Clustering the merged data

Reading the cleaned and merged data that was prepared in Python

The data "UpdatedTJHemaSamik.csv" has the following columns,

Country - Name of the Country

lessthan5_50 - percentage of people earning less than \$5.5 per day

FPI - Quantity of Forest Products Imported per year in 2019

FDI - Financial Development Index

FIEI - Financial Institutions Efficiency Index

```
linkcsv="https://github.com/tjvijapurapu/542_ComputationalThinking/raw/main/UpdatedTJHemaSamik.csv"

mydata2=read.csv(linkcsv)
```

Preparing data

a. Choosing the following three variables - lessthan5_50, FPI, and FDI for the clustering analysis

```
Clus_Mydata2 = mydata2[,c('lessthan5_50', 'FPI', 'FDI')]
summary(Clus_Mydata2)
```

```
##   lessthan5_50        FPI               FDI
##  Min.   : 0.000   Min.   :     196   Min.   :0.05297
##  1st Qu.: 3.725   1st Qu.:   30716   1st Qu.:0.14482
##  Median :31.750   Median :  237201   Median :0.25797
##  Mean   :40.545   Mean   : 1108193   Mean   :0.32989
##  3rd Qu.:78.175   3rd Qu.:  961296   3rd Qu.:0.46262
##  Max.   :97.300   Max.   :16442309   Max.   :0.96396
##  NA's   :15
```

b. Scaling the data:

This step scales all the values belonging to different scale to a uniform scale. This step is essential for comparing different types of values. The scaled data is stored into a new variable called Clus_Mydata2

```
Clus_Mydata2 = scale(Clus_Mydata2)
summary(Clus_Mydata2)
```

```
##   lessthan5_50        FPI               FDI
##  Min.   :-1.1298   Min.   :-0.4895   Min.   :-1.2303
##  1st Qu.:-1.0260   1st Qu.:-0.4760   1st Qu.:-0.8222
##  Median :-0.2451   Median :-0.3848   Median :-0.3195
```

```
##   Mean    : 0.0000    Mean    : 0.0000    Mean    : 0.0000
##   3rd Qu.: 1.0486    3rd Qu.:-0.0649    3rd Qu.: 0.5897
##   Max.    : 1.5815    Max.    : 6.7748    Max.    : 2.8170
##   NA's    :15
```

c. Renaming subset indexes and verifying the input

This step renames the indexes with the country names for ease of understanding

```
row.names(Clus_Mydata2) = mydata2$Country
head(Clus_Mydata2)
```

```
##                      lessthan5_50          FPI          FDI
## Albania               -0.1879592 -0.45973085 -0.58069978
## Algeria               -0.3328595 -0.04918446 -0.82217916
## Angola                 1.3585738 -0.45984042 -0.76893012
## Antigua and Barbuda           NA -0.48494579 -0.06874061
## Argentina             -0.7898531 -0.09859597 -0.03711607
## Australia             -1.1103058  0.45648214  2.53448894
```

d. Setting the seed for replicability

By setting seed even if the code is rerun we can ensure getting same results

```
set.seed(234)
```

e. Deciding distance method and computing distance matrix

The distance of each value from the mean is calculated. This step is essential to understand the outliers and anomalies in the data. The final distance matrix is stored into a new variable called "Clus_Mydata2_Dist"

```
library(cluster)
Clus_Mydata2_Dist = daisy(x=Clus_Mydata2) #daisy is only for numerical data
```

Partitioning technique

1. Applying function by using 4 clusters

Using a set of k mediods (4 in this case), the pam function constructs k clusters by assigning each observation to the nearest mediod.

```
NumCluster = 4
res.pam = pam(x=Clus_Mydata2_Dist, k=NumCluster, cluster.only = F)
```

2.Evaluate results

2.1 Visualizing the silhouette plot and reporting average silhouettes

The four clusters produced using the pam function are visualized. The plot above the base line are positive silhouettes and the ones below are negative. The negative silhouettes are considered as anomalies in the data.
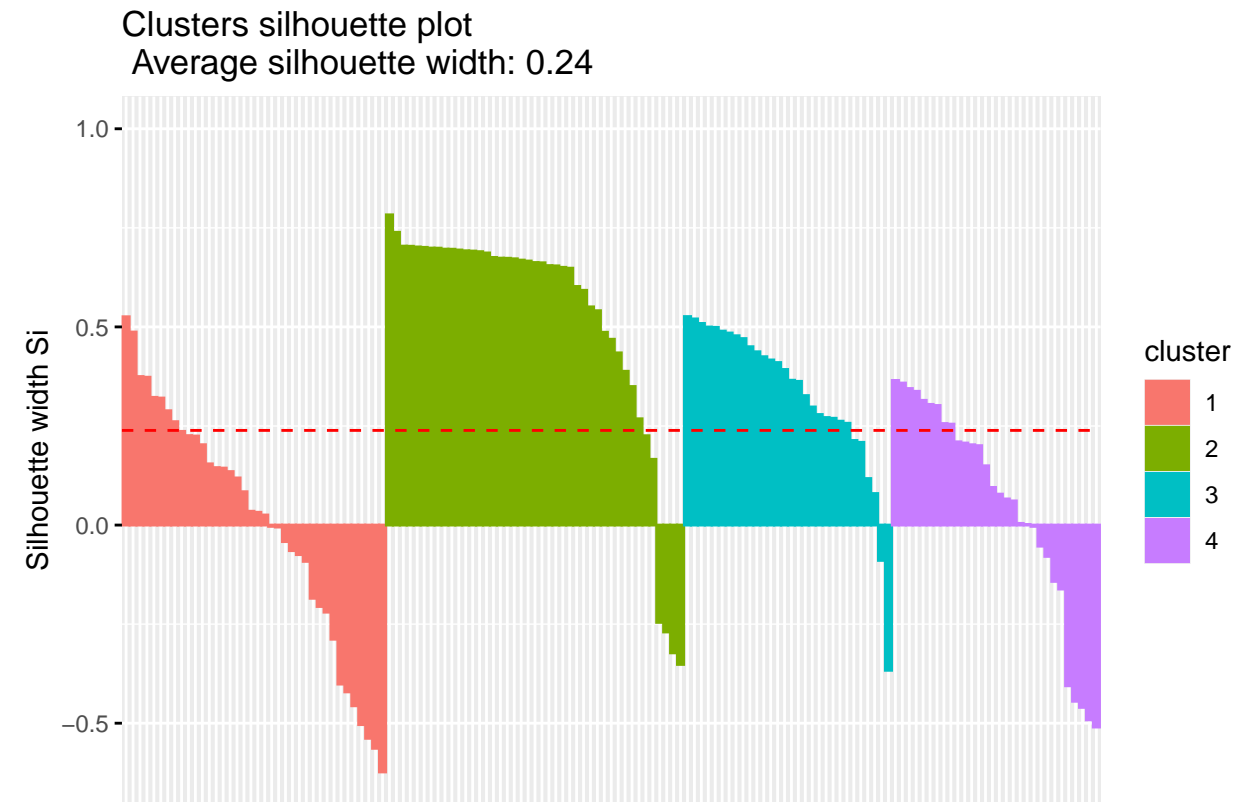
```
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_silhouette(res.pam)
```

```
##   cluster size ave.sil.width
## 1       1   38          0.00
## 2       2   43          0.52
## 3       3   30          0.33
## 4       4   30          0.05
```

Clusters silhouette plot
Average silhouette width: 0.24

2.2 Reporting and detecting anomalies

a. Individual silhouettes are saved in the column sil_width

```
pamEval = data.frame(res.pam$silinfo$widths)
head(pamEval)
```

```
##          cluster neighbor sil_width
## Dominica       1        2 0.5256818
## Grenada        1        3 0.4871110
## Samoa          1        3 0.3748396
## Albania        1        2 0.3730938
## Ecuador        1        3 0.3223090
## Bhutan         1        2 0.3205770
```

b. Requesting and filtering out negative silhouettes

If this happens in a research, the negative silhouette data are usually analyzed and the reasoning behind its anomaly is found. These data can be either removed or revisited and worked till

positive silhouettes are obtained. However, revisiting or removing data is beyond the scope of this project.

```
pamEval[pamEval$sil_width<0,]
```

```
##                        cluster neighbor    sil_width
## Uruguay                      1        3 -0.003289193
## Suriname                     1        2 -0.005090543
## Bosnia and Herzegovina       1        3 -0.041767287
## Latvia                       1        3 -0.064564594
## Maldives                     1        2 -0.074584825
## Costa Rica                   1        3 -0.091630563
## Estonia                      1        3 -0.184819413
## Seychelles                   1        3 -0.205502275
## Lebanon                      1        3 -0.219762777
## Pakistan                     1        2 -0.288064079
## Bangladesh                   1        2 -0.401348015
## Vanuatu                      1        2 -0.421005615
## Nigeria                      1        2 -0.456071044
## Papua New Guinea             1        2 -0.503528334
## Uzbekistan                   1        2 -0.538127338
## Kenya                        1        2 -0.563345814
## Togo                         1        2 -0.623012735
## Gabon                        2        1 -0.245553064
## Nicaragua                    2        1 -0.269797359
## Algeria                      2        1 -0.322443675
## Paraguay                     2        1 -0.351705786
## Egypt                        3        2 -0.089206755
## Antigua and Barbuda          3        1 -0.366448499
## India                        4        3 -0.003287645
## Turkey                       4        3 -0.053273674
## Luxembourg                   4        3 -0.079062270
## Norway                       4        3 -0.142203362
## Ireland                      4        3 -0.161523154
## Brazil                       4        3 -0.405547578
## Israel                       4        3 -0.444638797
## Saudi Arabia                 4        3 -0.460088088
## Greece                       4        3 -0.491502640
## New Zealand                  4        3 -0.509703987
```

Hierarchizing: agglomerative

1. Applying function

"agnes" function constructs a hierarchy of clusters. Two indices that are similar are clustered together and this keeps on getting built until all the values are clustered.

```
library(factoextra)

res.agnes= hcut(Clus_Mydata2_Dist,
                k = NumCluster,isdiss=T,
                hc_func='agnes',
                hc_method = "ward.D2")
```

2. Evaluate results

2.1 Reporting dendrogram

The hierarchy of clusters produced by agnes function is displayed as dendrogram

```r
library(factoextra)
library(ggplot2)
fviz_dend(res.agnes,k=NumCluster, cex = 0.7, horiz = T)
```

Cluster Dendrogram