1. Perform clustering (Both hierarchical and K means clustering) for the airlines data to obtain optimum number of clusters.

Draw the inferences from the clusters obtained.

Data Description:

The file EastWestAirlinescontains information on passengers who belong to an airline’s frequent flier program. For each passenger the data include information on their mileage history and on different ways they accrued or spent miles in the last year. The goal is to try to identify clusters of passengers that have similar characteristics for the purpose of targeting different segments for different types of mileage offers

ID --Unique ID

Balance--Number of miles eligible for award travel

Qual\_mile--Number of miles counted as qualifying for Topflight status

cc1\_miles? CHAR--Has member earned miles with airline freq. flyer credit card in the past 12 months (1=Yes/0=No)?

cc2\_miles?CHAR--Has member earned miles with Rewards credit card in the past 12 months (1=Yes/0=No)?

cc3\_miles?--Has member earned miles with Small Business credit card in the past 12 months (1=Yes/0=No)?

Bonus\_miles--Number of miles earned from non-flight bonus transactions in the past 12 months

Bonus\_trans--Number of non-flight bonus transactions in the past 12 months

Flight\_miles\_12mo--Number of flight miles in the past 12 months

Flight\_trans\_12--Number of flight transactions in the past 12 months

Days\_since\_enrolled--Number of days since enrolled in flier program

Award--whether that person had award flight (free flight) or not

Soln - >

**First we will apply the Non-hierarchical Clustering->**

Code - >

> #problem 1

> #to find Optimum number of clusters

> # Loading

> library("xlsx")

> inp<- read.xlsx(file.choose(),1)

> ninp<-scale(inp[,2:12])

>

> d<-dist(ninp,method="euclidean")

> fit<-hclust(d,method="complete")

>

> str(fit)

List of 7

$ merge : int [1:3998, 1:2] -3371 -2679 -3492 -3705 -2771 -1203 -3727 -2216 -3639 -3515 ...

$ height : num [1:3998] 0 0.00226 0.00271 0.00315 0.00315 ...

$ order : int [1:3999] 3236 3595 2016 3584 2365 3339 905 851 1947 1301 ...

$ labels : NULL

$ method : chr "complete"

$ call : language hclust(d = d, method = "complete")

$ dist.method: chr "euclidean"

- attr(\*, "class")= chr "hclust"

>

> fit$order

[1] 3236 3595 2016 3584 2365 3339 905 851 1947 1301 107 1918 1037 2284 2956 1245 2252 385 1879

[20] 1940 2490 511 314 3773 127 152 482 118 1189 319 1116 187 411 44 73 88 1657 630

[39] 824 176 291 428 276 221 467 472 489 421 1178 2606 834 1221 590 897 902 1338 1546

[58] 2437 3333 2143 3490 2970 3123 1696 2036 3222 3634 3748 1908 1477 2892 336 801 1562 1985 2176

[77] 2984 1226 3064 2560 3337 71 1036 1113 2474 3491 1416 533 865 2612 2853 3812 3280 3650 2552

[96] 2763 3627 2140 1630 1958 3053 3272 2794 3121 3039 3208 3421 3433 3296 3644 3702 827 1517 1668

[115] 677 316 1257 22 1501 653 3071 3019 1320 1514 574 1344 1244 1360 1533 1830 1443 194 1119

[134] 1132 1499 2181 2278 3028 2094 2345 2638 1699 1589 2123 651 178 806 208 485 744 308 821

[153] 2050 682 2268 2502 2674 3898 3912 773 1151 1303 3833 1227 1878 1643 2088 2277 1975 2363 2453

[172] 2832 2113 2550 1943 1104 1681 1103 424 270 370 892 973 211 222 726 1307 207 603 1225

[191] 1978 765 1270 956 1155 1259 1286 1380 1274 1543 1723 577 575 688 836 1497 106 86 148

[210] 506 547 686 487 204 292 138 807 21 329 500 640 1043 197 232 373 609 2071 384

[229] 1398 256 348 964 1395 153 33 156 189 516 1000 619 702 705 923 625 782 1059 1603

[248] 193 468 293 168 260 1841 438 466 245 1634 3138 906 1112 257 1669 3041 1429 2005 112

[267] 709 869 1624 3757 1460 2215 1504 2429 698 1127 1506 912 1625 169 479 173 694 1766 724

[286] 1085 408 431 1290 823 546 1080 558 1081 1557 115 238 46 1220 659 1670 2141 1844 889

[305] 1639 64 353 919 1003 2565 650 870 326 531 1961 2846 3099 2114 2733 1109 1522 1534 1214

[324] 2095 1717 3752 707 1147 1293 1241 2219 1869 1405 1521 2294 2542 2621 1149 1418 1600 2131 3400

[343] 2814 3864 3451 2818 3905 3109 3783 2837 2380 3085 2194 1962 2255 2291 2487 2997 1795 1799 3756

[362] 2188 2951 2574 3231 3253 3158 3832 1683 1848 1991 23 443 1343 1

5013 0.199232115 0.199339227

[ reached getOption("max.print") -- omitted 2998 entries ]

> plot(fit,hang=-1)

>

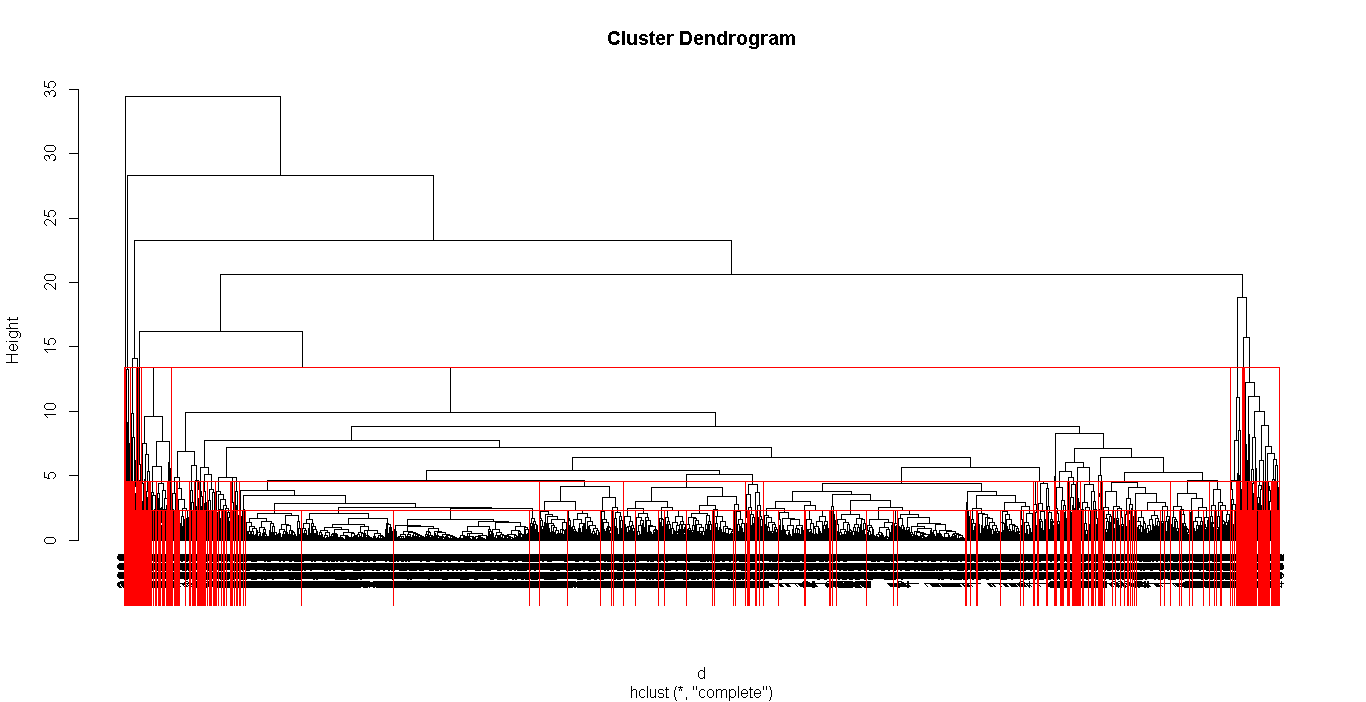
> rect.hclust(fit,k=10,border="red")

>

> rect.hclust(fit,k=100,border="red")

>

> rect.hclust(fit,k=350,border="red")



> #from the dendogram we can see it is difficult to draw any inference on the data

> #we will go for k=10 for now

> groups<- cutree(fit,k=10)

> #groups<- cutree(fit,k=350)

>

> membership<-as.matrix(groups) # groups or cluster numbers

> final <- data.frame(inp, membership)

>

> View(final)

>

> write.csv(final, file="final.csv",row.names = F)

> getwd()

[1] "C:/MY THING$/Data Science/Assignment 7 - Clustering"

>

> #now to get the mean values of the parameters in all the groups

> aggregate(final[,-1],by = list(final$membership),mean)

Group.1 Balance Qual\_miles cc1\_miles cc2\_miles cc3\_miles Bonus\_miles Bonus\_trans

1 1 64838.10 25.51528 2.048009 1.000000 1.000818 15685.25 10.735679

2 2 68876.58 23.25581 1.139535 2.348837 1.000000 14689.84 17.534884

3 3 95038.42 3143.51429 1.514286 1.000000 1.000000 11642.19 9.838095

4 4 131528.15 348.18033 2.565574 1.000000 1.000000 37762.43 29.983607

5 5 718125.39 44.52174 3.782609 1.000000 1.000000 63434.83 20.217391

6 6 138061.40 78.80000 3.466667 1.000000 4.066667 93927.87 28.066667

7 7 180579.20 0.00000 4.200000 1.000000 1.000000 225128.00 26.400000

8 8 1212649.60 1941.20000 2.600000 1.000000 1.000000 35308.80 25.800000

9 9 142376.18 8720.72727 2.181818 1.000000 1.000000 15125.36 15.454545

10 10 131999.50 347.00000 2.500000 1.000000 1.000000 65634.25 69.250000

Flight\_miles\_12mo Flight\_trans\_12 Days\_since\_enroll Award. membership

1 253.7139 0.7626841 4069.739 0.3434261 1

2 582.6279 2.2093023 3968.930 0.3953488 2

3 747.9810 2.4285714 4027.714 0.5428571 3

4 5375.0656 15.5819672 4950.566 0.8032787 4

5 929.1304 3.5652174 6616.348 0.8260870 5

6 506.6667 1.6000000 4613.867 0.5333333 6

7 2103.0000 4.8000000 4930.800 1.0000000 7

8 3254.6000 15.0000000 7845.800 1.0000000 8

9 1336.4545 4.0000000 5349.364 0.8181818 9

10 19960.0000 49.2500000 2200.250 1.0000000 10

**As we can see hierarchical clustering doesn’t makes any sense as the data as the dataset is too large.**

> #my code clustering -> k means

>

> library("xlsx")

> inp1<- read.xlsx(file.choose(),1)

> mydata<-scale(inp1[,-1] )

>

> #install.packages("kselection")

> library(kselection)

> k<-kselection(mydata,parallel = TRUE,k\_threshold = 0.95,max\_centers = 30)

> k

f(k) finds 15 clusters>

> twss=NULL

> for (i in 1:30)

+ {

+ twss[i]=sum(kmeans(mydata,i)$tot.withinss)

+

+ }

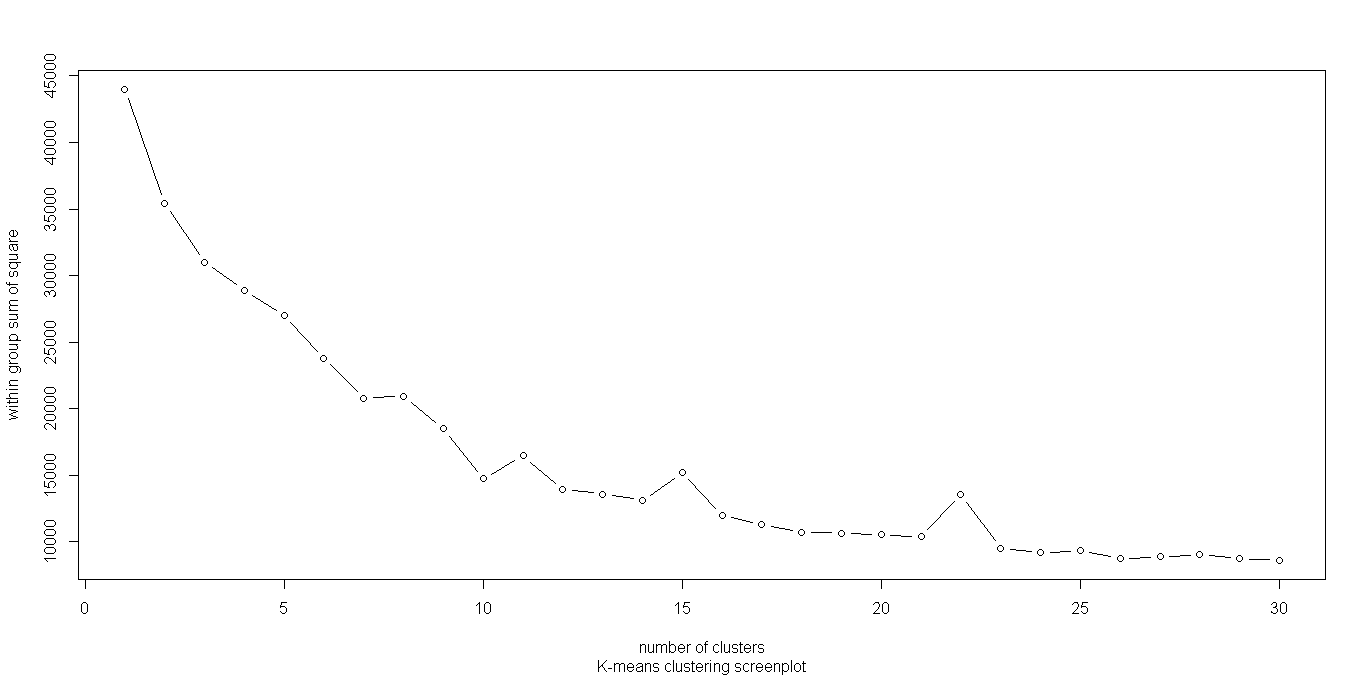
> warning()

Warning message:

> windows()

> plot(1:30,twss,type = "b",xlab = "number of clusters",ylab = "within group sum of square")

> title(sub="K-means clustering screenplot")

> 

> setwd("C:/MY THING$/Data Science/Assignment 7 - Clustering")

>

> **#from the plot we can see good clusters can be formed at k = 4 and k=6**

>

> fit<-kmeans(mydata,4)

> sum(fit$tot.withinss)

[1] 28587.01

> sum(kmeans(mydata,4)$tot.withinss)

[1] 28902.4

> str(fit)

List of 9

$ cluster : int [1:3999] 3 3 3 3 1 3 1 4 2 1 ...

$ centers : num [1:4, 1:11] 0.6316 1.2136 -0.2988 -0.1568 -0.0124 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:4] "1" "2" "3" "4"

.. ..$ : chr [1:11] "Balance" "Qual\_miles" "cc1\_miles" "cc2\_miles" ...

$ totss : num 43978

$ withinss : num [1:4] 10367 4814 6377 7029

$ tot.withinss: num 28587

$ betweenss : num 15391

$ size : int [1:4] 909 152 2097 841

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

> fit$cluster

[1] 3 3 3 3 1 3 1 4 2 1 3 1 3 3 3 1 1 4 1 4 1 4 3 3 3 3 3 3 1 4 1 3 1 3 3 1 4 3 1 4 3 1 2 1 1 4 3 4 1

[50] 3 2 3 1 4 3 3 1 4 3 1 1 3 3 3 3 2 3 1 1 4 4 1 1 3 1 1 3 1 1 1 4 3 3 3 3 1 4 1 3 4 4 3 1 4 2 3 3 1

[99] 3 4 1 4 3 4 3 1 1 4 1 1 1 4 2 4 4 1 3 1 1 4 3 1 1 1 1 1 1 2 4 1 4 4 4 4 4 2 1 1 4 3 3 3 3 1 3 4 3

[148] 1 1 3 1 1 1 3 1 1 1 4 1 3 1 1 4 1 4 3 3 1 1 1 1 3 1 4 1 1 4 4 4 4 3 4 3 4 1 3 2 3 1 3 1 2 1 3 4 3

[197] 1 1 4 1 1 4 3 1 3 3 1 4 4 3 1 3 3 4 3 1 3 3 1 1 2 1 4 3 1 3 4 1 3 1 1 1 3 4 4 4 3 4 2 3 1 4 1 2 1

[246] 2 3 4 1 3 3 3 1 4 4 1 2 4 3 1 3 3 3 3 4 1 3 3 1 1 4 1 1 3 3 2 3 1 3 1 3 1 3 4 1 1 1 3 1 1 1 1 1 3

[295] 3 1 1 1 2 3 4 3 1 1 3 1 3 2 3 3 1 1 1 1 4 4 3 3 1 4 3 1 1 2 3 4 3 2 1 3 1 3 1 4 1 1 1 3 3 3 1 4 3

[344] 3 3 1 1 1 1 4 3 1 3 4 1 3 3 1 3 3 2 3 1 3 3 4 4 3 3 1 4 2 1 3 4 2 3 1 3 1 2 4 1 1 1 3 3 1 1 3 3 4

[393] 4 3 3 3 3 3 3 3 3 3 3 3 2 1 3 4 3 3 2 1 2 1 3 4 4 4 3 3 2 3 1 1 1 1 3 1 3 1 4 4 4 1 4 3 1 1 1 3 3

[442] 1 3 3 1 4 1 3 1 3 1 3 1 1 3 3 4 3 1 3 3 1 4 3 3 1 2 1 3 4 3 2 1 3 3 2 3 1 1 1 3 1 4 3 4 4 1 4 2 1

[491] 4 3 2 3 1 3 4 4 3 1 2 1 3 1 3 1 1 4 3 3 1 2 1 3 1 1 4 1 1 4 3 3 1 3 4 3 1 1 1 4 4 2 1 3 1 4 4 1 3

[540] 3 3 4 3 3 4 4 1 1 4 2 3 4 3 3 1 4 3 4 3 3 4 4 4 1 1 1 3 1 3 3 4 1 3 3 1 3 1 3 4 3 1 3 3 3 3 3 3 1

[589] 3 4 3 3 4 1 2 3 3 3 1 1 1 1 1 1 4 4 4 3 1 3 3 2 4 4 4 1 3 4 1 1 1 3 2 1 1 1 3 1 3 1 2 3 3 3 3 3 3

[638] 4 1 1 4 3 3 3 3 3 3 4 3 4 4 4 4 1 3 3 3 1 4 4 1 3 2 4 1 3 3 3 3 1 1 3 1 4 1 4 4 4 1 3 3 2 1 2 3 1

[687] 3 1 3 1 3 1 1 1 3 3 3 1 3 3 3 1 2 4 1 1 4 3 2 3 4 1 3 4 1 3 1 4 2 3 3 4 4 1 3 1 1 3 3 3 1 1 3 3 4

[736] 3 1 3 3 1 4 1 3 2 4 1 3 3 3 3 1 4 3 4 1 3 1 1 4 4 4 3 3 1 1 1 3 4 4 1 3 3 2 1 1 3 4 3 3 4 1 1 4 3

[785] 3 3 3 1 4 4 4 1 1 1 3 4 1 1 1 4 1 3 3 4 3 4 1 1 3 3 3 4 1 1 1 4 3 3 1 3 4 3 4 1 3 3 4 3 1 4 1 1 1

[834] 2 4 1 1 1 3 2 1 1 1 3 1 3 3 1 3 1 1 2 3 3 1 1 3 3 1 3 2 1 4 3 1 1 4 3 2 4 4 4 4 3 3 4 3 1 3 3 1 1

[883] 3 1 1 4 4 1 3 4 3 1 3 1 3 3 4 3 3 3 1 1 3 3 1 1 3 3 3 1 3 1 3 3 2 4 4 1 3 1 3 3 1 4 3 4 4 3 3 3 1

[932] 3 1 3 1 1 1 1 4 3 4 4 4 3 3 3 1 1 4 3 3 3 3 1 1 1 3 3 2 1 3 3 1 1 4 4 2 3 4 1 3 4 1 3 3 1 2 1 3 1

[981] 1 3 3 1 1 3 4 3 3 3 3 4 1 1 4 1 3 1 1 1

[ reached getOption("max.print") -- omitted 2999 entries ]

> final2<-data.frame(fit$cluster,inp1)

> write.csv(final2, file="final2.csv",row.names = F)

> aggregate(final2[,-1],by = list(final2$fit.cluster),mean)

Group.1 ID. Balance Qual\_miles cc1\_miles cc2\_miles cc3\_miles Bonus\_miles Bonus\_trans

1 1 1496.204 137267.98 134.95614 4.098684 1.002193 1.006579 47622.625 19.665570

2 2 2280.785 43348.71 70.98669 1.302281 1.016160 1.000475 4817.212 7.219582

3 3 1714.517 198000.90 745.76159 2.390728 1.039735 1.278146 40711.715 28.953642

4 4 1965.218 57739.77 229.88942 1.679087 1.019231 1.000000 10634.041 10.695913

Flight\_miles\_12mo Flight\_trans\_12 Days\_since\_enroll Award.

1 365.9265 1.1206140 5065.708 0.5800439

2 162.1882 0.4857414 3618.301 0.0000000

3 5386.4702 15.7417219 4723.113 0.8211921

4 422.4002 1.2884615 4235.694 0.9951923

>

> fit1<-kmeans(mydata,6)

> final3<- data.frame(fit1$cluster,inp1)

> write.csv(final3, file="final3.csv",row.names = F)

> aggregate(final3[,-1],by = list(final3$fit1.cluster),mean)

Group.1 ID. Balance Qual\_miles cc1\_miles cc2\_miles cc3\_miles Bonus\_miles Bonus\_trans

1 1 1847.365 178141.83 919.08730 1.976190 1.031746 1.000000 28572.071 27.60317

2 2 1860.223 67249.24 224.58167 2.667450 1.000000 1.000000 22228.975 15.60752

3 3 3150.591 32030.91 96.45281 1.068398 1.000000 1.000000 2888.945 5.32381

4 4 1064.393 49568.84 67.55204 1.076923 1.000000 1.001131 2964.549 5.21267

5 5 1204.987 218262.32 146.78125 4.593750 1.000000 1.146875 73353.291 23.37187

6 6 1924.575 96528.26 77.37707 3.108597 1.081448 1.001508 25060.115 17.19457

Flight\_miles\_12mo Flight\_trans\_12 Days\_since\_enroll Award.

1 6071.0476 17.6507937 4507.262 0.80158730

2 457.5523 1.3783784 4413.268 1.00000000

3 147.5299 0.4493506 2066.797 0.10476190

4 200.0622 0.6006787 5789.183 0.15610860

5 673.2344 2.1468750 5596.703 0.80000000

6 185.1388 0.5414781 4299.819 0.02111614

**The Details of the Groups is given in the csv file named final and final2 and final3**

**Prob2->**

Perform Clustering for the crime data and identify the number of clusters formed and draw inferences.

Data Description:

Murder -- Muder rates in different places of United States

Assualt- Assualt rate in different places of United States

UrbanPop - urban population in different places of United States

Rape - Rape rate in different places of United States

Soln-> > #clustering prob2

>

>

> #to find Optimum number of clusters

>

>

> inp<- read.csv(file.choose())

> ninp<-scale(inp[,-1])

>

> d<-dist(ninp,method="euclidean")

> fit<-hclust(d,method="complete")

>

> str(fit)

List of 7

$ merge : int [1:49, 1:2] -15 -13 -14 -23 -36 -20 -37 -19 -46 -41 ...

$ height : num [1:49] 0.206 0.35 0.429 0.494 0.53 ...

$ order : int [1:50] 41 48 34 45 19 15 29 12 26 27 ...

$ labels : NULL

$ method : chr "complete"

$ call : language hclust(d = d, method = "complete")

$ dist.method: chr "euclidean"

- attr(\*, "class")= chr "hclust"

>

> fit$order

[1] 41 48 34 45 19 15 29 12 26 27 17 4 46 50 25 37 47 8 39 21 30 7 23 49 36 14 16 35 38 11 44 6 5

[34] 28 9 43 13 32 3 22 20 31 2 1 18 10 42 33 24 40

> fit$height

[1] 0.2058539 0.3502188 0.4287712 0.4940832 0.5303259 0.5353893 0.5935343 0.6457158 0.7038309 0.7108812

[11] 0.7389936 0.7722224 0.7781298 0.7865674 0.7977642 0.8286936 0.8412900 0.8457697 0.9824857 0.9971035

[21] 1.0122252 1.0354597 1.0709720 1.0800988 1.0918624 1.1314351 1.1826891 1.1968261 1.2117356 1.2502752

[31] 1.2716808 1.3329504 1.3988595 1.4668378 1.6230495 1.6448574 1.6585736 1.8537984 1.8649801 2.2630242

[41] 2.2952287 2.3374653 2.4458600 2.4748807 3.0883430 3.2554326 4.4005416 4.4200736 6.0766416

> plot(fit,hang=-1)

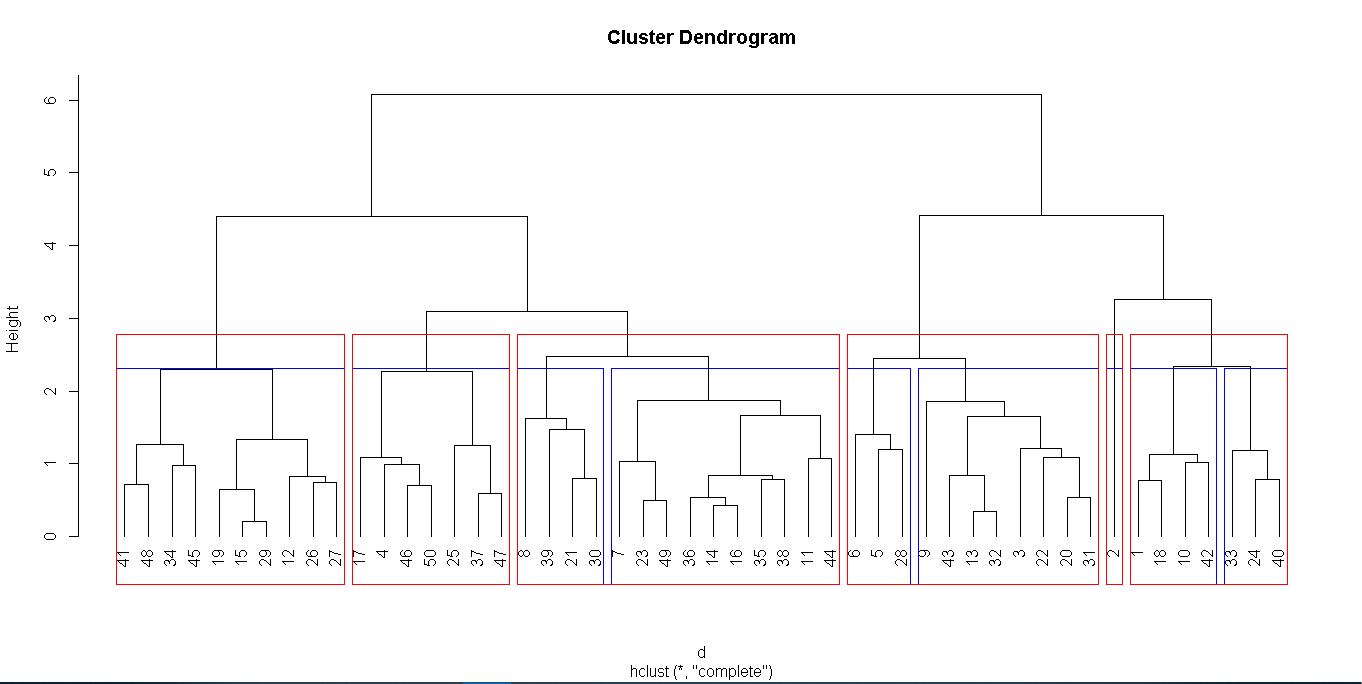
>

> rect.hclust(fit,k=9,border="blue")

>

> rect.hclust(fit,k=6,border="red")

>



>

> groups<- cutree(fit,k=6)

>

> membership<-as.matrix(groups) # groups or cluster numbers

> finalcrime <- data.frame(membership,inp)

>

> View(finalcrime)

>

> write.csv(finalcrime, file="finalcrime.csv",row.names = F)

> getwd()

[1] "C:/MY THING$/Data Science/Assignment 7 - Clustering"

>

> #now to get the mean values of the parameters in all the groups

> aggregate(finalcrime[,-c(1,2)],by = list(finalcrime$membership),mean)

Group.1 Murder Assault UrbanPop Rape

1 1 14.671429 251.2857 54.28571 21.68571

2 2 10.000000 263.0000 48.00000 44.50000

3 3 11.054545 264.0909 79.09091 32.61818

4 4 7.385714 156.8571 62.14286 22.25714

5 5 5.114286 123.2857 75.07143 16.74286

6 6 3.180000 78.7000 49.30000 11.63000

Inference->

**Thus, Group 1 has the max amount of crime Values**

**While Group 6 Seems to have least amount of crime yhus it’s a low crime**

**Area**

**The detail data of the group 1 and 2 is given the csv filed saved as finalcrime**

**#Now we will use hierarchical clustering i.e. k mean clustering**

> #my code clustering -> k means

> #prob 2

>

> inp1<- read.csv(file.choose())

> mydata<-scale(inp1[,-1] )

>

>

> #install.packages("kselection")

> library(kselection)

> k<-kselection(mydata,parallel = TRUE,k\_threshold = 0.95,max\_centers = 30)

> k

f(k) finds 2 clusters>

> twss=NULL

> for (i in 1:30)

+ {

+ twss[i]=sum(kmeans(mydata,i)$tot.withinss)

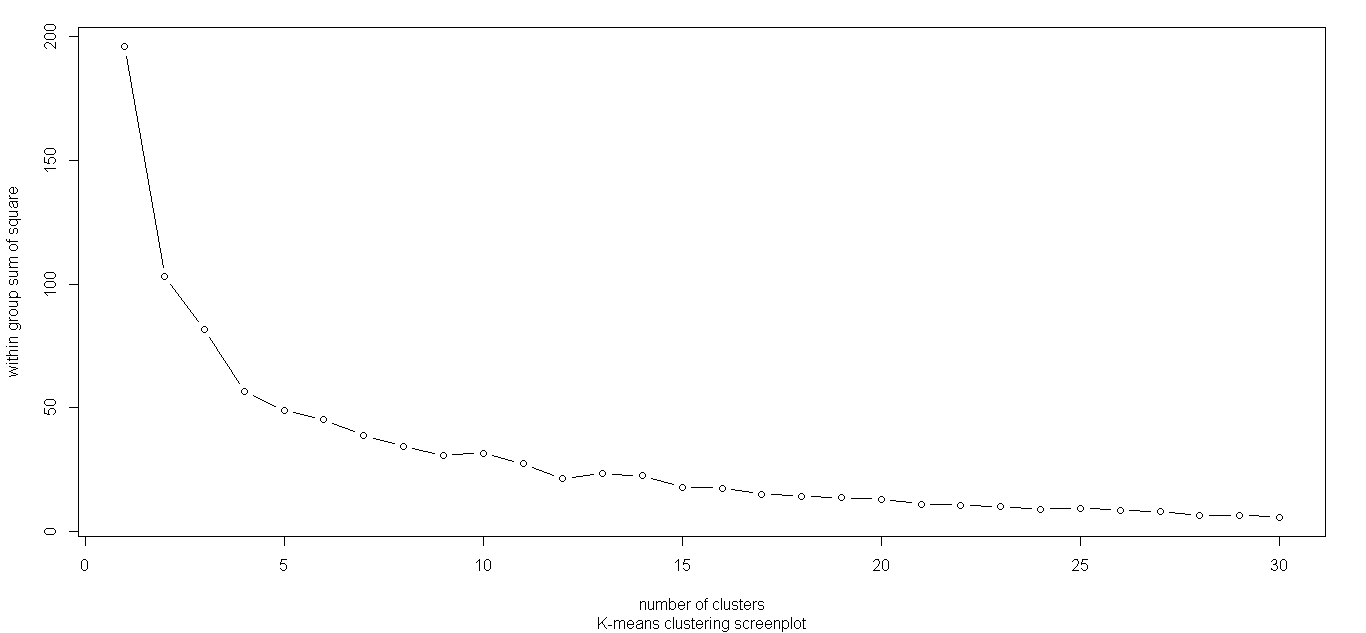
+

+ }

>

> windows()

> plot(1:30,twss,type = "b",xlab = "number of clusters",ylab = "within group sum of square")



> title(sub="K-means clustering screenplot")

>

>

> #from the plot we can see good clusters can be formed at k = 5 is best

>

> fit<-kmeans(mydata,5)

> sum(fit$tot.withinss)

[1] 51.63029

> sum(kmeans(mydata,5)$tot.withinss)

[1] 52.67144

> str(fit)

List of 9

$ cluster : int [1:50] 1 4 4 1 4 4 5 5 4 1 ...

$ centers : num [1:5, 1:4] 1.412 -0.793 -1.118 0.695 -0.507 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:5] "1" "2" "3" "4" ...

.. ..$ : chr [1:4] "Murder" "Assault" "UrbanPop" "Rape"

$ totss : num 196

$ withinss : num [1:5] 8.32 6.23 2.2 19.92 14.96

$ tot.withinss: num 51.6

$ betweenss : num 144

$ size : int [1:5] 8 9 5 13 15

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

> fit$cluster

[1] 1 4 4 1 4 4 5 5 4 1 5 2 4 5 2 5 2 1 3 4 5 4 2 1 4 2 2 4 2 5 4 4 1 3 5 5 5 5 5 1 3 1 4 5 3 5 5 3 2 2

> final2<-data.frame(fit$cluster,inp1)

> write.csv(final2, file="final2crime.csv",row.names = F)

>

> aggregate(final2[,-2],by = list(final2$fit.cluster),mean)

Group.1 fit.cluster Murder Assault UrbanPop Rape

1 1 1 13.937500 243.62500 53.75000 21.41250

2 2 2 4.333333 93.22222 58.44444 13.94444

3 3 3 2.920000 68.60000 42.20000 9.68000

4 4 4 10.815385 257.38462 76.00000 33.19231

5 5 5 5.580000 137.40000 74.80000 18.99333

**As we can see the group3 is the least crime state while group 1 is the group with high crime**

**The Details of the Groups is given in the csv file named final2crime.**