The Kalyan Wholesale Merchants Education Society's

LAXMAN DEVRAM SONAWANE DEGREE COLLEGE

OF

ARTS, COMMERCE AND SCIENCEKALYAN

(Affiliated To University Of Mumbai)

CERTIFICATE

This is certify that Mr./Miss. ____

Class __________________Roll No.___________Exam Seat No. _________________

Has completed the prescribed practical in _______________________________________

during the Academic 202 - 202

Practical	H.O.D.	Internal	External
In charge		Examiner	Examiner

INDEX

Sr.No	Title	Date	Sign

PRACTICAL NO :- 1

AIM :- *Practical of Principal Component Analysis*

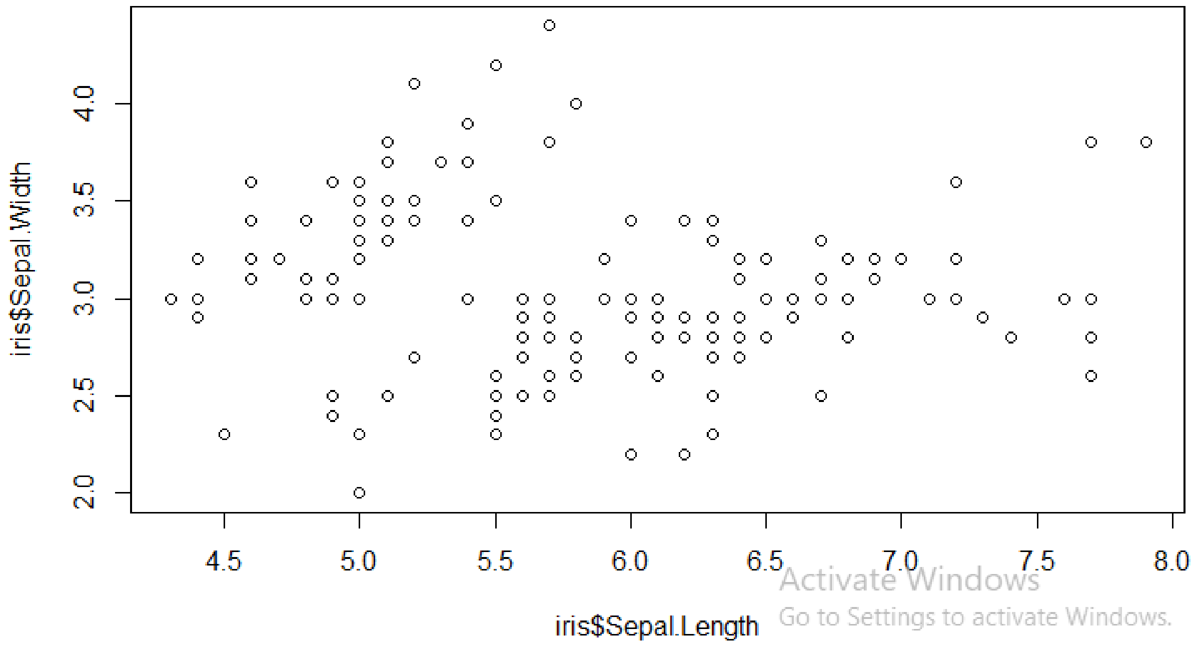
CODE :-

```
> data("iris")
> head(iris)
> summary(iris)
> library()
> "to find principal component"
> mypr <- prcomp(iris[,-5],scale=T)
> "to understand use of scale"
> plot(iris$Sepal.Length,iris$Sepal.Width)
> plot(scale(iris$Sepal.Length),scale(iris$Sepal.Width))
> mypr
> summary(mypr)
> plot(mypr,type="l")
> biplot(mypr,scale=0)
> "extract pc scores"
> str(mypr)
> mypr$x
> iris2 <- cbind(iris,mypr$x[,1:2])
> head(iris2)
> cor(iris[,-5],iris2[,6:7])
> names(iris)
```

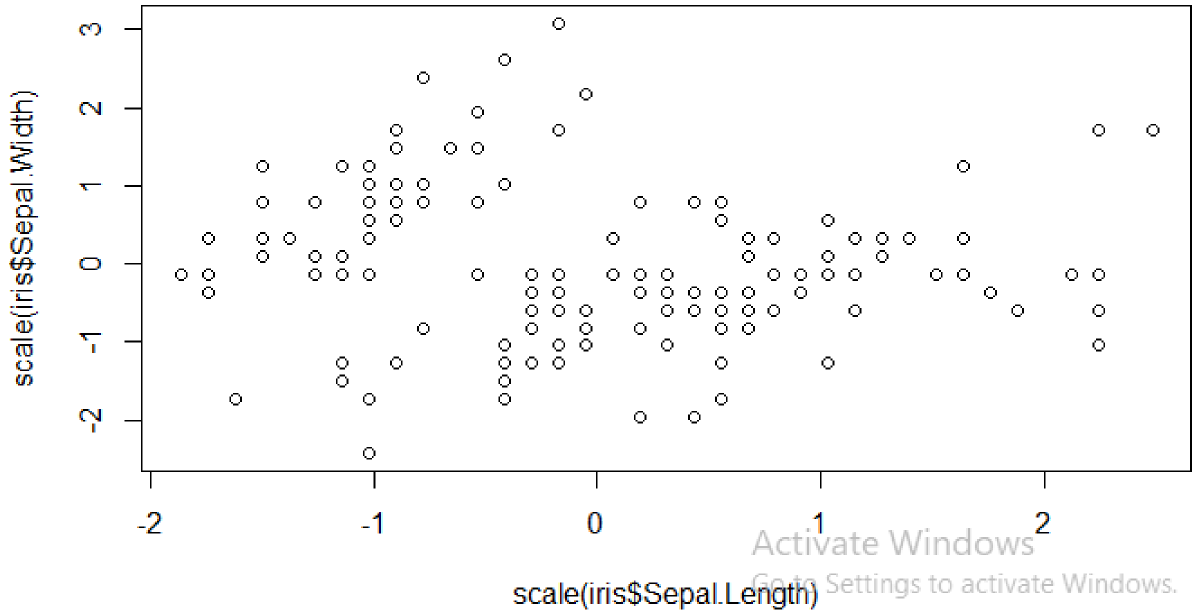
OUTPUT :-

```
> data("iris")
> head(iris)
```

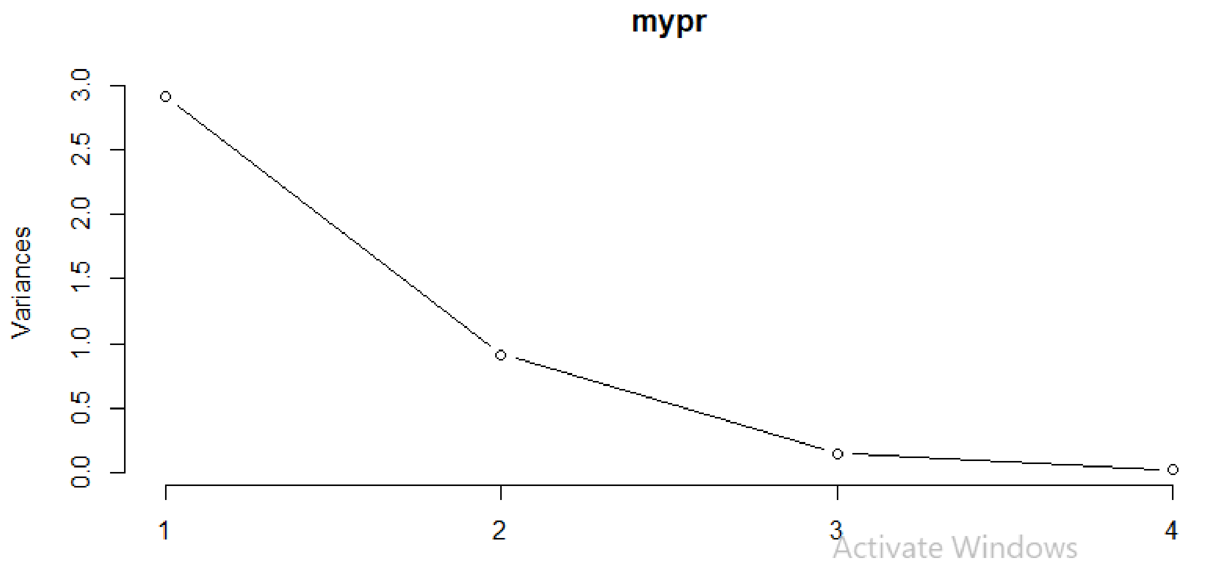
```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> summary(iris)
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica
:50
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
> library()
> mypr <- prcomp(iris[,-5],scale=T)
> plot(iris$Sepal.Length,iris$Sepal.Width)
```



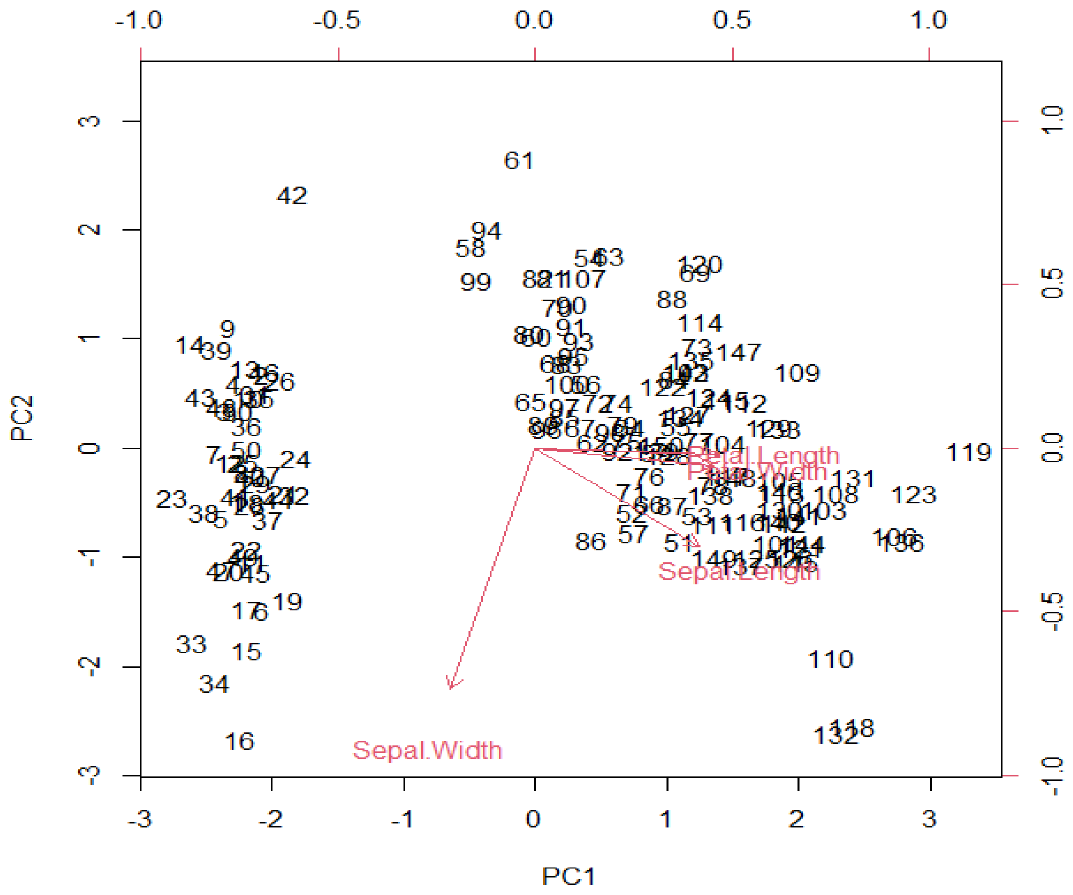
```
> plot(scale(iris$Sepal.Length),scale(iris$Sepal.Width))
```



```
> mypr
Standard deviations (1, .., p=4):
[1] 1.7083611 0.9560494 0.3830886 0.1439265
Rotation (n x k) = (4 x 4):
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
> summary(mypr)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
> plot(mypr,type="l")
```



```
> biplot(mypr,scale=0)
```



```
> str(mypr)
```

```
List of 5
 $ sdev    : num [1:4] 1.708 0.956 0.383 0.144
 $ rotation: num [1:4, 1:4] 0.521 -0.269 0.58 0.565 -0.377 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
  .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
 $ center  : Named num [1:4] 5.84 3.06 3.76 1.2
  ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length"
"Petal.Width"
 $ scale   : Named num [1:4] 0.828 0.436 1.765 0.762
  ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length"
"Petal.Width"
 $ x       : num [1:150, 1:4] -2.26 -2.07 -2.36 -2.29 -2.38 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
 - attr(*, "class")= chr "prcomp"
> mypr$x
              PC1          PC2          PC3          PC4
  [1,] -2.25714118 -0.478423832  0.127279624  0.024087508
  [2,] -2.07401302  0.671882687  0.233825517  0.102662845
  [3,] -2.35633511  0.340766425 -0.044053900  0.028282305
  [4,] -2.29170679  0.595399863 -0.090985297 -0.065735340
  [5,] -2.38186270 -0.644675659 -0.015685647 -0.035802870
  [6,] -2.06870061 -1.484205297 -0.026878250  0.006586116
  [7,] -2.43586845 -0.047485118 -0.334350297 -0.036652767
  [8,] -2.22539189 -0.222403002  0.088399352 -0.024529919
  [9,] -2.32684533  1.111603700 -0.144592465 -0.026769540
 [10,] -2.17703491  0.467447569  0.252918268 -0.039766068
 [11,] -2.15907699 -1.040205867  0.267784001  0.016675503
 [12,] -2.31836413 -0.132633999 -0.093446191 -0.133037725
 [13,] -2.21104370  0.726243183  0.230140246  0.002416941
 [14,] -2.62430902  0.958296347 -0.180192423 -0.019151375
 [15,] -2.19139921 -1.853846555  0.471322025  0.194081578
 [16,] -2.25466121 -2.677315230 -0.030424684  0.050365010
```

```
[17,] -2.20021676 -1.478655729  0.005326251  0.188186988
[18,] -2.18303613 -0.487206131  0.044067686  0.092779618
[19,] -1.89223284 -1.400327567  0.373093377  0.060891973
[20,] -2.33554476 -1.124083597 -0.132187626 -0.037630354
[21,] -1.90793125 -0.407490576  0.419885937  0.010884821
[22,] -2.19964383 -0.921035871 -0.159331502  0.059398340
[23,] -2.76508142 -0.456813301 -0.331069982  0.019582826
[24,] -1.81259716 -0.085272854 -0.034373442  0.150636353
[25,] -2.21972701 -0.136796175 -0.117599566 -0.269238379
[26,] -1.94532930  0.623529705  0.304620475  0.043416203
[27,] -2.04430277 -0.241354991 -0.086075649  0.067454082
[28,] -2.16133650 -0.525389422  0.206125707  0.010241084
[29,] -2.13241965 -0.312172005  0.270244895  0.083977887
[30,] -2.25769799  0.336604248 -0.068207276 -0.107918349
[31,] -2.13297647  0.502856075  0.074757996 -0.048027970
[32,] -1.82547925 -0.422280389  0.269564311  0.239069476
[33,] -2.60621687 -1.787587272 -0.047070727 -0.228470534
[34,] -2.43800983 -2.143546796  0.082392024 -0.048053409
[35,] -2.10292986  0.458665270  0.169706329  0.028926042
[36,] -2.20043723  0.205419224  0.224688852  0.168343905
[37,] -2.03831765 -0.659349230  0.482919584  0.195702902
[38,] -2.51889339 -0.590315163 -0.019370918 -0.136048774
[39,] -2.42152026  0.901161067 -0.192609402 -0.009705907
[40,] -2.16246625 -0.267981199  0.175296561  0.007023875
[41,] -2.27884081 -0.440240541 -0.034778398  0.106626042
[42,] -1.85191836  2.329610745  0.203552303  0.288896090
[43,] -2.54511203  0.477501017 -0.304745527 -0.066379077
[44,] -1.95788857 -0.470749613 -0.308567588  0.176501717
[45,] -2.12992356 -1.138415464 -0.247604064 -0.150539117
[46,] -2.06283361  0.708678586  0.063716370  0.139801160
[47,] -2.37677076 -1.116688691 -0.057026813 -0.151722682
[48,] -2.38638171  0.384957230 -0.139002234 -0.048671707
[49,] -2.22200263 -0.994627669  0.180886792 -0.014878291
[50,] -2.19647504 -0.009185585  0.152518539  0.049206884
[51,]  1.09810244 -0.860091033  0.682300393  0.034717469
```

```
[52,]  0.72889556 -0.592629362  0.093807452  0.004887251
[53,]  1.23683580 -0.614239894  0.552157058  0.009391933
[54,]  0.40612251  1.748546197  0.023024633  0.065549239
[55,]  1.07188379  0.207725147  0.396925784  0.104387166
[56,]  0.38738955  0.591302717 -0.123776885 -0.240027187
[57,]  0.74403715 -0.770438272 -0.148472007 -0.077111455
[58,] -0.48569562  1.846243998 -0.248432992 -0.040384912
[59,]  0.92480346 -0.032118478  0.594178807 -0.029779844
[60,]  0.01138804  1.030565784 -0.537100055 -0.028366154
[61,] -0.10982834  2.645211115  0.046634215  0.013714785
[62,]  0.43922201  0.063083852 -0.204389093  0.039992104
[63,]  0.56023148  1.758832129  0.763214554  0.045578465
[64,]  0.71715934  0.185602819  0.068429700 -0.164256922
[65,] -0.03324333  0.437537419 -0.194282030  0.108684396
[66,]  0.87248429 -0.507364239  0.501830204  0.104593326
[67,]  0.34908221  0.195656268 -0.489234095 -0.190869932
[68,]  0.15827980  0.789451008  0.301028700 -0.204612265
[69,]  1.22100316  1.616827281  0.480693656  0.225145511
[70,]  0.16436725  1.298259939  0.172260719 -0.051554138
[71,]  0.73521959 -0.395247446 -0.614467782 -0.083006045
[72,]  0.47469691  0.415926887  0.264067576  0.113189079
[73,]  1.23005729  0.930209441  0.367182178 -0.009911322
[74,]  0.63074514  0.414997441  0.290921638 -0.273304557
[75,]  0.70031506  0.063200094  0.444537765  0.043313222
[76,]  0.87135454 -0.249956017  0.471001057  0.101376117
[77,]  1.25231375  0.076998069  0.724727099  0.039556002
[78,]  1.35386953 -0.330205463  0.259955701  0.066604931
[79,]  0.66258066  0.225173502 -0.085577197 -0.036318171
[80,] -0.04012419  1.055183583  0.318506304  0.064571834
[81,]  0.13035846  1.557055553  0.149482697 -0.009371129
[82,]  0.02337438  1.567225244  0.240745761 -0.032663020
[83,]  0.24073180  0.774661195  0.150707074  0.023572390
[84,]  1.05755171  0.631726901 -0.104959762 -0.183354200
[85,]  0.22323093  0.286812663 -0.663028512 -0.253977520
[86,]  0.42770626 -0.842758920 -0.449129446 -0.109308985
```

```
 [87,]  1.04522645 -0.520308714  0.394464890  0.037084781
 [88,]  1.04104379  1.378371048  0.685997804  0.136378719
 [89,]  0.06935597  0.218770433 -0.290605718 -0.146653279
 [90,]  0.28253073  1.324886147 -0.089111491  0.008876070
 [91,]  0.27814596  1.116288852 -0.094172116 -0.269753497
 [92,]  0.62248441 -0.024839814  0.020412763 -0.147193289
 [93,]  0.33540673  0.985103828  0.198724011  0.006508757
 [94,] -0.36097409  2.012495825 -0.105467721  0.019505467
 [95,]  0.28762268  0.852873116 -0.130452657 -0.107043742
 [96,]  0.09105561  0.180587142 -0.128547696 -0.229191812
 [97,]  0.22695654  0.383634868 -0.155691572 -0.132163118
 [98,]  0.57446378  0.154356489  0.270743347 -0.019794366
 [99,] -0.44617230  1.538637456 -0.189765199  0.199278855
[100,]  0.25587339  0.596852285 -0.091572385 -0.058426315
[101,]  1.83841002 -0.867515056 -1.002044077 -0.049085303
[102,]  1.15401555  0.696536401 -0.528389994 -0.040385459
[103,]  2.19790361 -0.560133976  0.202236658  0.058986583
[104,]  1.43534213  0.046830701 -0.163083761 -0.234982858
[105,]  1.86157577 -0.294059697 -0.394307408 -0.016243853
[106,]  2.74268509 -0.797736709  0.580364827 -0.101045973
[107,]  0.36579225  1.556289178 -0.983598122 -0.132679346
[108,]  2.29475181 -0.418663020  0.649530452 -0.237246445
[109,]  1.99998633  0.709063226  0.392675073 -0.086221779
[110,]  2.25223216 -1.914596301 -0.396224508  0.104488870
[111,]  1.35962064 -0.690443405 -0.283661780  0.107500284
[112,]  1.59732747  0.420292431 -0.023108991  0.058136869
[113,]  1.87761053 -0.417849815 -0.026250468  0.145926073
[114,]  1.25590769  1.158379741 -0.578311891  0.098826244
[115,]  1.46274487  0.440794883 -1.000517746  0.274738504
[116,]  1.58476820 -0.673986887 -0.636297054  0.191222383
[117,]  1.46651849 -0.254768327 -0.037306280 -0.154811637
[118,]  2.41822770 -2.548124795  0.127454475 -0.272892966
[119,]  3.29964148 -0.017721580  0.700957033  0.045037725
[120,]  1.25954707  1.701046715  0.266643612 -0.064963167
[121,]  2.03091256 -0.907427443 -0.234015510  0.167390481
```

```
[122,]  0.97471535  0.569855257 -0.825362161  0.027662914
[123,]  2.88797650 -0.412259950  0.854558973 -0.126911337
[124,]  1.32878064  0.480202496  0.005410239  0.139491837
[125,]  1.69505530 -1.010536476 -0.297454114 -0.061437911
[126,]  1.94780139 -1.004412720  0.418582432 -0.217609339
[127,]  1.17118007  0.315338060 -0.129503907  0.125001677
[128,]  1.01754169 -0.064131184 -0.336588365 -0.008625505
[129,]  1.78237879  0.186735633 -0.269754304  0.030983849
[130,]  1.85742501 -0.560413289  0.713244682 -0.207519953
[131,]  2.42782030 -0.258418706  0.725386035 -0.017863520
[132,]  2.29723178 -2.617554417  0.491826144 -0.210968943
[133,]  1.85648383  0.177953334 -0.352966242  0.099675959
[134,]  1.11042770  0.291944582  0.182875741 -0.185721512
[135,]  1.19845835  0.808606364  0.164173760 -0.487849130
[136,]  2.78942561 -0.853942542  0.541093785  0.294893130
[137,]  1.57099294 -1.065013214 -0.942695700  0.035486875
[138,]  1.34179696 -0.421020154 -0.180271551 -0.214702016
[139,]  0.92173701 -0.017165594 -0.415434449  0.005220919
[140,]  1.84586124 -0.673870645  0.012629804  0.194543500
[141,]  2.00808316 -0.611835930 -0.426902678  0.246711805
[142,]  1.89543421 -0.687273065 -0.129640697  0.468128374
[143,]  1.15401555  0.696536401 -0.528389994 -0.040385459
[144,]  2.03374499 -0.864624030 -0.337014969  0.045036251
[145,]  1.99147547 -1.045665670 -0.630301866  0.213330527
[146,]  1.86425786 -0.385674038 -0.255418178  0.387957152
[147,]  1.55935649  0.893692855  0.026283300  0.219456899
[148,]  1.51609145 -0.268170747 -0.179576781  0.118773236
[149,]  1.36820418 -1.007877934 -0.930278721  0.026041407
[150,]  0.95744849  0.024250427 -0.526485033 -0.162533529

> iris2 <- cbind(iris,mypr$x[,1:2])
> head(iris2)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species       PC1
PC2
1          5.1         3.5          1.4         0.2  setosa -2.257141 -0.4784238
```

```
2          4.9         3.0         1.4         0.2  setosa -2.074013  0.6718827
3          4.7         3.2         1.3         0.2  setosa -2.356335  0.3407664
4          4.6         3.1         1.5         0.2  setosa -2.291707  0.5953999
5          5.0         3.6         1.4         0.2  setosa -2.381863 -0.6446757
6          5.4         3.9         1.7         0.4  setosa -2.068701 -1.4842053
```

```
> cor(iris[,-5],iris2[,6:7])
                    PC1         PC2
Sepal.Length  0.8901688 -0.36082989
Sepal.Width  -0.4601427 -0.88271627
Petal.Length  0.9915552 -0.02341519
Petal.Width   0.9649790 -0.06399985
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

PRACTICAL NO :- 2

AIM :- *Practical of K means clustering*

CODE :-

```
> "k-means clustering"
> data(iris)
> names(iris)
> new_data <- subset(iris,select=c(-Species))
> new_data
> cl <- kmeans(new_data,3)
> cl
> data <- new_data
> wss <-sapply(1:15,function(k){kmeans(data,k)$tot.withinss})
> wss
> plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab="Number of clusters
K",ylab = "Total Within-clusters sums of squares")
> library(cluster)
> clusplot(new_data,cl$cluster,color = TRUE,shade=TRUE,labels =
2,lines=0)
> cl$cluster
> cl$centers
> clusters <- hclust(dist(iris[,3:4]))
> plot(clusters)
> clusterCut <- cutree(clusters,3)
> table(clusterCut,iris$Species)
```

OUTPUT :-

```
> data(iris)
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```
> new_data <- subset(iris,select=c(-Species))
> new_data
    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
7          4.6         3.4          1.4         0.3
8          5.0         3.4          1.5         0.2
9          4.4         2.9          1.4         0.2
10         4.9         3.1          1.5         0.1
11         5.4         3.7          1.5         0.2
12         4.8         3.4          1.6         0.2
13         4.8         3.0          1.4         0.1
14         4.3         3.0          1.1         0.1
15         5.8         4.0          1.2         0.2
16         5.7         4.4          1.5         0.4
17         5.4         3.9          1.3         0.4
18         5.1         3.5          1.4         0.3
19         5.7         3.8          1.7         0.3
20         5.1         3.8          1.5         0.3
21         5.4         3.4          1.7         0.2
22         5.1         3.7          1.5         0.4
23         4.6         3.6          1.0         0.2
24         5.1         3.3          1.7         0.5
25         4.8         3.4          1.9         0.2
26         5.0         3.0          1.6         0.2
27         5.0         3.4          1.6         0.4
28         5.2         3.5          1.5         0.2
29         5.2         3.4          1.4         0.2
30         4.7         3.2          1.6         0.2
31         4.8         3.1          1.6         0.2
32         5.4         3.4          1.5         0.4
```

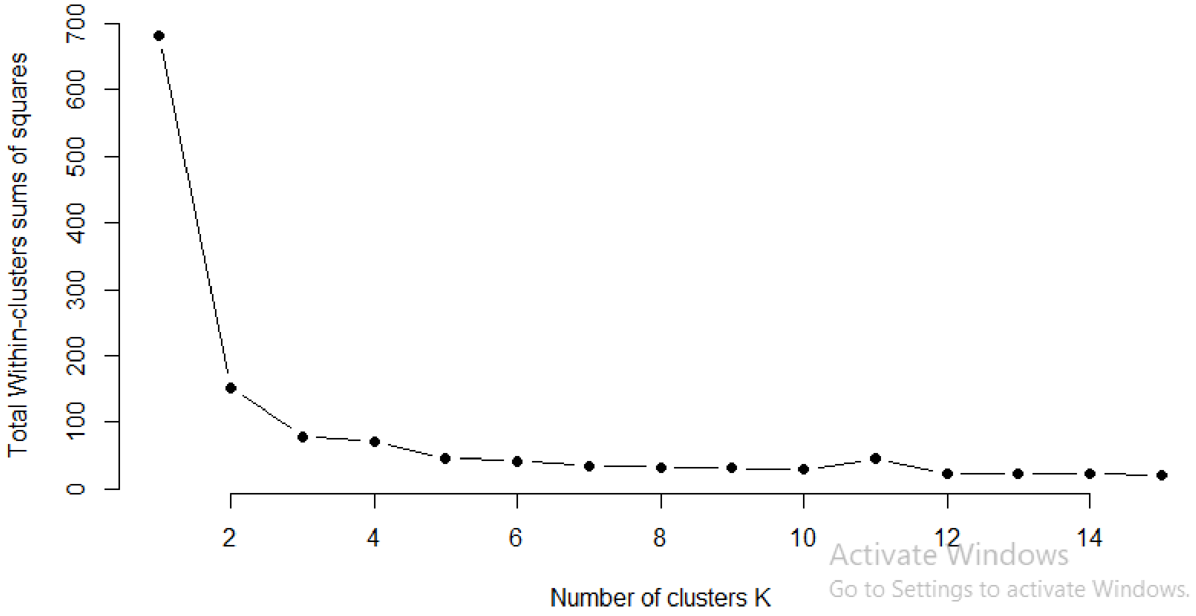
33	5.2	4.1	1.5	0.1
34	5.5	4.2	1.4	0.2
35	4.9	3.1	1.5	0.2
36	5.0	3.2	1.2	0.2
37	5.5	3.5	1.3	0.2
38	4.9	3.6	1.4	0.1
39	4.4	3.0	1.3	0.2
40	5.1	3.4	1.5	0.2
41	5.0	3.5	1.3	0.3
42	4.5	2.3	1.3	0.3
43	4.4	3.2	1.3	0.2
44	5.0	3.5	1.6	0.6
45	5.1	3.8	1.9	0.4
46	4.8	3.0	1.4	0.3
47	5.1	3.8	1.6	0.2
48	4.6	3.2	1.4	0.2
49	5.3	3.7	1.5	0.2
50	5.0	3.3	1.4	0.2
51	7.0	3.2	4.7	1.4
52	6.4	3.2	4.5	1.5
53	6.9	3.1	4.9	1.5
54	5.5	2.3	4.0	1.3
55	6.5	2.8	4.6	1.5
56	5.7	2.8	4.5	1.3
57	6.3	3.3	4.7	1.6
58	4.9	2.4	3.3	1.0
59	6.6	2.9	4.6	1.3
60	5.2	2.7	3.9	1.4
61	5.0	2.0	3.5	1.0
62	5.9	3.0	4.2	1.5
63	6.0	2.2	4.0	1.0
64	6.1	2.9	4.7	1.4
65	5.6	2.9	3.6	1.3
66	6.7	3.1	4.4	1.4
67	5.6	3.0	4.5	1.5

68	5.8	2.7	4.1	1.0
69	6.2	2.2	4.5	1.5
70	5.6	2.5	3.9	1.1
71	5.9	3.2	4.8	1.8
72	6.1	2.8	4.0	1.3
73	6.3	2.5	4.9	1.5
74	6.1	2.8	4.7	1.2
75	6.4	2.9	4.3	1.3
76	6.6	3.0	4.4	1.4
77	6.8	2.8	4.8	1.4
78	6.7	3.0	5.0	1.7
79	6.0	2.9	4.5	1.5
80	5.7	2.6	3.5	1.0
81	5.5	2.4	3.8	1.1
82	5.5	2.4	3.7	1.0
83	5.8	2.7	3.9	1.2
84	6.0	2.7	5.1	1.6
85	5.4	3.0	4.5	1.5
86	6.0	3.4	4.5	1.6
87	6.7	3.1	4.7	1.5
88	6.3	2.3	4.4	1.3
89	5.6	3.0	4.1	1.3
90	5.5	2.5	4.0	1.3
91	5.5	2.6	4.4	1.2
92	6.1	3.0	4.6	1.4
93	5.8	2.6	4.0	1.2
94	5.0	2.3	3.3	1.0
95	5.6	2.7	4.2	1.3
96	5.7	3.0	4.2	1.2
97	5.7	2.9	4.2	1.3
98	6.2	2.9	4.3	1.3
99	5.1	2.5	3.0	1.1
100	5.7	2.8	4.1	1.3
101	6.3	3.3	6.0	2.5
102	5.8	2.7	5.1	1.9

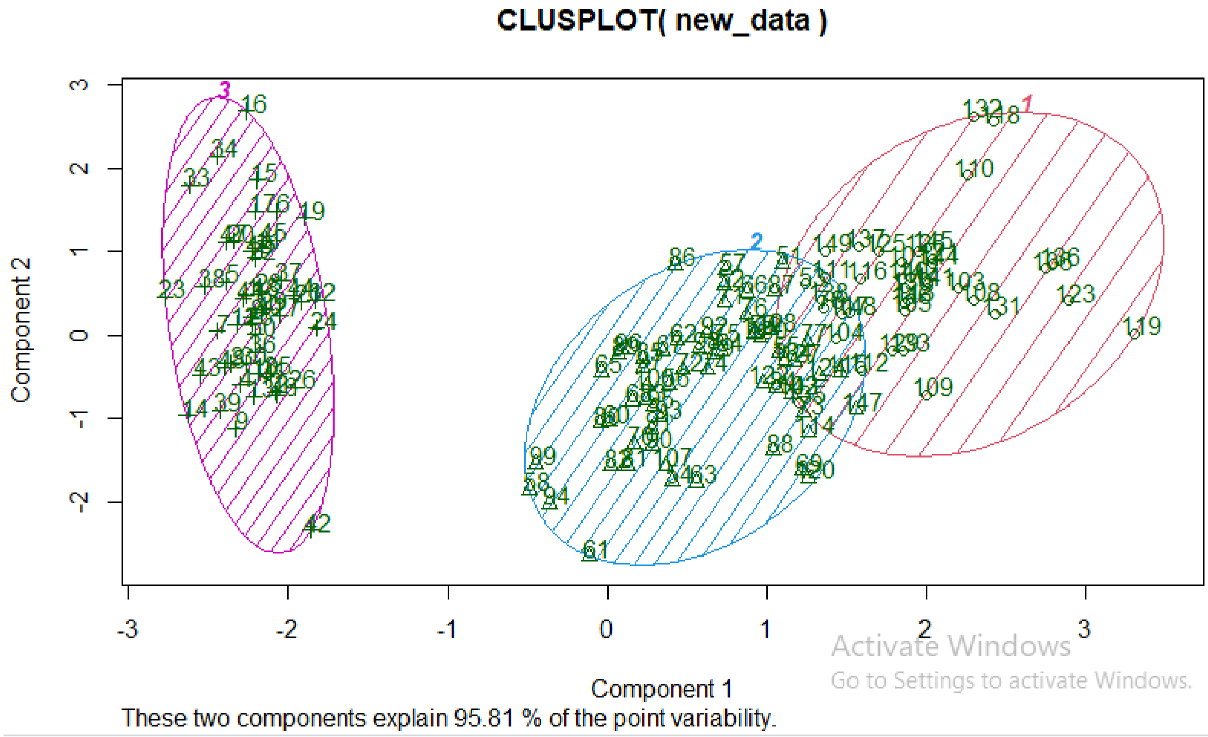
103	7.1	3.0	5.9	2.1
104	6.3	2.9	5.6	1.8
105	6.5	3.0	5.8	2.2
106	7.6	3.0	6.6	2.1
107	4.9	2.5	4.5	1.7
108	7.3	2.9	6.3	1.8
109	6.7	2.5	5.8	1.8
110	7.2	3.6	6.1	2.5
111	6.5	3.2	5.1	2.0
112	6.4	2.7	5.3	1.9
113	6.8	3.0	5.5	2.1
114	5.7	2.5	5.0	2.0
115	5.8	2.8	5.1	2.4
116	6.4	3.2	5.3	2.3
117	6.5	3.0	5.5	1.8
118	7.7	3.8	6.7	2.2
119	7.7	2.6	6.9	2.3
120	6.0	2.2	5.0	1.5
121	6.9	3.2	5.7	2.3
122	5.6	2.8	4.9	2.0
123	7.7	2.8	6.7	2.0
124	6.3	2.7	4.9	1.8
125	6.7	3.3	5.7	2.1
126	7.2	3.2	6.0	1.8
127	6.2	2.8	4.8	1.8
128	6.1	3.0	4.9	1.8
129	6.4	2.8	5.6	2.1
130	7.2	3.0	5.8	1.6
131	7.4	2.8	6.1	1.9
132	7.9	3.8	6.4	2.0
133	6.4	2.8	5.6	2.2
134	6.3	2.8	5.1	1.5
135	6.1	2.6	5.6	1.4
136	7.7	3.0	6.1	2.3
137	6.3	3.4	5.6	2.4

```
138         6.4       3.1        5.5       1.8
139         6.0       3.0        4.8       1.8
140         6.9       3.1        5.4       2.1
141         6.7       3.1        5.6       2.4
142         6.9       3.1        5.1       2.3
143         5.8       2.7        5.1       1.9
144         6.8       3.2        5.9       2.3
145         6.7       3.3        5.7       2.5
146         6.7       3.0        5.2       2.3
147         6.3       2.5        5.0       1.9
148         6.5       3.0        5.2       2.0
149         6.2       3.4        5.4       2.3
150         5.9       3.0        5.1       1.8

> cl <- kmeans(new_data,3)
> cl
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
42
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
63
  3   3   3   3   3   3   3   3   2   2   1   2   2   2   2   2   2   2   2   2   2
 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83
84
```

```
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2 
 85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 
104 105 
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   1   1   1 
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 
123 124 125 126 
  1   2   1   1   1   1   1   1   2   2   1   1   1   1   2   1   2   1   2   1   1 
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 
144 145 146 147 
  2   2   1   1   1   1   1   2   1   1   1   1   2   1   1   1   2   1   1   1   2 
148 149 150 
  1   1   2 

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"      
> data <- new_data
> wss <-sapply(1:15,function(k){kmeans(data,k)$tot.withinss})
> wss
 [1] 681.37060 152.34795  78.85144  71.44525  46.44618  42.42155 
34.61250  33.08052
 [9]  31.11917  29.05600  44.98542  22.74465  22.78841  22.22482 
20.42540
> plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab="Number of clusters 
K",ylab = "Total Within-clusters sums of squares")
```

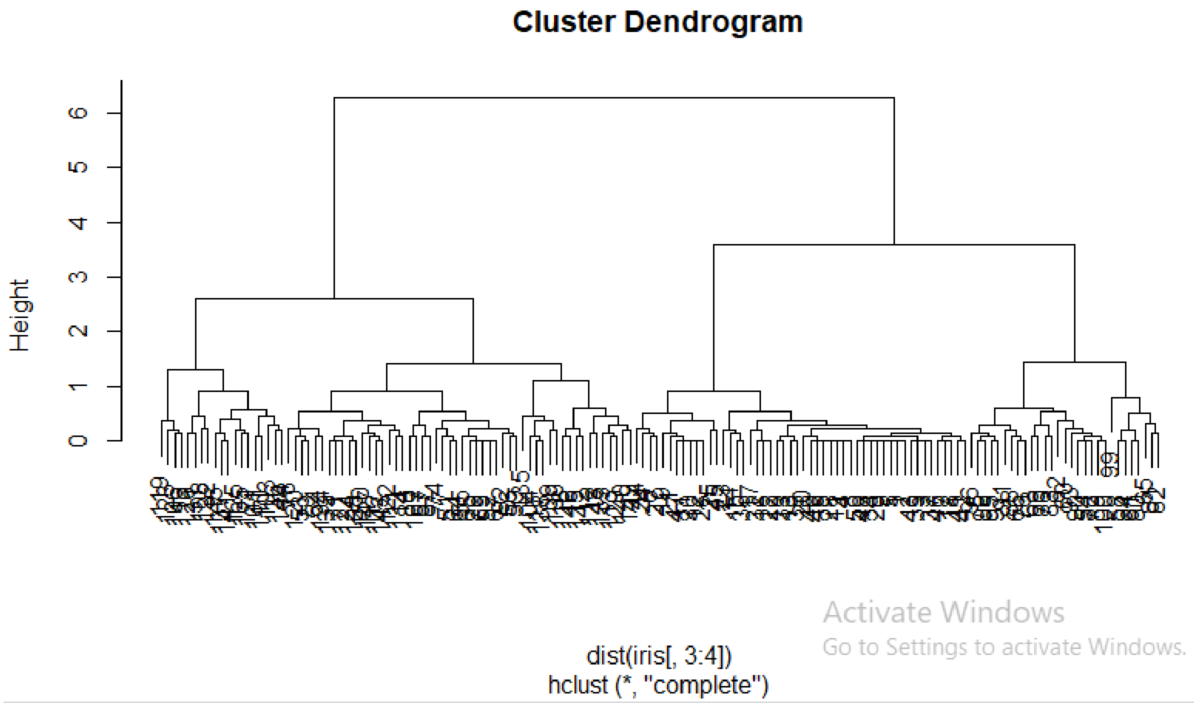


```
> library(cluster)
> clusplot(new_data,cl$cluster,color = TRUE,shade=TRUE,labels =
2,lines=0)
```



```
> cl$cluster
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
22  23  24
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
45  46  47  48
  3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
 49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
69  70  71  72
  3   3   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
 73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92
93  94  95  96
  2   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113
114 115 116 117 118 119 120
  2   2   2   2   1   2   1   1   1   1   2   1   1   1   1   1   1   2   2   1   1   1   1   2
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137
138 139 140 141 142 143 144
  1   2   1   2   1   1   2   2   1   1   1   1   1   2   1   1   1   1   2   1   1   1   2   1
145 146 147 148 149 150
  1   1   2   1   1   2

> cl$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000
> clusters <- hclust(dist(iris[,3:4]))
> plot(clusters)
```



```
> clusterCut <- cutree(clusters,3)
> table(clusterCut,iris$Species)

clusterCut setosa versicolor virginica
         1     50          0         0
         2      0         21        50
         3      0         29         0
```

PRACTICAL NO :- 3

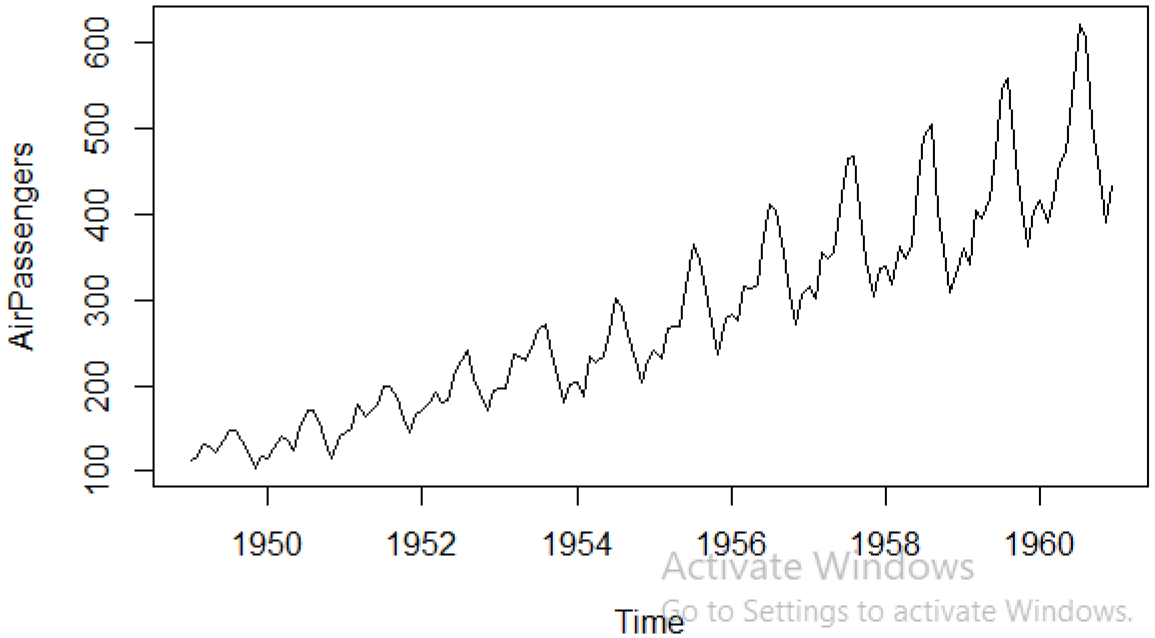
AIM :- *Practical of Time Series Forecasting*

CODE :-

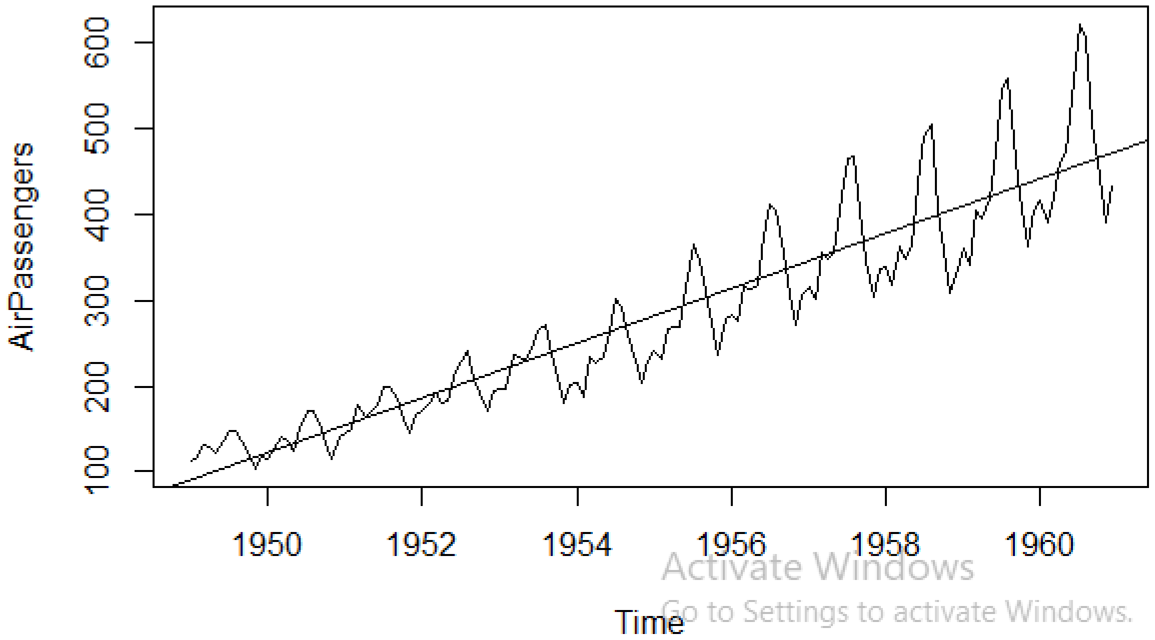
```
> data("AirPassengers")
> class(AirPassengers)
> start(AirPassengers)
> end(AirPassengers)
> frequency(AirPassengers)
> summary(AirPassengers)
> plot(AirPassengers)
> abline(reg=lm(AirPassengers ~ time(AirPassengers)))
> cycle(AirPassengers)
> plot(aggregate(AirPassengers,FUN=mean))
> boxplot(AirPassengers ~ cycle(AirPassengers))
```

OUTPUT :-

```
> data("AirPassengers")
> class("AirPassengers")
[1] "character"
> start("AirPassengers")
[1] 1 1
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
> plot(AirPassengers)
```



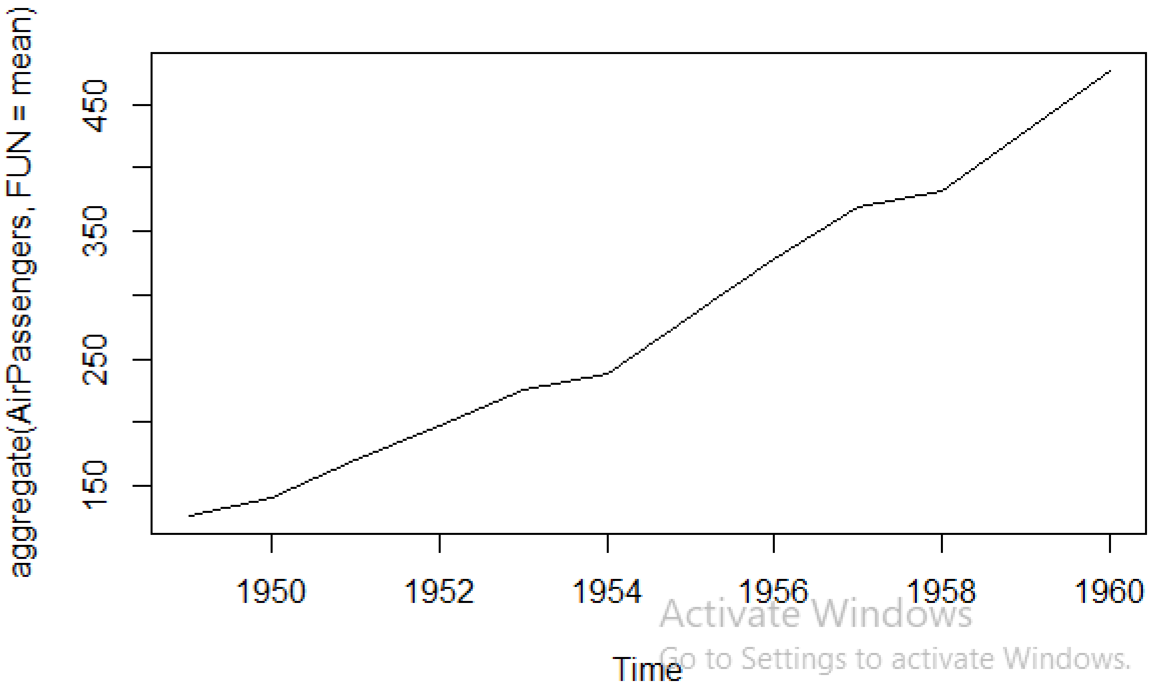
```
> abline(reg=lm(AirPassengers ~ time(AirPassengers)))
```



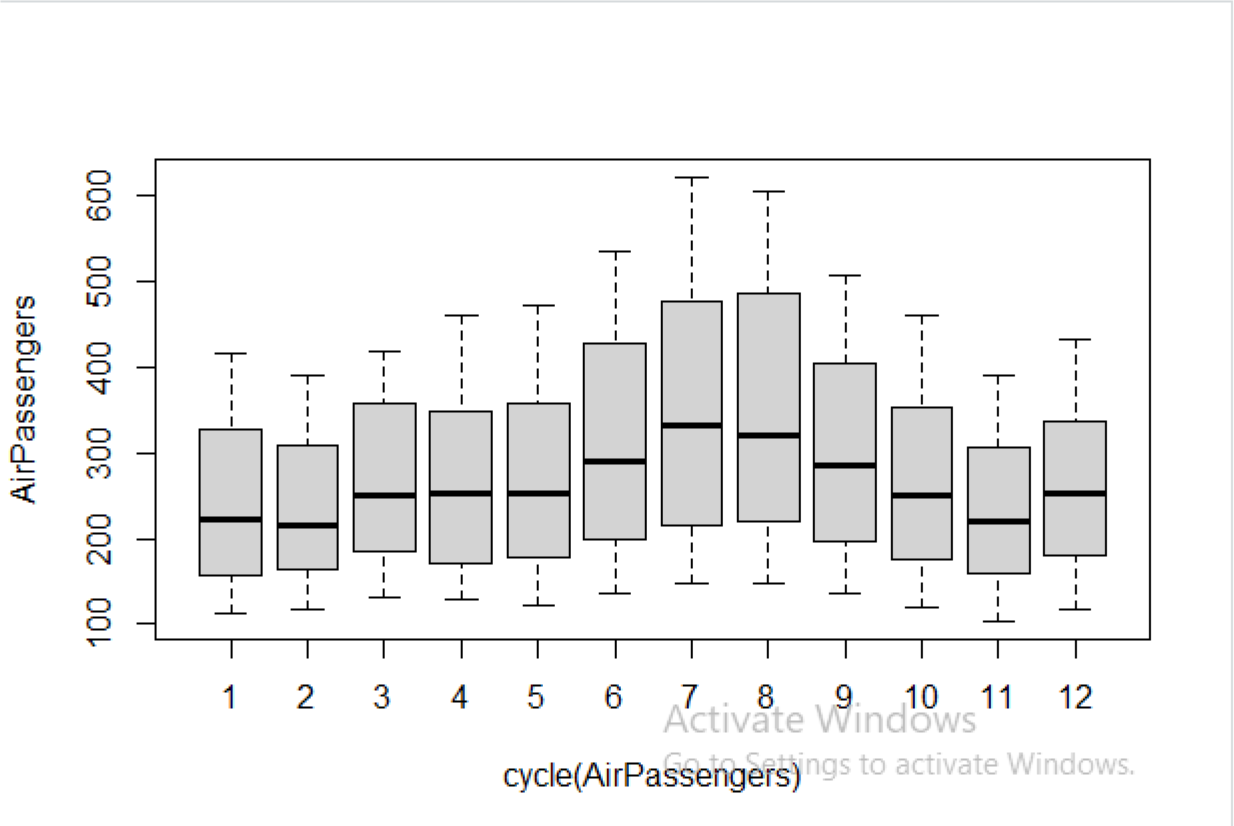
```
> cycle(AirPassengers)
```

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949   1   2   3   4   5   6   7   8   9  10  11  12
1950   1   2   3   4   5   6   7   8   9  10  11  12
1951   1   2   3   4   5   6   7   8   9  10  11  12
1952   1   2   3   4   5   6   7   8   9  10  11  12
1953   1   2   3   4   5   6   7   8   9  10  11  12
1954   1   2   3   4   5   6   7   8   9  10  11  12
1955   1   2   3   4   5   6   7   8   9  10  11  12
1956   1   2   3   4   5   6   7   8   9  10  11  12
1957   1   2   3   4   5   6   7   8   9  10  11  12
1958   1   2   3   4   5   6   7   8   9  10  11  12
1959   1   2   3   4   5   6   7   8   9  10  11  12
1960   1   2   3   4   5   6   7   8   9  10  11  12
```

```
> plot(aggregate(AirPassengers,FUN=mean))
```



```
> boxplot(AirPassengers ~ cycle(AirPassengers))
```


AirPassengers
100
200
300
400
500
600
1
2
3
4
5
6
7
8
9
10
11
12
cycle(AirPassengers)
Activate Windows
Go to Settings to activate Windows.

PRACTICAL NO :- 4

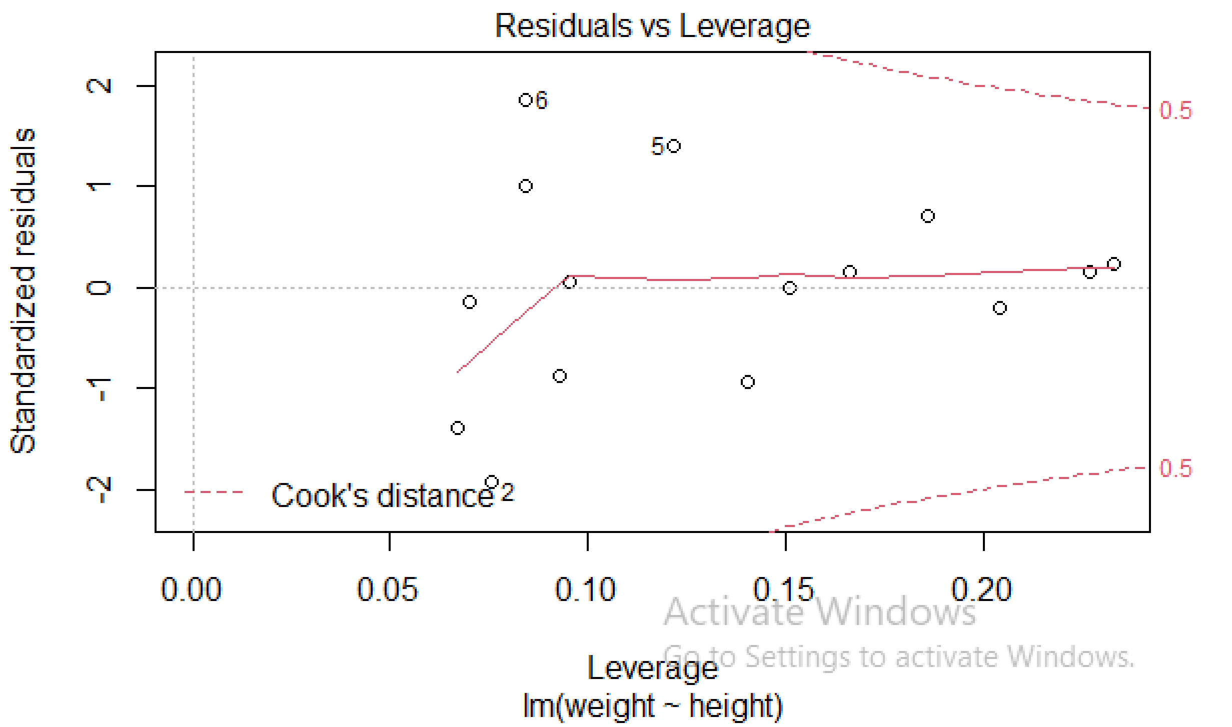
AIM :- *Practical of Simple / Multiple Linear Regression*

CODE :-

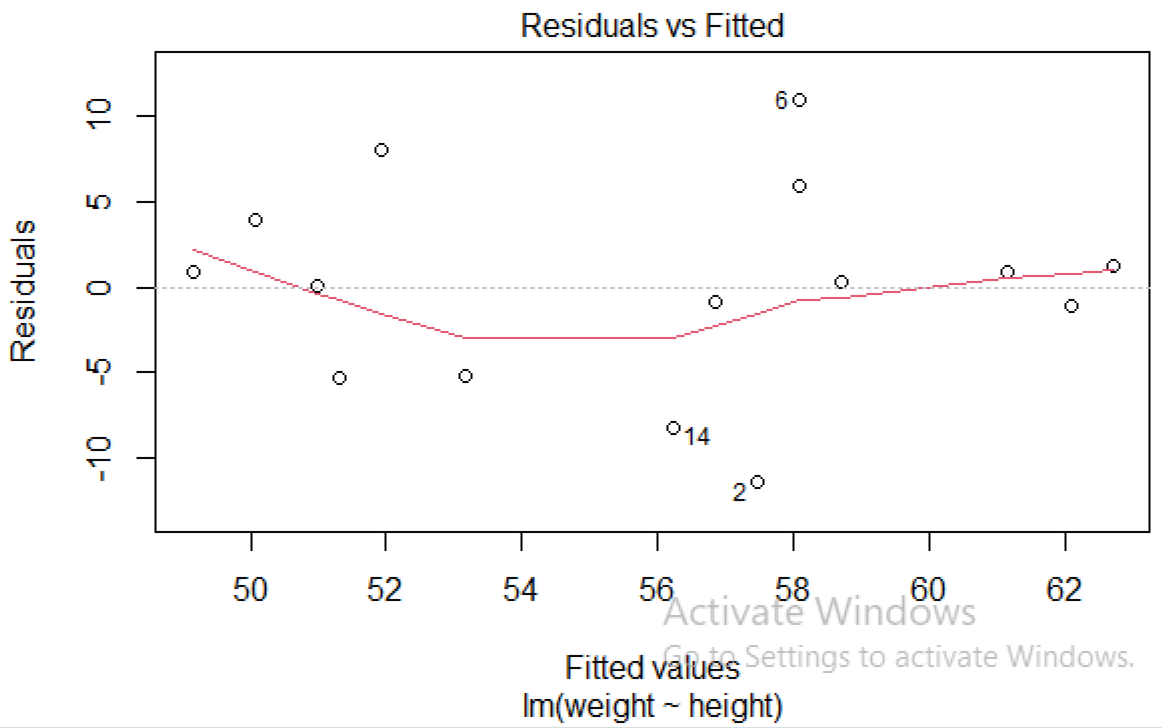
```
> height <-
c(102,117,105,141,135,115,138,144,137,100,131,119,115,121,113)
> weight <- c(61,46,62,54,60,69,51,50,46,64,48,56,64,48,59)
> student <- lm(weight ~ height)
> student
> predict(student,data,frame(height=199),interval="confidence")
> plot(student)
```

OUTPUT :-

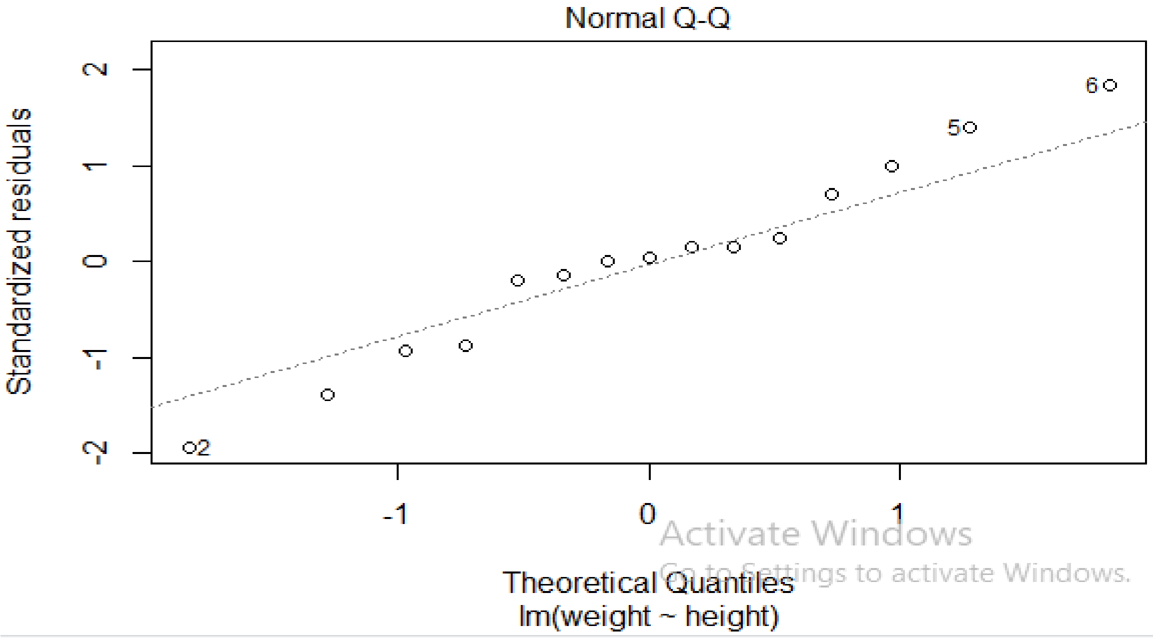
```
> height <-
c(102,117,105,141,135,115,138,144,137,100,131,119,115,121,113)
> weight <- c(61,46,62,54,60,69,51,50,46,64,48,56,64,48,59)
> student <- lm(weight ~ height)
> student

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)      height
   93.5530      -0.3084

> predict(student,data,frame(height=199),interval="confidence")
> plot(student)
Hit <Return> to see next plot:
```



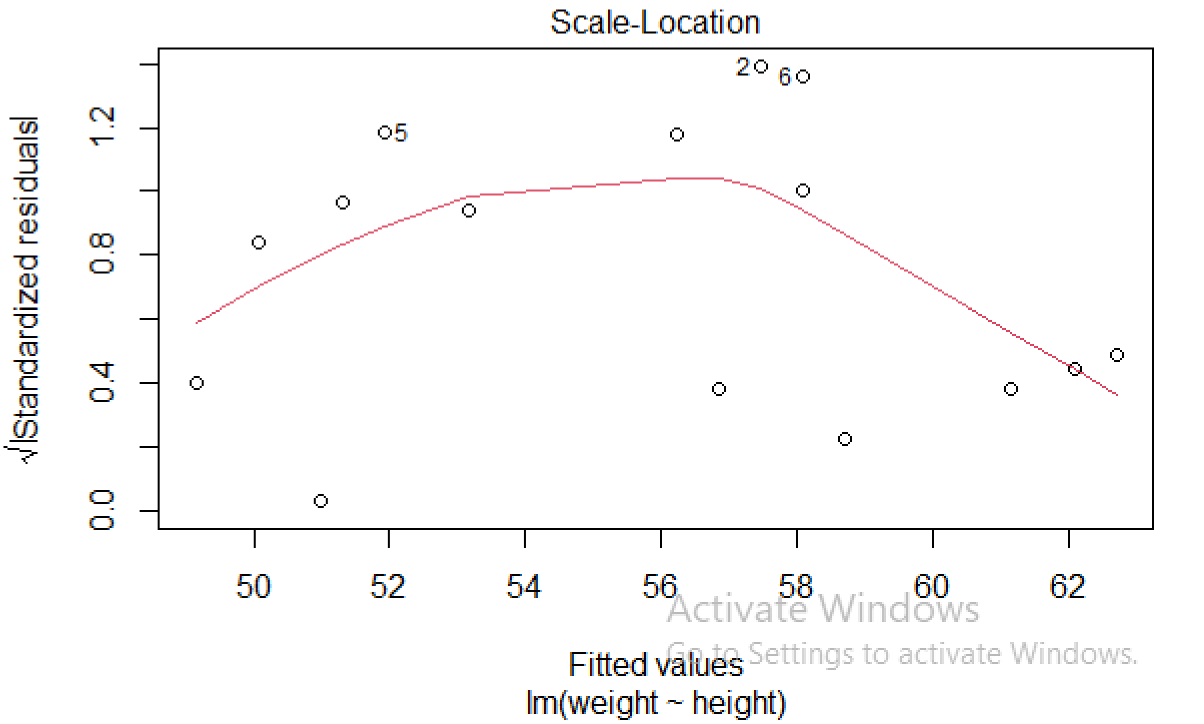
```
Hit <Return> to see next plot:
```



Hit <Return> to see next plot:



Hit <Return> to see next plot:



PRACTICAL NO. :- 5

AIM :- *Practical of Hypothesis testing*

CODE :-

```
> dataf <- seq(1,20,by=1)
> dataf
> mean(dataf)
> sd(dataf)
> a <- t.test(dataf,alternate="two sided",mu=10,conf.int=0.95)
> a
> a$p.value
> a$statistic
> (10.5-10)/(sd(dataf)/sqrt(length(dataf)))
> length(dataf)=1
> length(dataf)
> dataf
> dataf <- seq(1,20,by=1)
> length(dataf)-1
```

OUTPUT :-

```
> dataf <- seq(1,20,by=1)
> dataf
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> mean(dataf)
[1] 10.5
> sd(dataf)
[1] 5.91608
> a <- t.test(dataf,alternate="two sided",mu=10,conf.int=0.95)
> a

        One Sample t-test
```

```
data:  dataf
t = 0.37796, df = 19, p-value = 0.7096
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  7.731189 13.268811
sample estimates:
mean of x
     10.5

> a$p.value
[1] 0.7096465
> a$statistic
        t
0.3779645
> (10.5-10)/(sd(dataf)/sqrt(length(dataf)))
[1] 0.3779645

> length(dataf)=1
> length(dataf)
[1] 1
> dataf
[1] 1
> dataf <- seq(1,20,by=1)
> length(dataf)-1
[1] 19
```

Practical No. :- 6

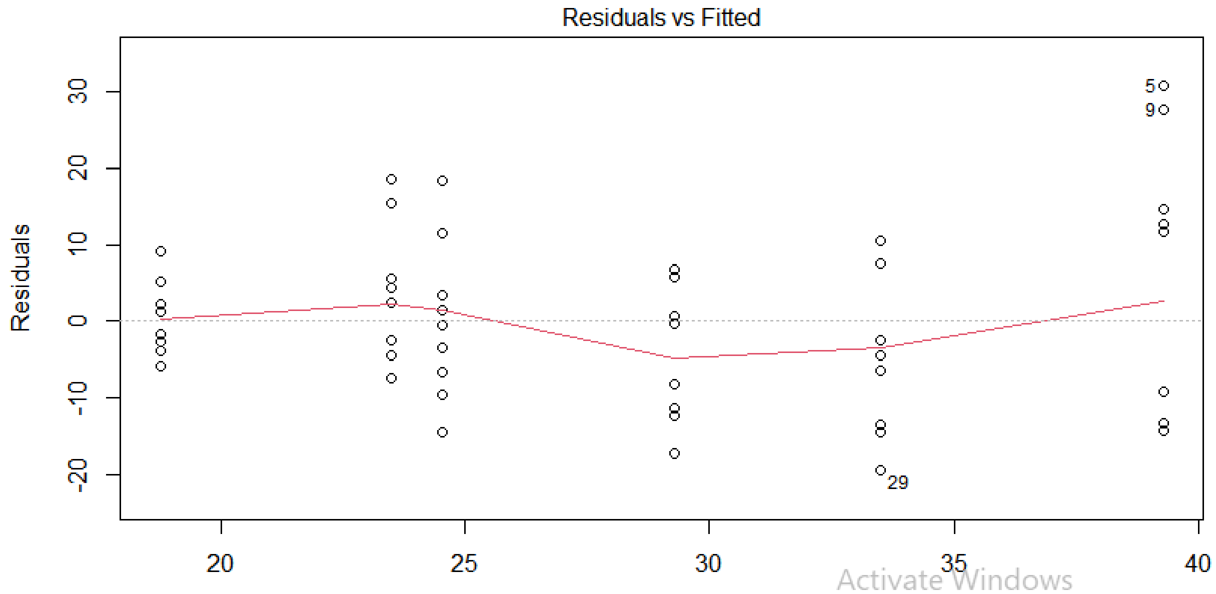
AIM :- *Practical of Analysis of Variance*

Code:-

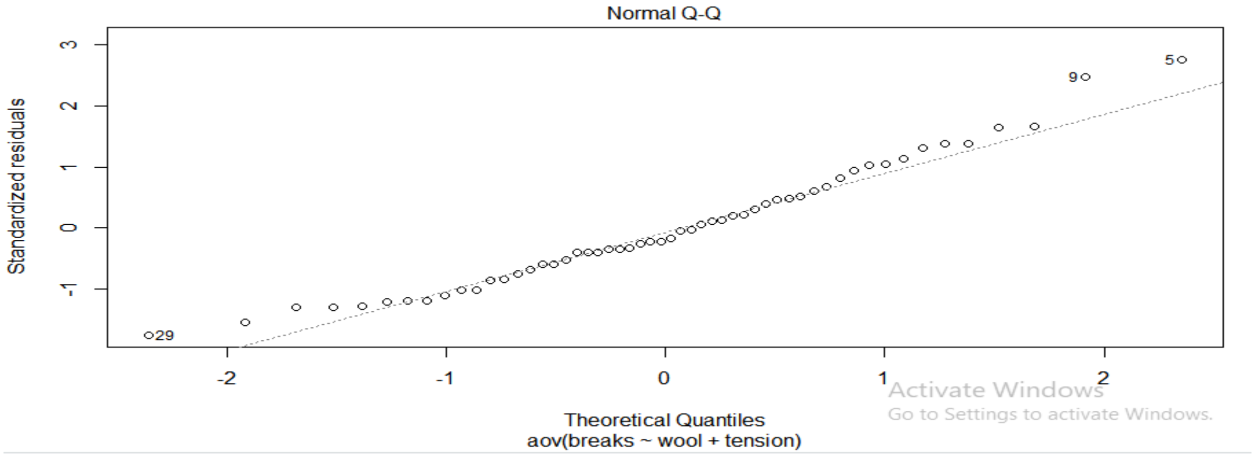
```
data("warpbreaks")
head(warpbreaks)
summary(warpbreaks)
Model_1 <- aov(breaks~wool+tension,data=warpbreaks)
summary(Model_1)
plot(Model_1)
Model_2 <-aov(breaks~wool+tension+wool:tension,data = warpbreaks)
summary(Model_2)
plot(Model_2)
```

Output :-

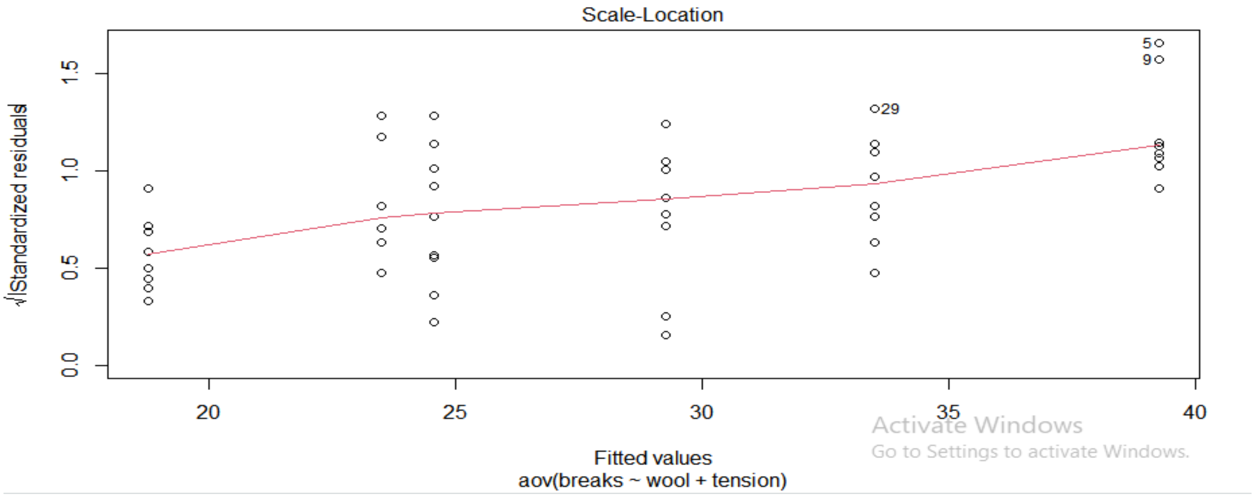
```
> data("warpbreaks")
> head(warpbreaks)
  breaks wool tension
1     26    A       L
2     30    A       L
3     54    A       L
4     25    A       L
5     70    A       L
6     52    A       L
> summary(warpbreaks)
     breaks      wool   tension
 Min.   :10.00   A:27   L:18
 1st Qu.:18.25   B:27   M:18
 Median :26.00          H:18
 Mean   :28.15
 3rd Qu.:34.00
 Max.   :70.00
> Model_1 <- aov(breaks~wool+tension,data=warpbreaks)
> summary(Model_1)
            Df Sum Sq Mean Sq F value  Pr(>F)
wool         1    451   450.7   3.339 0.07361 .
tension      2   2034  1017.1   7.537 0.00138 **
Residuals   50   6748   135.0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> plot(Model_1)
Hit <Return> to see next plot:
```



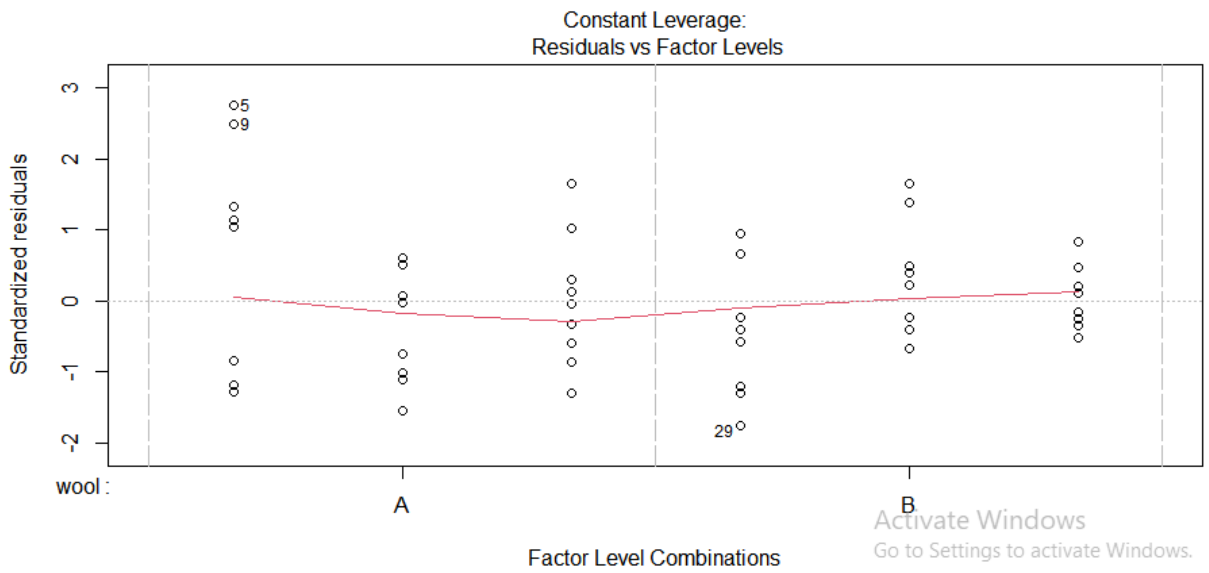
```
Hit <Return> to see next plot:
```



Hit <Return> to see next plot:

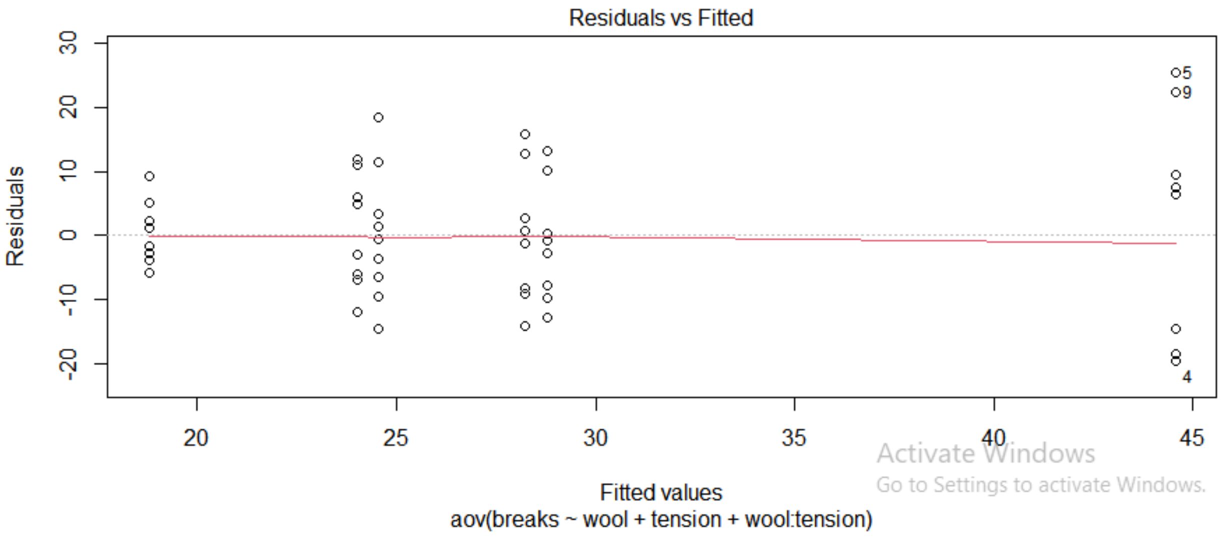


Hit <Return> to see next plot:

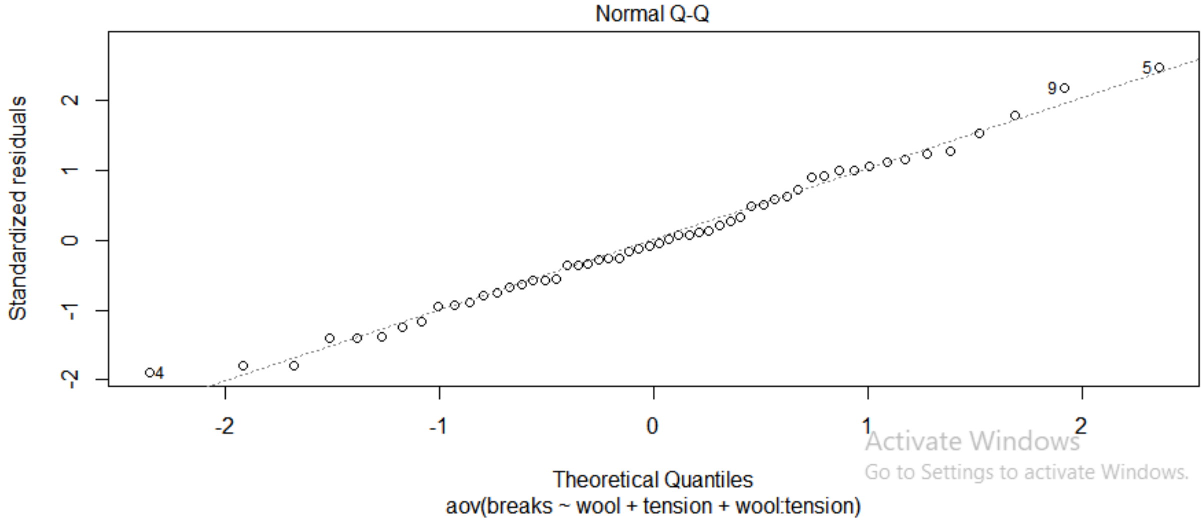


```
> Model_2 <-aov(breaks~wool+tension+wool:tension,data = warpbreaks)
> summary(Model_2)

             Df Sum Sq Mean Sq F value   Pr(>F)
wool          1    451   450.7   3.765 0.058213 .
tension       2   2034  1017.1   8.498 0.000693 ***
wool:tension  2   1003   501.4   4.189 0.021044 *
Residuals    48   5745   119.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

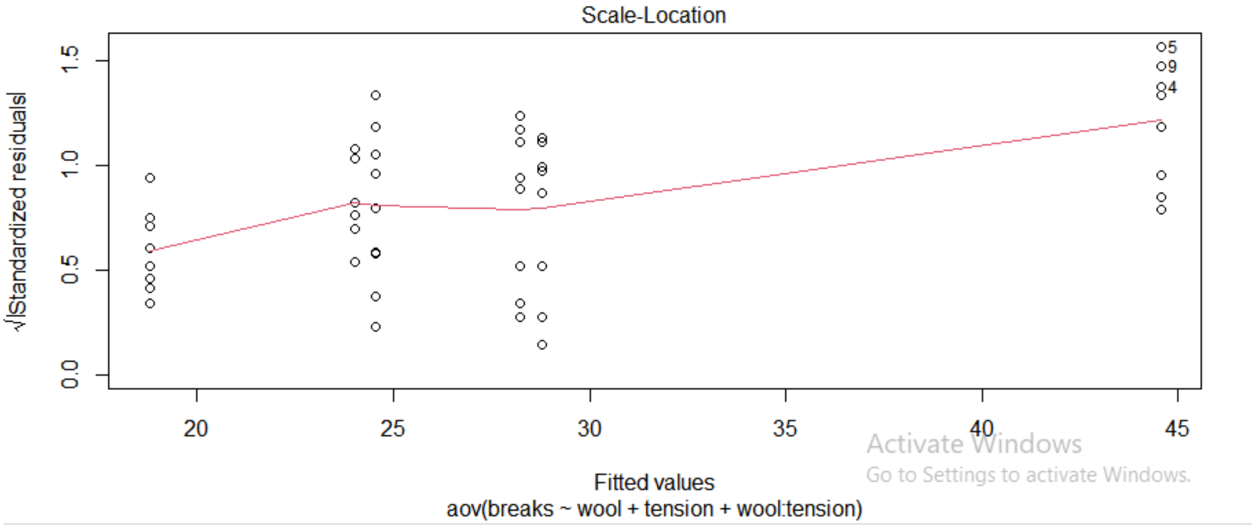
> plot(Model_2)
Hit <Return> to see next plot:



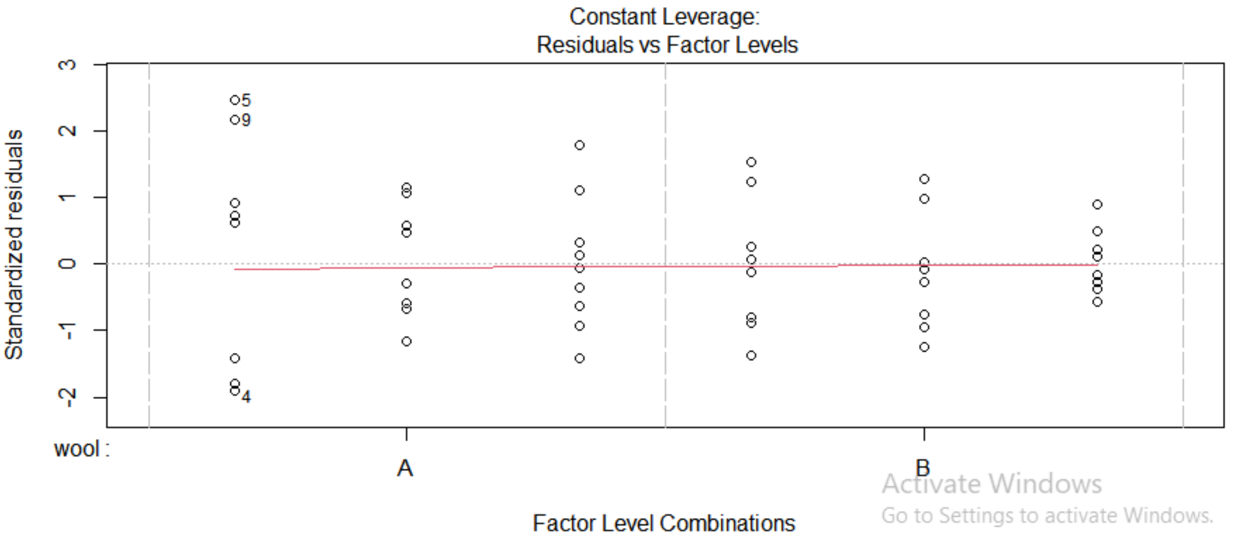
Hit <Return> to see next plot:



Hit <Return> to see next plot:



Hit <Return> to see next plot:



PRACTICAL NO. :- 7

AIM :- *Practical of Logistics Regression*

CODE :

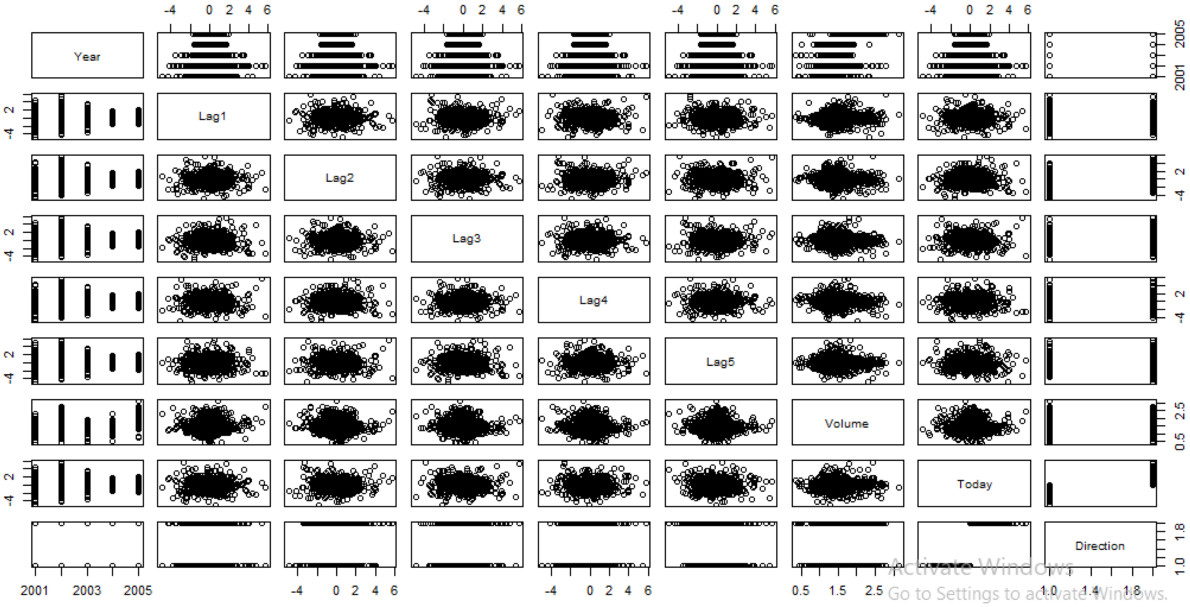
```
> rm(list=ls())
> library(ISLR)
> names(Smarket)
> dim(Smarket)
> summary(Smarket)
> pairs(Smarket)
> ?Smarket
> cor(Smarket[,-9])
> attach(Smarket)
> par(mfrow=c(1,1))
> plot(Volume)
> glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial)
> summary(glm.fits)
> coef(glm.fits)
> summary(glm.fits)$coef
> summary(glm.fits)$coef[,4]
> glm.probs=predict(glm.fits,type="response")
> glm.probs[1:10]
> contrasts(Direction)
> glm.pred=rep("Down",1250)
> glm.pred[glm.probs>.5]="Up"
> glm.probs[1:10]
> glm.pred[1:10]
> table(glm.pred,Direction)
> (507+145)/1250
> mean(glm.pred==Direction)
> train=(Year<2005)
> Smarket.2005 = Smarket[!train,]
> dim(Smarket.2005)
> Direction.2005=Direction[!train]
> glm.fits=glm(Direction ~
Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial,sub
set=train)
> summary(glm.fits)
> glm.probs = predict(glm.fits,Smarket.2005,type="response")
> glm.pred=rep("Down",252)
> glm.pred[glm.probs>.5]="Up"
> table(glm.pred,Direction.2005)
> mean(glm.pred == Direction.2005)
> mean(glm.pred!=Direction.2005)
> glm.fits=glm(Direction ~
Lag1+Lag2,data=Smarket,family=binomial,subset=train)
> glm.probs=predict(glm.fits,Smarket.2005,type = "response")
> mean(glm.pred == Direction.2005)
> (106+35)/252
> 106/(106+35)
> 76/(36+76)
```

<u>OUTPUT :-</u>

```
> rm(list=ls())
> library(ISLR)
> names(Smarket)
[1] "Year"    "Lag1"    "Lag2"    "Lag3"    "Lag4"    "Lag5"
[7] "Volume"   "Today"    "Direction"
> dim(Smarket)
[1] 1250    9
> summary(Smarket)
     Year          Lag1              Lag2              Lag3
```

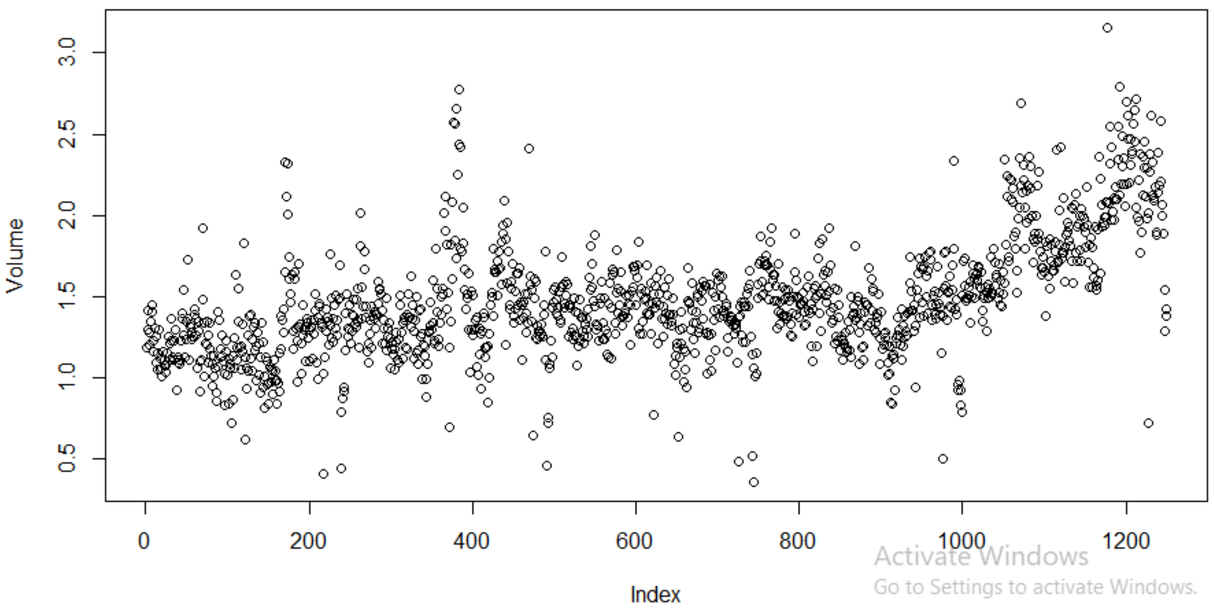
```
 Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000
 1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000
 Median :2003   Median : 0.039000   Median : 0.039000   Median :
0.038500
 Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716
 3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750
 Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000
      Lag4               Lag5             Volume          Today
 Min.   :-4.922000   Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000
 1st Qu.:-0.640000   1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500
 Median : 0.038500   Median : 0.03850   Median :1.4229   Median :
0.038500
 Mean   : 0.001636   Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138
 3rd Qu.: 0.596750   3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750
 Max.   : 5.733000   Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000
 Direction
 Down:602
 Up  :648
```

```
> pairs(Smarket)
```



```
> ?Smarket
> cor(Smarket[,-9])
            Year        Lag1        Lag2        Lag3        Lag4        Lag5
Volume
Year   1.00000000  0.029699649  0.030596422  0.033194581
0.035688718  0.029787995  0.53900647
Lag1   0.02969965  1.000000000 -0.026294328 -0.010803402 -
0.002985911 -0.005674606  0.04090991
Lag2   0.03059642 -0.026294328  1.000000000 -0.025896670 -
0.010853533 -0.003557949 -0.04338321
Lag3   0.03319458 -0.010803402 -0.025896670  1.000000000 -
0.024051036 -0.018808338 -0.04182369
Lag4   0.03568872 -0.002985911 -0.010853533 -0.024051036
1.000000000 -0.027083641 -0.04841425
Lag5   0.02978799 -0.005674606 -0.003557949 -0.018808338 -
0.027083641  1.000000000 -0.02200231
Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -
0.048414246 -0.022002315  1.00000000
Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -
0.006899527 -0.034860083  0.01459182
             Today
```

```
Year    0.030095229
Lag1   -0.026155045
Lag2   -0.010250033
Lag3   -0.002447647
Lag4   -0.006899527
Lag5   -0.034860083
Volume  0.014591823
Today   1.000000000

> attach(Smarket)
> par(mfrow=c(1,1))
> plot(Volume)
```



```
>
glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smark
et,family=binomial)
> summary(glm.fits)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket)
```

```
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6

Number of Fisher Scoring iterations: 3

> coef(glm.fits)
 (Intercept)         Lag1         Lag2         Lag3         Lag4         Lag5
Volume
-0.126000257 -0.073073746 -0.042301344  0.011085108  0.009358938
0.010313068  0.135440659

> summary(glm.fits)$coef
                Estimate Std. Error    z value  Pr(>|z|)
(Intercept) -0.126000257 0.24073574 -0.5233966 0.6006983
Lag1        -0.073073746 0.05016739 -1.4565986 0.1452272
Lag2        -0.042301344 0.05008605 -0.8445733 0.3983491
Lag3         0.011085108 0.04993854  0.2219750 0.8243333
```

```
Lag4         0.009358938 0.04997413  0.1872757 0.8514445
Lag5         0.010313068 0.04951146  0.2082966 0.8349974
Volume       0.135440659 0.15835970  0.8552723 0.3924004

> summary(glm.fits)$coef[,4]
(Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5      Volume
  0.6006983   0.1452272   0.3983491   0.8243333   0.8514445   0.8349974
0.3924004
> glm.probs=predict(glm.fits,type="response")
> glm.probs[1:10]
        1         2         3         4         5         6         7         8         9        10
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565
0.4926509 0.5092292 0.5176135 0.4888378
> contrasts(Direction)
     Up
Down  0
Up    1

> glm.pred=rep("Down",1250)
> glm.pred[glm.probs>.5]="Up"
> glm.probs[1:10]
        1         2         3         4         5         6         7         8         9        10
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565
0.4926509 0.5092292 0.5176135 0.4888378

> glm.pred[1:10]
 [1] "Up"   "Down" "Down" "Up"   "Up"   "Up"   "Down" "Up"   "Up"   "Down"

> table(glm.pred,Direction)
        Direction
glm.pred Down  Up
    Down  145 141
    Up    457 507
> (507+145)/1250
[1] 0.5216
```

```
> mean(glm.pred==Direction)
[1] 0.5216
> train=(Year<2005)
> Smarket.2005 = Smarket[!train,]
> dim(Smarket.2005)
[1] 252   9
> Direction.2005=Direction[!train]
> glm.fits=glm(Direction ~
Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial,sub
set=train)
> summary(glm.fits)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket, subset = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.302  -1.190   1.079   1.160   1.350

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.191213   0.333690   0.573    0.567
Lag1        -0.054178   0.051785  -1.046    0.295
Lag2        -0.045805   0.051797  -0.884    0.377
Lag3         0.007200   0.051644   0.139    0.889
Lag4         0.006441   0.051706   0.125    0.901
Lag5        -0.004223   0.051138  -0.083    0.934
Volume      -0.116257   0.239618  -0.485    0.628

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1383.3  on 997  degrees of freedom
Residual deviance: 1381.1  on 991  degrees of freedom
AIC: 1395.1
```

```
Number of Fisher Scoring iterations: 3

> glm.probs = predict(glm.fits,Smarket.2005,type="response")
> glm.pred=rep("Down",252)
> glm.pred[glm.probs>.5]="Up"
> table(glm.pred,Direction.2005)
        Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
> mean(glm.pred == Direction.2005)
[1] 0.4801587
> mean(glm.pred! = Direction.2005)
Error: unexpected '!' in "mean(glm.pred!"
> mean(glm.pred!=Direction.2005)
[1] 0.5198413
> glm.fits=glm(Direction ~
Lag1+Lag2,data=Smarket,family=binomial,subset=train)
> glm.probs=predict(glm.fits,Smarket.2005,type = "response")
> mean(glm.pred == Direction.2005)
[1] 0.4801587
> (106+35)/252
[1] 0.5595238
> 106/(106+35)
[1] 0.751773
> 76/(36+76)
[1] 0.6785714
```