**Perform Principal component analysis and perform clustering using first**

**3 principal component scores (both heirarchial and k mean clustering(scree plot or elbow curve) and obtain**

**optimum number of clusters and check whether we have obtained same number of clusters with the original data**

**(class column we have ignored at the begining who shows it has 3 clusters)df**

#**Clustering using principal companent analysis**

**#here using both hierarchical and k means clustering**

**#Loading the data**

wine<-read.csv(file.choose(),header=T)

View(wine)

**#display first few data in a data set**

head(wine)

Type Alcohol Malic Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids

1 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28

2 1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26

3 1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30

4 1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24

5 1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39

6 1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34

Proanthocyanins Color Hue Dilution Proline

1 2.29 5.64 1.04 3.92 1065

2 1.28 4.38 1.05 3.40 1050

3 2.81 5.68 1.03 3.17 1185

4 2.18 7.80 0.86 3.45 1480

5 1.82 4.32 1.04 2.93 735

6 1.97 6.75 1.05 2.85 1450

**#check if any null values present in the data set**

sum(is.na(wine))

**#there is no null value present in the data set**

**#display the structure of the data set**

str(wine)

data.frame': 178 obs. of 14 variables:

$ Type : int 1 1 1 1 1 1 1 1 1 1 ...

$ Alcohol : num 14.2 13.2 13.2 14.4 13.2 ...

$ Malic : num 1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...

$ Ash : num 2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...

$ Alcalinity : num 15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...

$ Magnesium : int 127 100 101 113 118 112 96 121 97 98 ...

$ Phenols : num 2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...

$ Flavanoids : num 3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...

$ Nonflavanoids : num 0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...

$ Proanthocyanins: num 2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...

$ Color : num 5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...

$ Hue : num 1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...

$ Dilution : num 3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...

$ Proline : int 1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...

**#display the column names of the data set**

colnames(wine)

[1] "Type" "Alcohol" "Malic" "Ash" "Alcalinity"

[6] "Magnesium" "Phenols" "Flavanoids" "Nonflavanoids" "Proanthocyanins"

[11] "Color" "Hue" "Dilution" "Proline"

**#convert the variables in to a pricipal component objects using principal component analysis(PCA)**

**## here we remove 'Type' variable from our data set**

pcaObj<-princomp(wine[-1], cor = TRUE, scores = TRUE, covmat = NULL)

summary(pcaObj)

Importance of components:

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6

Standard deviation 2.1692972 1.5801816 1.2025273 0.9586313 0.92370351 0.80103498

Proportion of Variance 0.3619885 0.1920749 0.1112363 0.0706903 0.06563294 0.04935823

Cumulative Proportion 0.3619885 0.5540634 0.6652997 0.7359900 0.80162293 0.85098116

Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12

Standard deviation 0.74231281 0.59033665 0.53747553 0.50090167 0.47517222 0.41081655

Proportion of Variance 0.04238679 0.02680749 0.02222153 0.01930019 0.01736836 0.01298233

Cumulative Proportion 0.89336795 0.92017544 0.94239698 0.96169717 0.97906553 0.99204785

Comp.13

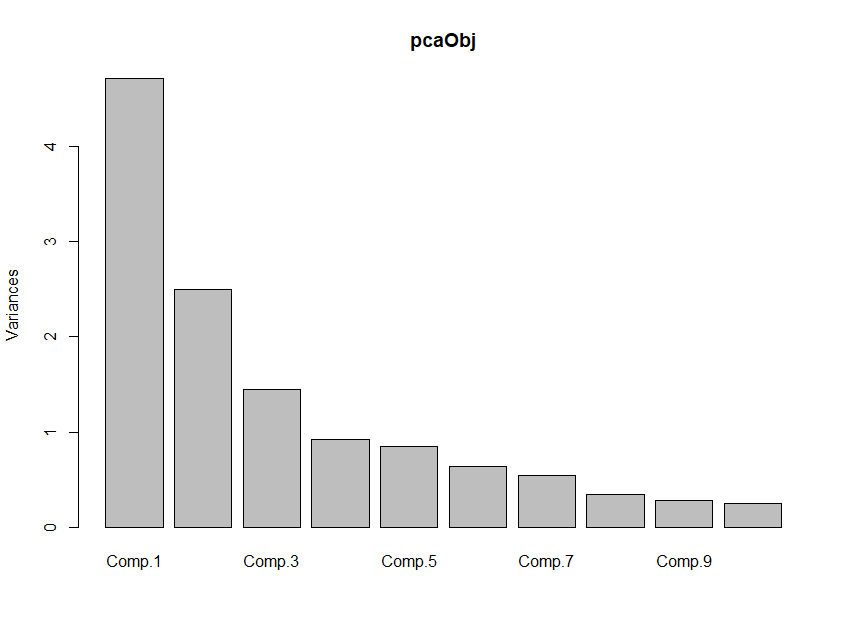
Standard deviation 0.321524394

Proportion of Variance 0.007952149

Cumulative Proportion 1.000000000

**# here creating 13 proncipal component object**

help(loadings)

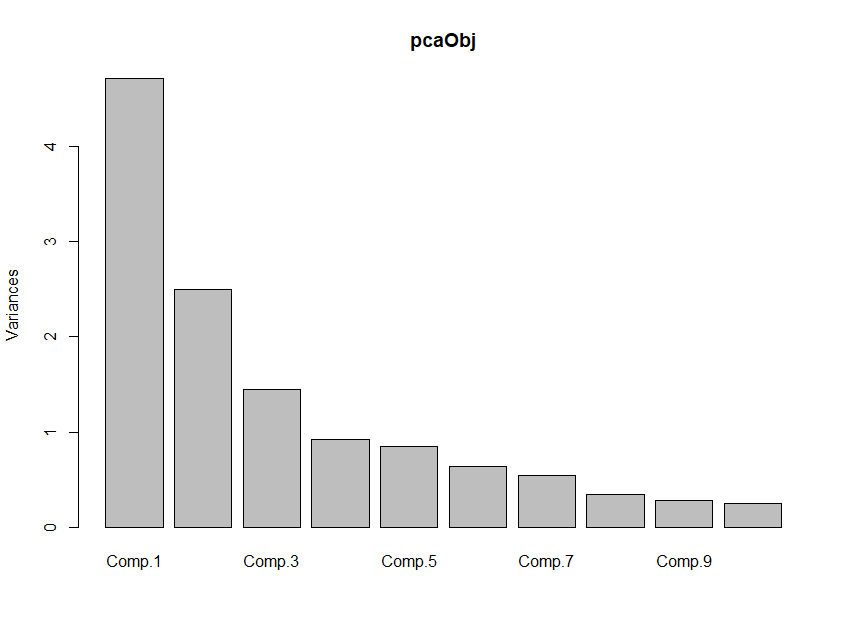


**#showing the weights of the variable using following code**

loadings(pcaObj)

**# graph showing importance of principal components**

plot(pcaObj)



**# graph showing importance of principal components**

**# Comp.1 having highest importance (highest variance)**

**#to check the scores of principal components**

pcaObj$scores

head(pcaObj$scores)

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8

[1,] 3.316751 1.4434626 0.1657390 0.2156312 -0.6930428 0.2238801 -0.59642655 -0.06513909

[2,] 2.209465 -0.3333929 2.0264574 0.2913583 0.2576546 0.9271202 -0.05377561 -1.02441595

[3,] 2.516740 1.0311513 -0.9828187 -0.7249023 0.2510331 -0.5492760 -0.42420545 0.34421613

[4,] 3.757066 2.7563719 0.1761918 -0.5679833 0.3118416 -0.1144310 0.38333730 -0.64359350

[5,] 1.008908 0.8698308 -2.0266882 0.4097658 -0.2984575 0.4065196 -0.44407446 -0.41670047

[6,] 3.050254 2.1224011 0.6293958 0.5156375 0.6320187 -0.1234306 -0.40165376 -0.39489342

Comp.9 Comp.10 Comp.11 Comp.12 Comp.13

[1,] 0.64144271 1.02095585 0.4515634 -0.5408104139 -0.066238631

[2,] -0.30884675 0.15970137 0.1426573 -0.3882377413 0.003636502

[3,] -1.17783447 0.11336086 0.2866728 -0.0005835732 0.021716510

[4,] 0.05254442 0.23941260 -0.7595843 0.2420195635 -0.369483531

[5,] 0.32681916 -0.07836648 0.5259451 0.2166641578 -0.079363566

[6,] -0.15214608 -0.10199582 -0.4055853 0.3794326839 0.145155331

**# Top 3 PCA Scores which represents the whole data**

pcaObj$scores[,1:3] **# Top 3 PCA Scores which represents the whole data**

head(pcaObj$scores[,1:3])

Comp.1 Comp.2 Comp.3

[1,] 3.316751 1.4434626 0.1657390

[2,] 2.209465 -0.3333929 2.0264574

[3,] 2.516740 1.0311513 -0.9828187

[4,] 3.757066 2.7563719 0.1761918

[5,] 1.008908 0.8698308 -2.0266882

[6,] 3.050254 2.1224011 0.6293958

**# cbind used to bind the data in column wise**

**# Considering top 3 principal component scores and binding them with wine data**

wine<-cbind(wine,pcaObj$scores[,1:3])

View(wine)

head(wine)

Type Alcohol Malic Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids

1 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28

2 1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26

3 1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30

4 1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24

5 1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39

6 1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34

Proanthocyanins Color Hue Dilution Proline Comp.1 Comp.2 Comp.3

1 2.29 5.64 1.04 3.92 1065 3.316751 1.4434626 0.1657390

2 1.28 4.38 1.05 3.40 1050 2.209465 -0.3333929 2.0264574

3 2.81 5.68 1.03 3.17 1185 2.516740 1.0311513 -0.9828187

4 2.18 7.80 0.86 3.45 1480 3.757066 2.7563719 0.1761918

5 1.82 4.32 1.04 2.93 735 1.008908 0.8698308 -2.0266882

6 1.97 6.75 1.05 2.85 1450 3.050254 2.1224011 0.6293958

**# preparing data for clustering (considering only pca scores as they represent the entire data)**

colnames(wine)

clus\_data<-wine[,15:17]

clus\_data

head(clus\_data)

Comp.1 Comp.2 Comp.3

1 3.316751 1.4434626 0.1657390

2 2.209465 -0.3333929 2.0264574

3 2.516740 1.0311513 -0.9828187

4 3.757066 2.7563719 0.1761918

5 1.008908 0.8698308 -2.0266882

6 3.050254 2.1224011 0.6293958

**# Normalizing the data**

**# Scale function is used to normalize data.but here we dont need to convert**

**#data into a normal form, because we already scale data using covarience matrix**

#norm\_clus<-scale(clus\_data)

wine\_distance<-dist(clus\_data,method = "euclidean") # method for finding the distance

###########################################################

**# HIERARCHICAL CLUSTERING**

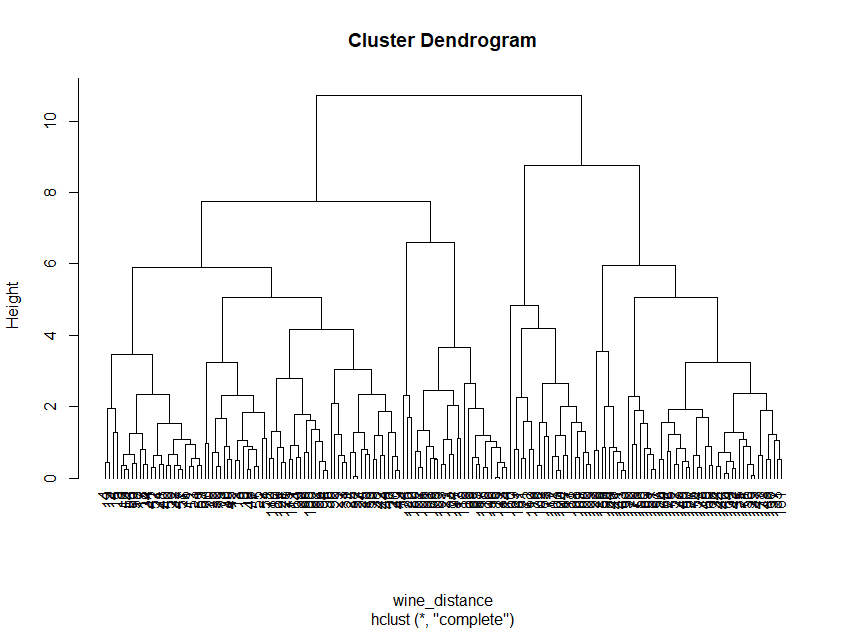
#############################################3

**# Clustering the data using hclust function --> Hierarchical clustering**

**# method here is complete linkage**

fit1<-hclust(wine\_distance,method="complete")

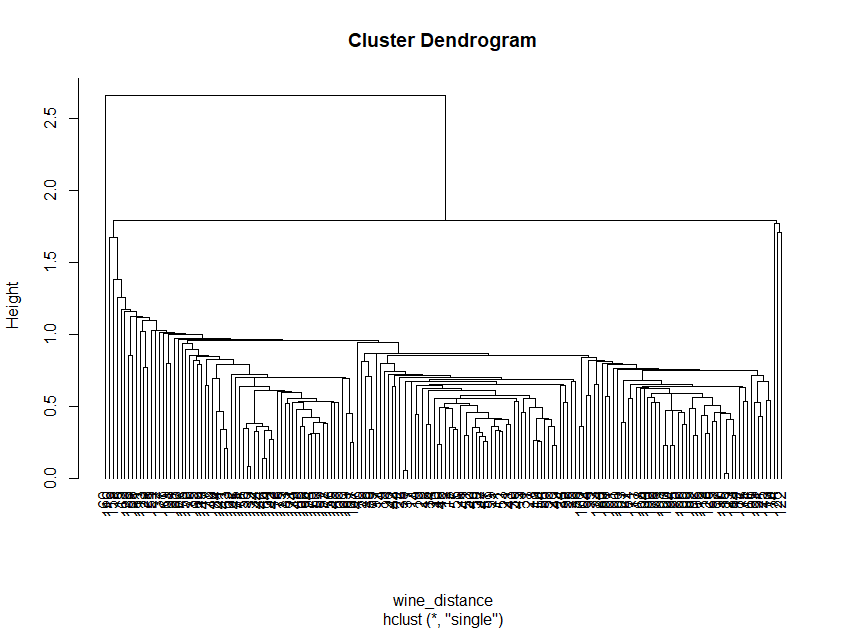
plot(fit1, hang=-1) # Displaying Dendrogram



**# method here is single linkage**

fit2<-hclust(wine\_distance,method="single")

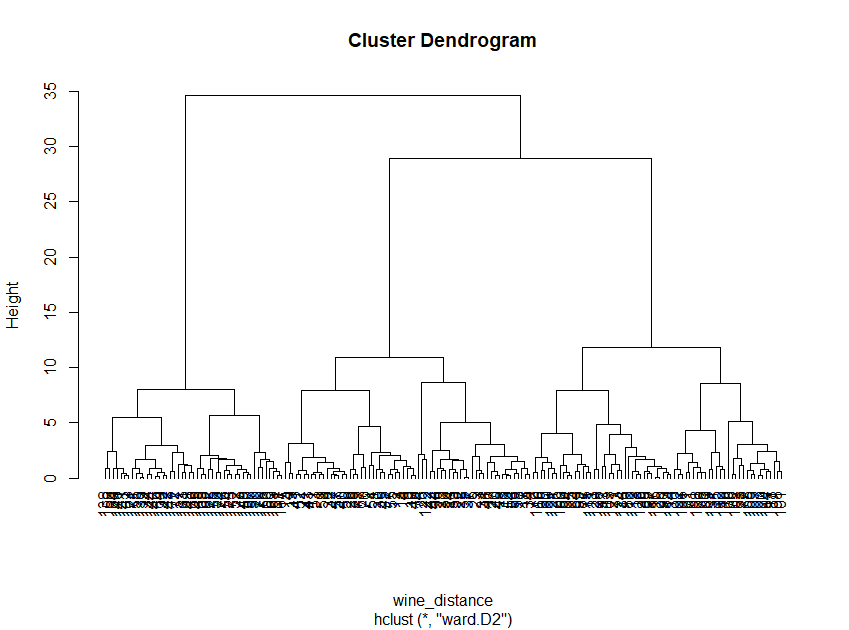
plot(fit2, hang=-1) # Displaying Dendrogram



**# method here is ward.D2 linkage**

fit3<-hclust(wine\_distance,method="ward.D2")

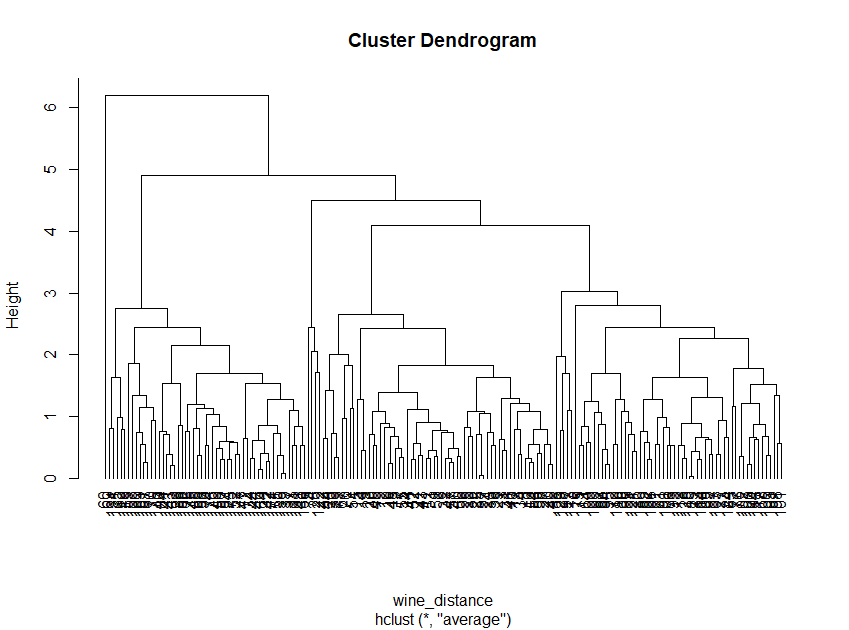
plot(fit3, hang=-1) # Displaying Dendrogram



**# method here is average linkage**

fit4<-hclust(wine\_distance,method="average")

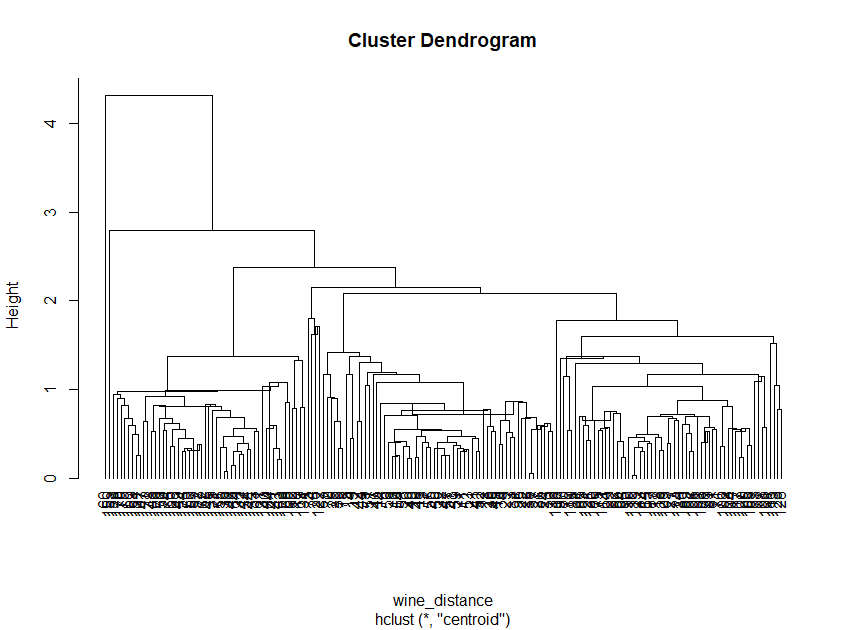
plot(fit4, hang=-1) # Displaying Dendrogram



**# method here is centroid linkage**

fit5<-hclust(wine\_distance,method="centroid")

plot(fit5, hang=-1) # Displaying Dendrogram



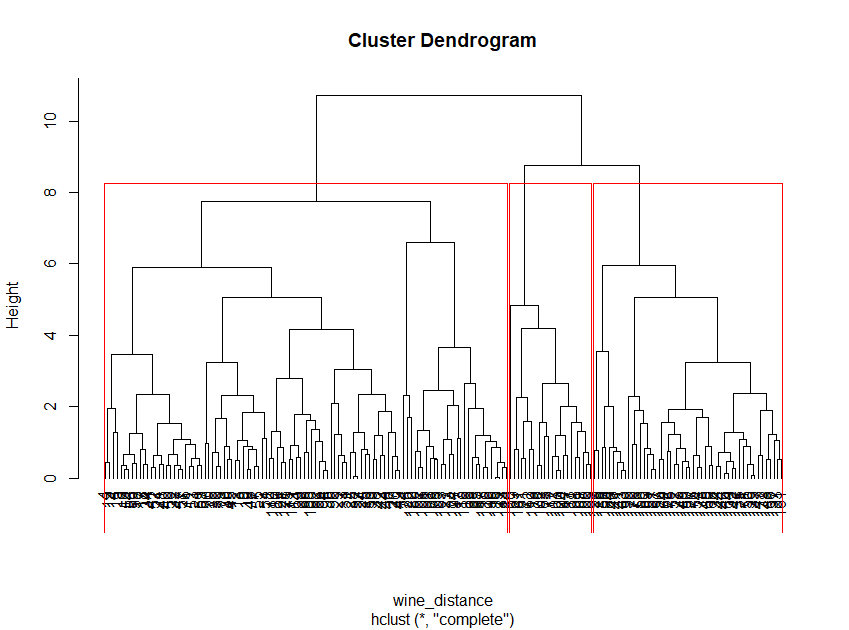
**##using complete linkage we got better dendrogram**

**#next step is to cut the dedrogram**

**#here i am going to cut dendrogram into three cluster**

wine\_groups <- cutree(fit1, k=3)# cut tree into 5 clusters

rect.hclust(fit1, k=3, border="red")



**#convert groups information into a matrix for better understanding**

wine\_cluster<-as.matrix(wine\_groups)

View(wine\_cluster)

head(wine\_cluster)

[,1]

[1,] 1

[2,] 1

[3,] 1

[4,] 1

[5,] 1

[6,] 1

**#create dataframe to combine wine cluster and original data**

final\_data <- data.frame(wine, wine\_cluster)

View(final\_data)

head(final\_data)

Type Alcohol Malic Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids

1 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28

2 1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26

3 1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30

4 1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24

5 1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39

6 1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34

Proanthocyanins Color Hue Dilution Proline Comp.1 Comp.2 Comp.3 wine\_cluster

1 2.29 5.64 1.04 3.92 1065 3.316751 1.4434626 0.1657390 1

2 1.28 4.38 1.05 3.40 1050 2.209465 -0.3333929 2.0264574 1

3 2.81 5.68 1.03 3.17 1185 2.516740 1.0311513 -0.9828187 1

4 2.18 7.80 0.86 3.45 1480 3.757066 2.7563719 0.1761918 1

5 1.82 4.32 1.04 2.93 735 1.008908 0.8698308 -2.0266882 1

6 1.97 6.75 1.05 2.85 1450 3.050254 2.1224011 0.6293958 1

**#here i am going to change the position of the column cluster in to first**

final\_data1 <- final\_data[,c(ncol(final\_data),1:(ncol(final\_data)-1))]

View(final\_data1)

head(final\_data1)

wine\_cluster Type Alcohol Malic Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids

1 1 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28

2 1 1 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26

3 1 1 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30

4 1 1 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24

5 1 1 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39

6 1 1 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34

Proanthocyanins Color Hue Dilution Proline Comp.1 Comp.2 Comp.3

1 2.29 5.64 1.04 3.92 1065 3.316751 1.4434626 0.1657390

2 1.28 4.38 1.05 3.40 1050 2.209465 -0.3333929 2.0264574

3 2.81 5.68 1.03 3.17 1185 2.516740 1.0311513 -0.9828187

4 2.18 7.80 0.86 3.45 1480 3.757066 2.7563719 0.1761918

5 1.82 4.32 1.04 2.93 735 1.008908 0.8698308 -2.0266882

6 1.97 6.75 1.05 2.85 1450 3.050254 2.1224011 0.629395

colnames(final\_data1)

**## Inferences can be drawn from the aggregate of the universities data on wine\_cluster**

View(aggregate(final\_data1[,-c(2,16:18)],by=list(wine\_cluster),FUN=mean))

|  | **Group.1** | | **wine\_cluster** | **Alcohol** | **Malic** | **Ash** | **Alcalinity** | **Magnesium** | **Phenols** | **Flavanoids** | **Nonflavanoids** | **Proanthocyanins** | **Color** | **Hue** | **Dilution** | **Proline** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1** | | 1 | 1 | 13.06500 | 1.993396 | 2.406509 | 18.75849 | 101.99057 | 2.632075 | 2.663019 | 0.3191509 | 1.858491 | 4.438585 | 1.063302 | 3.038113 | 853.7075 |
| **2** | | 2 | 2 | 12.42909 | 1.612727 | 1.984545 | 17.91818 | 91.77273 | 2.021818 | 1.668636 | 0.3495455 | 1.211818 | 3.138636 | 1.044364 | 2.534091 | 522.8636 |
| **3** | | 3 | 3 | 13.11560 | 3.381800 | 2.449800 | 21.75000 | 98.48000 | 1.701000 | 0.844400 | 0.4578000 | 1.190400 | 7.216000 | 0.694800 | 1.741800 | 619.0200 |

Showing 1 to 3 of 3 entries, 15 total columns

**#view a data frame that contain original cluster and predicted cluster**

Type<- wine$Type

View(data.frame(Type,wine\_cluster))

head(data.frame(Type,wine\_cluster))

Type wine\_cluster

1 1 1

2 1 1

3 1 1

4 1 1

5 1 1

6 1 1

**#########################################################################**

**# K-MEANS CLUSTERING**

**########################################################################**

**#lets create clusters using kmeans function**

**#initially we are going to create three clusters, k is the number of clusters**

wine\_fit1 <- kmeans(clus\_data, 3) # 3 cluster solution

str(wine\_fit1)

List of 9

$ cluster : int [1:178] 2 2 2 2 2 2 2 2 2 2 ...

$ centers : num [1:3, 1:3] -2.72 2.2758 -0.0727 1.1257 0.8945 ...

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

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

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

$ totss : num 1540

$ withinss : num [1:3] 117 180 217

$ tot.withinss: num 513

$ betweenss : num 1027

$ size : int [1:3] 51 63 64

$ iter : int 2

$ ifault : int 0

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

**#here we have total withiness between the observation and cluster centre should be very small**

**#and betweenss between the observation and cluster centre should be very large**

**#our objective of k means clustering into keep these values large and small**

**#so we have to try different k values to get higher betweeness and fewer withiness between the clusters**

**#elbow curve & k ~ sqrt(n/2) to decide the k value**

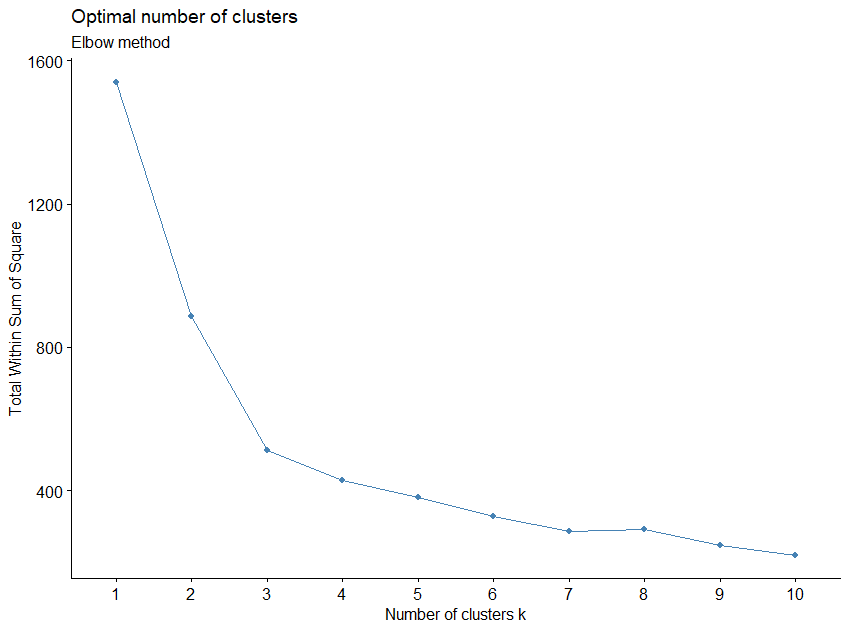
clus\_data

#**use thest clus\_data foe creating Elbow curve**

#install.packages("factoextra")

library(factoextra)

fviz\_nbclust(clus\_data,kmeans,method="wss")+labs(subtitle = "Elbow method")



**#the elbow curve we can see that 10 data points first data pont contain a big slope**

**#similarly data points2,3,4,5,6 also contain some slope.**

**#but the data points 7,8,9,10 contains no slope. this is our cutoff**

**# we can try k=7,8,9,10, which k value will give the result of smaller withiness and higher betweeness**

**#considerd as final fit**

wine\_fit2 <- kmeans(clus\_data, 7) # 7 cluster solution

str(wine\_fit2)

List of 9

$ cluster : int [1:178] 3 1 3 3 5 3 3 3 3 3 ...

$ centers : num [1:7, 1:3] 1.705 -2.616 2.584 -0.293 1.411 ...

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

.. ..$ : chr [1:7] "1" "2" "3" "4" ...

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

$ totss : num 1540

$ withinss : num [1:7] 36.9 24 50.3 111.1 30.9 ...

$ tot.withinss: num 313

$ betweenss : num 1227

$ size : int [1:7] 23 21 41 45 11 13 24

$ iter : int 3

$ ifault : int 0

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

**#tot.withinss: num 313**

**#betweenss : num 1227**

wine\_fit3 <- kmeans(clus\_data, 8) # 7 cluster solution

str(wine\_fit3)

List of 9

$ cluster : int [1:178] 7 7 8 7 8 7 7 7 7 7 ...

$ centers : num [1:8, 1:3] -2.5076 1.3031 -0.0351 -2.9976 -0.7799 ...

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

.. ..$ : chr [1:8] "1" "2" "3" "4" ...

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

$ totss : num 1540

$ withinss : num [1:8] 23.1 41.2 26.4 15.9 47.8 ...

$ tot.withinss: num 271

$ betweenss : num 1268

$ size : int [1:8] 14 24 15 19 29 21 40 16

$ iter : int 4

$ ifault : int 0

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

**#tot.withinss: num 271**

**#betweenss : num 1268**

wine\_fit4 <- kmeans(clus\_data, 9) # 7 cluster solution

str(wine\_fit4)

List of 9

$ cluster : int [1:178] 8 2 8 8 1 8 8 8 8 8 ...

$ centers : num [1:9, 1:3] 1.62 1.6 -2.57 -2.95 -2.8 ...

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

.. ..$ : chr [1:9] "1" "2" "3" "4" ...

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

$ totss : num 1540

$ withinss : num [1:9] 19.5 26.78 19.47 8.32 17.96 ...

$ tot.withinss: num 245

$ betweenss : num 1295

$ size : int [1:9] 7 22 18 9 22 18 17 37 28

$ iter : int 4

$ ifault : int 0

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

**#tot.withinss: num 248**

**#betweenss : num 1291**

wine\_fit5 <- kmeans(clus\_data, 10) # 7 cluster solution

str(wine\_fit5)

List of 9

$ cluster : int [1:178] 4 6 4 4 10 4 4 4 6 4 ...

$ centers : num [1:10, 1:3] -0.874 -1.014 -2.954 2.662 0.467 ...

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

.. ..$ : chr [1:10] "1" "2" "3" "4" ...

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

$ totss : num 1540

$ withinss : num [1:10] 38.48 25.3 8.32 37.33 24.14 ...

$ tot.withinss: num 222

$ betweenss : num 1318

$ size : int [1:10] 26 10 9 35 20 16 22 18 3 19

$ iter : int 4

$ ifault : int 0

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

**#tot.withinss: num 237**

**#betweenss : num 1303**

**##here k=10, we got small withiness and higher betweeness**

**#create final clusters using wine\_fit5$cluster**

wine\_final<- data.frame(wine[-1], wine\_fit5$cluster) # append cluster membership

View(wine\_final)

head(wine\_final)

|  |
| --- |
| Alcohol Malic Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids Proanthocyanins  1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29  2 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28  3 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81  4 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18  5 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82  6 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97  Color Hue Dilution Proline Comp.1 Comp.2 Comp.3 wine\_fit5.cluster  1 5.64 1.04 3.92 1065 3.316751 1.4434626 0.1657390 4  2 4.38 1.05 3.40 1050 2.209465 -0.3333929 2.0264574 6  3 5.68 1.03 3.17 1185 2.516740 1.0311513 -0.9828187 4  4 7.80 0.86 3.45 1480 3.757066 2.7563719 0.1761918 4  5 4.32 1.04 2.93 735 1.008908 0.8698308 -2.0266882 10  6 6.75 1.05 2.85 1450 3.050254 2.1224011 0.6293958 4 |
|  |
| |  | | --- | | > | |

**#find the average values of each clusters**

View(aggregate(wine\_final[,-c(2,16:18)],by=list(wine\_fit5$cluster),FUN=mean))

|  | **Group.1** | **Alcohol** | **Ash** | **Alcalinity** | **Magnesium** | **Phenols** | **Flavanoids** | **Nonflavanoids** | **Proanthocyanins** | **Color** | **Hue** | **Dilution** | **Proline** | **Comp.1** | **Comp.2** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1** | 1 | 12.21692 | 2.425769 | 22.33462 | 91.69231 | 2.023846 | 1.8969231 | 0.4323077 | 1.4784615 | 2.778077 | 1.0207692 | 2.729231 | 474.1923 | -0.8739850 | -1.5658763 |
| **2** | 2 | 12.67000 | 1.966000 | 16.53000 | 94.70000 | 1.842000 | 1.3390000 | 0.3980000 | 0.8850000 | 3.524000 | 1.0596000 | 2.080000 | 565.4000 | -1.0142249 | -1.5801475 |
| **3** | 3 | 12.83667 | 2.611111 | 23.50000 | 98.11111 | 1.775556 | 0.7488889 | 0.5355556 | 1.0322222 | 5.300000 | 0.7900000 | 1.851111 | 592.7778 | -2.9536093 | 0.5767106 |
| **4** | 4 | 13.93857 | 2.482286 | 16.81143 | 109.91429 | 3.000857 | 3.1708571 | 0.2874286 | 2.0011429 | 6.186286 | 1.0565714 | 3.162857 | 1197.7714 | 2.6618683 | 1.4611534 |
| **5** | 5 | 12.13900 | 2.097000 | 19.06000 | 88.00000 | 2.401500 | 2.2085000 | 0.3115000 | 1.6915000 | 2.804500 | 1.1160000 | 3.057500 | 457.4500 | 0.4674015 | -2.3266847 |
| **6** | 6 | 13.33938 | 2.108125 | 15.77500 | 100.81250 | 2.614375 | 2.7106250 | 0.2562500 | 1.9425000 | 4.687500 | 1.0700000 | 3.058750 | 984.6875 | 2.0414103 | -0.2166575 |
| **7** | 7 | 12.99455 | 2.305000 | 20.09091 | 95.90909 | 1.527727 | 0.7122727 | 0.4213636 | 0.9477273 | 6.430909 | 0.7009091 | 1.681818 | 618.8636 | -2.8045341 | 0.6022908 |
| **8** | 8 | 13.50111 | 2.505000 | 22.05556 | 103.27778 | 1.813333 | 0.9272222 | 0.4444444 | 1.4916667 | 9.480000 | 0.6155556 | 1.620000 | 655.5556 | -2.5650181 | 2.2808081 |
| **9** | 9 | 12.53333 | 3.016667 | 27.83333 | 127.33333 | 3.036667 | 3.5500000 | 0.3833333 | 1.9166667 | 4.310000 | 1.1233333 | 3.463333 | 760.0000 | 1.6295427 | 0.4847343 |
| **10** | 10 | 12.82526 | 2.505789 | 19.37368 | 103.63158 | 2.697895 | 2.6436842 | 0.2926316 | 2.0105263 | 3.816316 | 1.0468421 | 3.111579 | 795.1579 | 1.4344108 | -0.2934441 |

Showing 1 to 10 of 10 entries, 15 total columns

**#in our problem given three types of wine**

**#so here i am going to construct three clusters**

**#compare the predicted three clusters into original clusters**

predicted\_clust<-kmeans(clus\_data,3)

str(predicted\_clust)

List of 9

$ cluster : int [1:178] 2 2 2 2 2 2 2 2 2 2 ...

$ centers : num [1:3, 1:3] -2.72 2.2758 -0.0727 1.1257 0.8945 ...

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

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

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

$ totss : num 1540

$ withinss : num [1:3] 117 180 217

$ tot.withinss: num 513

$ betweenss : num 1027

$ size : int [1:3] 51 63 64

$ iter : int 2

$ ifault : int 0

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

**#compare with original cluster**

original\_clust<-wine$Type

View(data.frame(original\_clust,predicted\_clust$cluster))

|  | **original\_clust** | **predicted\_clust.cluster** |
| --- | --- | --- |
|  |  |  |
| **1** | 1 | 2 |
| **2** | 1 | 2 |
| **3** | 1 | 2 |
| **4** | 1 | 2 |
| **5** | 1 | 2 |
| **6** | 1 | 2 |
| **7** | 1 | 2 |
| **8** | 1 | 2 |
| **9** | 1 | 2 |
| **10** | 1 | 2 |
| **11** | 1 | 2 |
| **12** | 1 | 2 |
| **13** | 1 | 2 |
| **14** | 1 | 2 |
| **15** | 1 | 2 |
| **16** | 1 | 2 |
| **17** | 1 | 2 |
| **18** | 1 | 2 |
| **19** | 1 | 2 |
| **20** | 1 | 2 |
| **21** | 1 | 2 |
| **22** | 1 | 2 |
| **23** | 1 | 2 |
| **24** | 1 | 2 |
| **25** | 1 | 2 |
| **26** | 1 | 2 |
| **27** | 1 | 2 |
| **28** | 1 | 2 |
| **29** | 1 | 2 |
| **30** | 1 | 2 |
| **31** | 1 | 2 |
| **32** | 1 | 2 |
| **33** | 1 | 2 |
| **34** | 1 | 2 |
| **35** | 1 | 2 |
| **36** | 1 | 2 |
| **37** | 1 | 2 |
| **38** | 1 | 2 |
| **39** | 1 | 2 |
| **40** | 1 | 2 |
| **41** | 1 | 2 |
| **42** | 1 | 2 |
| **43** | 1 | 2 |
| **44** | 1 | 2 |
| **45** | 1 | 2 |
| **46** | 1 | 2 |
| **47** | 1 | 2 |
| **48** | 1 | 2 |
| **49** | 1 | 2 |
| **50** | 1 | 2 |
| **51** | 1 | 2 |
| **52** | 1 | 2 |
| **53** | 1 | 2 |
| **54** | 1 | 2 |
| **55** | 1 | 2 |
| **56** | 1 | 2 |
| **57** | 1 | 2 |
| **58** | 1 | 2 |
| **59** | 1 | 2 |
| **60** | 2 | 3 |
| **61** | 2 | 3 |
| **62** | 2 | 1 |
| **63** | 2 | 3 |
| **64** | 2 | 3 |
| **65** | 2 | 3 |
| **66** | 2 | 3 |
| **67** | 2 | 3 |
| **68** | 2 | 3 |
| **69** | 2 | 3 |
| **70** | 2 | 3 |
| **71** | 2 | 3 |
| **72** | 2 | 3 |
| **73** | 2 | 3 |
| **74** | 2 | 2 |
| **75** | 2 | 3 |
| **76** | 2 | 3 |
| **77** | 2 | 3 |
| **78** | 2 | 3 |
| **79** | 2 | 3 |
| **80** | 2 | 3 |
| **81** | 2 | 3 |
| **82** | 2 | 3 |
| **83** | 2 | 3 |
| **84** | 2 | 1 |
| **85** | 2 | 3 |
| **86** | 2 | 3 |
| **87** | 2 | 3 |
| **88** | 2 | 3 |
| **89** | 2 | 3 |
| **90** | 2 | 3 |
| **91** | 2 | 3 |
| **92** | 2 | 3 |
| **93** | 2 | 3 |
| **94** | 2 | 3 |
| **95** | 2 | 3 |
| **96** | 2 | 2 |
| **97** | 2 | 3 |
| **98** | 2 | 3 |
| **99** | 2 | 2 |

Showing 1 to 11 of 178 entries, 2 total columns

**#here we can see that original cluster and predicted clustes are different in case of k=3**