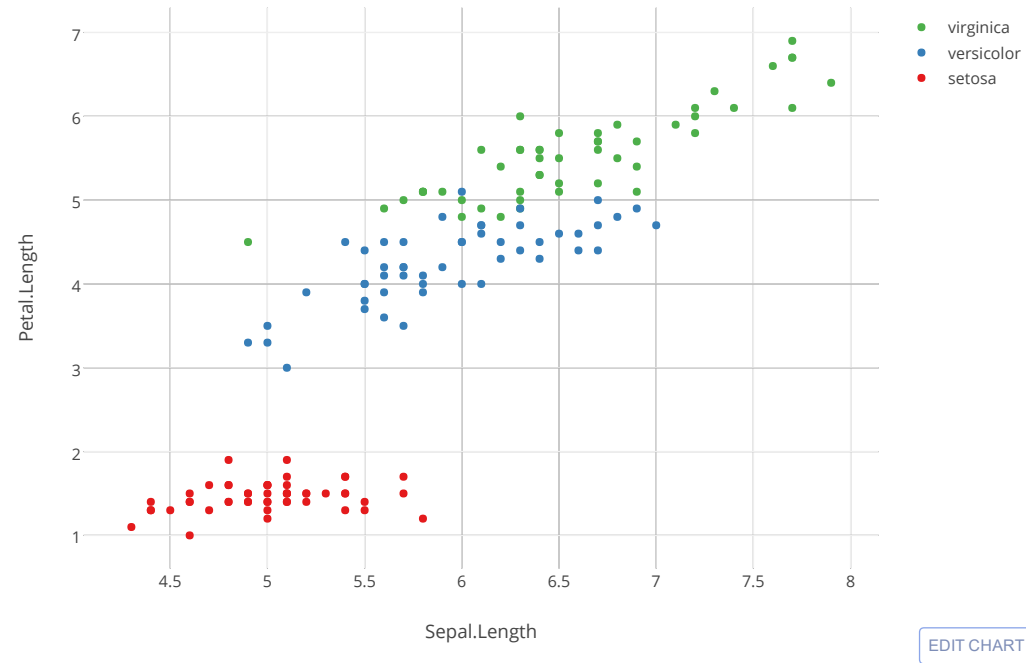
[Copy to clipboard!](#)



Custom Color Scales [↗](#)

```
# 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")
```

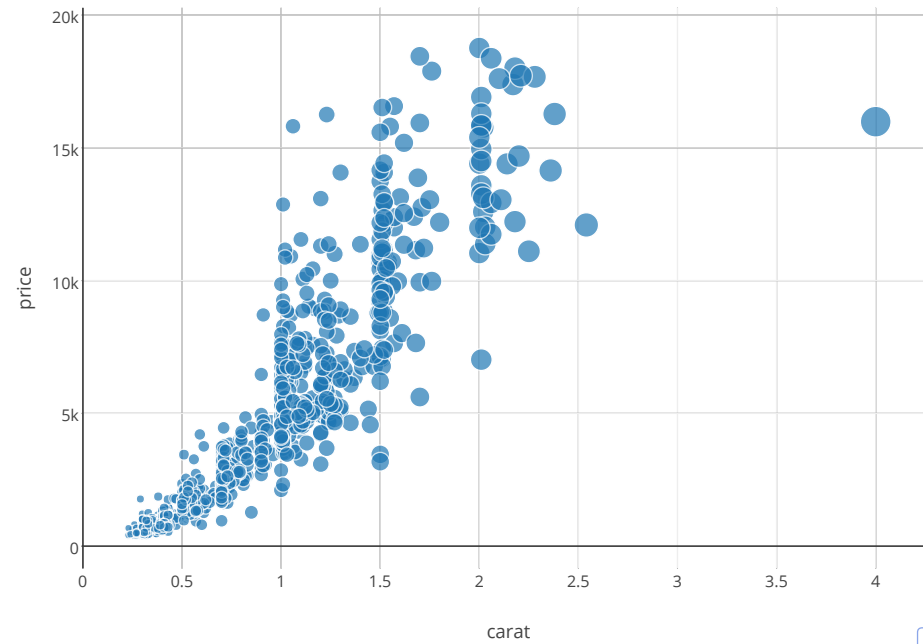
[Copy to clipboard!](#)



Adding Color and Size Mapping [↗](#)

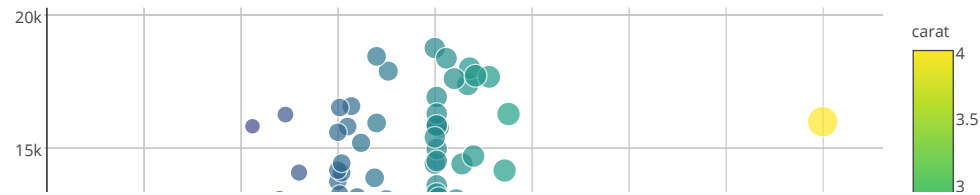
```
library(plotly)
d <- diamonds[sample(nrow(diamonds), 1000), ]
# note how size is automatically scaled and added as hover text
plot_ly(d, x = carat, y = price, size = carat, mode = "markers")
```

[Copy to clipboard!](#)

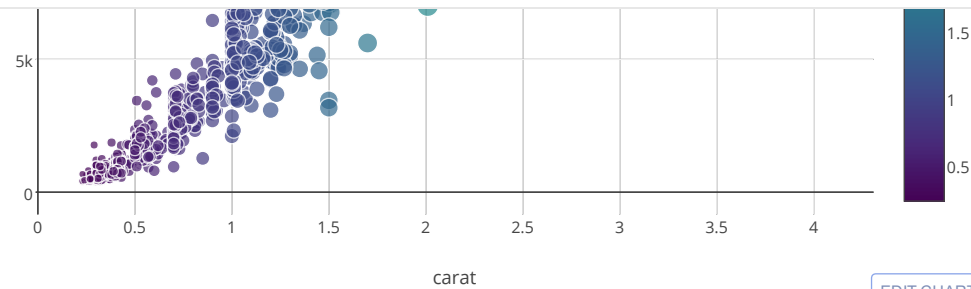
[EDIT CHART](#)

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

[Copy to clipboard!](#)

[Feed](#) [Pricing](#) [Make a Chart](#) [API](#)[? Help](#)[API Libraries](#)[R](#)[Line And Scatter Plots](#)[Fork on Github](#)

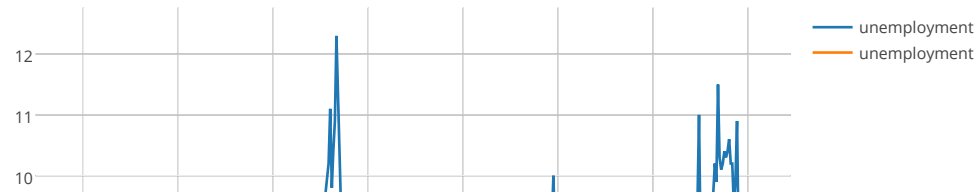
Navigation

[Line and Scatter Plots in R](#)[Scatter Plot with Qualitative
Colorscale](#)[ColorBrewer Palette Names](#)[Custom Color Scales](#)[Adding Color and Size
Mapping](#)[Basic Time Series Plot with
Loess Smooth](#)[Density Plot](#)[Line Interpolation Options](#)[← Back To R](#)

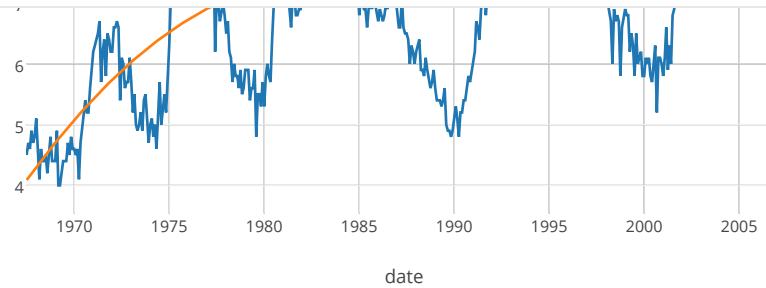
Basic Time Series Plot with Loess Smooth [↗](#)

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

[Copy to clipboard!](#)


[Feed](#) [Pricing](#) [Make a Chart](#) [API](#)
[? Help](#)
[API Libraries](#)
[R](#)
[Line And Scatter Plots](#)
[Fork on Github](#)

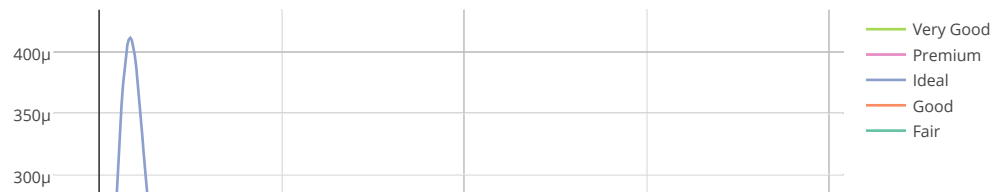
Navigation

[Line and Scatter Plots in R](#)
[Scatter Plot with Qualitative
Colorscale](#)
[ColorBrewer Palette Names](#)
[Custom Color Scales](#)
[Adding Color and Size
Mapping](#)
[Basic Time Series Plot with
Loess Smooth](#)
[Density Plot](#)
[Line Interpolation Options](#)
[← Back To R](#)

[EDIT CHART](#)

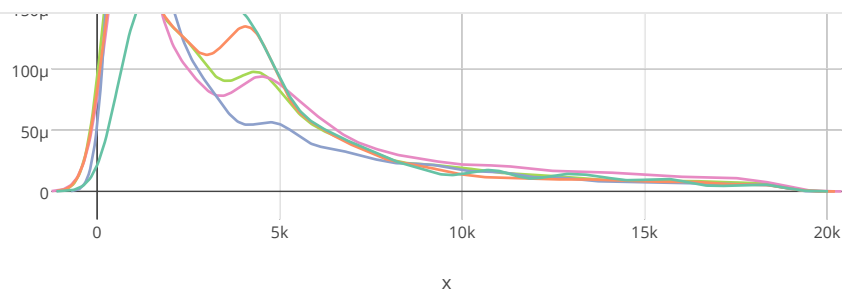
Density Plot [↗](#)

```
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)
```

[Copy to clipboard!](#)


[Feed](#) [Pricing](#) [Make a Chart](#) [API](#)
[? Help](#)
[API Libraries](#)
[R](#)
[Line And Scatter Plots](#)
[Fork on Github](#)

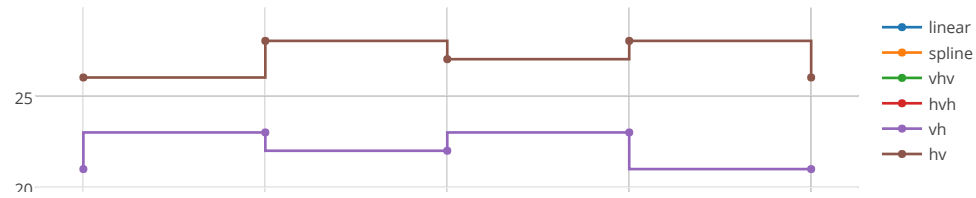
Navigation

[Line and Scatter Plots in R](#)
[Scatter Plot with Qualitative Colorscale](#)
[ColorBrewer Palette Names](#)
[Custom Color Scales](#)
[Adding Color and Size Mapping](#)
[Basic Time Series Plot with Loess Smooth](#)
[Density Plot](#)
[Line Interpolation Options](#)
[← Back To R](#)

[EDIT CHART](#)

Line Interpolation Options [↗](#)

```
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"))
```

[Copy to clipboard!](#)



Feed

Pricing

Make a Chart

API

Help

API Libraries

R

Line And Scatter Plots

Fork on Github

Navigation

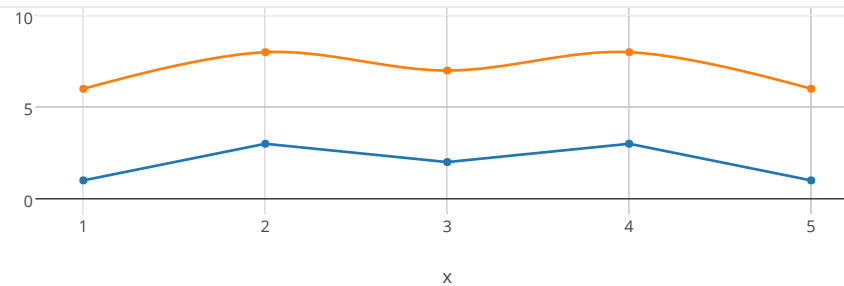
Line and Scatter Plots in R

Scatter Plot with Qualitative
Colorscale

ColorBrewer Palette Names

Custom Color Scales

Adding Color and Size



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.

API

About Us

Help

Documentation
API Libraries
REST APIs
Plotly.js
Hardware

Solutions

Plans & Pricing
Enterprise

Team
Careers
Plotly Blog
Modern Data

Connect


 Twitter  Facebook  Github  LinkedIn  Google+

Knowledge Base
Benchmarks
Workshop



Feed Pricing Make a Chart API

[? Help](#) | [API Libraries](#) | [R](#) | **Line And Scatter Plots**

 Fork on Github

Navigation

Line and Scatter Plots in R

Scatter Plot with Qualitative
Colorscale

ColorBrewer Palette Names

Custom Color Scales

Adding Color and Size
Mapping

Basic Time Series Plot with
Loess Smooth

Density Plot

Line Interpolation Options

 Back To R