DataExploration.R

Wow

Tue Sep 04 08:13:33 2018

```
############################################################
#                      Data Exploration
############################################################
data(iris)

################# Summary Statistics ########################
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  
##                 
##                 
## 
```

```
quantile(iris$Sepal.Length, prob = c(0,0.25,0.5,0.75,1)) #quantile
```

```
##   0%  25%  50%  75% 100% 
##  4.3  5.1  5.8  6.4  7.9
```

```
quantile(iris$Sepal.Length, prob = seq(0,1,by=0.1)) #quantile
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
## 4.30 4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90 7.90
```

```
range(iris$Sepal.Length) #provide max and min
```

```
## [1] 4.3 7.9
```

```
min(iris$Sepal.Length) #minimum
```

```
## [1] 4.3
```

```
max(iris$Sepal.Length) #maximum
```

```
## [1] 7.9
```

```r
mean(iris$Sepal.Length) #mean, average

## [1] 5.843333

median(iris$Sepal.Length) #median

## [1] 5.8

var(iris$Sepal.Length) #variance

## [1] 0.6856935

sd(iris$Sepal.Length) #standard deviaiton

## [1] 0.8280661

# Use apply function
apply(iris[1:4], MARGIN=2, range)

##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]          4.3         2.0          1.0         0.1
## [2,]          7.9         4.4          6.9         2.5

apply(iris[1:4], MARGIN=2, min)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##          4.3          2.0          1.0          0.1

apply(iris[1:4], MARGIN=2, max)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##          7.9          4.4          6.9          2.5

apply(iris[1:4], MARGIN=2, mean)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##     5.843333     3.057333     3.758000     1.199333

apply(iris[1:4], MARGIN=2, median)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##         5.80         3.00         4.35         1.30

apply(iris[1:4], MARGIN=2, var)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##    0.6856935    0.1899794    3.1162779    0.5810063

apply(iris[1:4], MARGIN=2, sd)

## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##    0.8280661    0.4358663    1.7652982    0.7622377

# covariance
cov(iris[,1:4]) #covariance between attributes
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

```
cov_objs <- cov(t(iris[,1:4])) #covariance between objects
cov_objs[1:10,1:10]
```

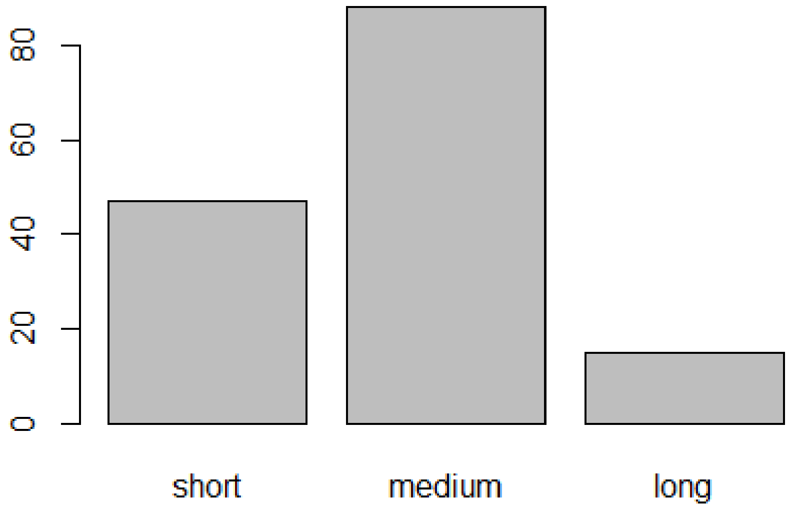
```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
##  [1,] 4.750000 4.421667 4.353333 4.160000 4.696667 4.860000 4.215000
##  [2,] 4.421667 4.149167 4.055000 3.885000 4.358333 4.515000 3.907500
##  [3,] 4.353333 4.055000 3.990000 3.813333 4.303333 4.453333 3.861667
##  [4,] 4.160000 3.885000 3.813333 3.656667 4.110000 4.256667 3.688333
##  [5,] 4.696667 4.358333 4.303333 4.110000 4.650000 4.810000 4.175000
##  [6,] 4.860000 4.515000 4.453333 4.256667 4.810000 4.976667 4.318333
##  [7,] 4.215000 3.907500 3.861667 3.688333 4.175000 4.318333 3.749167
##  [8,] 4.595000 4.284167 4.211667 4.031667 4.541667 4.701667 4.075833
##  [9,] 3.965000 3.707500 3.635000 3.485000 3.915000 4.055000 3.512500
## [10,] 4.493333 4.210000 4.120000 3.953333 4.433333 4.593333 3.976667
##           [,8]   [,9]    [,10]
##  [1,] 4.595000 3.9650 4.493333
##  [2,] 4.284167 3.7075 4.210000
##  [3,] 4.211667 3.6350 4.120000
##  [4,] 4.031667 3.4850 3.953333
##  [5,] 4.541667 3.9150 4.433333
##  [6,] 4.701667 4.0550 4.593333
##  [7,] 4.075833 3.5125 3.976667
##  [8,] 4.449167 3.8425 4.356667
##  [9,] 3.842500 3.3225 3.770000
## [10,] 4.356667 3.7700 4.280000
```

```
# correlation
cor(iris[,1:4]) #correlation between atrributes
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

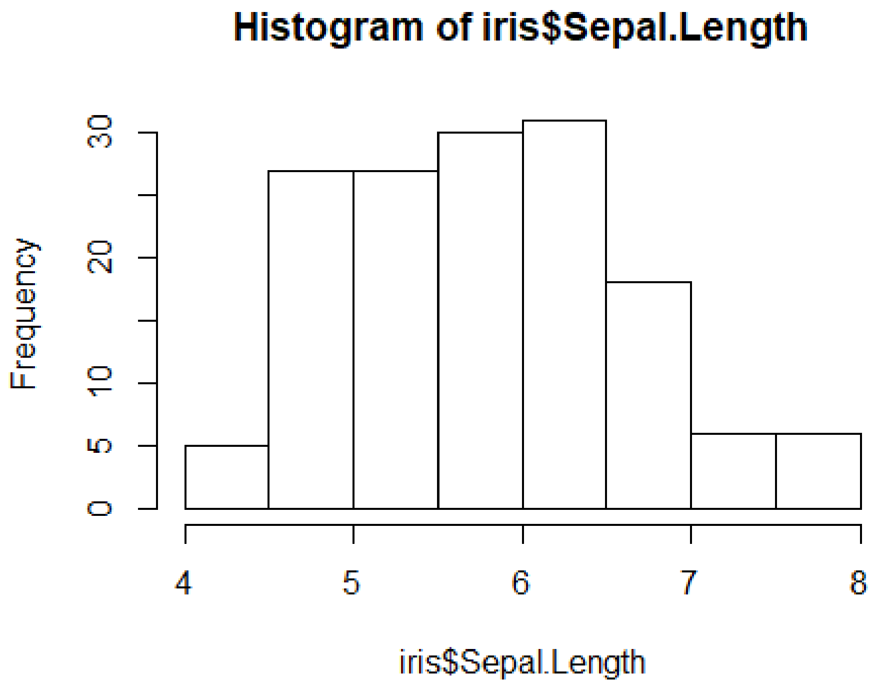
```
cor_objs <- cor(t(iris[,1:4])) #correlation between objects
cor_objs[1:10,1:10]
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
##  [1,] 1.0000000 0.9959987 0.9999739 0.9981685 0.9993473 0.9995861
##  [2,] 0.9959987 1.0000000 0.9966071 0.9973966 0.9922327 0.9935919
##  [3,] 0.9999739 0.9966071 1.0000000 0.9983335 0.9990611 0.9993773
##  [4,] 0.9981685 0.9973966 0.9983335 1.0000000 0.9967188 0.9978326
##  [5,] 0.9993473 0.9922327 0.9990611 0.9967188 1.0000000 0.9998833
##  [6,] 0.9995861 0.9935919 0.9993773 0.9978326 0.9998833 1.0000000
##  [7,] 0.9988112 0.9907206 0.9984377 0.9961394 0.9999140 0.9997226
##  [8,] 0.9995381 0.9971181 0.9996045 0.9995456 0.9985032 0.9991788
##  [9,] 0.9980766 0.9985463 0.9983561 0.9998333 0.9960309 0.9972157
## [10,] 0.9965520 0.9990329 0.9969856 0.9993068 0.9937612 0.9952606
##            [,7]      [,8]      [,9]     [,10]
##  [1,] 0.9988112 0.9995381 0.9980766 0.9965520
##  [2,] 0.9907206 0.9971181 0.9985463 0.9990329
##  [3,] 0.9984377 0.9996045 0.9983561 0.9969856
##  [4,] 0.9961394 0.9995456 0.9998333 0.9993068
##  [5,] 0.9999140 0.9985032 0.9960309 0.9937612
##  [6,] 0.9997226 0.9991788 0.9972157 0.9952606
##  [7,] 1.0000000 0.9979521 0.9952140 0.9927272
##  [8,] 0.9979521 1.0000000 0.9994062 0.9983737
##  [9,] 0.9952140 0.9994062 1.0000000 0.9997398
## [10,] 0.9927272 0.9983737 0.9997398 1.0000000

##################### Visualization #########################
# Cut each attribute into ordered factors with three levels
iris_ord <- data.frame(  # create the new data frame
  cut(iris[,1], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,2], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,3], 3, labels=c("short", "medium", "long"), ordered=T),
  cut(iris[,4], 3, labels=c("short", "medium", "long"), ordered=T),
  iris[,5])
colnames(iris_ord) <- colnames(iris) #assign column names

####################### Bar Plot ############################
sw <- table(iris_ord$Sepal.Width)
barplot(sw)
```

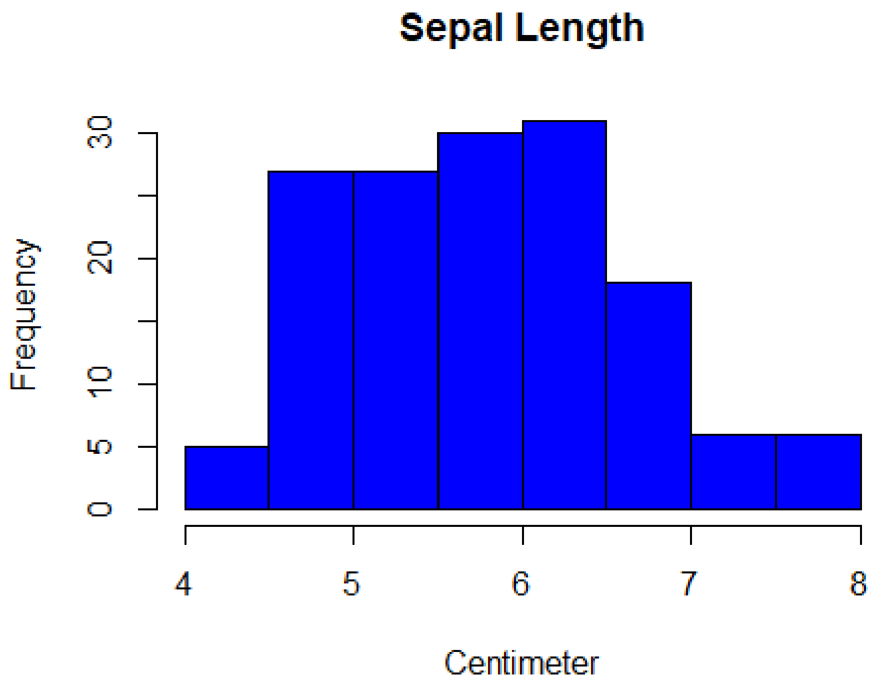


```
########### Stem and leaf plots of sepal length #############
sls <- sort(iris$Sepal.Length) #sort sepal length in ascending order
sls

##   [1] 4.3 4.4 4.4 4.4 4.5 4.6 4.6 4.6 4.6 4.7 4.7 4.8 4.8 4.8 4.8 4.8 4.9
##  [18] 4.9 4.9 4.9 4.9 4.9 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.1 5.1
##  [35] 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.2 5.2 5.2 5.2 5.3 5.4 5.4 5.4 5.4 5.4
##  [52] 5.4 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.6 5.6 5.6 5.6 5.6 5.6 5.7 5.7 5.7
##  [69] 5.7 5.7 5.7 5.7 5.7 5.8 5.8 5.8 5.8 5.8 5.8 5.8 5.9 5.9 5.9 6.0 6.0
##  [86] 6.0 6.0 6.0 6.0 6.1 6.1 6.1 6.1 6.1 6.1 6.2 6.2 6.2 6.2 6.3 6.3 6.3
## [103] 6.3 6.3 6.3 6.3 6.3 6.3 6.4 6.4 6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5
## [120] 6.5 6.6 6.6 6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.8 6.8 6.8 6.9 6.9 6.9
## [137] 6.9 7.0 7.1 7.2 7.2 7.2 7.3 7.4 7.6 7.7 7.7 7.7 7.7 7.9

slp1 <- stem(iris$Sepal.Length,scale=0.10) #each line contains 2 stems

##
##   The decimal point is at the |
##
##   4 |
34445666677888889999990000000000111111111222234444445555555666666777+3
##   6 | 0000001111112222333333333444444455555667777777788899990122234677779

slp1 <- stem(iris$Sepal.Length,scale=0.10, width=100) #default width = 80

##
##   The decimal point is at the |
##
```

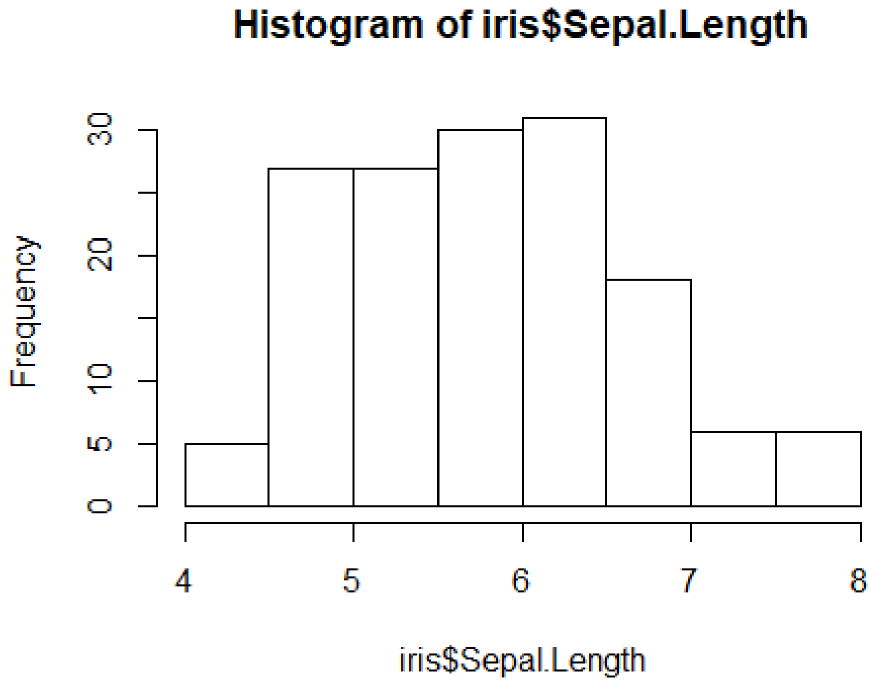
```
##   4 |
34445666677888889999990000000000111111111222234444445555555666666777777778888
888999
##   6 | 0000001111112222333333333444444455555667777777788899990122234677779

slp2 <- stem(iris$Sepal.Length,scale=0.25)

##
##   The decimal point is at the |
##
##   4 | 3444566667788888999999
##   5 | 0000000000111111111222234444445555555666666777777778888888999
##   6 | 0000001111112222333333333444444455555667777777788899999
##   7 | 0122234677779

slp3 <- stem(iris$Sepal.Length,scale=0.50) #two buckets per stem

##
##   The decimal point is at the |
##
##   4 | 3444
##   4 | 566667788888999999
##   5 | 000000000011111111122223444444
##   5 | 5555555666666777777778888888999
##   6 | 000000111111222233333333344444444
##   6 | 5555566777777778889999
##   7 | 0122234
##   7 | 677779

######################### Histograms ########################
hist(iris$Sepal.Length) # Histograms of sepal length
```



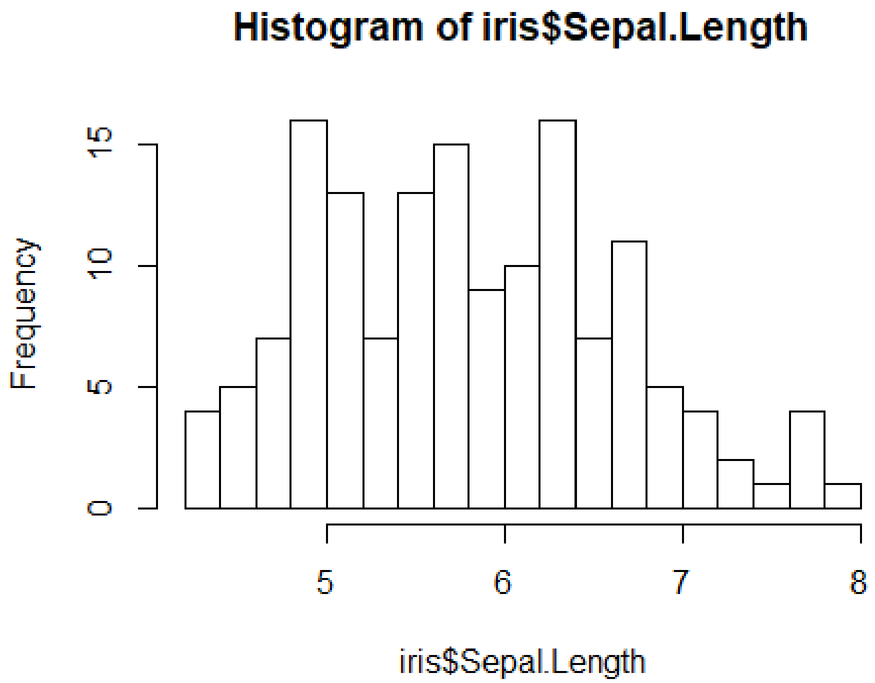
```
hist(iris$Sepal.Length, col = "blue", border = "black", main = "Sepal
Length", xlab = "Centimeter") #Use arguments to make the plot looks better
```



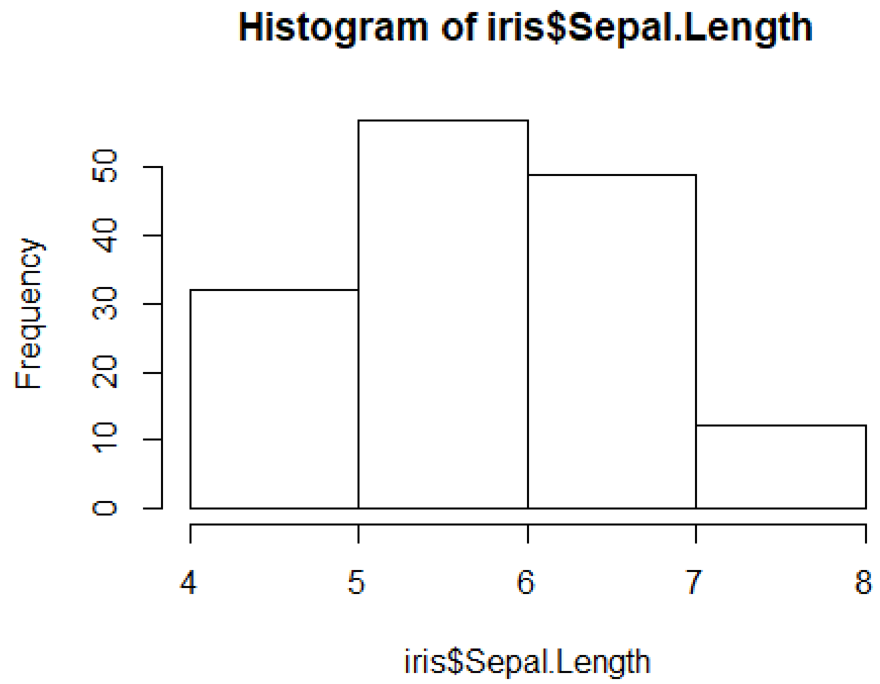
```
# See http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf for color options

# Specify number of bins
hist(iris$Sepal.Length, breaks=8) #8 bins
```



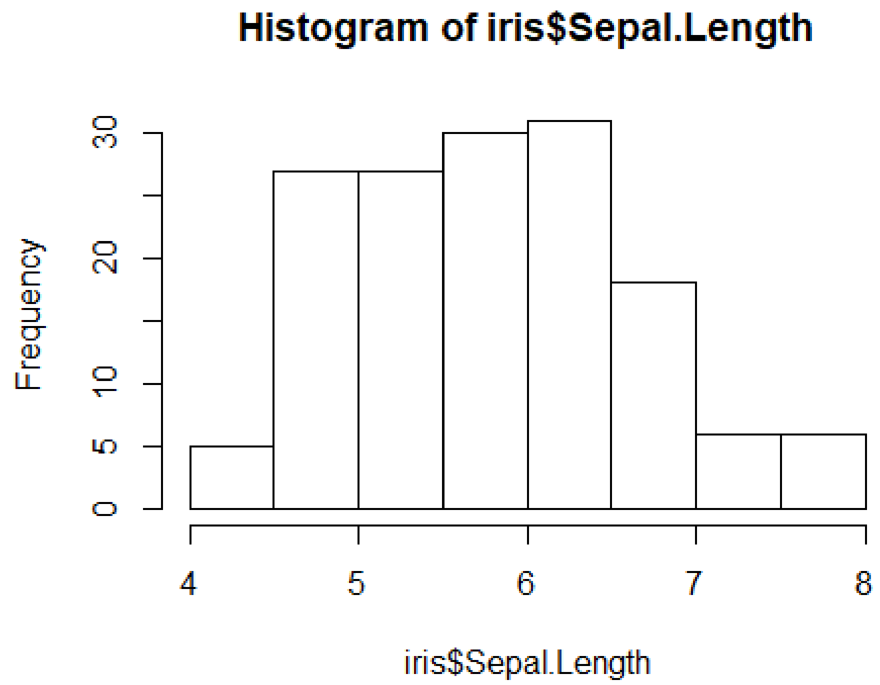
```
hist(iris$Sepal.Length, breaks=10) #10 bins (gets 8 bins, actually)
hist(iris$Sepal.Length, breaks=20) #20 bins (gets 19 bins, actually)
```



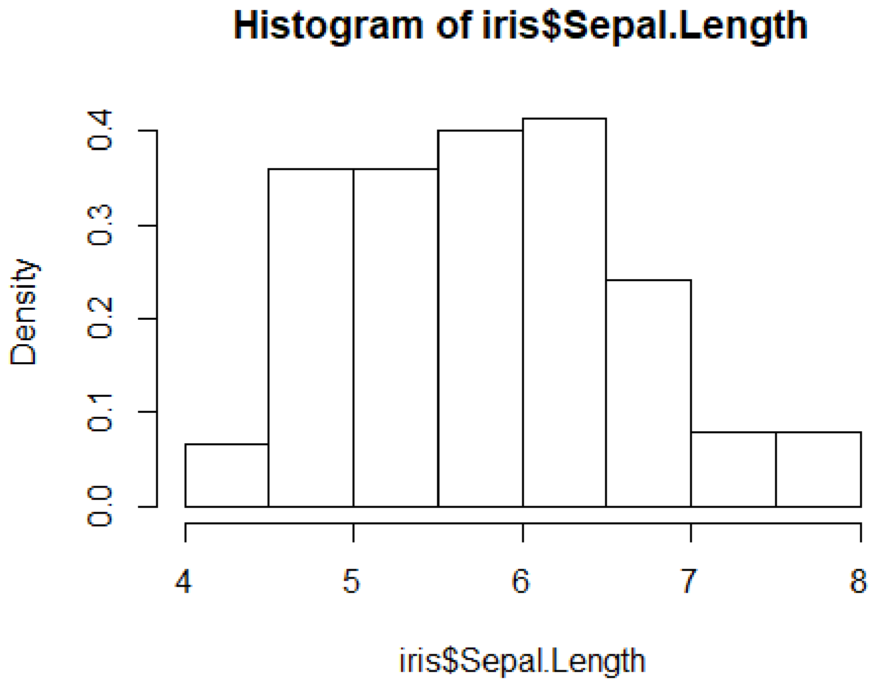
```
# The bins donâ  t correspond to exactly the number you put in, because of
the way R runs
# its algorithm to break up the data but it gives you generally what you
want. If you
# want more control over exactly the breakpoints between bins, you can be
more precise with
# the breaks() option and give it a vector of breakpoints, like this:
hist(iris$Sepal.Length, breaks=c(4.0,5.0,6.0,7.0,8.0)) #define split points
of bins using breaks()
```



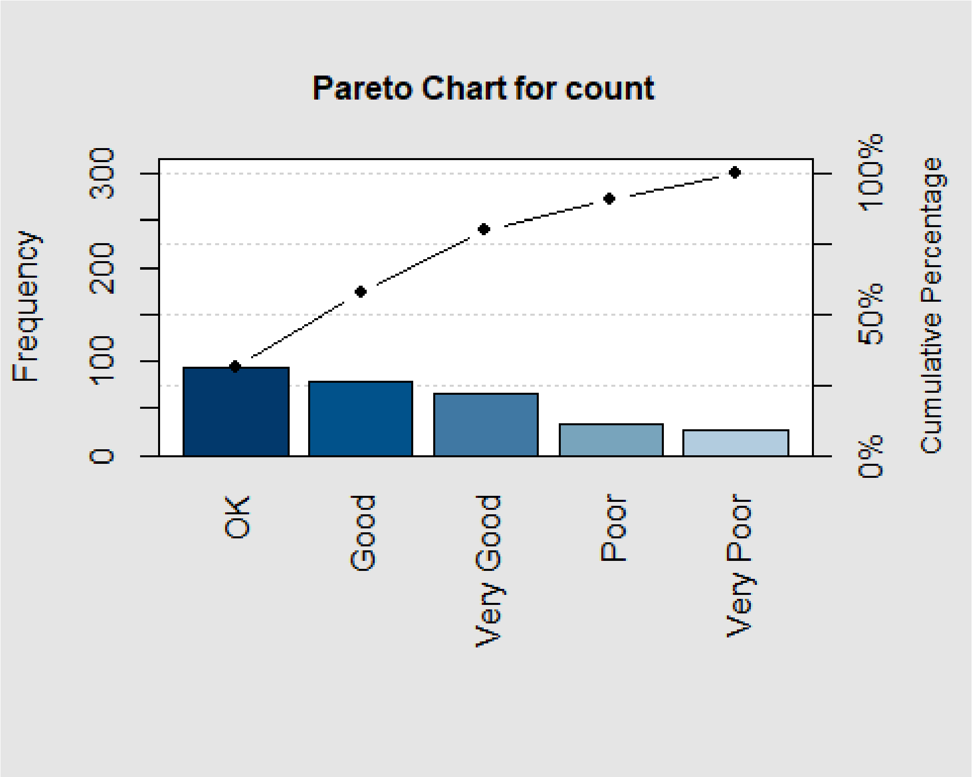
```r
hist(iris$Sepal.Length, breaks=seq(4.0,8.0,by=0.5)) #define breaks using
seq()
```



```
# Relative frequency histogram
hist(iris$Sepal.Length, freq = FALSE)
```



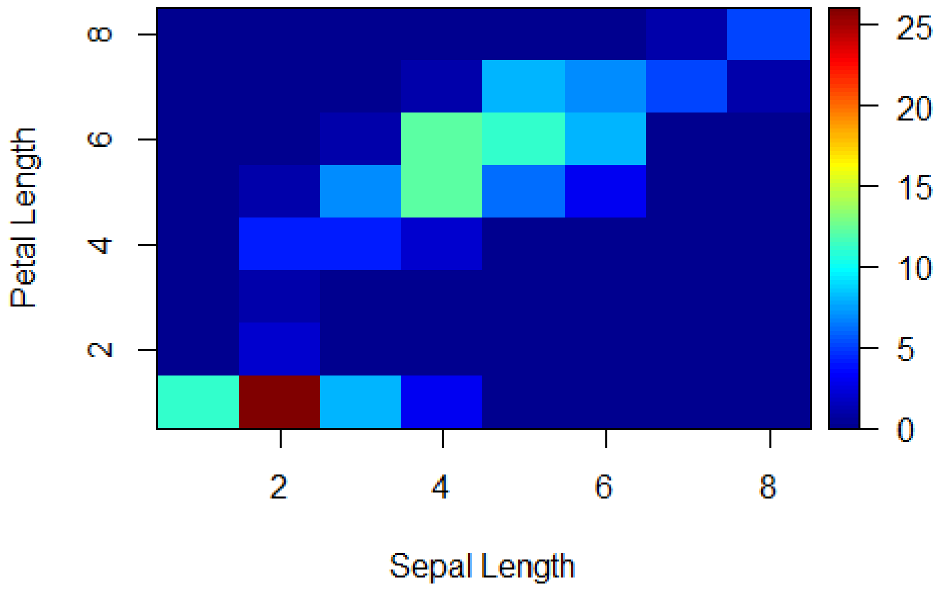
```
# Pareto histogram for categorical attribute
library(qcc)

## Package 'qcc' version 2.7

## Type 'citation("qcc")' for citing this R package in publications.

count <- c(80, 27, 66, 94, 33)
names(count) <- c("Good", "Very Poor", "Very Good", "OK", "Poor")
pareto.chart(count)
```



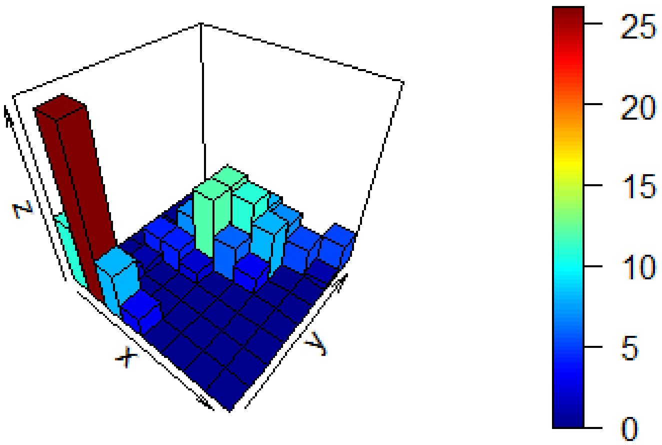
```
##
## Pareto chart analysis for count
##              Frequency Cum.Freq. Percentage Cum.Percent.
##   OK          94.00000  94.00000   31.33333     31.33333
##   Good        80.00000 174.00000   26.66667     58.00000
##   Very Good   66.00000 240.00000   22.00000     80.00000
##   Poor        33.00000 273.00000   11.00000     91.00000
##   Very Poor   27.00000 300.00000    9.00000    100.00000
```

```
# 2-D histograms
library(plot3D)
SepL <- cut(iris[,1],8) #cut sepal length into 8 bins
PetL <- cut(iris[,3],8) #cut petal length into 8 bins
tb <- table(SepL,PetL) #calculate joint counts at cut levels (cross
tabulation)
tb
```

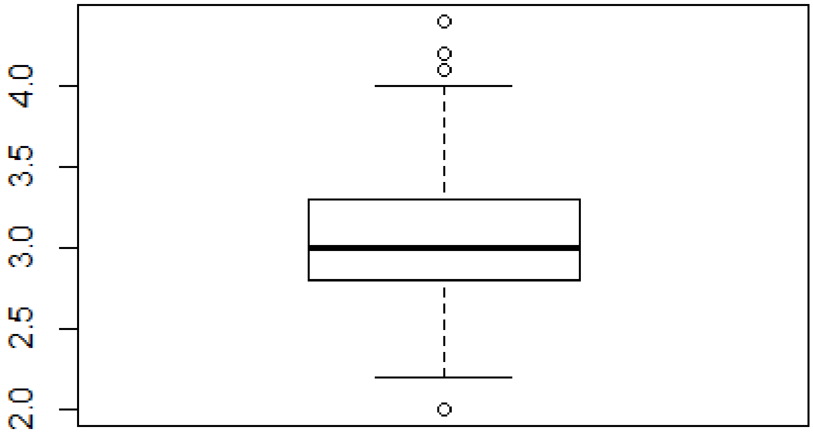
```
##              PetL
## SepL          (0.994,1.74] (1.74,2.48] (2.48,3.21] (3.21,3.95] (3.95,4.69]
##   (4.3,4.75]            11           0           0           0           0
##   (4.75,5.2]            26           2           1           4           1
##   (5.2,5.65]             8           0           0           4           7
##   (5.65,6.1]             3           0           0           2          12
##   (6.1,6.55]             0           0           0           0           6
##   (6.55,7]               0           0           0           0           3
##   (7,7.45]               0           0           0           0           0
##   (7.45,7.9]             0           0           0           0           0
##              PetL
## SepL          (4.69,5.43] (5.43,6.16] (6.16,6.91]
##   (4.3,4.75]            0           0           0
##   (4.75,5.2]            0           0           0
##   (5.2,5.65]            1           0           0
##   (5.65,6.1]           12           1           0
##   (6.1,6.55]           11           8           0
##   (6.55,7]              8           7           0
##   (7,7.45]              0           5           1
##   (7.45,7.9]            0           1           5

image2D(tb, x = 1:8, y = 1:8, xlab = "Sepal Length", ylab = "Petal Length")
```



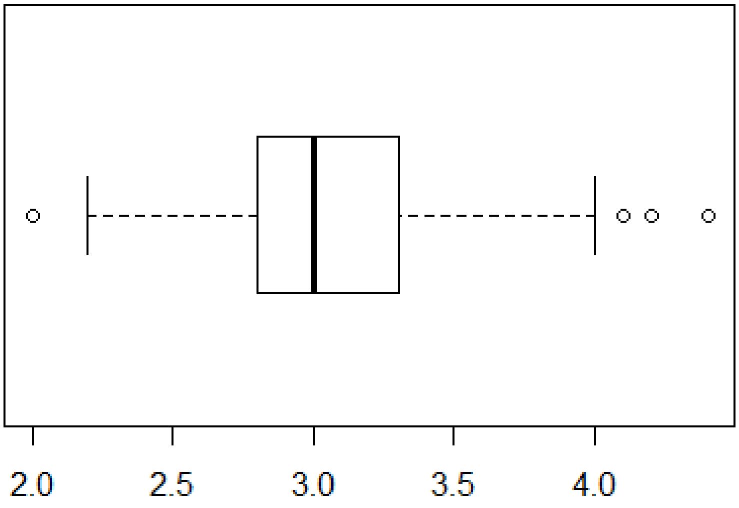
```
# 3-D histograms
hist3D(z=tb, border="black")
```



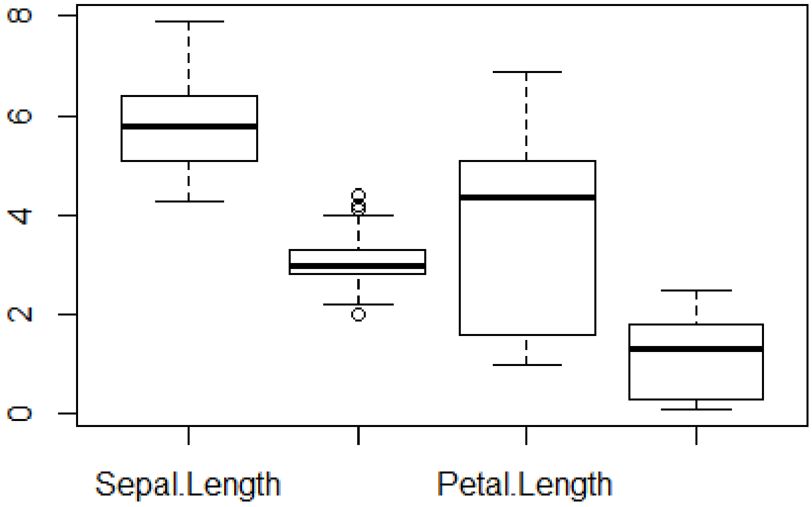
```
######################## Box plots ########################
boxplot(iris$Sepal.Width) #one attribute
```



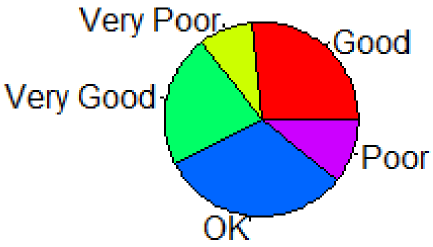
```r
boxplot(iris$Sepal.Width, horizontal = TRUE) #one attribute
```



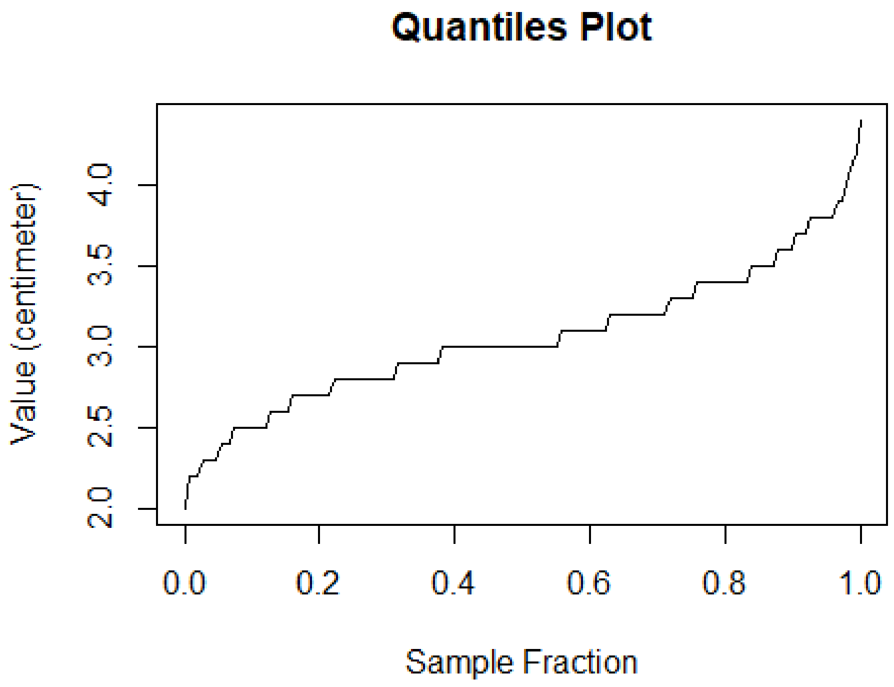
```r
boxplot(iris[,1:4]) #4 attributes
```



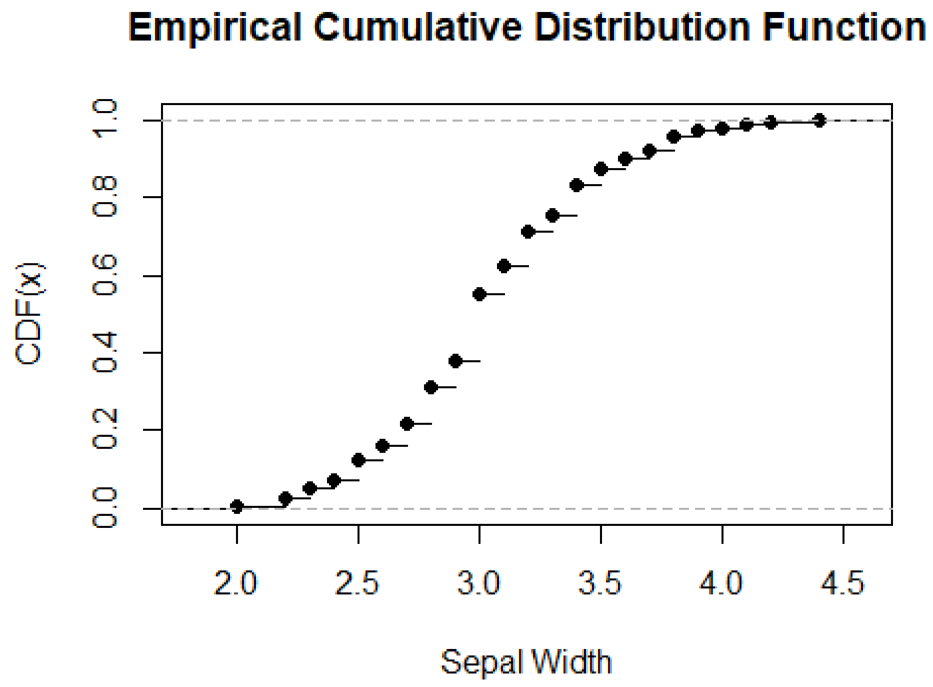
```
####################### Pie chart ########################
pie(count, col = rainbow(5), radius = 0.5)
```



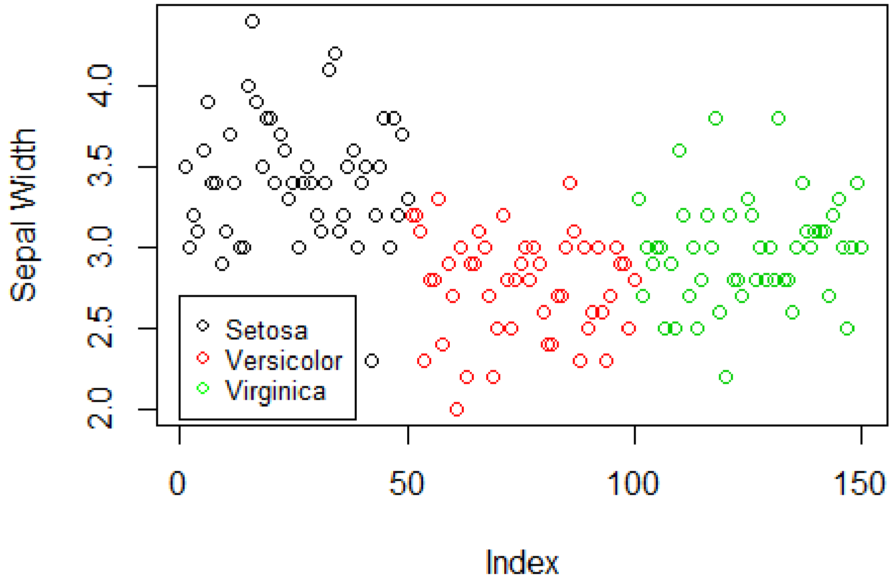
```
# for color palette see: https://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/palettes.html

##################### Quantiles plot ######################
# There is no built-in quantile plot in R, but it is relatively simple to produce one.
x <- iris$Sepal.Width
n <- length(x)
plot((1:n-1)/(n-1),sort(x),type="l",main="Quantiles Plot",xlab="Sample Fraction",ylab="Value (centimeter)")
```



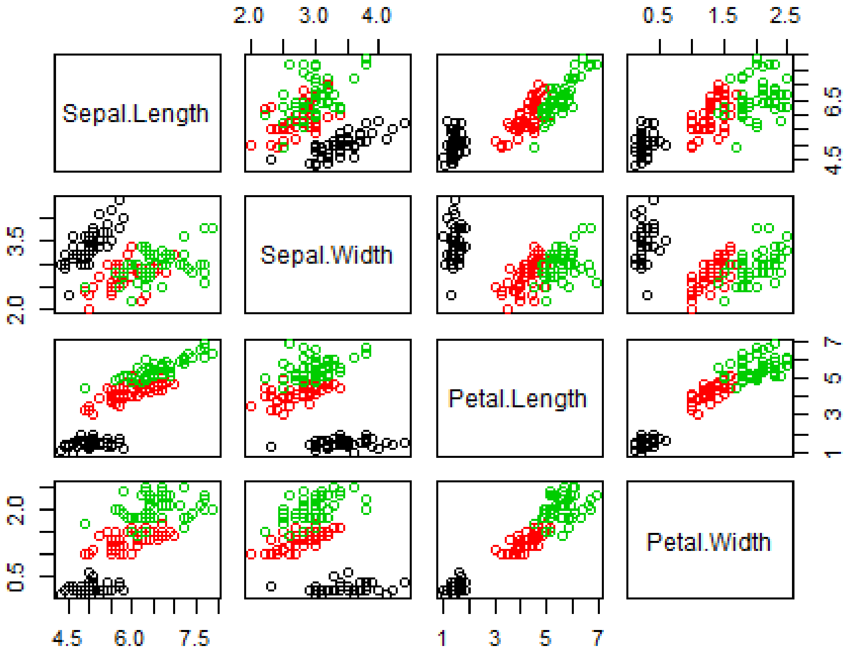
```
#### Empirical cumulative distribution function plot #####
x <- ecdf(iris$Sepal.Width)
plot(x, main = "Empirical Cumulative Distribution Function", xlab="Sepal
Width", ylab="CDF(x)", verticals = FALSE, col.01line = "gray70", pch = 19)
```



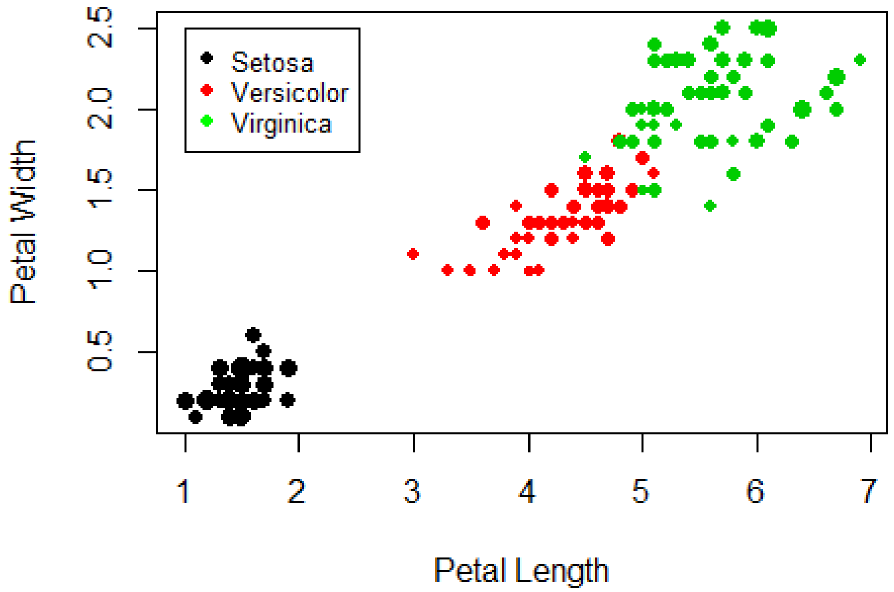
```
##################### Scatter plot #######################
plot(iris$Sepal.Width, ylab="Sepal Width", col=iris$Species)
legend(0,2.7,legend=c("Setosa", "Versicolor","Virginica"),
col=c("black","red","green"), pch=1, cex=0.8) # add legend to location x=0,
y=2.7
```



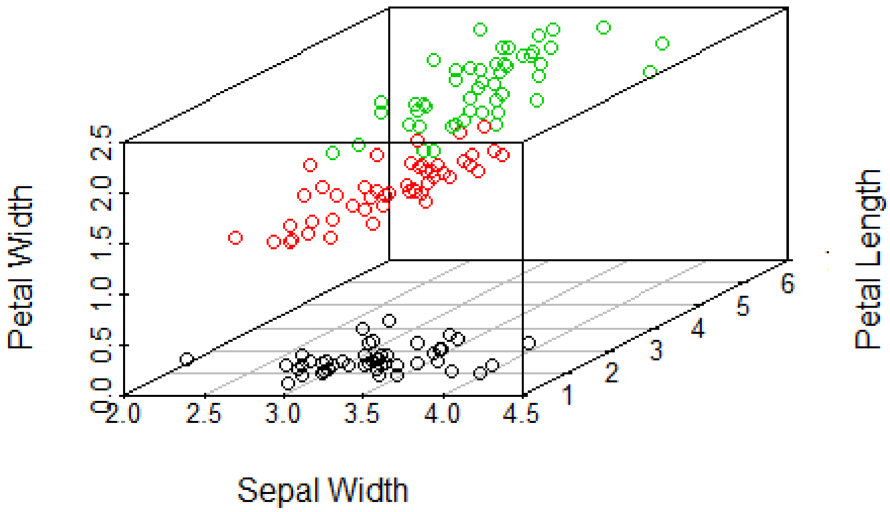
```
#Matrix of scatter plots
plot(iris[,1:4], col=iris$Species)
```



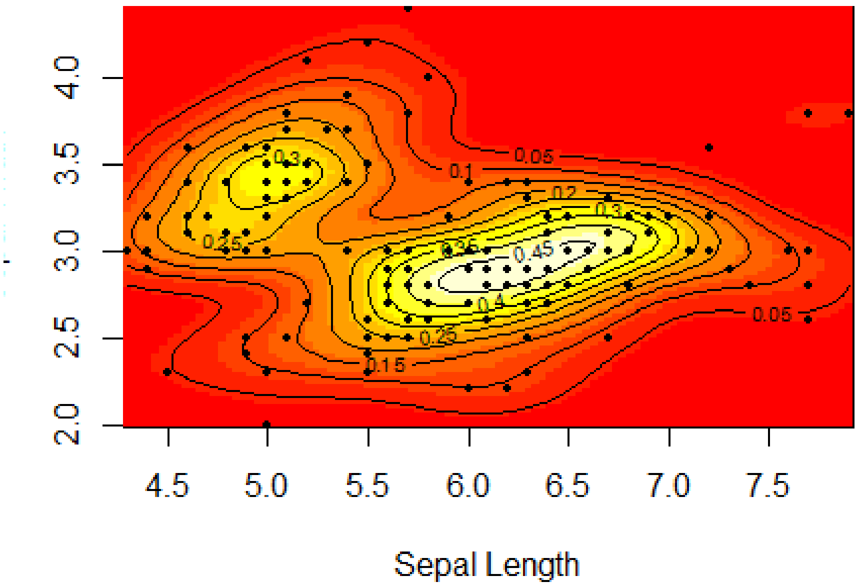
```
#Scatter plot of iris associating 4 attributes
plot(iris$Petal.Length,iris$Petal.Width, xlab = "Petal Length", ylab = "Petal
Width", col=iris$Species, pch=16, cex=(iris$Sepal.Width/3))
legend(1,2.5,legend=c("Setosa", "Versicolor","Virginica"),
col=c("black","red","green"), pch=16, cex=0.8)
```



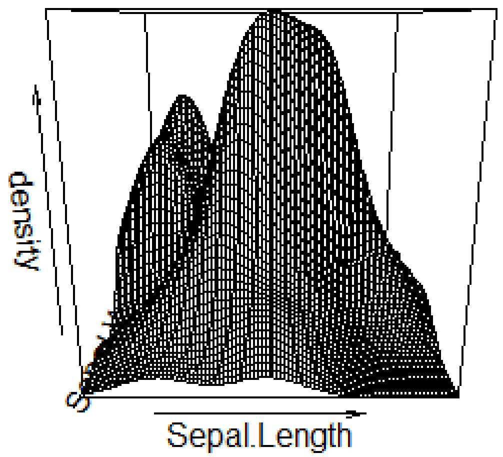
```
#3d scatter plot (associates 4 attributes)
library(scatterplot3d)
# Rename all levels in Species attribute (column 5) to number from 1-3
levels(iris$Species) <- c("1","2","3")
scatterplot3d(iris$Sepal.Width,iris$Petal.Length,iris$Petal.Width,
color=iris$Species,xlab="Sepal Width",ylab="Petal Length",zlab="Petal Width")
```



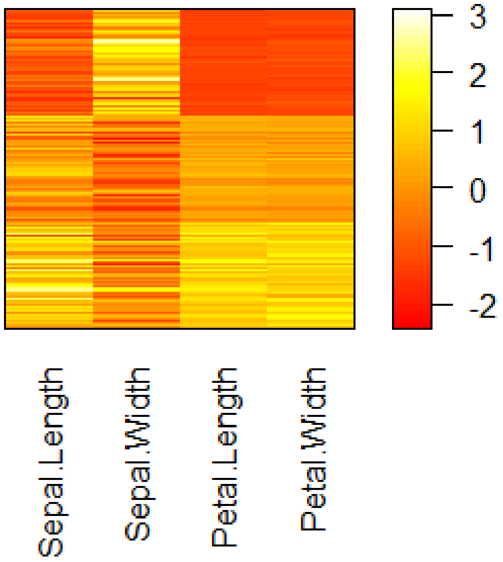
```
################# Contour plot of density ################
library(MASS)
dens <- kde2d(iris$Sepal.Length, iris$Sepal.Width, n=100)
image(dens, xlab="Sepal Length", ylab="Sepal Width")
contour(dens, add=TRUE) #add contour line
points(iris$Sepal.Length, iris$Sepal.Width, cex = 0.5, pch = 16) #add data
points
```



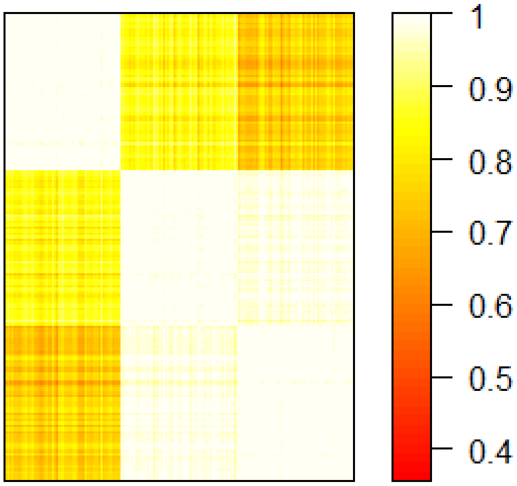
```
persp(dens, xlab="Sepal.Length", ylab="Sepal.Width", zlab="density")

##################### Matrix plots #######################
iris_s <- scale(iris[1:4]) #standardized all the data points to have mean of
0 and standard deviation of
library(seriation)
```



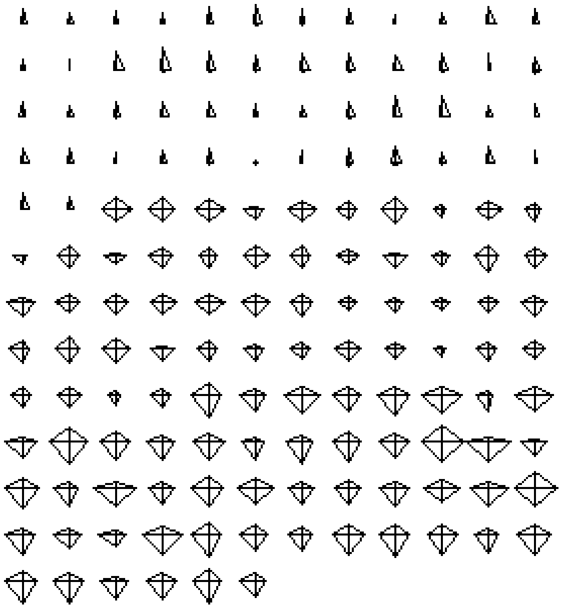
```
pimage(iris_s, col = heat.colors(50))
```



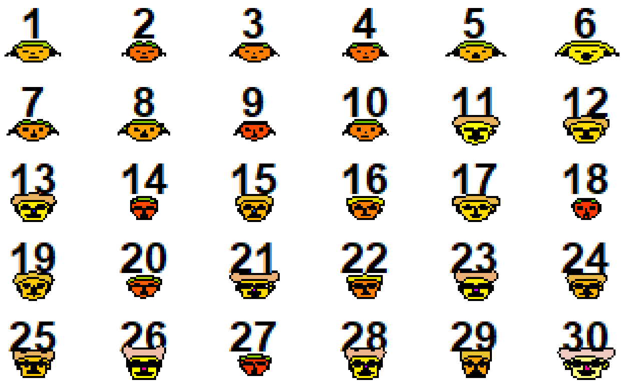
```
# Plot of the iris correlation matrix
cc <- cor(t(iris[,1:4])) #correlation between objects
pimage(cc, col = heat.colors(50))
```



```
###################### Star plots #########################
stars(iris_s)
```



```
#################### Chernoff face #########################
library(aplpack)

## Loading required package: tcltk

faces(iris_s[c(1:10,51:60,101:110),]) #first 10 flowers from each species
```



```
## effect of variables:
##  modified item       Var
##  "height of face   " "Sepal.Length"
##  "width of face    " "Sepal.Width"
##  "structure of face" "Petal.Length"
##  "height of mouth  " "Petal.Width"
##  "width of mouth   " "Sepal.Length"
##  "smiling          " "Sepal.Width"
##  "height of eyes   " "Petal.Length"
##  "width of eyes    " "Petal.Width"
##  "height of hair   " "Sepal.Length"
##  "width of hair   "  "Sepal.Width"
##  "style of hair   "  "Petal.Length"
##  "height of nose  "  "Petal.Width"
##  "width of nose   "  "Sepal.Length"
##  "width of ear    "  "Sepal.Width"
##  "height of ear   "  "Petal.Length"
```