

How can I plot multiple columns in R with examples?

Authored by
stats writer

April 20, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I plot multiple columns in R with examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137405>

R is a powerful statistical programming language that allows users to visualize and analyze data in various ways. One common task in data analysis is to plot multiple columns of data on a single graph for comparison and analysis. This can be easily achieved in R using various built-in functions and packages.

To plot multiple columns in R, the first step is to import the data into the R environment. This can be done using the "read.csv()" or "read.table()" functions, depending on the file format of the data. Once the data is imported, the "plot()" function can be used to create a basic graph with a single column. To add more columns to the graph, the "lines()" or "points()" function can be used. The "legend()" function can also be used to add a legend to the graph to differentiate between the different columns.

For example, if we have a dataset with three columns - "Year", "Sales", and "Expenses", we can plot the sales and expenses data for each year on a line graph using the following code:

```
sales <- c(100, 150, 200, 250, 300) #sample sales data
expenses <- c(80, 100, 120, 140, 160) #sample expenses data
year <- c(2015, 2016, 2017, 2018, 2019) #sample year data

plot(year, sales, type = "l", xlab = "Year", ylab = "Sales", main = "Sales vs Expenses") #plotting
sales data
lines(year, expenses, col = "red") #adding expenses data to the graph
legend("topright", legend = c("Sales", "Expenses"), col = c("black", "red"), lty = c(1, 1), cex = 0.8)
#adding a legend
```

This will result in a line graph with two lines representing sales and expenses data for each year.

In conclusion, plotting multiple columns in R is a simple and efficient way to compare and analyze data. By using the appropriate functions and packages, users can customize their graphs to effectively visualize their data and make meaningful interpretations.

Plot Multiple Columns in R (With Examples)

Often you may want to plot multiple columns from a data frame in R. Fortunately this is easy to do using the visualization library ggplot2.

This tutorial shows how to use ggplot2 to plot multiple columns of a data frame on the same graph and on different graphs.

Example 1: Plot Multiple Columns on the Same Graph

The following code shows how to generate a data frame, then "melt" the data frame into a long format, then use ggplot2 to create a line plot for each column in the data frame:

```
#load necessary libraries
```

```
library(ggplot2)
```

```
library(reshape2)
```

```
#create data frame
```

```
df <- data.frame(index=c(1, 2, 3, 4, 5, 6),
```

```
var1=c(4, 4, 5, 4, 3, 2),
```

```
var2=c(1, 2, 4, 4, 6, 9),
```

```
var3=c(9, 9, 9, 5, 5, 3))
```

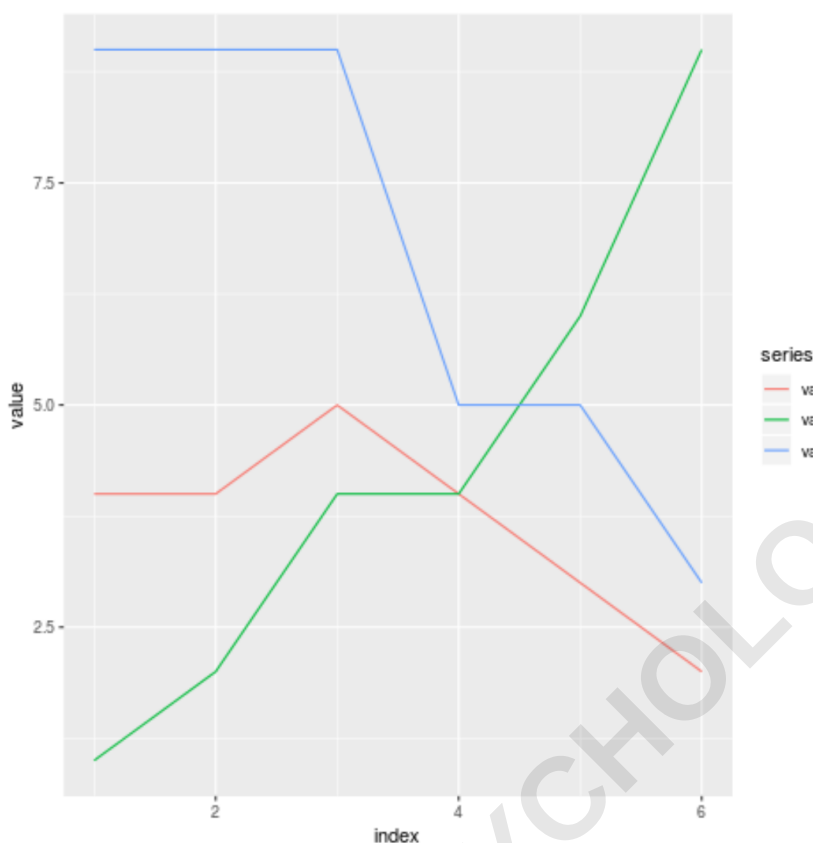
```
#melt data frame into long format
```

```
df <- melt(df , id.vars = 'index', variable.name = 'series')
```

```
#create line plot for each column in data frame
```

```
ggplot(df, aes(index, value)) +
```

`geom_line(aes(colour = series))`



Example 2: Plot Multiple Columns on Different Graphs

The following code shows how to generate a data frame, then "melt" the data frame into a long format, then use ggplot2 to create a line plot for each column in the data frame, splitting up each line into its own plot:

```
#load necessary libraries
```

```
library(ggplot2)
```

```
library(reshape2)
```

#create data frame

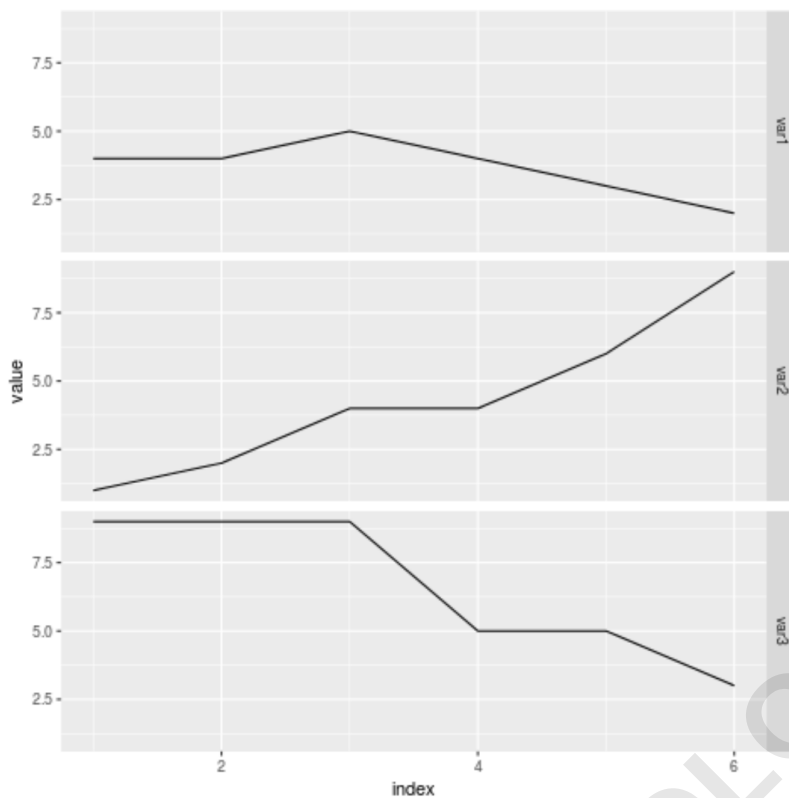
```
df <- data.frame(index=c(1, 2, 3, 4, 5, 6),  
var1=c(4, 4, 5, 4, 3, 2),  
var2=c(1, 2, 4, 4, 6, 9),  
var3=c(9, 9, 9, 5, 5, 3))
```

#melt data frame into long format

```
df <- melt(df , id.vars = 'index', variable.name = 'series')
```

#create line plot for each column in data frame

```
ggplot(df, aes(index, value)) +  
geom_line() +  
facet_grid(series ~ .)
```



How to Create Side-by-Side Plots in ggplot2

How to Create a Grouped Boxplot in R Using ggplot2