

How can I plot multiple plots on the same graph in R?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I plot multiple plots on the same graph in R?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=157282>

To plot multiple plots on the same graph in R, one can use the "par()" function to specify the layout of the graph, such as the number of rows and columns. Then, each individual plot can be created using the "plot()" function and added to the graph using the "lines()" or "points()" function. Alternatively, the "ggplot2" package can be used to create a single graph with multiple plots by specifying different data sets and variables for each plot within the same "ggplot()" function. This allows for easy comparison and analysis of multiple plots within one graph.

Plot Multiple Plots on Same Graph in R (3 Examples)

You can use the following methods to plot multiple plots on the same graph in R:

Method 1: Plot Multiple Lines on Same Graph

#plot first line

```
plot(x, y1, type='l')
```

#add second line to plot

```
lines(x, y2)
```

Method 2: Create Multiple Plots Side-by-Side

#define plotting area as one row and two columns

```
par(mfrow = c(1, 2))
```

#create first plot

```
plot(x, y1, type='l')
```

```
#create second plot  
plot(x, y2, type='l')
```

Method 3: Create Multiple Plots Stacked Vertically

```
#define plotting area as two rows and one column  
par(mfrow = c(2, 1))#create first plot  
plot(x, y1, type='l')
```

```
#create second plot  
plot(x, y2, type='l')
```

The following examples show how to use each method in practice.

Example 1: Plot Multiple Lines on Same Graph

The following code shows how to plot two lines on the same graph in R:

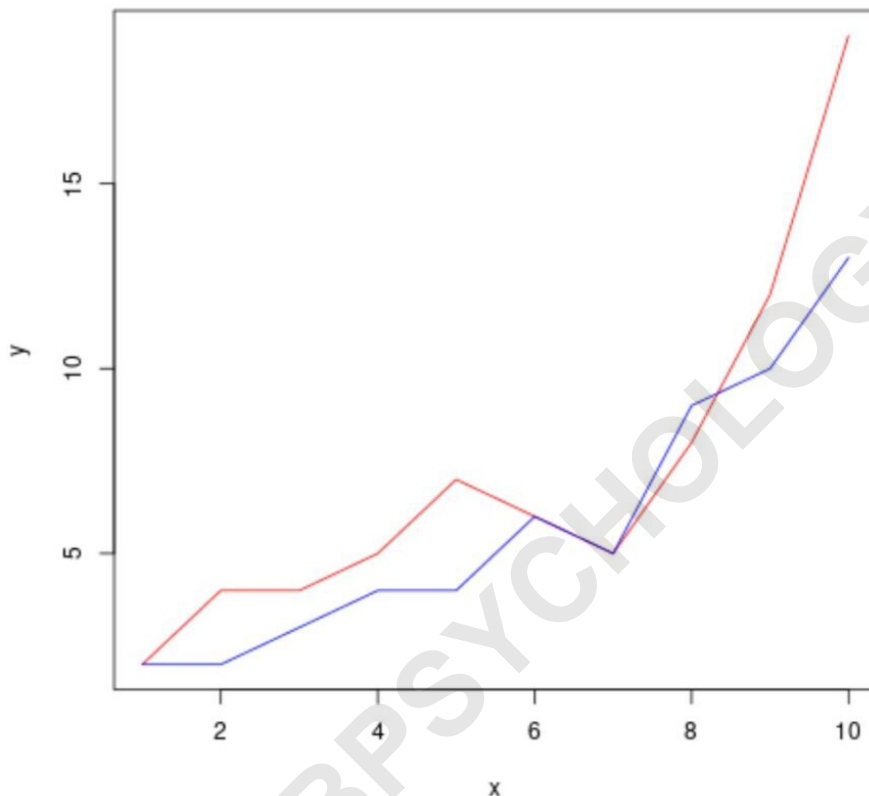
```
#define data to plot  
x <- 1:10  
y1 <- c(2, 4, 4, 5, 7, 6, 5, 8, 12, 19)  
y2 <- c(2, 2, 3, 4, 4, 6, 5, 9, 10, 13)
```

```
#plot first line
```

```
plot(x, y1, type='l', col='red', xlab='x', ylab='y')
```

```
#add second line to plot
```

```
lines(x, y2, col='blue')
```



Example 2: Create Multiple Plots Side-by-Side

The following code shows how to use the `par()` argument to plot multiple plots side-by-side:

```
#define data to plot
```

```
x <- 1:10
```

```
y1 <- c(2, 4, 4, 5, 7, 6, 5, 8, 12, 19)
```

```
y2 <- c(2, 2, 3, 4, 4, 6, 5, 9, 10, 13)
```

```
#define plotting area as one row and two columns
```

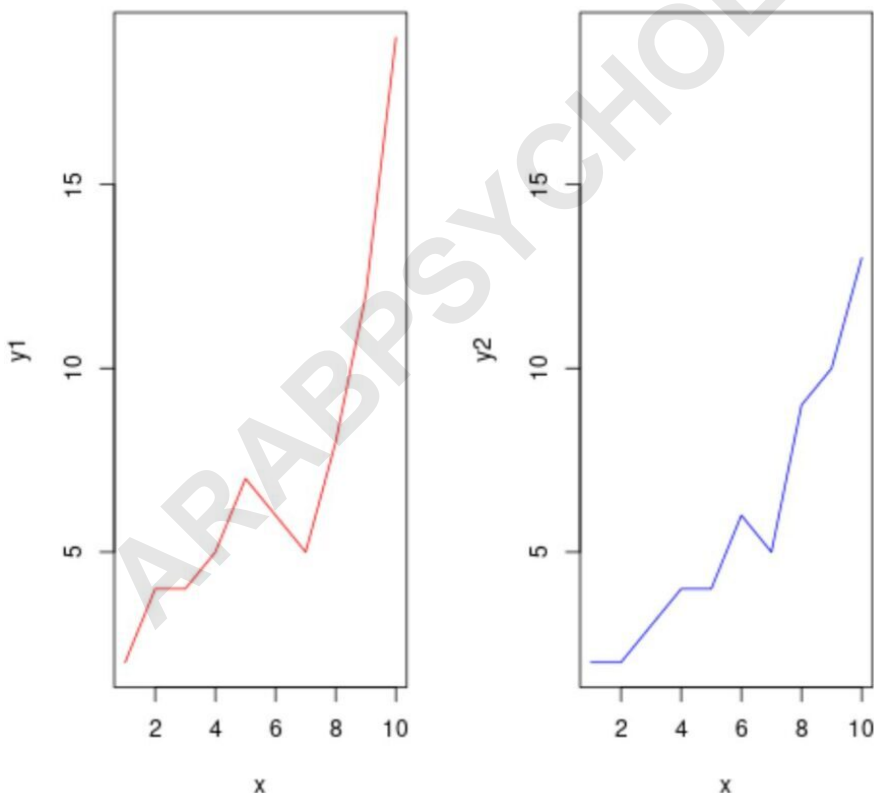
```
par(mfrow = c(1, 2))
```

```
#create first line plot
```

```
plot(x, y1, type='l', col='red')
```

```
#create second line plot
```

```
plot(x, y2, type='l', col='blue', ylim=c(min(y1), max(y1)))
```



Note that we used the argument in the second plot to

ensure that the two plots had the same y-axis limits.

Example 3: Create Multiple Plots Stacked Vertically

The following code shows how to use the `par()` argument to plot multiple plots stacked vertically:

```
#define data to plot
```

```
x <- 1:10
```

```
y1 <- c(2, 4, 4, 5, 7, 6, 5, 8, 12, 19)
```

```
y2 <- c(2, 2, 3, 4, 4, 6, 5, 9, 10, 13)
```

```
#define plotting area as two rows and one column
```

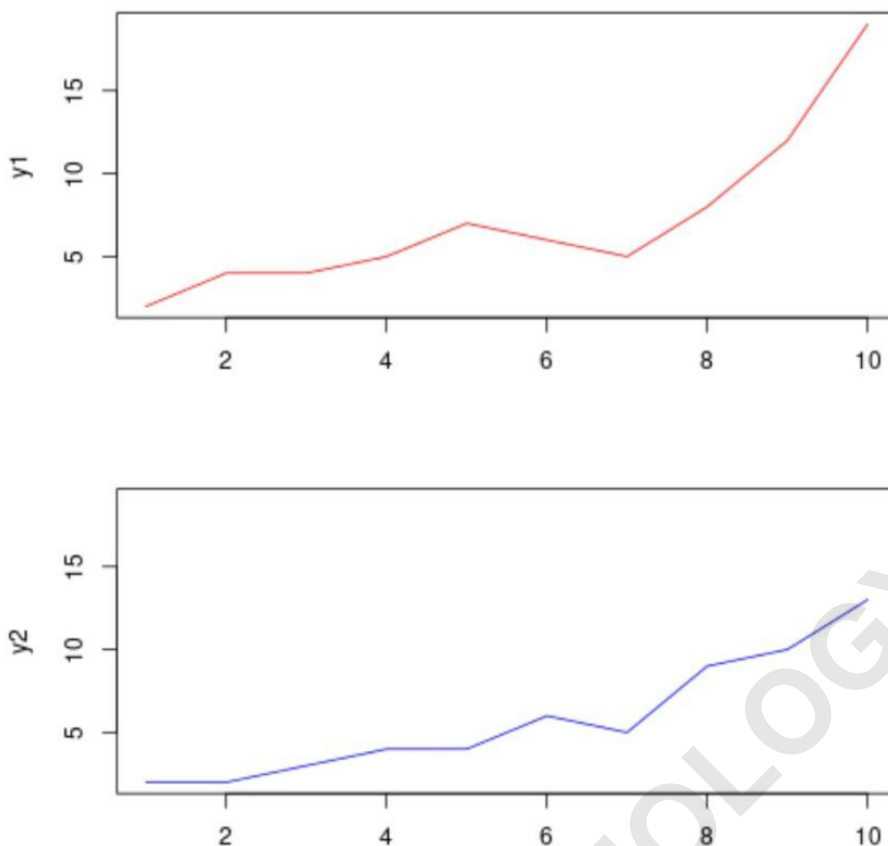
```
par(mfrow = c(2, 1), mar = c(2, 4, 4, 2))
```

```
#create first line plot
```

```
plot(x, y1, type='l', col='red')
```

```
#create second line plot
```

```
plot(x, y2, type='l', col='blue', ylim=c(min(y1), max(y1)))
```



Note that we used the mar argument to specify the (bottom, left, top, right) margins for the plotting area.

Note: The default is $\text{mar} = \text{c}(5.1, 4.1, 4.1, 2.1)$

Additional Resources

The following tutorials explain how to perform other common tasks in R: