Unit 8 - Exercises

Contents

1. Graphs 2

1.1. Examples 2

1.2. Exercises 28

2. ANALYSIS OF VARIANCE & REGRESSION 31

2.1. Examples 31

2.2. Exercises 47

3. Annexure 55

1. Graphs

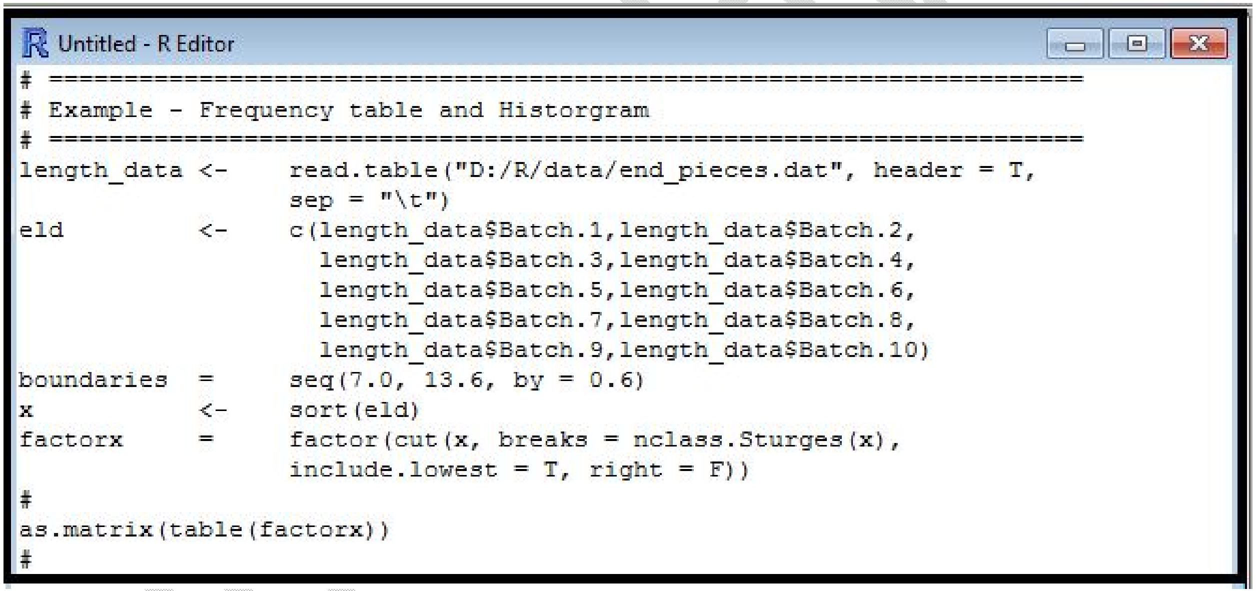
1.1. Examples

Example 1.1:

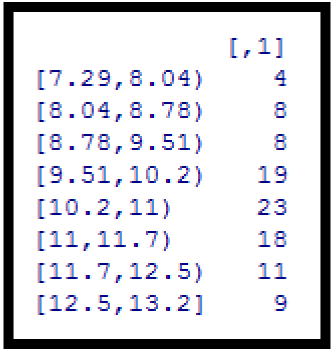
Data on lengths in centimeters (cm) of 100 end pieces in an engineering plant are specified in the file **end_pieces.dat**. (Refer to Annexure A-1)

Create a frequency table and also draw a histogram for the above data.

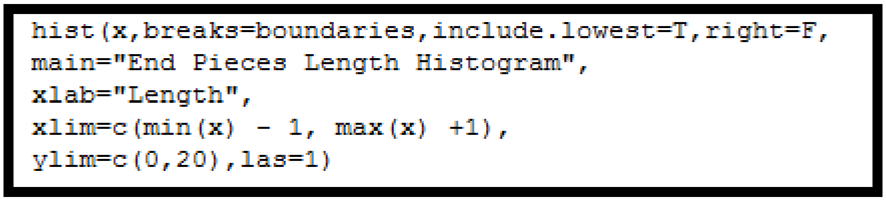
Solution:



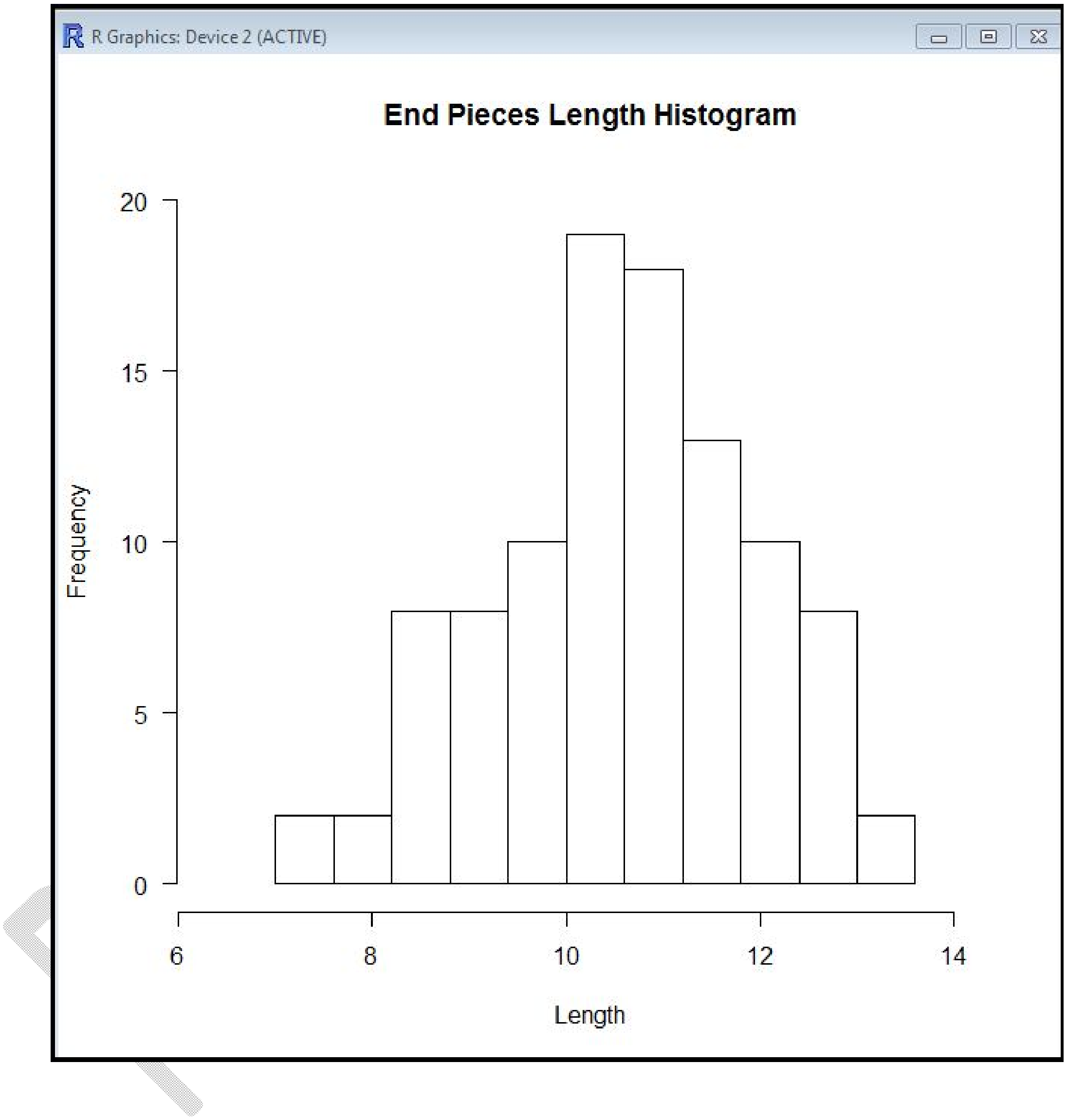
```
# ======================================================================
# Example - Frequency table and Historgram
# ======================================================================
length_data <-    read.table("D:/R/data/end_pieces.dat", header = T,
                  sep = "\t")
eld         <-    c(length_data$Batch.1,length_data$Batch.2,
                    length_data$Batch.3,length_data$Batch.4,
                    length_data$Batch.5,length_data$Batch.6,
                    length_data$Batch.7,length_data$Batch.8,
                    length_data$Batch.9,length_data$Batch.10)
boundaries  =     seq(7.0, 13.6, by = 0.6)
x           <-    sort(eld)
factorx     =     factor(cut(x, breaks = nclass.Sturges(x),
                  include.lowest = T, right = F))
#
as.matrix(table(factorx))
#
```



```
               [,1]
[7.29,8.04)       4
[8.04,8.78)       8
[8.78,9.51)       8
[9.51,10.2)      19
[10.2,11)        23
[11,11.7)        18
[11.7,12.5)      11
[12.5,13.2]       9
```

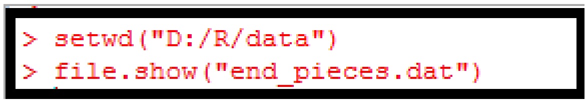


```
hist(x,breaks=boundaries,include.lowest=T,right=F,
main="End Pieces Length Histogram",
xlab="Length",
xlim=c(min(x) - 1, max(x) +1),
ylim=c(0,20),las=1)
```

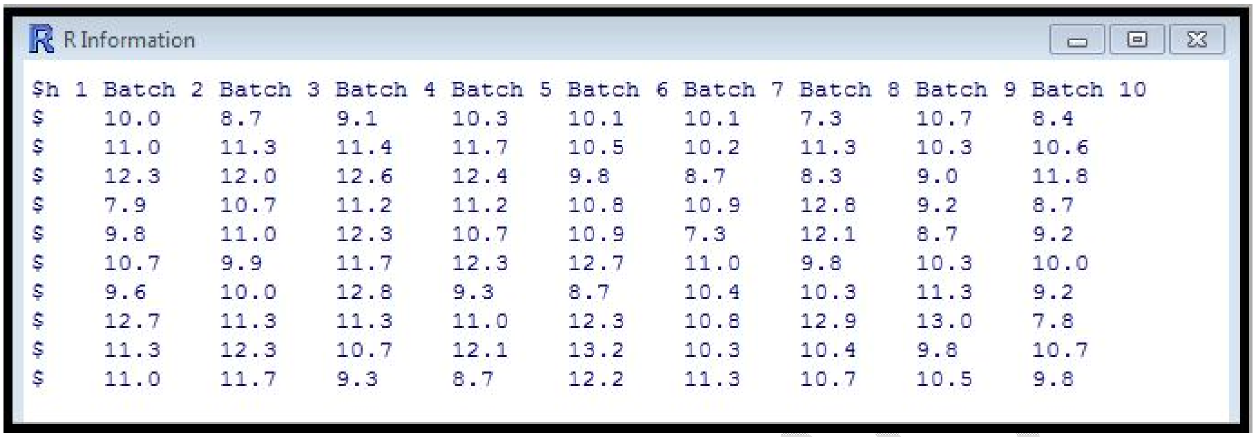


Explanation:

- *Set the directory to the location where the data file end_pieces.dat is stored.*
- *You can view the data by issuing the command file.show("end_pieces.dat")*



```
> setwd("D:/R/data")
> file.show("end_pieces.dat")
```



R Information

```
$h 1 Batch 2 Batch 3 Batch 4 Batch 5 Batch 6 Batch 7 Batch 8 Batch 9 Batch 10
$     10.0    8.7     9.1     10.3    10.1    10.1    7.3     10.7    8.4
$     11.0    11.3    11.4    11.7    10.5    10.2    11.3    10.3    10.6
$     12.3    12.0    12.6    12.4    9.8     8.7     8.3     9.0     11.8
$     7.9     10.7    11.2    11.2    10.8    10.9    12.8    9.2     8.7
$     9.8     11.0    12.3    10.7    10.9    7.3     12.1    8.7     9.2
$     10.7    9.9     11.7    12.3    12.7    11.0    9.8     10.3    10.0
$     9.6     10.0    12.8    9.3     8.7     10.4    10.3    11.3    9.2
$     12.7    11.3    11.3    11.0    12.3    10.8    12.9    13.0    7.8
$     11.3    12.3    10.7    12.1    13.2    10.3    10.4    9.8     10.7
$     11.0    11.7    9.3     8.7     12.2    11.3    10.7    10.5    9.8
```

Lines 1 - 5:

```
# ===================================================================== Line  1
# Example - Frequency table and Historgram                              Line  2
# ===================================================================== Line  3
length_data <-     read.table("D:/R/data/end_pieces.dat", header = T,  # Line  4
                   sep = "\t")                                          # Line  5
```

- *Lines 1 to 3 are the comment lines.*
- *Line 4: The command **read.table** reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.*
- *The data contains the heading and each piece of data is separated by a tab delimiter.*
- *Output of this command is assigned to the variable **length_data.***

Lines 6 – 10

```
eld          <-     c(length_data$Batch.1,length_data$Batch.2,          # Line  6
                      length_data$Batch.5,length_data$Batch.6,          # Line  8
                      length_data$Batch.7,length_data$Batch.8,          # Line  9
                      length_data$Batch.9,length_data$Batch.10)         # Line 10
```

- *The command **c** concatenates various columns in the variable **length_data** into a vector eld.*

Line 11

```
boundaries  =      seq(7.0, 13.6, by = 0.6)                            # Line 11
```

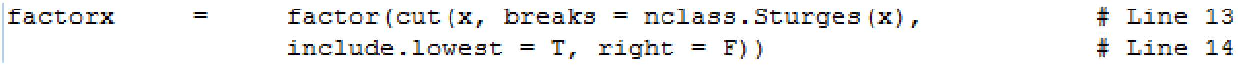
- *Create a vector, **boundaries** giving the breakpoints between histogram cells i.e.(7.0,7.6,8.2,8.8,9.4,10.0,10.6,11.2,11.8,12.6,13.2,13.8).*

Line 12

```
x            <-     sort(eld)                                           # Line 12
```

- *Sort **eld** into ascending order and store the result in x.*

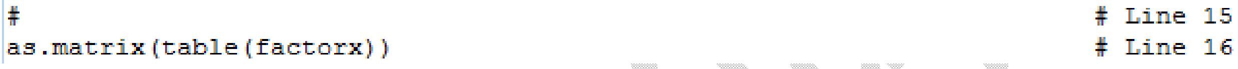
Line 13 - 14



```
factorx     =     factor(cut(x, breaks = nclass.Sturges(x),          # Line 13
                  include.lowest = T, right = F))                    # Line 14
```

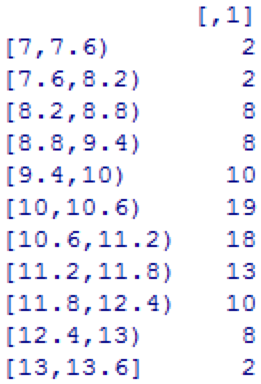
- *Now create a factor, **factorx** for the frequency table by*
  - a. *giving the breakpoints between histogram cells*
  - b. *creating a frequency table by using the option cut(x, breaks=breakpoints) where x is a numeric vector and breaks is the break points to divide x into different ranges based on the break points.*
  - c. *including the x[i] in the first bar, if x[i] = breaks value*
    If include.lowest= TRUE, an x[i] equal to the breaks value will be included in the first (or last, for right = FALSE) bar.
  - d. *creating right open intervals*
    If right = TRUE, the histograms cells are right-closed (left open) intervals.

Line 15 - 16



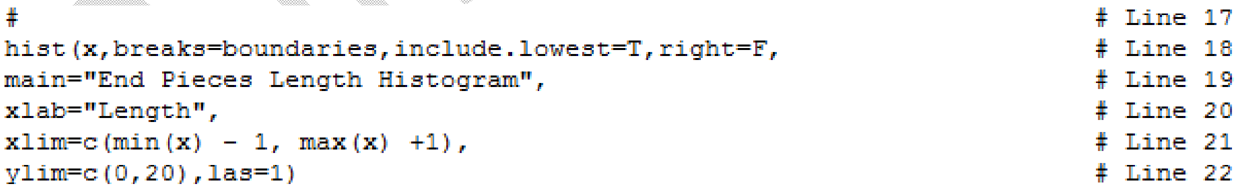
```
#                                                                    # Line 15
as.matrix(table(factorx))                                            # Line 16
```

- *Line 15 is a comment line.*
- *Convert the result into a matrix table which displays the frequency table.*



```
            [,1]
[7,7.6)        2
[7.6,8.2)      2
[8.2,8.8)      8
[8.8,9.4)      8
[9.4,10)      10
[10,10.6)     19
[10.6,11.2)   18
[11.2,11.8)   13
[11.8,12.4)   10
[12.4,13)      8
[13,13.6]      2
```

Line 17 – 22



```
#                                                         # Line 17
hist(x,breaks=boundaries,include.lowest=T,right=F,        # Line 18
main="End Pieces Length Histogram",                       # Line 19
xlab="Length",                                            # Line 20
xlim=c(min(x) - 1, max(x) +1),                            # Line 21
ylim=c(0,20),las=1)                                       # Line 22
```

- *Lines 17 is a comment line.*
- *Create a histogram for x with break calculated as in line 5;*
  - ○ *Including the x[i] in the first bar, if x[i] = breaks value*
  - ○ *disable right-closing of cell interval;*
  - ○ *set heading and x-axis label;*
  - ○ *make x-axis range from minimum of x to maximum of x;*
  - ○ *make y-axis range from 0 to 20;*
  - ○ *set axis-labels style horizontal*

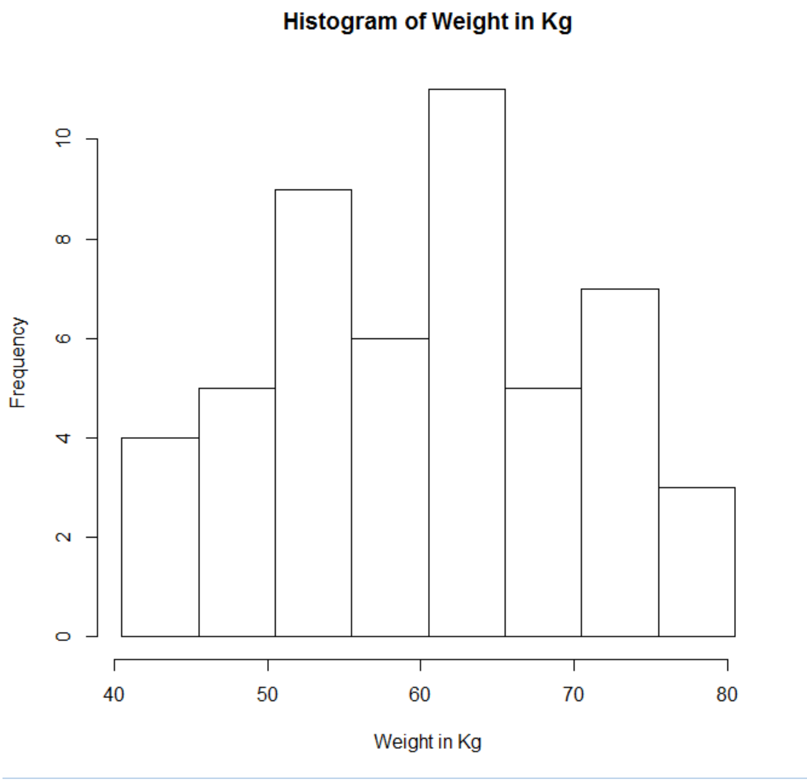
Example 1.2:

Construct a histogram for the following frequency distribution.

Weights in Kg.	40.5 – 45.5	45.5 – 50.5	50.5 – 55.5	55.5 – 60.5	60.5 – 65.5	65.5 – 70.5	70.5 – 75.5	75.5 – 80.5
Number of men	4	5	9	6	11	5	7	3

Solution:

```
> myhist <- list(breaks = seq(40.5,80.5,by=5),counts = c(4,5,9,6,11,5,7,3),
+ density=c(4,5,9,6,11,5,7,3)/50,xname="Weight in Kg")
> class(myhist)<- "histogram"
> plot(myhist)
```



Explanation:

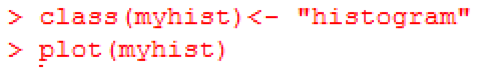
Line 1 -2

```
> myhist <- list(breaks = seq(40.5,80.5,by=5),counts = c(4,5,9,6,11,5,7,3),
+ density=c(4,5,9,6,11,5,7,3)/50,xname="Weight in Kg")
```

- *The function seq(40.5,80.5,by=5) creates the sequence 40.5,45.5,..80.5*

- *Construct a list myhist by coercing the boundaries, counts = number of men in each class interval, density = relative frequency being count divided by 50, x-axis name “Weight in Kg.”*

Line 3 - 4



```
> class(myhist)<- "histogram"
> plot(myhist)
```

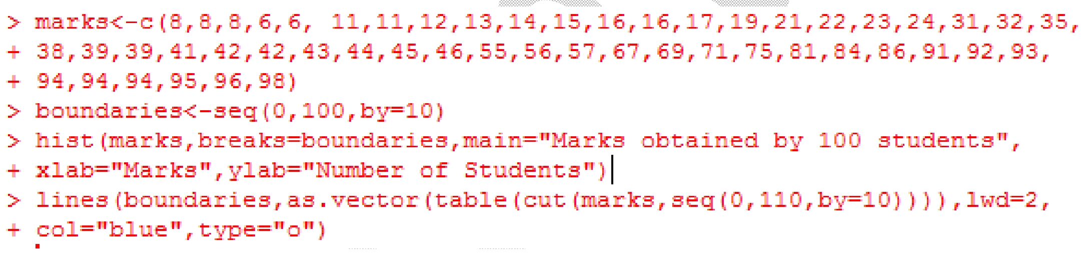
- *Make the class of the object myhist as histogram.*
- *Plot the histogram.*

Example 1.3:

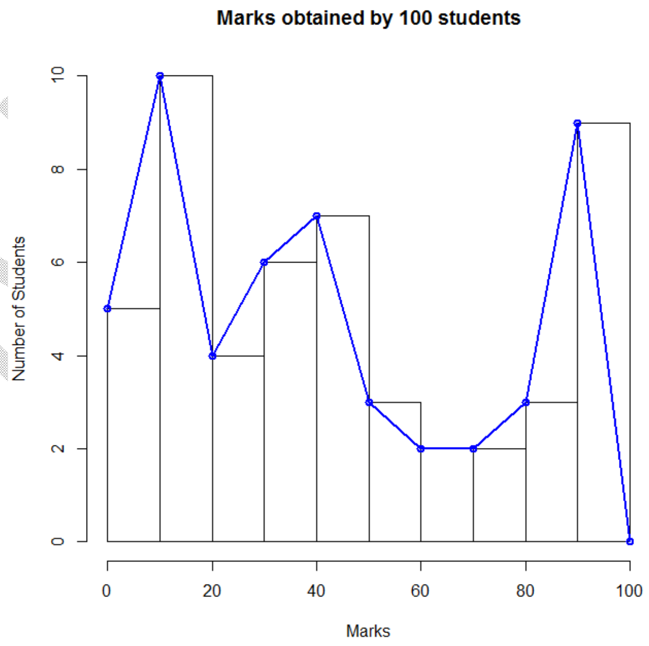
Construct a histogram and frequency polygon for the following data:

Consider the marks out of 100, obtained by the students in a class in a test.
8,8,8,6,6, 11,11,12,13,14,15,16,16,17,19,21,22,23,24,31,32,35,38,39,39,
41,42,42,43,44,45,46,55,56,57,67,69,71,75,81,84,86,91,92,93,94,94,94,95,96,98

Solution:



```
> marks<-c(8,8,8,6,6, 11,11,12,13,14,15,16,16,17,19,21,22,23,24,31,32,35,
+ 38,39,39,41,42,42,43,44,45,46,55,56,57,67,69,71,75,81,84,86,91,92,93,
+ 94,94,94,95,96,98)
> boundaries<-seq(0,100,by=10)
> hist(marks,breaks=boundaries,main="Marks obtained by 100 students",
+ xlab="Marks",ylab="Number of Students")
> lines(boundaries,as.vector(table(cut(marks,seq(0,110,by=10)))),lwd=2,
+ col="blue",type="o")
```



Explanation:

Line 1 – 3:

marks <- c(8,8,6,6,11,11,12,13,14,15,16,17,19,21,22,23,24,31,32,35,
38,39,39,41,42,42,43,44,45,46,55,56,67,69,71,75,81,84,86,91,92,93,
94,94,94,95,96,98)

*Construct a vector **marks** for the given data.*

Line 4

boundaries <- seq(10,100,by=10)

- *The option* **seq**(10,100,by=10) *creates the sequence 10,20,30,..100*

Line 5

hist(marks, breaks = boundaries, main = "Marks obtained by 100 students", xlab = "Marks", ylab="Number of Students")

- *The command **hist** is the basic command to construct a histogram.*
- *You can use the **breaks()** option to create the number of bins from the numeric vector, marks and give it a vector of breakpoints, boundaries.*
- *Use the option **main**, **xlab** and **ylab** to set the title for the graph, x-axis and y-axis labels.*

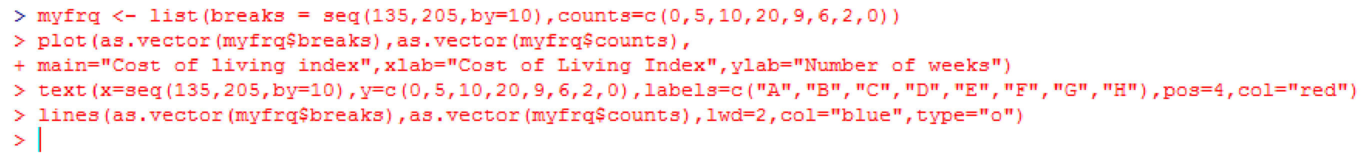
Example 1.4:

Construct a frequency polygon for the following data:

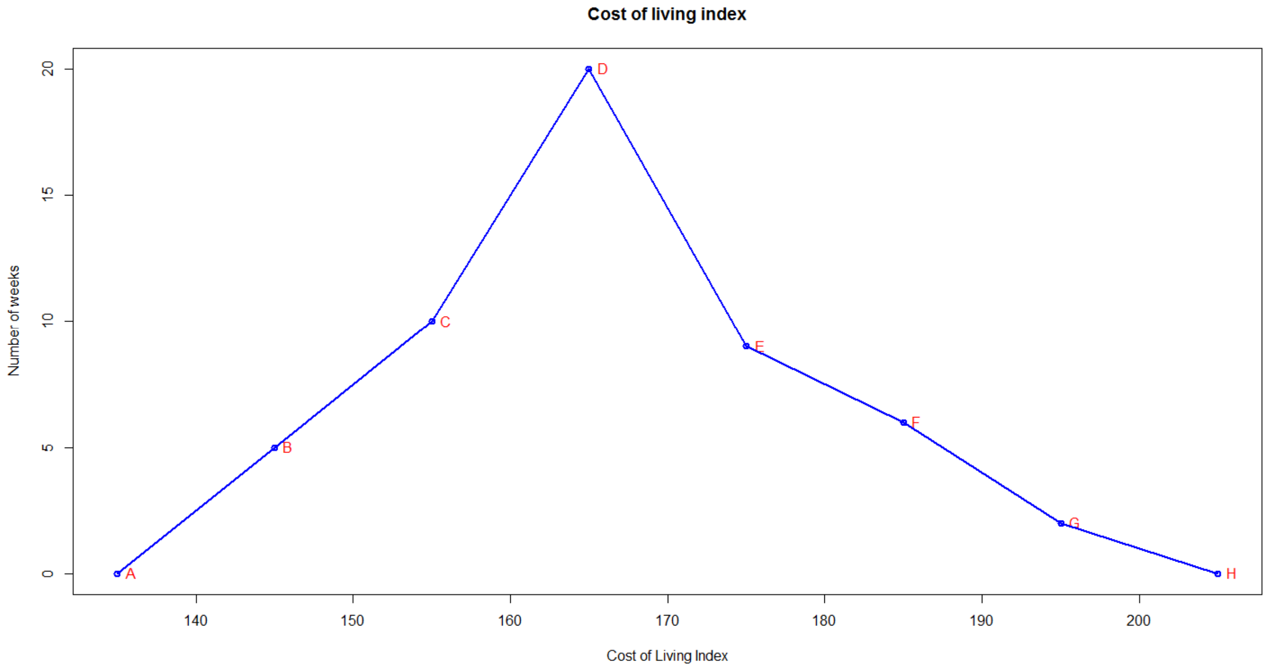
In a city, the weekly observations made in a study on the cost of living index are given in the following table.

Cost of living index	Number of weeks
140-150	5
150-160	10
160-170	20
170-180	9
180-190	6
190-200	2
Total	**52**

Solution:



```
> myfrq <- list(breaks = seq(135,205,by=10),counts=c(0,5,10,20,9,6,2,0))
> plot(as.vector(myfrq$breaks),as.vector(myfrq$counts),
+ main="Cost of living index",xlab="Cost of Living Index",ylab="Number of weeks")
> text(x=seq(135,205,by=10),y=c(0,5,10,20,9,6,2,0),labels=c("A","B","C","D","E","F","G","H"),pos=4,col="red")
> lines(as.vector(myfrq$breaks),as.vector(myfrq$counts),lwd=2,col="blue",type="o")
> 
```



Explanation:

We now draw a frequency polygon by plotting the cost-of-living index along with the horizontal axis, the frequencies along the vertical axis, and then plotting and joining the points B(145,5), C(155,10),D(165,20),E(175,9),F(185,6) and G(195,2) by line segments.

We should not forget to plot the point corresponding to the cost-of-living index in the class 130-140 (just before the lowest class 140-150) with zero frequency, that is A(135,0), and the point H(205,0) occurs immediately after G(195,2). So the resultant frequency polygon will be ABCDEFGH.

Line 1

```
> myfrq <- list(breaks = seq(135,205,by=10),counts=c(0,5,10,20,9,6,2,0))
```

- *The option **seq**(145,195,by=10) creates the sequence 135,155,165,175,..195,205*
- *Construct a list myfrq by coercing the breaks and counts(containing the cost of living in each class interval - each week)*

Line 2- 3

```
> plot(as.vector(myfrq$breaks),as.vector(myfrq$counts),
+ main="Cost of living index",xlab="Cost of Living Index",ylab="Number of weeks")
```

- *The command **plot** the points in vector, as.vector(myfrq$counts) versus the points in vector as.vector(myfrq$breaks);*
- *Set the **main** title for the plot as "Cost of living index";*
- *Set the X-axis title by specifying **xlab** as "Cost of Living Index";*
- *Set the Y-axis title by specifying **ylab** as "Number of weeks".*

Line 4

```
> text(x=seq(135,205,by=10),y=c(0,5,10,20,9,6,2,0),
+ labels=c("A","B","C","D","E","F","G","H"),pos=4,col="red")
```

*The command **text** draws the strings given in the vector labels at the coordinates given by x and y*

- *Option **pos** specifies a position parameter for the text*
    - *1 indicates position below the specified coordinates*
    - *2 indicates position to the left of the specified coordinates*
    - *3 indicates position above the specified coordinates*
    - *4 indicates to the right of the specified coordinates*
- *Option **col** is used to specify the color*

Line 5

```
> lines(as.vector(myfrq$breaks),as.vector(myfrq$counts),lwd=2,col="blue",type="o")
```

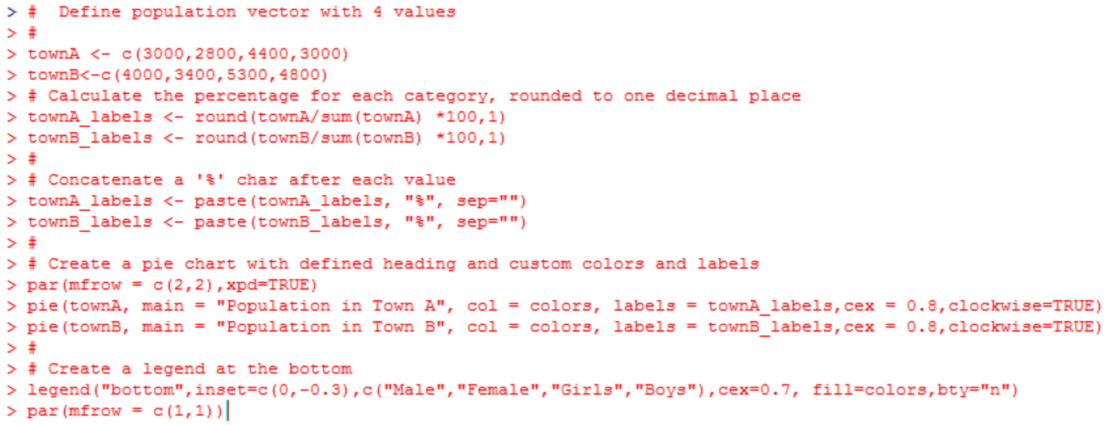
- *Line charts are created with the command **lines**(x,y,type=) where x and y are numeric vectors of (x,y) points to connect.*
- *The option **type** contains the following values:*
    - *p – point;*
    - *l- lines;*
    - *o – overplotted points and lines;*
    - *b.c points (empty if "c") joined by lines;*
    - *s,S – Stair steps;*
    - *h – histogram like vertical line;*
    - *n – does not produce any points or line*
- *The option **lwd** specifies the thickness of the line; 2 specify the line width in multiples of 1/96 inch*

Example 1.5:

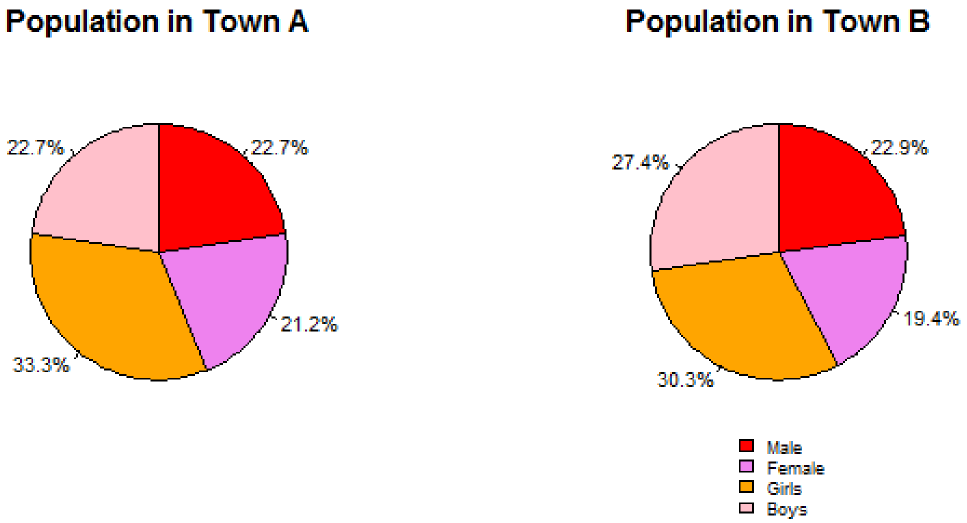
Draw a Pie diagram to represent the following population in a town:

Category	Town A	Town B
Male	3000	4000
Female	2800	3400
Girls	4400	5300
Boys	3000	4800
Total	13200	17500

Solution:



```
> #  Define population vector with 4 values
> #
> townA <- c(3000,2800,4400,3000)
> townB<-c(4000,3400,5300,4800)
> # Calculate the percentage for each category, rounded to one decimal place
> townA_labels <- round(townA/sum(townA) *100,1)
> townB_labels <- round(townB/sum(townB) *100,1)
> #
> # Concatenate a '%' char after each value
> townA_labels <- paste(townA_labels, "%", sep="")
> townB_labels <- paste(townB_labels, "%", sep="")
> #
> # Create a pie chart with defined heading and custom colors and labels
> par(mfrow = c(2,2),xpd=TRUE)
> pie(townA, main = "Population in Town A", col = colors, labels = townA_labels,cex = 0.8,clockwise=TRUE)
> pie(townB, main = "Population in Town B", col = colors, labels = townB_labels,cex = 0.8,clockwise=TRUE)
> #
> # Create a legend at the bottom
> legend("bottom",inset=c(0,-0.3),c("Male","Female","Girls","Boys"),cex=0.7, fill=colors,bty="n")
> par(mfrow = c(1,1))
```



Explanation:

Line 1 – 4

```
# Define population vector with 4 values
#
townA <- c(3000,2800,4400,3000)
townB<-c(4000,3400,5300,4800)
```

- *First two lines are comment lines.*
- *Third line constructs vector townA by concatenating the population values for various categories in town A.Fourth line constructs vector townB.*

Line 5 –12

```
# Calculate the percentage for each category, rounded to one decimal place
townA_labels <- round(townA/sum(townA) *100,1)
townB_labels <- round(townB/sum(townB) *100,1)
#
# Concatenate a '%' char after each value
townA_labels <- paste(townA_labels, "%", sep="")
townB_labels <- paste(townB_labels, "%", sep="")
#
```

- *Line 5 is a comment line.*
- *Line 6 creates the vector townA_labels, the percentage for each category of population in townA*
- *Line 7 creates the vector townB_lables, the percentage for each category of population in townB*
- *Line 8 and 9 are comment lines.*
- *Line 10 concatenates vector townA_lables with "%" symbol and this will be used in the diagram*
- *Similarly Line 11 concatenates vector townB_labels with "%" symbol.*
- *Line 12 is a comment line*

Line 13 –16

```
# Create a pie chart with defined heading and custom colors and labels
par(mfrow = c(2,2),xpd=TRUE)
pie(townA, main = "Population in Town A", col = colors, labels = townA_labels,cex = 0.8,clockwise=TRUE)
pie(townB, main = "Population in Town B", col = colors, labels = townB_labels,cex = 0.8,clockwise=TRUE)
```

- *Line 13 is a comment line.*
- *Line 14: The option **par** lets you set graphical parameters. The sub-option **mfrow** sets the number of rows and columns the plotting device will be divided in.*
  *Here, a vector of c(2,2) is given as the value to the **mfrow** subcommand. Subsequently, four plots are made as usual. Finally, you must set the graphical parameter to normal by entering vector c(1,1) to the **mfrow** subcommand.*

- *Set **xpd** as TRUE. If TRUE, all plotting is clipped to the figure region; if FALSE, all plotting is clipped to the plot region; if NA all plotting is clipped to the device region.*
- *Line 15 draws a pie diagram with the population values for townA with the defined heading and labels created in the above code*
- *Option **col** specifies a vector of colors to be used in filling or shading the slices. If missing, a set of 6 pastel colors is used.*
- *Option **cex** specifies the numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.*
- *Option **clockwise:** If TRUE, the slices are drawn clockwise; otherwise it is drawn anti-clockwise (default)*
- *Line 16 draws a pie diagram with population values for townB.*

Line 17 – 21

```
#
# Create a legend at the bottom
legend("bottom",inset=c(0,-0.3),c("Male","Female","Girls","Boys"),cex=0.7, fill=colors,bty="n")
par(mfrow = c(1,1))
```

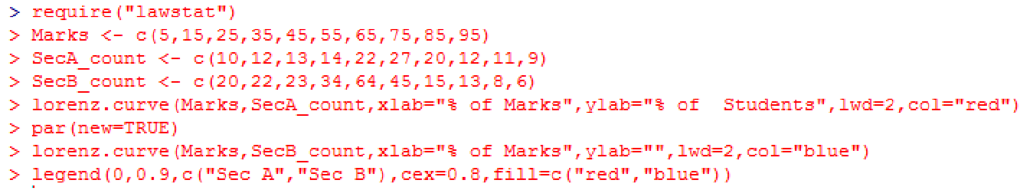
- *Line 17 and 18 are comment lines.*
- *Line 19 - 20: The command **legend** is used to add legends to the plot.*
- *Location of the legend can be specified by x and y coordinates. Also, we can specify a single keyword from the list ("bottomright" ,"bottom", "left", "topleft", "topright", "right", "center")*
- ***inset** distance(s) from the margins as a fraction of the plot region when legend is placed by a key word.*
- ***labels** specifies a character or expression vector of length > 1 to appear in the legend.*
- ***fill** will cause the boxes filled with the specified colors to appear beside the legend text*
- ***bty** specifies the type of box to be drawn around the legend. The allowed values are "o" and "n". Default is "o"*
- *Line 21 sets the graphical parameter to normal by specifying a vector c(1,1) to the **mfrow** subcommand.*

Example 1.6:

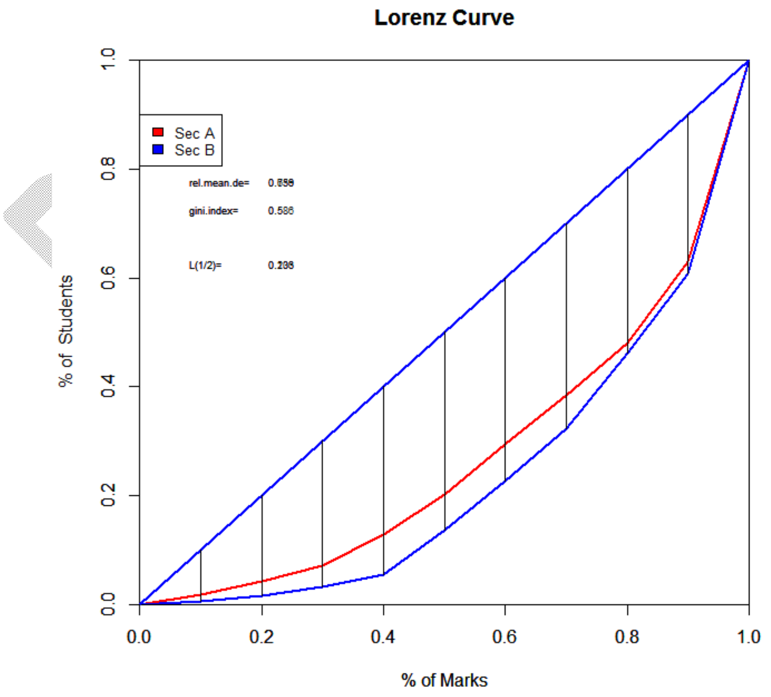
The frequency distribution of marks obtained by students in two sections are as follows. Draw a Lorenz curve in the same diagram and state your observation.

Marks	Number of Students	
	Sec A	Sec B
0 – 10	10	20
10 – 20	12	22
20 – 30	13	23
30 – 40	14	34
40 – 50	22	64
50 – 60	27	45
60 – 70	20	15
70 – 80	12	13
80 – 90	11	8
90 – 100	9	6

Solution:



```
> require("lawstat")
> Marks <- c(5,15,25,35,45,55,65,75,85,95)
> SecA_count <- c(10,12,13,14,22,27,20,12,11,9)
> SecB_count <- c(20,22,23,34,64,45,15,13,8,6)
> lorenz.curve(Marks,SecA_count,xlab="% of Marks",ylab="% of  Students",lwd=2,col="red")
> par(new=TRUE)
> lorenz.curve(Marks,SecB_count,xlab="% of Marks",ylab="",lwd=2,col="blue")
> legend(0,0.9,c("Sec A","Sec B"),cex=0.8,fill=c("red","blue"))
```



Explanation:

The Lorenz curve is a graphical method to display the concentration of activities within an area and it provides a good visual comparison of any observed differences.

We construct the following table to explain the solution.

Marks Mid values	Cumulative Marks	Cumulative % Marks	Sec A Students count	Sec A Students count – cum.	Sec A Students count – cum. %	Sec B Students count	Sec B Students count – cum.	Sec B Students count – cum. %
5	5	1	10	10	7	20	20	8
15	20	4	12	22	15	22	42	16.8
25	45	9	13	35	23	23	65	26
35	80	16	14	49	33	34	99	39.6
45	125	25	22	71	47	64	163	72.2
55	180	36	27	98	65	45	208	83.2
65	245	49	20	118	79	15	223	89.2
75	320	64	12	130	84	13	236	94.4
85	405	81	11	141	94	8	244	97.6
95	500	100	9	150	100	6	250	100

Before drawing a Lorenz curve, Install “lawstat” package.

Line 1

```
require("lawstat")
```

- *This command loads the package “lawstat” and updates the list of currently loaded packages and do not reload a package which is already loaded.*

Line 2 – 4

```
Marks <- c(5,15,25,35,45,55,65,75,85,95)
SecA_count <- c(10,12,13,14,22,27,20,12,11,9)
SecB_count <- c(20,22,23,34,64,45,15,13,8,6)
```

- *The above lines constructs vectors Marks from the Marks Mid value; SecA_count from the number of students in section A; SecB_count from the number of students in section B.*

Line 5

```
lorenz.curve(Marks,SecA_count,xlab="% of Marks",ylab="% of Students",lwd=2,col="red")
```

- *This command plots the Lorenz curve using Marks vector as the data, SecA_count vector as weight; set labels for x-axis and y-axis; set line width as 2; set color to "red".*

Line 6

```
par(new=TRUE)
```

- *To add to an existing plot use new = TRUE; to start a new plot, use the par(fig=starts)*

Line 7

```
lorenz.curve(Marks,SecB_count,xlab="% of Marks",ylab="",lwd=2,col="blue")
```

- *This command plots the Lorenz curve using Marks vector as the data, SecB_count vector as weight; set labels as empty strings; set line width as 2; set color to "blue".*
- *Note: to avoid overwriting the labels, set the axis labels to null.*

Line 8

```
legend(0,0.9,c("Sec A","Sec B"),cex=0.8,fill=c("red","blue"))
```

- *This command adds legends to the plot at the specified coordinates, fills the boxes with the specified colors and magnifies the size of the labels to the specified value.*

Observation:

- The further away the Lorenz curve is from the "line of perfect equality" (diagonal), the more diverse is the sample and the more unevenly the values are spread out.
- Lorenz curve for the marks of Section A students is closer to the line of perfect equality than the Lorenz curve for the marks of Section B students.
- Hence, t here is a great variability in the marks of Section B students.

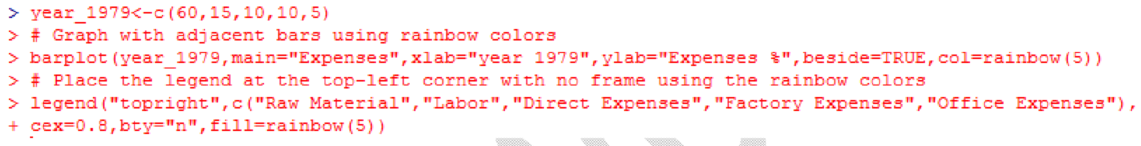
Example 1.7:

Particulars	Expressed in % for the year		
	‘1979	‘1980	‘1981
Raw Material	60	65	60
Labor	15	17.5	18
Direct Expenses	10	5	8
Factory Expenses	10	7.5	8
Office Expenses	5	5	6
Total Cost	100	100	100

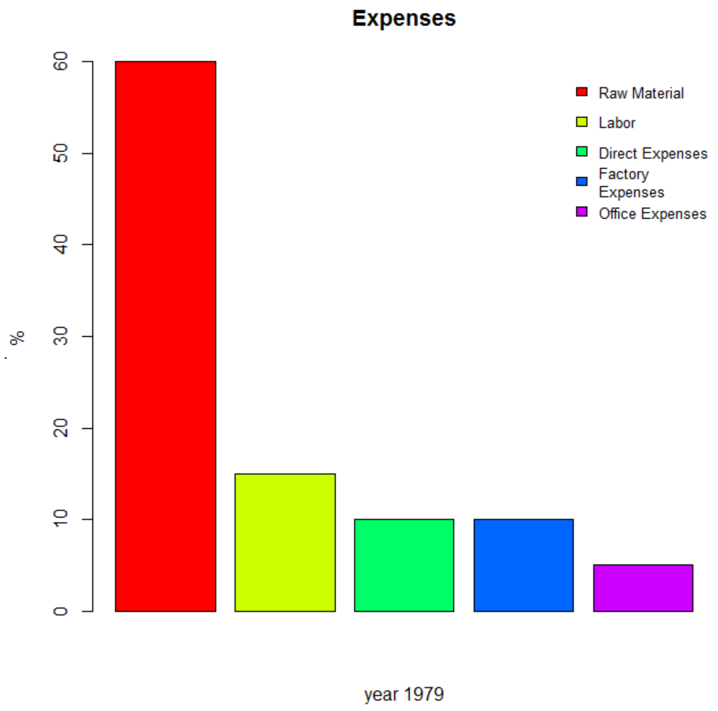
a. Represent the above data for 1979 by a simple bar diagram
b. Represent the above data by means of staked (a.k.a. sub-divided) bar diagrams:

Solution:

a)



```
> year_1979<-c(60,15,10,10,5)
> # Graph with adjacent bars using rainbow colors
> barplot(year_1979,main="Expenses",xlab="year 1979",ylab="Expenses %",beside=TRUE,col=rainbow(5))
> # Place the legend at the top-left corner with no frame using the rainbow colors
> legend("topright",c("Raw Material","Labor","Direct Expenses","Factory Expenses","Office Expenses"),
+ cex=0.8,bty="n",fill=rainbow(5))
```



Explanation:

Line 1

```
year_1979  <-  c(60,15,10,10,5)
```

- *The above command constructs vector year_1979 from the expenses for the year 1979 expressed as % for the year 1979.*

Line 2

```
# Graph with adjacent bars using rainbow colors
```

- *The above is a comment line.*

Line 3 - 4

```
barplot(year_1979,main="Expenses",xlab="year 1979",ylab="Expenses %",beside=TRUE,
col=rainbow(5))
```

- *The command **barplot** creates a plot with vertical or horizontal bars.*
- *The **title, x-axis label** and **y-axis** labels are set*
- *The option **beside** is a logical value .If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars*.
- *The option **horiz** has default value FALSE. If TRUE, the columns are drawn horizontally with first at the bottom; Otherwise, the bars are drawn vertically with the first bar to the left.*
- *The option **col** is* a vector of colors for the bars or bar components. By default, grey is used if `height` is a vector, and a gamma-corrected grey palette if `height` is a matrix.
- *The rainbow(5) creates a vector of 5 contiguous colors.*

Line 5

```
# Place the legend at the top-left corner with no frame using the rainbow colors
```

The above line is a comment line.

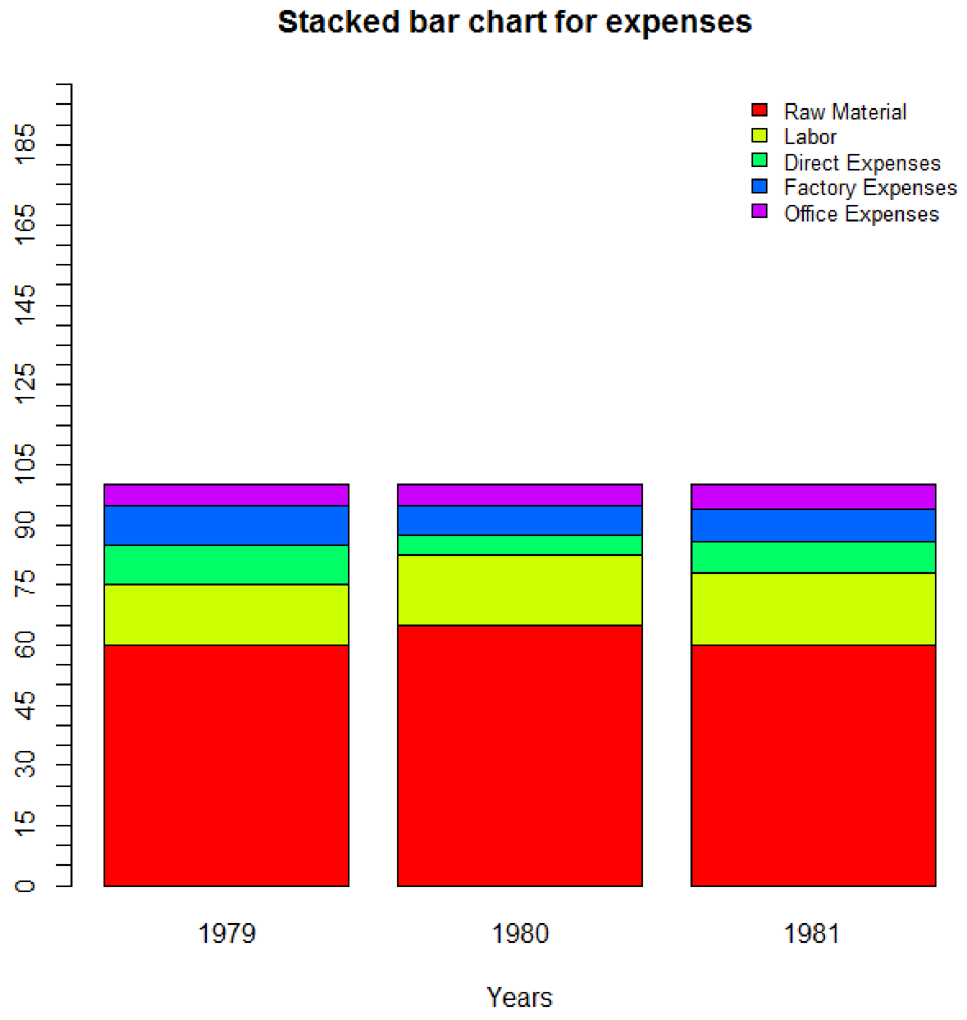
Line 6 – 8

```
legend("topright",c("Raw Material","Labor","Direct Expenses","Factory Expenses","Office
Expenses"),
cex=0.8,bty="n",fill=rainbow(5))
```

- *The command **legend** adds legend at the specified location ("topright")*
- *The height of the legend are magnified to the specified limit of 0.8.*
- *The option bty specifies the type of box to be drawn around the legend. The allowed values are "o" (the default) and "n". For value "o", the resulting box resembles the corresponding upper case letter. A value of "n" suppresses the box.*
- *The option fill causes the boxes filled with the specified colors to appear beside the legend text.*

b)

```
> # Stacked Bar Plot with Colors and Legend
> expenses <- c("Raw Material","Labor","Direct Expenses",
+ "Factory Expenses","Office Expenses")
> year_1979<-c(60,15,10,10,5)
> year_1980<-c(65,17.5,5,7.5,5)
> year_1981<-c(60,18,8,8,6)
> year_v <- cbind(year_1979,year_1980,year_1981)
>  # to modify the default way the axes are annotated
> par(lab=c(40,40,10)) # approximate number of tick marks on the x & y axes and label length
> barplot(year_v, main="Stacked bar chart for expenses ",
+   xlab="Years", col=rainbow(5),names.arg=c("1979","1980","1981"),ylim=c(0,200))
> legend("topright",expenses,cex=0.8,bty="n",fill=rainbow(5))
```



Explanation

Line 1 # Stacked Bar Plot with Colors and Legend

- *This is a comment line*

Line 2 - 7

```
expenses <- c("Raw Material","Labor","Direct Expenses",
"Factory Expenses","Office Expenses")
year_1979<-c(60,15,10,10,5)
year_1980<-c(65,17.5,5,7.5,5)
year_1981<-c(60,18,8,8,6)
year_v <- cbind(year_1979,year_1980,year_1981)
```

- *We construct vector, expenses from the expenses name, and the vectors, year_1979, year_1980, year_1981 from the expenditure in terms of percentages for the years 1979, 1980 & 1981 respectively*
- *By using command **cbind,** construct the matrix, year_v from the vectors* year_1979,year_1980 and year_1981

Line 8 # to modify the default way the axes are annotated

- *This is a comment line.*

Line 9 par(lab=c(40,40,10)) # approximate number of tick marks on the x & y axes and label length

- *By using the option lab in the command par, we set the number of tick marks of x and y axes. Default is (5,5,7)*

Line 10 - 11

```
barplot(year_v, main="Stacked bar chart for expenses ",
 xlab="Years", col=rainbow(5),names.arg=c("1979","1980","1981"),ylim=c(0,200))
```

- *By using the command **barplot,** we plot the stacked bar chart by specifying the vector or matrix describing the bars which make up the plot*
- *By using the options **main**, **xlab**, we set the title and x-axis label*
- *By using the option **col**, we use 5 contiguous colors.*
- *By using the option **names.arg**, we specify the labels to be plotted below each bar*
- *To have enough space above the bar chart for the legends, we set a higher y-axis limit by using the option **ylim***

Line 12 legend("topright",expenses,cex=0.8,bty="n",fill=rainbow(5))

- *The command **legend** adds legend at the specified location ("topright")*
- *The height of the legend are magnified to the specified limit of 0.8.*
- *The option bty specifies the type of box to be drawn around the legend. The allowed values are "o" (the default) and "n". For value "o", the resulting box resembles the corresponding upper case letter. A value of "n" suppresses the box.*
- *The option fill causes the boxes filled with the specified colors to appear beside the legend text.*

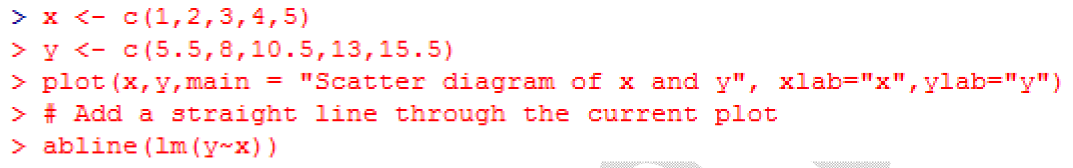
Scatter diagram

Example 1.8:

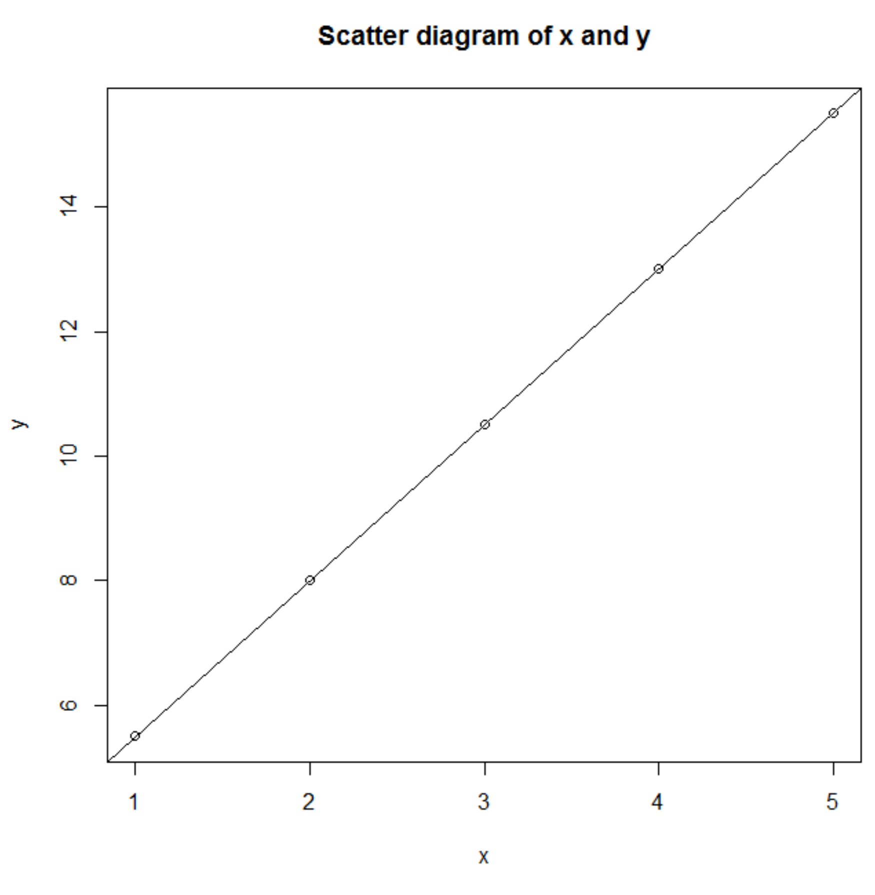
Find the scatter diagram of the variables x and y. Does it reveal any relationship between the variables?

x	y
1	5.5
2	8
3	10.5
4	13
5	15.5

Solution:



```
> x <- c(1,2,3,4,5)
> y <- c(5.5,8,10.5,13,15.5)
> plot(x,y,main = "Scatter diagram of x and y", xlab="x",ylab="y")
> # Add a straight line through the current plot
> abline(lm(y~x))
```



Explantation:

Line 1 - 2

```
x <- c(1,2,3,4,5)
y <- c(5.5,8,10.5,13,15.5)
```

- *Construct vectors x and y with the given values.*

Line 3

```
plot(x,y,main = "Scatter diagram of x and y", xlab="x",ylab="y")
```

- *A scatter plot is used when you have two variables to plot against one another. The command plot performs this task.*
- *Set the main heading and x-axis and y-axis label*

Line 4

```
# Add a straight line through the current plot
```

- *This is a comment line*

Line 5

```
abline(lm(y~x))
```

- *Add a best line of fit to the plot.*

Observation:

- ***There is a positive correlation among x and y variables.***
- *The shape of the line drawn through the data points, gauge the nature of relationship between the two variables.*
- *A straight line is interpreted as a linear relationship; a curved shape suggests a quadratic relationship.*
- *A line that lies relatively flat before suddenly shooting up or down is interpreted as an exponential relationship.*
- *When we examine the scatter plot for outliers, values that lie abnormally far from the cluster of data points. Outliers distort the relationship between the variables.*
- *A positive association is indicated by a upward trend (positive slope) - For example, higher income correspond to higher education levels.*
- *A negative association is indicated by negative slope.*
- *Absence of significant association is indicated by the scatter plot indicate no trend at all.*

Ogive Curves

Example 1.9:

Below is the frequency distribution of marks in Statistics obtained by 100 students in a class. Draw the Ogives (“less than” and “more than” type) for this distribution and use it to determine the median.

Marks Class Interval	Mid Marks	Frequency
20 - 30	29.5	7
30 - 40	39.5	11
40 - 50	49.5	24
50 - 60	59.5	32
60 - 70	69.5	9
70 - 80	79.5	14
80 - 90	89.5	2
90 - 100	99.5	1

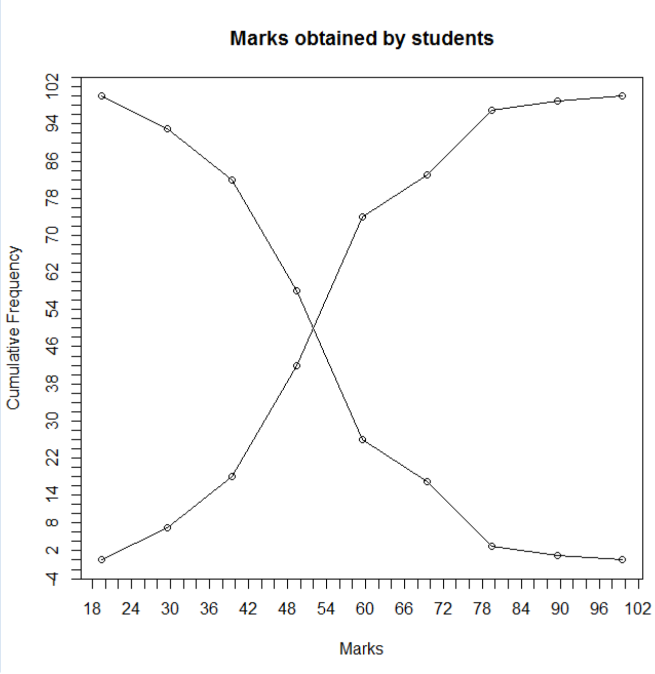
Solution:

An ogive is a curve that represents cumulative frequencies of the given variables. The Less than cumulative frequencies are plotted as points against the class limits. These points are joined by a smooth curve and it is know as "Less than Ogive".

Similarly, More than cumulative frequencies are plotted as points against the class limits. These points are joined by a smooth curve and it is known as "More than Ogive".

Class Limits	**Frequency**	**Cumulative Frequency**	
		Less than	**More than**
20 - 30	7	7	100
30 - 40	11	18	93
40 - 50	24	42	82
50 - 60	32	74	58
60 - 70	9	83	26
70 - 80	14	97	17
80 - 90	2	99	3
90 - 100	1	100	1

```
> freqmatrix <- function(x) {
+ cumfreq  =  cumsum(x)
+ sumv = sum(x)
+ morethanfrq = sumv
+ xlen = length(x)
+ rvcumfrq = rep(0,xlen)
+ for ( i in 1:xlen) {
+ rvcumfrq[i] =  morethanfrq
+ morethanfrq = morethanfrq - x[i]
+  }
+ m <- as.matrix(cbind(x,cumfreq,rvcumfrq))
+ return (m)
+ }
> plot_ogive <- function(x,m,titlelabel,xlabel,ylabel)  {
+ y1  <-  m[,2]
+ z1 <-  m[,3]
+ y   <- c("0",y1)
+ z   <- c(z1,"0")
+ par(lab=c(50,50,10))
+ # plot less than ogive curve
+ plot(x,y, main = titlelabel, xlab = xlabel,ylab=ylabel,axes=TRUE)
+ lines(x,y)
+ par(new=TRUE)
+ # plot more than ogive curve
+ plot(x,z, main = "", xlab ="",ylab="",axes=F)
+ lines(x,z)
+ }
> m= freqmatrix(c(7,11,24,32,9,14,2,1))
> plot_ogive(seq(19.5,99.5,by=10),m,"Marks obtained by students","Marks","Cumulative Frequency")
> locator()
$x
[1] 51.88613

$y
[1] 49.75674
```



Explanation

We write two functions, namely freqmatrix and plot_ogive.

*The function **freqmatrix** constructs a matrix containing the frequency and cumulative frequency for both less than and more than cases.*

Line 1 - 13 *contains the code for the function freqmatrix*

```
freqmatrix <- function(x) {
        cumfreq  =  cumsum(x)
        sumv = sum(x)
        morethanfrq = sumv
        xlen = length(x)
        rvcumfrq = rep(0,xlen)
        for ( i in 1:xlen) {
                rvcumfrq[i] =  morethanfrq
                morethanfrq = morethanfrq - x[i]
         }
        m <- as.matrix(cbind(x,cumfreq,rvcumfrq))
        return (m)
}
```

Line 2

- *Calculate the cumulative frequency for less than type and store it in the vector, cumfreq*

Line 3

- *Calculate the sum of frequency and store it in sumv*

Line 4

- *Set morethanfrq to sumv*

Line 5

- *Find the length of x and store it in xlen*

Line 6

- *Initialize the values for the vector rvcumfrq as 0*

Line 7 - 10

```
for ( i in 1:xlen) {
        rvcumfrq[i] =  morethanfrq
        morethanfrq = morethanfrq - x[i]
 }
```

- *Calculate more than cumulative frequency values.*
- *Store the sum of x as the first value in the vector, rvcumfrq.*
- *Then subtract the frequency value from rvcumfrq and store it as the next value in the vector, rvcumfrq.*
- *Repeat the above step for all values of the frequency*

Line 11

- *Construct a matrix by using the command cbind and store vector x, cumfreq and rvcumfrq*

Line 12

- *Return the matrix m*

The function **plot_ogive** *plots Less than and more than Ogive curves.*

Line 14 - 27 *contains the code for the function* **plot_ogive.**

```
plot_ogive <- function(x,m,titlelabel,xlabel,ylabel)  {
          y1  <-  m[,2]
          z1 <-  m[,3]
          y   <- c("0",y1)
          z   <- c(z1,"0")
          par(lab=c(50,50,10))
          # plot less than ogive curve
          plot(x,y, main = titlelabel, xlab = xlabel,ylab=ylabel,axes=TRUE)
          lines(x,y)
          par(new=TRUE)
          # plot more than ogive curve
          plot(x,z, main = "", xlab ="",ylab="",axes=F)
          lines(x,z)
}
```

Line 16 – 17

```
          y   <- c("0",y1)
          z   <- c(z1,"0")
```

- *Concatenate "0" as the first value to the vector cumfreq.*
- *Concatenate "0" as the last value to the vector rvcumfrq.*

Line 18

```
          par(lab=c(50,50,10))
```

- *Modify the default way that axes are annotated.*
- *The option* ***lab*** *= c(x,y,len) specifies x and y to give the approximate number of tick marks on the x axis and y-axis; len specifies the the label length.*
- *Default is c(5,5,7).*

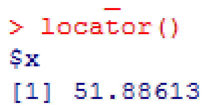
Line 23

```
          par(new=TRUE)
```

To add to an existing plot use new = TRUE; to start a new plot, use the par(fig=starts)

- *After plotting the less than Ogive curve, set the option par with the option new = TRUE so that the more than Ogive curve is drawn in the same plot.*

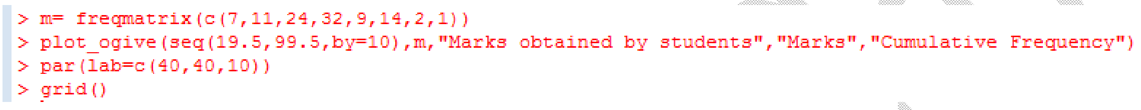
Observation



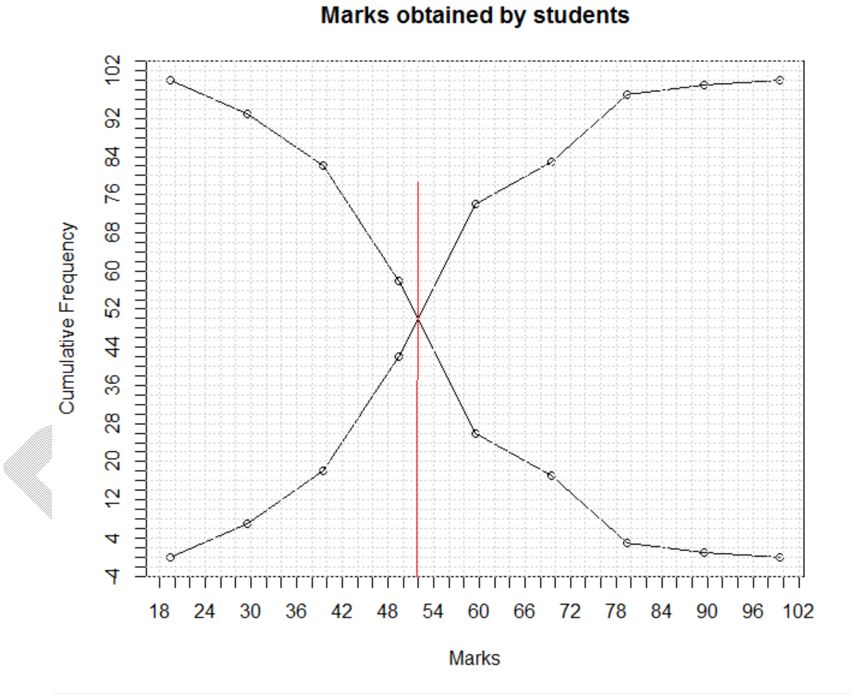
```
> locator()
$x
[1] 51.88613
```

- *The intersection point of the less than Ogive curve and More than Ogive curve gives the median.*
- *We use the Interactive graphics here.*
- *The* ***locator()*** *reads the position of the graphics cursor when the mouse button is pressed.*
- *We place the cursor at the point of intersection and press the mouse button, the value 51.88613*
- *So, we find the median to be approximately 52.*

Alternatively, we can get the location of intersection very easily, as shown below:



```
> m= freqmatrix(c(7,11,24,32,9,14,2,1))
> plot_ogive(seq(19.5,99.5,by=10),m,"Marks obtained by students","Marks","Cumulative Frequency")
> par(lab=c(40,40,10))
> grid()
```



By mere visual inspection (we have drawn a line in red color), we see that the median is 52.

1.2. Exercises

Exercise 1.1:

Draw a histogram to present the following data:

Income (in '000s – Rs)	Number of Individuals
100 – 149	21
150 – 199	32
200 – 249	52
250 – 299	105
300 – 349	62
350 – 399	43
400 – 449	18
450 – 499	9
Total	342

Exercise 1.2:

The heights of 50 students, measured to the nearest centimeter (cm) have been found to be as follows:

161	150	154	165	168	161	154	162	150	151
162	164	171	165	158	154	156	172	160	170
153	159	161	170	162	165	166	168	165	164
154	152	153	156	158	162	160	161	163	166
161	159	162	167	168	159	158	153	154	159

a) Represent the data given above by a grouped frequency table taking the class intervals as 160-165, 165-170, etc.,

b) Draw a histogram to present the above data

Exercise 1.3:

Consider the marks, out of 50. Obtained by 25 students of a class to a test given in the table below:

Marks	Number of students
0 – 10	1
10 – 20	4
20 – 30	14
30 – 40	4
40 – 50	2
Total	25

Draw a frequency polygon corresponding to this frequency distribution table.

Exercise 1.4:

Represent the following data by a simple bar diagram:

Year	Production (in Tonnes)
1974	45
1975	40
1976	44
1977	41
1978	49
1979	55
1980	50

Exercise 1.5:

Present the profit before tax and profit after tax for the year ended 30th September 2009, 2010, 2011,2012 and 2013 respectively of the public limited company mentioned below by a bar diagram.

Financial highlights of the Public Limited Co.

Year ended 30th September	Profit before Tax (In Lakhs of Rupees)	Profit after Tax (In Lakhs of Rupees)
2009	190	79
2010	191	71
2011	200	90
2012	109	36
2013	127	89

Exercise 1.6:

Represent the following data by a stacked bar diagram:

Details	Commodity A (in Rs.)	Commodity B (in Rs.)
Value of Raw material	175	100
Other production Expenses	30	25
Profits	20	25
Total	225	150

Exercise 1.7:

Draw a Pie diagram to represent the following population in a town:

Male	2000
Female	1800
Girls	4200
Boys	2000
Total	10,000

Exercise 1.8:

Draw a Pie diagram to represent the following data:

Type of commodity	Expenditure in '000 Rs.	
	Family A	Family B
Food	5	6
Rent	20	30
Clothes	2	3
Education	10	14
Miscellaneous	3	5
Savings	8	12
Total	48	70

Exercise 1.9:

The frequency distribution of wages in a certain factory is as follows:

Wages (in Rs.)	No. of Workers
7000 – 7499	10
7500 – 7999	18
8000 – 8499	27
8500 – 8999	20
9000 – 9499	15
9500 – 9999	8
10000 – 10499	2

Draw an Ogive curve for this distribution.

Exercise 1.10:

Draw a Lorenz curve from the following data to study the extent of dispersion graphically.

Amount of profits (in Rs.)	No. of companies
150	28
160	20
600	34
840	30
1050	28
1500	26
1700	22
4000	12

2. ANALYSIS OF VARIANCE & REGRESSION

2.1. Examples

Example 2.1:

To test the lifetime of batteries, 12 toy drummers are fitted with new batteries of three types: Amazing, Superlong, Endurance.

The lengths of time (in hours) that the drummers continue to drum are summarized in the table below:

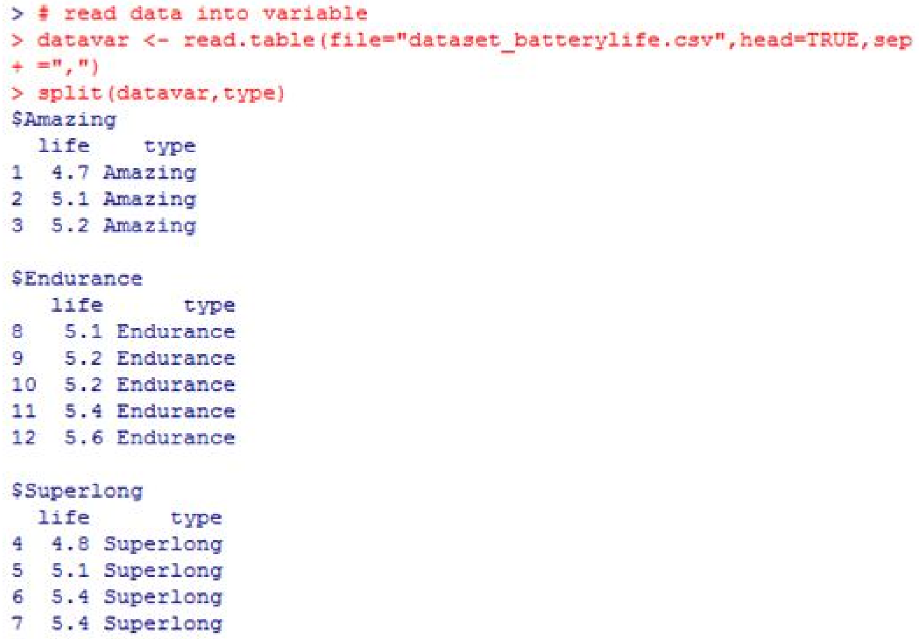
Amazing	4.7, 5.1, 5.2
Superlong	4.8,5.1,5.4,5.4
Endurance	5.1,5.2,5.2,5.4,5.6

Determine whether there is significant evidence, at the 5 % level, of a difference between the mean lifetimes of the three types of batteries:

Summarize the findings in the anova table.

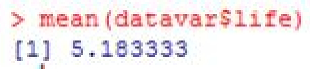
Solution:

First, we create a comma separated file "dataset_batterylife.csv" from the given values. Then check the data entered.



```
> # read data into variable
> datavar <- read.table(file="dataset_batterylife.csv",head=TRUE,sep
+ =",")
> split(datavar,type)
$Amazing
  life    type
1  4.7 Amazing
2  5.1 Amazing
3  5.2 Amazing

$Endurance
   life      type
8   5.1 Endurance
9   5.2 Endurance
10  5.2 Endurance
11  5.4 Endurance
12  5.6 Endurance

$Superlong
  life      type
4  4.8 Superlong
5  5.1 Superlong
6  5.4 Superlong
7  5.4 Superlong
```

Then we estimate the mean life time.



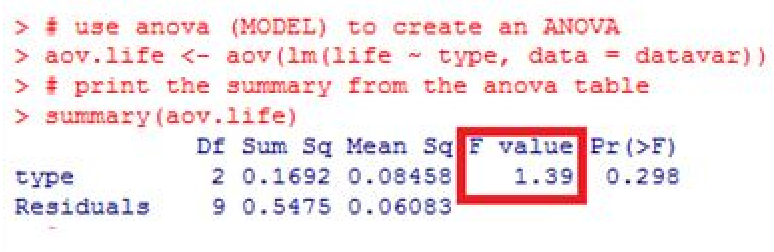
```
> mean(datavar$life)
[1] 5.183333
```

We find that the mean life time is 5.2 hours.

Let us form the hypothesis:

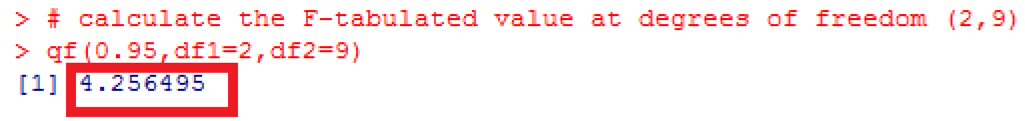
H_0 (Null): The three types of battery have the same mean life time, estimated by 5.2 hours.

H_1 (Alternate): The three types of battery have the different mean life time, estimated by 5.2 hours.



```
> # use anova (MODEL) to create an ANOVA
> aov.life <- aov(lm(life ~ type, data = datavar))
> # print the summary from the anova table
> summary(aov.life)
            Df Sum Sq Mean Sq F value Pr(>F)
type         2 0.1692 0.08458    1.39  0.298
Residuals    9 0.5475 0.06083
```

We now proceed to calculate the F (tabulated) value for degrees of freedom (df1= 2, df2 =9).



```
> # calculate the F-tabulated value at degrees of freedom (2,9)
> qf(0.95,df1=2,df2=9)
[1] 4.256495
```

Conclusion:

We find Fcal (=1.39) < Ftab(=4.26) at 5 % level of an F-distribution.

Hence, we accept the null hypothesis that the three types of battery have the same mean life time, estimated by 5.2 hours.

Example 2.2:

A fast food franchise is test marketing 3 new menu items. To find out if they have the same popularity, 6 franchisee restaurants are randomly chosen for participation in the study. In accordance with the randomized block design, each restaurant will be test marketing all 3 new menu items.

The testing order of the menu items for each restaurant is randomly assigned as well.

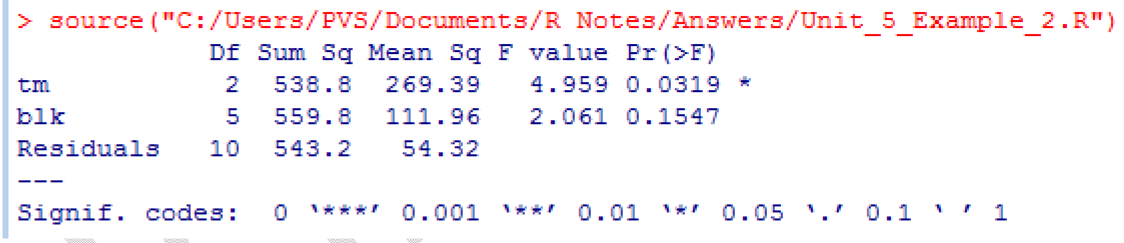
Suppose each row in the following table represents the sale figures of the 3 new menu in a restaurant after a week of test marketing. At 0.5 level of significance, test whether the mean sale volume for the 3 new menu items are equal.

Item1	Item2	Item3
31	27	24
31	28	31
45	29	46
21	18	48
42	36	46
32	17	40

Solution

Let us form the hypotheses for this problem.

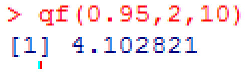
H_0 (Null): Mean sales volume of the new menu items are all equal.
H_1 : Mean sales volume of the new menu items are significantly different.



```
> source("C:/Users/PVS/Documents/R Notes/Answers/Unit_5_Example_2.R")
             Df Sum Sq Mean Sq F value Pr(>F)  
tm            2  538.8  269.39   4.959 0.0319 *
blk           5  559.8  111.96   2.061 0.1547  
Residuals    10  543.2   54.32                 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

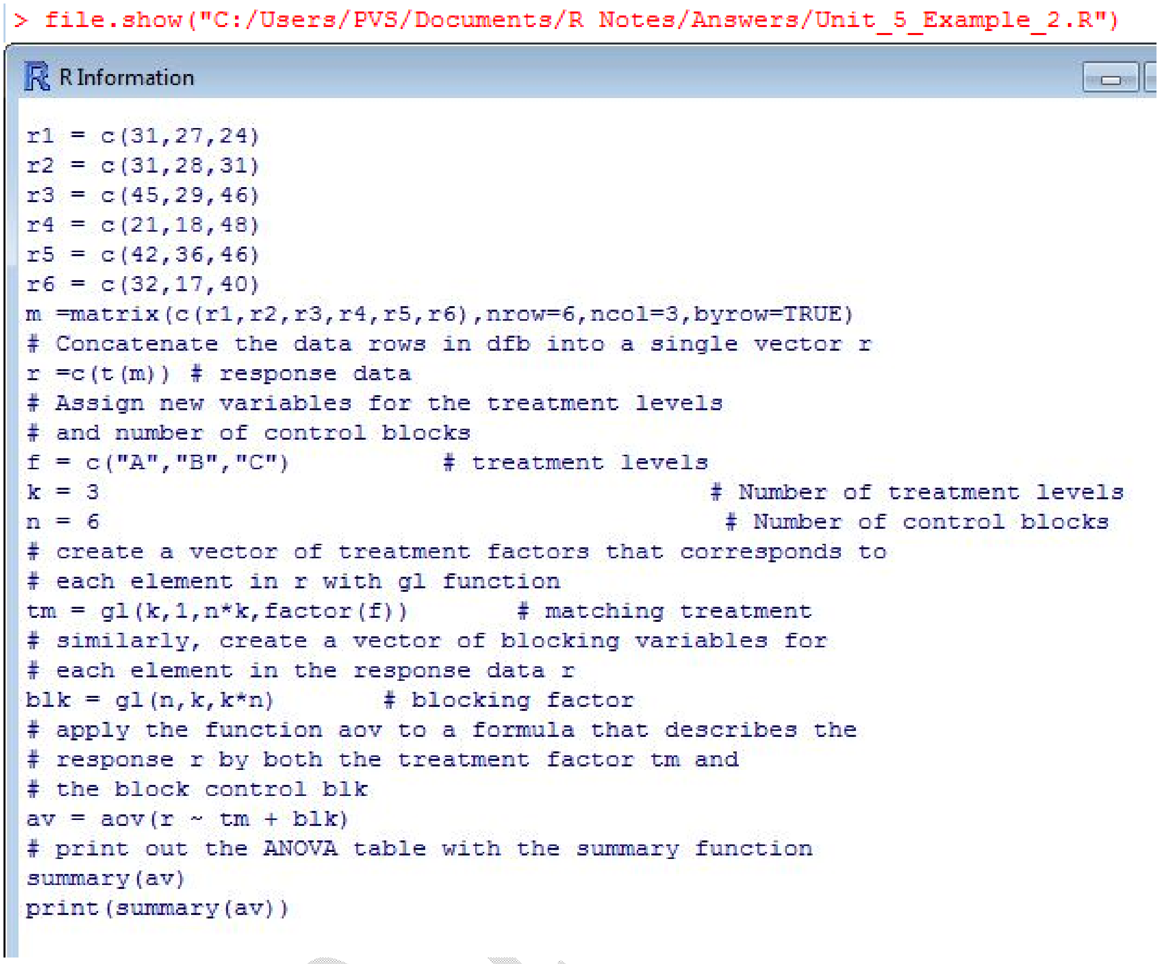
Since the p-value of 0.032 is less than the 0.05 significance level, we reject the null hypothesis that the mean sales volume of the new menu items are all equal.

We find the F (tabulated) vale at 5 % significance level for (2,5) degrees of freedom is 4.102821. Here Fcal (=4.959) > Ftab (=4.102821). Hence we reject the Null hypothesis.



```
> qf(0.95,2,10)
[1] 4.102821
```

The source code of this solution is written in the file Unit_5_Example_2.R and stored in the R workspace.



```
> file.show("C:/Users/PVS/Documents/R Notes/Answers/Unit_5_Example_2.R")
```

```
r1 = c(31,27,24)
r2 = c(31,28,31)
r3 = c(45,29,46)
r4 = c(21,18,48)
r5 = c(42,36,46)
r6 = c(32,17,40)
m =matrix(c(r1,r2,r3,r4,r5,r6),nrow=6,ncol=3,byrow=TRUE)
# Concatenate the data rows in dfb into a single vector r
r =c(t(m)) # response data
# Assign new variables for the treatment levels
# and number of control blocks
f = c("A","B","C")          # treatment levels
k = 3                                            # Number of treatment levels
n = 6                                             # Number of control blocks
# create a vector of treatment factors that corresponds to
# each element in r with gl function
tm = gl(k,1,n*k,factor(f))        # matching treatment
# similarly, create a vector of blocking variables for
# each element in the response data r
blk = gl(n,k,k*n)        # blocking factor
# apply the function aov to a formula that describes the
# response r by both the treatment factor tm and
# the block control blk
av = aov(r ~ tm + blk)
# print out the ANOVA table with the summary function
summary(av)
print(summary(av))
```

Explanation:

Line 1 – 7 *Create a matrix, m to contain the sales figures*

```
r1 = c(31,27,24)
r2 = c(31,28,31)
r3 = c(45,29,46)
r4 = c(21,18,48)
r5 = c(42,36,46)
r6 = c(32,17,40)
m =matrix(c(r1,r2,r3,r4,r5,r6),nrow=6,ncol=3,byrow=TRUE)
```

Line 9 Concatenate the data rows in m into a single vector.

```
r =c(t(m)) # response data
```

Line 12 – 14 *Assign new variables for the treatment levels and number of control blocks.*

```
f = c("A","B","C")              # treatment levels
k = 3                           # Number of treatment levels
```

```
n = 6                                   # Number of control blocks
```

Line 17 *Create a vector of treatment factors that corresponds to each that corresponds to the each element in r with the **gl** function.*

***gl** () function generates factors by specifying the pattern of their levels.*

***Syntax** gl (n, k, length = n * k, labels = 1:n, ordered = FALSE)*
where

n	*number of levels*
k	*number of replications*
length	*length of the result*
labels	*labels for the resulting factor levels*
ordered	*whether the result should be ordered or not*

```
tm = gl(k,1,n*k, factor(f))      # matching treatment
```

Line 20 *Similarly, create a vector of blocking factors for each element in the response data r.*

```
blk = gl(n,k,k*n)      # blocking factor
```

Line 24 *Apply the function **aov** to a formula that describes the response r by both the treatment factor tm and the block control blk.*

```
av = aov(r ~ tm + blk)
```

Line 26 *Print out the ANOVA table with the summary function*

```
summary(av)
```

Example 2.3: Factorial Design

A fast food franchise is test marketing 3 new items in both East and West Coasts of Continental United States. To find out if they enjoy the same popularity, 12 franchisee restaurants from each Coast are randomly chosen for participation in the study. In accordance with the factorial design, within the 12 restaurants from East Coast, 4 are randomly chosen to test market the first new item, another 4 for the second menu item, and the remaining 4 for the last menu item. The 12 restaurants from West Coast are arranged likewise.

Suppose the following tables represent the sales figures of the 3 new menu items after a week of test marketing. Each row in the upper table represents the sales figures of 3 different East Coast restaurants. The lower half represents West Coast restaurants.

At 0.05 level of significance, test whether the mean sales volume for the new menu items are all equal.

Decide also whether the mean sales volumes of the two coastal regions differ.

Explain the significant differences, if any.

East Coast

Region	Item 1	Item 2	Item 3
E1	25	39	36
E2	36	42	24
E3	31	39	28
E4	26	35	29

Wast Coast

Region	Item 1	Item 2	Item 3
W1	51	43	42
W2	47	39	36
W3	47	53	32
W4	52	46	33

Solution:

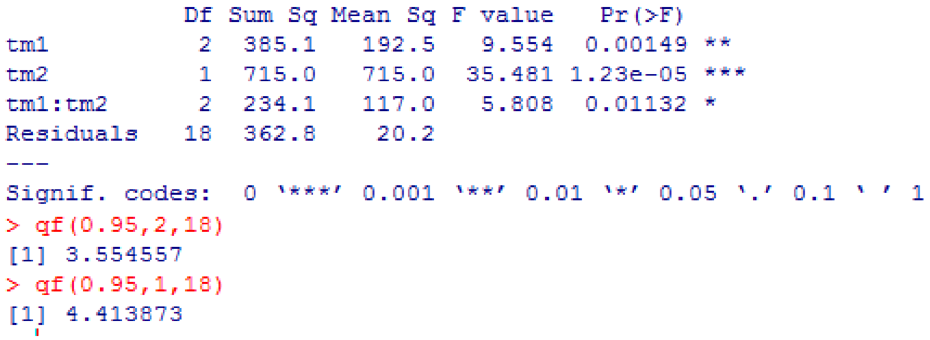
Let us form the hypothesis for this problem.

H_0 (Null): Mean sales volume for the new menu items are all equal.
Mean sales volume of the coastal regions differ.
There is no significant difference in means sales volume due to interaction of menu items and location of coastal regions

H1 (Alternate): Mean sales volume of the coastal regions are same.

Mean sales volume of the coastal regions differ.

There is significant difference in means sales volume due to interaction of menu items and location of coastal regions



```
              Df Sum Sq Mean Sq F value    Pr(>F)    
tm1            2  385.1   192.5   9.554   0.00149 ** 
tm2            1  715.0   715.0  35.481  1.23e-05 ***
tm1:tm2        2  234.1   117.0   5.808   0.01132 *  
Residuals     18  362.8    20.2                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> qf(0.95,2,18)
[1] 3.554557
> qf(0.95,1,18)
[1] 4.413873
```

Conclusion:

1 **For new menu items:**

Since the p-value of menu items (0.00149) is less than 0.05 at 5% significance level and Fcal (9.554) > Ftab (3.554557), we reject the null hypothesis.
Hence, mean sales volume of the new menu items differ.

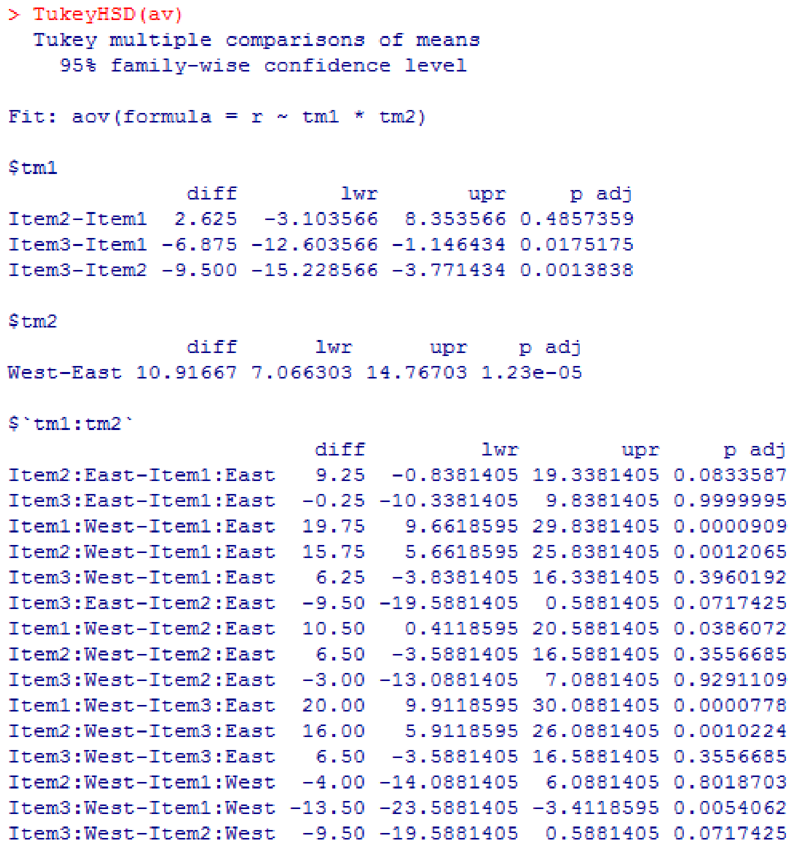
2 **For Coastal Regions:**

Since the p-value of menu items (1.2e-05) is less than 0.05 at 5% significance level and Fcal (35.481) > Ftab (4.413873), we reject the null hypothesis.
Hence, mean sales volume of the coastal regions differ.

3 **For Interaction of new menu items and Coastal Regions:**

Since the p-value of menu items (0.01132) is less than 0.05 at 5% significance level and Fcal (5.808) > Ftab (3.554557), we reject the null hypothesis.
Hence, there is significant difference in means sales volume due to interaction of menu items location of coastal regions

TukeyHSD tells us whether any particular sample mean significantly differs from any particular other.



```
> TukeyHSD(av)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = r ~ tm1 * tm2)

$tm1
              diff        lwr       upr     p adj
Item2-Item1  2.625  -3.103566  8.353566 0.4857359
Item3-Item1 -6.875 -12.603566 -1.146434 0.0175175
Item3-Item2 -9.500 -15.228566 -3.771434 0.0013838

$tm2
              diff      lwr      upr    p adj
West-East 10.91667 7.066303 14.76703 1.23e-05

$`tm1:tm2`
                         diff         lwr        upr     p adj
Item2:East-Item1:East    9.25  -0.8381405 19.3381405 0.0833587
Item3:East-Item1:East   -0.25 -10.3381405  9.8381405 0.9999995
Item1:West-Item1:East   19.75   9.6618595 29.8381405 0.0000909
Item2:West-Item1:East   15.75   5.6618595 25.8381405 0.0012065
Item3:West-Item1:East    6.25  -3.8381405 16.3381405 0.3960192
Item3:East-Item2:East   -9.50 -19.5881405  0.5881405 0.0717425
Item1:West-Item2:East   10.50   0.4118595 20.5881405 0.0386072
Item2:West-Item2:East    6.50  -3.5881405 16.5881405 0.3556685
Item3:West-Item2:East   -3.00 -13.0881405  7.0881405 0.9291109
Item1:West-Item3:East   20.00   9.9118595 30.0881405 0.0000778
Item2:West-Item3:East   16.00   5.9118595 26.0881405 0.0010224
Item3:West-Item3:East    6.50  -3.5881405 16.5881405 0.3556685
Item2:West-Item1:West   -4.00 -14.0881405  6.0881405 0.8018703
Item3:West-Item1:West  -13.50 -23.5881405 -3.4118595 0.0054062
Item3:West-Item2:West   -9.50 -19.5881405  0.5881405 0.0717425
```

From the above output, we conclude that

a) *For Item 2 - Item3 interaction, P-value (0.0013838) is less than 0.05, at 5 % level.*

Here we reject the null hypothesis and conclude that there is significant difference in the mean sales volume between Item 2 and Item 3.

b) *For West - Item 1 and East Item 1, p- value (0.0000909) is very much less than 0.05.*

Here also we reject the null hypothesis. The mean sales volume of Item 1 sold in West Coast significantly differs from the mean sales volume of Item 1 sold in East Coast.

c) *For West - Item 2 and East Item 1, p- value (0.0012065) is very much less than 0.05.*

Here also we reject the null hypothesis. The mean sales volume of Item 2 sold in West Coast significantly differs from the mean sales volume of Item 1 sold in East Coast.

d) *For West - Item 1 and East Item 2, p-value (0.0386072) is very much less than 0.05.*

Here also we reject the null hypothesis. The mean sales volume of Item 1 sold in West Coast significantly differs from the mean sales volume of Item 2 sold in East Coast.

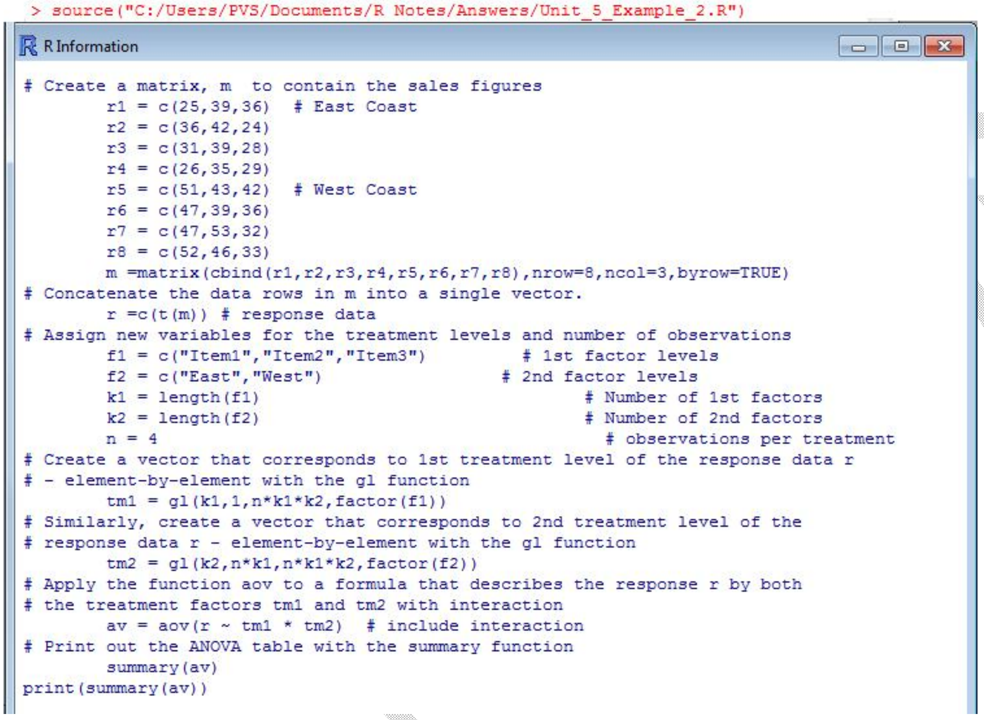
e) *For West - Item 1 and East Item 3, p-value (0.0000708) is very much less than 0.05.*

Here also we reject the null hypothesis. The mean sales volume of Item 1 sold in West Coast significantly differs from the mean sales volume of Item 3 sold in East Coast.

f) *For West - Item 2 and East Item 3, p-value (0.0010224) is very much less than 0.05. Here also we reject the null hypothesis. The mean sales volume of Item 2 sold in West Coast significantly differs from the mean sales volume of Item 3 sold in East Coast.*

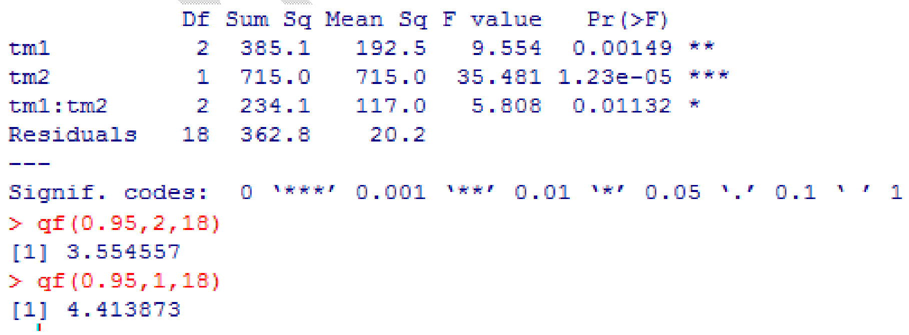
g) *For West - Item 2 and East Item 3, p-value (0.0010224) is very much less than 0.05. Here also we reject the null hypothesis. The mean sales volume of Item 2 sold in West Coast significantly differs from the mean sales volume of Item 3 sold in East Coast.*

```
> source("C:/Users/PVS/Documents/R Notes/Answers/Unit_5_Example_2.R")
```



```
# Create a matrix, m  to contain the sales figures
        r1 = c(25,39,36)  # East Coast
        r2 = c(36,42,24)
        r3 = c(31,39,28)
        r4 = c(26,35,29)
        r5 = c(51,43,42)  # West Coast
        r6 = c(47,39,36)
        r7 = c(47,53,32)
        r8 = c(52,46,33)
        m =matrix(cbind(r1,r2,r3,r4,r5,r6,r7,r8),nrow=8,ncol=3,byrow=TRUE)
# Concatenate the data rows in m into a single vector.
        r =c(t(m)) # response data
# Assign new variables for the treatment levels and number of observations
        f1 = c("Item1","Item2","Item3")        # 1st factor levels
        f2 = c("East","West")                # 2nd factor levels
        k1 = length(f1)                              # Number of 1st factors
        k2 = length(f2)                              # Number of 2nd factors
        n = 4                                          # observations per treatment
# Create a vector that corresponds to 1st treatment level of the response data r
# - element-by-element with the gl function
        tm1 = gl(k1,1,n*k1*k2,factor(f1))
# Similarly, create a vector that corresponds to 2nd treatment level of the
# response data r - element-by-element with the gl function
        tm2 = gl(k2,n*k1,n*k1*k2,factor(f2))
# Apply the function aov to a formula that describes the response r by both
# the treatment factors tm1 and tm2 with interaction
        av = aov(r ~ tm1 * tm2)  # include interaction
# Print out the ANOVA table with the summary function
        summary(av)
print(summary(av))
```

ANOVA table



```
            Df Sum Sq Mean Sq F value   Pr(>F)    
tm1          2  385.1   192.5   9.554  0.00149 ** 
tm2          1  715.0   715.0  35.481 1.23e-05 ***
tm1:tm2      2  234.1   117.0   5.808  0.01132 *  
Residuals   18  362.8    20.2                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> qf(0.95,2,18)
[1] 3.554557
> qf(0.95,1,18)
[1] 4.413873
```

Explanation:

Line 1 - 10 *We create a matrix to contain the sales figures.*

```
# Create a matrix, m  to contain the sales figures
r1 = c(25,39,36)  # East Coast
r2 = c(36,42,24)
r3 = c(31,39,28)
r4 = c(26,35,29)
r5 = c(51,43,42)  # West Coast
r6 = c(47,39,36)
r7 = c(47,53,32)
r8 = c(52,46,33)
m =matrix(cbind(r1,r2,r3,r4,r5,r6,r7,r8),nrow=8,ncol=3,byrow=TRUE)
```

Line 11 - 12 *Create a vector, r by concatenating the data rows of matrix m.*

```
# Concatenate the data rows in m into a single vector.
r =c(t(m)) # response data
```

Line 13 - 18 *Create variables for first factor level (f1), second factor level (f2), lengths of f1, f2 and number of observations per treatment.*

```
# Assign new variables for the treatment levels and number of observations
f1 = c("Item1","Item2","Item3")       # 1st factor levels
f2 = c("East","West")                  # 2nd factor levels
k1 = length(f1)                    # Number of 1st factors
k2 = length(f2)                    # Number of 2nd factors
n = 4                                 # observations per treatment
```

Line 19 – 21 *We create a vector, tm1 corresponding to the first treatment level of the response vector, r by using the function,* ***gl ().***

As explained in the previous example, ***gl*** *() function generates factors by specifying the pattern of their levels.*

```
# Create a vector that corresponds to 1st treatment level of the response data r
# - element-by-element with the gl function
tm1 = gl(k1,1,n*k1*k2,factor(f1))
```

Line 22 – 24 *Similarly, we create a vector tm2 corresponding to 2nd level of response vector, r by using the function* ***gl().***

```
# Similarly, create a vector that corresponds to 2nd treatment level of the
# response data r - element-by-element with the gl function
tm2 = gl(k2,n*k1,n*k1*k2,factor(f2))
```

Line 25 – 27 *Apply the function* ***aov*** *to a formula that describes the response r by both the treatment factors tm1 and tm2 along with their interaction.*

*To analyze the interaction between two independent variables tm1 and tm2, use r ~ tm1 * tm2.*

If the interactions are not important, then we use r ~ tm1 + tm2.

```
# Apply the function aov to a formula that describes the response r by both
# the treatment factors tm1 and tm2 with interaction
av = aov(r ~ tm1 * tm2)  # include interaction
```

Line 28 – 30 *Print the summary of ANOVA table.*

```
# Print out the ANOVA table with the summary function
summary(av)
print(summary(av))
```

Regression

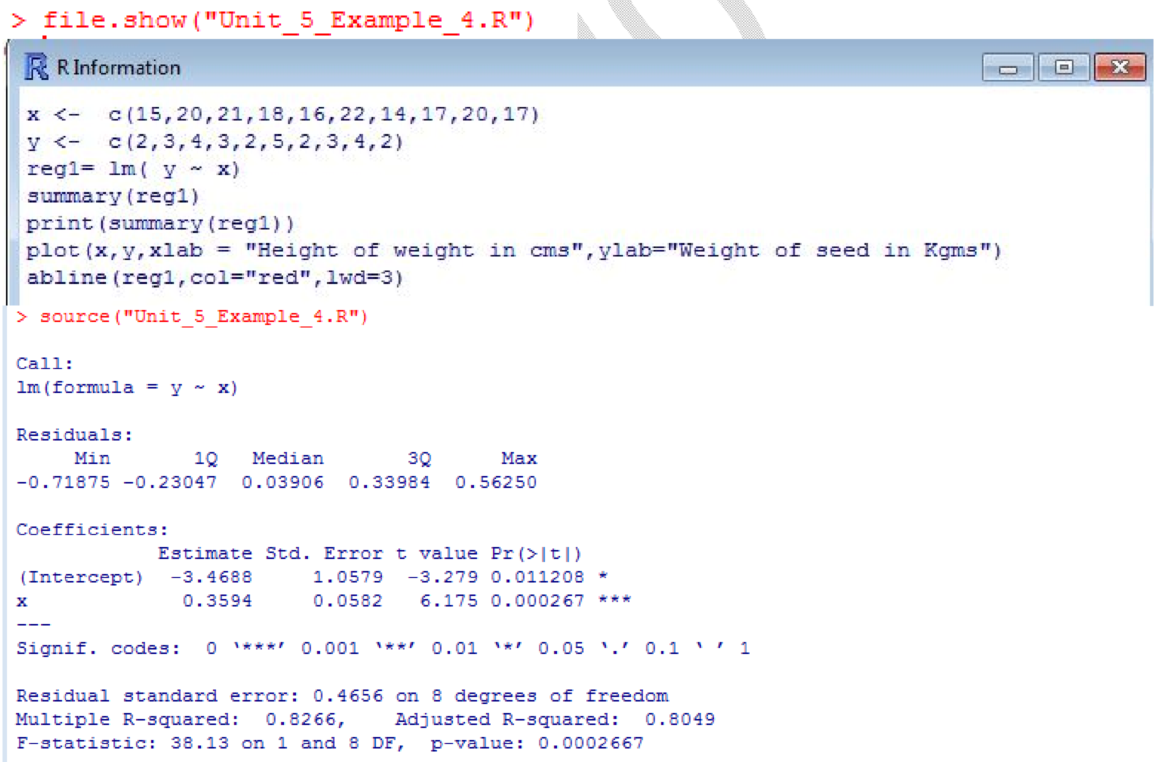
Example 2.4:

Obtain the regression of height of plant on weight of seed, and the R^2

Height of plant (in cms)	Weight of seed in (kgm)
15	2
20	3
21	4
18	3
16	2
22	5
14	2
17	3
20	4
17	2

Solution:

We shall look into the source code for solving the above problem.



```
> file.show("Unit_5_Example_4.R")
```

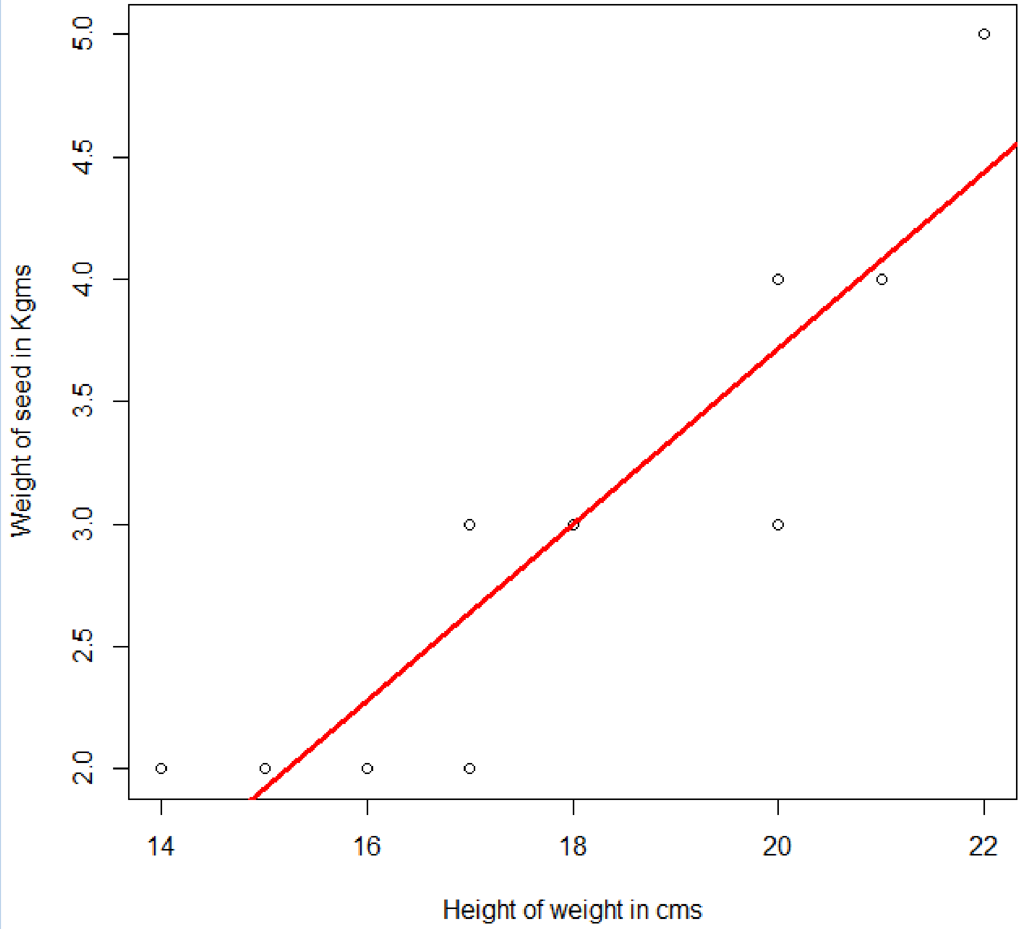
```
x <-  c(15,20,21,18,16,22,14,17,20,17)
y <-  c(2,3,4,3,2,5,2,3,4,2)
reg1= lm( y ~ x)
summary(reg1)
print(summary(reg1))
plot(x,y,xlab = "Height of weight in cms",ylab="Weight of seed in Kgms")
abline(reg1,col="red",lwd=3)
```

```
> source("Unit_5_Example_4.R")

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.71875 -0.23047  0.03906  0.33984  0.56250 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.4688     1.0579  -3.279 0.011208 *  
x             0.3594     0.0582   6.175 0.000267 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.4656 on 8 degrees of freedom
Multiple R-squared:  0.8266,	Adjusted R-squared:  0.8049 
F-statistic: 38.13 on 1 and 8 DF,  p-value: 0.0002667
```

Regression equation y on x: y = 0.3594 x – 3.4688

Multiple R^2 = 0.8266. This means that 82 % of the variation is explained in this model.

Note: R is the coefficient of **multiple correlation,** a measure of how well a given variable can be predicted using a linear function of a set of other variables.



The above figure shows the scatter diagram and the regression line for this problem.

Regression – Curve fitting

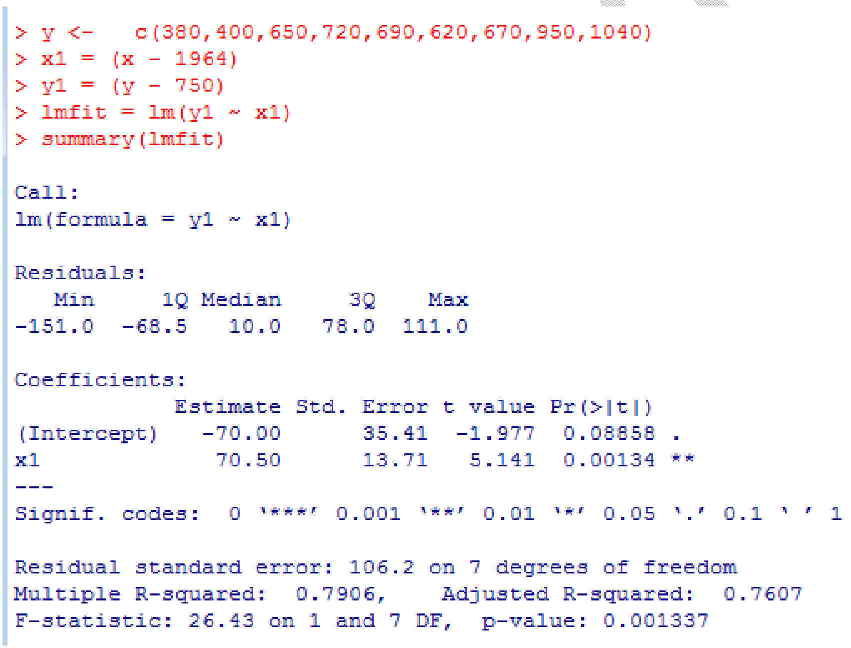
Example 2.5:

Obtain the linear regression line (Straight line – trend equation) and tabulate against each year after estimation of the trend and short term fluctuations.

Use X = (x- 1964) and Y = (y – 750) for plotting year values.

x -> Year	y ->Value
1960	380
1961	400
1962	650
1963	720
1964	690
1965	620
1966	670
1967	950
1968	1040

Solution:

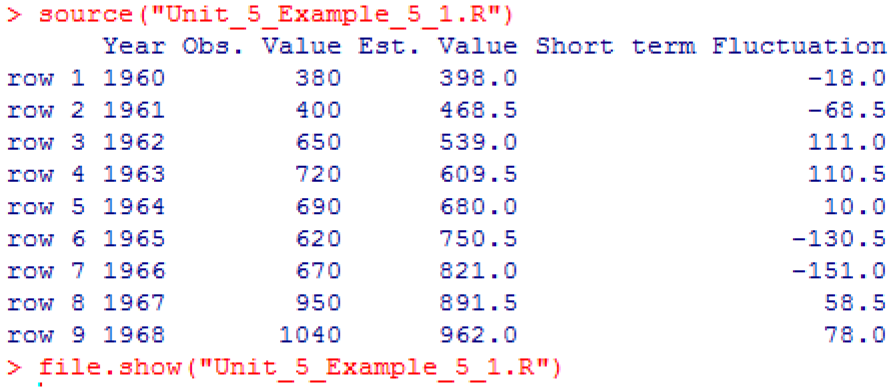


```
> y <-   c(380,400,650,720,690,620,670,950,1040)
> x1 = (x - 1964)
> y1 = (y - 750)
> lmfit = lm(y1 ~ x1)
> summary(lmfit)

Call:
lm(formula = y1 ~ x1)

Residuals:
   Min     1Q Median     3Q    Max 
-151.0  -68.5   10.0   78.0  111.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   -70.00      35.41  -1.977  0.08858 . 
x1             70.50      13.71   5.141  0.00134 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 106.2 on 7 degrees of freedom
Multiple R-squared:  0.7906,    Adjusted R-squared:  0.7607 
F-statistic: 26.43 on 1 and 7 DF,  p-value: 0.001337
```

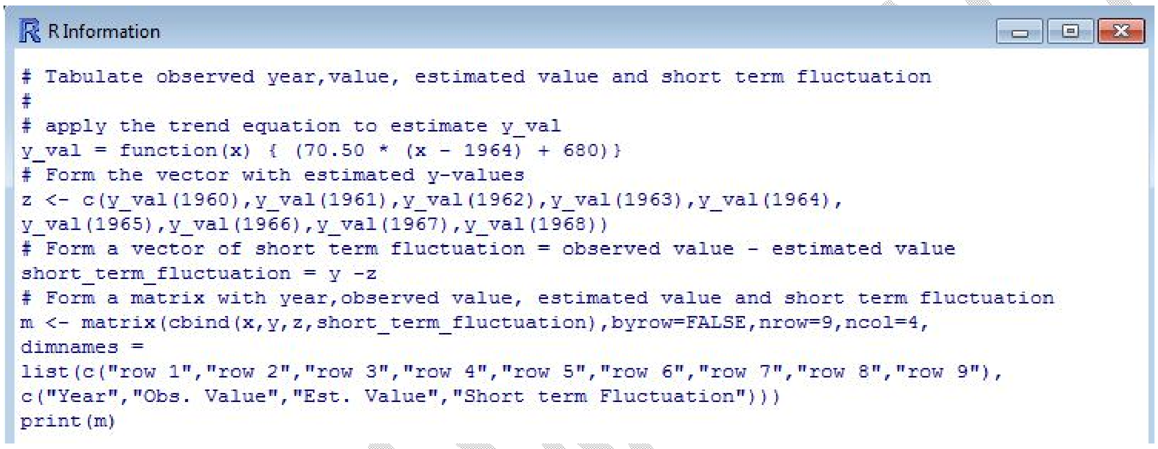
Trend line is y – 750 = (70.50 * (x – 1964)) – 70

y = (70.50 * (x – 1964) -+ 680

Trend value for 1960, 1961, 1962,… are obtained by applying the above regression equation by supplying each value of x , i.e. year 1960, 1961, 1962. Short term fluctuation = y – z, where y is the observed value and z is the trend value.



```
> source("Unit_5_Example_5_1.R")
      Year Obs. Value Est. Value Short term Fluctuation
row 1 1960        380      398.0                  -18.0
row 2 1961        400      468.5                  -68.5
row 3 1962        650      539.0                  111.0
row 4 1963        720      609.5                  110.5
row 5 1964        690      680.0                   10.0
row 6 1965        620      750.5                 -130.5
row 7 1966        670      821.0                 -151.0
row 8 1967        950      891.5                   58.5
row 9 1968       1040      962.0                   78.0
> file.show("Unit_5_Example_5_1.R")
```



```
# Tabulate observed year,value, estimated value and short term fluctuation
#
# apply the trend equation to estimate y_val
y_val = function(x) { (70.50 * (x - 1964) + 680)}
# Form the vector with estimated y-values
z <- c(y_val(1960),y_val(1961),y_val(1962),y_val(1963),y_val(1964),
y_val(1965),y_val(1966),y_val(1967),y_val(1968))
# Form a vector of short term fluctuation = observed value - estimated value
short_term_fluctuation = y -z
# Form a matrix with year,observed value, estimated value and short term fluctuation
m <- matrix(cbind(x,y,z,short_term_fluctuation),byrow=FALSE,nrow=9,ncol=4,
dimnames =
list(c("row 1","row 2","row 3","row 4","row 5","row 6","row 7","row 8","row 9"),
c("Year","Obs. Value","Est. Value","Short term Fluctuation")))
print(m)
```

Example 2.6:

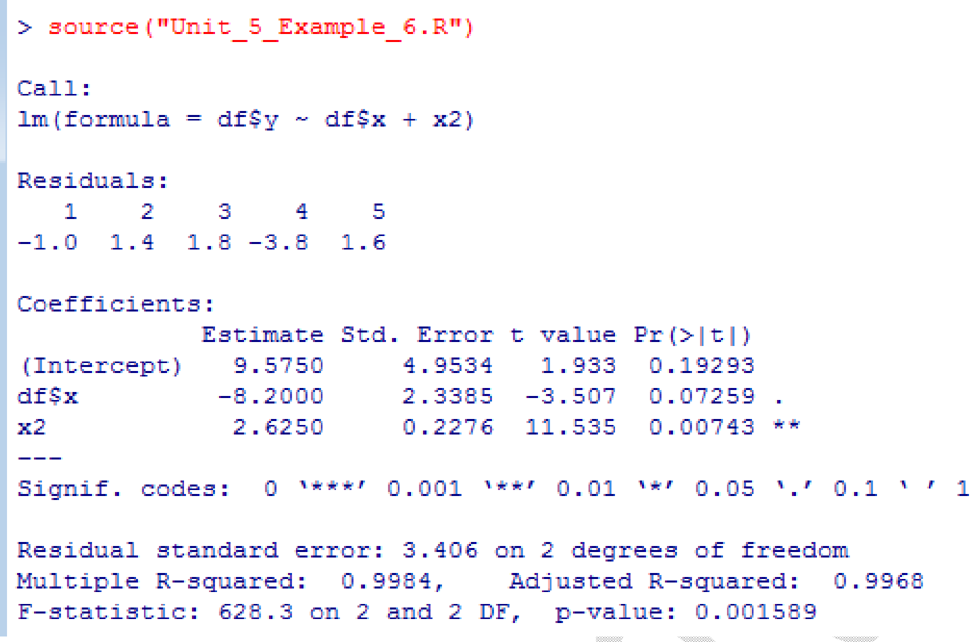
Fit a polynomial of the second degree for y in terms of x to the following data.

X	Y
1	3
3	10
5	36
7	77
9	150

Solution:

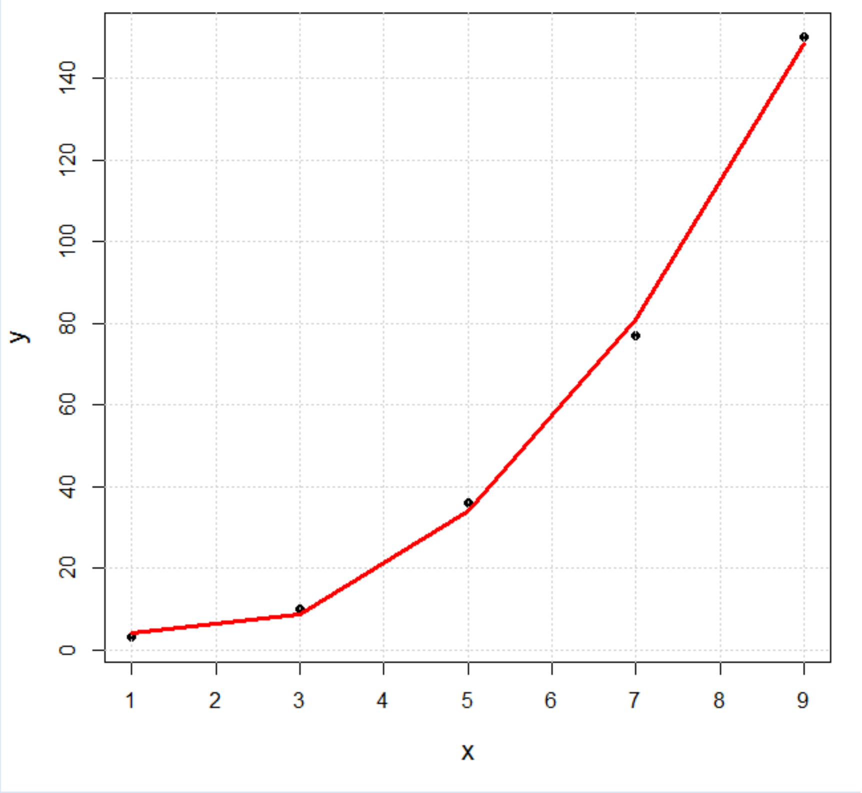
- We fit a model that is quadratic in nature. We create a variable, x2 which is the square of the variable, x.
- Note the syntax involved in fitting a quadratic model with two or more predictors. We include each predictor and put a plus sign between them.
- Our quadratic model is basically a linear model in two variables, one of which is the square of the other.
- This quadratic model explains 99.84 % of variance.
- Now we plot the quadratic model by setting up a grid of values for x running from 1 to 9 with increments of 2.

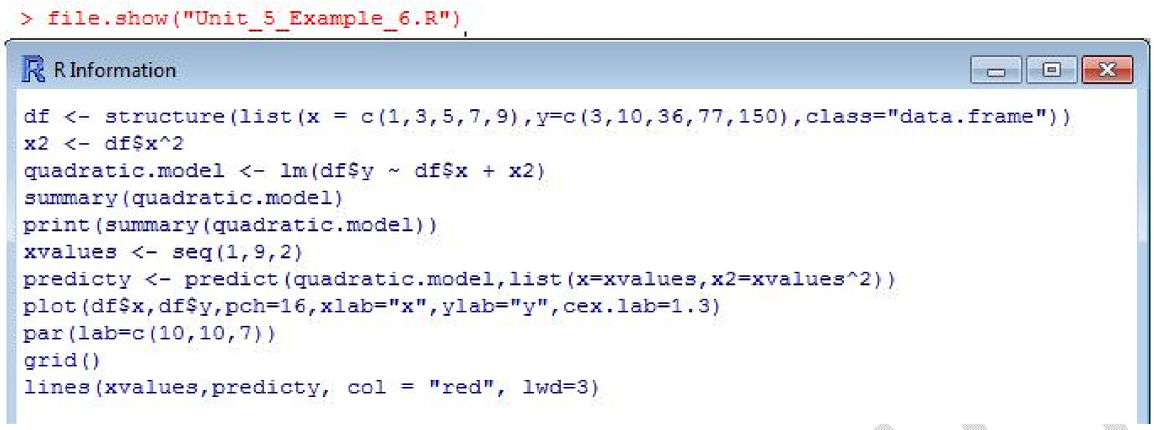
- Now add the quadratic model to the plot using the lines() command.



```
> source("Unit_5_Example_6.R")

Call:
lm(formula = df$y ~ df$x + x2)

Residuals:
   1    2    3    4    5 
-1.0  1.4  1.8 -3.8  1.6 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   9.5750     4.9534   1.933  0.19293   
df$x         -8.2000     2.3385  -3.507  0.07259 . 
x2            2.6250     0.2276  11.535  0.00743 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.406 on 2 degrees of freedom
Multiple R-squared:  0.9984,    Adjusted R-squared:  0.9968 
F-statistic: 628.3 on 2 and 2 DF,  p-value: 0.001589
```

We see that the quadratic model appears to fit the data very well in the graph shown below:





```
> file.show("Unit_5_Example_6.R")
```

```
df <- structure(list(x = c(1,3,5,7,9),y=c(3,10,36,77,150),class="data.frame"))
x2 <- df$x^2
quadratic.model <- lm(df$y ~ df$x + x2)
summary(quadratic.model)
print(summary(quadratic.model))
xvalues <- seq(1,9,2)
predicty <- predict(quadratic.model,list(x=xvalues,x2=xvalues^2))
plot(df$x,df$y,pch=16,xlab="x",ylab="y",cex.lab=1.3)
par(lab=c(10,10,7))
grid()
lines(xvalues,predicty, col = "red", lwd=3)
```

2.2. Exercises

Exercise 2.1:

An experiment is conducted to compare three diets A, B and C on growth rate in mice, grouped according to their weight to different blocks. Within each block the animals were then assigned at random to one of the diets. The results (gm / week) are shown in the table.

Treatment	Block 1	Block 2	Block 3	Treatment Totals
A	10	12	7	29
B	7	8	5	20
C	5	6	4	15
Block totals	22	26	16	Grand Total 64

Find out if
a. there is any significant difference in growth due to blocking
b. there is any significant difference in growth due to treatment

Exercise 2.2:

An experiment is conducted to compare the effect four different chemicals (A,B,C,D) in producing water resistance (y) in textiles. A strip of material, randomly selected from each bolt, is cut into four pieces (samples); the pieces are randomly assigned to receive one of the four chemical treatments. This process is replicated three times producing a Randomized Block design.

Moisture resistance (y) were measured for each of the samples. (Low readings indicate low moisture penetration).

Analyze the data to decide whether, at the 5% level, there are significant differences

a) between chemicals
b) between blocks

The data is given in the table below:

9.9	C
10.1	A
11.4	B
12.1	D

13.4	D
12.9	B
12.2	A
12.3	C

12.7	B
12.9	D
11.4	C
11.9	A

Exercise 2.3:

A research study was conducted to examine the clinical efficacy of a new antidepressant. Depressed patients were randomly assigned to one of three groups: a placedo group, a group that received a low dose of the drug, and a group that received a moderate dose of the drug. After four weeks of treatment, the patients completed the Beck Depression Inventory. The higher the score, the more depressed the patient. The data are presented below:

Compute the appropriate test.

Paclebo	Low Dose	High Dose
38	22	14
47	19	26
39	8	11
25	23	18
42	31	5

Exercise 2.4:

The heights (in cms) and weights (in kg) of a random sample of three adult male are shown in the following table.

a. Obtain the regression equation of x on y
b. Estimate the height when weight is 60 kgs.
c. Draw a scatter diagram and a regression line

Height (x)	Weight (y)
177	71
163	67
173	77
182	85
171	69
168	62
174	73
184	80

Exercise 2.5:

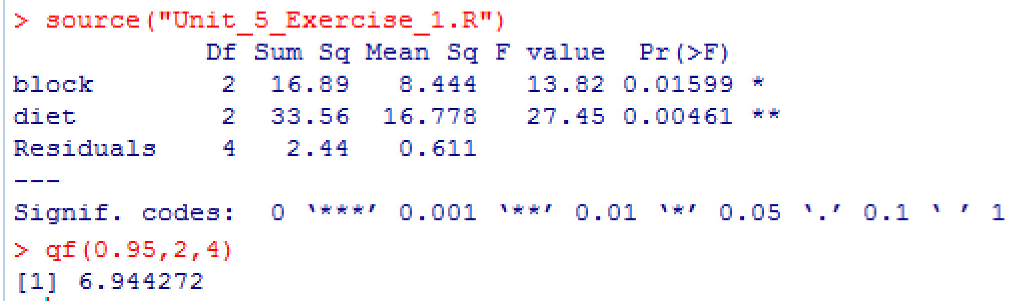
The profits (Rs. Lakhs) of a certain company in the x^{th} year of its life are given by the following table:

Fit a second degree parabola $y = a + bx + cx^2$ to this data.

Age (x)	Profit (y)
1	2.18
2	2.44
3	2.78
4	3.25

Answers

Exercise 2.1

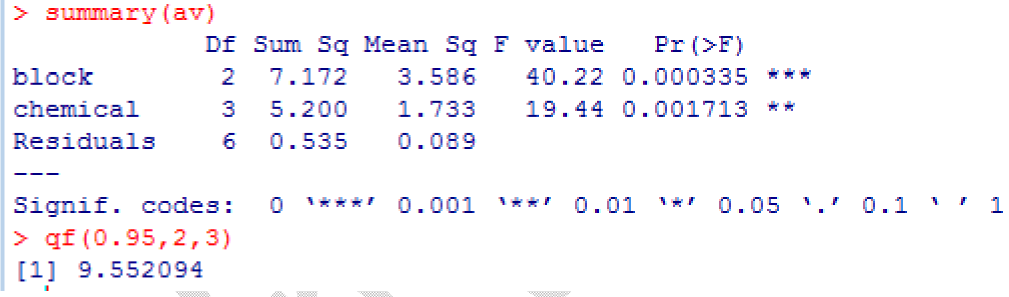


```
> source("Unit_5_Exercise_1.R")
            Df Sum Sq Mean Sq F value  Pr(>F)  
block        2  16.89   8.444   13.82 0.01599 *
diet         2  33.56  16.778   27.45 0.00461 **
Residuals    4   2.44   0.611                  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> qf(0.95,2,4)
[1] 6.944272
```

Analysis:

At the 5% significant level,
a. There is significant difference in growth due to blocking
b. There is significant difference in growth due to treatment (diet)

Exercise 2.2
Source(“Unit_5_Exercise_2.R”)



```
> summary(av)
            Df Sum Sq Mean Sq F value   Pr(>F)    
block        2  7.172   3.586   40.22 0.000335 ***
chemical     3  5.200   1.733   19.44 0.001713 ** 
Residuals    6  0.535   0.089                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> qf(0.95,2,3)
[1] 9.552094
```

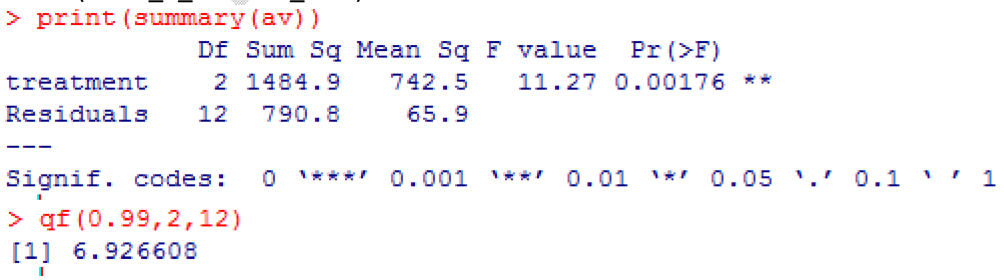
Analysis:

At the 5% significant level, there are significant differences

a) between chemicals
b) between blocks

Exercise 2.3

Source(“Unit_5_Exercise_3.R”)

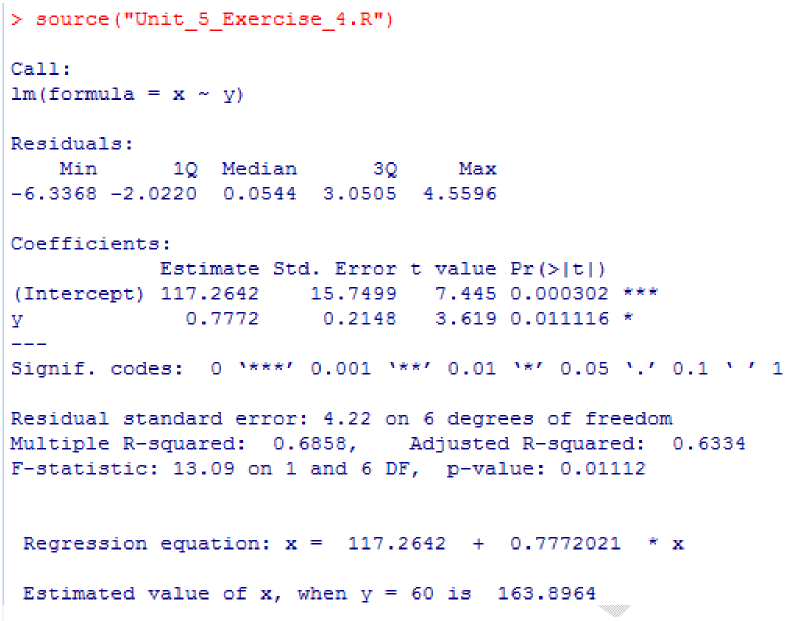


```
> print(summary(av))
            Df Sum Sq Mean Sq F value  Pr(>F)   
treatment    2 1484.9   742.5   11.27 0.00176 **
Residuals   12  790.8    65.9                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> qf(0.99,2,12)
[1] 6.926608
```

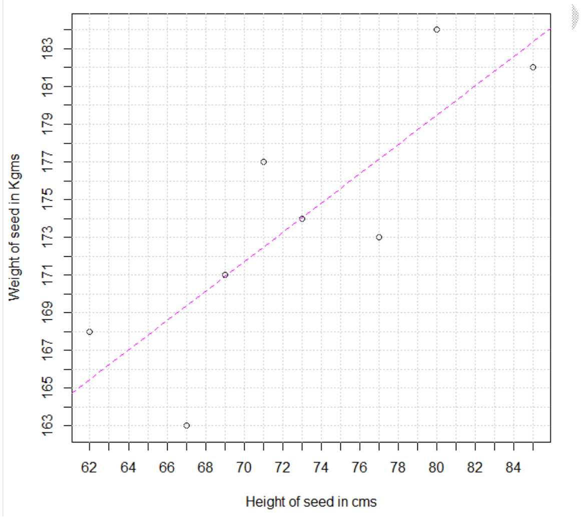
Analysis:

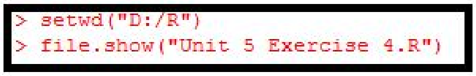
- 5 % is chosen as the significant level or the probability level because of the risk involved in wrongly classifying the drug works when it does not.
- Fcal = 11.27 at (2,12)
- Ftab = 6.926608 at (2,2) at probability level 0.0101
- Since Fcal > Ftab, we reject the null hypothesis that there is no significant difference in the treatment of taking placedo or low dose or high dose of the drug.

Exercise 2.4

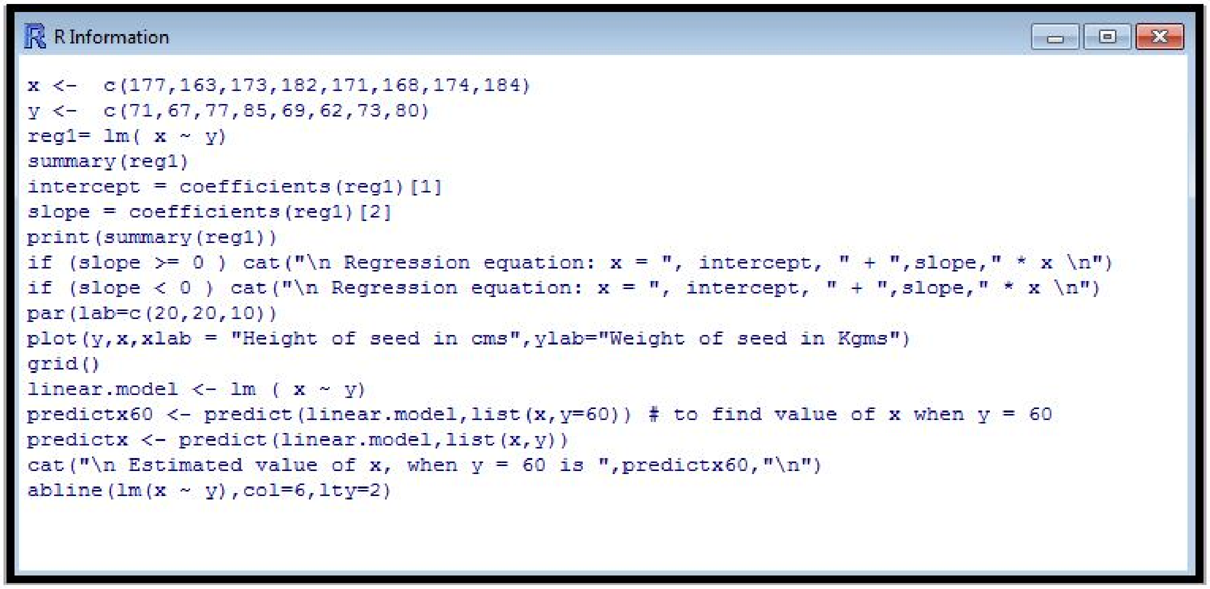


```
> source("Unit_5_Exercise_4.R")

Call:
lm(formula = x ~ y)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.3368 -2.0220  0.0544  3.0505  4.5596 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 117.2642    15.7499   7.445 0.000302 ***
y             0.7772     0.2148   3.619 0.011116 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.22 on 6 degrees of freedom
Multiple R-squared:  0.6858,    Adjusted R-squared:  0.6334 
F-statistic: 13.09 on 1 and 6 DF,  p-value: 0.01112


 Regression equation: x =  117.2642  +  0.7772021  * x

 Estimated value of x, when y = 60 is  163.8964
```



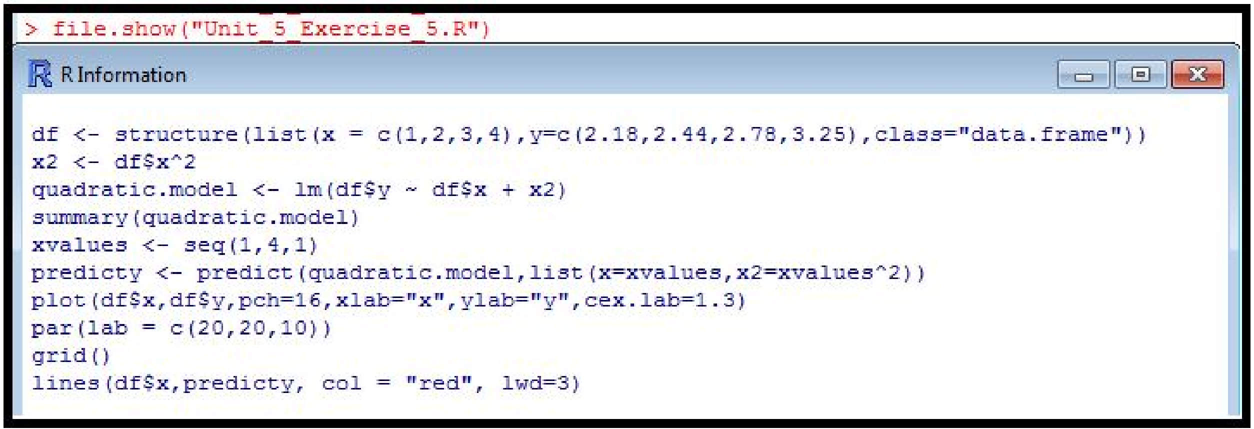


```
> setwd("D:/R")
> file.show("Unit 5 Exercise 4.R")
```



```
x <-  c(177,163,173,182,171,168,174,184)
y <-  c(71,67,77,85,69,62,73,80)
reg1= lm( x ~ y)
summary(reg1)
intercept = coefficients(reg1)[1]
slope = coefficients(reg1)[2]
print(summary(reg1))
if (slope >= 0 ) cat("\n Regression equation: x = ", intercept, " + ",slope," * x \n")
if (slope < 0 ) cat("\n Regression equation: x = ", intercept, " + ",slope," * x \n")
par(lab=c(20,20,10))
plot(y,x,xlab = "Height of seed in cms",ylab="Weight of seed in Kgms")
grid()
linear.model <- lm ( x ~ y)
predictx60 <- predict(linear.model,list(x,y=60)) # to find value of x when y = 60
predictx <- predict(linear.model,list(x,y))
cat("\n Estimated value of x, when y = 60 is ",predictx60,"\n")
abline(lm(x ~ y),col=6,lty=2)
```

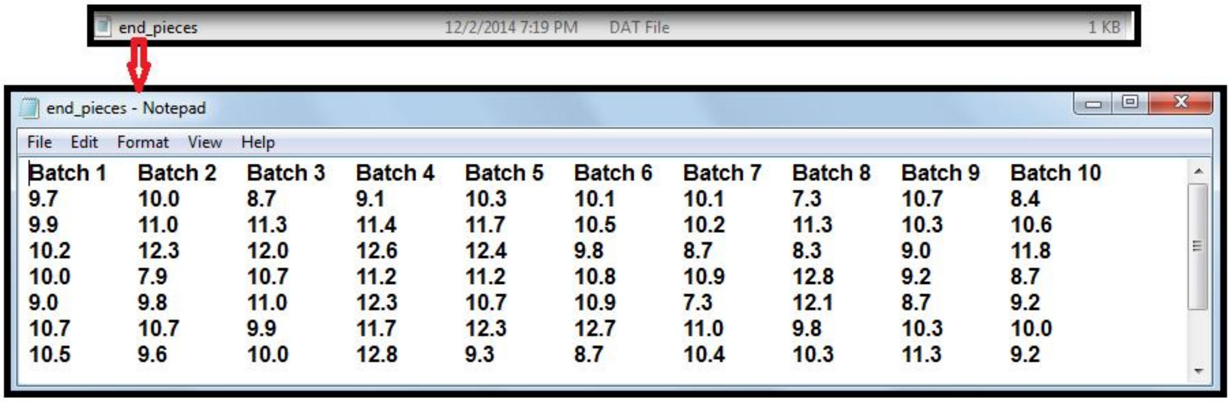
Exercise 2. 5



```
> file.show("Unit_5_Exercise_5.R")
```

```
df <- structure(list(x = c(1,2,3,4),y=c(2.18,2.44,2.78,3.25),class="data.frame"))
x2 <- df$x^2
quadratic.model <- lm(df$y ~ df$x + x2)
summary(quadratic.model)
xvalues <- seq(1,4,1)
predicty <- predict(quadratic.model,list(x=xvalues,x2=xvalues^2))
plot(df$x,df$y,pch=16,xlab="x",ylab="y",cex.lab=1.3)
par(lab = c(20,20,10))
grid()
lines(df$x,predicty, col = "red", lwd=3)
```

3. Annexure



```
Batch 1  Batch 2  Batch 3  Batch 4  Batch 5  Batch 6  Batch 7  Batch 8  Batch 9  Batch 10
9.7      10.0     8.7      9.1      10.3     10.1     10.1     7.3      10.7     8.4
9.9      11.0     11.3     11.4     11.7     10.5     10.2     11.3     10.3     10.6
10.2     12.3     12.0     12.6     12.4     9.8      8.7      8.3      9.0      11.8
10.0     7.9      10.7     11.2     11.2     10.8     10.9     12.8     9.2      8.7
9.0      9.8      11.0     12.3     10.7     10.9     7.3      12.1     8.7      9.2
10.7     10.7     9.9      11.7     12.3     12.7     11.0     9.8      10.3     10.0
10.5     9.6      10.0     12.8     9.3      8.7      10.4     10.3     11.3     9.2
```

Contents of the file: end_pieces.dat