```
library(tidyverse)
> library(factoextra)
> library(ISLR)
> library(stats)
> library(HAC)
> library(flexclust)
> library(dplyr)
> library(stats)
> Universities1 <- read.csv("C:/Users/Ghirghir/Desktop/Mashine Learning/Assignment/Unive
rsities.csv")
>
> set.seed(123)
>
>
> # Remove all NA from the table and summarize with this method:
>
> Universities <- Universities1[complete.cases(Universities1), ]
> U <- Universities[,c(-1,-2,-3)]  # Excluding categorical measurements
>
> # Scaling the data frame (z-score)
>
> df <- scale(U)
> head(U)
> distance <- get_dist(df)
> fviz_dist(distance) # plot
>
>
> # K-Means algorithm to find out the clusters
>
> k4 <- kmeans(df, centers = 4, nstart = 25) # k = 4, number of restarts = 25
>


k4$centers # output the centers
  X..appli..rec.d X..appl..accepted X..new.stud..enrolled X..new.stud..from.top.10.
1       1.9817966          2.2299227           2.444722e+00                  0.1334215
2      -0.3692895         -0.3314846          -3.967692e-01                  0.0102519
3      -0.3033156         -0.2989118          -2.276979e-01                 -0.6785172
4       0.4402622          0.1551461          -2.000371e-05                  1.6526422
  X..new.stud..from.top.25. X..FT.undergrad X..PT.undergrad in.state.tuition
1                 0.2545856       2.5228452      1.74868491       -1.0500277
2                 0.1080080      -0.4049392     -0.25785122        0.4057712
3                -0.7279285      -0.1972688     -0.04353747       -0.7234450
4                 1.4315089      -0.1108205     -0.38259215        1.5022093
  out.of.state.tuition        room      board   add..fees estim..book.costs estim..perso
nal..
1           -0.4918168 -0.03883300 -0.1745795  0.49531762       0.163585669         0.93
85863
2            0.2956208  0.08357902  0.3292398 -0.18996619      -0.158302104        -0.29
78018
3           -0.8237908 -0.53385193 -0.6791344  0.03928218       0.003218005         0.25
31393
4            1.6819156  1.19276784  0.9944521  0.07619136       0.311659604        -0.49
21884
  X..fac..w.PHD stud..fac..ratio Graduation.rate
1     0.6840794        0.6139980      -0.2538234
2     0.0835866       -0.1828501       0.3971948
3    -0.6684106        0.4582141      -0.7769793
4     1.0478784       -1.1189523       1.1188151
>
> k4$size # Number of universities in each cluster
[1]  46 183 175  67
>
>
```

```
> k4$cluster[120] # Identify the cluster of the 20th observation as an example
377
  3
>
>
> fviz_cluster(k4, data = df) # Visualize the output(plot)
>
>
> # Elbow Method
> fviz_nbclust(df, kmeans, method = "wss")


# Compute k-means clustering with k = 4
> set.seed(123)
> k3 <- kmeans(df, centers =3, nstart = 25)
> print(k3)
K-means clustering with 3 clusters of sizes 46, 150, 275

Cluster means:
  X..appli..rec.d X..appl..accepted X..new.stud..enrolled X..new.stud..from.top.10.
1      1.98179657        2.22992267             2.4447222                 0.1334215
2      0.05140256       -0.04367128            -0.1683551                 0.8795798
3     -0.35953828       -0.34918455            -0.3171053                -0.5020886
  X..new.stud..from.top.25. X..FT.undergrad X..PT.undergrad in.state.tuition
1                 0.2545856       2.5228452       1.7486849       -1.0500277
2                 0.8620961      -0.2324464      -0.3130216        1.0620416
3                -0.5128195      -0.2952142      -0.1217682       -0.4036544
  out.of.state.tuition       room      board   add..fees estim..book.costs estim..person
al..
1           -0.4918168 -0.0388330 -0.1745795  0.49531762        0.16358567        0.9385
8632
2            1.1158839  0.6698444  0.7756859 -0.04496556        0.07122705       -0.3966
5857
3           -0.5263964 -0.3588740 -0.3938990 -0.05832646       -0.06621454        0.0593
5933
  X..fac..w.PHD stud..fac..ratio Graduation.rate
1     0.6840794        0.6139980      -0.2538234
2     0.7659627       -0.7036167       0.8426062
3    -0.5322257        0.2810858      -0.4171456




Clustering vector:
   1    3   10   12   22   26   32   38   39   46   49   50   63   77   78   79   81   9
0   92
   3    3    2    3    3    3    3    3    3    3    1    1    1    2    2    2    2
2    3
  95   96   97  108  110  112  120  121  122  123  126  127  130  134  139  140  146  14
8  149
   2    3    2    2    1    2    2    3    3    2    3    3    3    3    3    2    3
2    2
 151  152  153  154  156  158  160  161  164  168  169  172  174  176  181  186  188  19
0  194
   2    2    2    3    1    2    2    2    2    1    3    3    2    3    3    2    2
2    1
 208  210  220  228  230  235  236  239  244  245  246  247  248  250  251  252  258  25
9  260
   3    2    3    2    1    3    3    3    3    3    3    2    3    3    3    2    3
3    3
 262  263  264  265  268  269  270  272  275  277  287  294  297  298  302  304  312  31
7  319
   3    3    3    3    3    3    3    3    3    2    3    2    2    2    3    2    3
3    3
```

```
 320  321  322  323  324  326  329  330  331  332  333  336  340  349  352  359  360  36
2  365
   3    2    1    1    3    3    2    2    2    3    3    3    3    3    3    3    2
2    2
 366  368  369  370  376  377  382  387  390  395  398  400  403  405  412  415  418  42
1  428
   2    3    3    3    3    3    3    3    3    2    3    3    3    2    3    3    1
2    2
 433  436  438  441  442  443  444  451  452  454  456  457  460  462  463  464  467  47
7  479
   2    2    2    1    3    2    2    3    2    3    2    3    3    3    3    2    1
1    2
 481  483  484  494  500  505  511  513  514  515  520  522  525  526  527  528  537  53
8  541
   2    2    2    2    3    1    3    2    2    3    3    3    3    2    2    3    2
2    3
 543  544  551  552  556  558  563  564  565  567  571  572  577  578  579  582  584  58
7  589
   1    3    3    3    1    3    2    3    3    2    2    2    3    3    3    3    1
3    3
 591  595  598  599  605  606  607  608  614  615  616  617  618  626  627  629  630  63
3  638
   3    3    3    3    3    3    1    2    2    3    3    3    3    3    3    3    3
3    3
 642  646  647  651  654  655  656  658  659  660  662  666  667  670  673  674  676  67
7  681
   3    1    3    3    2    2    1    3    3    3    3    3    3    3    3    3    3
2    1
 682  684  687  688  689  691  693  696  697  698  702  703  704  705  710  712  713  71
6  717
   1    3    3    3    3    3    3    3    3    3    2    3    3    3    3    1    3
2    3
 720  723  724  726  732  733  736  737  738  739  742  744  745  746  750  751  754  75
6  757
   3    1    3    3    2    3    3    3    3    3    2    3    2    3    3    1    3
3    3
 769  771  772  777  778  782  783  789  792  793  794  795  801  803  804  814  815  82
3  824
   2    3    3    2    2    2    2    3    2    2    2    3    2    2    3    3    2
2    2
 825  826  828  831  833  834  836  837  838  839  840  841  843  844  845  846  848  85
1  860
   2    3    3    2    1    1    3    3    3    3    3    3    3    3    3    2    2
2    3
 869  870  872  874  875  878  879  882  885  889  891  892  893  894  896  898  901  90
3  904
   3    3    1    2    3    3    3    2    3    2    3    2    3    3    2    1    3
1    2
 907  911  912  916  917  928  929  931  932  933  939  943  945  950  952  955  958  95
9  963
   3    2    3    1    3    3    3    3    1    3    3    3    2    2    2    2    2
2    2
 965  967  969  971  974  975  977  978  979  986  987  988  989  991  992  994  996  99
7 1001
   2    2    2    2    3    2    3    3    2    2    3    2    2    2    3    3    2
2    3
1009 1010 1014 1017 1020 1021 1023 1024 1025 1026 1027 1029 1030 1031 1032 1033 1035 103
6 1037
   3    3    3    3    3    2    1    3    2    1    2    2    2    3    3    3    3
3    3
1039 1041 1043 1047 1048 1051 1052 1053 1055 1059 1060 1061 1064 1065 1075 1079 1081 108
4 1087
   1    2    2    1    3    3    3    3    3    3    2    3    3    2    3    3    3
3    3
```

```
1089 1090 1095 1096 1098 1101 1102 1105 1107 1110 1111 1115 1117 1118 1121 1125 1127 113
1 1132
   3    3    3    3    3    3    3    3    2    3    3    1    2    2    3    3    3
3    3
1138 1139 1143 1146 1152 1154 1156 1158 1163 1164 1166 1168 1172 1176 1177 1181 1185 118
8 1189
   1    3    3    3    3    3    3    3    2    2    3    1    1    3    1    1    3
3    2
1192 1194 1195 1196 1198 1204 1206 1212 1214 1218 1221 1222 1223 1227 1231 1232 1236 123
7 1238
   3    2    3    2    3    3    1    3    1    3    3    2    1    3    3    3    3
2    3
1239 1246 1253 1257 1258 1262 1268 1269 1273 1274 1275 1284 1285 1292 1302
   2    2    2    2    3    2    2    2    3    3    3    3    3    3    3

Within cluster sum of squares by cluster:
[1] 1044.680 1424.892 2562.342
 (between_SS / total_SS =  37.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweens
s"
[7] "size"         "iter"         "ifault"
```

```
fviz_cluster(k3, data = df) # Cluster Plot
>
> # Silhouette Method
>
>
> fviz_nbclust(df, kmeans, method = "silhouette")
>
>
> #Creating the cluster index for 3 clusters
> set.seed(123)
> k3_median = kcca(U, k=3, kccaFamily("kmedians"))
> Clusters_3 <- predict(k3_median)
> Un_scale <- scale(U, center=T, scale= T)
> U.dis <- dist(Un_scale)
> cah.ward <- hclust(U.dis, method = "ward.D2")
> plot(cah.ward)
>
> # partition in 3 groups
> groupes.cah <- cutree(cah.ward,k=3)
> #Function for calculating summary statistics -y cluster membership variable
> stat.comp <-function(x,y){ #number of clusters
+ K <- length(unique(y))
+ #nb. Of instances
+ n <- length(x)
+ #overall mean
+ m <- mean(x)
+ #total sum of squares
+ TSS <- sum((x-m)^2)
+ #size of clusters
+ nk <- table(y)
+ #conditional mean
+ mk <- tapply(x,y,mean)
+ #between (explained) sum of squares
+ BSS <- sum(nk * (mk-m)^2)
+ #collect in a vector the means and the proportion of variance explained
+ result <- c(mk,100.0*BSS/TSS)
```

```
+ #set a name to the values
+ names(result) <- c(paste("G",1:K),"% epl.")
+ #return the results
+ return(result)
+ }


> print(sapply(U,stat.comp ,y=groupes.cah))
       X..appli..rec.d X..appl..accepted X..new.stud..enrolled X..new.stud..from.top.10.
G 1          1751.22287         1262.59531              505.83578                   20.98534
G 2          4305.92500         2202.68750              691.96250                   56.05000
G 3         10814.78000         7297.84000             2797.30000                   31.08000
% epl.         47.59883           53.97229               58.30018                   49.96857
       X..new.stud..from.top.25. X..FT.undergrad X..PT.undergrad in.state.tuition
G 1                    48.28152       2222.73314        560.75953        8256.88270
G 2                    83.03750       2700.85000        208.90000       17339.78750
G 3                    62.10000      14082.48000       3353.40000        4554.90000
% epl.                 41.51523         60.55538         33.25373          46.57664
       out.of.state.tuition       room      board add..fees estim..book.costs estim..per
sonal..
G 1              9239.48974 2049.13490 1998.38416 360.12317        532.947214        131
5.45748
G 2             17339.78750 2909.68750 2637.85000 357.62500        587.762500         97
2.52500
G 3              8861.04000 2292.24000 2139.14000 542.14000        594.420000        183
1.02000
% epl.             50.54138   20.19134   17.55565   2.50144          2.486459          1
0.38515
       X..fac..w.PHD stud..fac..ratio Graduation.rate
G 1          67.54839          14.5871         61.63343
G 2          90.12500          10.1925         84.50000
G 3          84.74000          15.7360         62.06000
% epl.       30.99962          19.9780         22.33416
> print(sapply(U,stat.comp ,y=groupes.cah))
       X..appli..rec.d X..appl..accepted X..new.stud..enrolled X..new.stud..from.top.10.
G 1          1751.22287         1262.59531              505.83578                   20.98534
G 2          4305.92500         2202.68750              691.96250                   56.05000
G 3         10814.78000         7297.84000             2797.30000                   31.08000
% epl.         47.59883           53.97229               58.30018                   49.96857
       X..new.stud..from.top.25. X..FT.undergrad X..PT.undergrad in.state.tuition
G 1                    48.28152       2222.73314        560.75953        8256.88270
G 2                    83.03750       2700.85000        208.90000       17339.78750
G 3                    62.10000      14082.48000       3353.40000        4554.90000
% epl.                 41.51523         60.55538         33.25373          46.57664
       out.of.state.tuition       room      board add..fees estim..book.costs estim..per
sonal..
G 1              9239.48974 2049.13490 1998.38416 360.12317        532.947214        131
5.45748
G 2             17339.78750 2909.68750 2637.85000 357.62500        587.762500         97
2.52500
G 3              8861.04000 2292.24000 2139.14000 542.14000        594.420000        183
1.02000
% epl.             50.54138   20.19134   17.55565   2.50144          2.486459          1
0.38515
       X..fac..w.PHD stud..fac..ratio Graduation.rate
G 1          67.54839          14.5871         61.63343
G 2          90.12500          10.1925         84.50000
G 3          84.74000          15.7360         62.06000
% epl.       30.99962          19.9780         22.33416



#Merging the clusters to the original data frame
```

```
> set.seed(123)
> clusters <- data.frame(Clusters_3)
> Universities <- cbind(Universities, clusters)
> Universities$room_board_fees <- Universities$room + Universities$board + Universities$
add..fees + Universities$estim..book.costs + Universities$estim..personal..
> #All
> set.seed(123)
> Summary_cont <- Universities %>%
+     group_by(Clusters_3) %>%
+     summarise( Acceptance_rate = sum(X..appl..accepted)/ sum(X..appli..rec.d), Avg_out
_state_tuition=mean(out.of.state.tuition), Avg_int_state_tuition=mean(in.state.tuition),
room_board_fees=mean(room_board_fees), mean_PHD_fac=mean(X..fac..w.PHD), mean_stud_fac_r
atio=mean(stud..fac..ratio), mean_grad_rate=mean(Graduation.rate), priv_count = sum(Publ
ic..1...Private..2. == 2), pub_count = sum(Public..1...Private..2. == 1))
> Summary_cont
> #Private
> Summary_cont_priv <- Universities %>% filter(Public..1...Private..2. == 2) %>%
+     group_by(Clusters_3) %>%
+     summarise( Acceptance_rate = sum(X..appl..accepted)/ sum(X..appli..rec.d), Avg_out
_state_tuition=mean(out.of.state.tuition), Avg_int_state_tuition=mean(in.state.tuition),
room_board_fees=mean(room_board_fees), mean_PHD_fac=mean(X..fac..w.PHD), mean_stud_fac_r
atio=mean(stud..fac..ratio), mean_grad_rate=mean(Graduation.rate))
> Summary_cont_priv
> #Public
> Summary_cont_pub <- Universities %>% filter(Public..1...Private..2. == 1) %>%
+     group_by(Clusters_3) %>%
+     summarise( Acceptance_rate = sum(X..appl..accepted)/ sum(X..appli..rec.d), Avg_out
_state_tuition=mean(out.of.state.tuition), Avg_int_state_tuition=mean(in.state.tuition),
room_board_fees=mean(room_board_fees), mean_PHD_fac=mean(X..fac..w.PHD), mean_stud_fac_r
atio=mean(stud..fac..ratio), mean_grad_rate=mean(Graduation.rate))
> Summary_cont_pub
> ###  Cluster 3 contains as the only cluster the majority public schools which means th
at the average tution rates are low.Moreove, the state of the school, the operating budg
et of the university, or the amount of academic endowments of the university are additio
nal information that could help to explain the data.There are differences between public
and private schools which would explain the reason of falling the public schools into a
different cluster.
> # Isolating the data to Tufts University cluster 2 that have the lowest distance:
>
> norm_Tufts <- scale(Universities1[, c(4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18
, 19, 20)])
> Tufts <- filter(Universities1, College.Name == "Tufts University")
>
> #Euclidean Distance for Each Clusters. Cluster 1 has the lowest distance from Tufts:

> #Cluster 1

> dist(rbind(Tufts[, -c(1, 2, 3, 10)], k3$centers[1,]))
                  1
2 29816.76


> #Cluster 2
> dist(rbind(Tufts[, -c(1, 2, 3, 10)], k3$centers[2,]))
        1
2 29817.8
> #Cluster 3
> dist(rbind(Tufts[, -c(1, 2, 3, 10)], k3$centers[3,]))
         1
2 29819.09
> #Impute NAs with average from cluster 2
> cluster_2 <- filter(Universities, Clusters_3 == 2)
> avg_cluster_2 <- mean(cluster_2[,c(10)])
> Tufts[, c(10)] <- avg_cluster_2
```

```
> Tufts[, c(10)]
[1] 396.7658
```