***library(cluster)***

***install.packages("fpc")***

***library(fpc)***

***install.packages("NbClust")***

***library(NbClust)***

***library(ggplot2)***

***wine <- read.csv("G:/Data Science/ExcelR/LM Portal/Assignments/PCA/wine.csv",1)***

# cor = TRUE use correlation matrix for getting PCA scores

***help(princomp)*** #to understand the api for princomp

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

***str(pca.wine)***

List of 7

$ sdev : Named num [1:13] 2.169 1.58 1.203 0.959 0.924 ...

..- attr(\*, "names")= chr [1:13] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...

$ loadings: 'loadings' num [1:13, 1:13] 0.14433 -0.24519 -0.00205 -0.23932 0.14199 ...

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

.. ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...

.. ..$ : chr [1:13] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...

$ center : Named num [1:13] 13 2.34 2.37 19.49 99.74 ...

..- attr(\*, "names")= chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...

$ scale : Named num [1:13] 0.81 1.114 0.274 3.33 14.242 ...

..- attr(\*, "names")= chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...

$ n.obs : int 178

$ scores : num [1:178, 1:13] 3.32 2.21 2.52 3.76 1.01 ...

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

.. ..$ : NULL

.. ..$ : chr [1:13] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...

$ call : language princomp(x = wine[, -1], cor = TRUE, scores = TRUE, covmat = NULL)

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

***summary(pca.wine)***

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

#the first 7 variables contribute ~90% of the information required for the entire data.

#hence the 13 components can be reduced to 7 for further analysis with 90% information.

#the standard deviation of the components is stored in a named element called “sdev” of the output variable made by “princomp”:

***pca.wine$sdev***

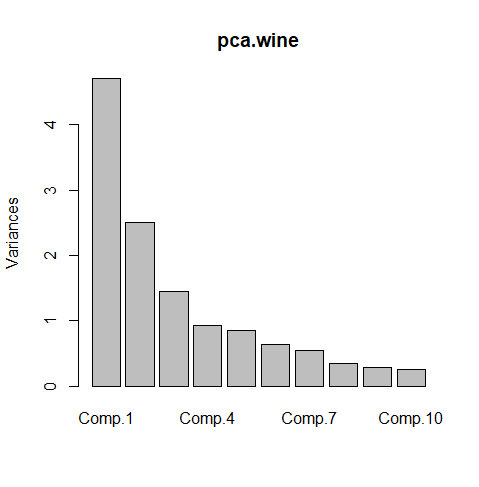
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9

2.1692972 1.5801816 1.2025273 0.9586313 0.9237035 0.8010350 0.7423128 0.5903367 0.5374755

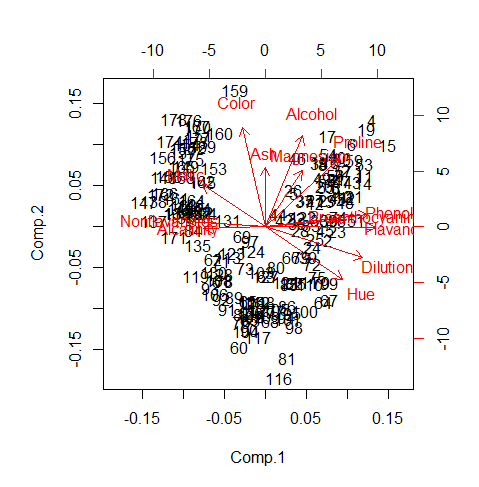
Comp.10 Comp.11 Comp.12 Comp.13

0.5009017 0.4751722 0.4108165 0.3215244

***plot(pca.wine)***

******

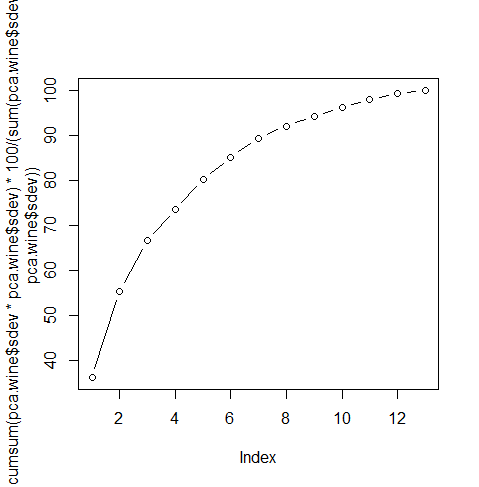
***biplot(pca.wine)***

******

# showing the increase of variance with considering principal components

# which helps in choosing number of principal components

***plot(cumsum(pca.wine$sdev\*pca.wine$sdev)\*100/(sum(pca.wine$sdev\*pca.wine$sdev)),type="b")***

******

#we can also extract the PC loadings

***pca.wine$loadings OR***

***loadings(pca.wine)***

Loadings:

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10

Alcohol 0.144 0.484 0.207 0.266 0.214 0.396 0.509 0.212

Malic -0.245 0.225 -0.537 0.537 -0.421 -0.309

Ash 0.316 -0.626 0.214 0.143 0.154 0.149 -0.170 -0.308

Alcalinity -0.239 -0.612 -0.101 0.287 0.428 0.200

Magnesium 0.142 0.300 -0.131 0.352 -0.727 -0.323 -0.156 0.271

Phenols 0.395 -0.146 -0.198 0.149 -0.406 0.286 -0.320

Flavanoids 0.423 -0.151 -0.152 0.109 -0.187 -0.163

Nonflavanoids -0.299 -0.170 0.203 0.501 -0.259 -0.595 -0.233 0.196 0.216

Proanthocyanins 0.313 -0.149 -0.399 -0.137 -0.534 -0.372 0.368 -0.209 0.134

Color 0.530 0.137 -0.419 0.228 -0.291

Hue 0.297 -0.279 0.428 0.174 0.106 -0.232 0.437 -0.522

Dilution 0.376 -0.164 -0.166 -0.184 0.101 0.266 0.137 0.524

Proline 0.287 0.365 0.127 0.232 0.158 0.120 0.120 -0.576 0.162

Comp.11 Comp.12 Comp.13

Alcohol 0.226 0.266

Malic -0.122

Ash 0.499 -0.141

Alcalinity -0.479

Magnesium

Phenols -0.304 0.304 -0.464

Flavanoids 0.832

Nonflavanoids -0.117 0.114

Proanthocyanins 0.237 -0.117

Color -0.604

Hue -0.259

Dilution -0.601 -0.157

Proline -0.539

Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10

SS loadings 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000

Proportion Var 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077

Cumulative Var 0.077 0.154 0.231 0.308 0.385 0.462 0.538 0.615 0.692 0.769

Comp.11 Comp.12 Comp.13

SS loadings 1.000 1.000 1.000

Proportion Var 0.077 0.077 0.077

Cumulative Var 0.846 0.923 1.000

#Note:

#1.The function above provides the estimate loadings for the data for each component

#2.Note that some values are missing. These values are not zero but are close to zero and should be ignored as we only concern about the largest loadings in the absolute values.

#3.Take the first PC as an example, the three largest loadings are -0.423 for 7, -0.395 for 6, and -0.376 for 12th component.

#4.The first PC makes up 36.19% of the variability in the original data

#We can also extract the PC scores

***winescores <- pca.wine$scores***

#Top 3 PCA Scores which represents the whole data

***winescores[,1:3]***

Comp.1 Comp.2 Comp.3

[1,] 3.31675081 1.44346263 0.165739045

[2,] 2.20946492 -0.33339289 2.026457374

[3,] 2.51674015 1.03115130 -0.982818670

[4,] 3.75706561 2.75637191 0.176191842

[5,] 1.00890849 0.86983082 -2.026688219

[6,] 3.05025392 2.12240111 0.629395827

[7,] 2.44908967 1.17485013 0.977094891

[8,] 2.05943687 1.60896307 -0.146281883

[9,] 2.51087430 0.91807096 1.770969027

[10,] 2.75362819 0.78943767 0.984247490

[11,] 3.47973668 1.30233324 0.422735217

[12,] 1.75475290 0.61197723 1.190878320

[13,] 2.11346234 0.67570634 0.865086426

[14,] 3.45815682 1.13062988 1.204276353

[15,] 4.31278391 2.09597558 1.263912752

[16,] 2.30518820 1.66255173 -0.217902616

[17,] 2.17195527 2.32730534 -0.831729866

[18,] 1.89897118 1.63136888 -0.794913792

[19,] 3.54198508 2.51834367 0.485458508

[20,] 2.08452220 1.06113799 0.164746678

[21,] 3.12440254 0.78689711 0.364887083

[22,] 1.08657007 0.24174355 -0.936961600

[23,] 2.53522408 -0.09184062 0.311932659

[24,] 1.64498834 -0.51627893 -0.143885095

[25,] 1.76157587 -0.31714893 -0.890285647

[26,] 0.99007910 0.94066734 -3.820908008

[27,] 1.77527763 0.68617513 0.086700406

[28,] 1.23542396 -0.08980704 1.386896545

[29,] 2.18840633 0.68956962 -1.394566881

[30,] 2.25610898 0.19146194 1.092657258

[31,] 2.50022003 1.24083383 -1.386017855

[32,] 2.67741105 1.47187365 0.332261728

[33,] 1.62857912 0.05270445 0.167128706

[34,] 1.90269086 1.63306043 -1.172082119

[35,] 1.41038853 0.69793432 -0.479743025

[36,] 1.90382623 0.17671095 -0.450835040

[37,] 1.38486223 0.65863985 -0.458438581

[38,] 1.12220741 0.11410976 0.039107277

[39,] 1.50219450 -0.76943201 1.426177346

[40,] 2.52980109 1.80300198 0.343152389

[41,] 2.58809543 0.77961630 0.118477466

[42,] 0.66848199 0.16996094 0.783362548

[43,] 3.07080699 1.15591896 0.312758084

[44,] 0.46220914 0.33074213 0.201476496

[45,] 2.10135193 -0.07100892 0.655849415

[46,] 1.13616618 1.77710739 -0.028705736

[47,] 2.72660096 1.19133469 0.539773261

[48,] 2.82133927 0.64625860 1.155552411

[49,] 2.00985085 1.24702946 0.057293988

[50,] 2.70749130 1.75196741 0.643113612

[51,] 3.21491747 0.16699199 1.973571680

[52,] 2.85895983 0.74527880 -0.004719502

[53,] 3.50560436 1.61273386 0.520774530

[54,] 2.22479138 1.87516800 -0.339549850

[55,] 2.14698782 1.01675154 0.957762762

[56,] 2.46932948 1.32900831 -0.513437453

[57,] 2.74151791 1.43654878 0.612473396

[58,] 2.17374092 1.21219984 -0.261779593

[59,] 3.13938015 1.73157912 0.285661413

[60,] -0.92858197 -3.07348616 4.585064007

[61,] -1.54248014 -1.38144351 0.874683112

[62,] -1.83624976 -0.82998412 1.605702186

[63,] 0.03060683 -1.26278614 1.784408010

[64,] 2.05026161 -1.92503260 0.007368777

[65,] -0.60968083 -1.90805881 -0.679357938

[66,] 0.90022784 -0.76391147 -0.573361302

[67,] 2.24850719 -1.88459248 2.031840193

[68,] 0.18338403 -2.42714611 1.069745560

[69,] -0.81280503 -0.22051399 0.707005396

[70,] 1.97562050 -1.40328323 1.238276220

[71,] -1.57221622 -0.88498314 0.628997950

[72,] 1.65768181 -0.95671220 -1.952584217

[73,] -0.72537239 -1.06364540 -0.080332229

[74,] 2.56222717 0.26019855 -3.374393962

[75,] 1.83256757 -1.28787820 -0.458280027

[76,] -0.86799290 -2.44410119 1.563333179

[77,] 0.37001440 -2.15390698 2.449386348

[78,] -1.45737704 -1.38335177 0.227306902

[79,] 1.26293085 -0.77084953 1.184224517

[80,] 0.37615037 -1.02704340 -1.794466295

[81,] 0.76206390 -3.37505381 0.357470056

[82,] 1.03457797 -1.45070974 0.363011773

[83,] -0.49487676 -2.38124353 -1.335743176

[84,] -2.53897708 -0.08744336 -0.474251393

[85,] 0.83532015 -1.47367055 -0.610093576

[86,] 0.78790461 -2.02662652 0.254723404

[87,] -0.80683216 -2.23383039 -0.772855797

[88,] -0.55804262 -2.37298543 -2.307611404

[89,] -1.11511104 -1.80224719 -0.959253308

[90,] -0.55572283 -2.65754004 -0.849126898

[91,] -1.34928528 -2.11800147 0.047652321

[92,] -1.56448261 -1.85221452 -0.781067031

[93,] -1.93255561 -1.55949546 0.089274676

[94,] 0.74666594 -2.31293171 -0.114679769

[95,] 0.95745536 -2.22352843 -0.142444774

[96,] 2.54386518 0.16927402 -0.788696991

[97,] -0.54395259 -0.36892655 -1.308895932

[98,] 1.03104975 -2.56556935 1.086390174

[99,] 2.25190942 -1.43274138 0.230208244

[100,] 1.41021602 -2.16619177 -0.748896411

[101,] 0.79771979 -2.37694880 1.568112531

[102,] -0.54953173 -2.29312864 1.498935323

[103,] -0.16117374 -1.16448332 -1.003713103

[104,] -0.65979494 -2.67996119 0.764920868

[105,] 0.39235441 -2.09873171 0.471850008

[106,] -1.77249908 -1.71728847 -0.947033174

[107,] -0.36626736 -2.16935330 0.481324235

[108,] -1.62067257 -1.35558339 -0.287159001

[109,] 0.08253578 -2.30623459 0.463574989

[110,] 1.57827507 -1.46203429 -1.779645955

[111,] 1.42056925 -1.41820664 -0.139275829

[112,] -0.27870275 -1.93056809 -0.078670553

[113,] -1.30314497 -0.76317231 -1.999596510

[114,] -0.45707187 -2.26941561 -1.061338968

[115,] -0.49418585 -1.93904505 -1.323938072

[116,] 0.48207441 -3.87178385 -1.344271223

[117,] -0.25288888 -2.82149237 0.302639785

[118,] -0.10722764 -1.92892204 -0.690148243

[119,] -2.43301260 -1.25714104 1.903027404

[120,] -0.55108954 -2.22216155 0.356228830

[121,] 0.73962193 -1.40895667 -1.125345492

[122,] 1.33632173 0.25333693 -5.345388179

[123,] -1.17708700 -0.66396684 -3.010221888

[124,] -0.46233501 -0.61828818 -0.483442366

[125,] 0.97847408 -1.44557050 -1.481236975

[126,] -0.09680973 -2.10999799 -0.434826116

[127,] 0.03848715 -1.26676211 -0.687577913

[128,] -1.59715850 -1.20814357 -3.361175555

[129,] -0.47956492 -1.93884066 -1.296507519

[130,] -1.79283347 -1.15028810 -0.782800173

[131,] -1.32710166 0.17038923 1.180013355

[132,] -2.38450083 0.37458261 0.723822595

[133,] -2.93694010 0.26386183 0.167639816

[134,] -2.14681113 0.36825495 0.453301301

[135,] -2.36986949 -0.45963481 1.101399789

[136,] -3.06384157 0.35341284 1.099124104

[137,] -3.91575378 0.15458252 -0.221827800

[138,] -3.93646339 0.65968723 -1.712215419

[139,] -3.09427612 0.34884276 1.026831413

[140,] -2.37447163 0.29198035 -1.241914333

[141,] -2.77881295 0.28680487 -0.609670124

[142,] -2.28656128 0.37250784 0.971643032

[143,] -2.98563349 0.48921791 -0.946952932

[144,] -2.37519470 0.48233372 0.252883994

[145,] -2.20986553 1.16005250 1.245125226

[146,] -2.62562100 0.56316076 0.855961082

[147,] -4.28063878 0.64967096 1.458196962

[148,] -3.58264137 1.27270275 0.110784038

[149,] -2.80706372 1.57053379 0.472527935

[150,] -2.89965933 2.04105701 0.495959810

[151,] -2.32073698 2.35636608 -0.437681744

[152,] -2.54983095 2.04528309 0.312267999

[153,] -1.81254128 1.52764595 -1.362589782

[154,] -2.76014464 2.13893235 0.964628688

[155,] -2.73715050 0.40988627 1.190404684

[156,] -3.60486887 1.80238422 0.094036861

[157,] -2.88982600 1.92521861 0.782322556

[158,] -3.39215608 1.31187639 -1.602025969

[159,] -1.04818190 3.51508969 -1.160038566

[160,] -1.60991228 2.40663816 -0.548559697

[161,] -3.14313097 0.73816104 0.090998724

[162,] -2.24015690 1.17546529 0.101376932

[163,] -2.84767378 0.55604397 -0.804215218

[164,] -2.59749706 0.69796554 0.884939521

[165,] -2.94929937 1.55530896 0.983400727

[166,] -3.53003227 0.88252680 0.466029128

[167,] -2.40611054 2.59235618 -0.428226211

[168,] -2.92908473 1.27444695 1.213358272

[169,] -2.18141278 2.07753731 -0.763782552

[170,] -2.38092779 2.58866743 -1.418044029

[171,] -3.21161722 -0.25124910 0.847129152

[172,] -3.67791872 0.84774784 1.339420231

[173,] -2.46555580 2.19379830 0.918780960

[174,] -3.37052415 2.21628914 0.342569512

[175,] -2.60195585 1.75722935 -0.207581355

[176,] -2.67783946 2.76089913 0.940941877

[177,] -2.38701709 2.29734668 0.550696197

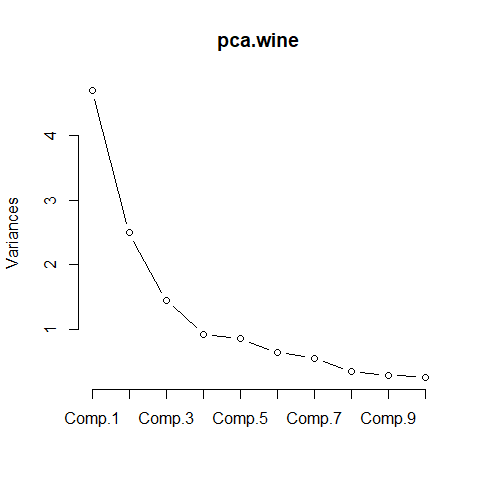
[178,] -3.20875816 2.76891957 -1.013913664

#Determine how many principal components to retain

#screeplot

***screeplot(pca.wine, main = "screeplot")***

***screeplot(pca.wine, type = "lines")***

******

#Cluster Analysis (considering only pca scores as they represent the entire data)

***clust.wine <- NbClust(wine[], distance = "euclidean", min.nc = 2, max.nc = 10, method = "complete", index = "all")***

The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a

significant increase of the value of the measure i.e the significant peak in Hubert

index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex

second differences plot) that corresponds to a significant increase of the value of

the measure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Among all indices:

\* 6 proposed 2 as the best number of clusters

\* 5 proposed 3 as the best number of clusters

\* 1 proposed 4 as the best number of clusters

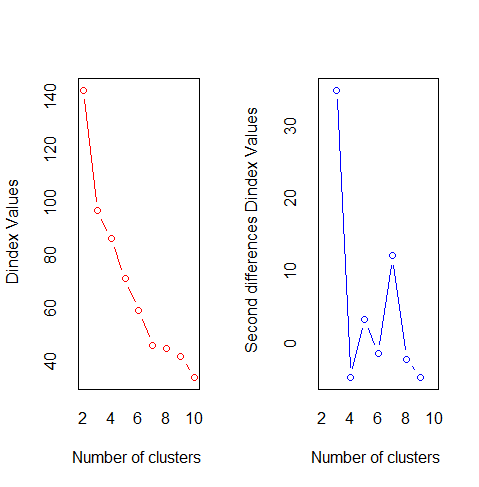
\* 7 proposed 7 as the best number of clusters

\* 1 proposed 8 as the best number of clusters

\* 4 proposed 10 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 7



#plot bar chart for the clusters

***install.packages("factoextra")***

***library(factoextra)***

***fviz\_nbclust(clust.wine)+theme\_minimal()***

Among all indices:

===================

\* 2 proposed 0 as the best number of clusters

\* 6 proposed 2 as the best number of clusters

\* 5 proposed 3 as the best number of clusters

\* 1 proposed 4 as the best number of clusters

\* 7 proposed 7 as the best number of clusters

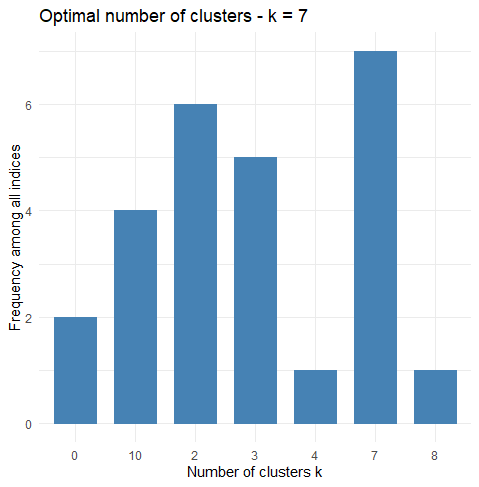
\* 1 proposed 8 as the best number of clusters

\* 4 proposed 10 as the best number of clusters

Conclusion

=========================

\* According to the majority rule, the best number of clusters is 7.

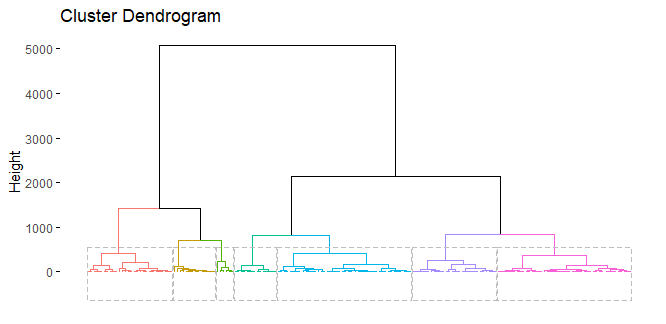
******

***dev.off()***

#Hierarchical clustering - All Variables

***hclust.complete = eclust(wine, "hclust", k=7, method = "complete", graph = FALSE)***

***fviz\_dend(hclust.complete, rect = TRUE, show\_labels = FALSE)***

******

#K-means clustering - All Variables

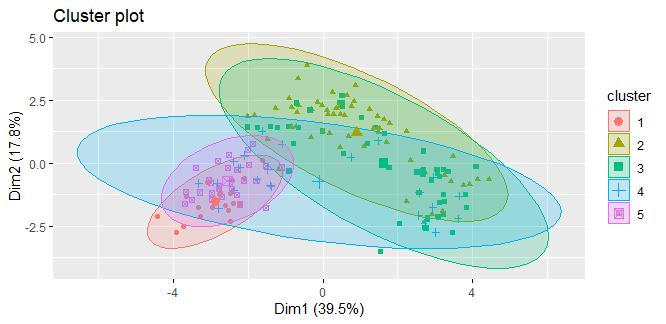
***kmeans.wine = eclust(wine, "kmeans", k=5, nstart = 25, graph = FALSE)***

***fviz\_cluster(kmeans.wine, geom = "point", frame.type = "norm")***

Warning messages:

1: argument frame is deprecated; please use ellipse instead.

2: argument frame.type is deprecated; please use ellipse.type instead.

******

#Cluster Analysis- with PCA components

#Number of clusters suggested by the NbClust function are **7**

***wine.pca = wine[,2:14]***

***no\_clust = NbClust(wine.pca, distance = "euclidean", min.nc = 2, max.nc = 10, method = "complete", index = "all")***

\*\*\* : The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a

significant increase of the value of the measure i.e the significant peak in Hubert

index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dinde

second differences plot) that corresponds to a significant increase of the value of the measure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Among all indices:

\* 6 proposed 2 as the best number of clusters

\* 5 proposed 3 as the best number of clusters

\* 1 proposed 4 as the best number of clusters

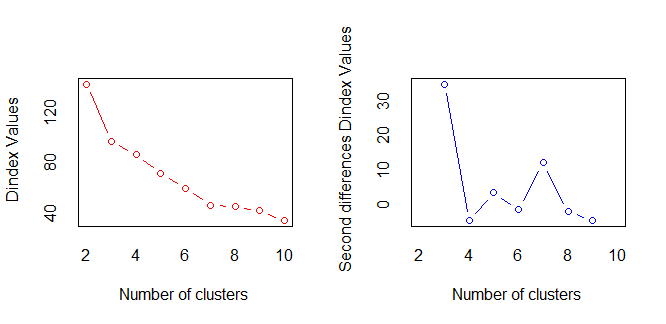
\* 7 proposed 7 as the best number of clusters

\* 1 proposed 8 as the best number of clusters

\* 4 proposed 10 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 7



#Plot barchart for the clusters

***fviz\_nbclust(no\_clust)+theme\_minimal()***

Among all indices:

===================

\* 2 proposed 0 as the best number of clusters

\* 6 proposed 2 as the best number of clusters

\* 5 proposed 3 as the best number of clusters

\* 1 proposed 4 as the best number of clusters

\* 7 proposed 7 as the best number of clusters

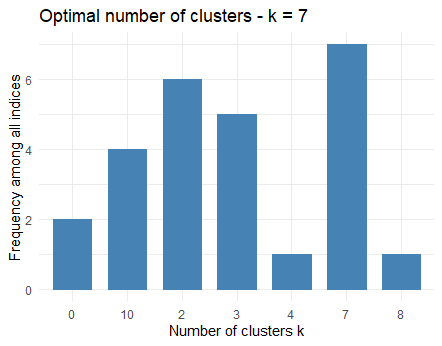
\* 1 proposed 8 as the best number of clusters

\* 4 proposed 10 as the best number of clusters

Conclusion

=========================

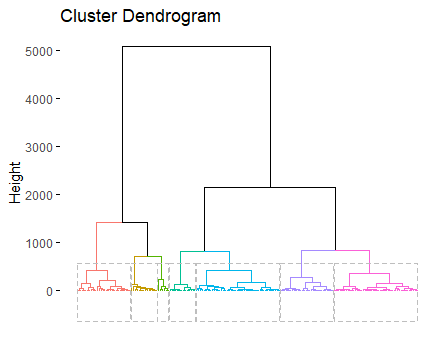
\* According to the majority rule, the best number of clusters is 7 .

******

# H clust using PCA components

***hclust.complete = eclust(wine.pca, "hclust", k=7, method = "complete", graph = FALSE)***

***fviz\_dend(hclust.complete, rect = TRUE, show\_labels = FALSE)***



#K-means clustering - All Variables

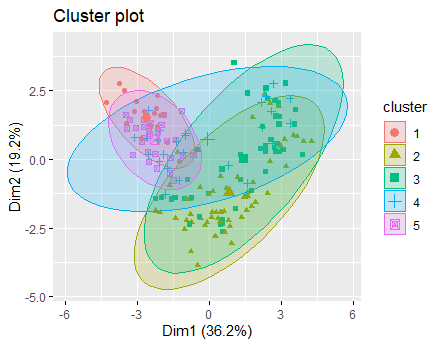
***kmeans.wine = eclust(wine.pca, "kmeans", k=3, nstart = 25, graph = FALSE)***

***fviz\_cluster(kmeans.wine, geom = "point", frame.type = "norm")***

Warning messages:

1: argument frame is deprecated; please use ellipse instead.

2: argument frame.type is deprecated; please use ellipse.type instead.



#Observation(s):

#When PCA was applied on the entire set of varibles (13);

#PCA suggested that 90% of the information can be inferred from the first 7 variables.

#We then plotted dendrogram for both 13 variables and 7 variables data and found that the number of clustered required are 7 and the dendrogram seem identical.