SSIE 637 - DATA VISUALIZATION

11/14/2016

BELOW ARE VISUALIZATIONS USING PYTHON AND R THAT TALKED ABOUT IN CLASS TODAY .I COMBINED ALL INTO ONE RMARKDOWN FILE!THE OUTPUT ARE ALL IN THERE .IF YOU ARE INTERESTED IN RUNING THE CODE ,YOU CAN COPY AND PAST INTO PYTHON OR R DEPENDING ON THE CODE.//Dieudonne

DATA VISUALIZATION

To understand thousands of rows of data in a limited time there is no alternative to visual representation. Objective of visualization is to reveal the hidden information through simple charts and diagrams. Visual representation of data is the first step towards data exploration and formulation of analytical relationship among the variables. In a whirl of complex and voluminous data, visualization in one, two and three dimension helps data analysts to sift through data in a logical manner and understand the data dynamics. It is instrumental in identifying patterns and relationships among groups of variables. Visualization techniques depend on the type of variables. Techniques available to represent nominal variables are generally not suitable for visualizing continuous variables and vice versa. Data often contains complex information. It is easy to internalize complex information through visual mode. Graphs, charts and other visual representation provide quick and focused summarization

Histogram

Histograms are the most common grapical tool to represent continuous data. On the horizontal axis the range of the sample is plotted. On the vertical axis is plotted the frequencies or relative frequencies of each class. The class width has an impact on the shape of the histogram. The histograms in the previous section were drawn from a random sample generated from theoretical distributions. Here we consider a real example to construct histograms.

The data set used for this purpose is the Wage data that is included in the ISLR package in R. A full description of the data is given in the package. The following R code produces the figure below which illustrates the distribution of wage for all 3000 workers.

PYTHON

```
import matplotlib.pyplot as plt
import pandas as pd
df= pd.read_csv('Visualization637.2.csv')
print df
fig=plt.figure() #Plots in matplotlib reside within a figure object, use plt.figure to create new figur
```

```
#Create one or more subplots using add_subplot, because you can't create blank figure
ax = fig.add_subplot(1,1,1)
#Variable
ax.hist(df['Age'],bins = 2) # Here you can play with number of bins
#Labels and Tit
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('#Patient')
plt.show()
import matplotlib.pyplot as plt
import pandas as pd
fig=plt.figure()
ax = fig.add_subplot(1,1,1)
#Variable
ax.boxplot(df['Age'])
plt.show()
#
#df['COST']
import seaborn as sns
sns.violinplot(df['Age'], df['Gender']) #Variable Plot
sns.despine()
var = df.groupby('Gender').Cost.sum() #grouped Total(sum) Cost  at Gender level
fig = plt.figure()
ax1 = fig.add_subplot(1,1,1)
ax1.set_xlabel('Gender')
ax1.set_ylabel('total Cost')
ax1.set_title("Gender wise total Cost")
var.plot(kind='bar')
var = df.groupby('Body').Cost.sum()
fig = plt.figure()
ax1 = fig.add_subplot(1,1,1)
ax1.set_xlabel('Body')
ax1.set_ylabel('Total Cost')
ax1.set_title("Body wise total Cost")
var.plot(kind='line')
var = df.groupby(['Body','Gender']).Cost.sum()
var.unstack().plot(kind='bar',stacked=True,  color=['red','blue'], grid=False)
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.scatter(df['Age'],df['Cost']) #You can also add more variables here to represent color and size.
plt.show()
var=df.groupby(['Gender']).sum().stack()
temp=var.unstack()
type(temp)
x_list = temp['Cost']
label_list = temp.index
plt.axis("equal") #The pie chart is oval by default. To make it a circle use pyplot.axis("equal")
#To show the percentage of each pie slice, pass an output format to the autopctparameter
plt.pie(x_list,labels=label_list,autopct="%1.1f%%")
plt.title("Cost distribution")
plt.show()

import numpy as np
```

```
#Generate a random number, you can refer your data values also
data = np.random.rand(4,2)
rows = list('1234') #rows categories
columns = list('MF') #column categories
fig,ax=plt.subplots()
#Advance color controls
ax.pcolor(data,cmap=plt.cm.Reds,edgecolors='k')
ax.set_xticks(np.arange(0,2)+0.5)
ax.set_yticks(np.arange(0,4)+0.5)
# Here we position the tick labels for x and y axis
ax.xaxis.tick_bottom()
ax.yaxis.tick_left()
#Values against each labels
ax.set_xticklabels(columns,minor=False,fontsize=20)
ax.set_yticklabels(rows,minor=False,fontsize=20)
plt.show()
```

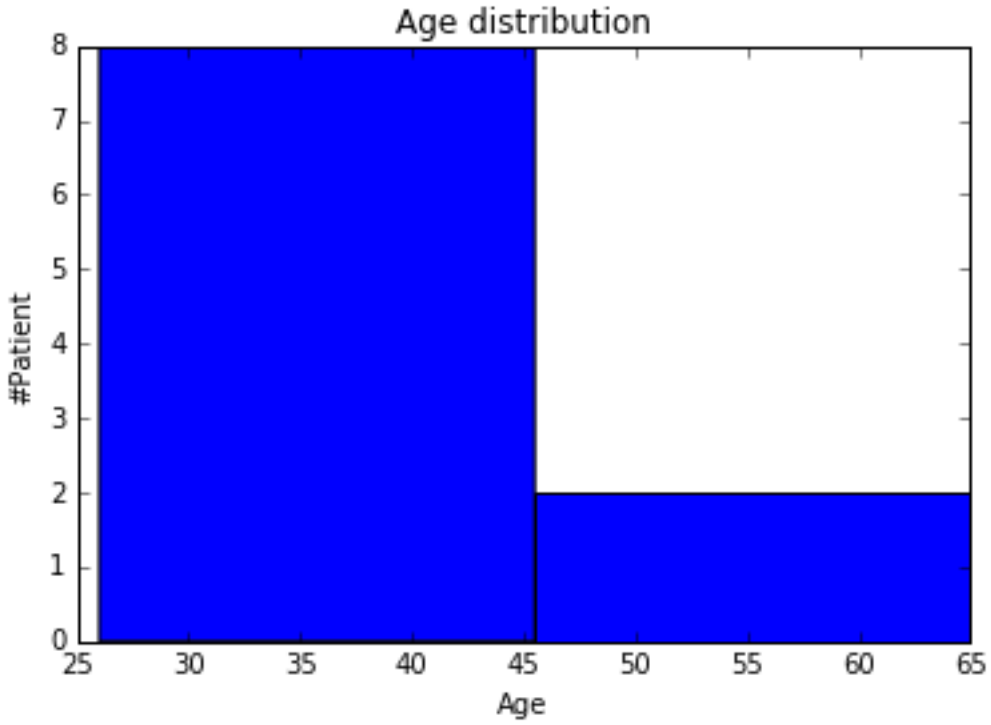


Figure 1: Histogram of Age

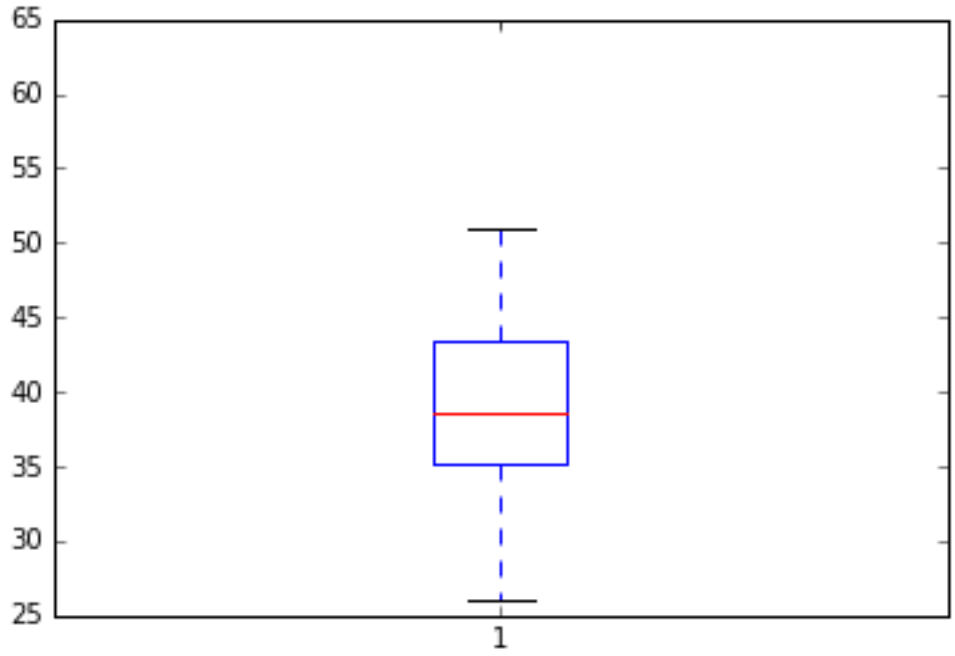


Figure 2: BOXPLOT

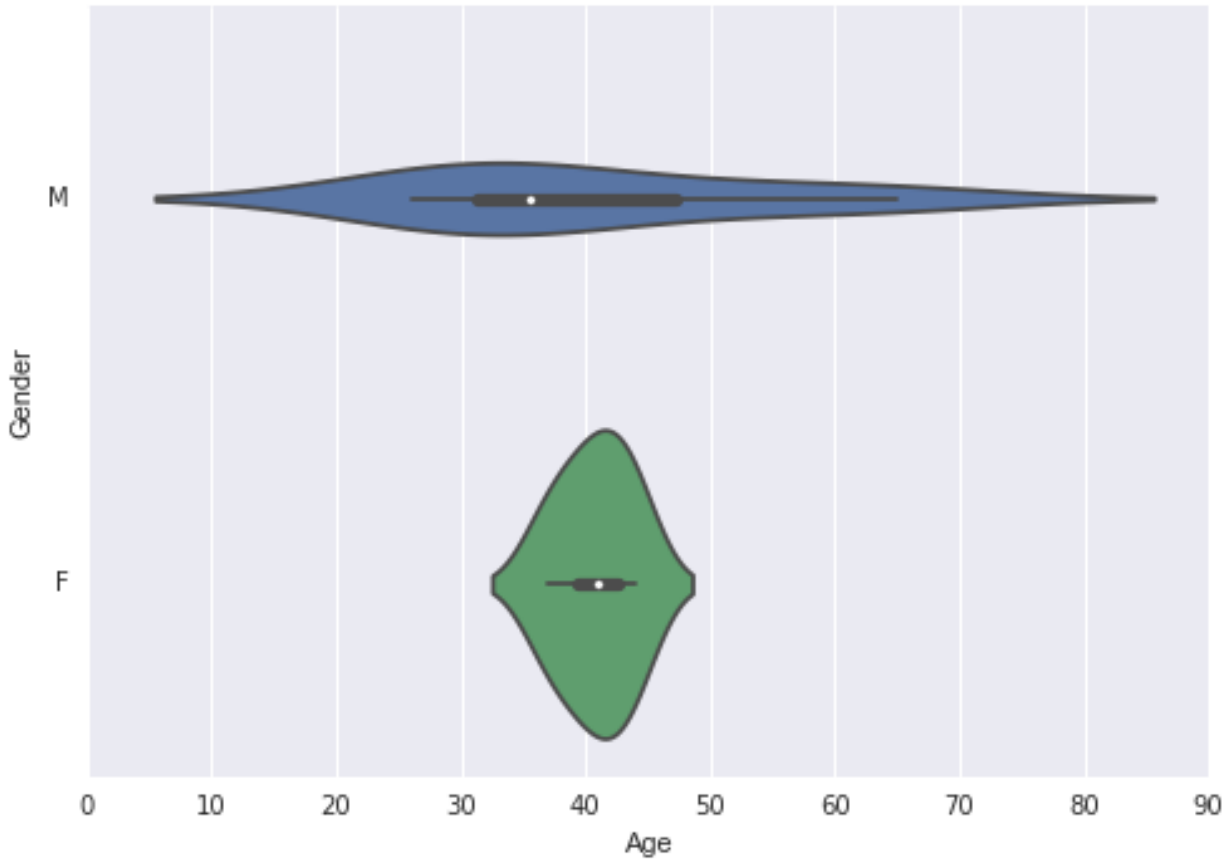
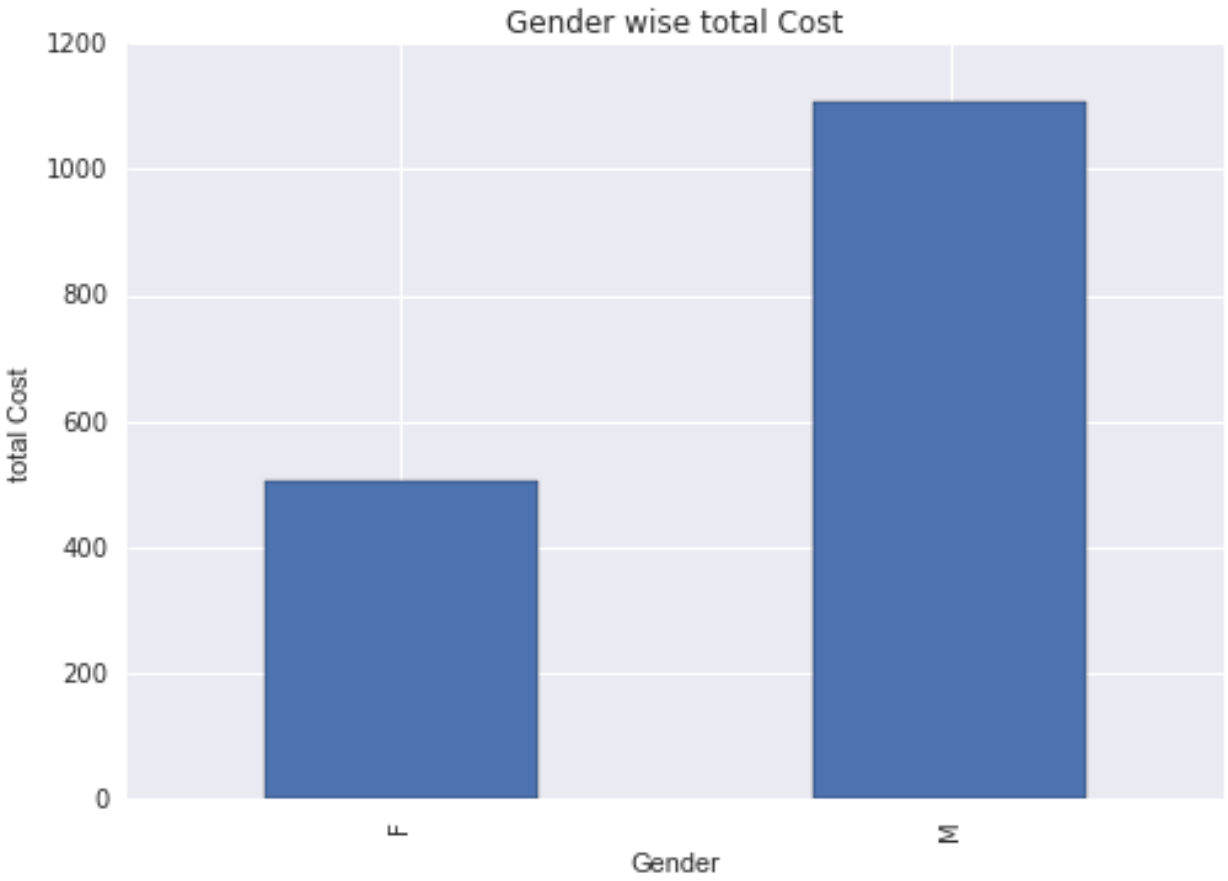
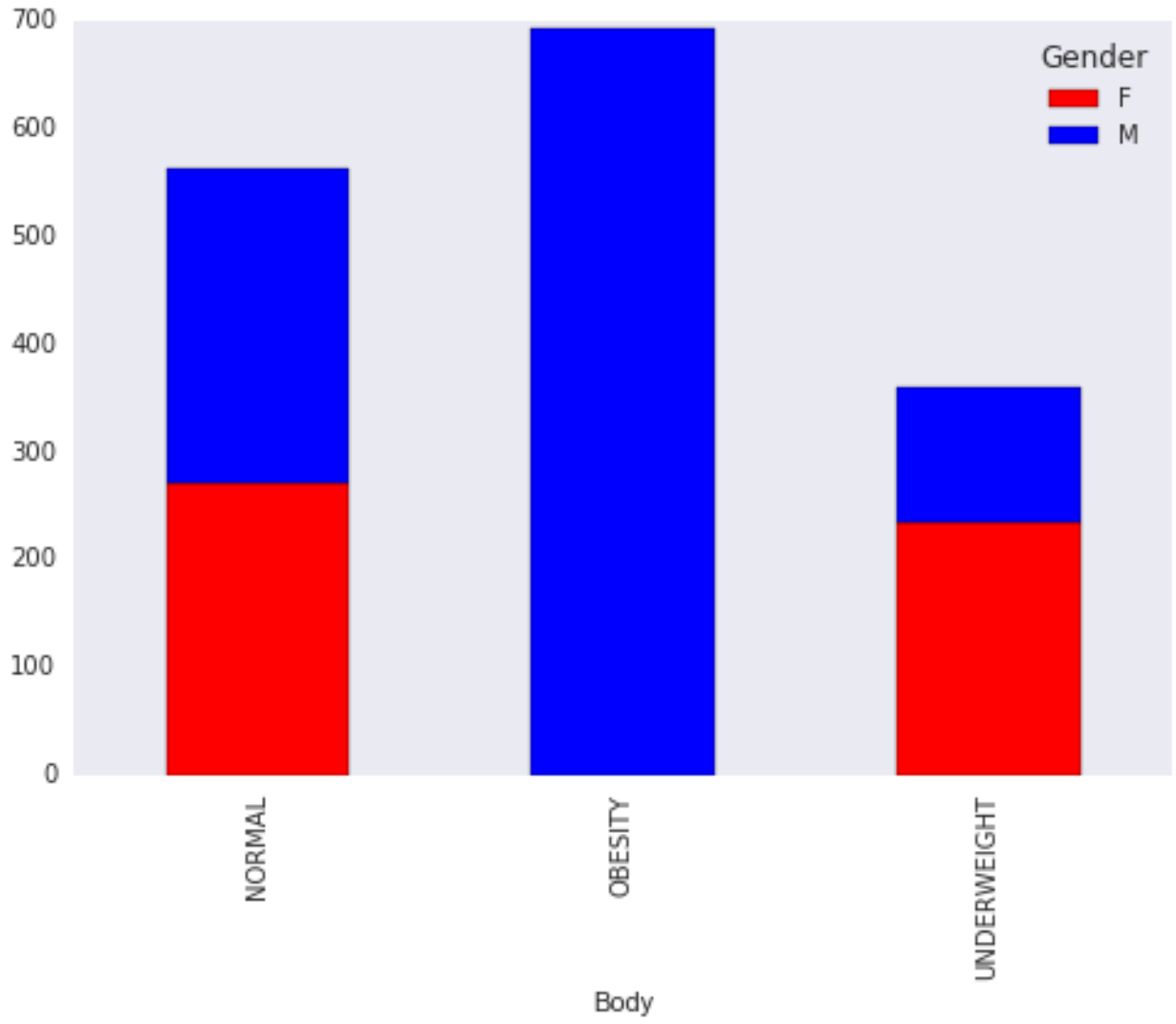
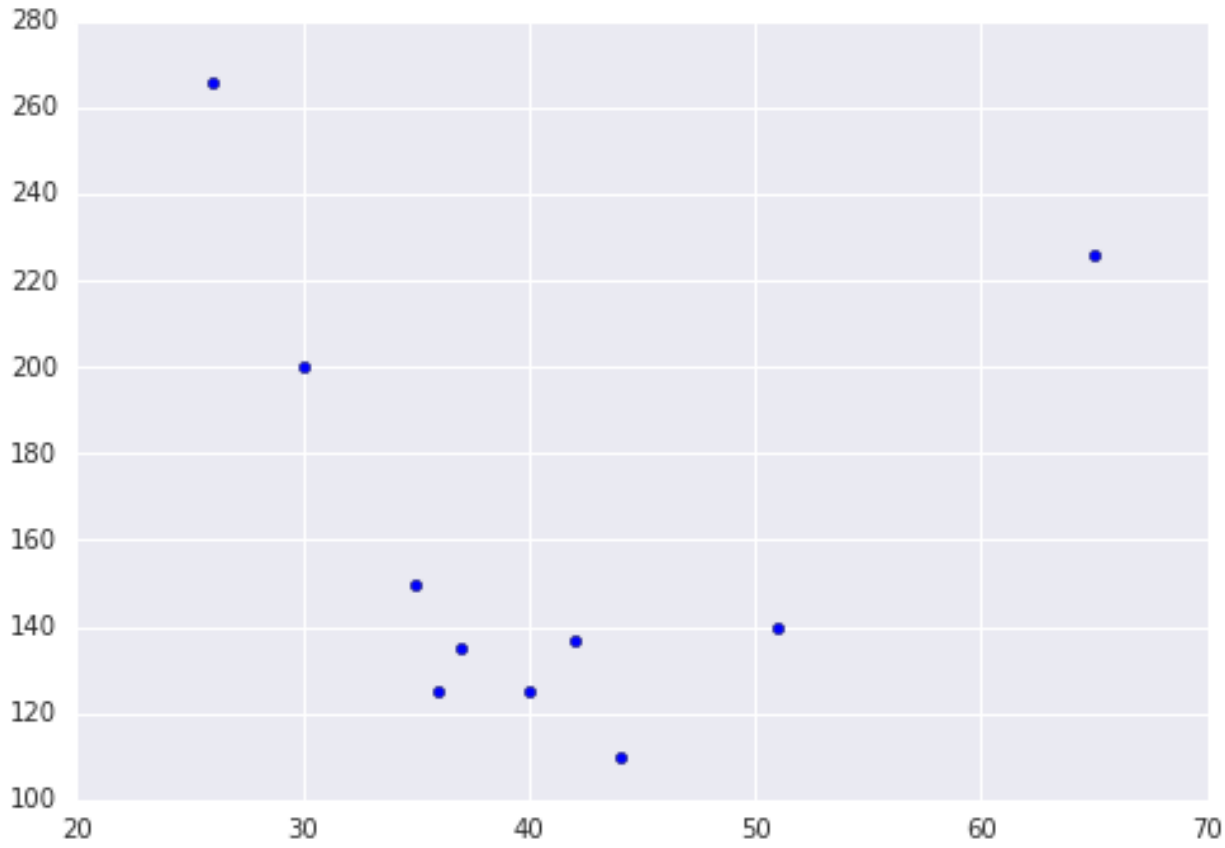


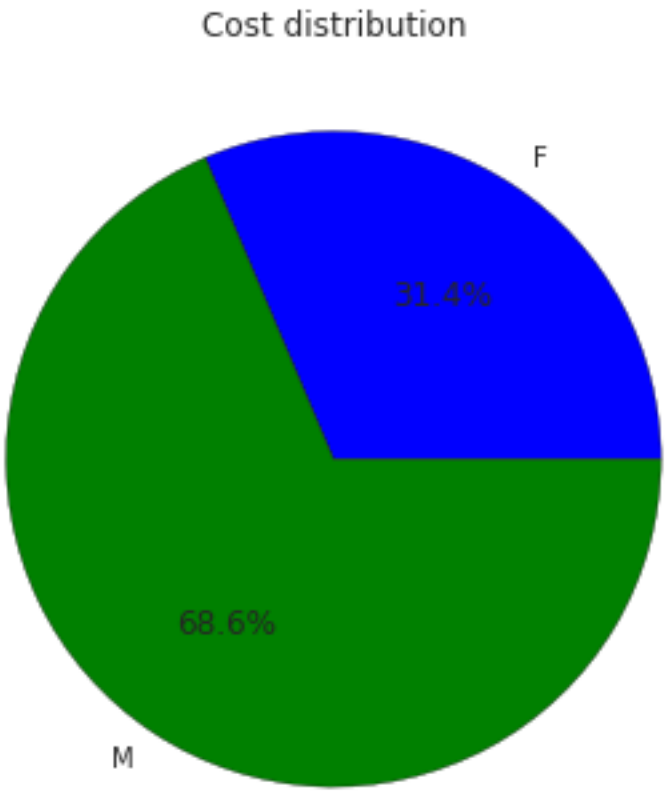
Figure 3: GENDER VS AGE

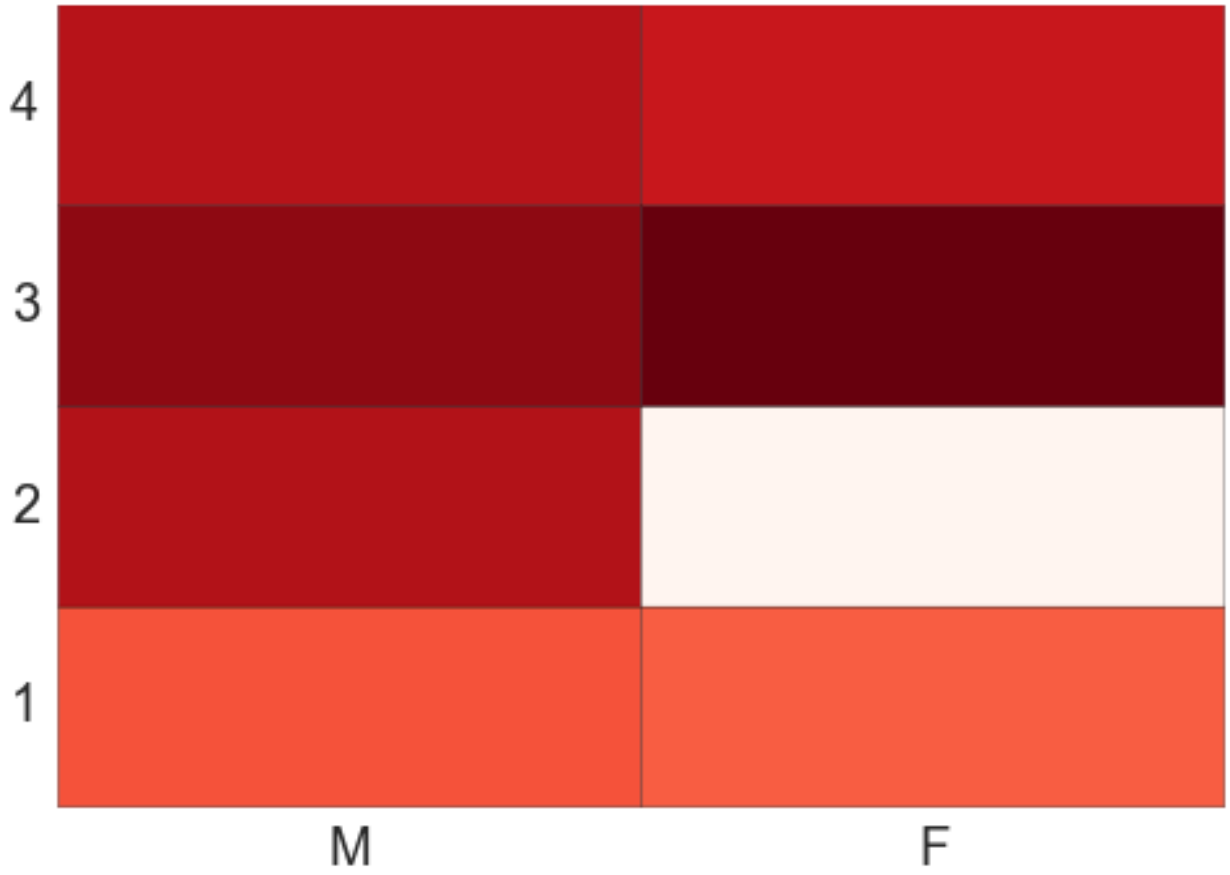

Gender wise total Cost
1200
1000
800
600
400
200
0
total Cost
F
M
Gender


Body wise total Cost
700
650
600
550
500
450
400
350
Total Cost
NORMAL
OBESITY
UNDERWEIGHT
Body


700
600
500
400
300
200
100
0
Gender
F
M
NORMAL
OBESITY
UNDERWEIGHT
Body


280
260
240
220
200
180
160
140
120
100
20
30
40
50
60
70

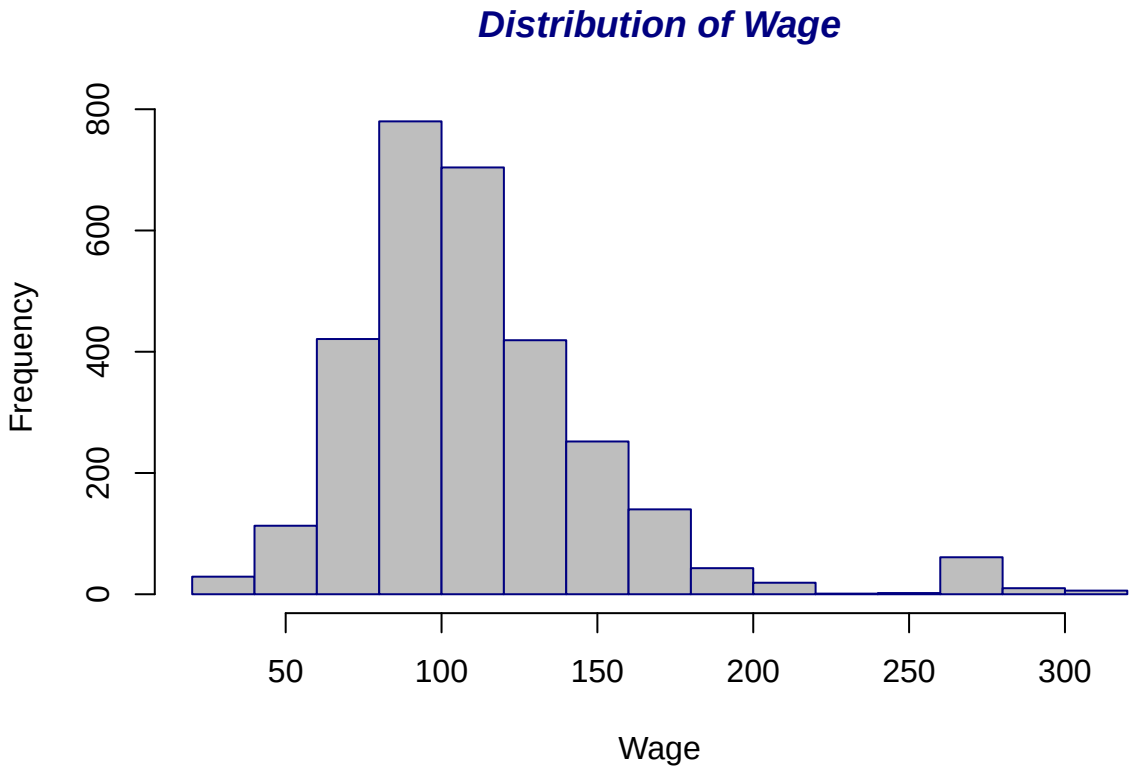

Cost distribution
F
31.4%
68.6%
M



NOW WE ARE MOVING INTO R BELOW

R

```
#install.packages("ISLR")
library(ISLR)
#View(Wage)
with(Wage, hist(wage, nclass=20, col="grey", border="navy", main="", xlab="Wage", cex=1.2))
title(main = "Distribution of Wage", cex=1.2, col.main="navy", font.main=4)
```



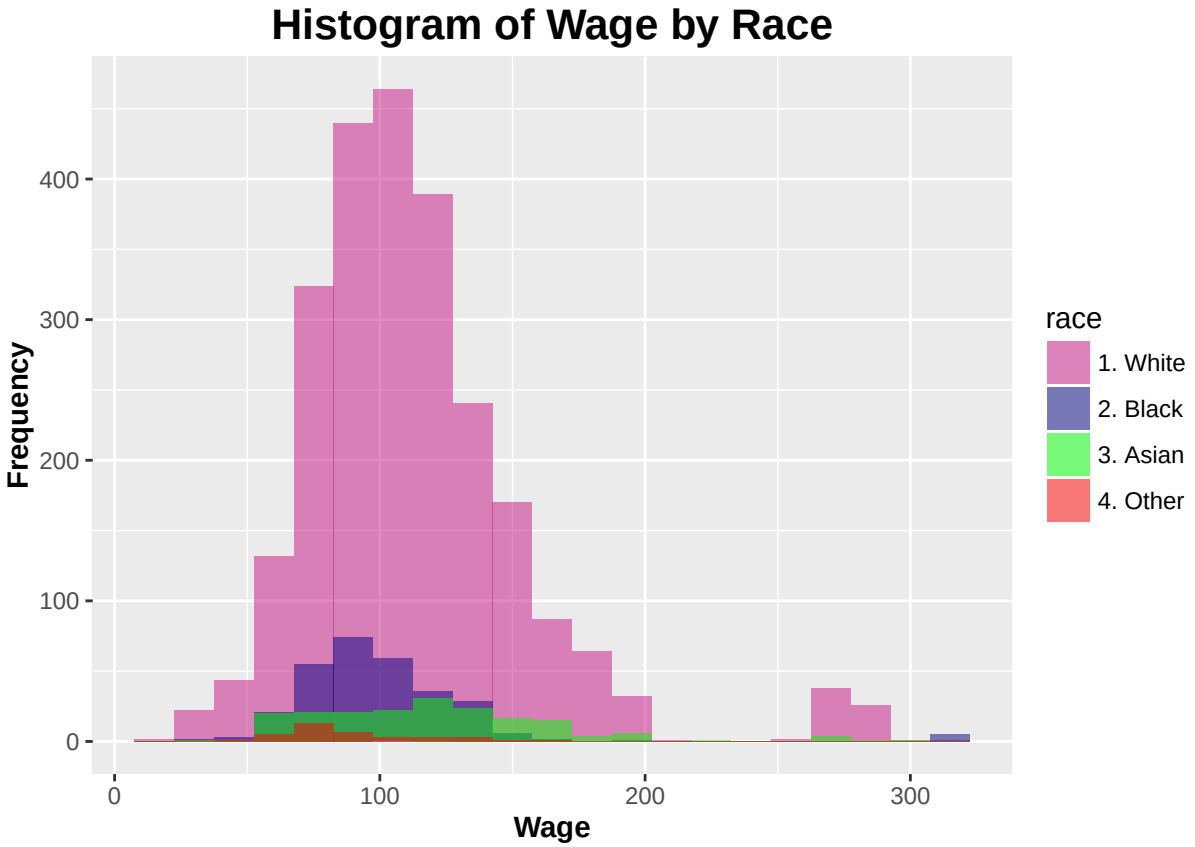
The data is mostly symmetrically distributed but there is a small bimodality in the data which is indicated by a small hump towards the right tail of the distribution

The data set contains a number of categorical variables one of which is Race. A natural question is whether the wage distribution is the same across Race. There are several libraries in R which may be used to construct histograms across levels of a categorical variables and many other sophisticated graphs and charts. One such library is ggplot2.

```
# Histogram: Wage data by Race
library(ggplot2)
library(ISLR)
p <- ggplot(data = Wage, aes(x=wage))
p <- p + geom_histogram(binwidth=25, aes(fill=race))
p <- p + scale_fill_brewer(palette="Set1")
p <- p + facet_wrap( ~ race, ncol=2)
p <- p + labs(x="Wage", y="Frequency")+ theme(axis.title =
element_text(color="black", face="bold"))
p <- p + ggtitle("Histogram of Wage by Race") + theme(plot.title =
element_text(color="black", face="bold", size=16))
p
```

Because of huge disparity among the counts of the different races, the above histograms may not be very informative. Code for an alternative visual display of the same information is shown below, followed by the plot.

```
library(ggplot2)
library(ISLR)
p1 <- ggplot(Wage, aes(x=wage,fill=race))+geom_histogram(binwidth=15,position="identity")
p11 <- p1 + scale_fill_manual(values = alpha(c("mediumvioletred","navy",
"green", "red"), 0.5))
p2 <- p11 + labs(x="Wage", y="Frequency")+ theme(axis.title =
element_text(color="black", face="bold"))
p3 <- p2 + ggtitle("Histogram of Wage by Race") + theme(plot.title =
element_text(color="black", face="bold", size=16))
p3
```

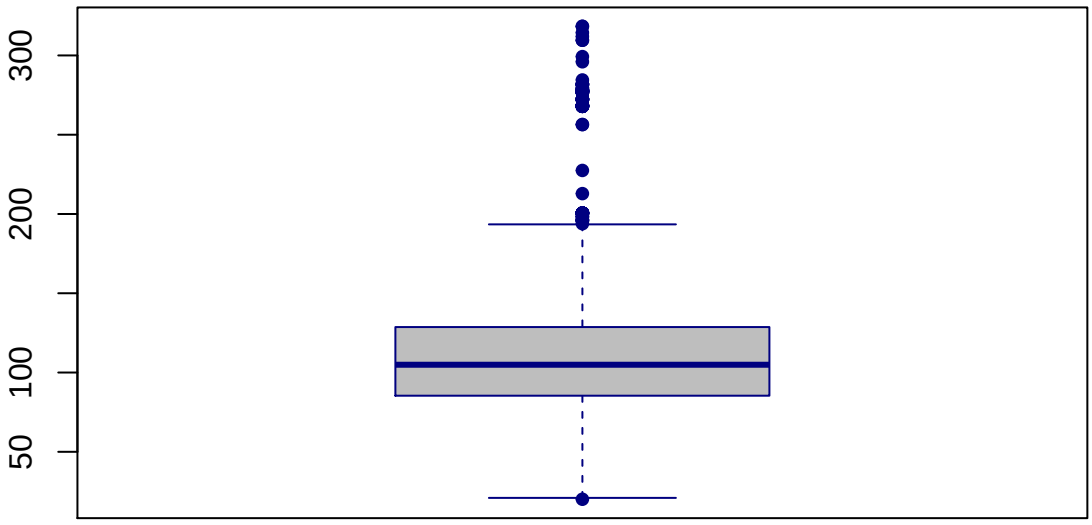


The second type of histogram also may not be the best way of presenting all the information. However further clarity is seen in the small concentration at the right tail.

BOXPLOT

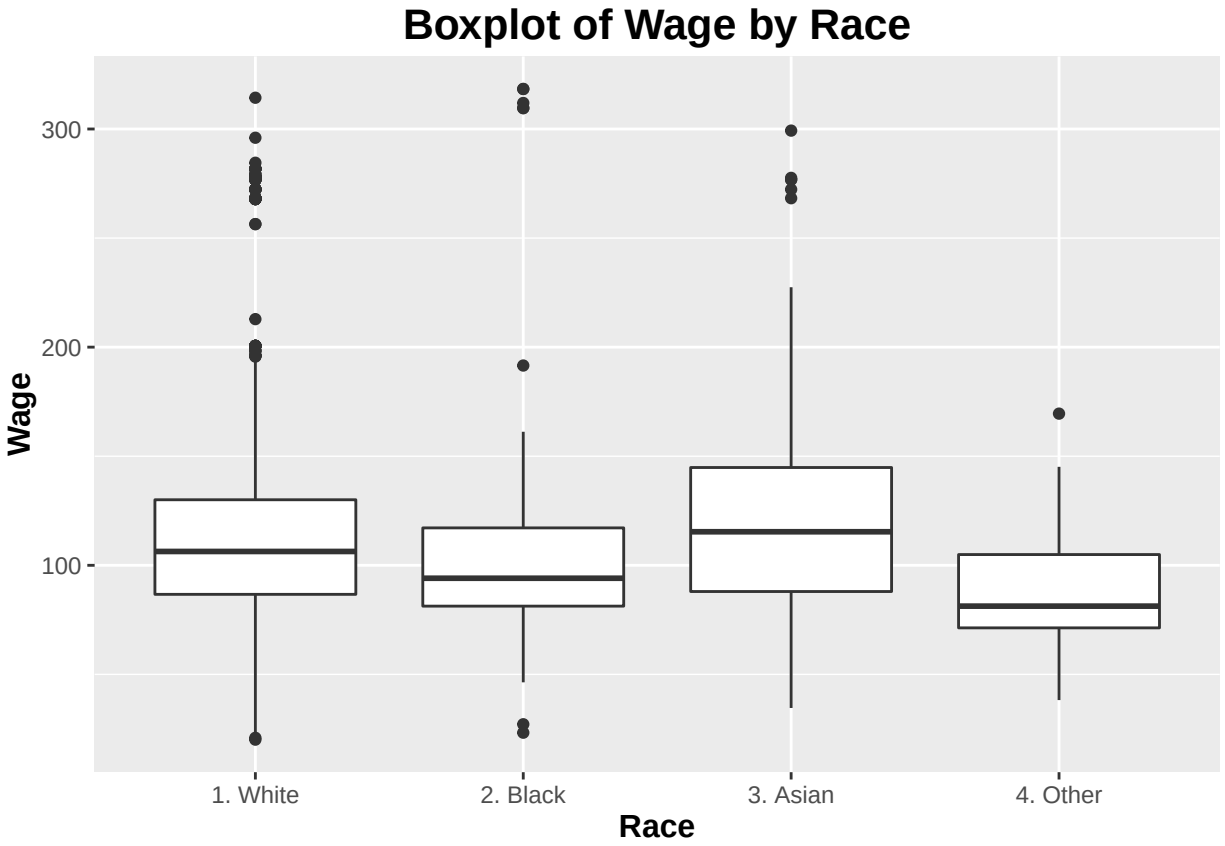
Boxplot is used to describe shape of a data distribution and especially to identify outliers. Typically an observation is an outlier if it is either less than Q1 - 1.5 IQR or greater than Q3 + 1.5 IQR, where IQR is the inter-quartile range defined as Q3 - Q1. This rule is conservative and often too many points are identified as outliers. Hence sometimes only those points outside of [Q1 - 3 IQR, Q3 + 3 IQR] are only identified as outliers

```
with(Wage,boxplot(wage,col="grey", border="navy", main="", xlab="Wage",pch = 19, cex=0.8))
title(main = "Distribution of Wage", cex=1.2, col.main="navy", font.main=4)
```



The boxplot of the Wage distribution clearly identifies many outliers. It is a reflection of the histogram depicting the distribution of Wage. The story is clearer from the boxplots drawn on the wage distribution for individual races

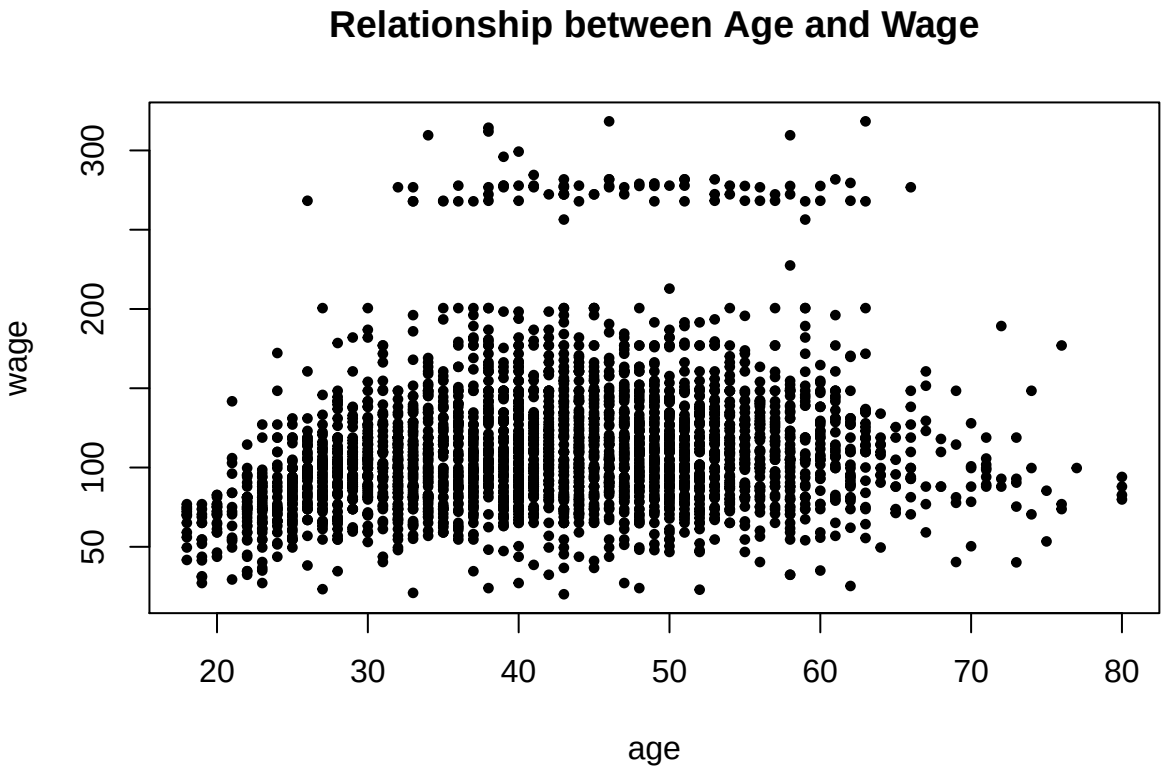
```
# Boxplot: Wage data by race
library(ggplot2)
library(ISLR)
p1 <- ggplot(Wage, aes(x=race,y=wage))+geom_boxplot()
p2 <- p1 + labs(x="Race", y="Wage")+ theme(axis.title =
element_text(color="black", face="bold", size = 12))
p3 <- p2 + ggtitle("Boxplot of Wage by Race") + theme(plot.title =
element_text(color="black", face="bold", size=16))
p3
```



Tools for Displaying Relationships Between Two Variables #SCATTERPLOT

The most standard way to visualize relation between two variables is a scatterplot. It shows the direction and strength of association between two variables, but does not quantify. Scatterplots also help to identify unusual observations. In the previous section (Section 1(b).2) a set of scatterplots are drawn for different values of the correlation coefficient. The data there is generated from a theoretical distribution of multivariate normal distribution with various values of the correlation paramater. Below is the R code used to obtain a scatterplot for these data:

```
library(ISLR)
with(Wage, plot(age, wage, pch = 19, cex=0.6))
title(main = "Relationship between Age and Wage")
```



It is clear from the scatterplot that the Wage does not seem to depend on Age very strongly. However a set of points is towards top are very different from the rest. A natural follow-up question is whether Race has any impact on the Age-Wage dependency, or the lack of it. Here is the R code and then the new plot:

```
# Scatterplot: Wage vs. Age by race
library(ISLR)
with(Wage, plot(age, wage, col = c("lightgreen","navy", "mediumvioletred",
"red")[race], pch = 19, cex=0.6))
legend(70, 310, legend=levels(Wage$race), col=c("lightgreen","navy",
"mediumvioletred", "red"), bty="n", cex=0.7, pch=19)
title(main = "Relationship between Age and Wage by Race")
```

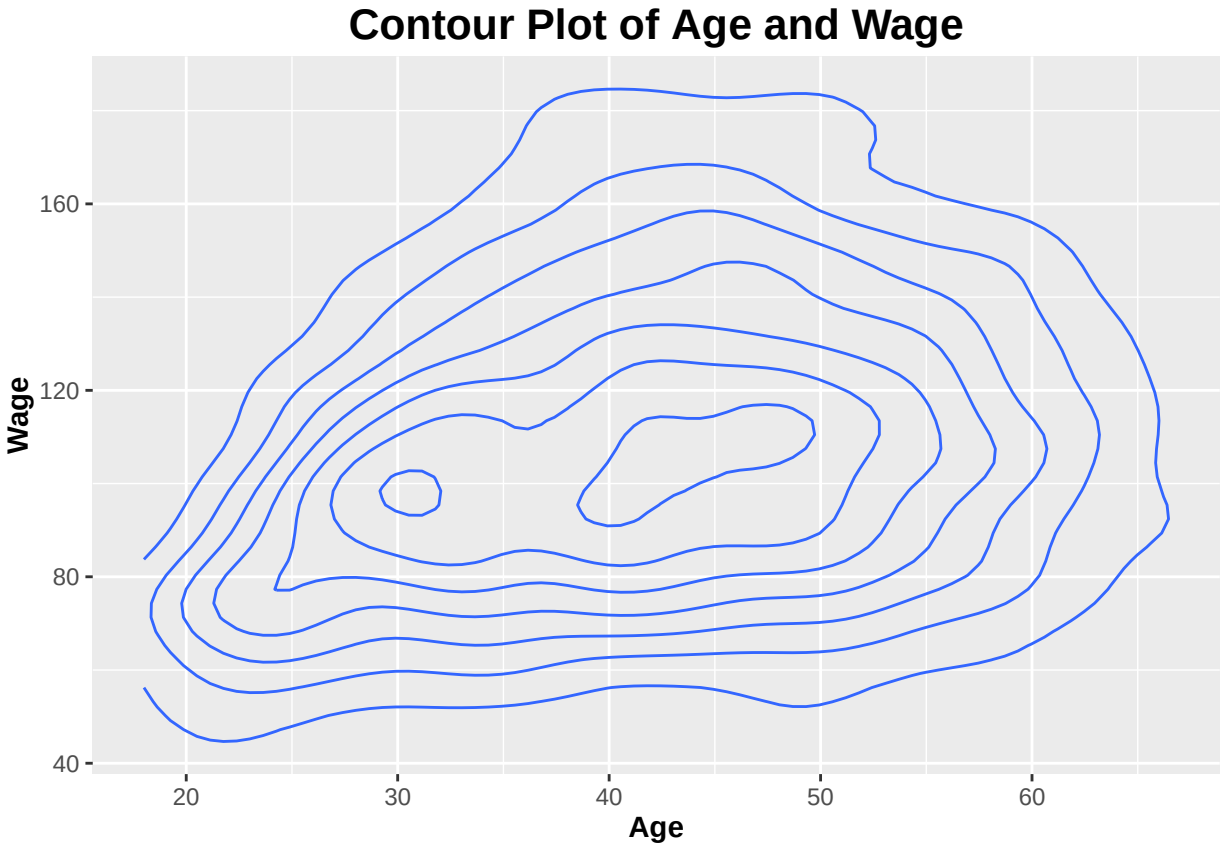
We have noted before that the disproportionately high number of Whites in the data masks the effects of the other races. There does not seem to be any association between Age and Wage, controlling for Race.

CONTOUR PLOT

This is useful when a continuous attribute is measured on a spatial grid. They partition the plane into regions of similar values. The contour lines that form the boundaries of these regions connect points with equal values. In spatial statistics contour plots have a lot of applications.

Contour plots join points of equal probability. Within the contour lines concentration of bivariate distribution is the same. One may think of the contour lines as slices of a bivariate density, sliced horizontally. Contour plots are concentric; if they are perfect circles then the random variables are independent. The more oval shaped they are, the farther they are from independence. Note the conceptual similarity in the scatterplot series in Sec 1.(b).2. In the following plot the two disjoint shapes in the interior-most part indicate that a small part of the data is very different from the rest.

```
# Contour Plot: Age and Wage
library(ggplot2)
library(ISLR)
d0 <- ggplot(Wage,aes(age, wage))+ stat_density2d()
d0 <- d0 +labs(x="Age", y="Wage")+ theme(axis.title =
element_text(color="black", face="bold"))
d0 + ggtitle("Contour Plot of Age and Wage") + theme(plot.title =
element_text(color="black", face="bold", size=16))
```

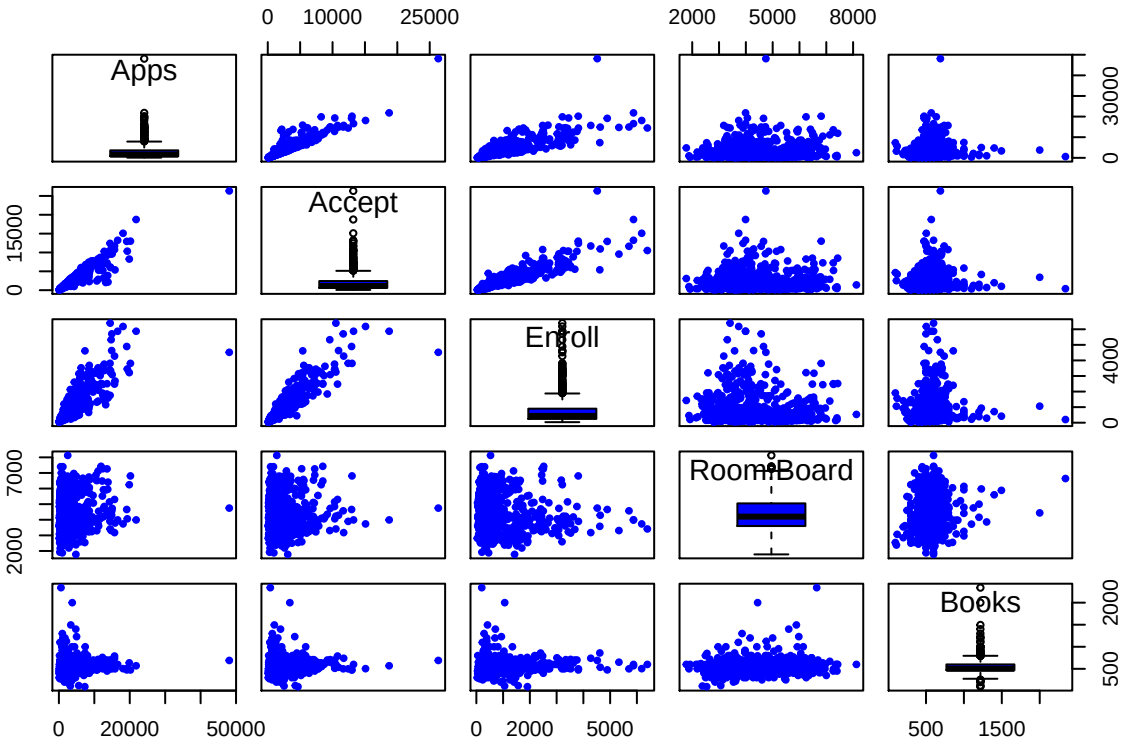


Tools for Displaying More Than Two Variables Scatterplot Matrix

Displaying more than two variables on a single scatterplot is not possible. Scatterplot matrix is one possible visualization of three or more continuous variables taken two at a time.

The data set used to display scatterplot matrix is the College data that is included in the ISLR package. A full description of the data is given in the package. Here is the R code for the scatterplot matrix that follows:

```
library(ISLR)
attach(College)
library(car)
X <- cbind(Apps, Accept, Enroll, Room.Board, Books)
scatterplotMatrix(X, diagonal=c("boxplot"), reg.line=F, smoother=F, pch=19, cex=0.6, col="blue")
title (main="Scatterplot Matrix of College Attributes", col.main="navy", font.main=4, line = 3)
```



MORE VISUALIZATIONS USING THE DATASET PROVIDED IN CLASS

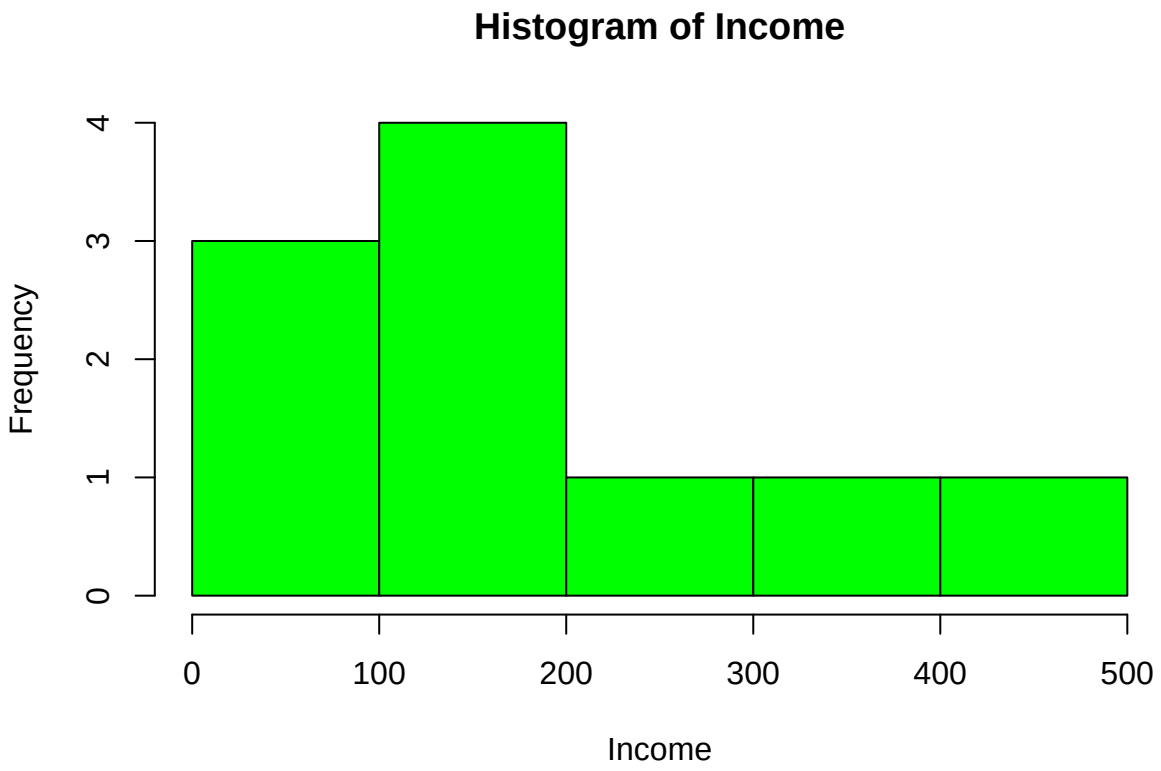
```
library(readr)
data<- read_csv("~/Downloads/Visualization637.csv")
summary(data)
```

```
##   PatientID            Gender               age            weight
##  Length:10          Length:10          Min.   :26.00   Min.   :110.0
##  Class :character   Class :character   1st Qu.:35.25   1st Qu.:127.5
##  Mode  :character   Mode  :character   Median :38.50   Median :138.5
##                                        Mean   :40.60   Mean   :161.4
##                                        3rd Qu.:43.50   3rd Qu.:187.5
##                                        Max.   :65.00   Max.   :266.0
##      Body               Income
##  Length:10          Min.   : 75.0
##  Class :character   1st Qu.: 97.5
##  Mode  :character   Median :175.0
##                     Mean   :194.0
##                     3rd Qu.:218.8
##                     Max.   :450.0
```

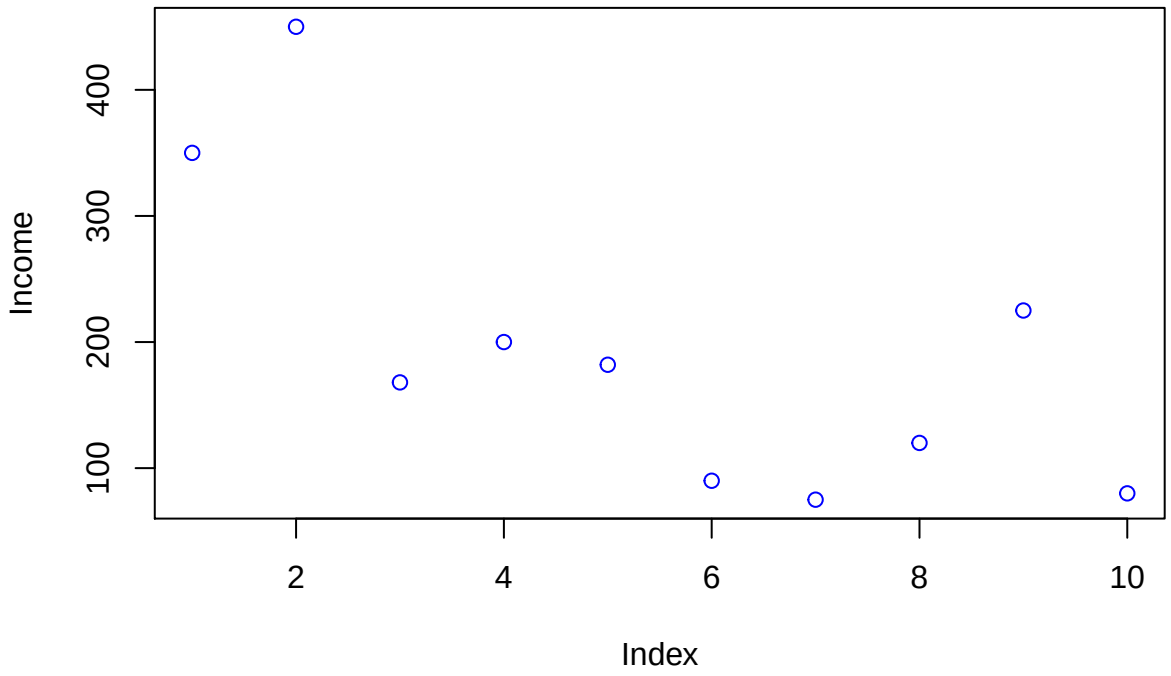
```
names(data)
```

```
## [1] "PatientID" "Gender"    "age"       "weight"    "Body"      "Income"
```

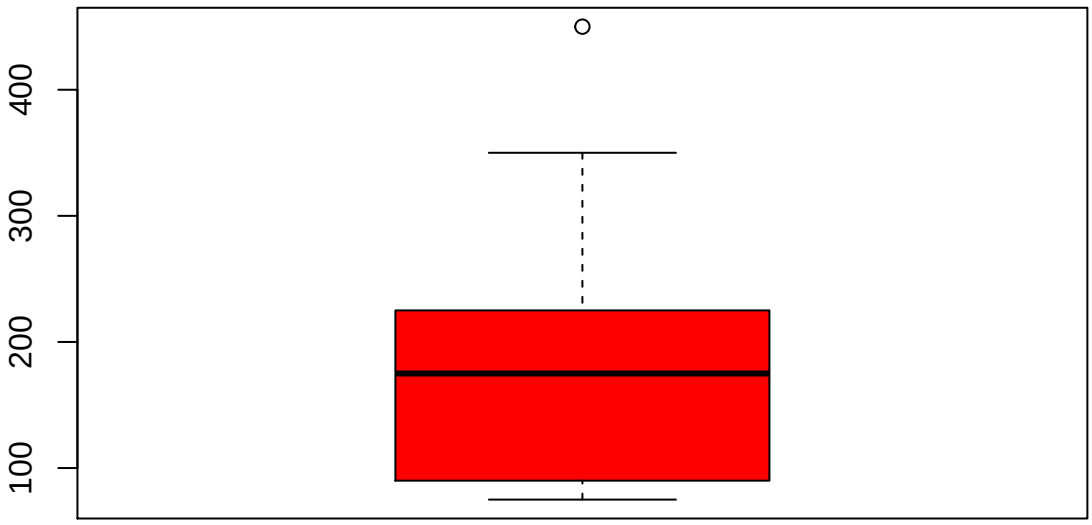
```r
attach(data)
hist(Income,col='green')
```



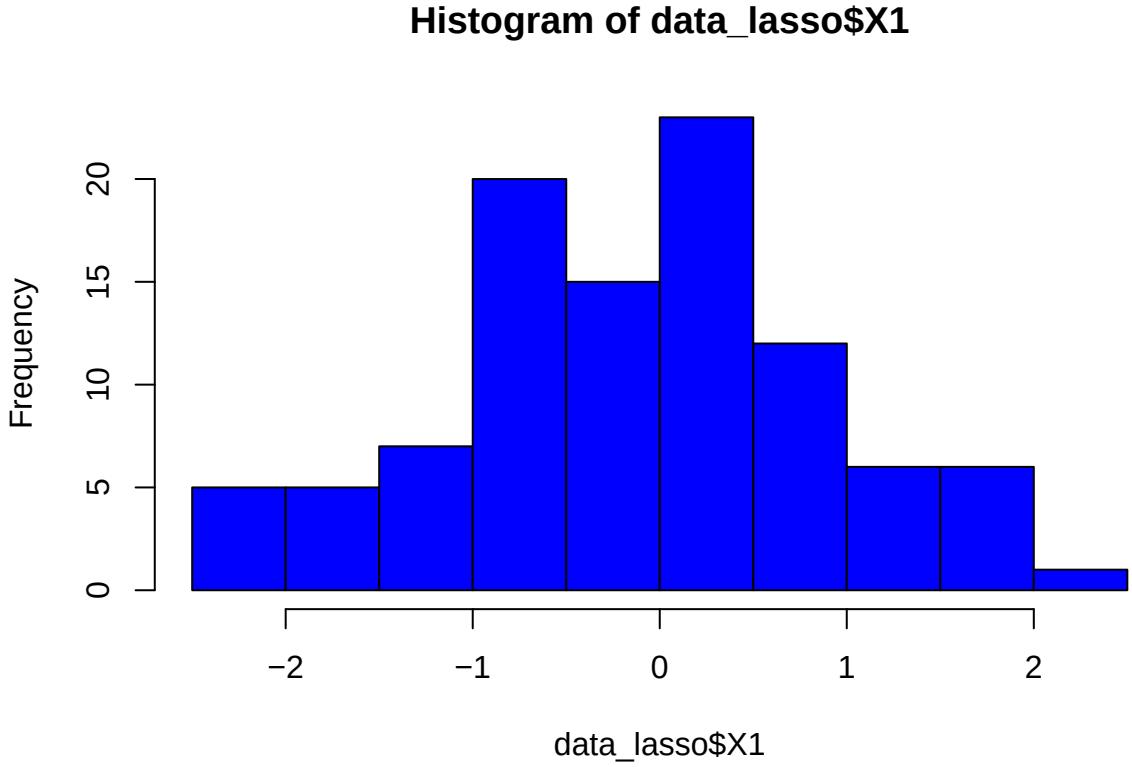

```r
plot(Income,col='blue')
```



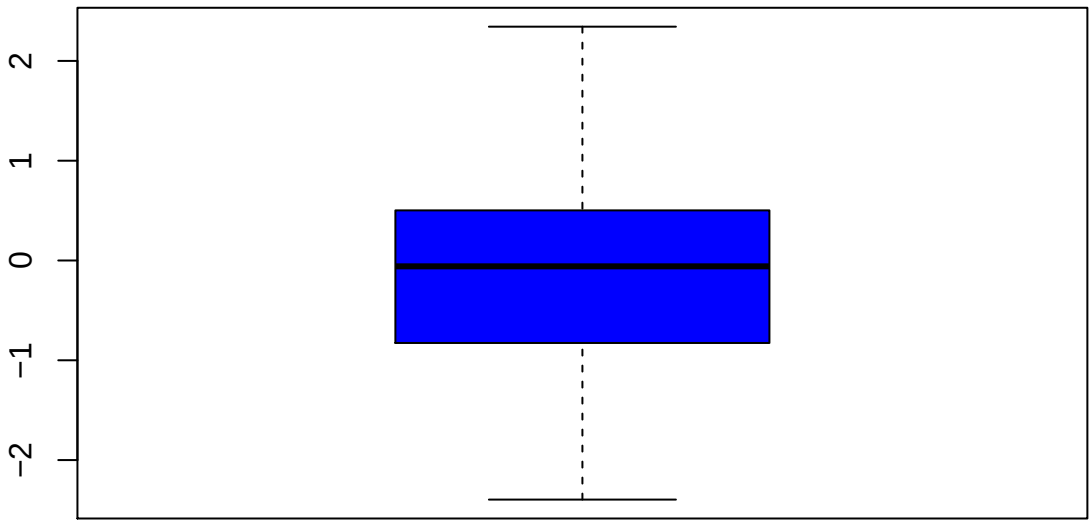

```r
boxplot(Income,col='red')
```



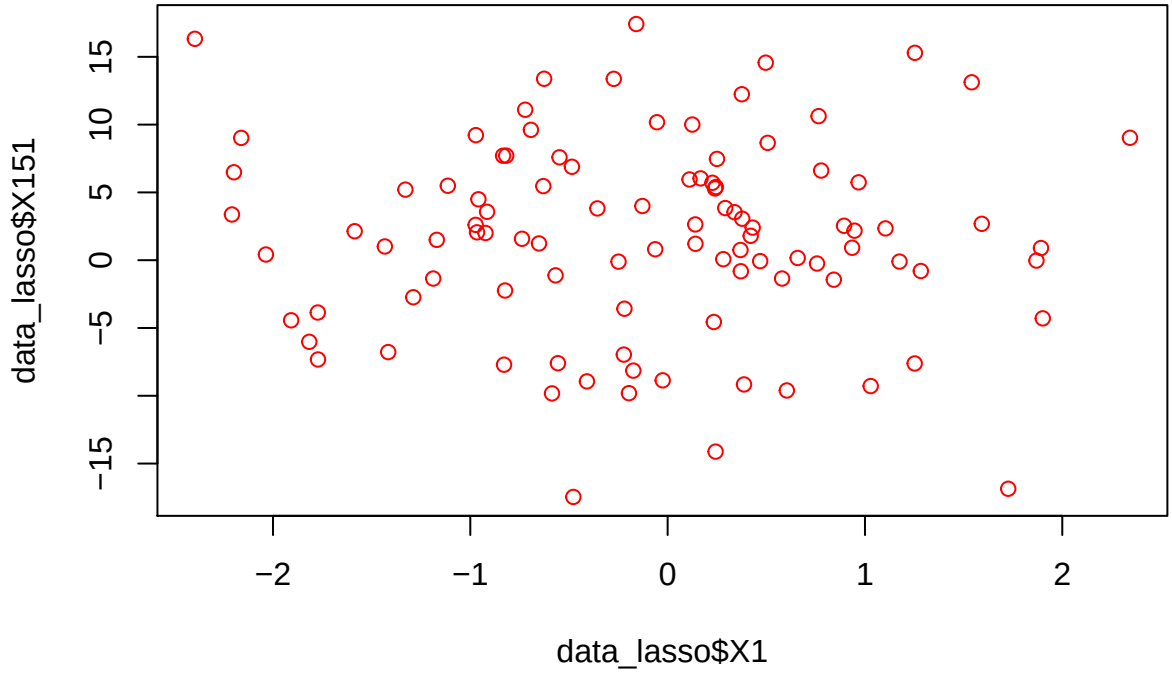

```r
library(readr)
data_lasso <- read_csv("~/Downloads/hw3_csv/data_lasso.csv",col_names = FALSE)
#View(data_lasso)
#NICE TO SEE BUIT TOO LONG TO PRINT
#str(data_lasso)
#summary(data_lasso)
hist(data_lasso$X1,col='blue')
```




```r
boxplot(data_lasso$X1,col='blue')
```




```
plot(data_lasso$X1,data_lasso$X151,col='red')
```




```
#
library(RColorBrewer)
#
data(VADeaths)
names(data)
```

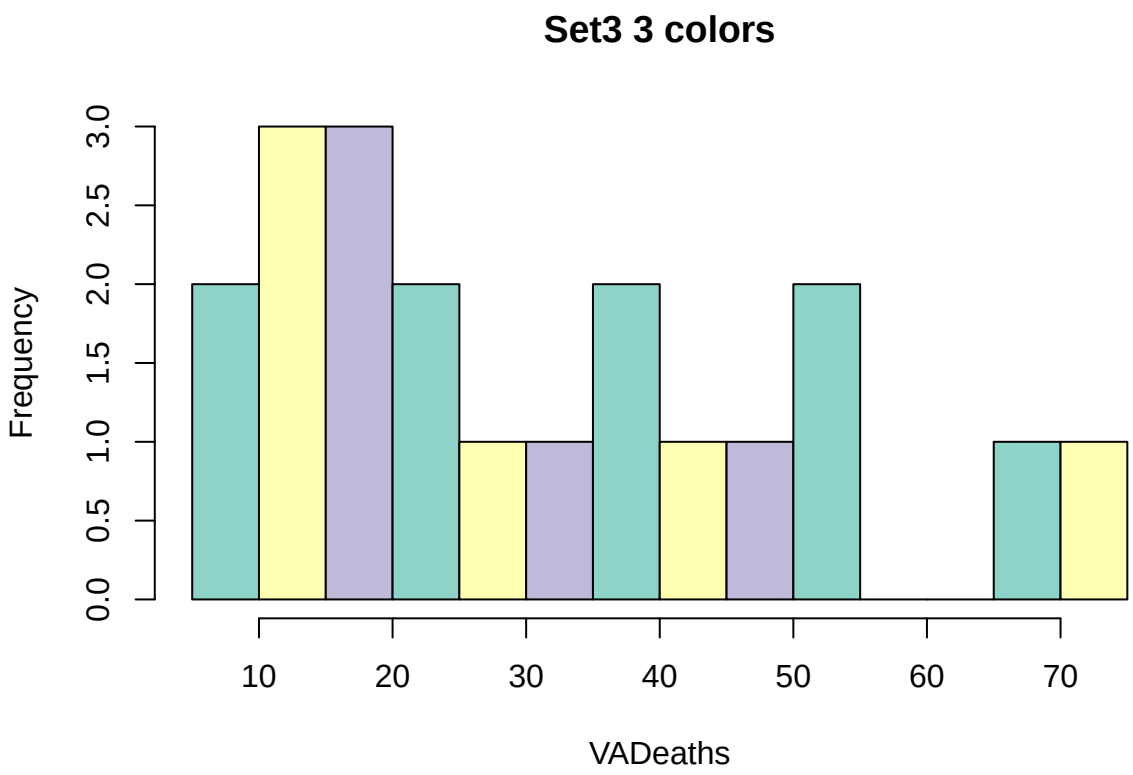
```
## [1] "PatientID" "Gender"    "age"       "weight"    "Body"      "Income"
```

```
summary(data)
```

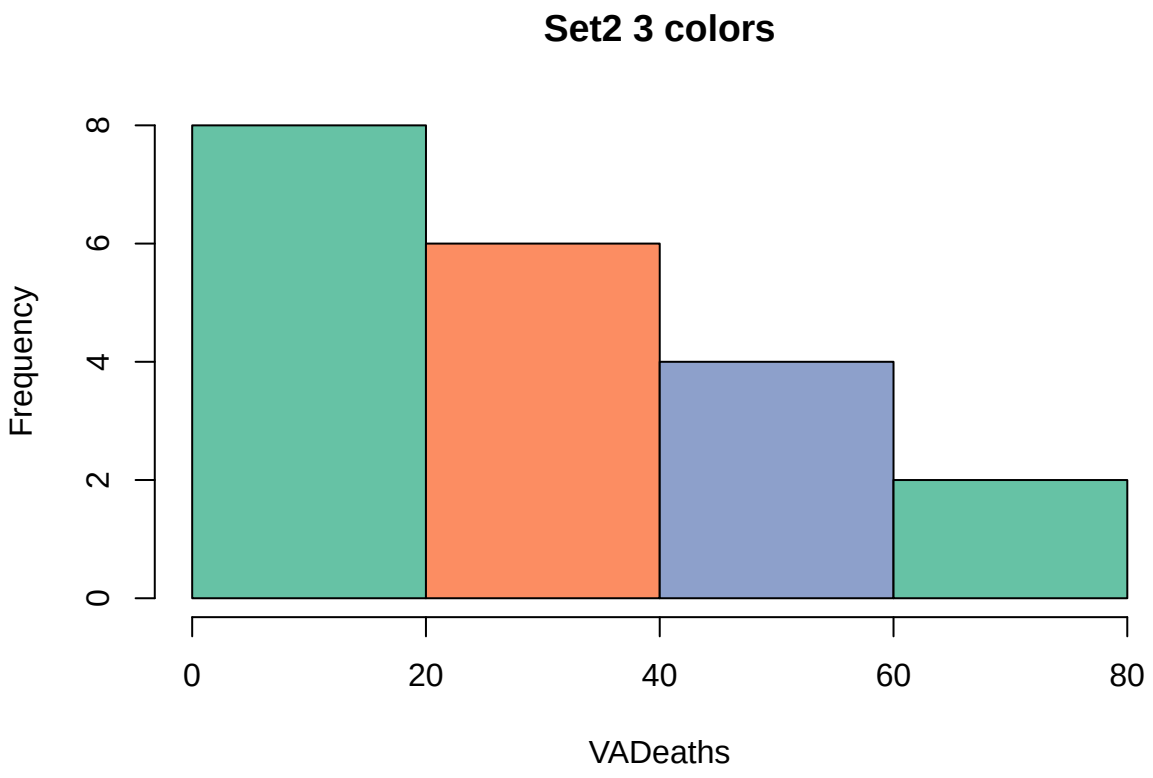
```
##   PatientID            Gender                age            weight
##  Length:10          Length:10          Min.   :26.00   Min.   :110.0
##  Class :character   Class :character   1st Qu.:35.25   1st Qu.:127.5
##  Mode  :character   Mode  :character   Median :38.50   Median :138.5
##                                        Mean   :40.60   Mean   :161.4
```

```
##                                      3rd Qu.:43.50   3rd Qu.:187.5
##                                      Max.   :65.00   Max.   :266.0
##      Body               Income
##  Length:10          Min.   : 75.0
##  Class :character   1st Qu.: 97.5
##  Mode  :character   Median :175.0
##                     Mean   :194.0
##                     3rd Qu.:218.8
##                     Max.   :450.0
```

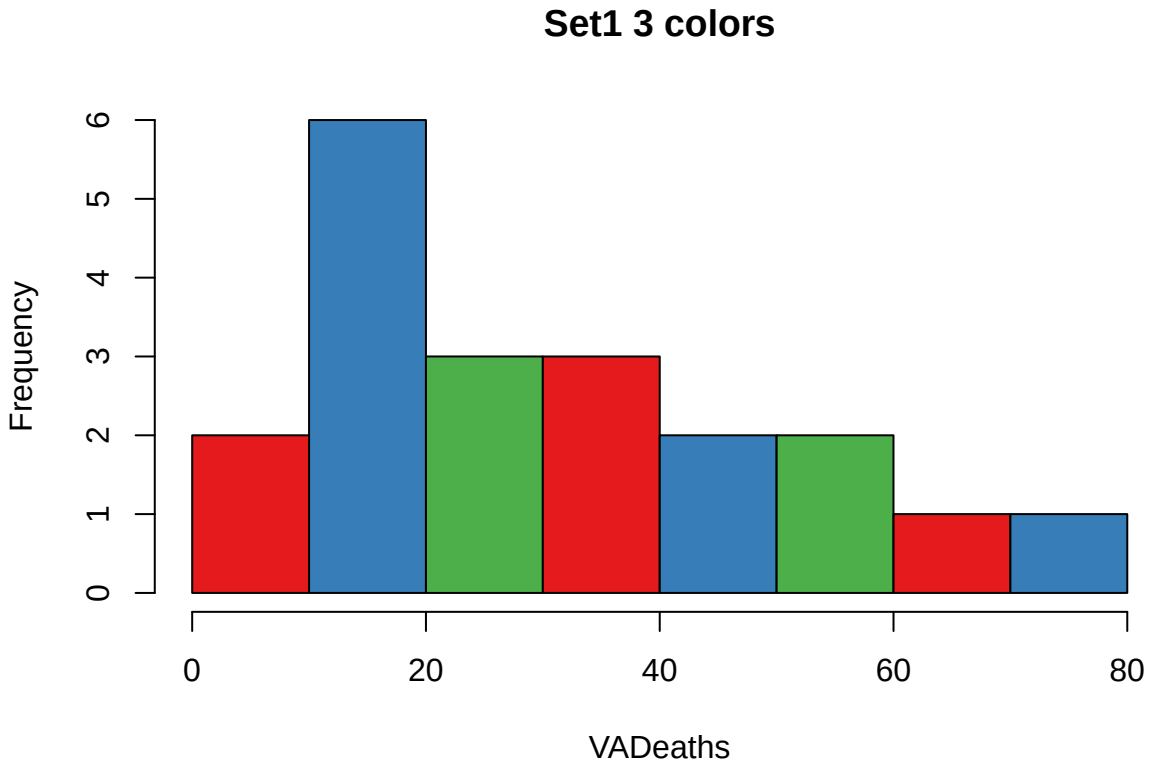
```
#par(mfrow=c(2,3))
hist(VADeaths,breaks=10, col=brewer.pal(3,"Set3"),main="Set3 3 colors")
```



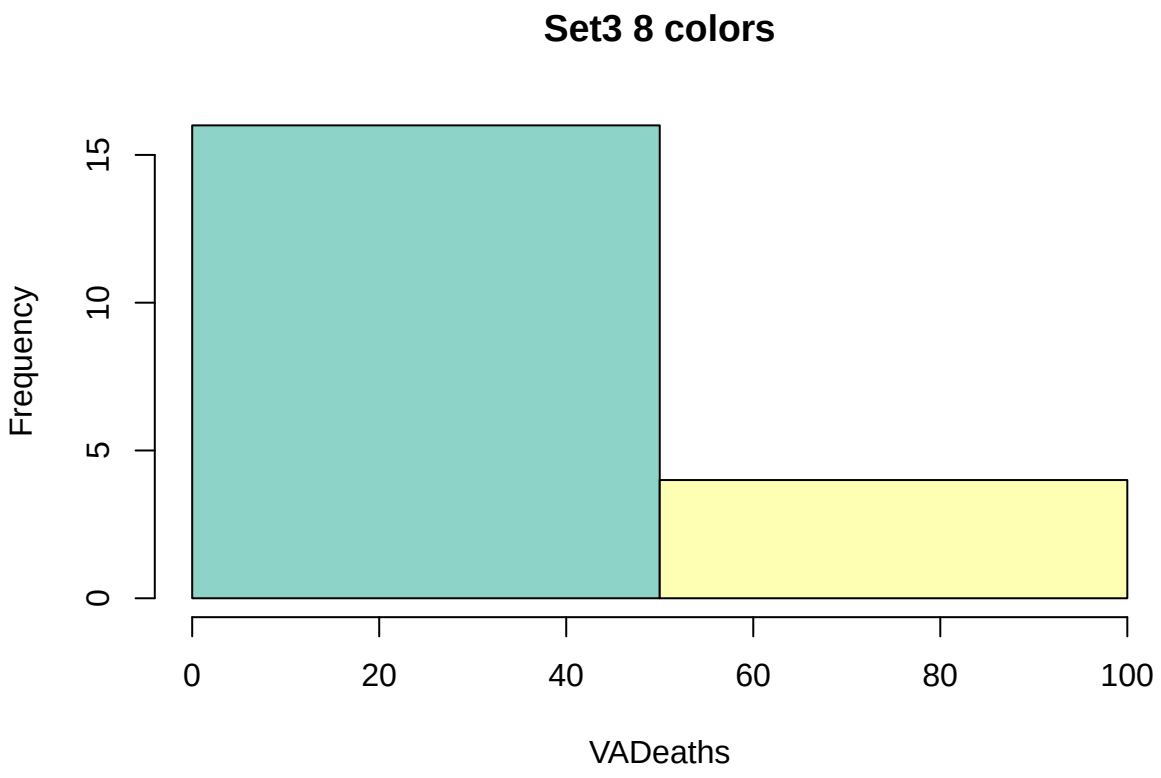

```
hist(VADeaths,breaks=3 ,col=brewer.pal(3,"Set2"),main="Set2 3 colors")
```



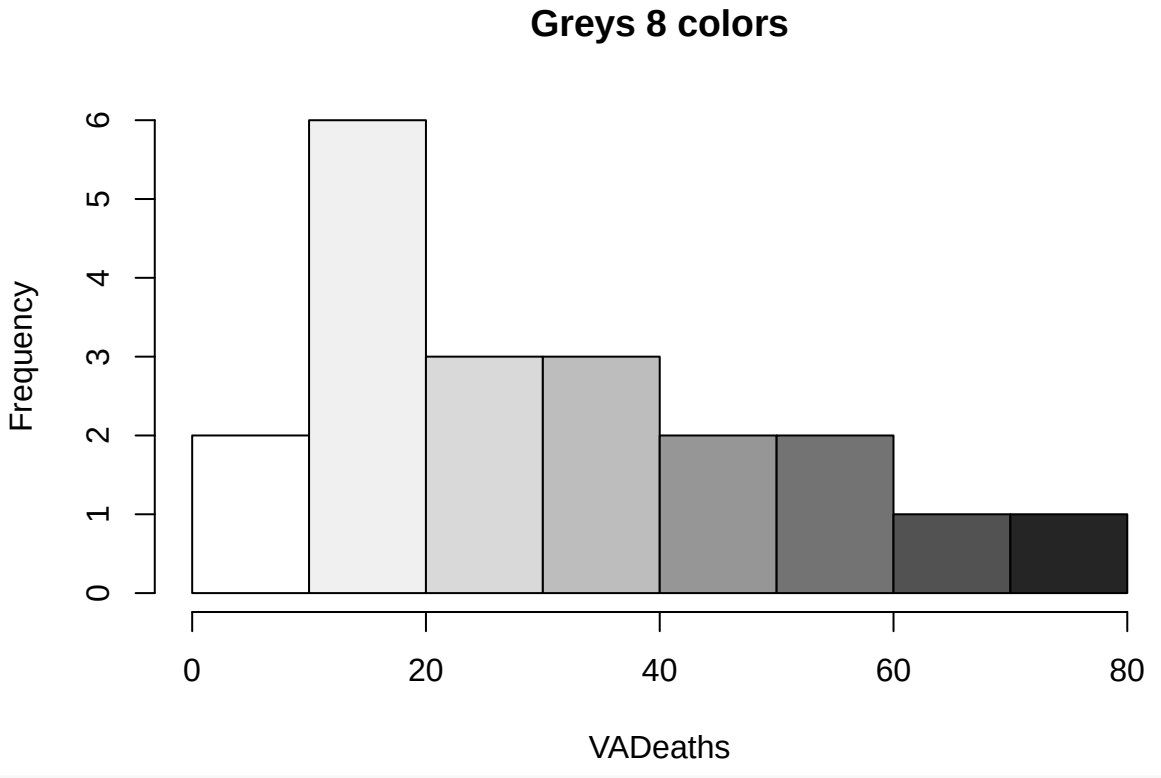

```
hist(VADeaths,breaks=7, col=brewer.pal(3,"Set1"),main="Set1 3 colors")
```



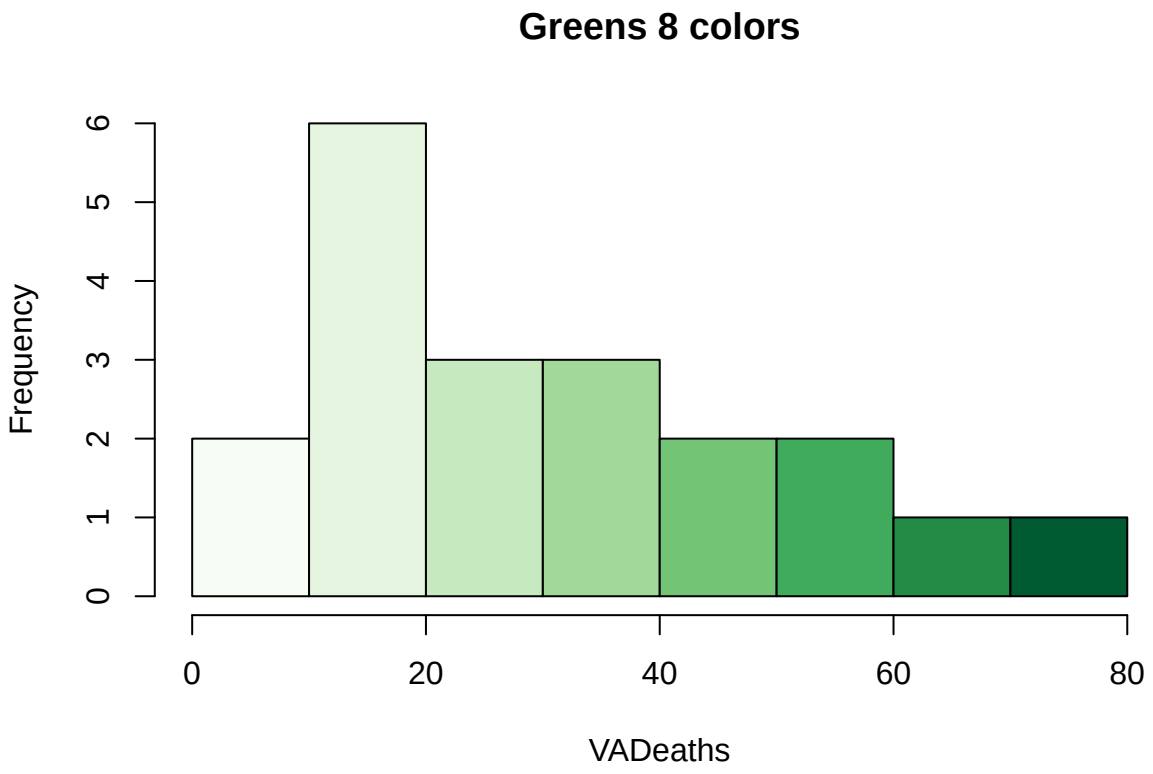

```
hist(VADeaths,,breaks= 2, col=brewer.pal(8,"Set3"),main="Set3 8 colors")
```



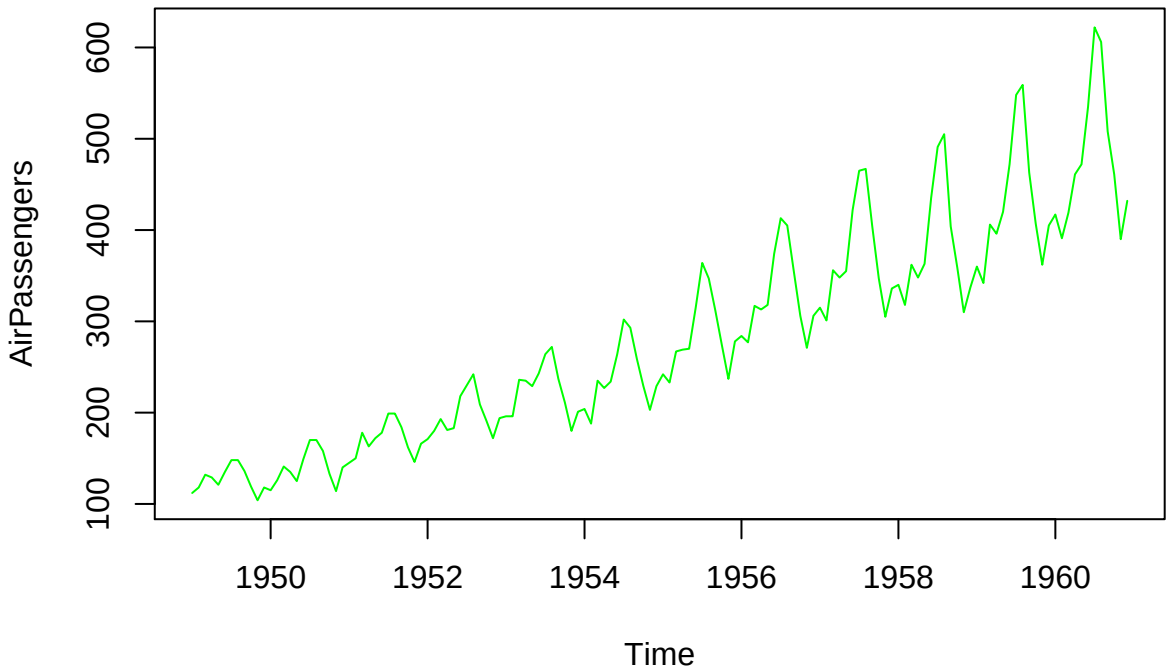

```
hist(VADeaths,col=brewer.pal(8,"Greys"),main="Greys 8 colors")
```



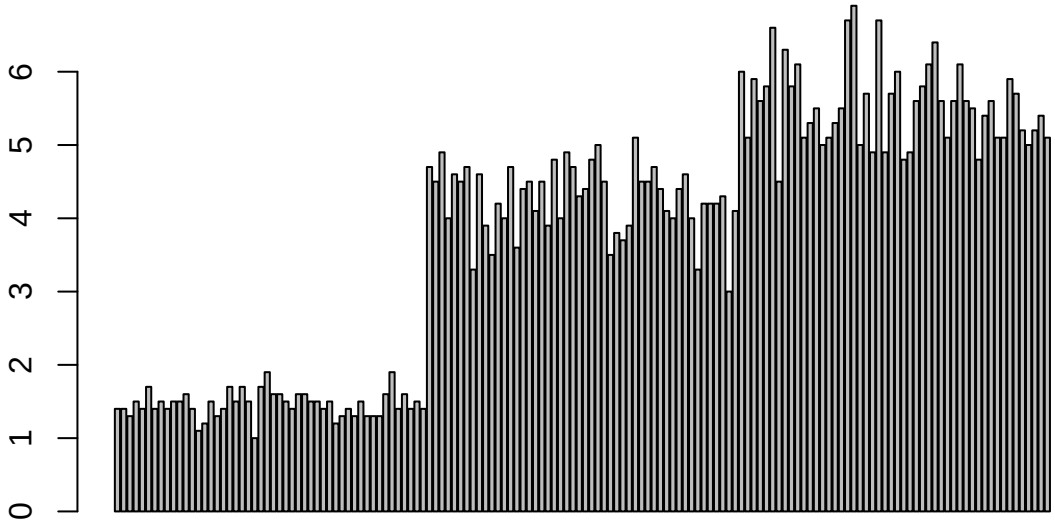

```
hist(VADeaths,col=brewer.pal(8,"Greens"),main="Greens 8 colors")
```



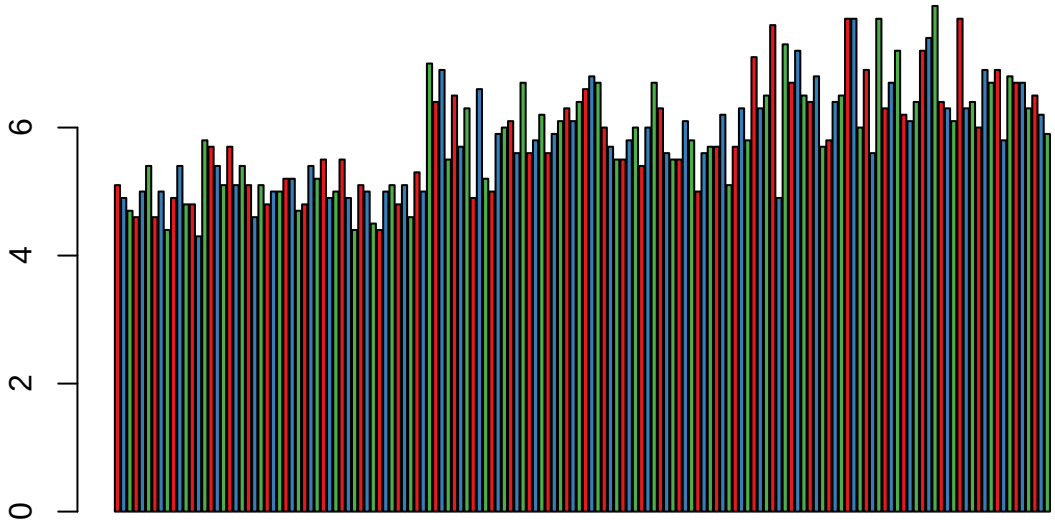

```
par(mfrow=c(1,1))
plot(AirPassengers,type="l",col='green')  #Simple Line Plot
```



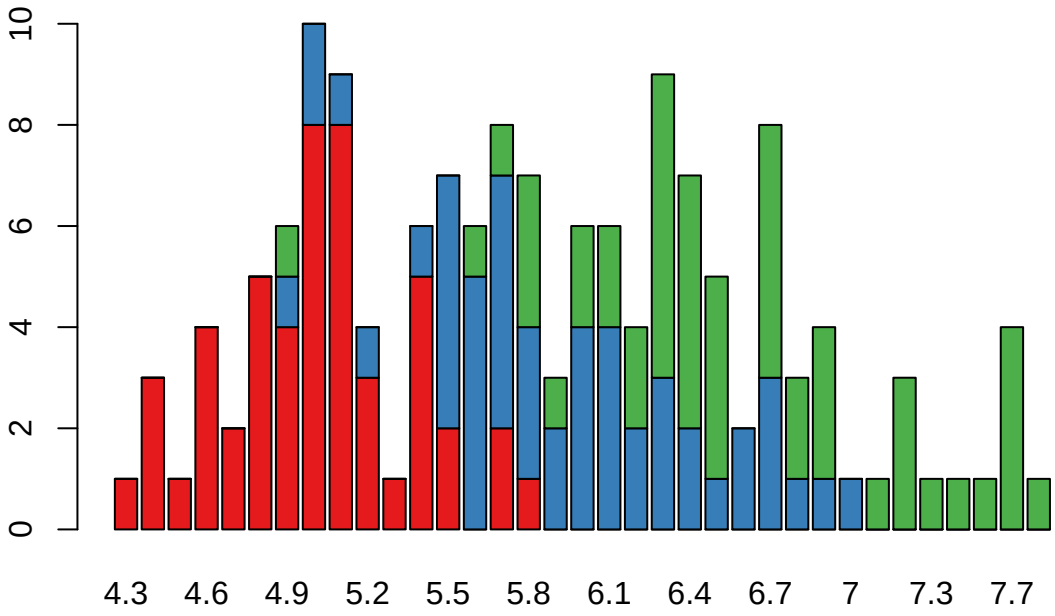

```
barplot(iris$Petal.Length) #Creating simple Bar Graph
```



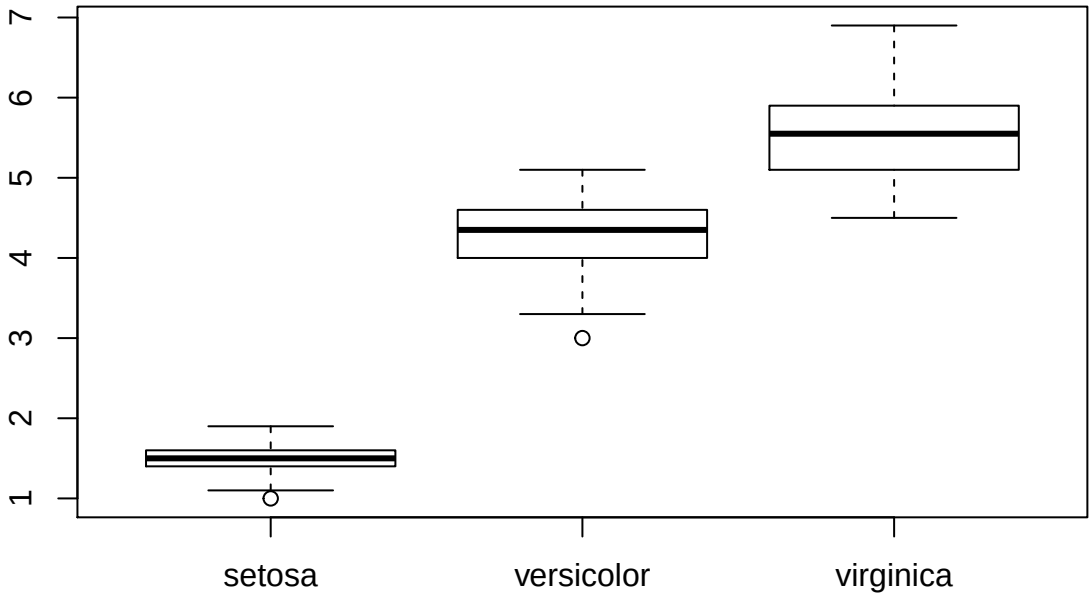

```
barplot(iris$Sepal.Length,col  = brewer.pal(3,"Set1"))
```



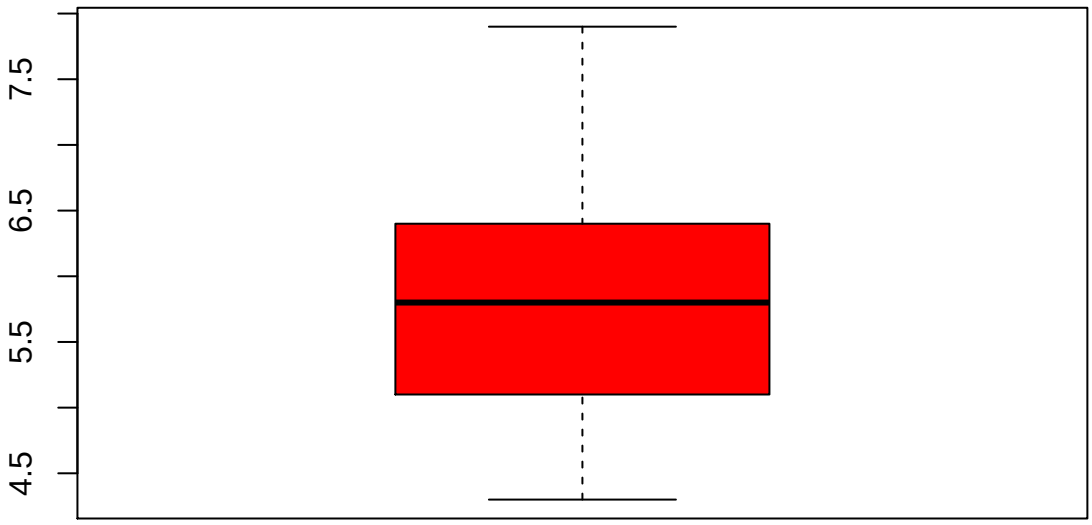

```
barplot(table(iris$Species,iris$Sepal.Length),col  = brewer.pal(3,"Set1")) #Stacked Plot
```



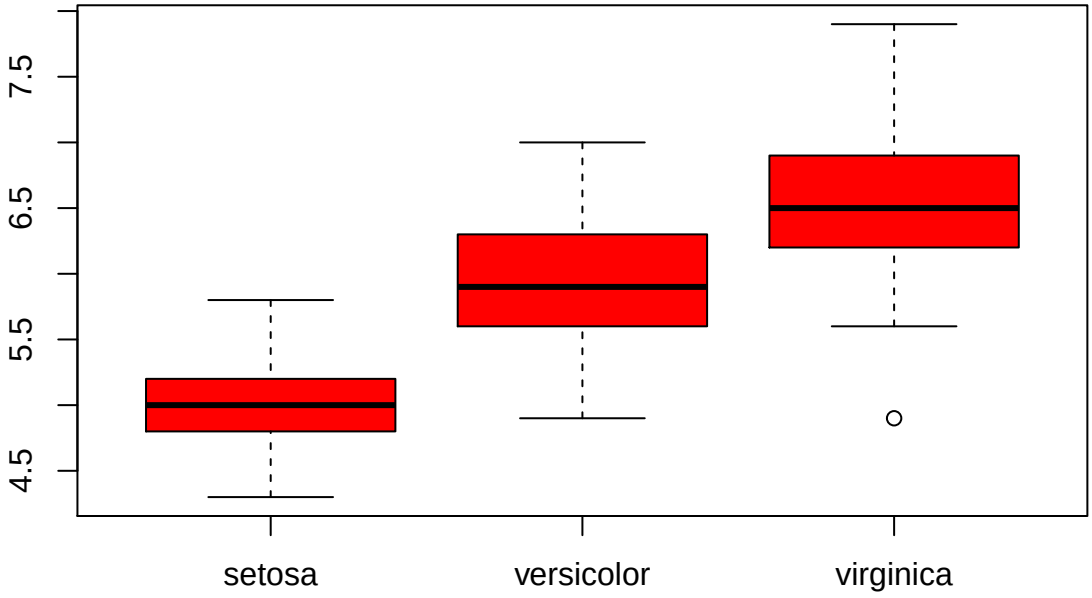

```
boxplot(iris$Petal.Length~iris$Species) #Creating Box Plot between two variable
```



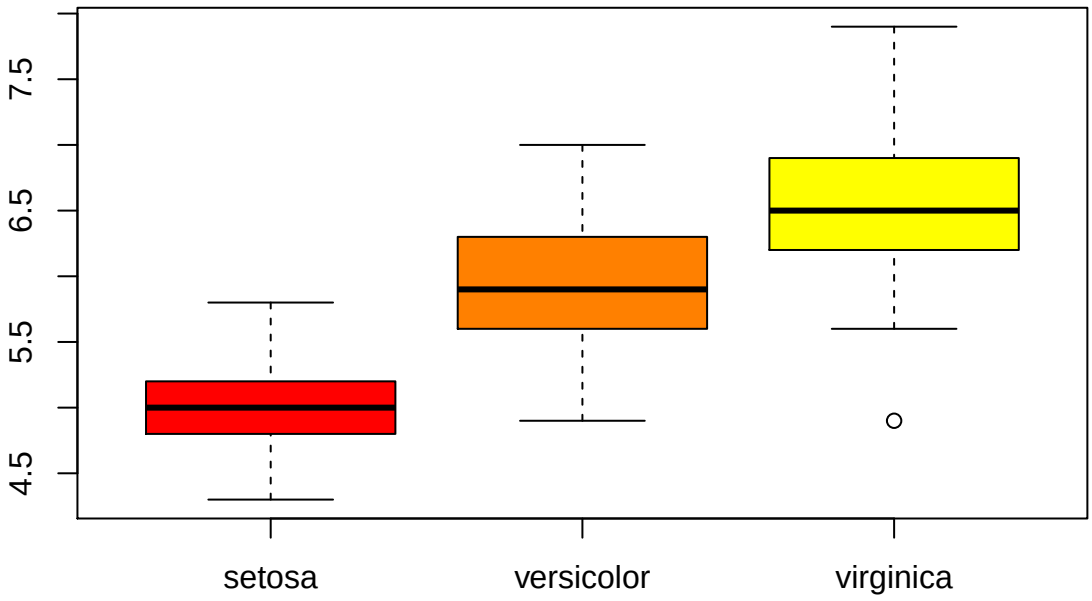

```
data(iris)
#par(mfrow=c(2,2))
boxplot(iris$Sepal.Length,col="red")
```



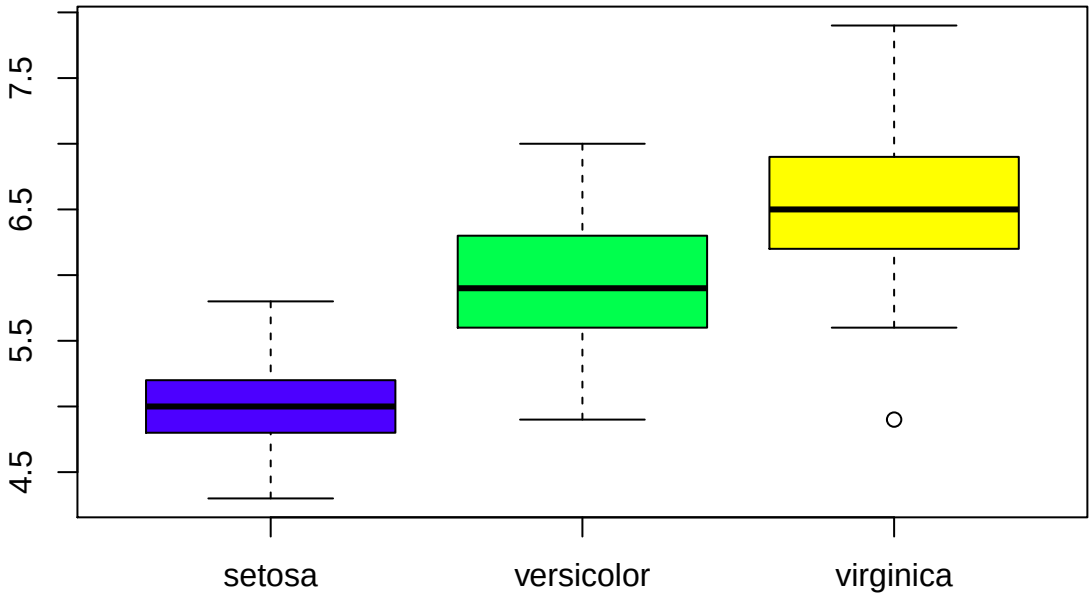

```
boxplot(iris$Sepal.Length~iris$Species,col="red")
```



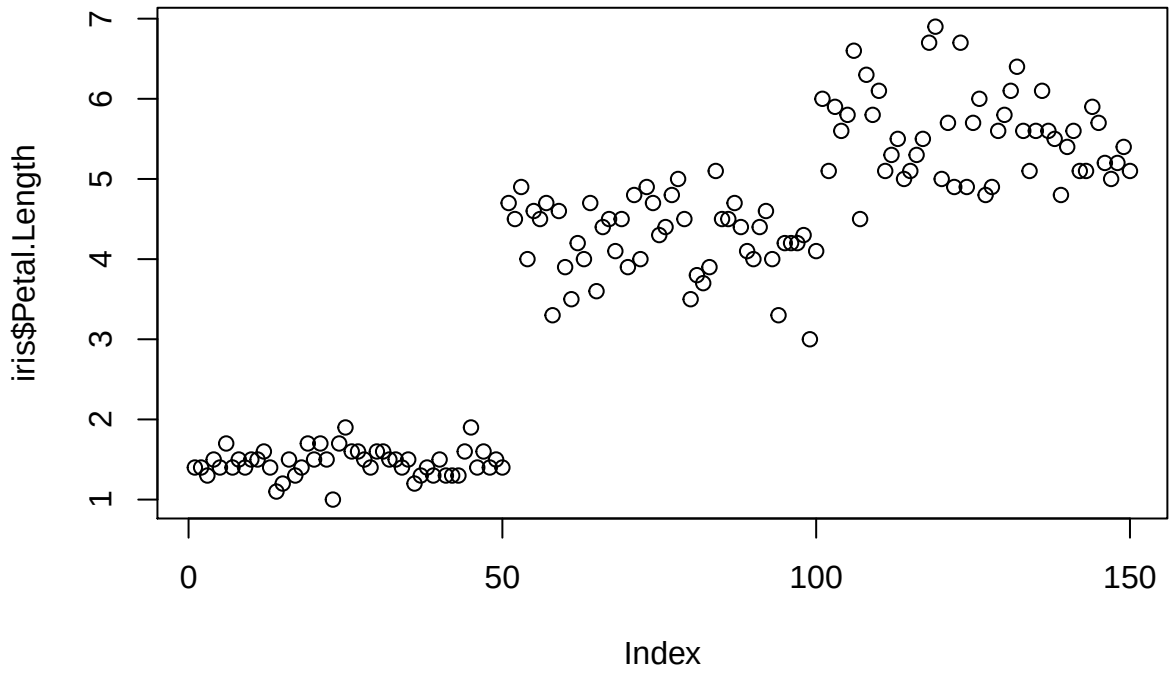

```
boxplot(iris$Sepal.Length~iris$Species,col=heat.colors(3))
```



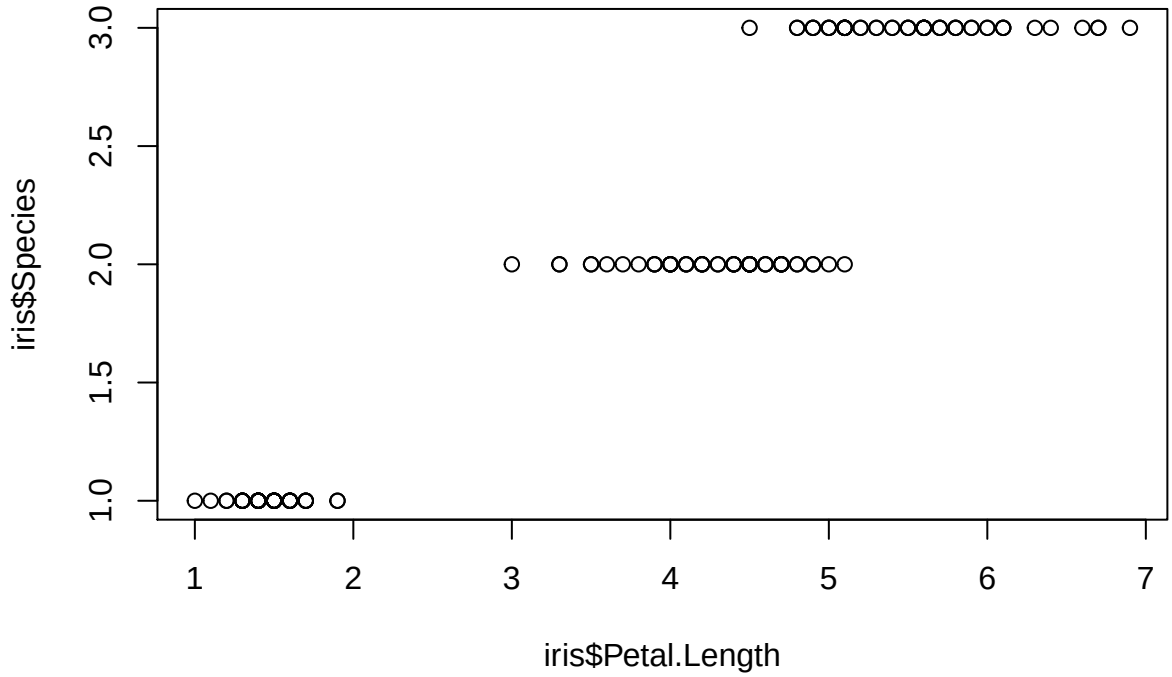

```
boxplot(iris$Sepal.Length~iris$Species,col=topo.colors(3))
```



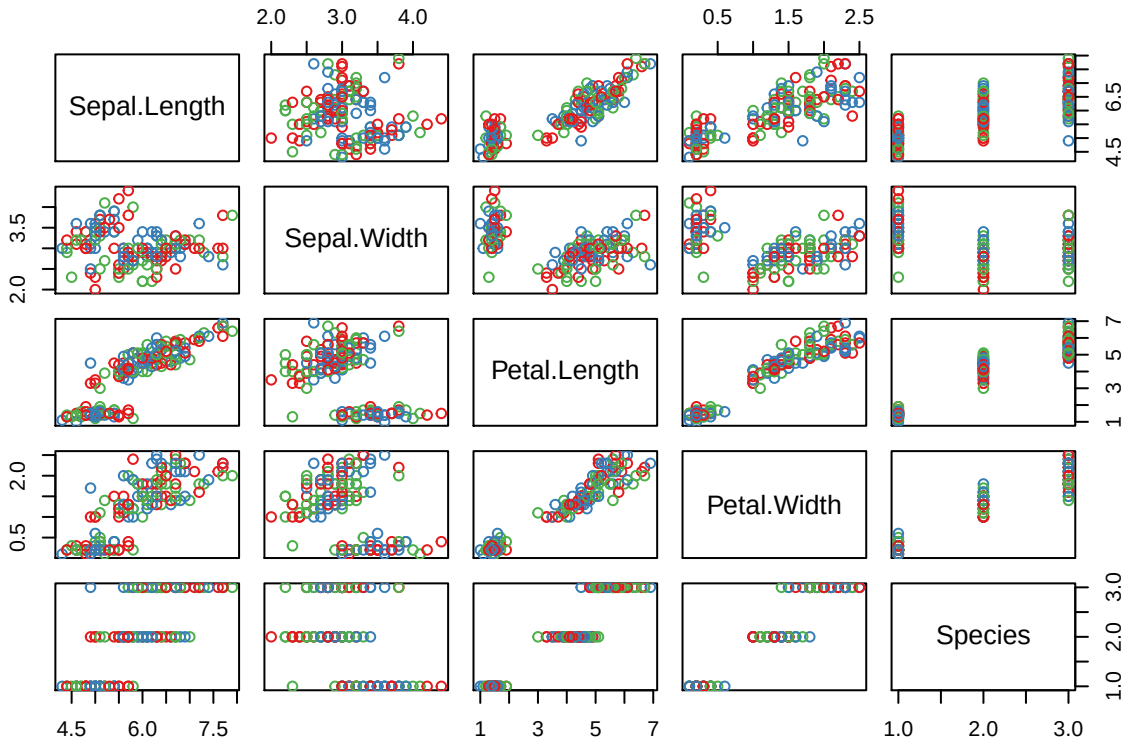

```
plot(x=iris$Petal.Length) #Simple Scatter Plot
```



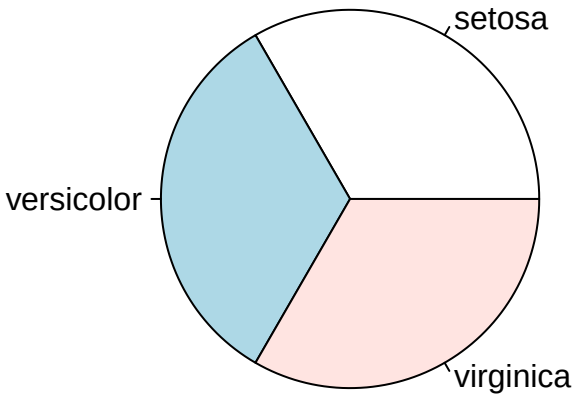

```
plot(x=iris$Petal.Length,y=iris$Species) #Multivariate Scatter Plot
```



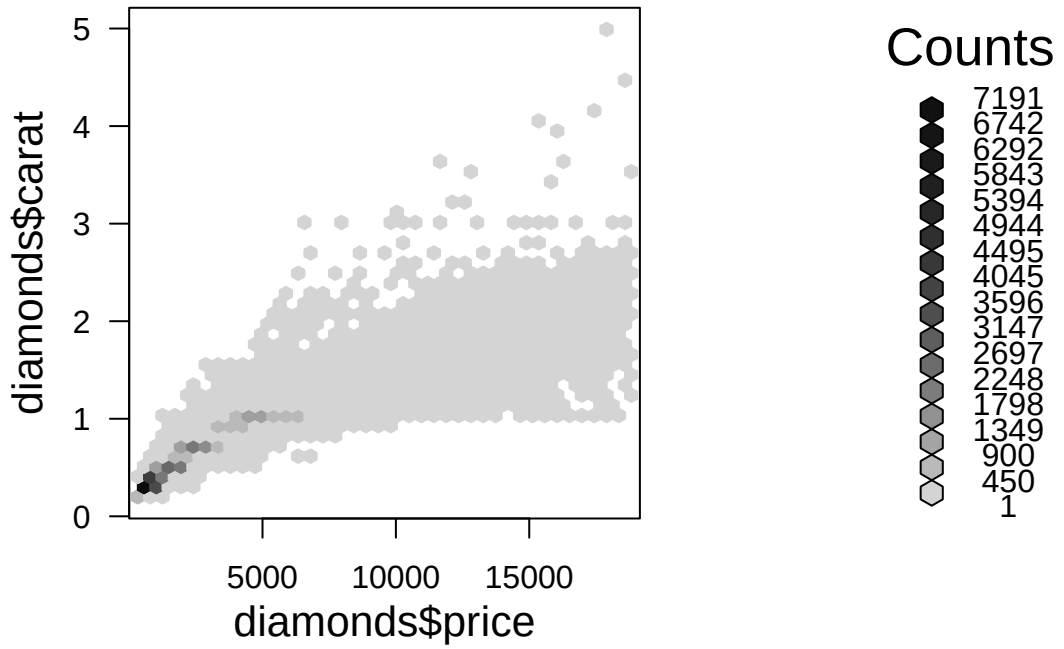

```
plot(iris,col=brewer.pal(3,"Set1"))
```




```
pie(table(iris$Species))
```



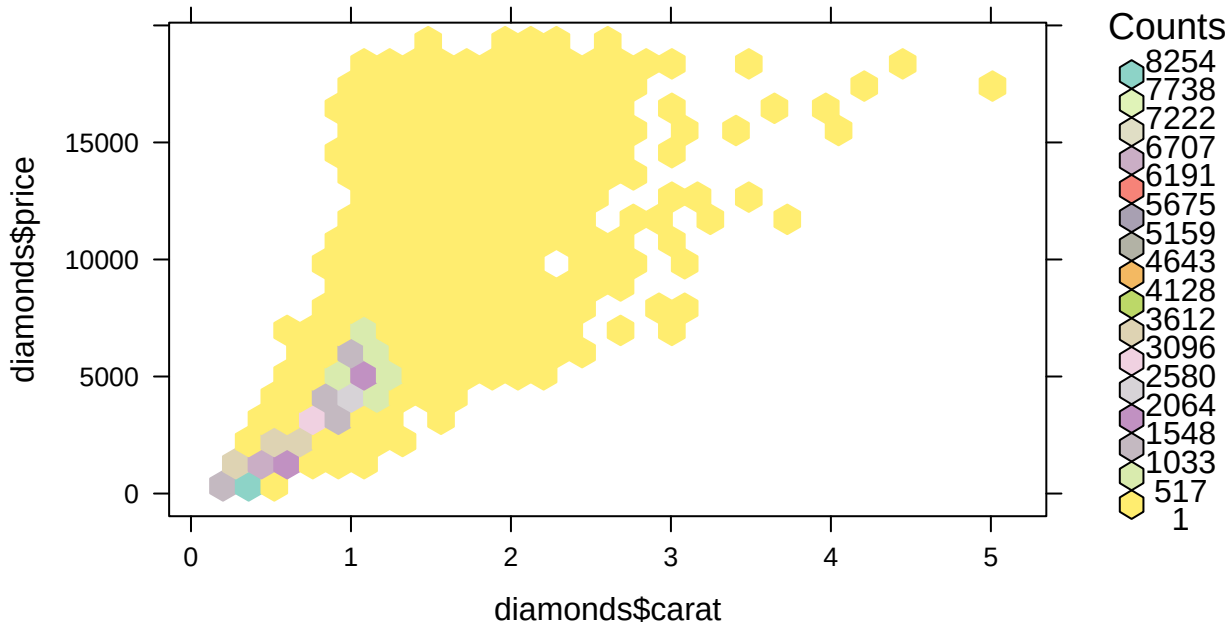

```
library(ggplot2)
data(diamonds)
library(hexbin)
a=hexbin(diamonds$price,diamonds$carat,xbins=40)
library(RColorBrewer)
plot(a)
```



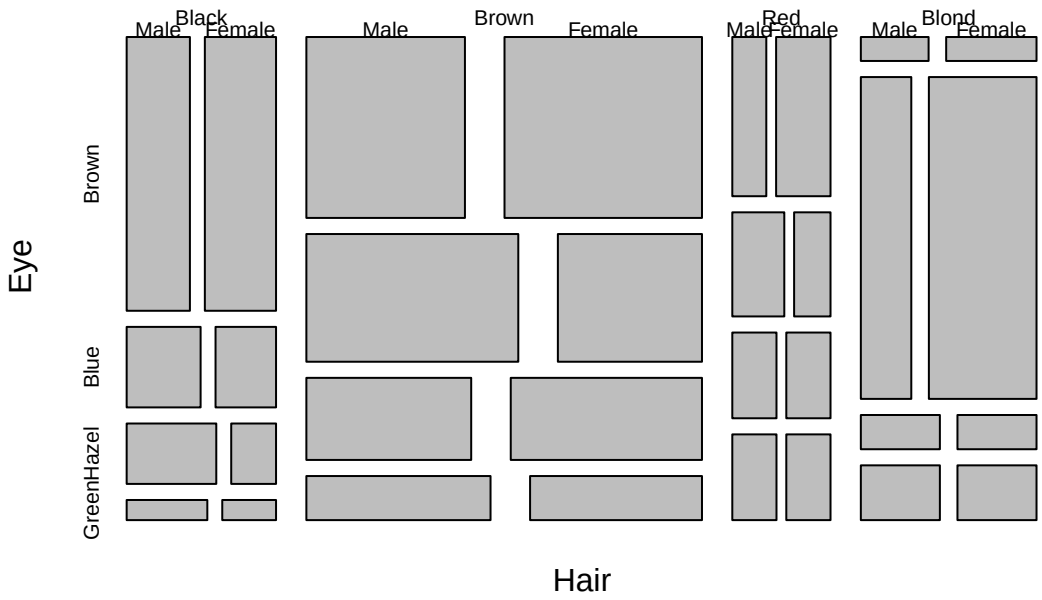

```
library(RColorBrewer)
rf <- colorRampPalette(rev(brewer.pal(40,'Set3')))
```

```
## Warning in brewer.pal(40, "Set3"): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```

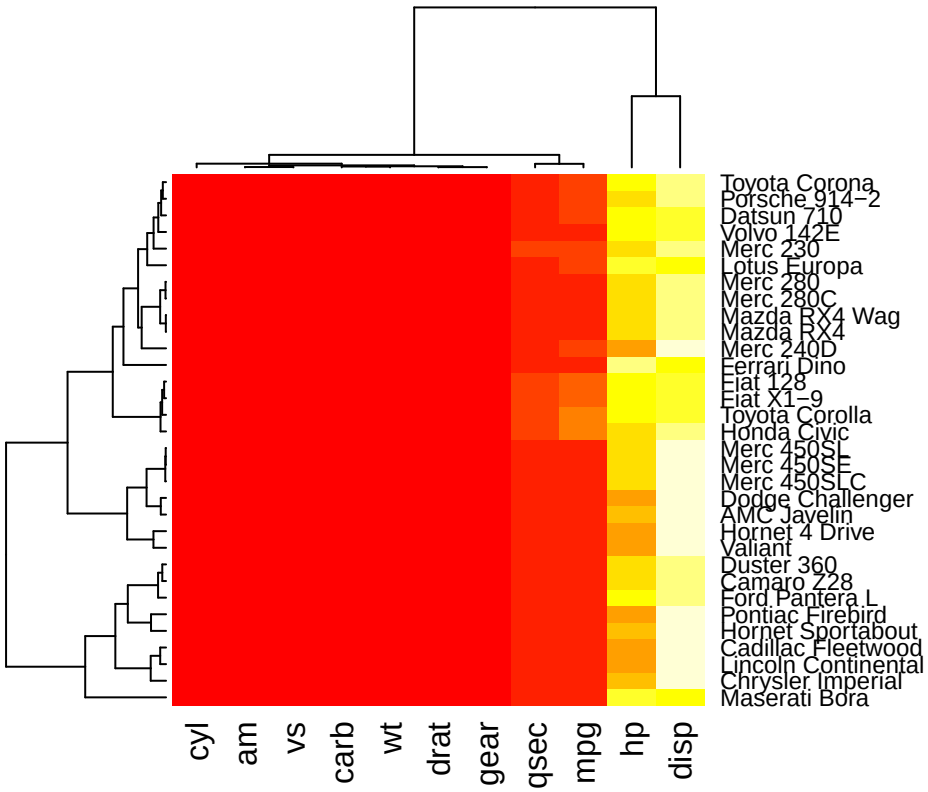
```
hexbinplot(diamonds$price~diamonds$carat, data=diamonds, colramp=rf)
```



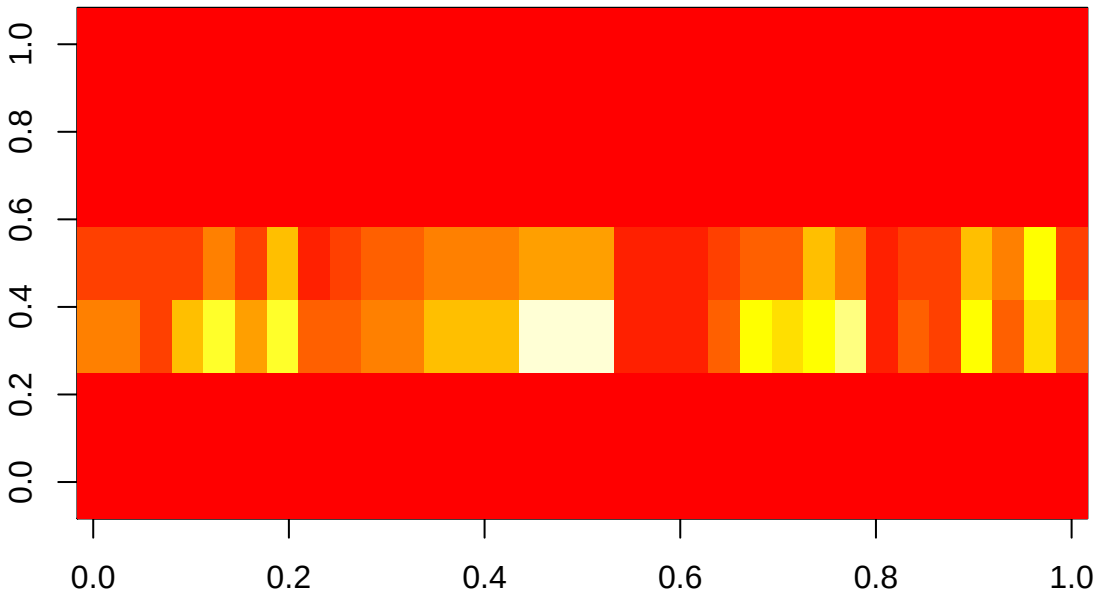

```
data(HairEyeColor)
mosaicplot(HairEyeColor)
```




```
heatmap(as.matrix(mtcars))
```




```
image(as.matrix(mtcars[1:7]))
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0