

R Programming

Bency Nyota

2024-03-21

R Programming

R Markdown presentation

We will be using R Markdown for our outline

R is optimized for vector operations and as well, it is an open source software

CRAN :- Comprehensive R Archive Network

loading the data sets

```
library(datasets)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
## 4         4.6         3.1         1.5         0.2   setosa
## 5         5.0         3.6         1.4         0.2   setosa
## 6         5.4         3.9         1.7         0.4   setosa
```

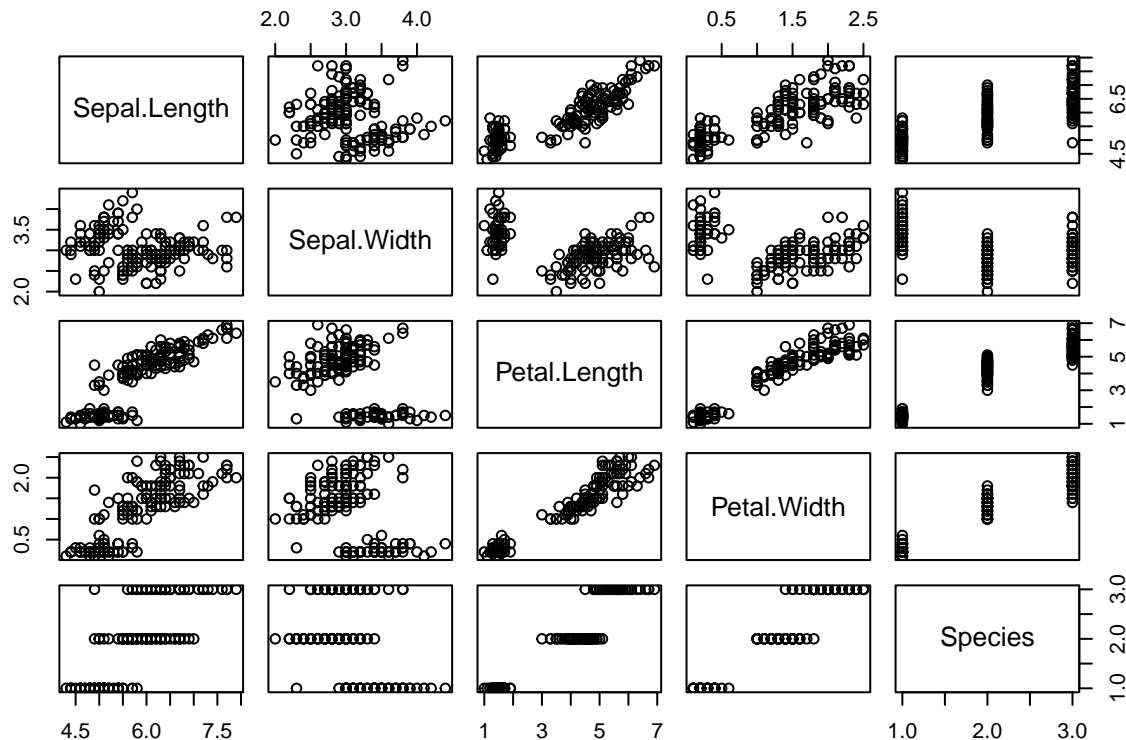
To see the summary of the data set “iris”

```
summary(iris)
```

```
##   Sepal.Length   Sepal.Width   Petal.Length   Petal.Width
## Min.    :4.300   Min.    :2.000   Min.    :1.000   Min.    :0.100
## 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
## Median :5.800   Median :3.000   Median :4.350   Median :1.300
## Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
## 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
## Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##      Species
## setosa    :50
## versicolor:50
## virginica :50
##
##
##
```

Sometimes we can even get to visualize the data directly

```
plot(iris)
```



We can install one packages that carries it many packages we would need

```
install.packages("pacman")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
require("pacman")
```

```
## Loading required package: pacman
```

There are two types of packages in R

- **Base Packages** - download with r (inbuilt) but not loaded o the environment
- **Contributed packages** - they must be downloaded

```
pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggris, lubridate, plotly, rio, rmarkdown, shiny)
```

To load the entire collection

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
## Warning: package 'ggris' is not available for this version of R
##
```

```
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: 'BiocManager' not available. Could not check Bioconductor.
##
```

```
## Please use `install.packages('BiocManager')` and then retry.
```

```
## Warning in p_install(package, character.only = TRUE, ...):
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'ggris'
## Warning in pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggris, : Failed to install/load:
## ggris
```

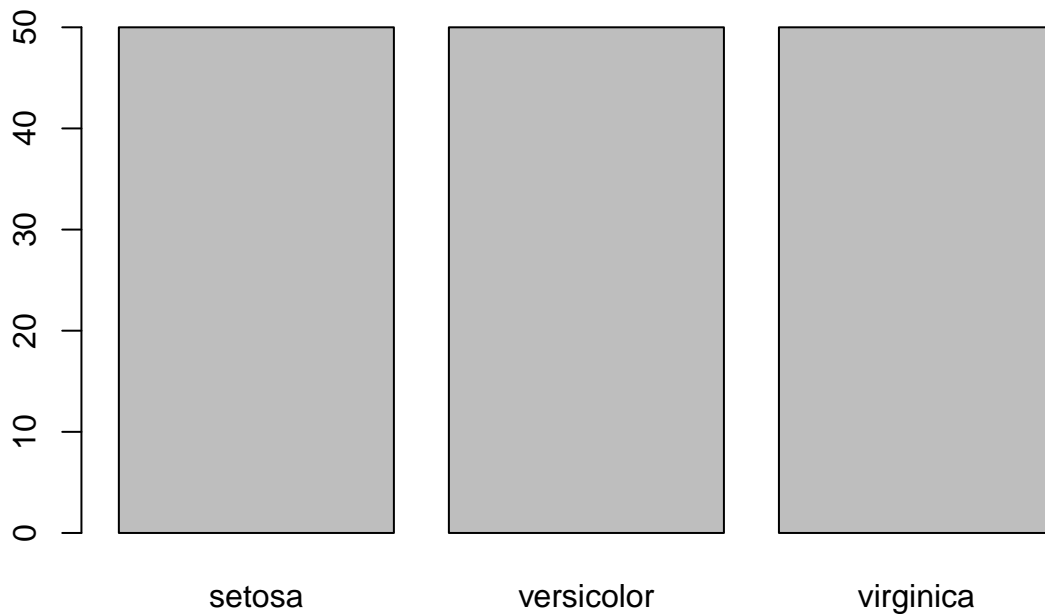
Basic Graphics

we can plot categorical data

```
head(iris)
```

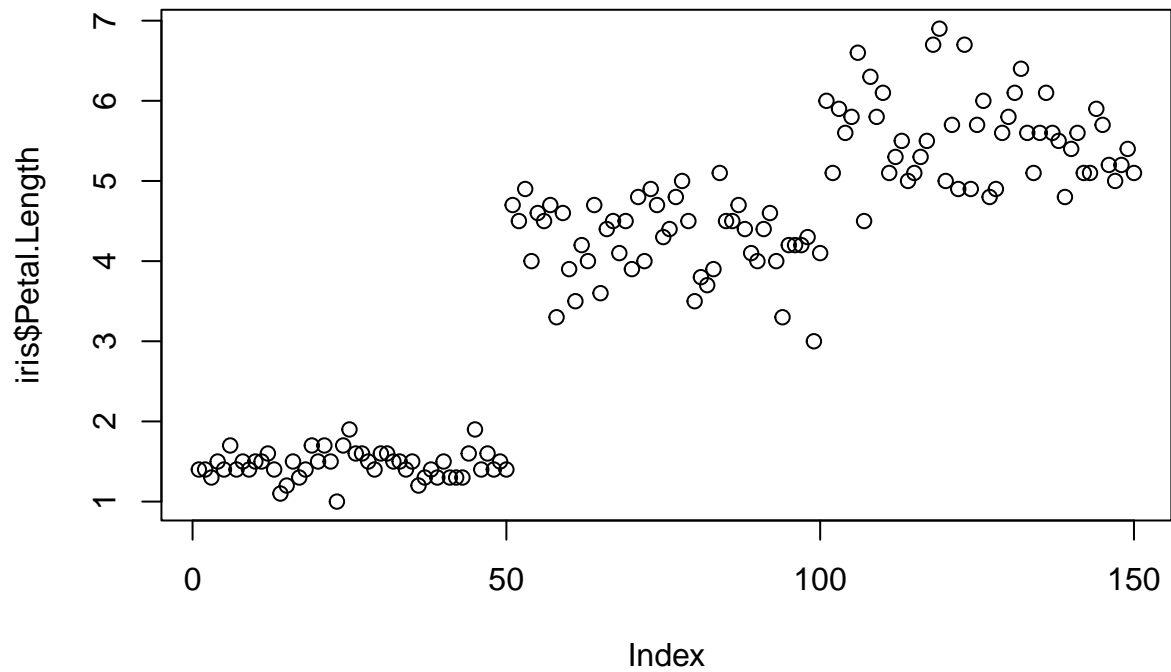
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
## 4         4.6         3.1         1.5         0.2   setosa
## 5         5.0         3.6         1.4         0.2   setosa
## 6         5.4         3.9         1.7         0.4   setosa
```

```
plot(iris$Species)
```



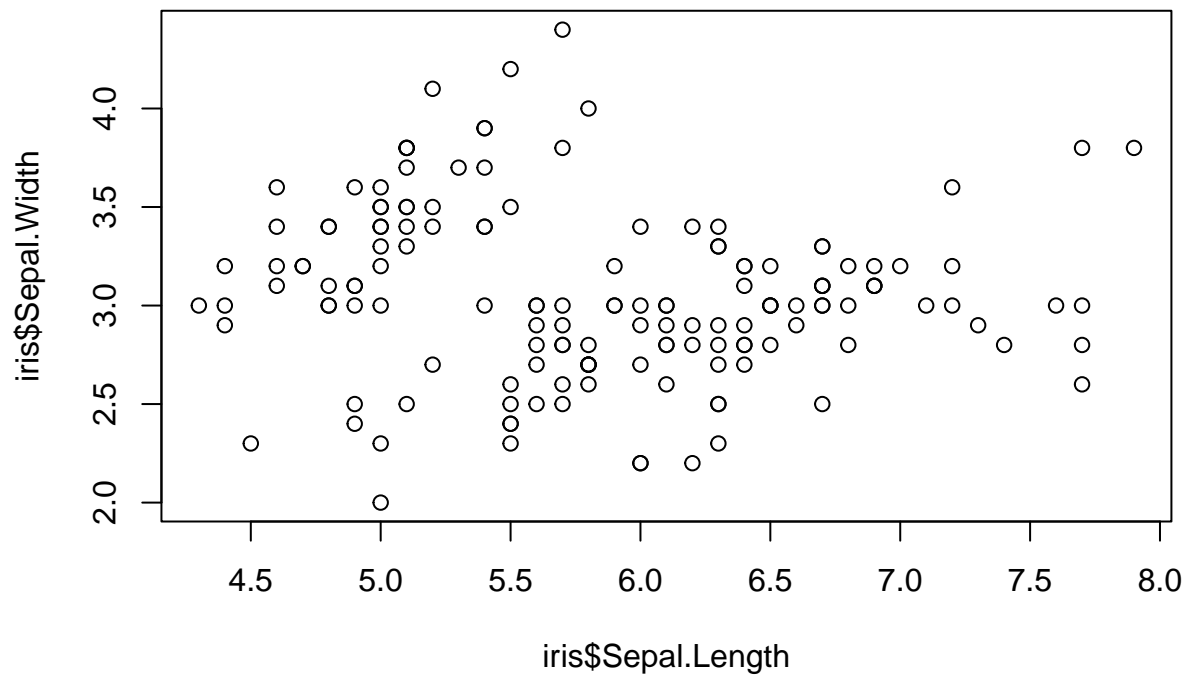
- quantitative variable

```
plot(iris$Petal.Length)
```



Or else plot both quantitative data

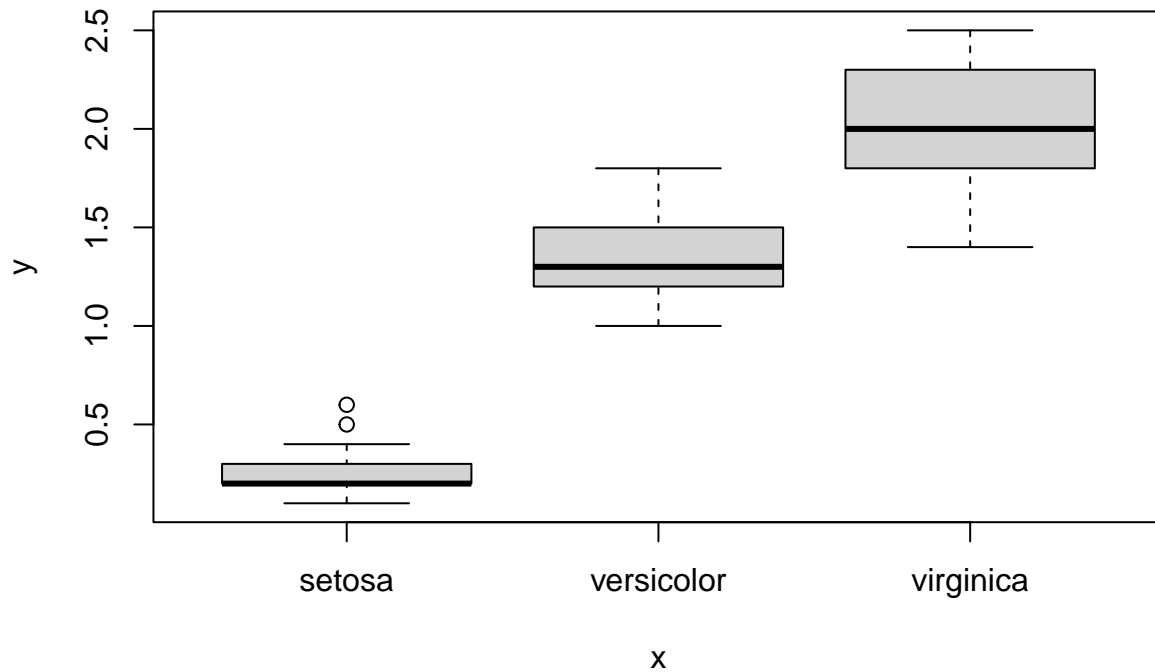
```
plot(iris$Sepal.Length, iris$Sepal.Width)
```



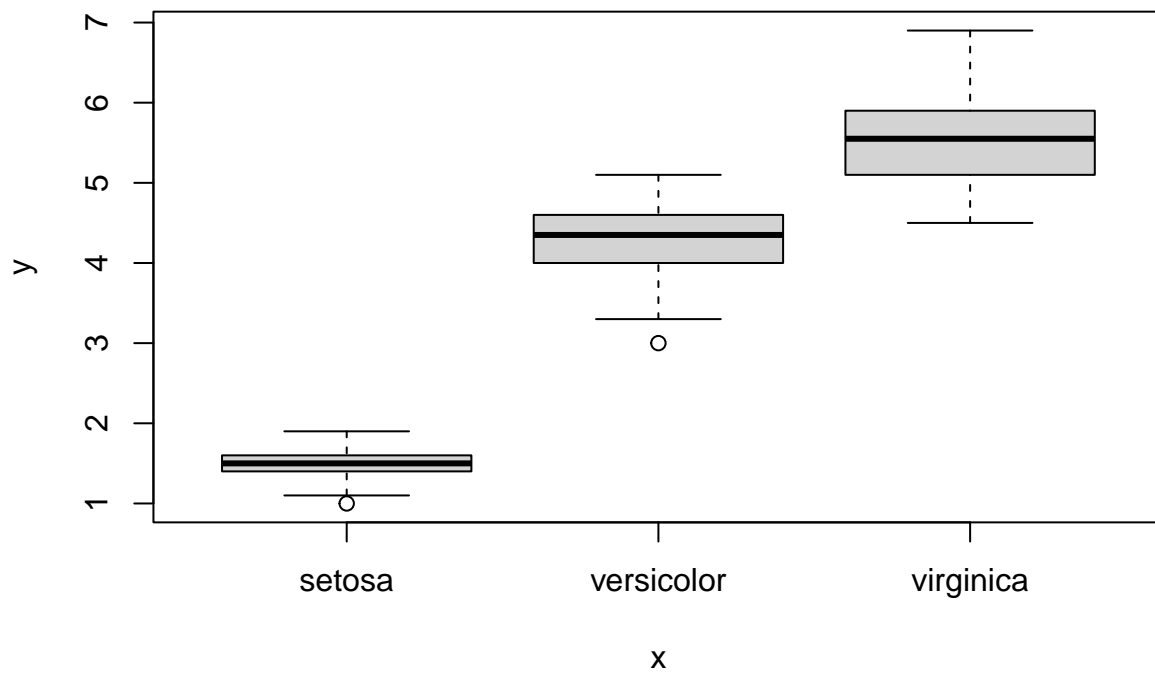
categorical vs quantitative

```
plot(iris$Species, iris$Petal.Width, main="Species vs Petal Width")
```

Species vs Petal Width



```
plot(iris$Species, iris$Petal.Length)
```

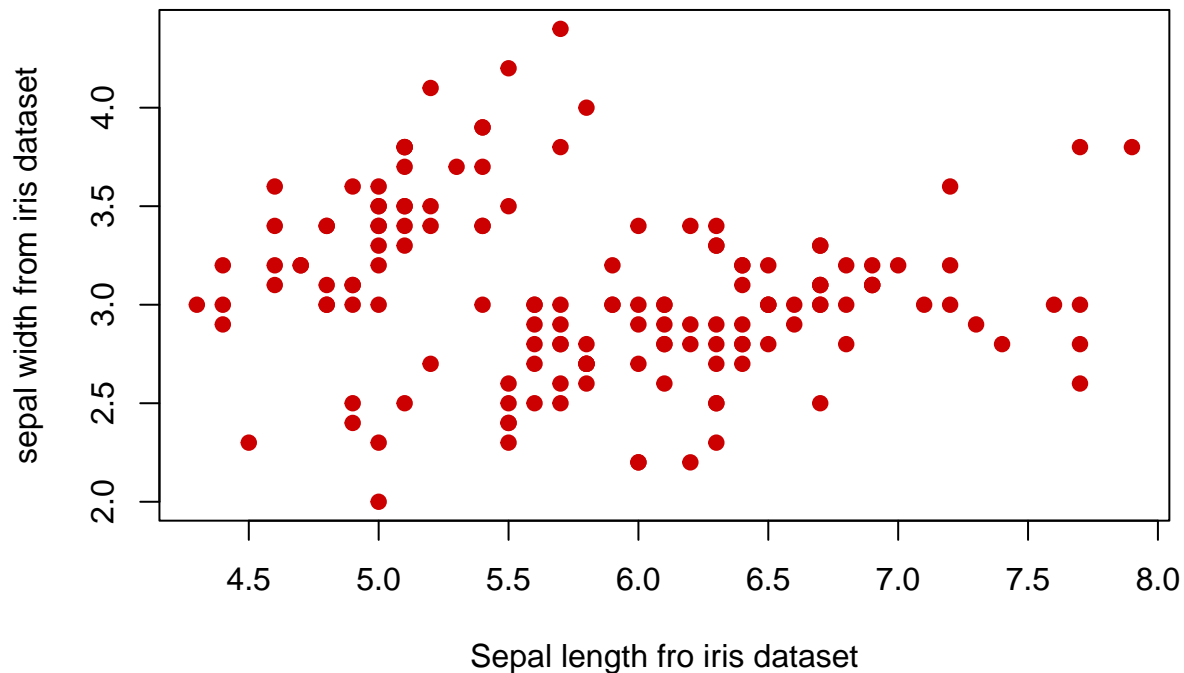


coloring the graphs

```
plot(iris$Sepal.Length, iris$Sepal.Width,
     col = "#cc0000",
     pch = 19, #shape of the pointers for this case is circles
     main = "Sepa Length vs Sepal Width",
     xlab = "Sepal length fro iris dataset",
```

```
ylab = "sepal width from iris dataset")
```

Sepa Length vs Sepal Width



Bar charts

we use mtcars as our dataset

```
head(mtcars)
```

```
##           mpg  cyl  disp  hp  drat    wt  qsec vs  am  gear  carb
## Mazda RX4      21.0   6  160  110 3.90  2.620 16.46 0   1    4    4
## Mazda RX4 Wag  21.0   6  160  110 3.90  2.875 17.02 0   1    4    4
## Datsun 710     22.8   4  108   93 3.85  2.320 18.61 1   1    4    1
## Hornet 4 Drive  21.4   6  258  110 3.08  3.215 19.44 1   0    3    1
## Hornet Sportabout 18.7   8  360  175 3.15  3.440 17.02 0   0    3    2
## Valiant        18.1   6  225  105 2.76  3.460 20.22 1   0    3    1
```

```
summary(mtcars)
```

```
##           mpg           cyl           disp           hp
##  Min.   :10.40  Min.   :4.000  Min.   : 71.1  Min.   : 52.0
## 1st Qu.:15.43  1st Qu.:4.000  1st Qu.:120.8  1st Qu.: 96.5
## Median :19.20  Median :6.000  Median :196.3  Median :123.0
## Mean   :20.09  Mean   :6.188  Mean   :230.7  Mean   :146.7
## 3rd Qu.:22.80  3rd Qu.:8.000  3rd Qu.:326.0  3rd Qu.:180.0
## Max.   :33.90  Max.   :8.000  Max.   :472.0  Max.   :335.0
##           drat           wt           qsec           vs
##  Min.   :2.760  Min.   :1.513  Min.   :14.50  Min.   :0.0000
## 1st Qu.:3.080  1st Qu.:2.581  1st Qu.:16.89  1st Qu.:0.0000
## Median :3.695  Median :3.325  Median :17.71  Median :0.0000
## Mean   :3.597  Mean   :3.217  Mean   :17.85  Mean   :0.4375
## 3rd Qu.:3.920  3rd Qu.:3.610  3rd Qu.:18.90  3rd Qu.:1.0000
```

```
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000
##      am      gear      carb
## Min. :0.0000 Min. :3.000 Min. :1.000
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000
## Median :0.0000 Median :4.000 Median :2.000
## Mean :0.4062 Mean :3.688 Mean :2.812
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000
## Max. :1.0000 Max. :5.000 Max. :8.000
```

Number of rows

```
paste(nrow(mtcars), "rows")
```

```
## [1] "32 rows"
```

Number of columns

```
paste(ncol(mtcars), "Columns")
```

```
## [1] "11 Columns"
```

```
str(mtcars)
```

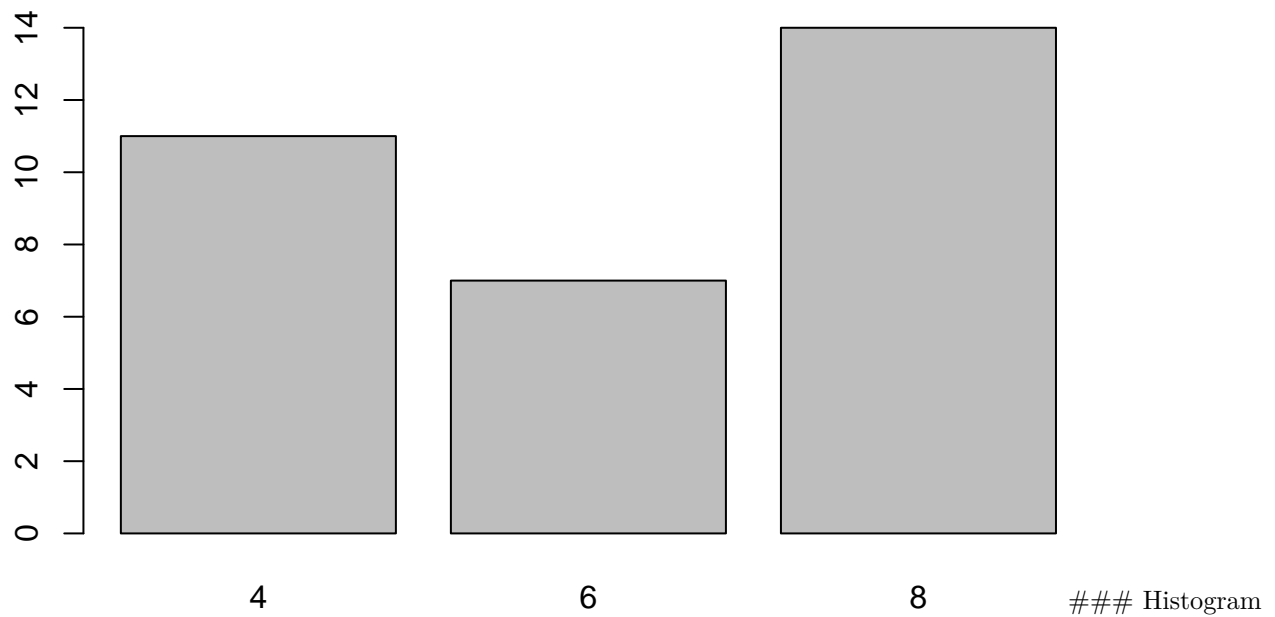
```
## 'data.frame': 32 obs. of 11 variables:
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num 160 160 108 258 360 ...
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...
```

Factoring the vs column

```
mtcars$vs <- factor(mtcars$vs)
```

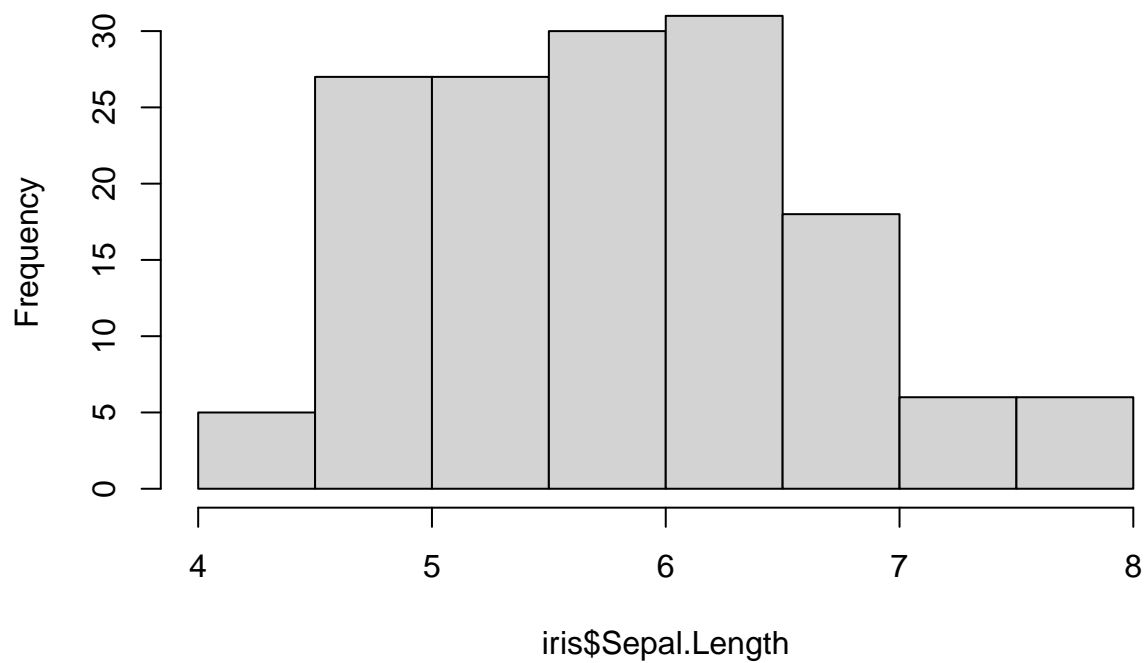
```
#This is more of like counting distinct to create a table
cylinders <- table(mtcars$cyl)
```

```
barplot(cylinders)
```



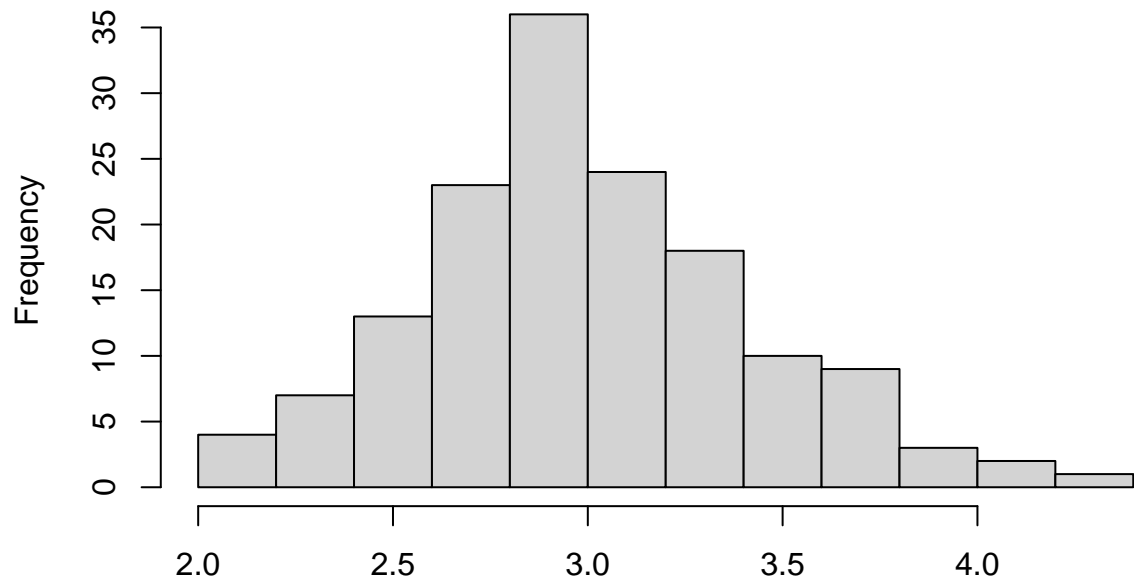
```
hist(iris$Sepal.Length)
```

Histogram of iris\$Sepal.Length



```
hist(iris$Sepal.Width,  
     main="Sepal Width Histogram")
```


Sepal Width Histogram



iris\$Sepal.Width

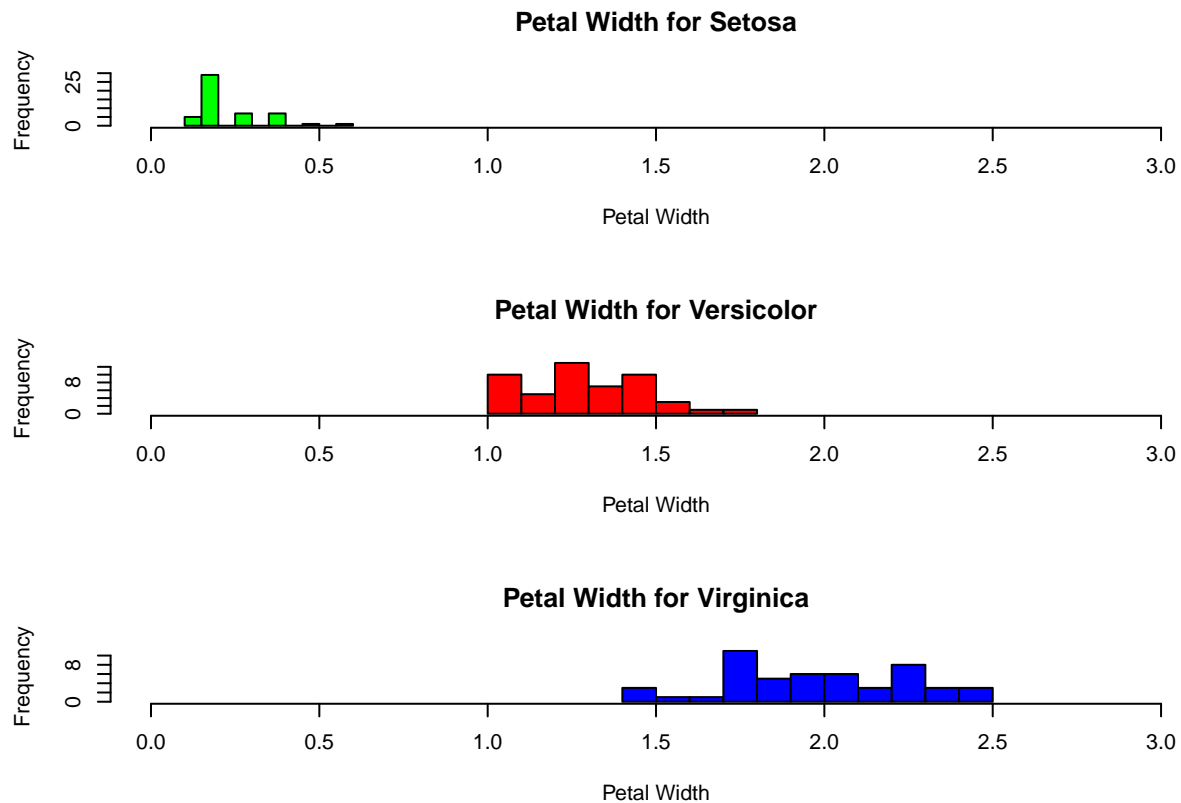
Iris data

```
summary(iris$Species)
```

```
##      setosa versicolor  virginica  
##       50       50       50
```

We can arrange the graphs into different rows and columns

```
par(mfrow = c(3,1)) #3 rows and 1 columns  
#plot histogram for each species  
  
hist(iris$Petal.Width [iris$Species=="setosa"],  
     xlim = c(0,3),  
     breaks = 9,  
     main = "Petal Width for Setosa",  
     xlab = "Petal Width",  
     col = "green")  
  
hist(iris$Petal.Width [iris$Species=="versicolor"],  
     xlim = c(0,3), #xlim?  
     breaks = 9,  
     main = "Petal Width for Versicolor",  
     xlab = "Petal Width",  
     col = "red")  
  
hist(iris$Petal.Width [iris$Species=="virginica"],  
     xlim = c(0,3),  
     breaks = 9,  
     main = "Petal Width for Virginica",  
     xlab = "Petal Width",  
     col = "blue")
```



Restoring the graphics parameters

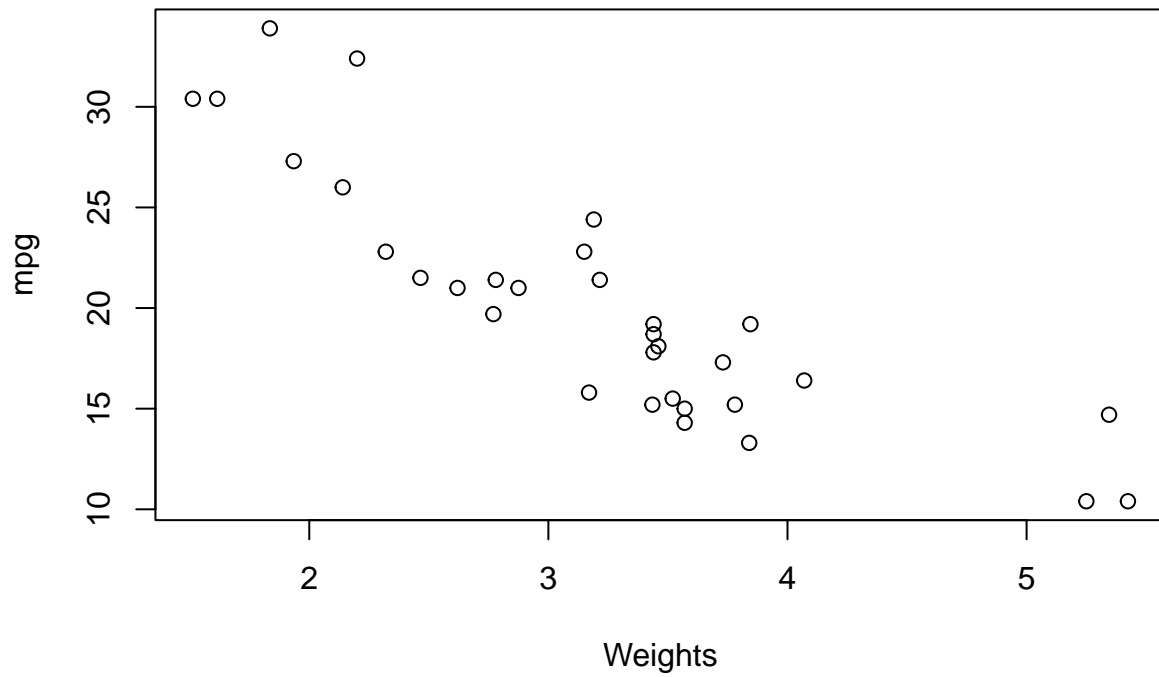
```
par(mfrow=c(1,1))
```

Scatter plot

This case you are plotting between two quantitative variables

```
plot(mtcars$wt, mtcars$mpg,
     main = "MTCARS",
     xlab = "Weights",
     ylab = "mpg")
```

MTCARS



adding

some more plots

```
plot(mtcars$wt, mtcars$mpg,  
     pch = 19,  
     cex = 1.5, #circumference of the circle  
     col = "violet",  
     main = "MTCARS",  
     xlab = "Weights",  
     ylab = "mpg")
```

MTCARS

