

How can I use the par() function in R?

Authored by
stats writer

May 3, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the par() function in R?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=142234>

The `par()` function in R is a powerful tool that allows users to customize and control various graphical parameters in their plots. With `par()`, users can adjust aspects such as plot size, color palette, grid lines, and axis labels. This function is particularly useful for creating visually appealing and informative graphs, as well as for making comparisons between multiple plots. By using the `par()` function, users can easily modify the appearance of their plots to suit their specific needs and preferences. This function is essential for anyone looking to create high-quality and customizable plots in R.

Use the par() Function in R

You can use the `par()` function in R to create multiple plots at once.

This function uses the following basic syntax:

#define plot area as four rows and two columns

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

#create plots

```
plot(1:5)
```

```
plot(1:20)
```

```
...
```

The following examples show how to use this function in practice.

Example 1: Display Multiple Plots with `par()`

The following code shows how to use the `par()` function

to define a plotting area with 3 rows and 1 column:

#define plot area as three rows and one column

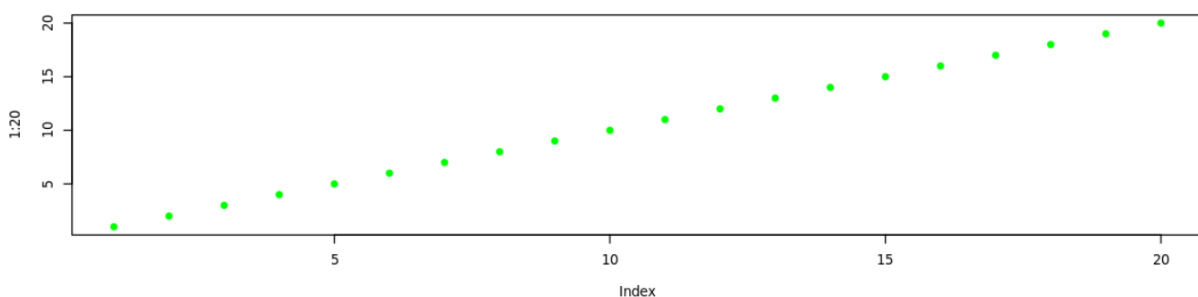
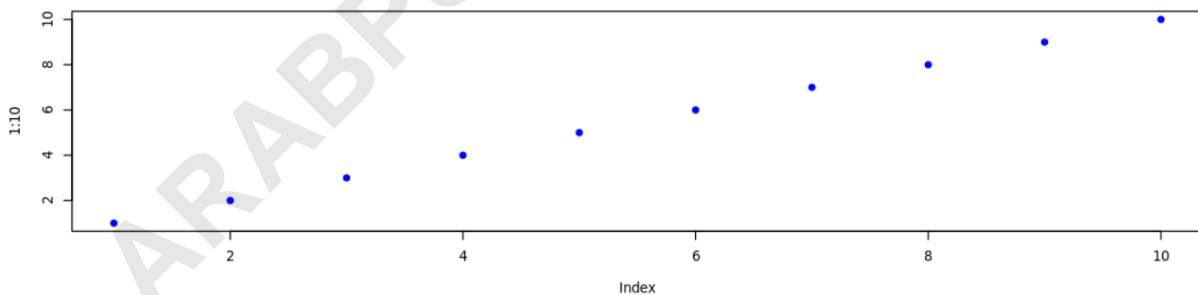
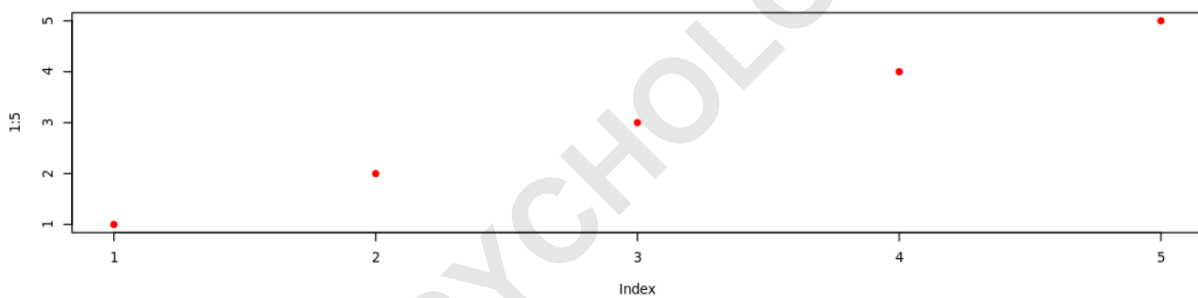
```
par(mfrow = c(3, 1))
```

#create plots

```
plot(1:5, pch=19, col='red')
```

```
plot(1:10, pch=19, col='blue')
```

```
plot(1:20, pch=19, col='green')
```



Example 2: Specify Margins of Plots with mar()

The following code shows how to use the mar() argument to specify the margins around each plot in the following order: bottom, left, top, right.

Note: The default is mar = c(5.1, 4.1, 4.1, 2.1)

#define plot area with tiny bottom margin and huge right margin

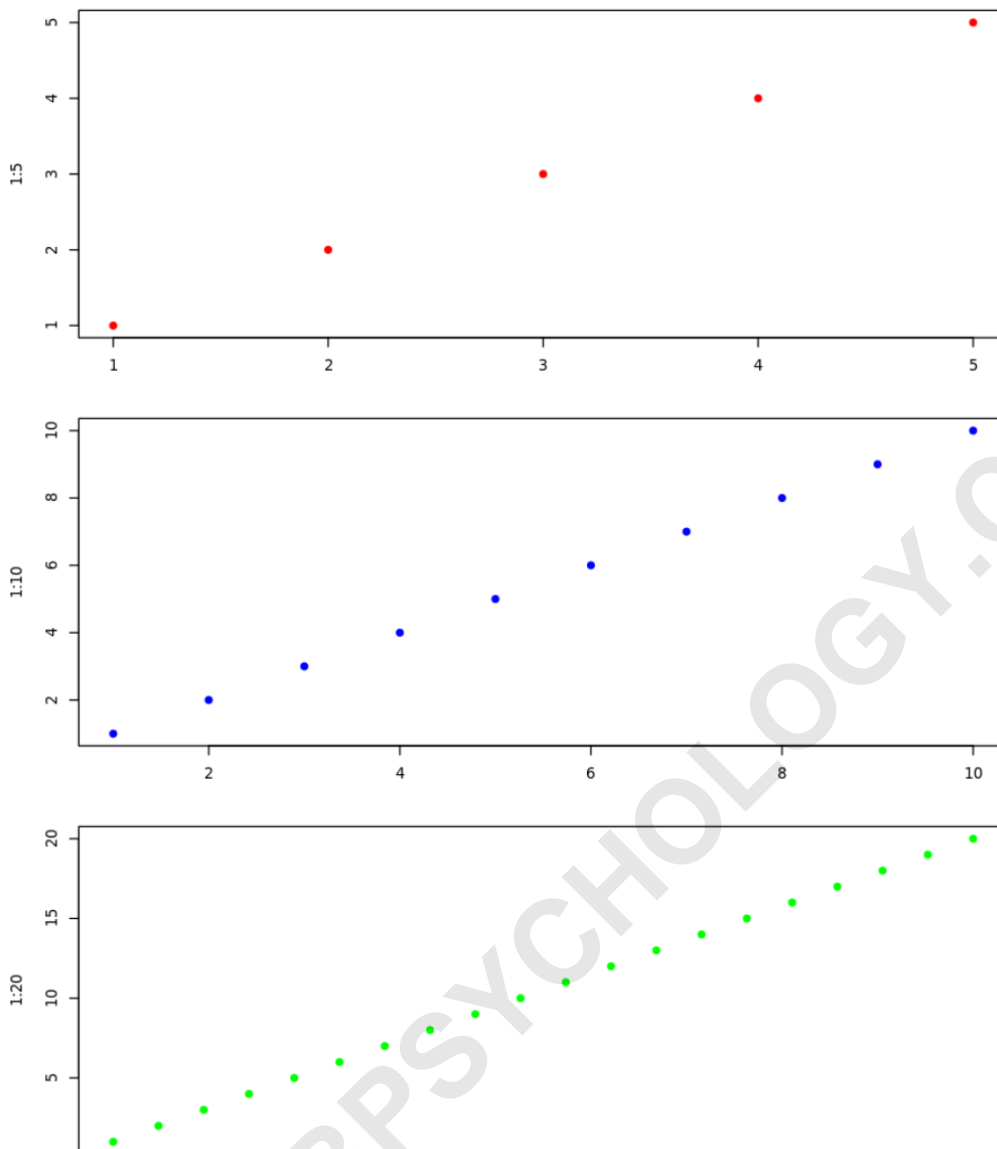
```
par(mfrow = c(3, 1), mar = c(0.5, 4, 4, 20))
```

#create plots

```
plot(1:5, pch=19, col='red')
```

```
plot(1:10, pch=19, col='blue')
```

```
plot(1:20, pch=19, col='green')
```



Notice how the plots look less wide because we made the margin on the right so large.

Example 3: Specify Text Size of Plots with `cex()`

The following code shows how to use the `cex.lab()` and `cex.axis()` arguments to specify the size of the axis labels and the tick labels, respectively.

Note: The default is `cex.lab = 1` and `cex.axis = 1`

#define plot area with large axis labels

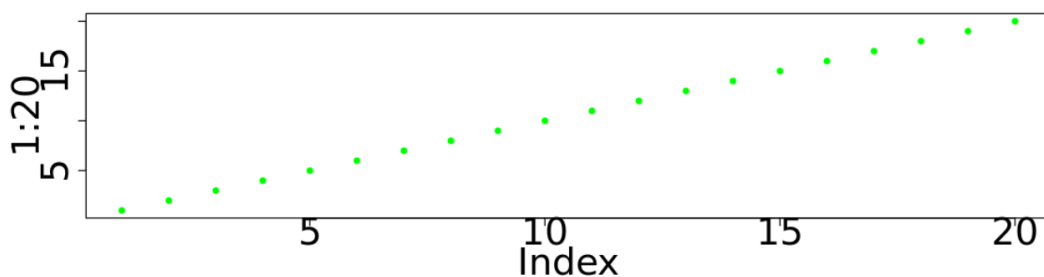
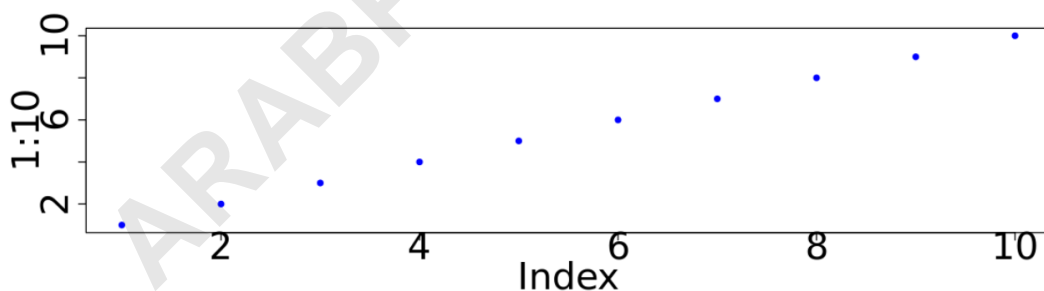
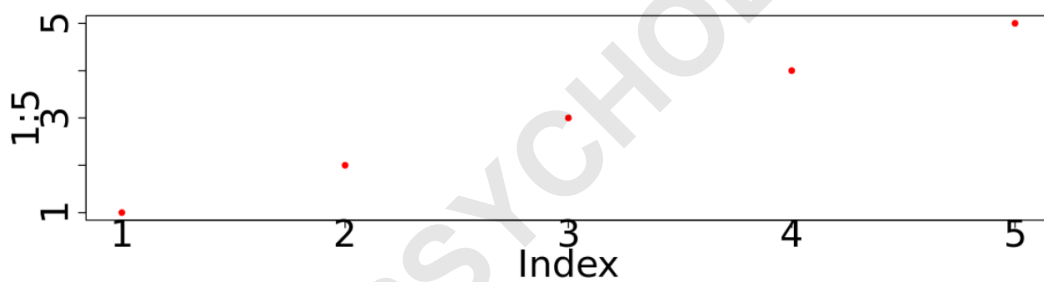
```
par(mfrow = c(3, 1), mar = c(5, 10, 4, 1), cex.axis = 3,  
cex.lab = 3)
```

#create plots

```
plot(1:5, pch=19, col='red')
```

```
plot(1:10, pch=19, col='blue')
```

```
plot(1:20, pch=19, col='green')
```



Once you're finished using the par() function, you can use the dev.off() function to reset the par options.

```
#reset par() options  
dev.off()
```

It's a good to use dev.off() each time you're done using the par() function.

ARABPSYCHOLOGY.COM