1.

Finding the Attributes and instances in the data:  
Using the Attributes function we can find the attributes of the data.   
 Using the attributes(hw1) in R Studio gives the following output.

$names

[1] "Weight" "Height" "gender" "avg\_exc\_mnth" "exc\_past\_year"

[6] "fried\_food\_week"

$class

[1] "data.frame"

$row.names

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

[31] 31 32 33 34 35 36 37

b)

All the variables are displayed using the “view” command  
view(hw1)

c) Displaying the statistics can be done in many ways.  
 ***Method 1:***

***Commands:***  
weights=hw1$Weight

mean(weights)

median(weights)

var(weights)

sd(weights)

***Output:***

|  |
| --- |
| weights=hw1$Weight  > mean(weights)  [1] 150.4595  > median(weights)  [1] 154  > var(weights)  [1] 1256.977  > sd(weights)  [1] 35.45388 |
|  |
| |  | | --- | | > | |

**Method 2:**

library("pastecs")

stat.desc(weights)

**Output:**stat.desc(weights)

nbr.val nbr.null nbr.na min max range sum

37.0000000 0.0000000 0.0000000 21.0000000 230.0000000 209.0000000 5567.0000000

median mean SE.mean CI.mean.0.95 var std.dev coef.var

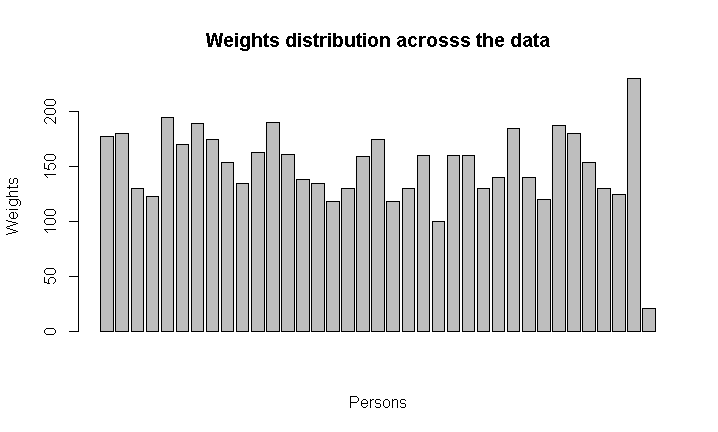
154.0000000 150.4594595 5.8285817 11.8209115 1256.9774775 35.4538782 0.2356374

***d)***

The normality of the data can be found based on the normality tests.

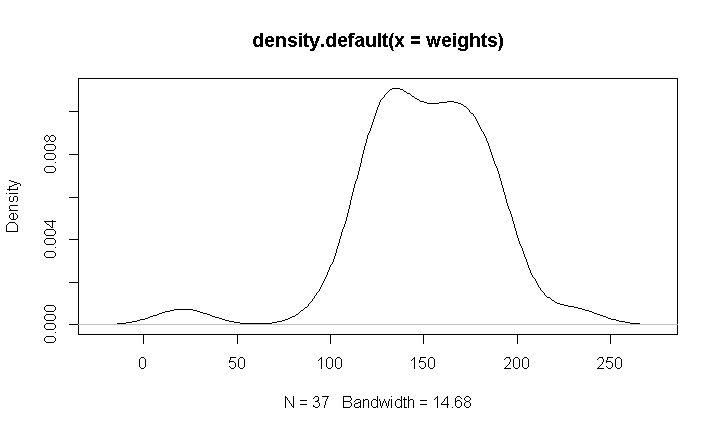
barplot(weights,xlab='number',ylab='weights',main ='Weights distribution acrosss the data')

***The distribution of the weights is shown in the bar chart plot.***



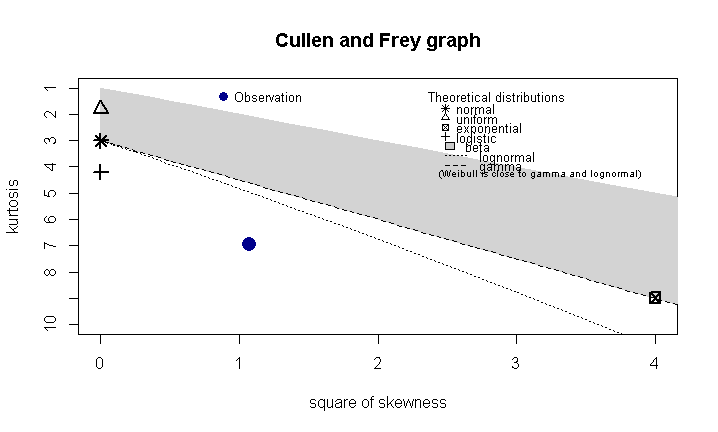
***The density distribution of the weights is shown can be plotted as shown below.***

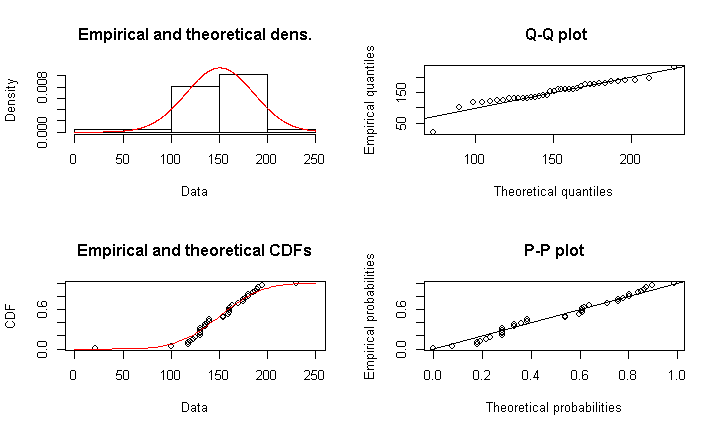
plot(density(weights))



***Method 2:***

**By Using the package “fitdistrplus” 🡪 fits the distribution to the respective data.**





***The output of the R is as below:***

> descdist(weights,discrete=FALSE)

summary statistics

------

min: 21 max: 230

median: 154

mean: 150.4595

estimated sd: 35.45388

estimated skewness: -1.035904

estimated kurtosis: 6.946055

>

> fit.norm <- fitdist(weights, "norm")

> plot(fit.norm)

> # Fit of the normal distribution to the given data at QQ plot and find the normality of the function

> fit.norm$aic

[1] 372.0369

***Looking at the QQ plot, we can observe that the data is almost fitting perfectly to the normality. So, the data is almost normal.***

***Question 2:***

***a)***

***Code***

gen = hw1$gender

gen.freq = table(hw1$gender)

gen.freq

Output:

Female Male

27 10

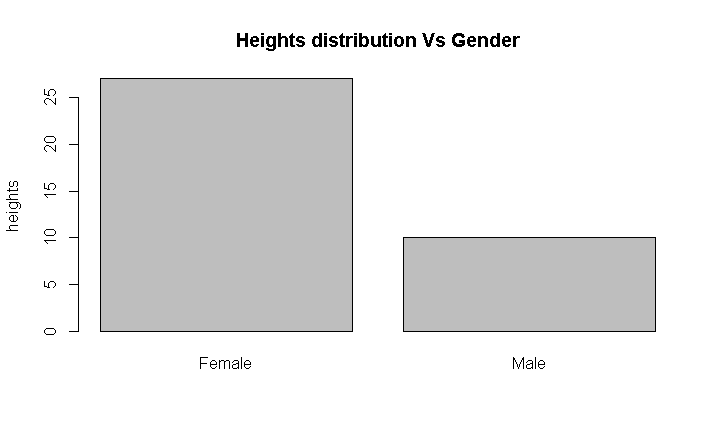
b)

Barplot of the data:

Code:

barplot(gen.freq,ylab='heights',main ='Heights distribution Vs Gender')

Output:



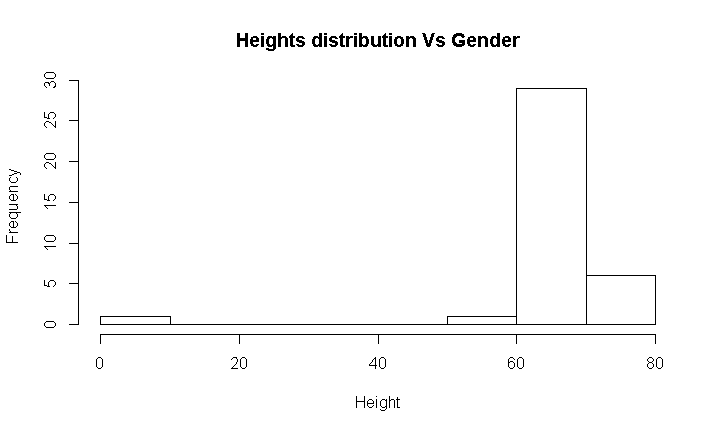
***c)***

***Code:***

> ht = hw1$Height

> hist(ht)

***Output:***



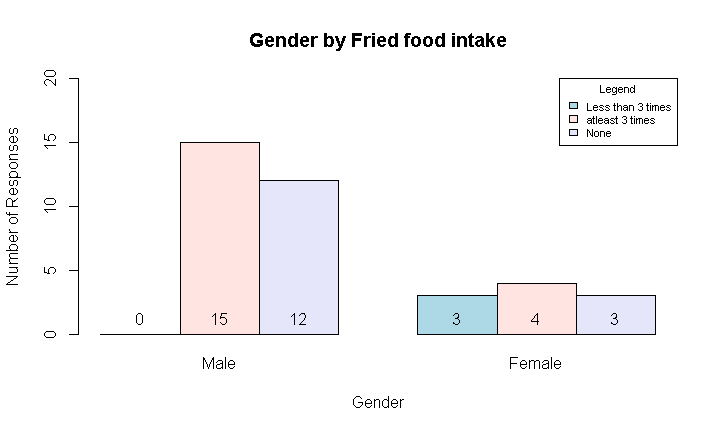
***d) Cluster bar graph***

***Code:***

> bp<-barplot(table(hw1$fried\_food\_week,hw1$gender),beside=T, main = "Gender by Fried food intake",col = c("lightblue", "mistyrose", "lavender"),xlab = "Gender", names = c("Male", "Female"), ylab = "Number of Responses", legend = c("Less than 3 times", "atleast 3 times", "None"), args.legend = list(title = "Legend", x = "topright", cex = .7), ylim = c(0, 20))

> text(bp, 0, round(table(hw1$fried\_food\_week,hw1$gender), 1),cex=1,pos=3)

***Output:***



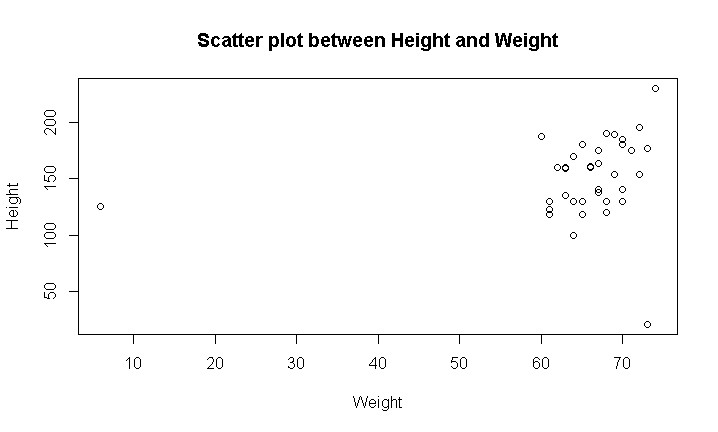
***e) Scatter plot between Height and Weight:***

***Scatter plot can be done in many ways.***

***Method 1:***

***Code:***

> plot(hw1$Height,hw1$Weight,xlab='Weight',ylab='Height',main="Scatter plot between Height and Weight")



***Method 2 :***

***Code:***

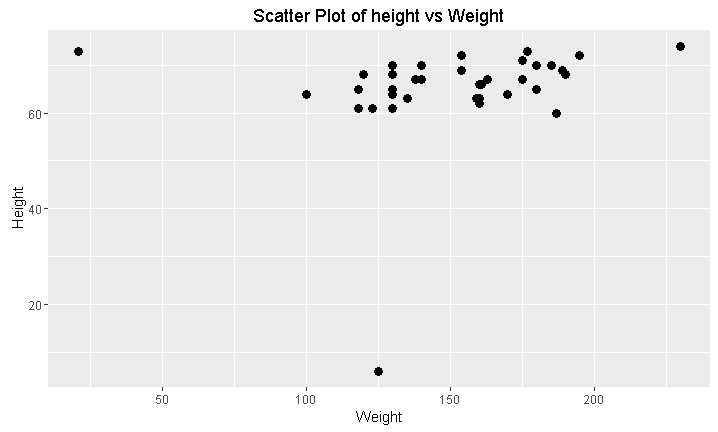
***Using ggplot2()***

>library(ggplot2)

> sc\_plot<-ggplot(hw1,aes(hw1$Weight,hw1$Height))

> sc\_plot+geom\_point(size = 3)+xlab("Weight")+ylab("Height")+ggtitle("Scatter Plot of height vs Weight")

***Output:***



***f. Five Point Summary:***

***Code:***

summary(hw1$Height)

***Output:***

summary(hw1$Height)

Min. 1st Qu. Median Mean 3rd Qu. Max.

6.00 63.00 67.00 65.05 70.00 74.00

**Boxplots:**

***3.***

******

***a) Given Function:***

(Give Birth = no) ^ (Can Fly = yes) 🡪 Birds

WKT

* Support = = proportion of records predicted correctly.
* Confidence = = proportion of correct predictions among records where it applies.

Here A = (Give Birth = no) ^ (Can Fly = yes)

B = (Can Fly = yes) -> Birds

Target = Birds

From the table, we see that they are independent events.

Therefore

Support = 3/20

Confidence = ¾

b)Using the 1-rule method discussed in class, find the relevant sets of classification rules for the  
target \Class" by testing each of the input attributes Blood Type, Give Birth, Can Fly, and  
Live in Water. Which of these three sets of rules has the lowest misclassification rate?

Target = Class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluating Attributes in the Animals Data** | | | | |
|  | Attribute | Rules | Errors | Total Errors |
|  |  |  |  |  |
| 1. | Blood Type | Warm-> Mammals | 4/11 | 9/20 |
|  |  | Cold -> Reptiles | 5/9 |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 2. | Gives Birth | Yes-> Mammals | 1/7 | 1/2 |
|  |  | No-> Reptiles | 9/13 |  |
|  |  |  |  |  |
| 3. | Can Fly |  |  |  |
|  |  | Yes -> Birds | 1/4 | 11/20 |
|  |  | No -> Mammals | 10/16 |  |
|  |  |  |  |  |
| 4. | Live in Water | Yes -> Fishes | 2/5 | 1/2 |
|  |  | No -> Mammals | 6/11 |  |
|  |  | Sometimes -> Amphibians | 2/4 |  |

Based on the minimum error, we select the blood type as the attribute to classify our data. But as the percentage of success is no better than a coin toss, we should further classify this to reach a pure dataset.