Practical No. 1

What if analysis

- To find total expenditure and saving
- Make a data source in Microsoft Excel

Product	**Expenditure**	**Saving**
TC	500	60
TCC	600	70
WR	700	80
AR	800	90
K-TYPE	900	100
DC	1000	110
BK	1100	120
MM	1200	130
YKK	1300	140
MKL	1400	150

- In B13 type =SUM (B2:B11) and press enter
- In C14 type =C2-B13 and press enter

❖ To create scenario

- Select data tab, in the data tools select what-if analysis -> scenario manager
- In add scenario dialogue box, which appears. Click the add button.
- In the ‘Scenario Name’ type -> Original Budget.
- In ‘Changing Cells’ type -> B2:B11. Click ok.

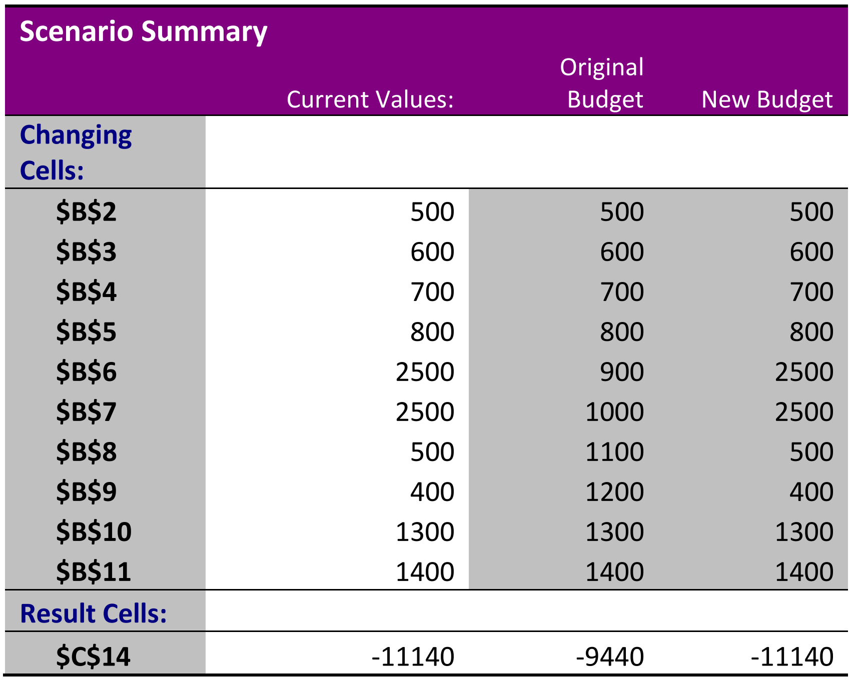
- Scenario values dialog box opens to retain original values. Click ok

❖ To create new scenario

- Click on add button again.
- In scenario and type New Budget. Then click ok.

We get scenario values dialogue box. Change the values in B6, B7,B8 and B9 to 2500,2500,500 and 400 respectively. Then click ok.

Output:



Scenario Summary	Current Values:	Original Budget	New Budget
Changing Cells:			
$B$2	500	500	500
$B$3	600	600	600
$B$4	700	700	700
$B$5	800	800	800
$B$6	2500	900	2500
$B$7	2500	1000	2500
$B$8	500	1100	500
$B$9	400	1200	400
$B$10	1300	1300	1300
$B$11	1400	1400	1400
Result Cells:			
$C$14	-11140	-9440	-11140

Practical No. 2

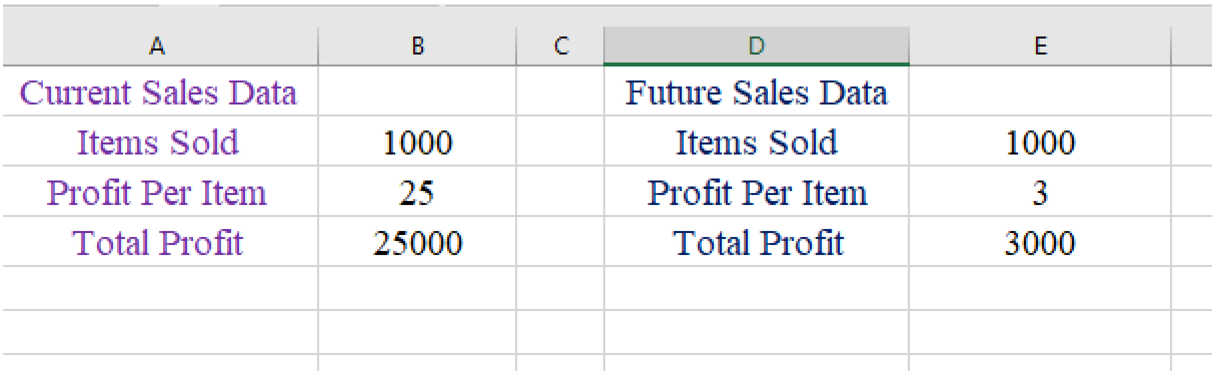
Goal Seek

- Select A1 and type Current Sales Data.
- Select A2 and type Items Sold.
- Select A3 and type Profit Per Item.
- Select A4 and type Total Profit.
- In B2 type 1000, in B3 type 25 and in B4 type =B2*B3.
- Select D1 and type Future Sales Data.
- Select D2 and type Items Sold.
- Select D3 and type Profit Per Item.
- Select D4 and type Total Profit.
- In E2 type 1000 and in E4 type =E2*E3.

❖ Goal Seek

  - Select data tab then in data tools select what if analysis then select goal seek.
  - In goal seek dialog box which appears, “In Set Cell” type E4.
  - In “To Value” type 3000.
  - In “By Changing Cells” type E3 then click ok.

Output

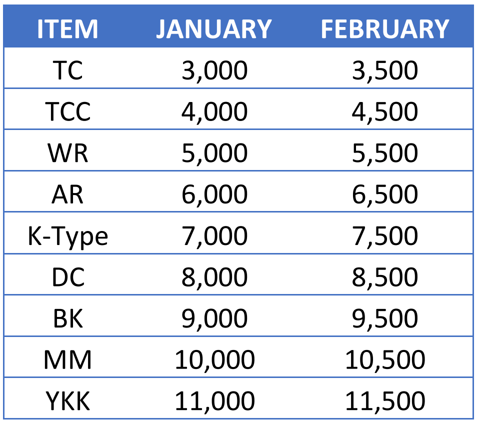


A	B	C	D	E
Current Sales Data			Future Sales Data	
Items Sold	1000		Items Sold	1000
Profit Per Item	25		Profit Per Item	3
Total Profit	25000		Total Profit	3000

Practical No. 3

Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system. Make data source in Microsoft Excel.

- Make a data source in Microsoft Excel



ITEM	JANUARY	FEBRUARY
TC	3,000	3,500
TCC	4,000	4,500
WR	5,000	5,500
AR	6,000	6,500
K-Type	7,000	7,500
DC	8,000	8,500
BK	9,000	9,500
MM	10,000	10,500
YKK	11,000	11,500

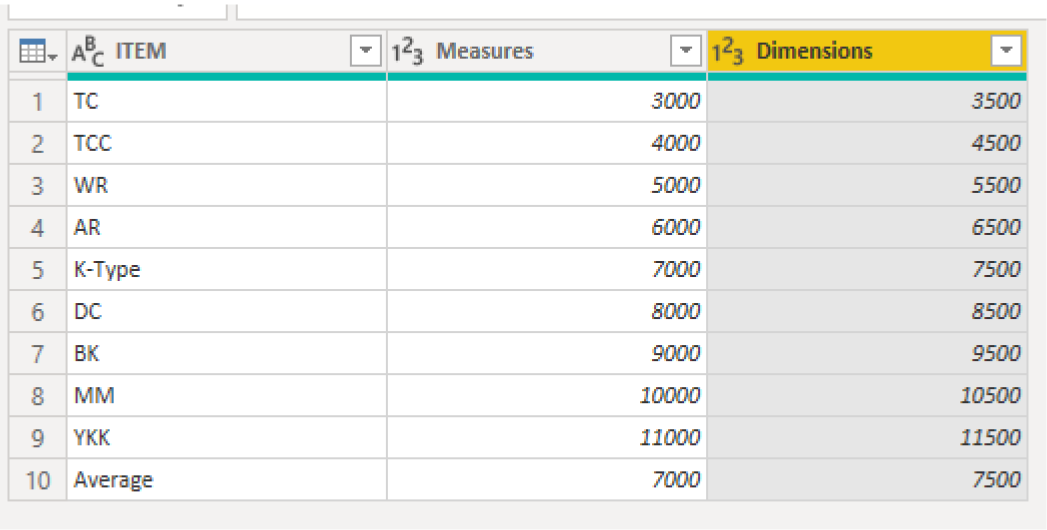
- Find the average
- Get data -> Microsoft Excel -> choose sheet from Excel -> OK -> Load

- The data will be displayed on navigator.

- Now you can Go to Edit Queries and make changes to various fields

- The fields of the table are shown on right side of screen. It has ∑ symbol before some field name such values are called Measures and other fields are called Dimensions.

Output



	ITEM	Measures	Dimensions
1	TC	3000	3500
2	TCC	4000	4500
3	WR	5000	5500
4	AR	6000	6500
5	K-Type	7000	7500
6	DC	8000	8500
7	BK	9000	9500
8	MM	10000	10500
9	YKK	11000	11500
10	Average	7000	7500

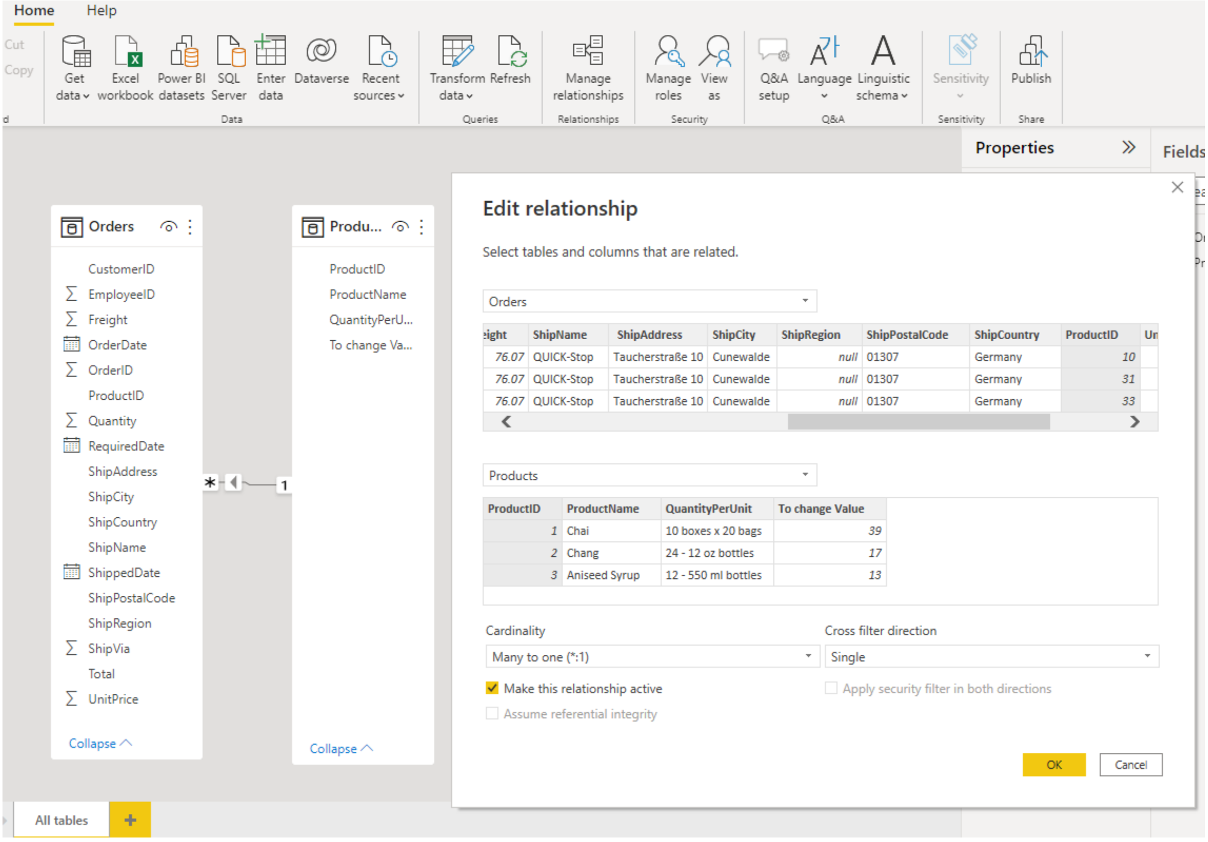
Practical No. 4

Perform the Extraction Transformation and Loading (ETL) process to construct the database in Power BI.

- Get Data -> OData feed -> Put URL- https://services.odata.org/v3/Northwind/Northwind.svc -> OK -> Connect.

- Select tables such as product and order table -> Load.
- Open Edit Queries (New window will be displayed as Power Query Editor)
- Choose product ID, Name, Quantity Per Unit ,Unit In Stock -> OK
- To change Data type
- Select Column -> Click on Data type in Ribbon.
- To change Name of the column
- Double click on column -> Change Name
- To change Value
- Click Replace Value OR Right Click on Value
- To Load another table
- Go to Recent Sources -> Tick Order Table -> OK
- Go to Edit Queries -> Search Column Order_Details OR Click on the icon (which shows all the hidden columns) -> Select Product ID, Unit Price, Quantity -> Save and Apply.
- Add column -> custom column -> Name- Total
- Formula = select Order_Details.Unit Price*Order_Details.Quantity -> OK -> Save File and Apply.
- On Dashboard -> On left side -> click on Relationship Icon.

Output



Edit relationship

Select tables and columns that are related.

Orders

ight	ShipName	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	ShipCountry	ProductID	Ur
76.07	QUICK-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	10	
76.07	QUICK-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	31	
76.07	QUICK-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	33	

Products

ProductID	ProductName	QuantityPerUnit	To change Value
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13

Cardinality: Many to one (*:1)

Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

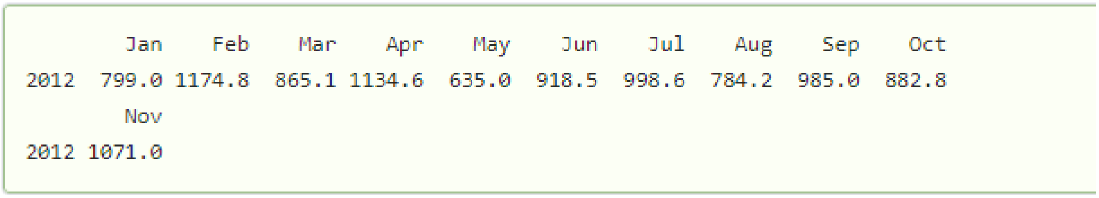
□ Assume referential integrity

Practical No:5

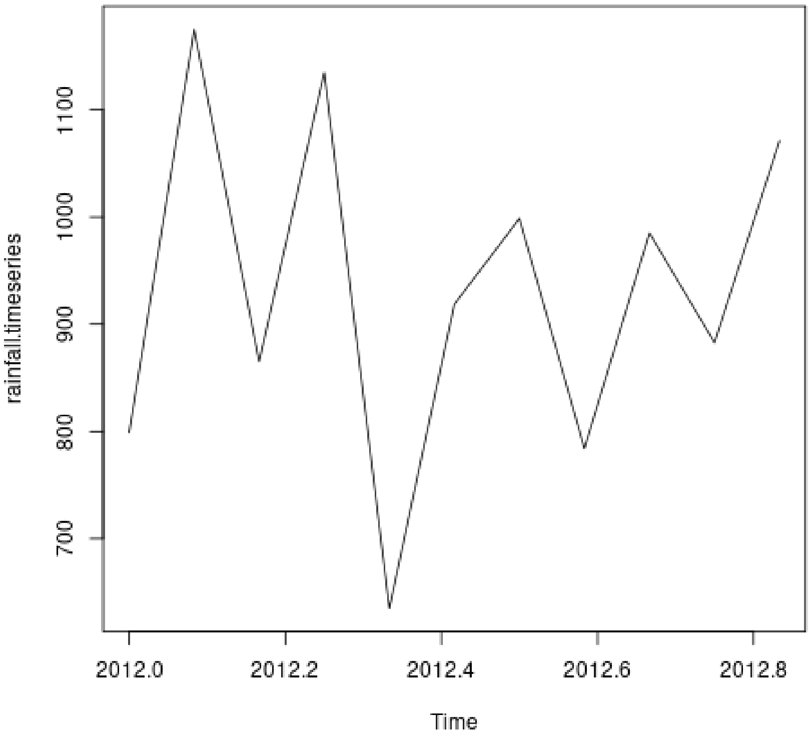
Implementation of Classification algorithm in R Programming.

- Open R tool
- Write on Console
- Code
- rainfall<c(799,1174.8,865.1,1134.6,635,918.5,998.6,784.2,985,882.8,10)
- rainfall.timeseries<-ts(rainfall,start=c(2012,1),frequency=12)
- print(rainfall.timeseries)
- plot(rainfall.timeseries)

<u>Output</u>



```
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
2012  799.0 1174.8  865.1 1134.6  635.0  918.5  998.6  784.2  985.0  882.8
        Nov
2012 1071.0
```

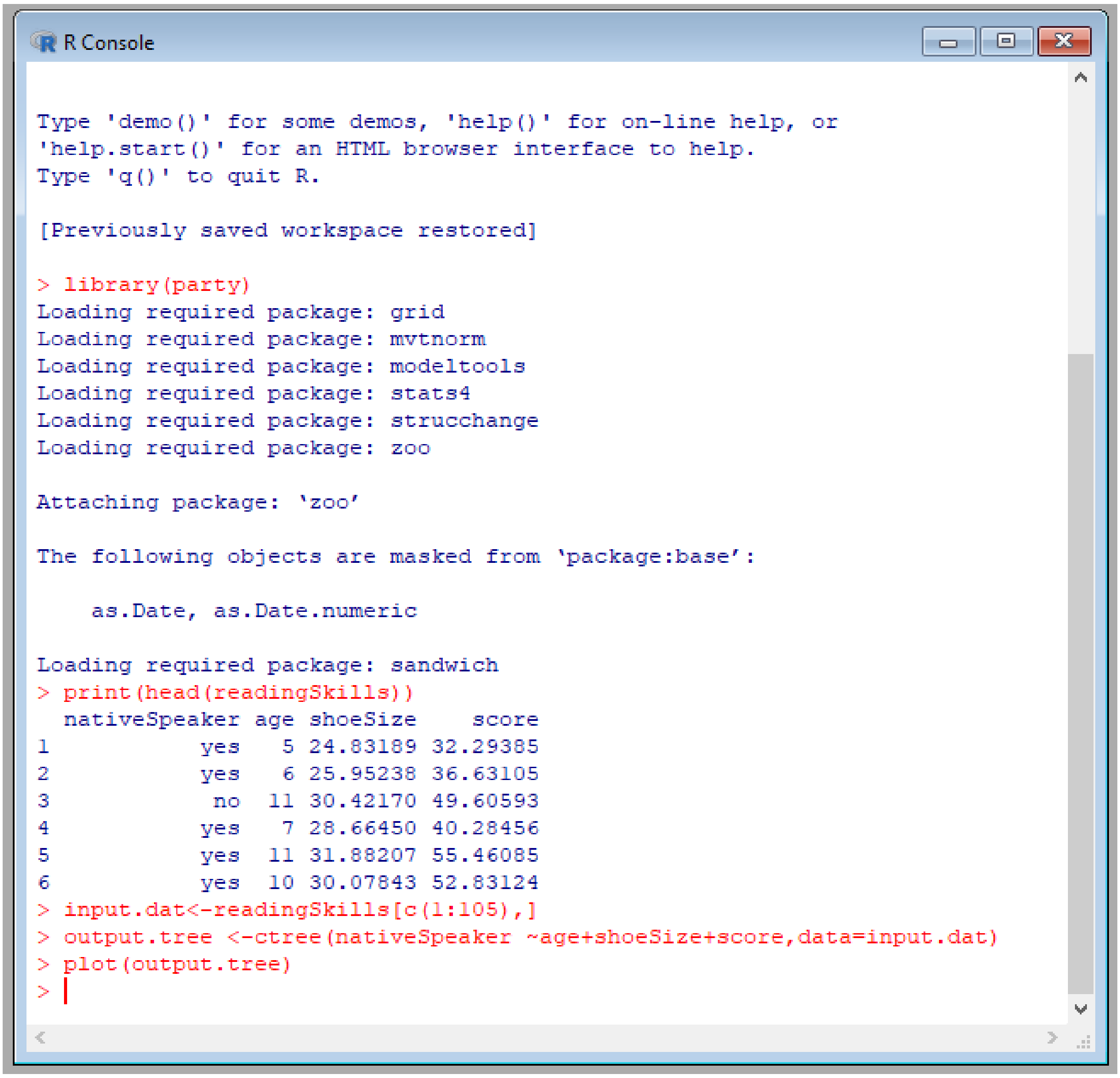


Practical No:6

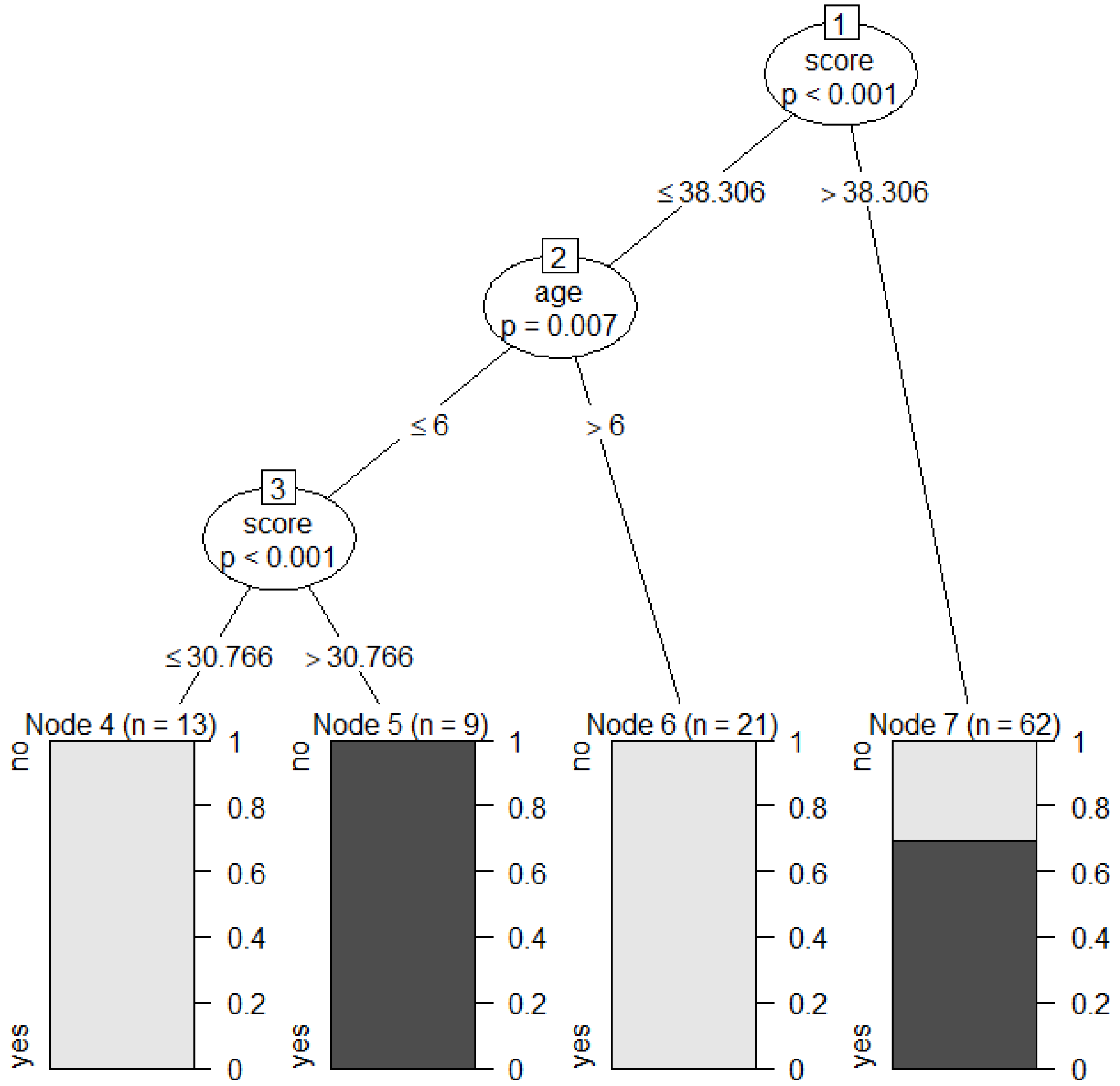
Practical Implementation of Decision Tree using R Tool

- Open R tool
- Write on Console
- Code
- 4. Go to package -> find "party" package and install.
- Load party package
- Library(party)
- print(head(readingSkills))
- input.dat<-readingSkills[c(1:105),]
- output.tree <-ctree(nativeSpeaker ~age+shoeSize+score,data=input.dat)
- plot(output.tree)

Output



```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: sandwich
> print(head(readingSkills))
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
> input.dat<-readingSkills[c(1:105),]
> output.tree <-ctree(nativeSpeaker ~age+shoeSize+score,data=input.dat)
> plot(output.tree)
> 
```

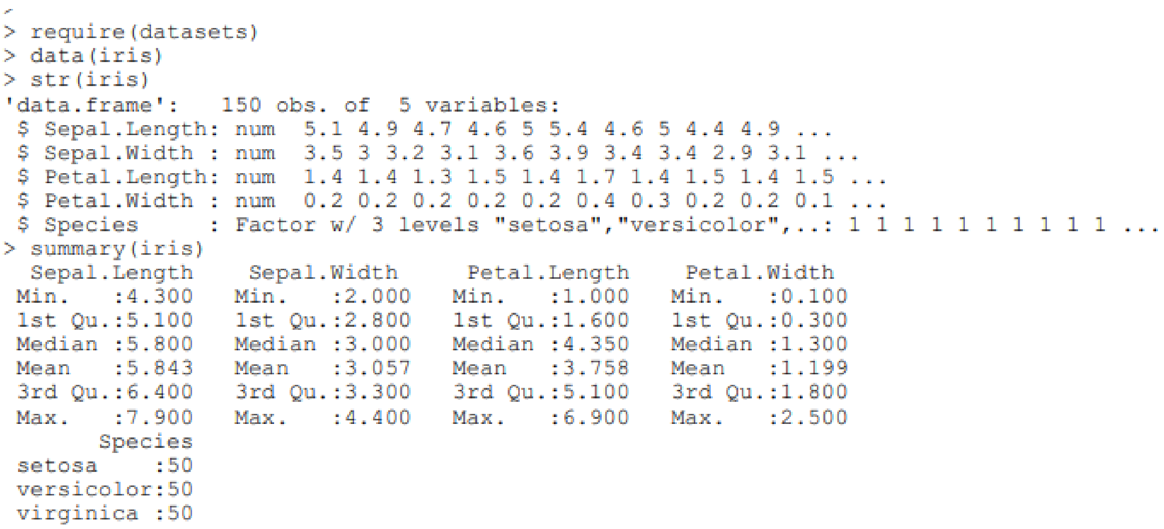

1
score
p < 0.001
≤ 38.306
> 38.306
2
age
p = 0.007
≤ 6
> 6
3
score
p < 0.001
≤ 30.766
> 30.766
Node 4 (n = 13)
Node 5 (n = 9)
Node 6 (n = 21)
Node 7 (n = 62)
no
yes
1
0.8
0.6
0.4
0.2
0

Practical No:7

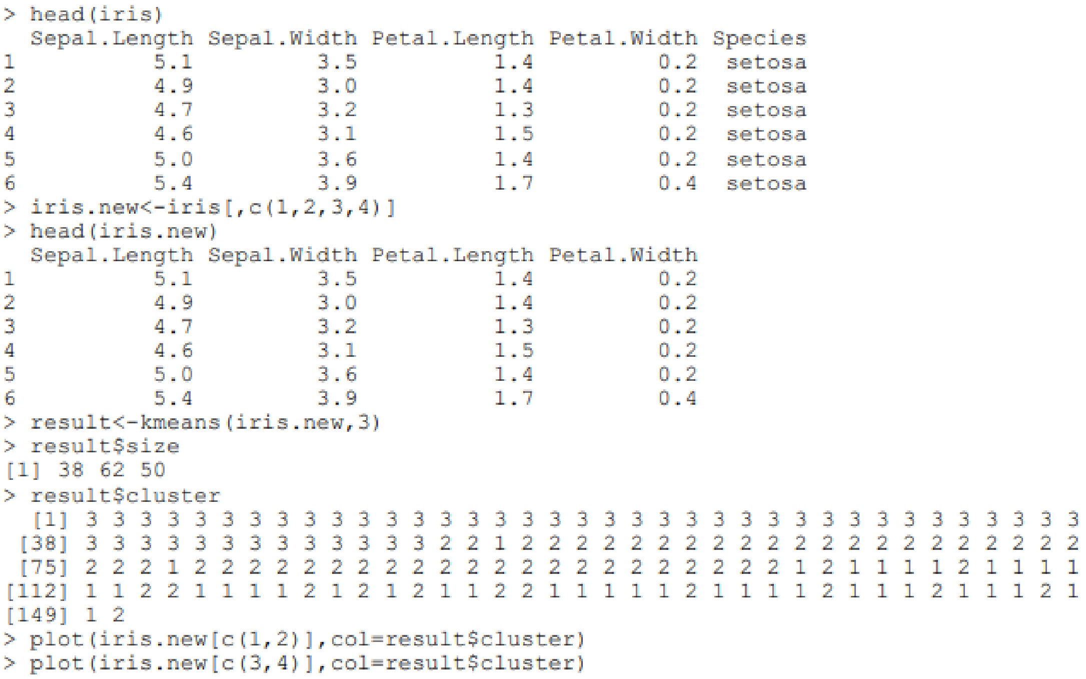
k-means clustering using R

- Open R tool
- Write on Console
- Code
- require(datasets)
- data(iris)
- str(iris)
- summary(iris)
- head(iris)
- iris.new<-iris[,c(1,2,3,4)]
- head(iris.new)
- result<-kmeans(iris.new,3)
- result$size
- result$cluster
- plot(iris.new[c(1,2)],col=result$cluster)
- plot(iris.new[c(3,4)],col=result$cluster)

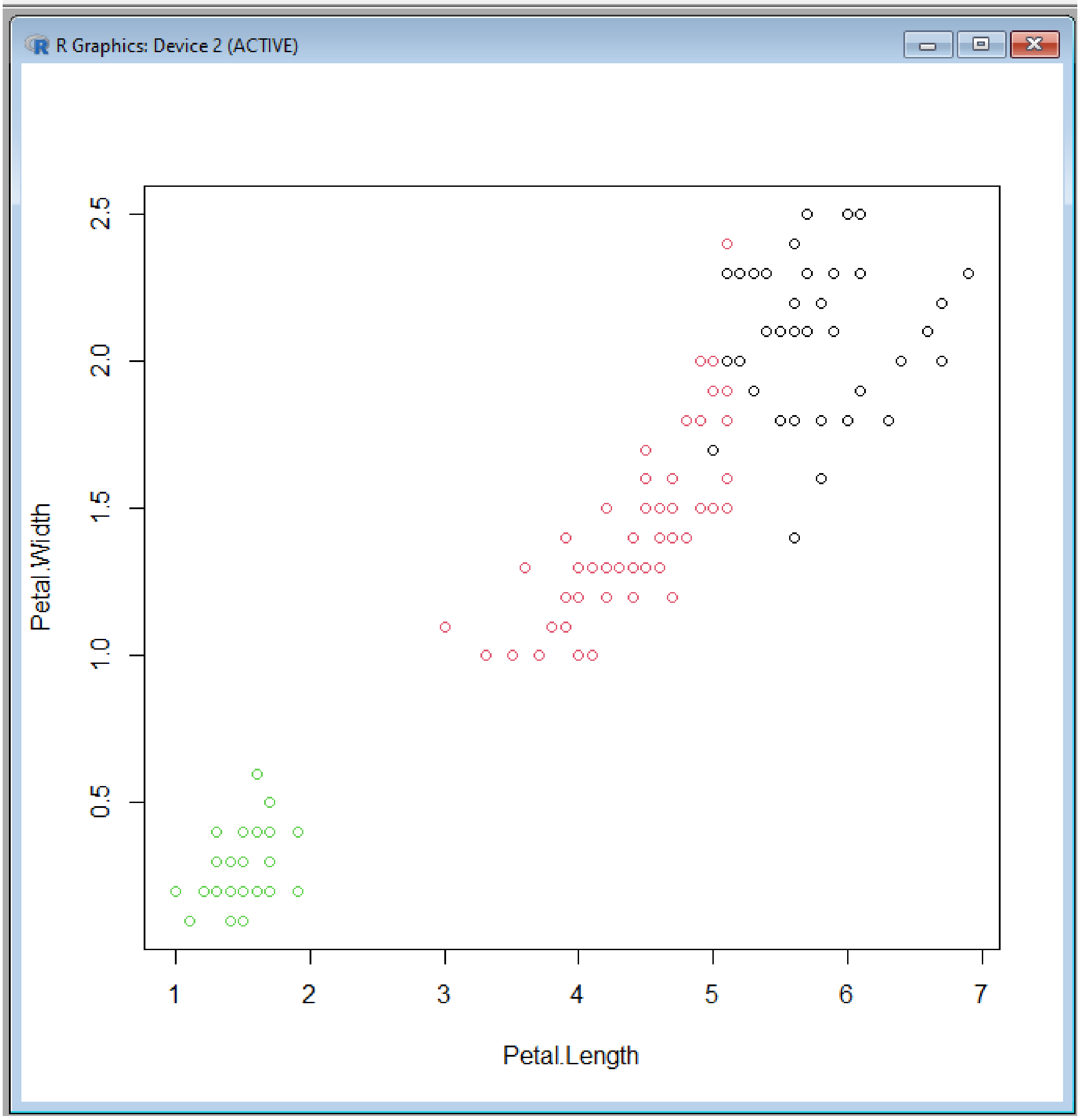
Output



```
> require(datasets)
> data(iris)
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```



```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> iris.new<-iris[,c(1,2,3,4)]
> head(iris.new)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
> result<-kmeans(iris.new,3)
> result$size
[1] 38 62 50
> result$cluster
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
[149] 1 2
> plot(iris.new[c(1,2)],col=result$cluster)
> plot(iris.new[c(3,4)],col=result$cluster)
```

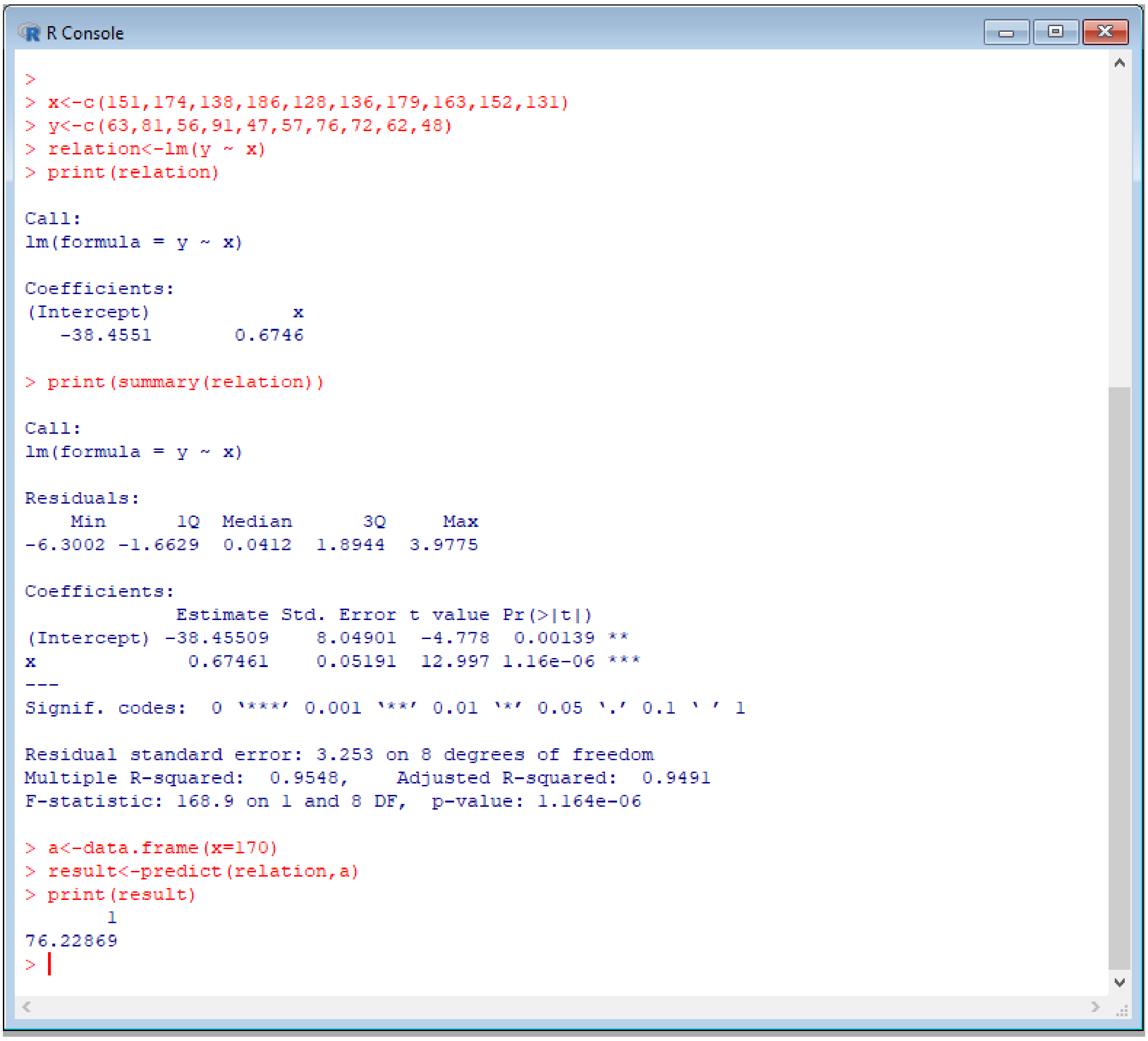

R Graphics: Device 2 (ACTIVE)
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

Practical No:8

Prediction Using Linear Regression

- Open R tool
- Write on Console
- Code
- x<-c(151,174,138,186,128,136,179,163,152,131)
- y<-c(63,81,56,91,47,57,76,72,62,48)
- relation<-lm(y ~ x)
- print(relation)
- print(summary(relation))
- a<-data.frame(x=170)
- result<-predict(relation,a)
- print(result)

Output



```
>
> x<-c(151,174,138,186,128,136,179,163,152,131)
> y<-c(63,81,56,91,47,57,76,72,62,48)
> relation<-lm(y ~ x)
> print(relation)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746

> print(summary(relation))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06

> a<-data.frame(x=170)
> result<-predict(relation,a)
> print(result)
       1
76.22869
>
```