6.0 Clustering Assignment

Data Setup

Europe employment dataset showing the percentage of people employed in nine industry sectors in Europe for the years 1989 to 1995.

Variables:

- AGR: Agriculture, forestry, and fishing
- MIN: Mining and quarrying
- MAN: Manufacturing
- PS: Power and water supplies
- CON: Construction
- SER: Services
- FIN: Finance
- SPS: Social and personal services
- TC: Transport and communications

1. Read in data

```
euro <- read.csv("https://bit.ly/3ktLWfr", header=TRUE, row.names=1)
head(euro)
```

	Group <chr>	AGR <dbl>	MIN <dbl>	MAN <dbl>	PS <dbl>	CON <dbl>	SER <dbl>	FIN <dbl>	SPS <dbl>
Belgium	EU	2.6	0.2	20.8	0.8	6.3	16.9	8.7	36.9
Denmark	EU	5.6	0.1	20.4	0.7	6.4	14.5	9.1	36.3
France	EU	5.1	0.3	20.2	0.9	7.1	16.7	10.2	33.1
Germany	EU	3.2	0.7	24.8	1.0	9.4	17.2	9.6	28.4
Greece	EU	22.2	0.5	19.2	1.0	6.8	18.2	5.3	19.8
Ireland	EU	13.8	0.6	19.8	1.2	7.1	17.8	8.4	25.5

6 rows | 1-10 of 11 columns

2. Remove outliers Albania and Giblartar as well as the non-numerical “Group” column:

```
# Remove outlier countries
euro.c <- euro[-c(19,28), ]

# Remove the first column
mydata = euro.c[, -1]
head(mydata)
```

	AGR <dbl>	MIN <dbl>	MAN <dbl>	PS <dbl>	CON <dbl>	SER <dbl>	FIN <dbl>	SPS <dbl>	TC <dbl>
Belgium	2.6	0.2	20.8	0.8	6.3	16.9	8.7	36.9	6.8
Denmark	5.6	0.1	20.4	0.7	6.4	14.5	9.1	36.3	7.0
France	5.1	0.3	20.2	0.9	7.1	16.7	10.2	33.1	6.4
Germany	3.2	0.7	24.8	1.0	9.4	17.2	9.6	28.4	5.6
Greece	22.2	0.5	19.2	1.0	6.8	18.2	5.3	19.8	6.9
Ireland	13.8	0.6	19.8	1.2	7.1	17.8	8.4	25.5	5.8

6 rows

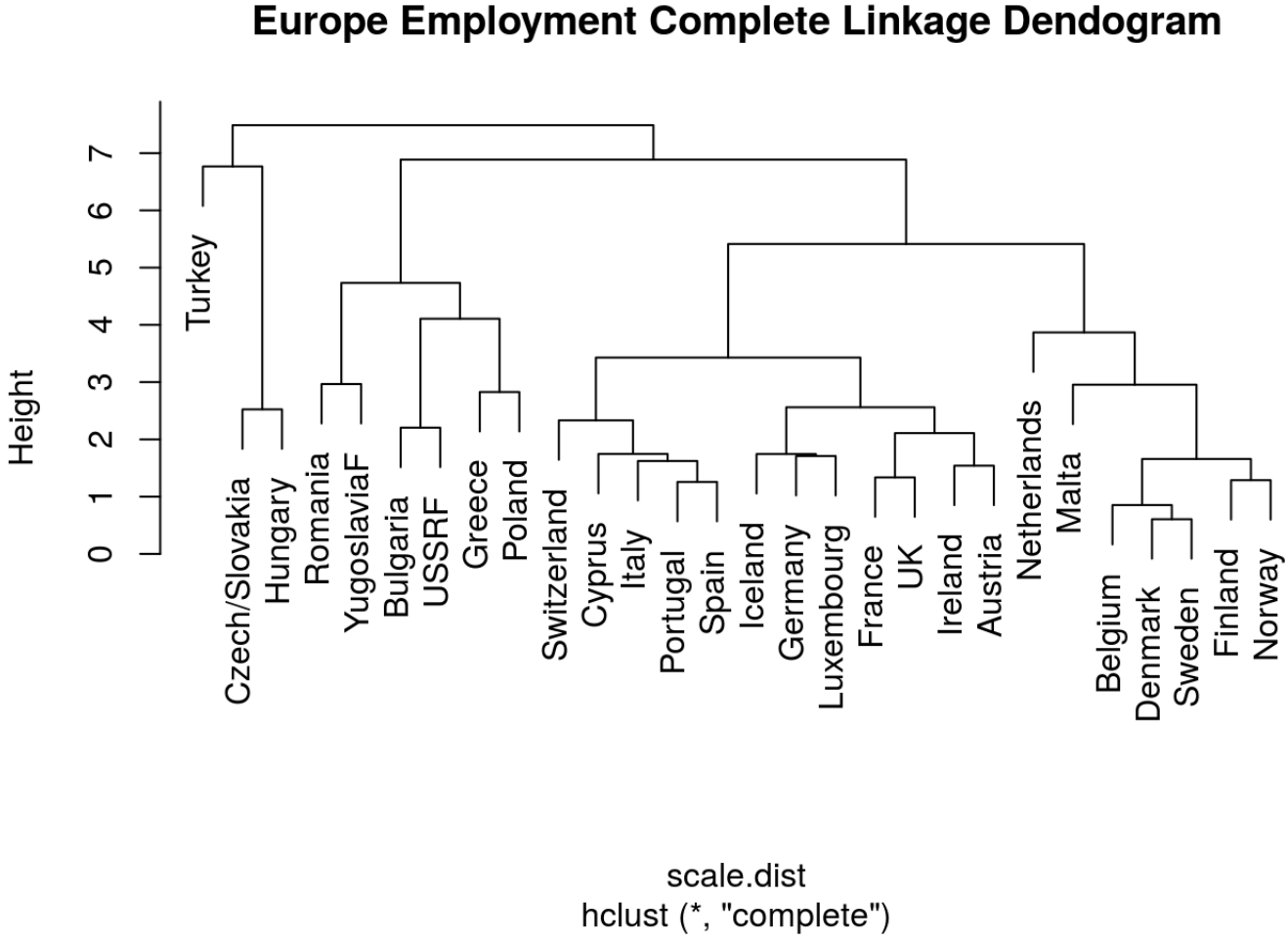
a) Create a hierarchical clustering dendrogram

Based on complete linkage (default).

This requires bothing scaling and creating a distance matrix of our original data which we can accomplish by wrapping mydata with the `scale()` and `dist()` functions.

We then perform the hierarchical clustering with `hclust()` and plot the dendogram:

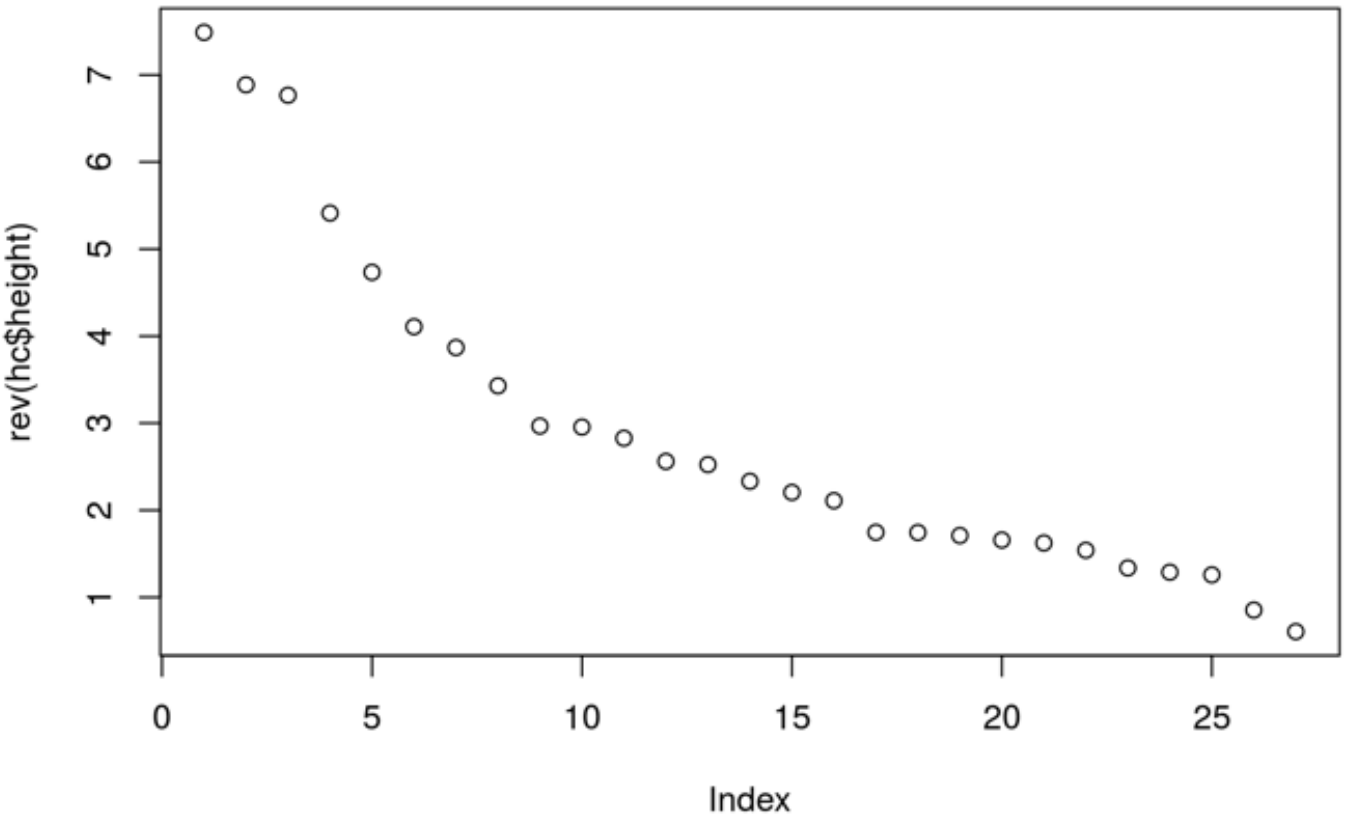
```
scale.dist = dist(scale(mydata))
hc <- hclust(scale.dist)
plot(hc, main = "Europe Employment Complete Linkage Dendogram")
```



b) Identify the appropriate number of clusters in Hierarchical clustering using a scree plot

To create this scree plot, we simply reverse the height of our clustering.Based on the drop-up point at 3, we will use that many clusters.

```
plot(rev(hc$height))
```



c) Based on your decision in part b, determine what countries are in which group?

Using our assumption of 3 clusters above, we can cut the clustering and see which countries land in the three groups.

1. Cut the clustering, display number of countires in each cluster:

```
ct <- cutree(hc, 3)
table(ct)
```

```
## ct
##  1  2  3
## 19  6  3
```

2. List the countries in cluster 1:

```
cluster1 = subset(rownames(mydata), ct==1)
cluster1
```

```
##  [1] "Belgium"     "Denmark"     "France"      "Germany"     "Ireland"
##  [6] "Italy"       "Luxembourg"  "Netherlands" "Portugal"    "Spain"
## [11] "UK"          "Austria"     "Finland"     "Iceland"     "Norway"
## [16] "Sweden"      "Switzerland" "Cyprus"      "Malta"
```

3. List the countries in cluster 2:

```
cluster2 = subset(rownames(mydata), ct==2)
cluster2
```

```
## [1] "Greece"      "Bulgaria"    "Poland"      "Romania"     "USSRF"
## [6] "YugoslaviaF"
```

4. List the countries in cluster 3:

```
cluster3 = subset(rownames(mydata), ct==3)
cluster3
```

```
## [1] "Czech/Slovakia" "Hungary"        "Turkey"
```

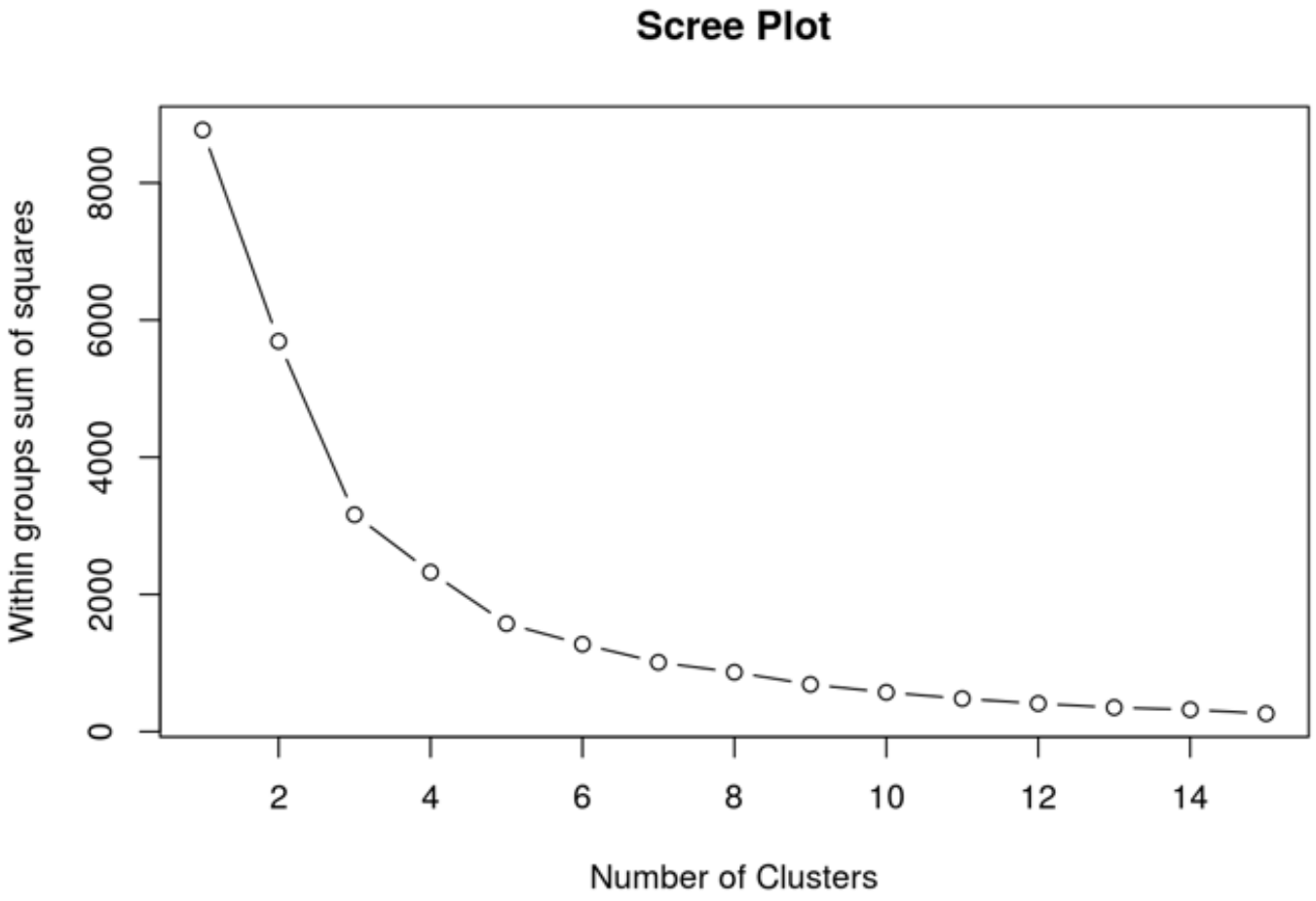
d) Identify the appropriate number of clusters in kmeans clustering based on the WGSS scree plot function.

Another method of identifying the best number of clusters is k-means clustering. For this we use the actual data instance of the distance matrix.

In this plot we limit the data to 15 points because additional clusters likely aren't helpful and make reading the elbow more difficult. For this analysis, we see that 4 clusters might actually be a better number to work with.

```
plot.wgss = function(mydata, maxc) {
  wss = numeric(maxc)
  for (i in 1:maxc)
    wss[i] = kmeans(mydata,centers=i, nstart = 10)$tot.withinss
  plot(1:maxc, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares", main="Scree Plot")
}

# Set max groups to 15 to see relevant clusters more easily
plot.wgss(mydata, 15)
```



e) Based on your decision in part e, perform k-means clustering and determine what countries are in which group?

With our number of clusters (centers), we can scale the data and perform the k-means clustering. The number of countries in each cluster is printed below along with the country groupings.

```
mydata.s <- scale(mydata)
km <- kmeans(mydata.s, centers = 4)
table(km$cluster)
```

```
##
##  1  2  3  4
##  8  7  2 11
```

```
km$cluster
```

```
##         Belgium         Denmark          France         Germany          Greece 
##               1               1               1               4               2 
##         Ireland           Italy      Luxembourg     Netherlands        Portugal 
##               4               4               4               1               4 
##           Spain              UK         Austria         Finland         Iceland 
##               4               4               4               1               4 
##          Norway          Sweden     Switzerland        Bulgaria Czech/Slovakia 
##               1               1               4               2               3 
##         Hungary          Poland         Romania           USSRF     YugoslaviaF 
##               3               2               2               2               2 
##          Cyprus           Malta          Turkey 
##               4               1               2
```

f) Attempt to identify the meanings of the clusters you found in part e by finding and interpreting the cluster centroids.

By pulling the centers from our k-means analysis, we can see some patterns in the centroids:

1. Cluster 1 contains countries that have heavy employment in construction, services, and finance with low employment elsewhere
2. Cluster 2 represents countries that have extremely high mining employment and some employment in transport. They have particularly low manufacturing employment.
3. Cluster 3 countries countries have relatively low employment across the board, with some representation in finance and social services
4. Cluster 4 countries are almost entirely employed in agriculture and manufacturing industries.

```
km$centers
```

```
##          AGR        MIN        MAN         PS         CON         SER
## 1 -0.6857861 -0.3058799 -0.1547026  0.3046104 -0.86656250 -0.08058308
## 2  1.1921536 -0.1805872  0.8141802  0.1969199 -0.16220498 -1.00290033
## 3  0.3200335  3.4907997 -2.5230363 -1.3538241  0.02820956 -0.89598812
## 4 -0.3180776 -0.2973137  0.0531302 -0.1006977  0.72831962  0.85972211
##          FIN        SPS          TC
## 1  0.6632947  1.2886558  0.47722126
## 2 -1.1123092 -1.0689538  0.08239791
## 3 -1.4707416 -0.3596020  1.17417028
## 4  0.4928445 -0.1915787 -0.61299055
```

g) Attempt to identify the meanings of the clusters you found in part f

Plot different pairs of principal component scores, the (PC1,PC2), (PC1,PC3), and (PC2,PC3) scatterplots, with points labeled (or colored) according to the assigned cluster. You can look at the loading of the first three PCs to find a meaning for each PC.

To try to interpret these meanings, we can use Principal Component Analysis and examine the first few loadings. The first two components appear very high in traditional economic employment, like agriculture or manufacturing.

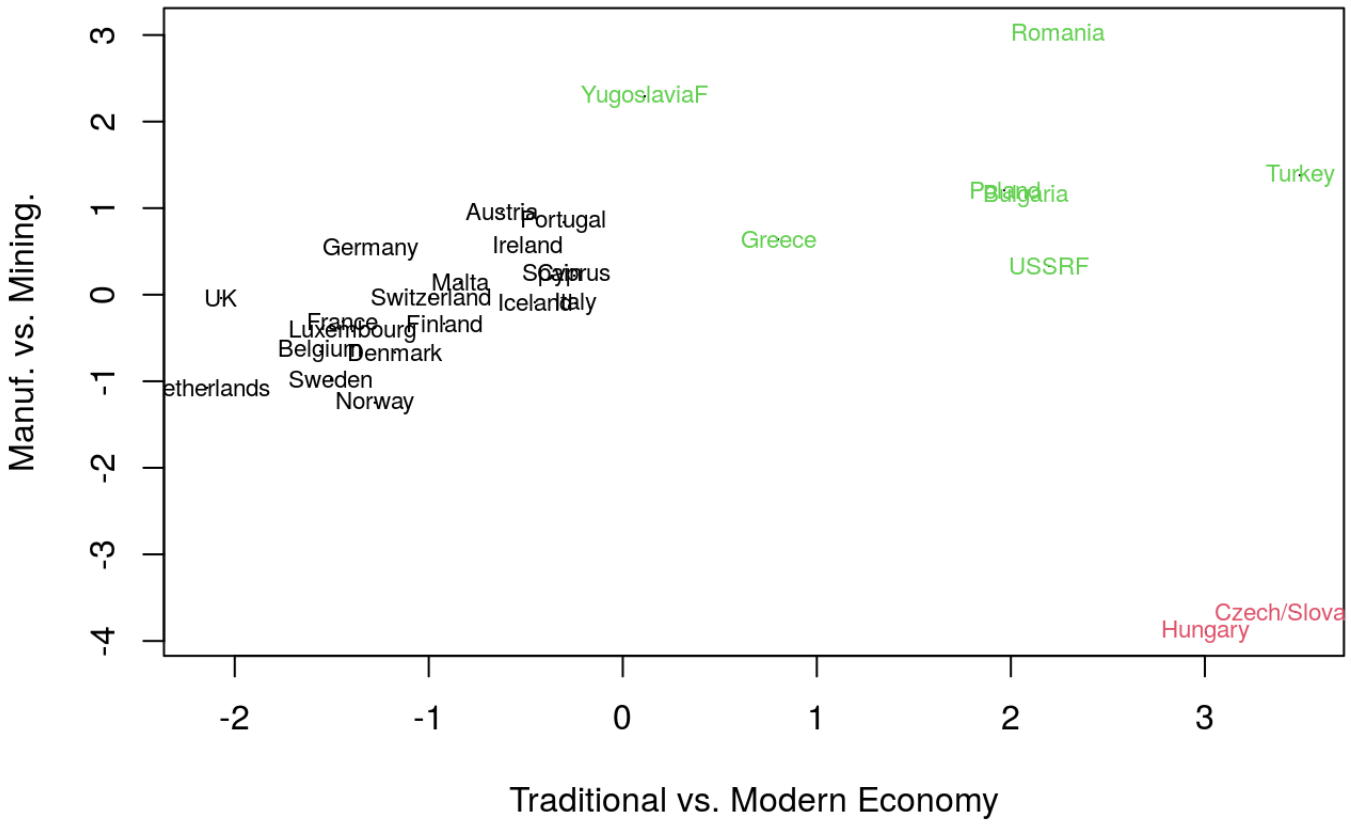
There appears to be some specialization within those traditional employment countries, such as component 2 being high in manufacturing as opposed to component 1. Component 3 reads high in what we'd consider "modern" industries, like services and power supplies.

```
pca <- princomp(mydata, cor = T)
pca$loadings[,1:3]
```

```
##          Comp.1      Comp.2      Comp.3
## AGR  0.48364840  0.21179179  0.15321979
## MIN  0.34260148 -0.49066649 -0.03803186
## MAN -0.09502752  0.61846227 -0.20394526
## PS  -0.20066474  0.36413697 -0.38254381
## CON  0.02671019  0.04530866  0.39543849
## SER -0.38420771 -0.06921631  0.48893685
## FIN -0.53994266 -0.08089840  0.15245056
## SPS -0.39848393 -0.33498839 -0.29358673
## TC   0.02748388 -0.27146598 -0.53129903
```

These clusters become more apparently if we plot the k-means with country labels and cluster colors:

```
km <- kmeans(scale(mydata), 3, nstart = 20)
plot(pca$scores[,c(1,2)], pch = ".", xlab = "Traditional vs. Modern Economy", ylab =
"Manuf. vs. Mining.")
text(pca$scores[,c(1,2)], label = rownames(mydata), cex = 0.7, col = km$cluster)
```



h) Perform model-based clustering without identifying the number of clusters

Plot the result of classification. How many groups are identified in your data? Determine what countries are in which group?

First, the “Mclust” module is needed so we install and load it. Then we can run Mclust on our dataset and plot the classifications. This classification identifies two groups, the countries in these groups are identified below.

```
install.packages("mclust")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(mclust)
```

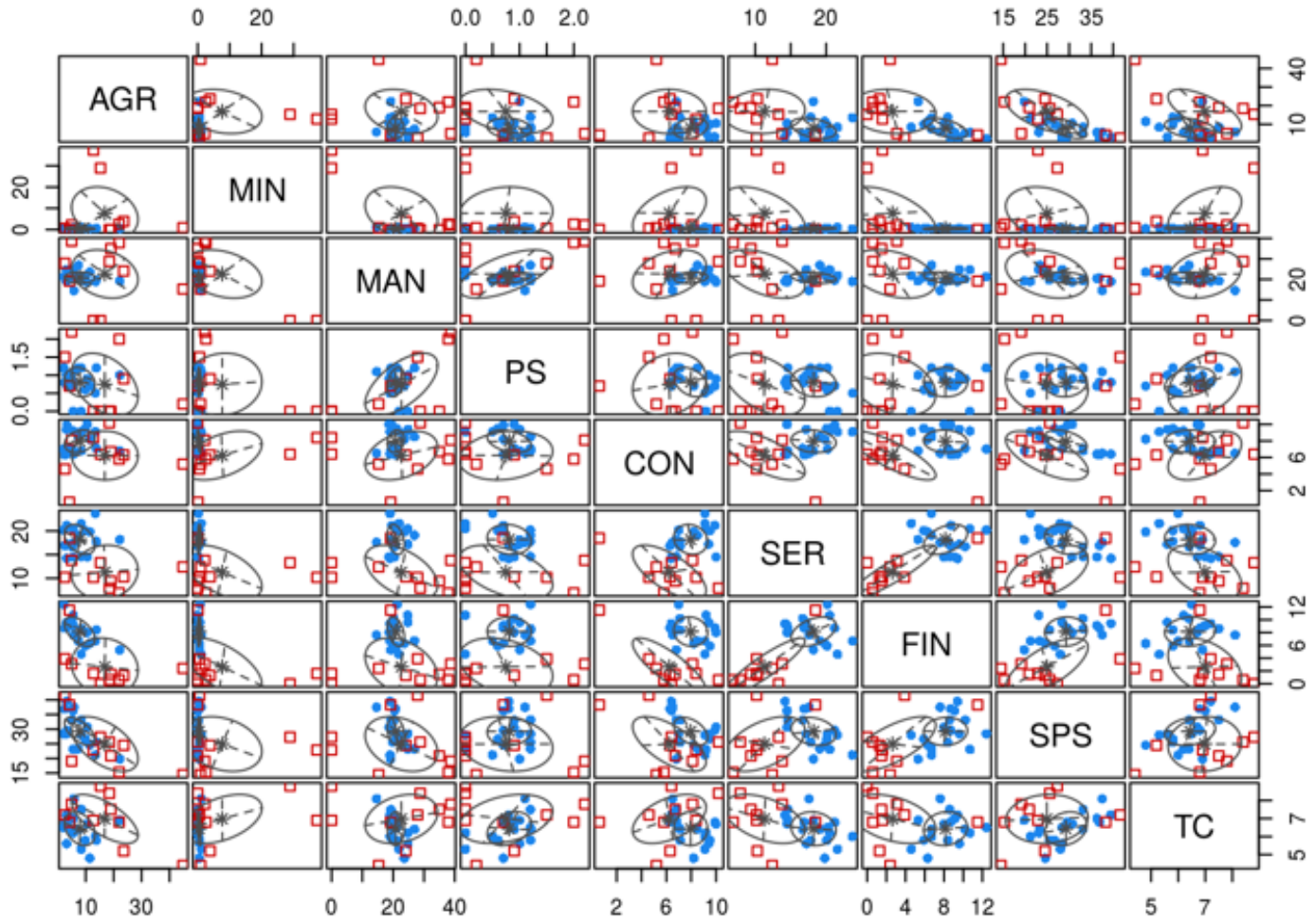
```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
mc <- Mclust(mydata)
mc$classification
```

```
##        Belgium        Denmark         France        Germany         Greece
##              1              1              1              1              1
##        Ireland          Italy     Luxembourg    Netherlands       Portugal
##              1              1              1              2              1
##          Spain             UK        Austria        Finland        Iceland
##              1              1              1              1              1
##         Norway         Sweden    Switzerland       Bulgaria Czech/Slovakia
##              1              1              1              2              2
##        Hungary         Poland        Romania          USSRF    YugoslaviaF
##              2              2              2              2              2
##         Cyprus          Malta         Turkey
##              1              2              2
```

Here we can see the color-coded plots of the clusters for each industry variable.

```
plot(mc, what = "classification")
```

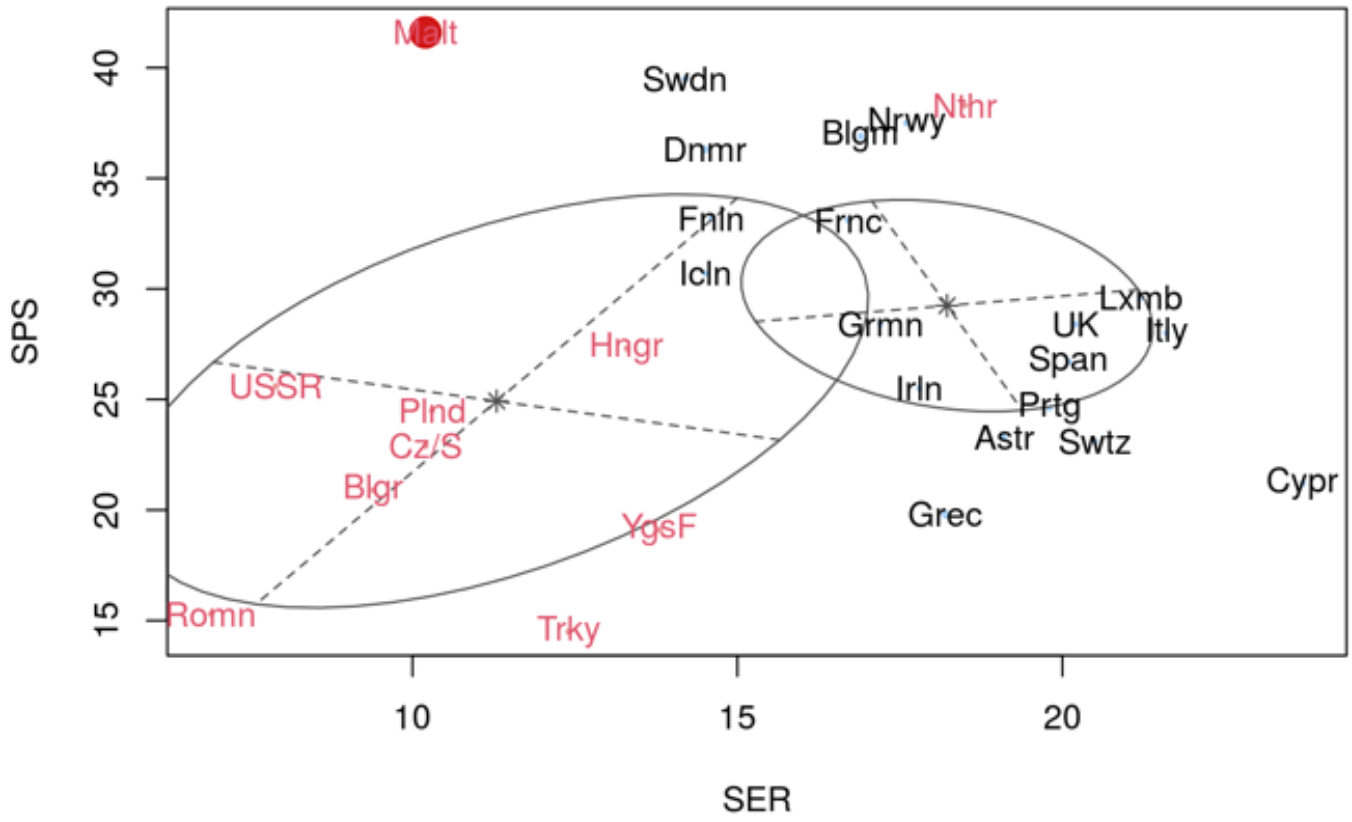


i) Use “plot” on your fitted mclust object, and report the “uncertainty” plot for variables (SER, SPS)

Explain the grouping of what country is more uncertain with what probability of uncertainty

We can plot the uncertainty of the countries by specifying `what = uncertainty` as a plot parameter, and specifying the dimensions for the two variables of interest:

```
plot(mc, what = "uncertainty", dimens = c(6,8))
text(mc$data[,c(6,8)], labels = abbreviate(rownames(mydata)), col = mc$classification
)
```



To find the most uncertain country, we can create a dataframe of the rows, classification, and uncertainty, then order by uncertainty. By doing this we see that Malta is the most uncertain between these variables with a probability of 0.045.

```
clust.data = cbind(rownames(mydata), mc$classification, mc$uncertainty)
clust.data[order(mc$uncertainty),]
```

```
##                [,1]             [,2] [,3]
## Germany        "Germany"        "1"  "0"
## Italy          "Italy"          "1"  "0"
## Luxembourg     "Luxembourg"     "1"  "0"
## Portugal       "Portugal"       "1"  "0"
## Austria        "Austria"        "1"  "0"
## Czech/Slovakia "Czech/Slovakia" "2"  "0"
## Hungary        "Hungary"        "2"  "0"
## Romania        "Romania"        "2"  "0"
## YugoslaviaF    "YugoslaviaF"    "2"  "0"
## Bulgaria       "Bulgaria"       "2"  "1.11022302462516e-15"
## Poland         "Poland"         "2"  "3.99680288865056e-15"
## Netherlands    "Netherlands"    "2"  "5.77315972805081e-15"
## Turkey         "Turkey"         "2"  "1.04893871366585e-12"
## Cyprus         "Cyprus"         "1"  "7.89013299140606e-12"
## UK             "UK"             "1"  "2.27684537890127e-11"
## Iceland        "Iceland"        "1"  "1.48895118456949e-10"
## Spain          "Spain"          "1"  "1.28949110278498e-08"
## Finland        "Finland"        "1"  "1.34680809393828e-07"
## France         "France"         "1"  "1.48729669469105e-07"
## Switzerland    "Switzerland"    "1"  "2.29876298885046e-07"
## USSRF          "USSRF"          "2"  "1.89374313774859e-06"
## Ireland        "Ireland"        "1"  "3.1970088886224e-06"
## Norway         "Norway"         "1"  "4.03895216360972e-06"
## Sweden         "Sweden"         "1"  "3.0297814936131e-05"
## Belgium        "Belgium"        "1"  "3.82251565148595e-05"
## Denmark        "Denmark"        "1"  "0.000125426603032497"
## Greece         "Greece"         "1"  "0.000284492832784422"
## Malta          "Malta"          "2"  "0.0454836912519381"
```

j) Construct the appropriate contingency table between the given grouping in the original cleaned data (euro.c$Group) and the groups we found in the model-based clustering.

Interpret the table, and explain how well do the model-based clusters correspond?

Looking at the contingency table, we see there is a significant disconnect between the original EU groups and the model classifications. Namely, the model classified way too many countries in the Eastern categories, and very few in the EU category.

```
table(euro.c$Group, mc$classification)
```

```
##
##             1  2
##   Eastern   0  7
##   EFTA      6  0
##   EU       11  1
##   Other     1  2
```