R Notebook

Code ▾

.

Hide

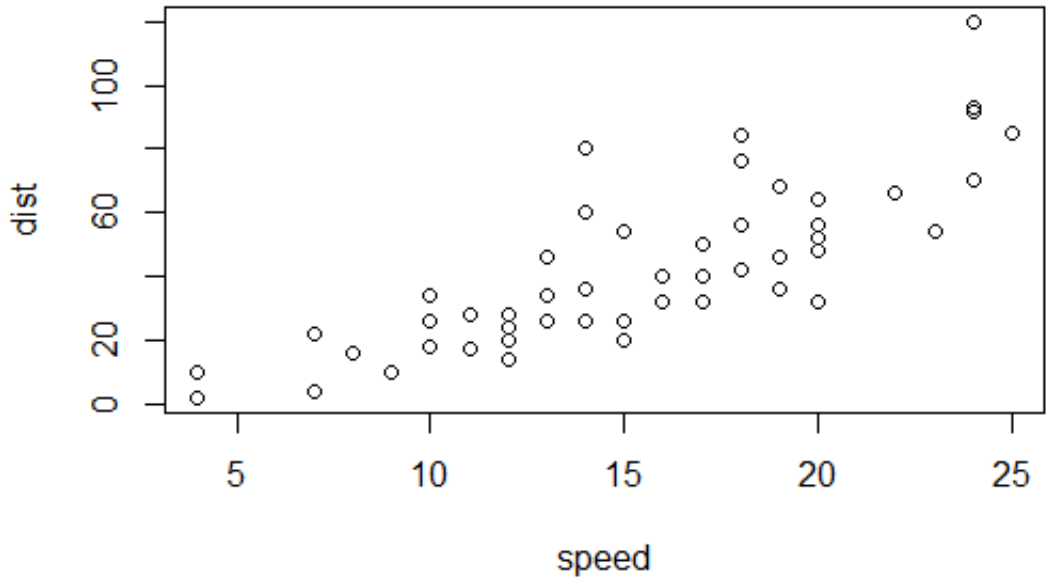
```
library(readr)
D1 <- read_csv("D1.csv")
```

```
Rows: 173 Columns: 7── Column specification ────────────────────────────────────────
Delimiter: ","
chr (1): Country
dbl (6): years, Population, Gini Index, Unemployment Rate, ...
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
View(D1)

plot(cars)
```



import D1 from directory

Hide

```
dataToCluster=D1[,-c(1)]
row.names(dataToCluster)=D1$Country
```

```
Warning: Setting row names on a tibble is deprecated.
```

subsetting the data to cluster

Hide

```
set.seed(999)
```

set random seed

Hide

```
library(cluster)
distanceMatrix=daisy(x=dataToCluster, metric = "gower")
```

Decide distance method and using gower

Hide

```
projectedData = cmdscale(distanceMatrix, k=2)
```

Representing the distance of 2

Hide

```
D1$dim1 = projectedData[,1]
D1$dim2 = projectedData[,2]
D1[,c('dim1','dim2')][1:7,]
```

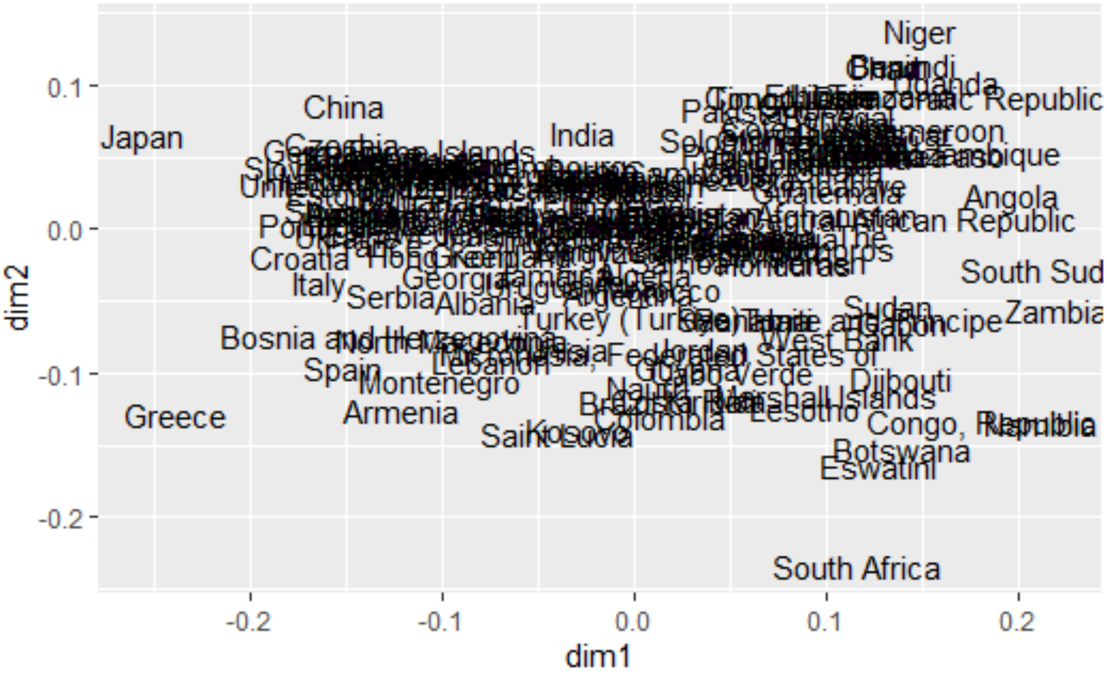
	dim1 <dbl>	dim2 <dbl>
	-0.25482166	0.06603283
	-0.15936096	0.05527021
	-0.16258625	-0.03601123
	-0.09837471	-0.01731216
	-0.23903006	-0.12850592
	-0.17168828	0.04346390
	-0.16528273	0.00388423

7 rows

saving coordinates for each element in the data

Hide

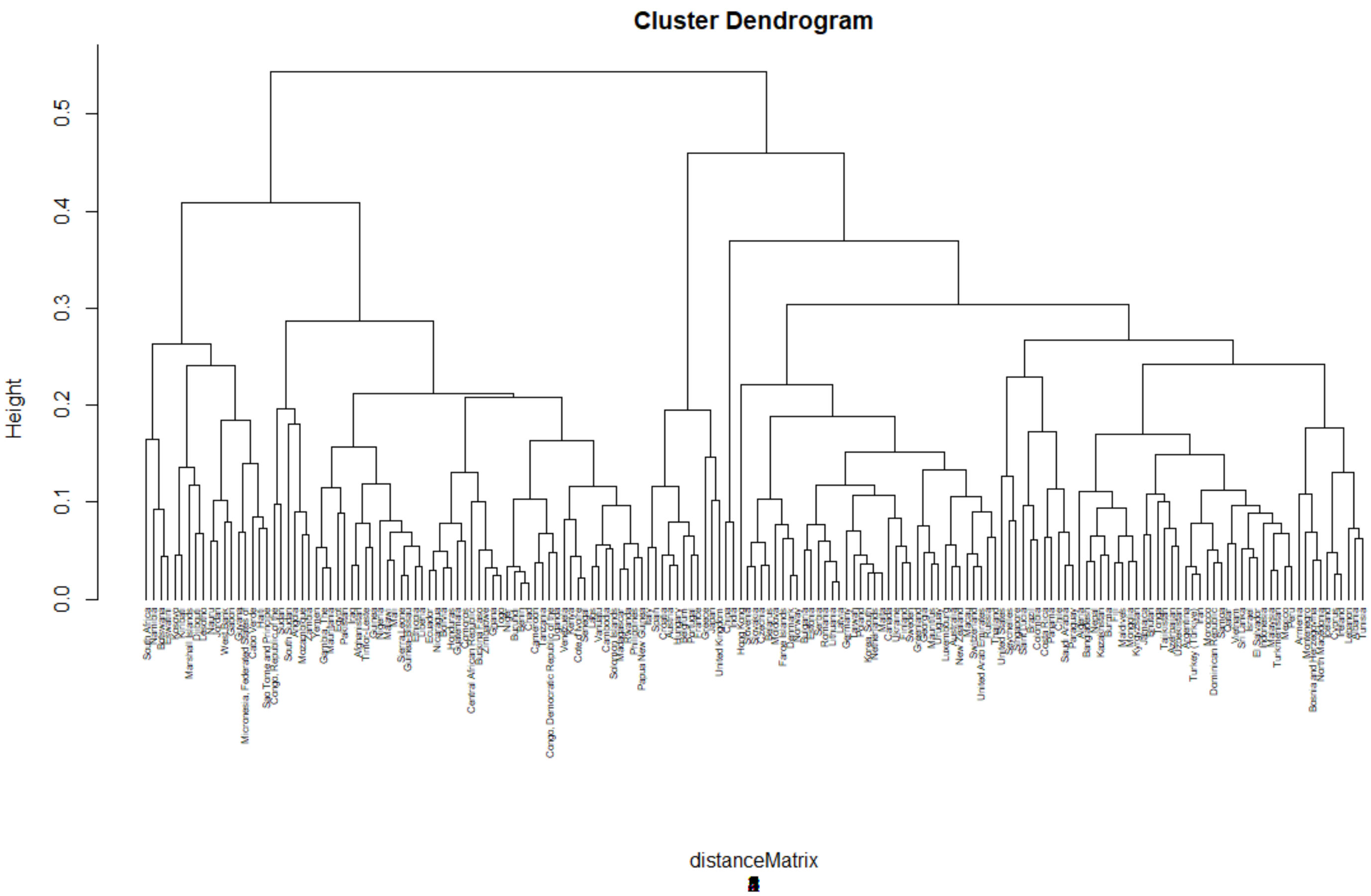
```
library(ggplot2)
base= ggplot(data=D1,
             aes(x=dim1, y=dim2,
                 label=Country))
base + geom_text(size=4)
```



simple map of average age and population clustering

Hide

```
hc = hclust(distanceMatrix)
subtree <- cutree(hc, k = 5)
plot(hc,hang=-1,cex=0.5,sub=subtree)
```



simple dendogram

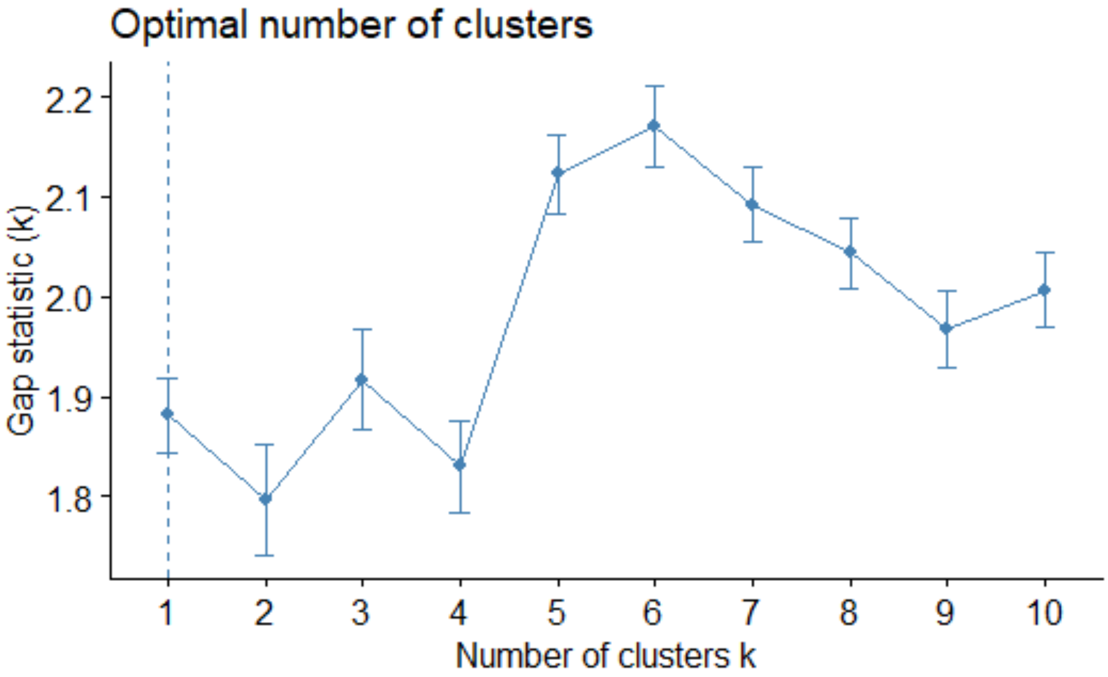
Hide

```
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
a
```

Hide

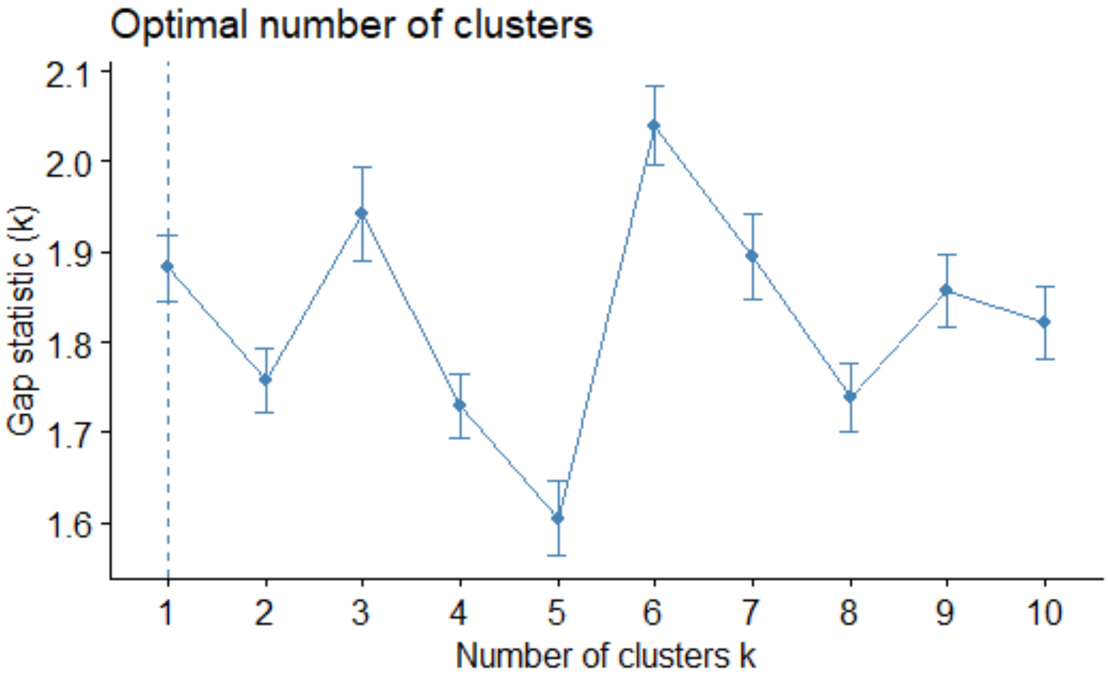
```
fviz_nbclust(dataToCluster,
             hcut,
             diss=distanceMatrix,
             method = "gap_stat",
             k.max = 10,
             verbose = F,
             hc_func = "agnes")
```



clustering suggestion(agglomerative) which outputs 6 as the optimal number of clusters

Hide

```
fviz_nbclust(dataToCluster,
             hcut,
             diss=distanceMatrix,
             method = "gap_stat",
             k.max = 10,
             verbose = F,
             hc_func = "diana")
```



clustering suggestion(divisive) which also outputs 6 as the optimal number of clusters

Hide

```


NumberOfClusterDesired=6
library(factoextra)
res.agnes= hcut(distanceMatrix,
                k = NumberOfClusterDesired,
                isdiss=TRUE,
                hc_func='agnes',
                hc_method = "ward.D2")


res.diana= hcut(distanceMatrix,
                k = NumberOfClusterDesired,
                isdiss=TRUE,
                hc_func='diana',
                hc_method = "ward.D2")
```

running two different methods using the suggested number of clustering 6

Hide

```
D1$agn=as.factor(res.agnes$cluster)
D1$dia=as.factor(res.diana$cluster)
```

save results to original data frame

Hide

```
library(dplyr)
D1$agn=dplyr::recode_factor(D1$agn,
                    `4`='1',`1` = '2',`2`='3',`3`='4',.ordered = T)
D1$dia=dplyr::recode_factor(D1$dia,
                    `1` = '1',`3`='2',`2`='3',`4`='4',.ordered = T)
```

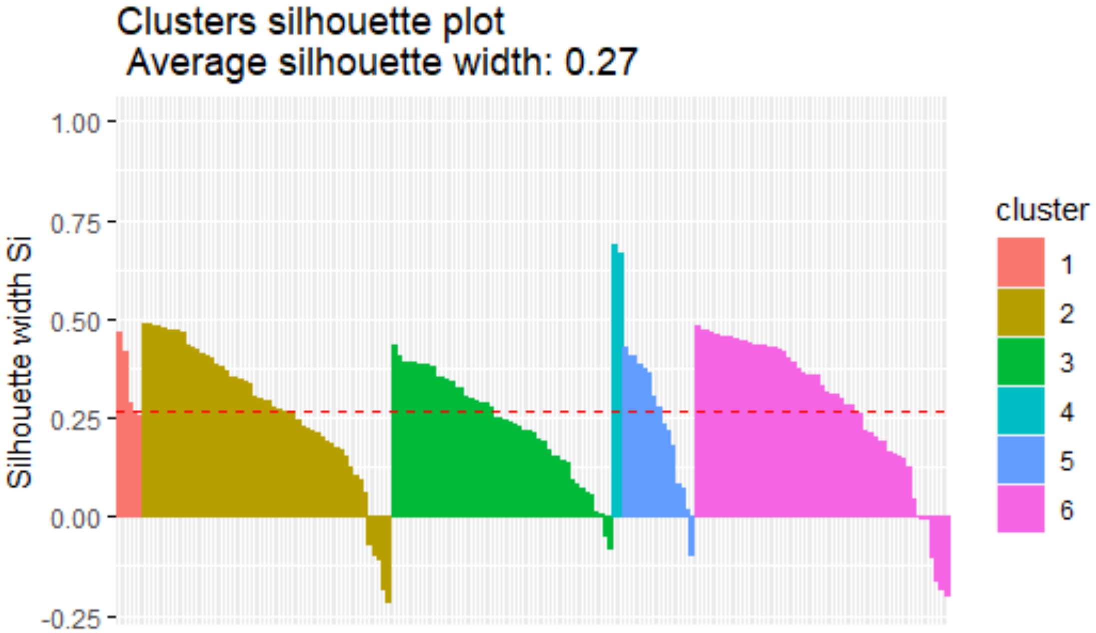
Ascending order

Hide

```
fviz_silhouette(res.agnes)
```

	cluster <fctr>	size <int>	ave.sil.width <dbl>
1	1	5	0.34
2	2	52	0.27
3	3	46	0.24
4	4	2	0.68
5	5	15	0.24
6	6	53	0.28

6 rows

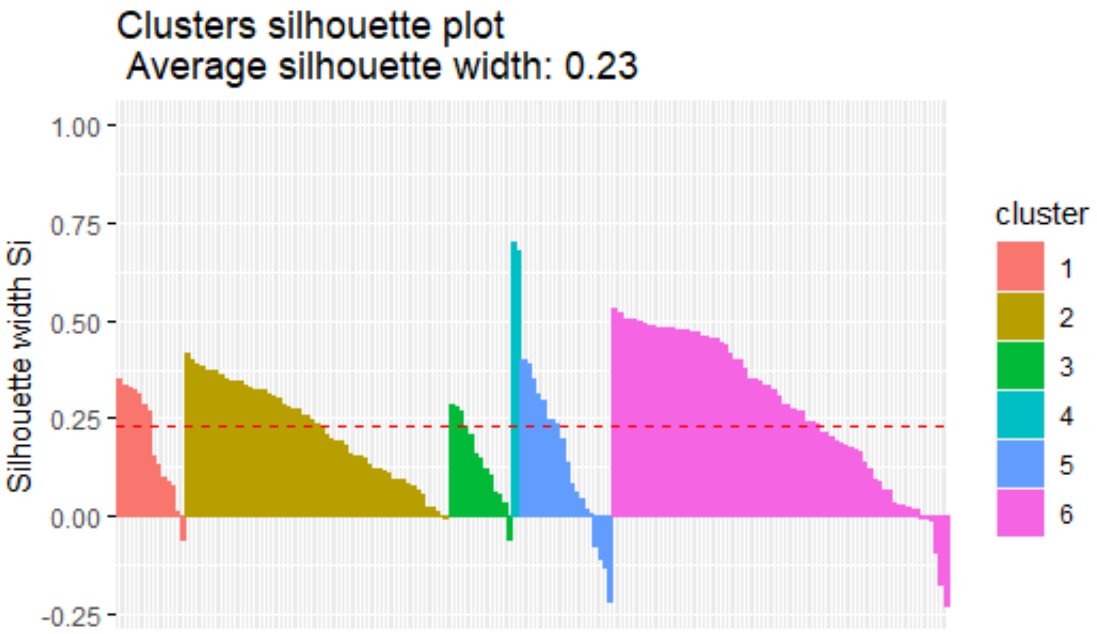


Hide

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

	cluster <fctr>	size <int>	ave.sil.width <dbl>
1	1	14	0.19
2	2	55	0.22
3	3	13	0.15
4	4	2	0.69
5	5	19	0.13
6	6	70	0.28

6 rows



The clustering average shows a somewhat okay clustering

Hide

```
agnEval=data.frame(res.agnes$silinfo$widths)
diaEval=data.frame(res.diana$silinfo$widths)
agnPoor=rownames(agnEval[agnEval$sil_width<0,])
diaPoor=rownames(diaEval[diaEval$sil_width<0,])
```

Hide

```
library("qpcR")
bad_Clus=as.data.frame(qpcR:::cbind.na(sort(agnPoor),
                                       sort(diaPoor)))
names(bad_Clus)=c("agn","dia")
bad_Clus
```

agn	**dia**
<chr>	<chr>
Albania	Algeria
Bolivia	Bhutan
Cambodia	Colombia
Chile	Costa Rica
Egypt	El Salvador
France	Guyana
Gabon	Iran
Guyana	Kosovo
Honduras	Lebanon
Lebanon	Micronesia, Federated States of

1-10 of 14 rows Previous **1** 2 Next

These a the listed countries that are poorly clustered by both methods

Hide

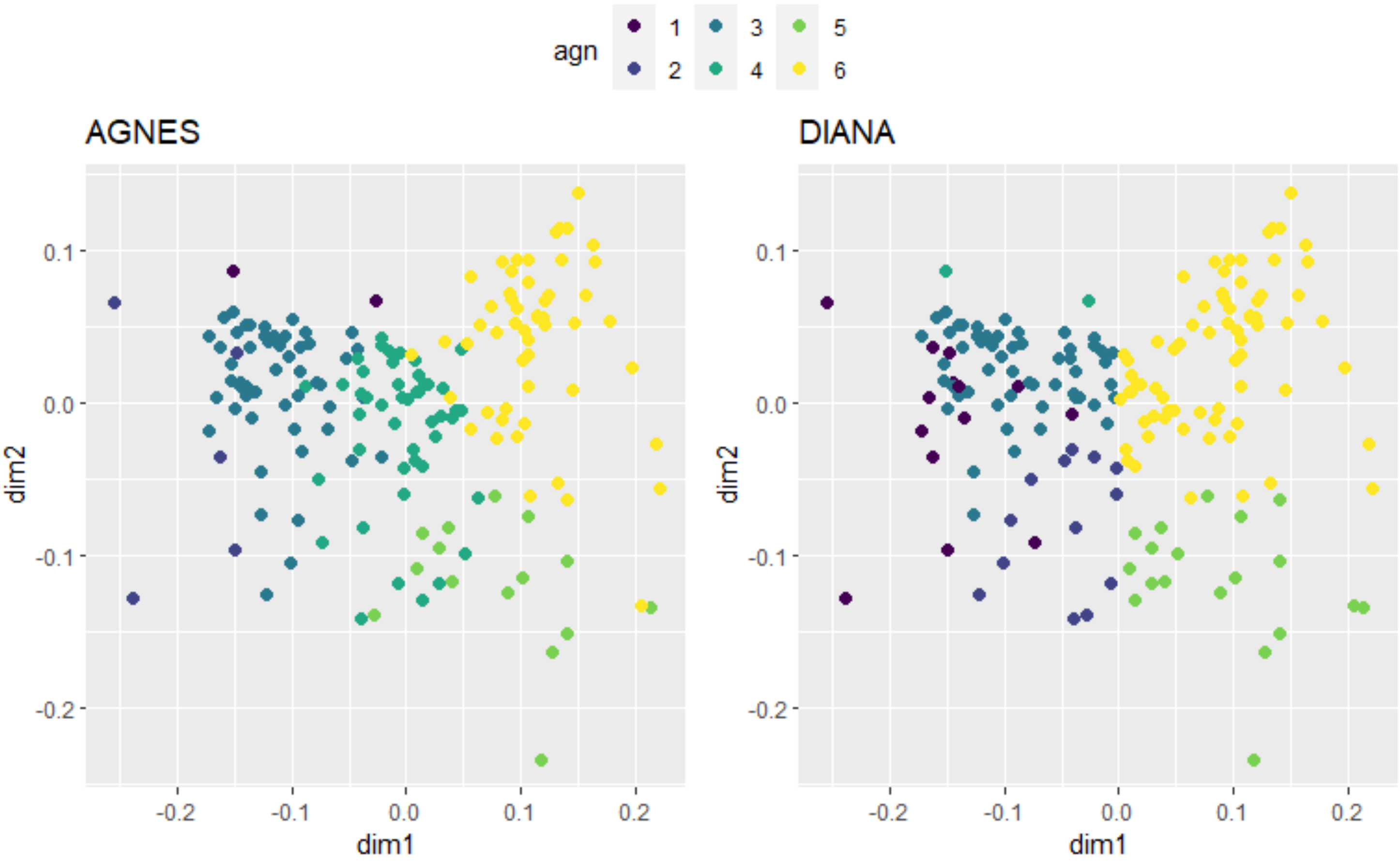
```
base= ggplot(data=D1,
             aes(x=dim1, y=dim2,
                 label=Country))
agnPlot=base + labs(title = "AGNES") + geom_point(size=2,
                                              aes(color=agn),
                                              show.legend = T)
```

Hide

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

Hide

```
library(ggpubr)
ggarrange(agnPlot, diaPlot,ncol = 2,common.legend = T)
```



Hide

```
# If name of country in black list, use it, else get rid of it
LABELdia=ifelse(D1$Country%in%diaPoor,D1$Country,"")
LABELagn=ifelse(D1$Country%in%agnPoor,D1$Country,"")
```

Hide

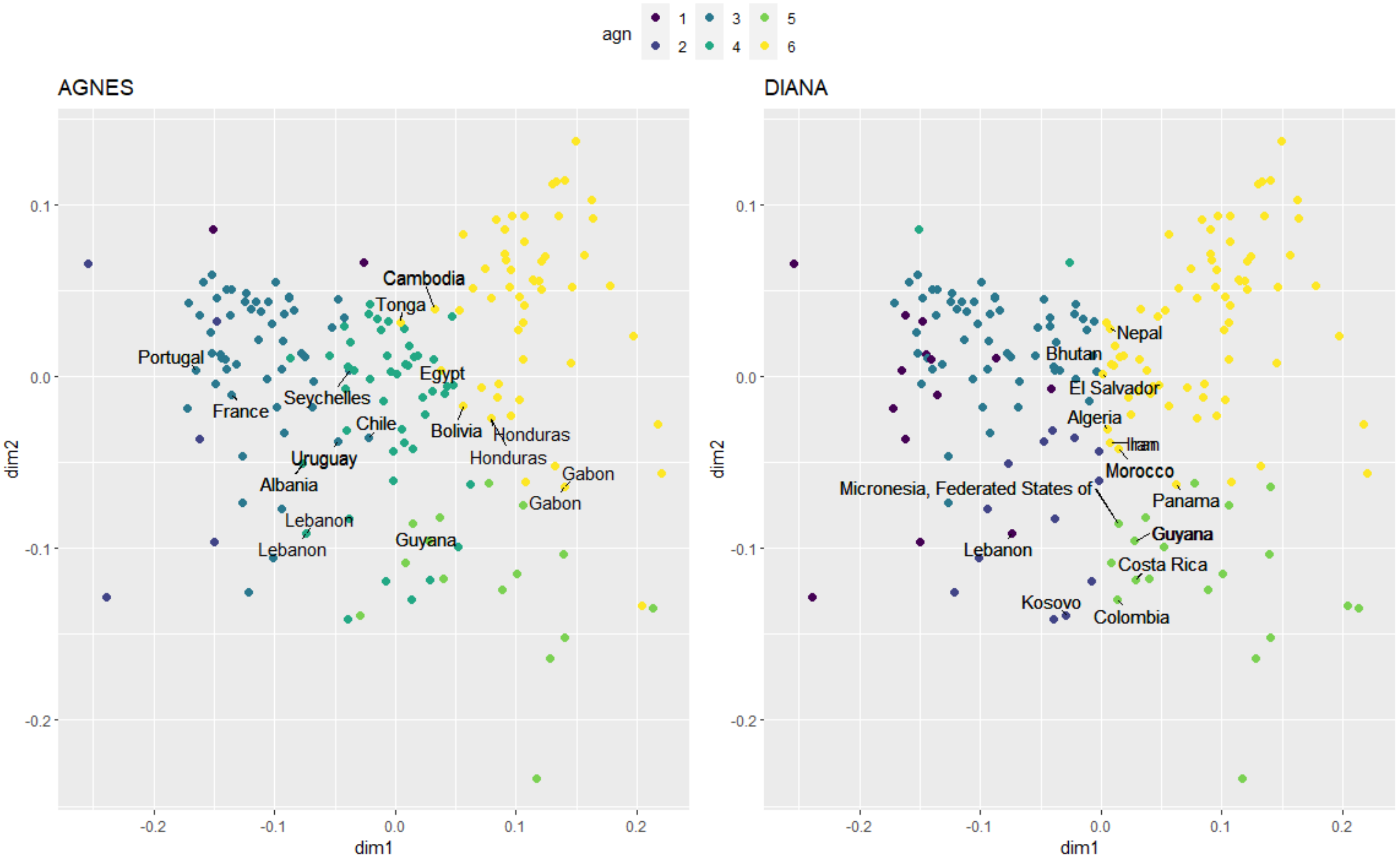
```
library(ggrepel)
diaPlot=diaPlot +
        geom_text_repel(aes(label=LABELdia),
                        max.overlaps=50,
                        min.segment.length =unit(0,'lines'))
```

Hide

```
agnPlot= agnPlot +
         geom_text_repel(aes(label=LABELagn),
                         max.overlaps = 50,
                         min.segment.length = unit(0, 'lines'))
```

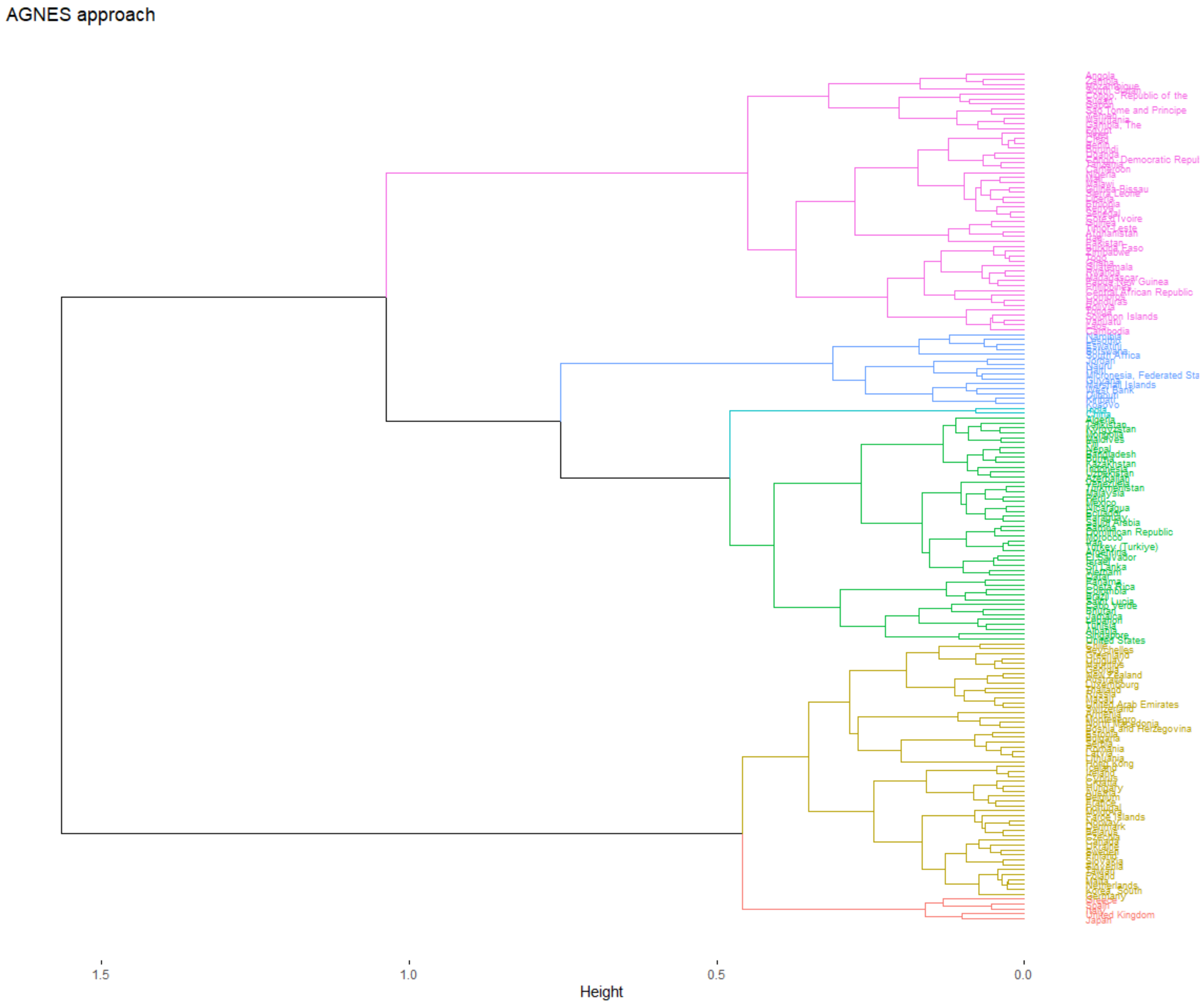
Hide

```
ggarrange(agnPlot,
          diaPlot,
          ncol = 2,
          common.legend = T)
```



Hide

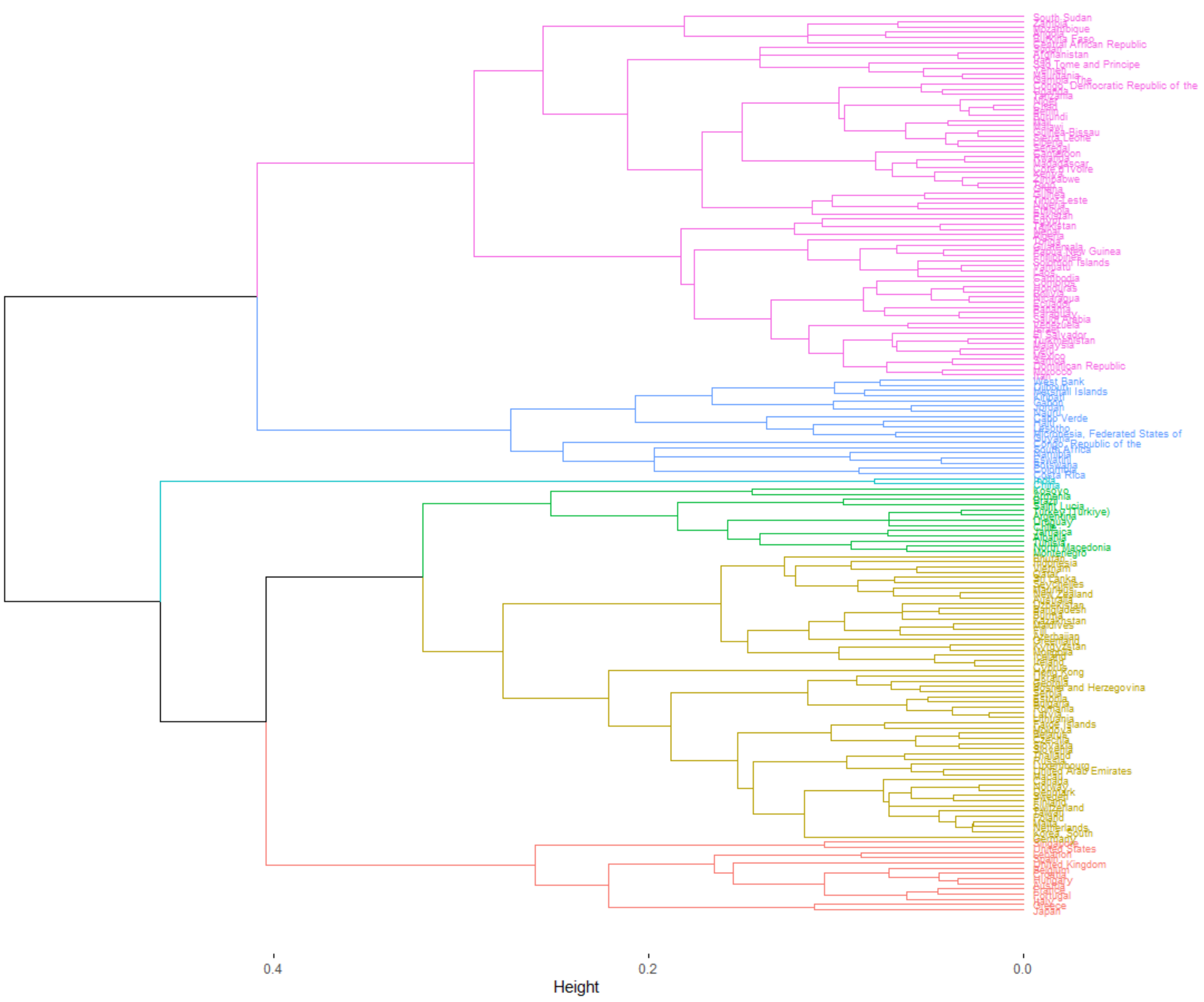
```
fviz_dend(res.agnes,
          k=NumberOfClusterDesired,
          cex = 0.45,
          horiz = T,
          main = "AGNES approach")
```



Hide

```
fviz_dend(res.diana,
          k=NumberOfClusterDesired,
          cex = 0.45,
          horiz = T,
          main = "DIANA approach")
```

DIANA approach



Hide

```
table(D1$`Unemployment Rate`,D1$agn)
```

```
        1 2 3 4 5 6
  0.26  0 0 0 1 0 0
  0.61  0 0 0 0 0 1
  0.75  0 0 0 0 0 1
  1.03  0 0 0 0 0 1
  1.26  0 0 0 0 0 1
  1.42  0 0 1 0 0 0
  1.57  0 0 0 0 0 1
  1.61  0 0 0 0 0 1
  1.79  0 0 0 0 0 1
  1.88  0 0 0 0 0 1
  2.17  0 0 0 2 0 0
  2.18  0 0 0 0 0 1
  2.2   0 0 1 0 0 0
  2.41  0 0 0 0 0 1
  2.59  0 0 0 0 0 1
  2.65  0 0 0 0 0 1
  2.75  0 0 0 0 0 1
  2.8   0 1 0 0 0 0
  2.89  0 0 1 0 0 0
  2.94  0 0 0 0 0 1
  3     0 0 1 0 0 0
  3.01  0 0 1 0 0 0
  3.36  0 0 1 0 0 0
  3.37  0 0 1 0 0 0
  3.47  0 0 0 0 0 1
  3.5   0 0 1 0 0 0
  3.53  0 0 1 0 0 0
  3.54  0 0 1 0 0 0
  3.57  0 0 0 0 0 1
  3.62  0 0 0 1 0 0
  3.69  0 0 0 0 0 1
  3.72  0 0 0 0 0 1
  3.73  0 0 1 0 0 0
  3.87  0 0 0 0 0 1
  3.96  0 0 1 0 0 0
  3.97  0 0 0 0 0 1
  3.98  0 0 0 0 0 1
  4     0 0 0 0 0 1
  4.01  0 0 1 0 0 0
  4.09  0 0 0 0 0 1
  4.12  0 0 2 0 0 0
  4.33  0 0 0 1 0 0
  4.35  0 0 0 0 0 1
  4.38  0 0 0 1 0 0
  4.41  0 0 0 1 0 0
  4.42  0 0 1 0 0 0
  4.53  0 1 0 0 0 0
  4.61  0 0 0 1 0 0
  4.7   0 0 0 0 0 1
  4.74  0 0 1 0 0 0
  4.76  0 0 0 0 0 1
  4.8   0 0 1 0 0 0
  4.82  1 0 0 0 0 0
  4.83  0 0 0 1 0 0
  4.9   0 0 0 1 0 0
  4.99  0 0 1 0 0 0
  5.01  0 0 1 0 0 0
  5.05  0 0 0 2 0 0
  5.07  0 0 0 0 0 1
  5.08  0 0 0 1 0 0
  5.11  0 0 1 0 0 0
  5.17  0 0 1 0 0 1
  5.23  0 0 1 1 0 0
  5.24  0 0 0 1 0 0
  5.32  0 0 2 0 0 0
  5.33  0 0 0 0 0 1
  5.39  0 0 0 1 0 0
  5.4   0 0 1 0 0 0
  5.42  0 0 1 0 0 0
  5.43  0 0 0 0 0 1
  5.46  0 0 0 1 0 0
  5.74  0 0 0 0 0 1
  5.94  0 0 0 1 0 0
  5.96  0 0 0 1 0 0
  5.98  1 0 0 0 0 0
  6.08  0 0 0 1 0 0
  6.13  0 0 1 0 0 0
  6.3   0 0 1 0 0 0
  6.33  0 0 1 0 0 0
  6.34  0 0 0 0 0 1
  6.41  0 0 0 1 0 0
  6.42  0 0 1 0 0 0
  6.43  0 0 0 1 0 0
  6.57  0 0 0 0 0 1
  6.58  0 0 0 1 0 0
  6.63  0 0 1 0 0 0
  6.65  0 0 1 0 0 0
  6.74  0 0 1 0 0 0
  6.76  0 0 0 0 0 1
  7.02  0 0 0 0 0 1
  7.08  0 0 0 1 0 0
  7.16  0 0 0 1 0 0
  7.21  0 0 0 1 0 0
  7.36  0 0 0 1 0 0
  7.41  0 0 1 0 0 0
  7.51  0 0 1 0 0 0
  7.53  0 0 1 0 0 0
  7.6   0 0 1 0 0 0
  7.72  0 0 0 0 0 1
  7.75  0 0 0 1 0 0
  7.9   0 0 1 0 0 0
  8.06  0 0 1 0 0 0
  8.5   0 0 0 1 0 0
  8.51  0 0 0 0 0 2
  8.53  0 0 0 0 0 1
  8.66  0 0 1 0 0 0
  8.68  0 0 1 0 0 0
  8.88  0 0 1 0 0 0
  9.1   0 0 1 1 0 0
  9.13  0 0 1 0 0 0
  9.18  0 0 0 1 0 0
  9.33  0 0 0 0 0 1
  9.45  0 0 0 0 0 1
  9.79  0 0 0 0 0 1
  9.83  0 1 0 0 0 0
  9.84  0 0 0 1 0 0
  10.45 0 0 1 0 0 0
  10.66 0 0 1 0 0 0
  10.9  0 0 0 1 0 0
  11.21 0 0 0 0 0 1
  11.46 0 0 0 1 0 1
  11.47 0 0 0 1 0 0
  11.81 0 0 1 0 0 0
  11.82 0 0 0 1 0 0
  12.09 0 0 0 1 0 0
  12.7  0 0 0 1 0 0
  13.03 0 0 0 0 0 1
  13.28 0 0 0 0 0 1
  13.39 0 0 0 1 0 0
  13.57 0 0 0 0 0 1
  13.91 0 0 0 0 0 1
  14.19 0 0 0 0 0 1
  14.34 0 0 0 1 0 0
  14.4  0 0 0 1 0 0
  14.49 0 0 0 1 0 0
  14.73 0 1 0 0 0 0
  14.8  0 1 0 0 0 0
  15.22 0 0 1 0 0 0
  15.42 0 0 0 1 0 0
  15.73 0 0 0 0 1 0
  15.91 0 0 0 0 0 1
  16.2  0 0 1 0 1 0
  16.42 0 0 0 0 1 0
  16.82 0 0 0 1 0 0
  16.91 0 0 0 1 0 0
  17.95 0 0 0 1 0 0
  18.49 0 0 1 0 0 0
  19.25 0 0 0 0 1 0
  19.81 0 0 0 0 0 1
  20.9  0 0 1 0 0 0
  21.68 0 0 0 0 1 0
  22.26 0 0 0 0 0 1
  23    0 0 0 0 1 0
  23.01 0 0 0 0 0 1
  24.6  0 0 0 0 1 0
  24.72 0 0 0 0 1 0
  24.9  0 0 0 0 1 0
  25.76 0 0 0 0 1 0
  28.39 0 0 0 0 1 0
  30.5  0 0 0 0 1 0
  30.6  0 0 0 0 1 0
  33.56 0 0 0 0 1 0
  36    0 0 0 0 1 0
```

Hide

```
table(D1$`Unemployment Rate`,D1$dia)
```

```

        1 2 3 4 5 6
  0.26  0 0 1 0 0 0
  0.61  0 0 0 0 0 1
  0.75  0 0 0 0 0 1
  1.03  0 0 0 0 0 1
  1.26  0 0 0 0 0 1
  1.42  0 0 1 0 0 0
  1.57  0 0 0 0 0 1
  1.61  0 0 0 0 0 1
  1.79  0 0 0 0 0 1
  1.88  0 0 0 0 0 1
  2.17  0 0 2 0 0 0
  2.18  0 0 0 0 0 1
  2.2   0 0 1 0 0 0
  2.41  0 0 0 0 0 1
  2.59  0 0 0 0 0 1
  2.65  0 0 0 0 0 1
  2.75  0 0 0 0 0 1
  2.8   1 0 0 0 0 0
  2.89  0 0 1 0 0 0
  2.94  0 0 0 0 0 1
  3     0 0 1 0 0 0
  3.01  0 0 1 0 0 0
  3.36  0 0 1 0 0 0
  3.37  0 0 1 0 0 0
  3.47  0 0 0 0 0 1
  3.5   0 0 1 0 0 0
  3.53  0 0 1 0 0 0
  3.54  0 0 1 0 0 0
  3.57  0 0 0 0 0 1
  3.62  1 0 0 0 0 0
  3.69  0 0 0 0 0 1
  3.72  0 0 0 0 0 1
  3.73  0 0 1 0 0 0
  3.87  0 0 0 0 0 1
  3.96  0 0 1 0 0 0
  3.97  0 0 0 0 0 1
  3.98  0 0 0 0 0 1
  4     0 0 0 0 0 1
  4.01  0 0 1 0 0 0
  4.09  0 0 0 0 0 1
  4.12  1 0 1 0 0 0
  4.33  0 0 1 0 0 0
  4.35  0 0 0 0 0 1
  4.38  0 0 0 0 0 1
  4.41  0 0 1 0 0 0
  4.42  0 0 1 0 0 0
  4.53  1 0 0 0 0 0
  4.61  0 0 0 0 0 1
  4.7   0 0 0 0 0 1
  4.74  0 0 1 0 0 0
  4.76  0 0 0 0 0 1
  4.8   0 0 1 0 0 0
  4.82  0 0 0 1 0 0
  4.83  0 0 0 0 0 1
  4.9   0 0 1 0 0 0
  4.99  0 0 1 0 0 0
  5.01  0 0 1 0 0 0
  5.05  0 0 0 0 0 2
  5.07  0 0 0 0 0 1
  5.08  0 0 0 0 0 1
  5.11  0 0 1 0 0 0
  5.17  0 0 1 0 0 1
  5.23  0 0 2 0 0 0
  5.24  0 0 1 0 0 0
  5.32  0 0 2 0 0 0
  5.33  0 0 0 0 0 1
  5.39  0 0 1 0 0 0
  5.4   0 0 1 0 0 0
  5.42  0 0 1 0 0 0
  5.43  0 0 0 0 0 1
  5.46  1 0 0 0 0 0
  5.74  0 0 0 0 0 1
  5.94  0 0 0 0 0 1
  5.96  0 0 0 0 0 1
  5.98  0 0 0 1 0 0
  6.08  0 0 1 0 0 0
  6.13  0 0 1 0 0 0
  6.3   1 0 0 0 0 0
  6.33  0 0 1 0 0 0
  6.34  0 0 0 0 0 1
  6.41  0 0 0 0 0 1
  6.42  1 0 0 0 0 0
  6.43  0 0 0 0 0 1
  6.57  0 0 0 0 0 1
  6.58  0 0 1 0 0 0
  6.63  0 0 1 0 0 0
  6.65  1 0 0 0 0 0
  6.74  0 0 1 0 0 0
  6.76  0 0 0 0 0 1
  7.02  0 0 0 0 0 1
  7.08  0 0 1 0 0 0
  7.16  0 0 1 0 0 0
  7.21  0 0 0 0 0 1
  7.36  0 0 0 0 0 1
  7.41  0 0 1 0 0 0
  7.51  0 0 1 0 0 0
  7.53  0 0 1 0 0 0
  7.6   0 0 1 0 0 0
  7.72  0 0 0 0 0 1
  7.75  0 0 0 0 0 1
  7.9   0 0 1 0 0 0
  8.06  1 0 0 0 0 0
  8.5   0 0 0 0 0 1
  8.51  0 0 0 0 0 2
  8.53  0 0 0 0 0 1
  8.66  0 0 1 0 0 0
  8.68  1 0 0 0 0 0
  8.88  0 0 1 0 0 0
  9.1   0 0 2 0 0 0
  9.13  0 1 0 0 0 0
  9.18  0 1 0 0 0 0
  9.33  0 0 0 0 0 1
  9.45  0 0 0 0 0 1
  9.79  0 0 0 0 0 1
  9.83  1 0 0 0 0 0
  9.84  0 0 0 0 0 1
  10.45 0 1 0 0 0 0
  10.66 0 0 1 0 0 0
  10.9  0 1 0 0 0 0
  11.21 0 0 0 0 0 1
  11.46 0 0 0 0 0 2
  11.47 0 0 0 0 0 1
  11.81 0 0 1 0 0 0
  11.82 0 1 0 0 0 0
  12.09 0 0 0 0 0 1
  12.7  0 0 0 0 0 1
  13.03 0 0 0 0 0 1
  13.28 0 0 0 0 0 1
  13.39 0 1 0 0 0 0
  13.57 0 0 0 0 0 1
  13.91 0 0 0 0 0 1
  14.19 0 0 0 0 0 1
  14.34 0 0 0 0 1 0
  14.4  0 1 0 0 0 0
  14.49 1 0 0 0 0 0
  14.73 1 0 0 0 0 0
  14.8  1 0 0 0 0 0
  15.22 0 0 1 0 0 0
  15.42 0 0 0 0 1 0
  15.73 0 0 0 0 1 0
  15.91 0 0 0 0 0 1
  16.2  0 1 0 0 1 0
  16.42 0 0 0 0 1 0
  16.82 0 1 0 0 0 0
  16.91 0 1 0 0 0 0
  17.95 0 0 0 0 1 0
  18.49 0 1 0 0 0 0
  19.25 0 0 0 0 1 0
  19.81 0 0 0 0 0 1
  20.9  0 1 0 0 0 0
  21.68 0 0 0 0 1 0
  22.26 0 0 0 0 1 0
  23    0 0 0 0 1 0
  23.01 0 0 0 0 1 0
  24.6  0 0 0 0 1 0
  24.72 0 0 0 0 1 0
  24.9  0 0 0 0 1 0
  25.76 0 0 0 0 1 0
  28.39 0 0 0 0 1 0
  30.5  0 1 0 0 0 0
  30.6  0 0 0 0 1 0
  33.56 0 0 0 0 1 0
  36    0 0 0 0 1 0
```

I wish to answer Question 1 Section B

what technique (diana or agned) did you use and why?

I used both method and I wanted to see what differences do they make.I did not have a better model in mind therefore I chose to run both models.

how many clusters did you accept?

I accpted 6 clusters as both models computed exact same number of clusters.

how many cases (rows) will be badly clustered?

For the Agn model, there are total of 14 bad clusters For the Diana model there are total of 13 bad clusters

```
```