

R Multiple Plot Using `par()` Function



R Multiple Plot Using par() Function

In this article, you will learn to use par() function to put multiple graphs in a single plot by passing graphical parameters mfrow and mfcoll.

Sometimes we need to put two or more graphs in a single plot.

R par() function

We can put multiple graphs in a single plot by setting some graphical parameters with the help of par() function. R programming has a lot of graphical parameters which control the way our graphs are displayed.

The par() function helps us in setting or inquiring about these parameters. For example, you can look at all the parameters and their value by calling the function without any argument.

```
>par()
$xlog
[1] FALSE
...
$yaxt
[1] "s"
$ylbias
[1] 0.2
```

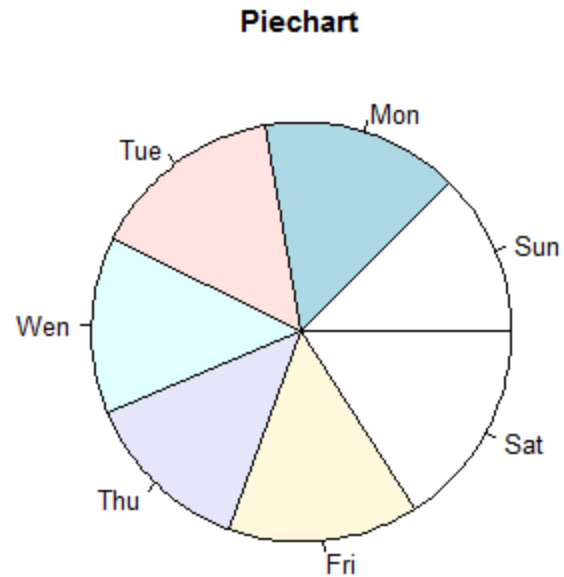
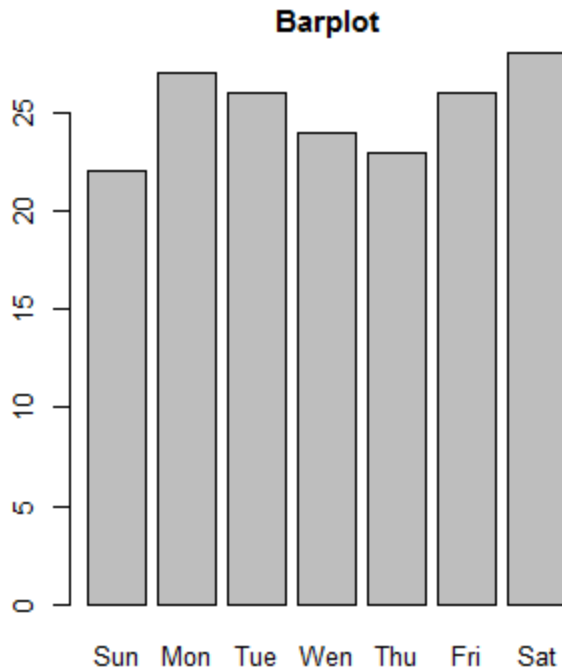
You will see a long list of parameters and to know what each does you can check the help section ?par. Here we will focus on those which help us in creating subplots.

Graphical parameter mfrow can be used to specify the number of subplot we need.

It takes in a vector of form c(m, n) which divides the given plot into m*n array of subplots. For example, if we need to plot two graphs side by side, we would have m=1 and n=2. Following example illustrates this.

R Multiple Plot Using par() Function

```
>max.temp # a vector used for plotting
Sun Mon Tue Wen Thu Fri Sat
22 27 26 24 23 26 28
par(mfrow=c(1,2)) # set the plotting area into a 1*2 array
barplot(max.temp, main="Barplot")
pie(max.temp, main="Piechart", radius=1)
```

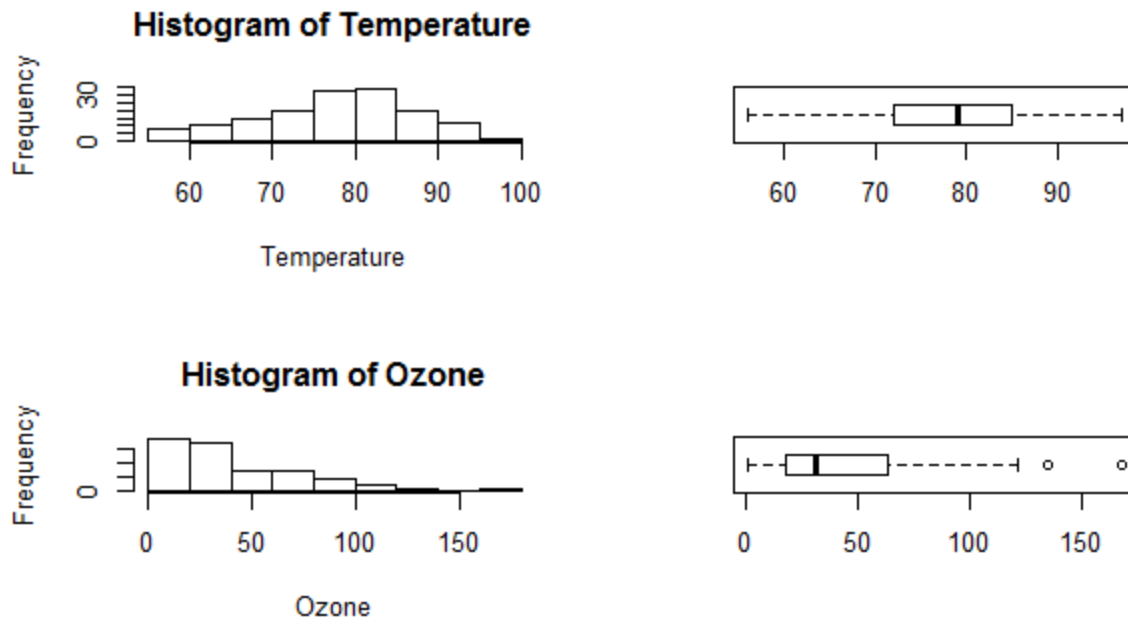


This same phenomenon can be achieved with the graphical parameter **mfcol**.

The only difference between the two is that, **mfrow** fills in the subplot region row wise while **mfcol** fills it column wise.

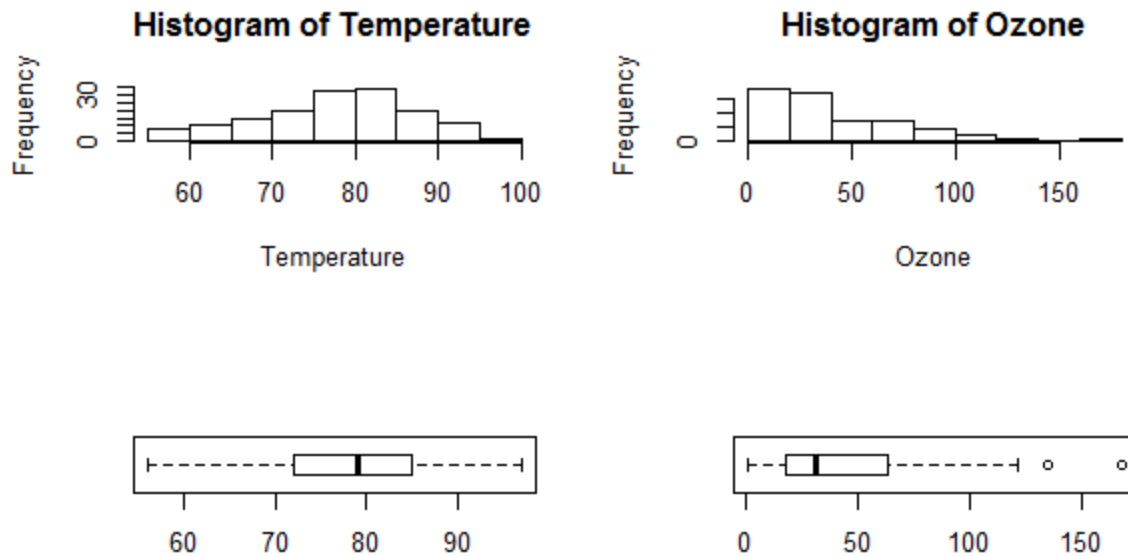
```
Temperature <- airquality$Temp
Ozone <- airquality$Ozone
par(mfrow=c(2,2))
hist(Temperature)
boxplot(Temperature, horizontal=TRUE)
hist(Ozone)
boxplot(Ozone, horizontal=TRUE)
```

R Multiple Plot Using par() Function



Same plot with the change `par(mfcol = c(2, 2))` would look as follows. Note that only the ordering of the subplot is different.

R Multiple Plot Using par() Function



More Precise Control

The graphical parameter `fig` lets us control the location of a figure precisely in a plot.

We need to provide the coordinates in a normalized form as `c(x1, x2, y1, y2)`. For example, the whole plot area would be `c(0, 1, 0, 1)` with `(x1, y1) = (0, 0)` being the lower-left corner and `(x2, y2) = (1, 1)` being the upper-right corner.

Note: we have used parameters `cex` to decrease the size of labels and `mai` to define margins.

R Multiple Plot Using par() Function

```
# make labels and margins smaller
par(cex=0.7, mai=c(0.1,0.1,0.2,0.1))
Temperature <- airquality$Temp
# define area for the histogram
par(fig=c(0.1,0.7,0.3,0.9))
hist(Temperature)
# define area for the boxplot
par(fig=c(0.8,1,0,1), new=TRUE)
boxplot(Temperature)
# define area for the stripchart
par(fig=c(0.1,0.67,0.1,0.25), new=TRUE)
stripchart(Temperature, method="jitter")
```

The numbers assigned to **fig** were arrived at with a hit-and-trial method to achieve the best looking plot.

