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.Clustering results

2.1 Add results to the original data frame

```
mydata2$pam=as.factor(res.pam$clustering)
```

2.2 Query data frame as required. Some examples are given here

Example 1

```
mydata2[mydata2$pam==1,'Country']
```

```
##  [1] "Albania"                "Bangladesh"             "Belarus"
##  [4] "Belize"                 "Bhutan"                 "Bosnia and Herzegovina"
##  [7] "Botswana"               "Costa Rica"             "Dominica"
## [10] "Dominican Republic"     "Ecuador"                "El Salvador"
## [13] "Estonia"                "Fiji"                   "Georgia"
## [16] "Grenada"                "Guatemala"              "Honduras"
## [19] "Jamaica"                "Kenya"                  "Latvia"
## [22] "Lebanon"                "Lithuania"              "Maldives"
## [25] "Nigeria"                "Pakistan"               "Papua New Guinea"
## [28] "Samoa"                  "Seychelles"             "Sri Lanka"
## [31] "Suriname"               "Togo"                   "Tonga"
## [34] "Tunisia"                "Ukraine"                "Uruguay"
## [37] "Uzbekistan"             "Vanuatu"
```

Example 2

```
mydata2[mydata2$Country=="Ukraine",'pam']
```

```
## [1] 1
## Levels: 1 2 3 4
```

2.3 Reporting table of clusters

```
table(mydata2$pam)
```

```
##
##  1  2  3  4
## 38 43 30 30
```

3.Evaluate results

3.1 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.

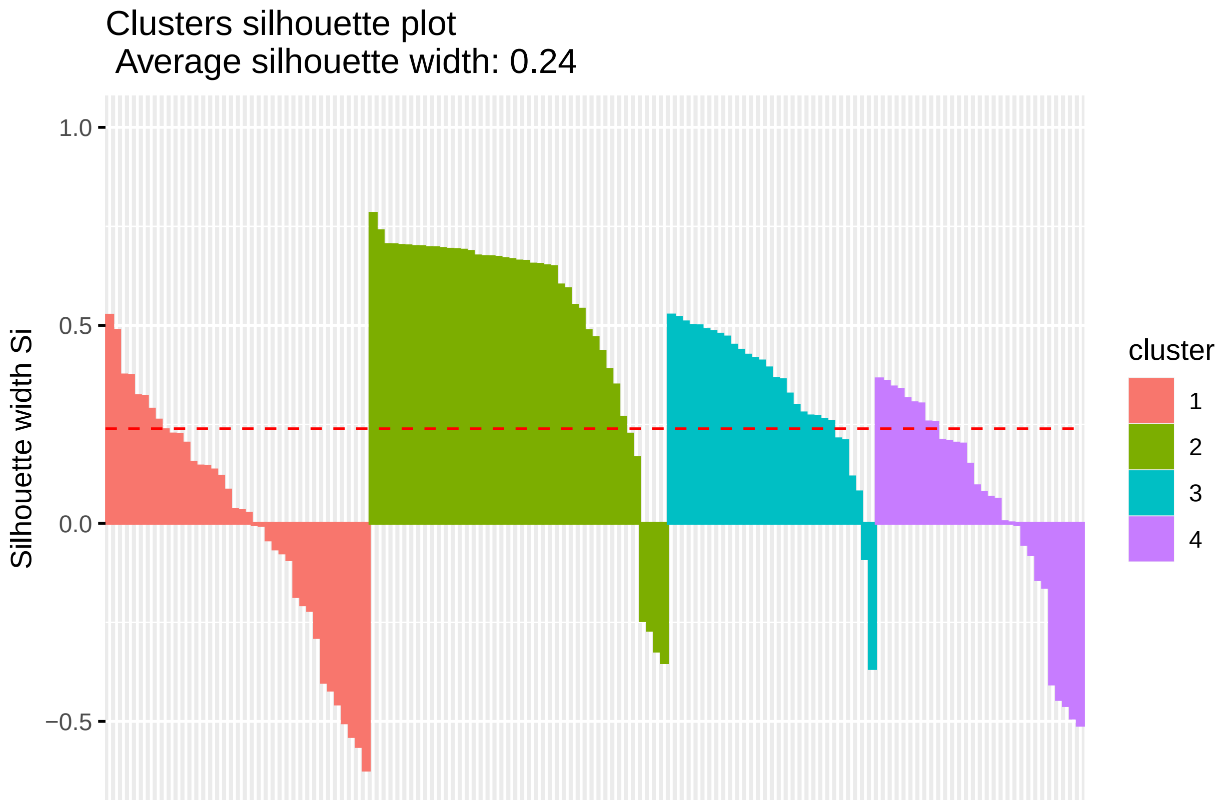
```
library(factoextra)
```

```
## 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
```



3.2 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.Clustering results

2.1 Add results to the original data frame

```
mydata2$agn=as.factor(res.agnes$cluster)
```

2.2 Query data frame as required. Some examples are given here

Example 1

```r
mydata2[mydata2$agn==1,'Country']
```

```
##  [1] "Albania"              "Algeria"              "Antigua and Barbuda"
##  [4] "Argentina"            "Barbados"             "Belarus"
##  [7] "Belize"               "Bhutan"               "Bosnia and Herzegovina"
## [10] "Botswana"             "Brazil"               "Bulgaria"
## [13] "Chile"                "Colombia"             "Costa Rica"
## [16] "Croatia"              "Cyprus"               "Dominica"
## [19] "Dominican Republic"   "Ecuador"              "Egypt"
## [22] "El Salvador"          "Estonia"              "Fiji"
## [25] "Gabon"                "Georgia"              "Greece"
## [28] "Grenada"              "Guatemala"            "Honduras"
## [31] "Hungary"              "Iceland"              "Indonesia"
## [34] "Israel"               "Jamaica"              "Jordan"
## [37] "Kazakhstan"           "Kuwait"               "Latvia"
## [40] "Lebanon"              "Libya"                "Lithuania"
## [43] "Malta"                "Mauritius"            "Mongolia"
## [46] "Morocco"              "Namibia"              "New Zealand"
## [49] "Nicaragua"            "Oman"                 "Panama"
## [52] "Paraguay"             "Peru"                 "Philippines"
## [55] "Qatar"                "Romania"              "Samoa"
## [58] "Saudi Arabia"         "Seychelles"           "Slovenia"
## [61] "South Africa"         "Sri Lanka"            "Suriname"
## [64] "Tonga"                "Trinidad and Tobago"  "Tunisia"
## [67] "Ukraine"              "United Arab Emirates" "Uruguay"
```

Example 2

```r
mydata2[mydata2$Country=="Ukraine",'pam']
```

```
## [1] 1
## Levels: 1 2 3 4
```

2.3 Reporting table of clusters

```r
table(mydata2$agn)
```

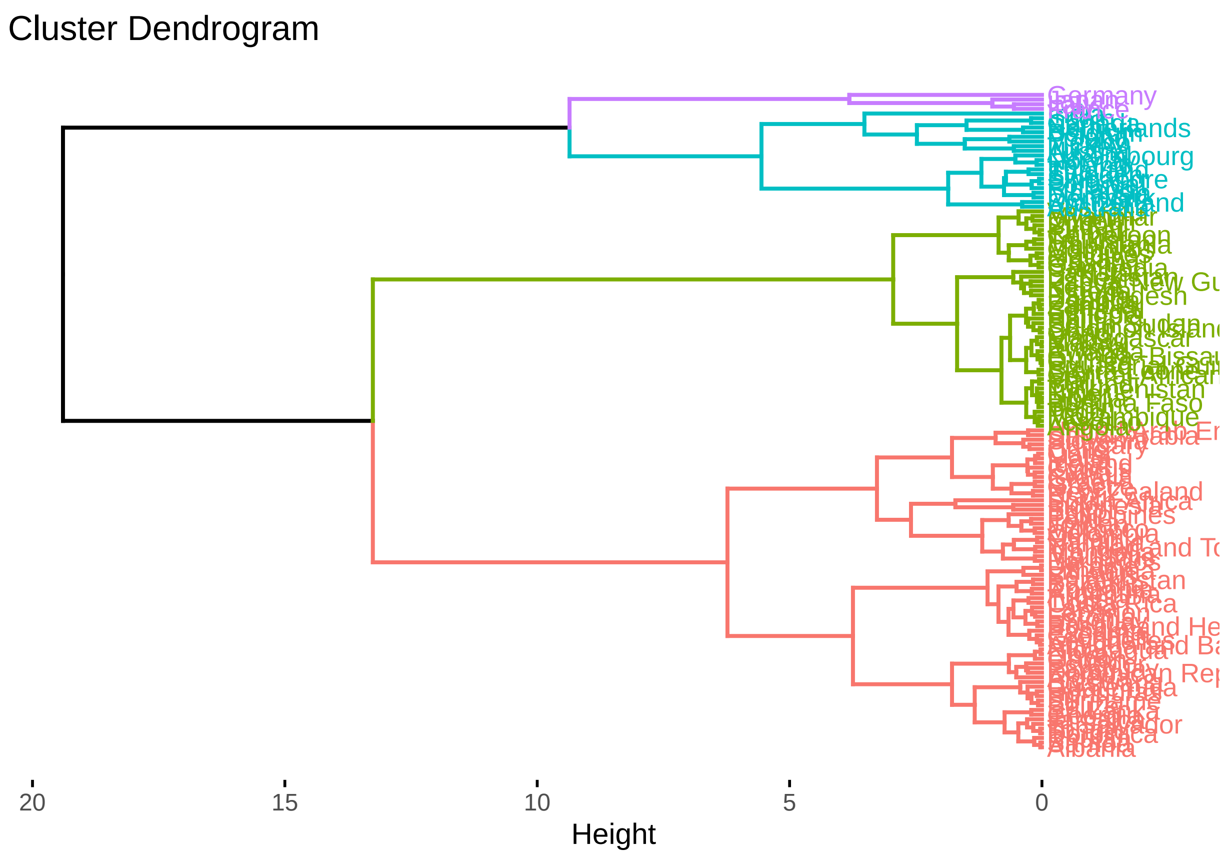
```
##
##  1  2  3  4
## 69 47 21  4
```

3. Evaluate results

3.1 Reporting dendrogram

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

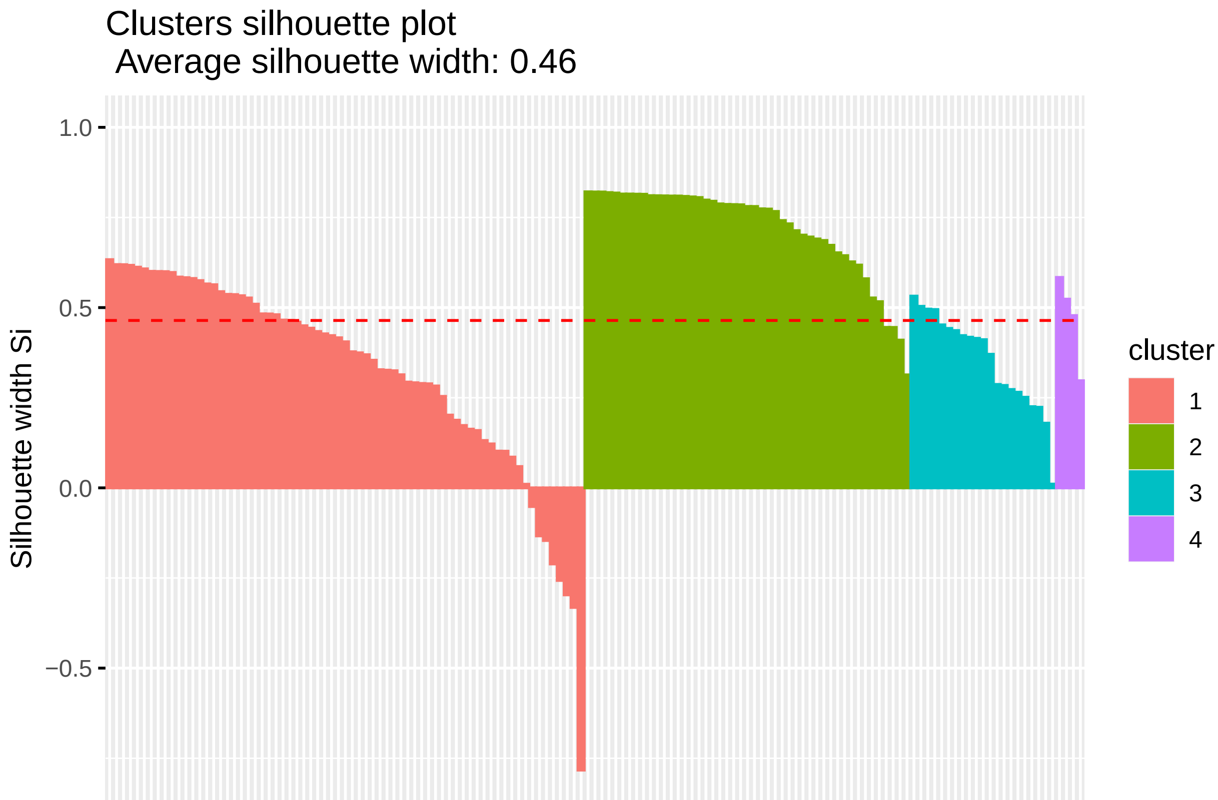
```r
library(factoextra)
library(ggplot2)
fviz_dend(res.agnes,k=NumCluster, cex = 0.7, horiz = T)
```



3.2 Reporting average silhouettes

```r
library(factoextra)
fviz_silhouette(res.agnes)
```

```
##   cluster size ave.sil.width
## 1       1   69          0.32
## 2       2   47          0.72
## 3       3   21          0.35
## 4       4    4          0.47
```



Detecting anomalies

a. Saving silhouettes

```
agnEval=data.frame(res.agnes$silinfo$widths)
head(agnEval)
```

```
##            cluster neighbor sil_width
## Kazakhstan       1        2 0.6330666
## Panama           1        2 0.6193669
## Jordan           1        2 0.6190718
## Argentina        1        2 0.6173024
## Lebanon          1        2 0.6121815
## Seychelles       1        2 0.6074691
```

b. Finding negative silhouettes

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

```
##          cluster neighbor   sil_width
## Fiji           1        2 -0.05186469
## Honduras       1        2 -0.13321911
## Egypt          1        2 -0.14605380
## Belize         1        2 -0.21102507
## Botswana       1        2 -0.25681540
## Suriname       1        2 -0.29680214
## Dominica       1        2 -0.33166380
```

```
## Libya          1        2 -0.78293367
```

Hierarchizing: Divisive

1. Applying function

"diana" function constructs a hierarchy of clusters returning an object of class "diana"

```
library(factoextra)

res.diana= hcut(Clus_Mydata2_Dist,
                k = NumCluster,
                hc_func='diana',
                hc_method = "ward.D")
```

2.Clustering results

2.1 Add results to the original data frame

```
mydata2$dia=as.factor(res.diana$cluster)
```

2.2 Query data frame as required. Some examples are given here

Example 1

```
mydata2[mydata2$dia==1,'Country']
```

```
##   [1] "Albania"                  "Algeria"
##   [3] "Angola"                   "Antigua and Barbuda"
##   [5] "Argentina"                "Bangladesh"
##   [7] "Barbados"                 "Belarus"
##   [9] "Belize"                   "Benin"
##  [11] "Bhutan"                   "Bosnia and Herzegovina"
##  [13] "Botswana"                 "Bulgaria"
##  [15] "Burkina Faso"             "Burundi"
##  [17] "Cambodia"                 "Cameroon"
##  [19] "Central African Republic" "Chad"
##  [21] "Chile"                    "Colombia"
##  [23] "Comoros"                  "Costa Rica"
##  [25] "Croatia"                  "Cyprus"
##  [27] "Djibouti"                 "Dominica"
##  [29] "Dominican Republic"       "Ecuador"
##  [31] "Egypt"                    "El Salvador"
##  [33] "Equatorial Guinea"        "Eritrea"
##  [35] "Estonia"                  "Ethiopia"
##  [37] "Fiji"                     "Gabon"
##  [39] "Georgia"                  "Ghana"
##  [41] "Grenada"                  "Guatemala"
##  [43] "Guinea"                   "Guinea-Bissau"
##  [45] "Guyana"                   "Haiti"
##  [47] "Honduras"                 "Hungary"
##  [49] "Iceland"                  "Indonesia"
##  [51] "Jamaica"                  "Jordan"
##  [53] "Kazakhstan"               "Kenya"
##  [55] "Kiribati"                 "Kuwait"
##  [57] "Latvia"                   "Lebanon"
```

```
##  [59] "Lesotho"                   "Liberia"
##  [61] "Libya"                     "Lithuania"
##  [63] "Madagascar"                "Malawi"
##  [65] "Maldives"                  "Mali"
##  [67] "Malta"                     "Mauritania"
##  [69] "Mauritius"                 "Mongolia"
##  [71] "Morocco"                   "Mozambique"
##  [73] "Myanmar"                   "Namibia"
##  [75] "New Zealand"               "Nicaragua"
##  [77] "Niger"                     "Nigeria"
##  [79] "Oman"                      "Pakistan"
##  [81] "Panama"                    "Papua New Guinea"
##  [83] "Paraguay"                  "Peru"
##  [85] "Philippines"               "Qatar"
##  [87] "Romania"                   "Rwanda"
##  [89] "Samoa"                     "Saudi Arabia"
##  [91] "Senegal"                   "Seychelles"
##  [93] "Sierra Leone"              "Slovenia"
##  [95] "Solomon Islands"           "South Africa"
##  [97] "South Sudan"               "Sri Lanka"
##  [99] "Sudan"                     "Suriname"
## [101] "Tajikistan"                "Togo"
## [103] "Tonga"                     "Trinidad and Tobago"
## [105] "Tunisia"                   "Turkmenistan"
## [107] "Uganda"                    "Ukraine"
## [109] "United Arab Emirates"      "Uruguay"
## [111] "Uzbekistan"                "Vanuatu"
## [113] "Zambia"
```

Example 2

```
mydata2[mydata2$Country=="Ukraine",'pam']
```

```
## [1] 1
## Levels: 1 2 3 4
```

2.3 Reporting table of clusters

```
table(mydata2$dia)
```

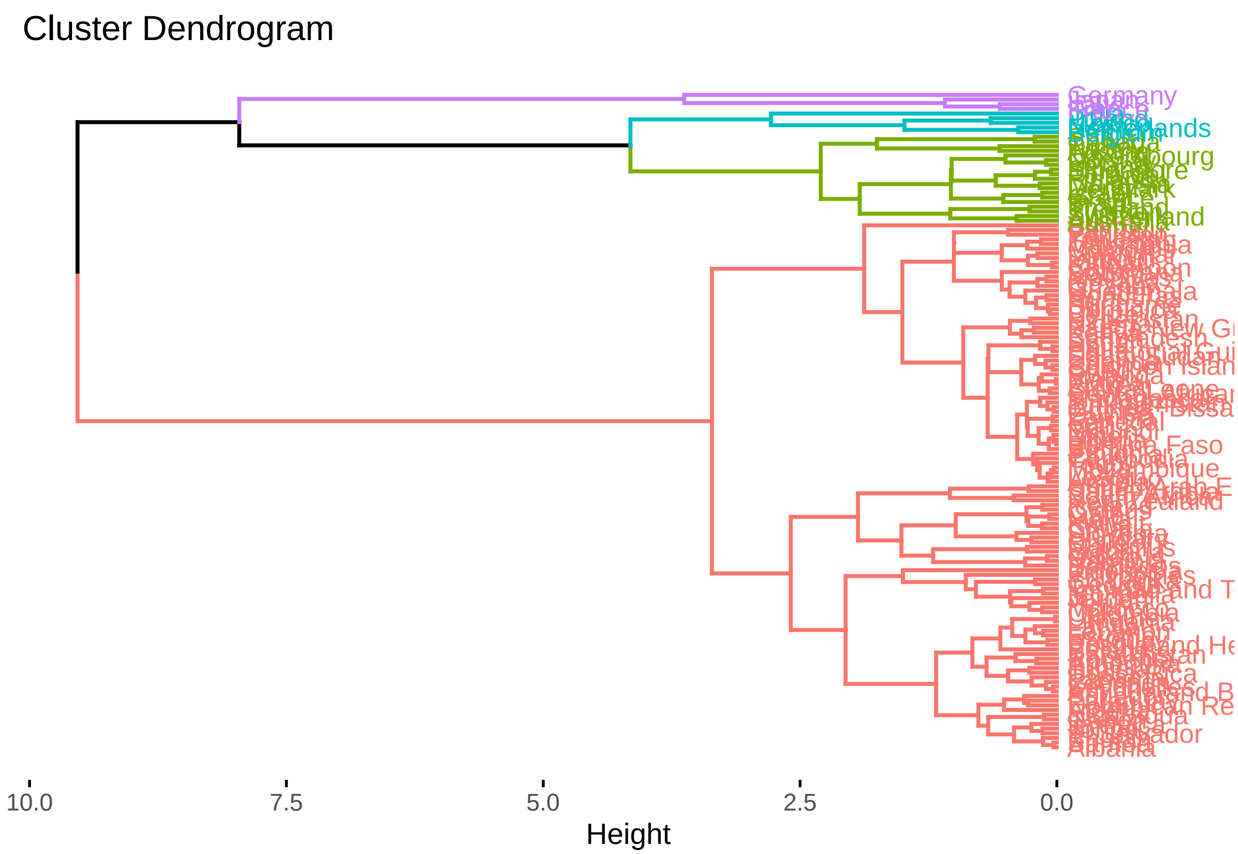
```
##
##   1   2   3   4
## 113  19   5   4
```

3. Evaluate results

3.1 Reporting dendrogram

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

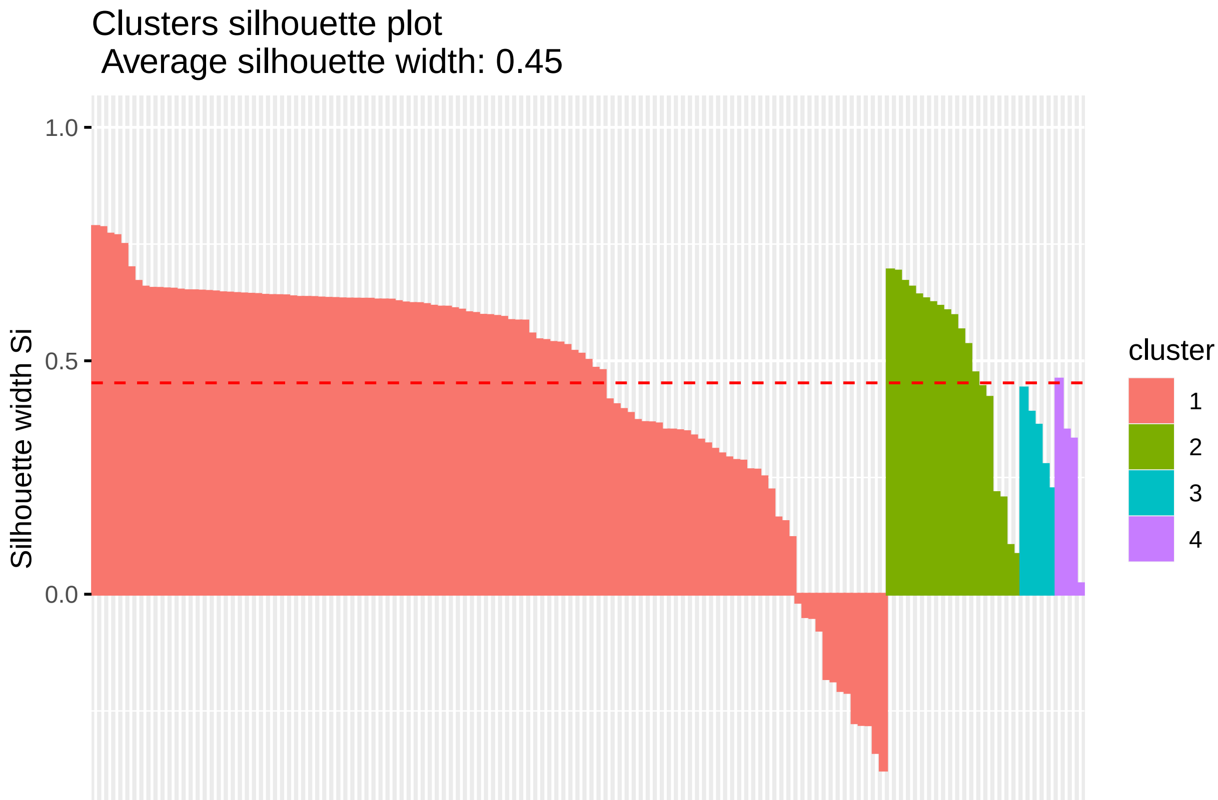
```
library(factoextra)
library(ggplot2)
fviz_dend(res.diana,k=NumCluster, cex = 0.7, horiz = T)
```



3.2 Reporting average silhouettes

```r
library(factoextra)
fviz_silhouette(res.diana)
```

```
##   cluster size ave.sil.width
## 1       1  113          0.46
## 2       2   19          0.50
## 3       3    5          0.34
## 4       4    4          0.29
```



Detecting anomalies

a. Saving silhouettes

```
diaEval=data.frame(res.diana$silinfo$widths)
head(diaEval)
```

```
##                    cluster neighbor sil_width
## Libya                    1        2 0.7875739
## Cambodia                 1        2 0.7853280
## Dominica                 1        2 0.7713236
## Equatorial Guinea        1        2 0.7680873
## Eritrea                  1        2 0.7492745
## Grenada                  1        2 0.6993301
```

b. Finding negative silhouettes

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

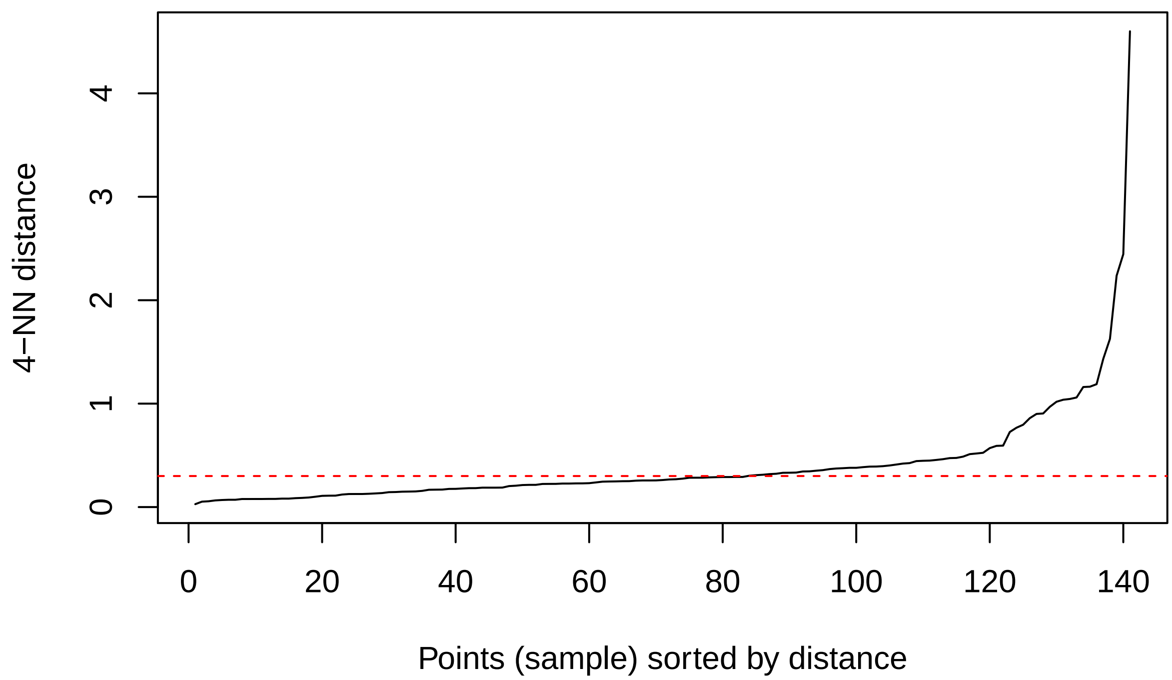
```
##                       cluster neighbor   sil_width
## Slovenia                    1        2 -0.01722643
## United Arab Emirates        1        2 -0.04797647
## Kuwait                      1        2 -0.04987818
## Qatar                       1        2 -0.07700742
## South Africa                1        2 -0.18074656
## Cyprus                      1        2 -0.18572715
## Chile                       1        2 -0.20624326
```

```
## Hungary                       1        2 -0.21026654
## Iceland                       1        2 -0.27520859
## Saudi Arabia                  1        2 -0.27900881
## Croatia                       1        2 -0.27953582
## Malta                         1        2 -0.33890718
## New Zealand                   1        2 -0.37673829
```

2.2 Density based clustering

Generated by inputting the distance and minimal number of neighbors that form a cluster

```
library(dbscan)
#minNeighs>= num cols in data
minNeighs=4
kNNdistplot(Clus_Mydata2_Dist, k = minNeighs)
abline(h=0.3, col = "red", lty=2)
```



2.3 Reporting Clusters

To find the number of clusters produced for similar values and the outliers present, we used "dbscan" function in the dbscan library

```
distance=0.3
res.db = dbscan::dbscan(Clus_Mydata2_Dist,
                        eps=distance,
                        minPts=minNeighs)
res.db
```

```
## DBSCAN clustering for 141 objects.
## Parameters: eps = 0.3, minPts = 4
## The clustering contains 1 cluster(s) and 38 noise points.
##
##   0   1
##  38 103
##
## Available fields: cluster, eps, minPts
```

Saving results in the original data frame

```r
mydata2$db=as.factor(res.db$cluster)
```

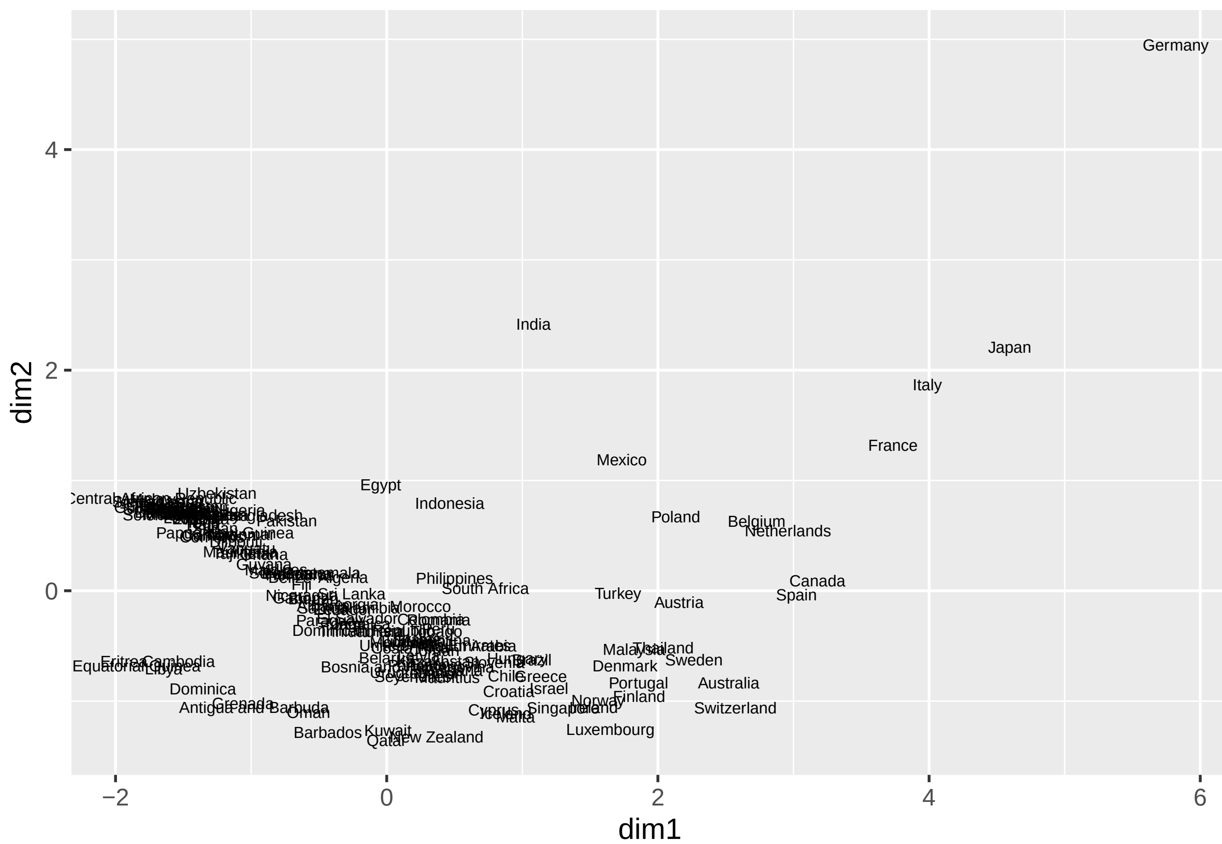
Comparing different clusters

1. We start by preparing a bidimensional map

```r
projectedData = cmdscale(Clus_Mydata2_Dist, k=2)
mydata2$dim1 = projectedData[,1]
mydata2$dim2 = projectedData[,2]
```

2. Visualizing the map

```r
base= ggplot(data=mydata2,
             aes(x=dim1, y=dim2,
                 label=Country))
base + geom_text(size=2)
```



Plotting PAM results

```r
pamPlot=base + labs(title = "PAM") + geom_point(size=2,
                                                aes(color=pam),
                                                show.legend = F)
```

Plotting hierarchical agnes results

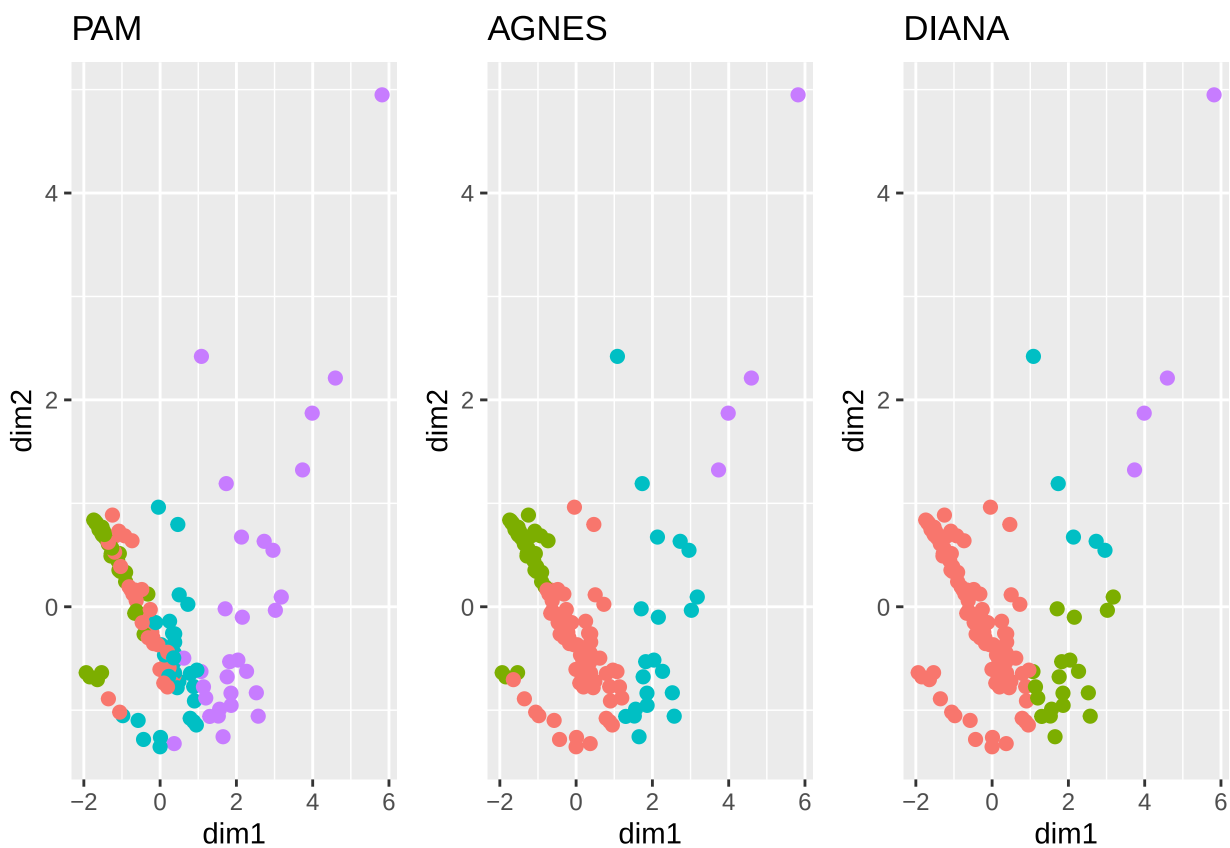
```r
agnPlot=base + labs(title = "AGNES") + geom_point(size=2,
                                                  aes(color=agn),
                                                  show.legend = F)
```

Plotting divisive diana results

```r
diaPlot=base + labs(title = "DIANA") + geom_point(size=2,
                                                  aes(color=dia),
                                                  show.legend = F)
```

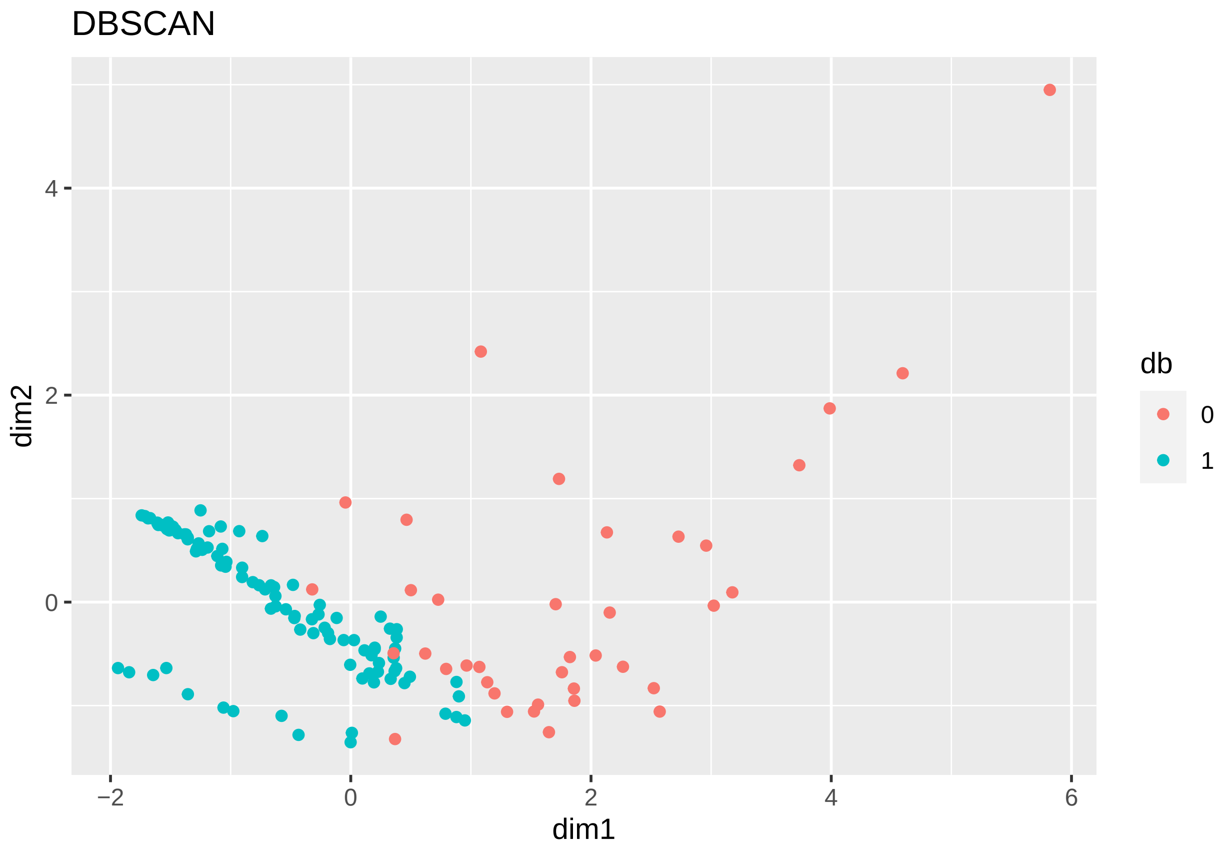
Comparing all the three plots visually

```r
library(ggpubr)
ggarrange(pamPlot, agnPlot, diaPlot, ncol = 3)
```



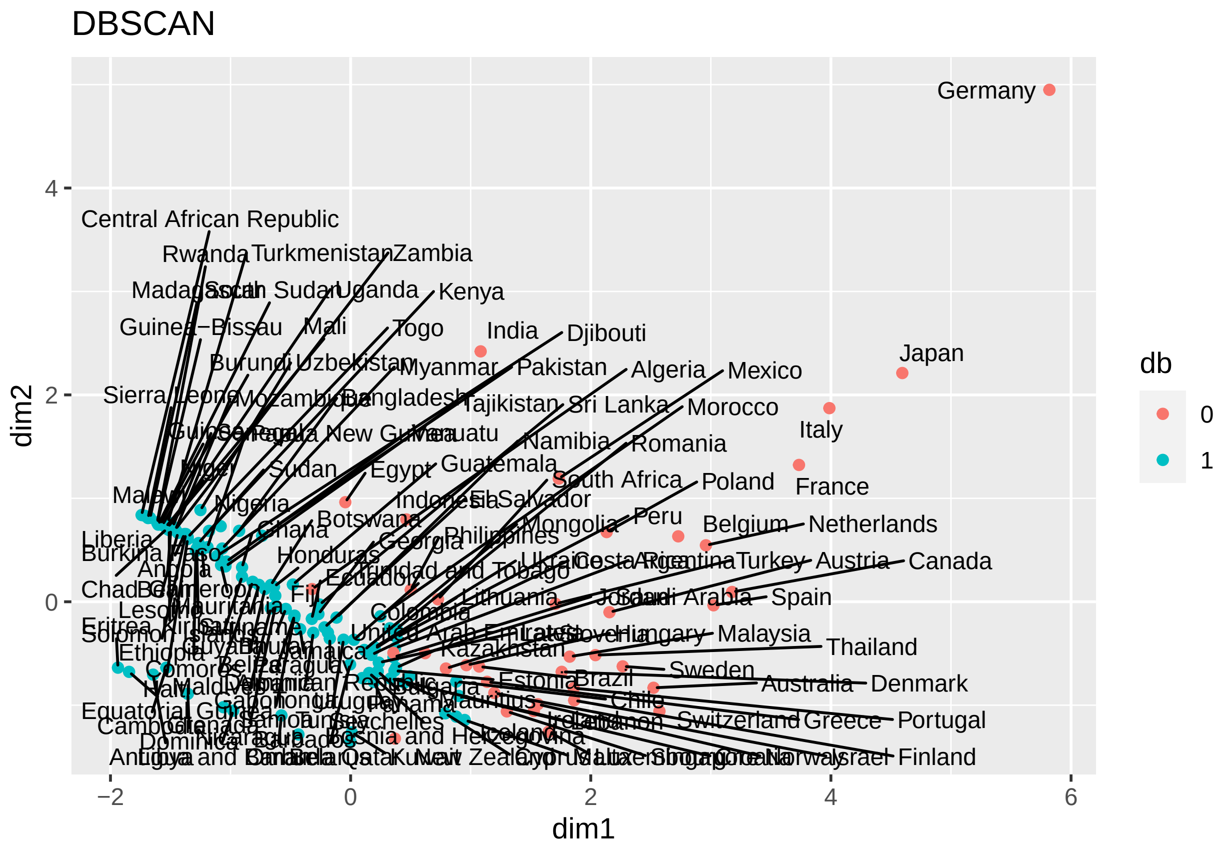
Plotting results from DBSCAN

```
dbPlot= base + labs(title = "DBSCAN") + geom_point(aes(color=db),
                                               show.legend = T)
dbPlot
```



Annotating

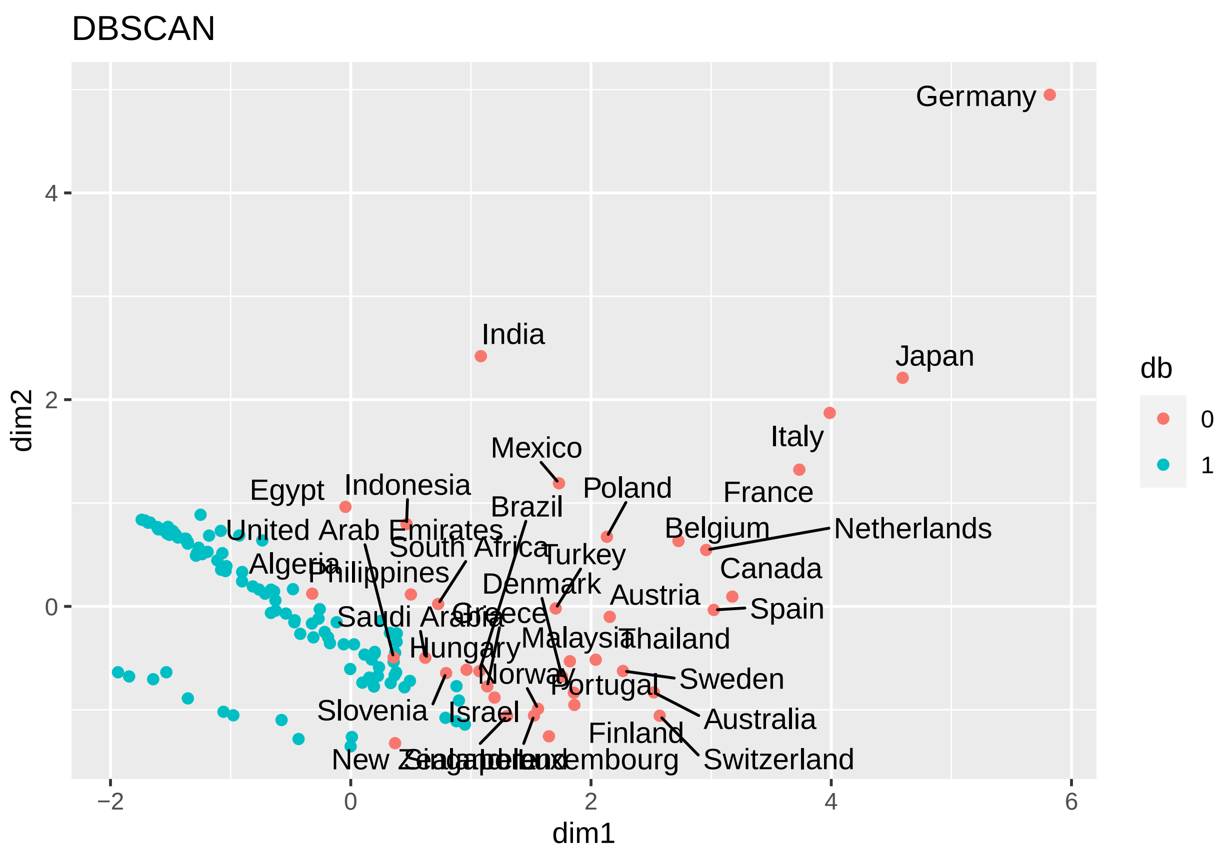
```
library(ggrepel)
dbPlot + geom_text_repel(size=3,aes(label=Country), max.overlaps = 200)
```



Annotating only the outliers

```
LABEL=ifelse(mydata2$db==0,mydata2$Country,"")

dbPlot + geom_text_repel(aes(label=LABEL), max.overlaps = 200)
```



DBSCAN
dim1
dim2
db
0
1
Germany
Japan
India
Italy
France
Mexico
Poland
Netherlands
Belgium
Canada
Spain
Austria
Egypt
Indonesia
Brazil
United Arab Emirates
South Africa
Turkey
Algeria
Philippines
Denmark
Saudi Arabia
Greece
Hungary
Malaysia
Thailand
Sweden
Norway
Portugal
Slovenia
Israel
Australia
Finland
Switzerland
New Zealand
Singapore
Ireland
Luxembourg