Assignment #8: Cluster Analysis (0 points)

Data: The data for this assignment is the European employment data set. This data will be made available by your instructor.

Data Description: Employment in various industry segments reported as a percent for thirty European nations. Note that EU stands for European Union, EFTA stands for European Free Trade Association, and Eastern stand for Eastern European nations or the former Eastern Block.

For convenience here are the definitions of the abbreviated industries.

AGR:	agriculture
MIN:	mining
MAN:	manufacturing
PS:	power and water supply
CON:	construction
SER:	services
FIN:	finance
SPS:	social and personal services
TC:	transport and communications

Assignment Instructions:

In this assignment we will learn how to perform an exploratory data analysis for a clustering problem, fit a hierarchical cluster analysis, fit a k-means cluster analysis, how to integrate principal components analysis and cluster analysis, how to use cluster analysis as a predictive model, and how to make a variety of R graphics applicable to cluster analysis and multivariate analysis in general.

Part 1: The Data

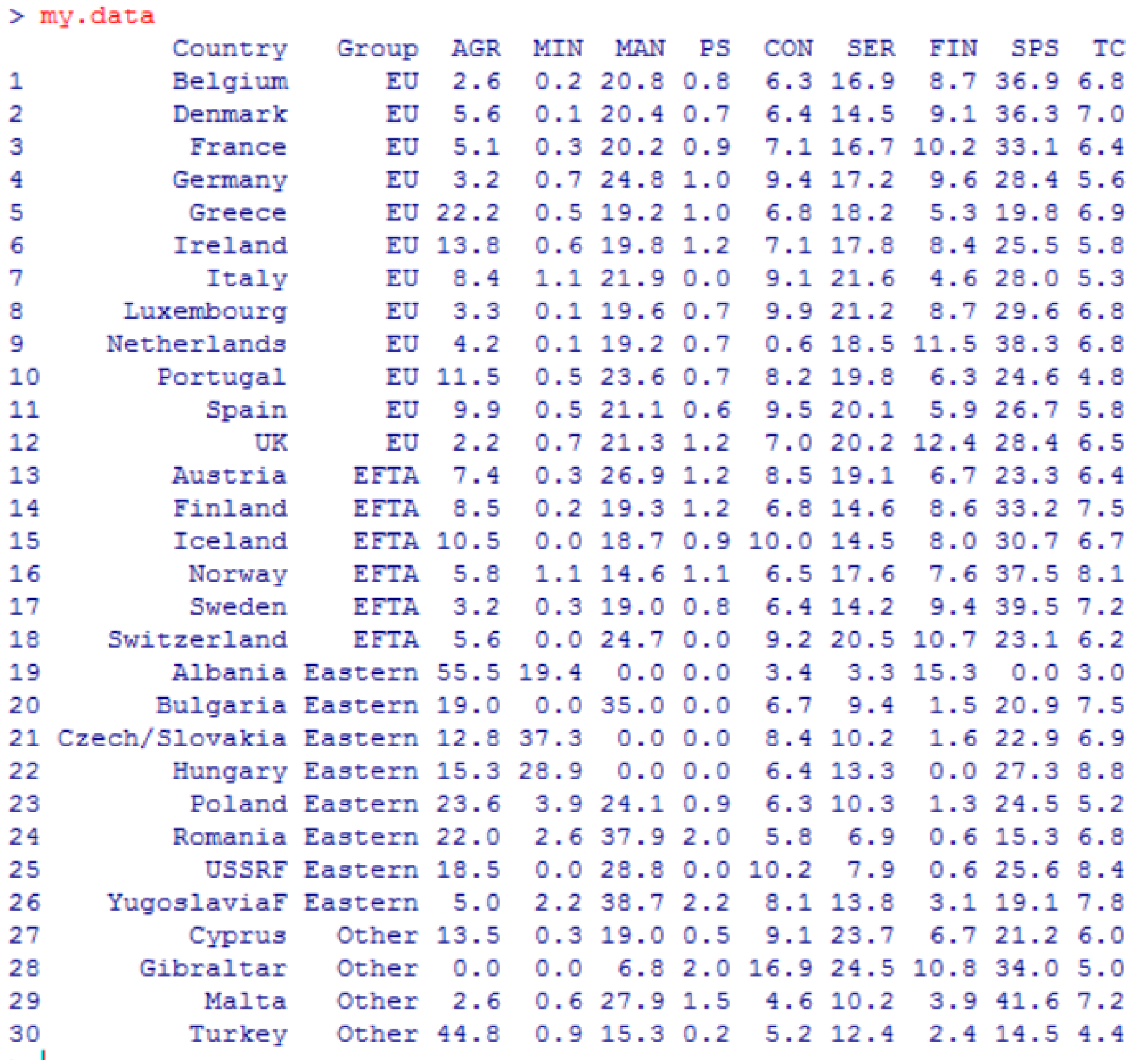
Let's begin by reading in the data.

```
# Change my.path to point to your file;
my.path <- ‘Specify your file path';
my.file <- paste(my.path,'EuropeanEmployment.csv',sep='');
my.data <- read.csv(my.file,header=TRUE);

str(my.data)
head(my.data)
```

If we look at the data, then we will see that we have multivariate data in which the observations are assigned to classes, or are said to have *labels* (EU, EFTA, Eastern, or Other). With this type of data we could perform a segmentation, an *unsupervised learning* problem where we assign the countries to different groups based on their similarity as determined by a clustering algorithm, or we could define a classification problem (see Chapter 11 in Applied Multivariate Data Analysis) and use a clustering algorithm as a *supervised learning* algorithm that would predict the class/label of each observation based on its cluster assignment.

Since the data set is small enough, we can easily view and comprehend the entire data set by simply printing it out. Take a look. What do we have?



```
> my.data
          Country   Group  AGR  MIN  MAN  PS  CON  SER  FIN  SPS  TC
1         Belgium      EU  2.6  0.2 20.8 0.8  6.3 16.9  8.7 36.9 6.8
2         Denmark      EU  5.6  0.1 20.4 0.7  6.4 14.5  9.1 36.3 7.0
3          France      EU  5.1  0.3 20.2 0.9  7.1 16.7 10.2 33.1 6.4
4         Germany      EU  3.2  0.7 24.8 1.0  9.4 17.2  9.6 28.4 5.6
5          Greece      EU 22.2  0.5 19.2 1.0  6.8 18.2  5.3 19.8 6.9
6         Ireland      EU 13.8  0.6 19.8 1.2  7.1 17.8  8.4 25.5 5.8
7           Italy      EU  8.4  1.1 21.9 0.0  9.1 21.6  4.6 28.0 5.3
8      Luxembourg      EU  3.3  0.1 19.6 0.7  9.9 21.2  8.7 29.6 6.8
9     Netherlands      EU  4.2  0.1 19.2 0.7  0.6 18.5 11.5 38.3 6.8
10       Portugal      EU 11.5  0.5 23.6 0.7  8.2 19.8  6.3 24.6 4.8
11          Spain      EU  9.9  0.5 21.1 0.6  9.5 20.1  5.9 26.7 5.8
12             UK      EU  2.2  0.7 21.3 1.2  7.0 20.2 12.4 28.4 6.5
13        Austria    EFTA  7.4  0.3 26.9 1.2  8.5 19.1  6.7 23.3 6.4
14        Finland    EFTA  8.5  0.2 19.3 1.2  6.8 14.6  8.6 33.2 7.5
15        Iceland    EFTA 10.5  0.0 18.7 0.9 10.0 14.5  8.0 30.7 6.7
16         Norway    EFTA  5.8  1.1 14.6 1.1  6.5 17.6  7.6 37.5 8.1
17         Sweden    EFTA  3.2  0.3 19.0 0.8  6.4 14.2  9.4 39.5 7.2
18    Switzerland    EFTA  5.6  0.0 24.7 0.0  9.2 20.5 10.7 23.1 6.2
19        Albania Eastern 55.5 19.4  0.0 0.0  3.4  3.3 15.3  0.0 3.0
20       Bulgaria Eastern 19.0  0.0 35.0 0.0  6.7  9.4  1.5 20.9 7.5
21 Czech/Slovakia Eastern 12.8 37.3  0.0 0.0  8.4 10.2  1.6 22.9 6.9
22        Hungary Eastern 15.3 28.9  0.0 0.0  6.4 13.3  0.0 27.3 8.8
23         Poland Eastern 23.6  3.9 24.1 0.9  6.3 10.3  1.3 24.5 5.2
24        Romania Eastern 22.0  2.6 37.9 2.0  5.8  6.9  0.6 15.3 6.8
25          USSRF Eastern 18.5  0.0 28.8 0.0 10.2  7.9  0.6 25.6 8.4
26     YugoslaviaF Eastern  5.0  2.2 38.7 2.2  8.1 13.8  3.1 19.1 7.8
27         Cyprus   Other 13.5  0.3 19.0 0.5  9.1 23.7  6.7 21.2 6.0
28      Gibraltar   Other  0.0  0.0  6.8 2.0 16.9 24.5 10.8 34.0 5.0
29          Malta   Other  2.6  0.6 27.9 1.5  4.6 10.2  3.9 41.6 7.2
30         Turkey   Other 44.8  0.9 15.3 0.2  5.2 12.4  2.4 14.5 4.4
```

Part 2: Initial Exploratory Data Analysis

Since we have a relatively small number of variables, we will begin our exploratory data analysis with a pairwise scatterplot. Note that when you have a small number of variables, the pairwise scatterplot is a useful statistical graphic. However, when you have a large number of variables, then the pairwise scatterplot is not useful. Typically the individual plots become too small to be of any use when we have more than twelve variables.

Another note about scatterplots – they are not very useful when we have too many data points. A scatterplot is a more useful statistical graphic when you have 100 data points than when you have 1MM data points.

Since we are interested in applying cluster analysis to this data, we can use the pairs plot to scan the individual 2-dimensional views of the data. In cluster analysis we typically focus on 2D and 3D representations of the data in order to avoid the *curse of dimensionality*. With multivariate data as the dimension grows the distance between the observations grow, and it is difficult for the observations to be ‘close’ to one another, and hence be grouped into a small number of clusters.

```
# Pairwise scatterplot
pairs(my.data[,-c(2)])
```

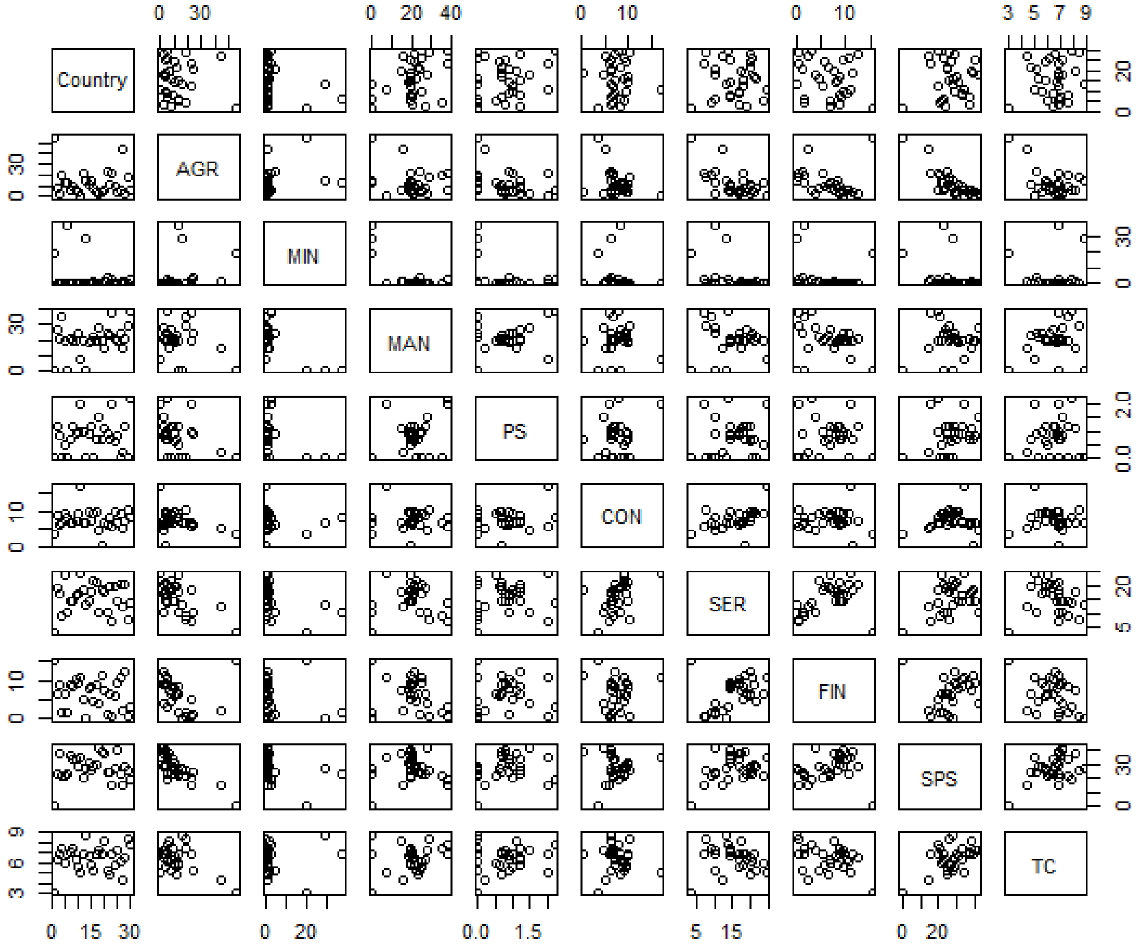


Figure 2.1 Pairwise Scatterplot

Do you see any interesting 2D views of the data? What would be ‘interesting’? Remember, we are interesting in applying cluster analysis so 2D plots that show clusters are the plots that would be interesting. Why don’t we consider MAN versus SER and SER versus FIN? Do these 2D views look interesting?

Part 3: Visualizing the Data with Labelled Scatterplots

While the pairs plot allows us to scan all of the pairwise scatterplots easily and efficiently, it is not the ideal visualization of the data. After we have honed in on some interesting dimensions we can create more specialized plots for those dimensions. Specialized plots should always include labels and color. The objective is to compress more than two dimensions of information into a two dimensional plot.

Let’s begin by plotting FIN versus SER. Note that in order to make this plot we will need to separate our data set into four data sets, one for each class. If an R function operates on a data frame, then it is designed to operate on the entire data frame. We could embed the `subset()` function in the other function calls, but in our case it will be easier and cleaner to subset the data outright.

```
eu.df <- subset(my.data,Group=='EU');
efta.df <- subset(my.data,Group=='EFTA');
eastern.df <- subset(my.data,Group=='Eastern');
other.df <- subset(my.data,Group=='Other');

# Plot of FIN versus SER;
plot(my.data$SER,my.data$FIN,xlab='Services',ylab='Finance',xlim=c(0,27),ylim=c(0,17))
text(eu.df$SER,eu.df$FIN,labels=eu.df$Country,cex=0.75,pos=4,col='green')
text(efta.df$SER,efta.df$FIN,labels=efta.df$Country,cex=0.75,pos=4,col='blue')
text(eastern.df$SER,eastern.df$FIN,labels=eastern.df$Country,cex=0.75,pos=4,col='red')
text(other.df$SER,other.df$FIN,labels=other.df$Country,cex=0.75,pos=4,col='grey')
```

The `text()` function allows us to add the labels to the plot. Can we figure out what the other graphical parameters are doing? Hint: The R help page can be found using `help(par)`.

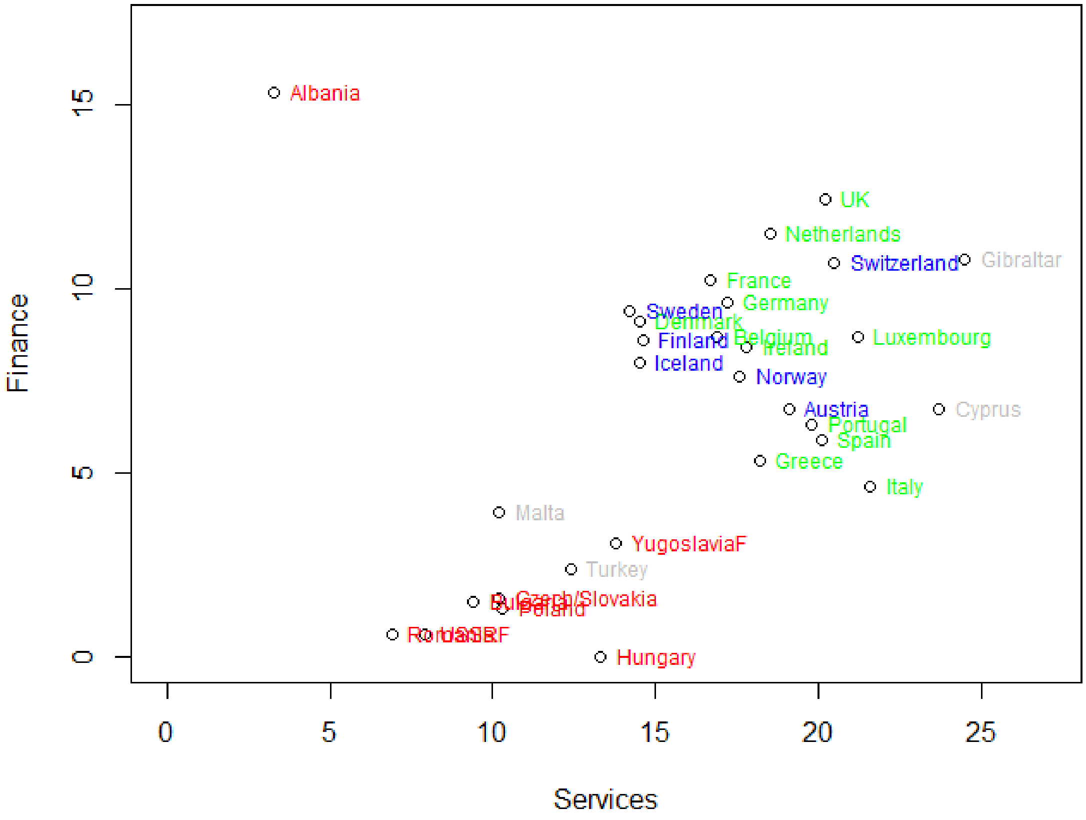


Figure 3.1 Labelled Scatterplot of Finance vs Services

Do we see some clusters in this plot? How many clusters do we have? How many clusters would you have if you were creating a segmentation? How many clusters would you need if you wanted to classify the RED, BLUE, and GREEN points accurately? Which countries do our 'others' belong with?

Now let's make the same plot for MAN versus SER.

```
# Plot MAN versus SER;
plot(my.data$MAN,my.data$FIN,xlab='Manufacturing',ylab='Finance',xlim=c(0,32),ylim=c(0,17))
text(eu.df$MAN,eu.df$FIN,labels=eu.df$Country,cex=0.75,pos=4,col='green')
text(efta.df$MAN,efta.df$FIN,labels=efta.df$Country,cex=0.75,pos=4,col='blue')
text(eastern.df$MAN,eastern.df$FIN,labels=eastern.df$Country,cex=0.75,pos=4,col='red')
text(other.df$MAN,other.df$FIN,labels=other.df$Country,cex=0.75,pos=4,col='grey')
```

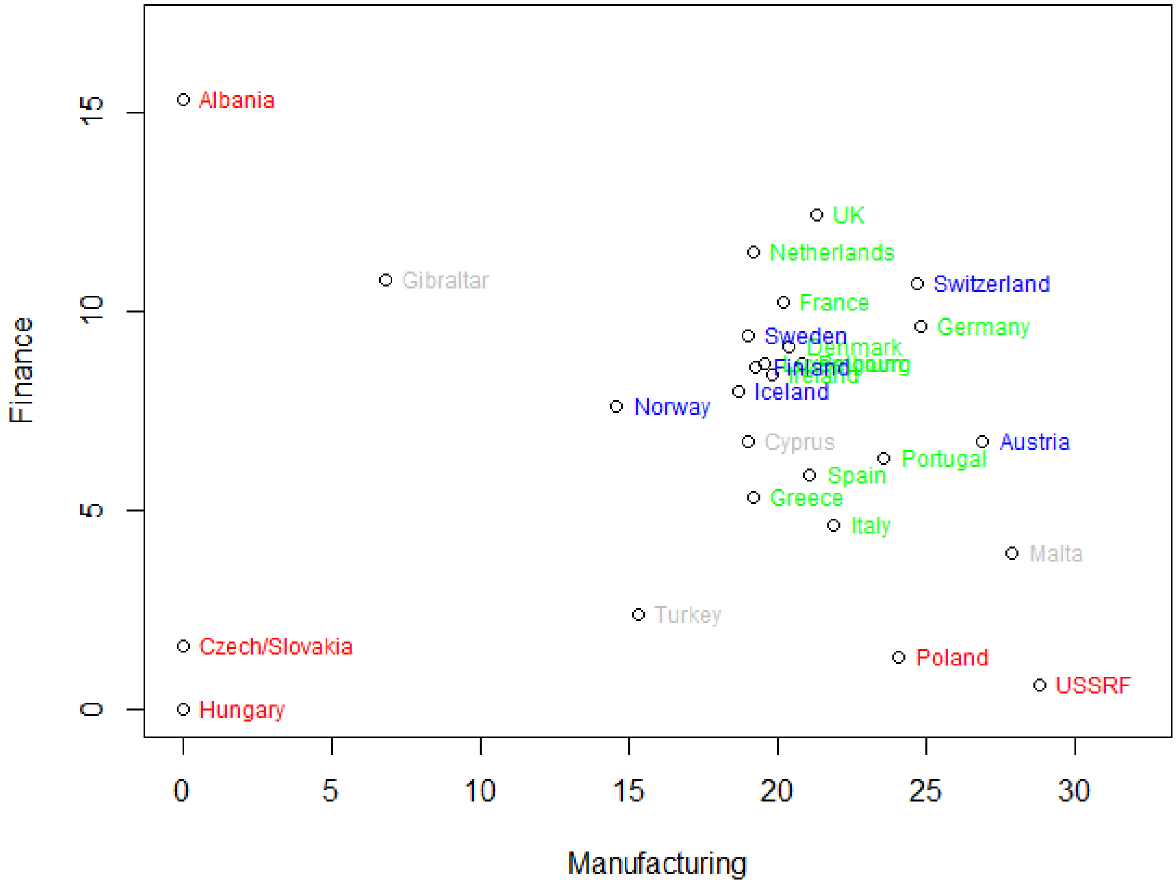


Figure 3.2 Labelled Scatterplot of Finance vs Manufacturing

Do we see some clusters in this plot? How many clusters do we have? Are they the same clusters as we saw in the previous plot? How many clusters would you have if you were creating a segmentation? How many clusters would you need if you wanted to classify the RED, BLUE, and GREEN points accurately? Which countries do our 'others' belong with?

Of the two 2D views of the data which one do you think would be the better view for supervised clustering, i.e. using a clustering algorithm to create a classifier that will assign the countries to the correct class/label? Why?

Part 4: Creating a 2D Projection Using Principal Components Analysis

We can use principal components analysis to reduce the dimension of the data. We can project the data down from 9D to 2D by performing PCA and using the first and second principal components. By doing so we are creating a new 2D view of the data, and a view of the data that contains information from more than two dimensions.

Note that in this application we will not standardize this data to be mean zero with unit variance before we perform the PCA. In general we almost always want to standardize our data before we perform PCA to keep the variables with the largest scales from getting the largest loadings. Remember – large scale means large variance. However, in this case we have a type of data called *compositional data*. Compositional data represent the components of a whole. In our case the dimensions sum to 100, and each dimension represents a component of the economy. We are not standardizing the data since large components in some dimensions will require small components in other dimensions in order to sum to 100. This natural constraint creates the natural separations that we have seen in Part 3.

The nature of compositional data can cause a variety of problems. Most statistical methods are designed for continuous data, and the question is how 'continuous' is our compositional data. In practice we would run this analysis both ways. We would run the PCA as is, and we would run the PCA on the standardized data, and then we would compare the results. In particular we would compare the results in our final application, which in this assignment would be the cluster analysis.

```
apply(my.data[,-c(1,2)],MARGIN=1,FUN=sum)
pca.out <- princomp(x=my.data[,-c(1,2)],cor=FALSE);
names(pca.out)

pc.1 <- pca.out$scores[,1];
pc.2 <- pca.out$scores[,2];

my.pca <- as.data.frame(list(Country=my.data$Country,Group=my.data$Group,pc1=pc.1,pc2=pc.2));
# Do we know why I used list() instead of cbind()?;

eu.pca <- subset(my.pca,Group=='EU');
efta.pca <- subset(my.pca,Group=='EFTA');
eastern.pca <- subset(my.pca,Group=='Eastern');
other.pca <- subset(my.pca,Group=='Other');


plot(eu.pca$pc1,eu.pca$pc2,xlab='Principal Component 1',ylab='Principal Component 2',
xlim=c(-60,25),ylim=c(-25,30))points(efta.pca$pc1,efta.pca$pc2)
points(eastern.pca$pc1,eastern.pca$pc2)
points(other.pca$pc1,other.pca$pc2)
text(eu.pca$pc1,eu.pca$pc2,labels=eu.pca$Country,cex=0.75,col='green',pos=4)
text(efta.pca$pc1,efta.pca$pc2,labels=efta.pca$Country,cex=0.75,col='blue',pos=1)
text(eastern.pca$pc1,eastern.pca$pc2,labels=eastern.pca$Country,cex=0.75,col='red',pos=1)
text(other.pca$pc1,other.pca$pc2,labels=other.pca$Country,cex=0.75,col='grey',pos=3)
```

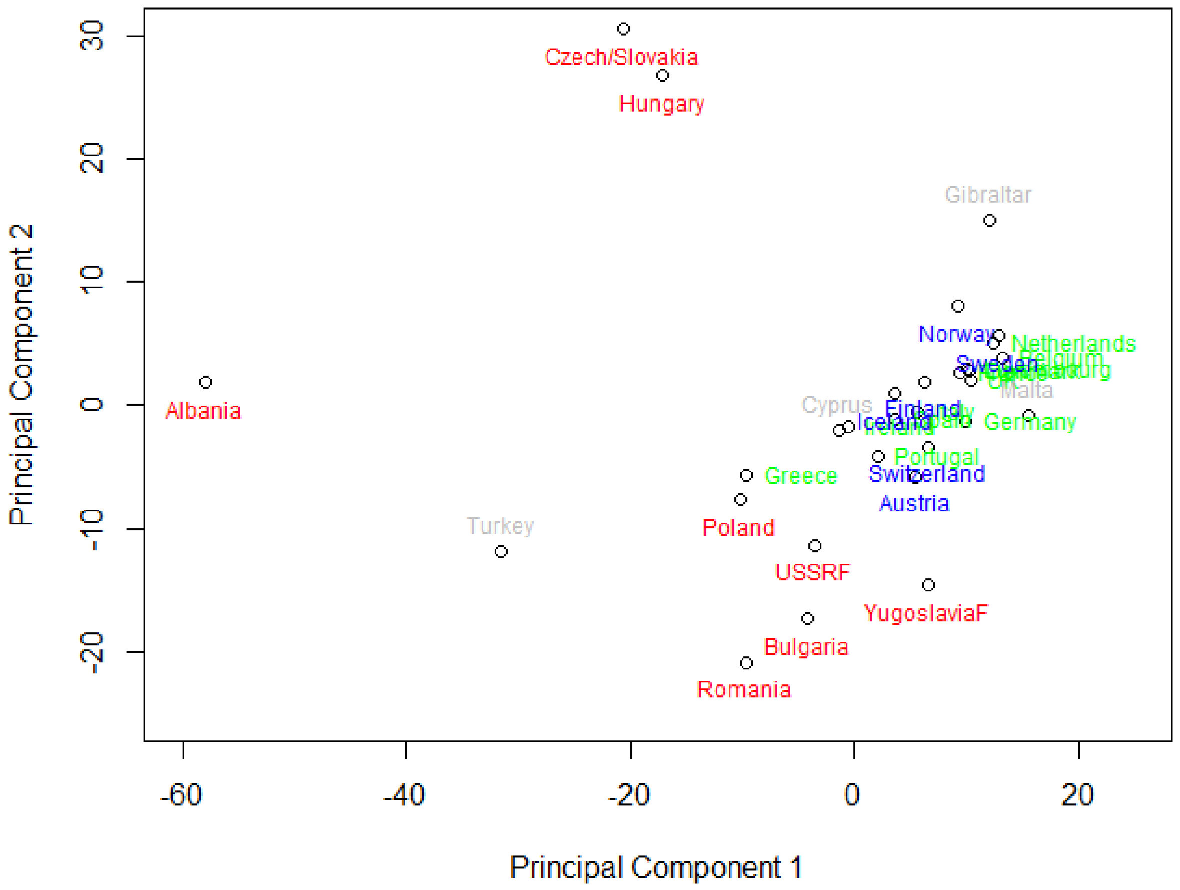


Figure 4.1 Labelled Scatterplot of PC1 vs PC2

How does this 2D projection of the data compare to the two other views of the data that we are considering? How many clusters does this 2D projection have? Clearly, our data can have different degrees of separation in different 2D profiles, and hence some low dimension representations will be better clustered than others.

If you are interested in examining how standardizing the data would change this view of the data, then simply rerun the code but set `cor=TRUE` in the `princomp()` function.

Part 5: Hierarchical Clustering Analysis

The first cluster analysis that we perform on this data will use hierarchical clustering. In the previous exploratory data analysis of the data we kept the ‘Other’ category on the data. Since the derived clusters are affect by all included data points, especially outlier data points, we will remove the ‘Other’ observations from the data so that the clustering algorithms will only use the proper data when creating the clusters.

```
# Drop the 'Other' Category;
label.data <- subset(my.data,Group != 'Other');

# Cluster in FIN*SER view;
fin.ser <- hclust(d=dist(label.data[,c('FIN','SER')]),method='complete');
plot(fin.ser,labels=label.data[,1],xlab='Hierarchical Clustering FIN vs SER 2D View',sub='');
```

We will begin by clustering in the FIN*SER 2D view of the data. Hierarchical clustering is performed in R by using the R function `hclust()`. Hierarchical clustering algorithms fit a tree of clusters from k=2 to k=N, where N is the number of data points in the sample. This tree of clusters can be visualized using a dendrogram, and all software programs that have a hierarchical clustering algorithm should produce a dendrogram. When the data is small enough, then dendograms are useful for visualizing the tree of clusters. However, like many statistical graphics, when the data gets large (large N) the tree, and hence the dendogram, becomes too large to be an effective display of the clusters.

Since the cluster tree stores all possible cluster assignments, we must cut the tree using `cutree()` to force an assignment of the observations to a particular number of clusters. Let’s cut the tree to k=3 and k=6 and compare the classification accuracy of two cluster tree cuts.

```
fin.ser.3 <- cutree(fin.ser,k=3);
finser.3df <- as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=fin.ser.3))
finser.t3 <- table(finser.3df$Group,finser.3df$Cluster)
finser.comp3 <- t(finser.t3[1:3,])*(1/apply(finser.t3[1:3,],FUN=sum,MARGIN=2))
finser.accuracy3 <-
sum(apply(finser.t3[1:3,],FUN=max,MARGIN=2))/sum(apply(finser.t3[1:3,],FUN=sum,MARGIN=2));


fin.ser.6 <- cutree(fin.ser,k=6);
finser.6df <- as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=fin.ser.6))
finser.t6 <- table(finser.6df$Group,finser.6df$Cluster)
finser.comp6 <- t(finser.t6[1:3,])*(1/apply(finser.t6[1:3,],FUN=sum,MARGIN=2))
finser.accuracy6 <-
sum(apply(finser.t6[1:3,],FUN=max,MARGIN=2))/sum(apply(finser.t6[1:3,],FUN=sum,MARGIN=2));
```

How are we comparing the cluster accuracy in this R code? Which set of clusters is more accurate?

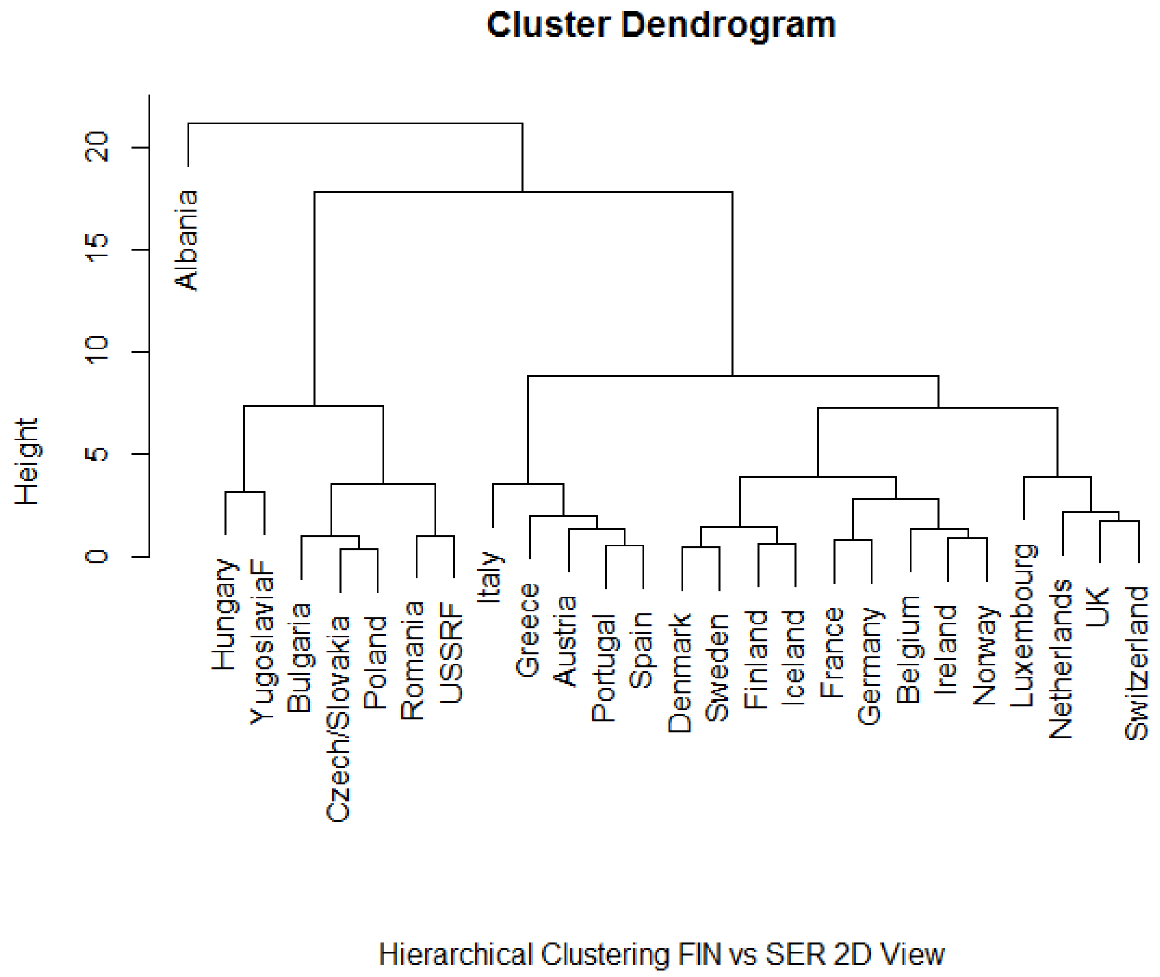


Figure 5.1 Dendrogram for FIN vs SER

Now let's perform the same analysis in the principal component space using the first and second principal components.

```
# Cluster in PC1*PC2 view;
pca.out <- princomp(x=label.data[,-c(1,2)],cor=FALSE);
my.pca <- as.data.frame(list(Country=label.data$Country,Group=label.data$Group,
               pc1=pca.out$scores[,1],pc2=pca.out$scores[,2]) );

eu.pca <- subset(my.pca,Group=='EU');
efta.pca <- subset(my.pca,Group=='EFTA');
eastern.pca <- subset(my.pca,Group=='Eastern');
other.pca <- subset(my.pca,Group=='Other');

pc1.pc2 <- hclust(d=dist(my.pca[,c('pc1','pc2')]),method='complete');
plot(pc1.pc2,labels=my.pca[,1],xlab='Hierarchical Clustering PC1 vs PC2 2D View',sub='');
```

```
pca.3 <- cutree(pc1.pc2,k=3);
pca.3df <- as.data.frame(list(Country=my.pca[,1],Group=my.pca[,2],Cluster=pca.3))
pca.t3 <- table(pca.3df$Group,pca.3df$Cluster)
pca.comp3 <- t(pca.t3[1:3,])*(1/apply(pca.t3[1:3,],FUN=sum,MARGIN=2))
pca.accuracy3 <- sum(apply(pca.t3[1:3,],FUN=max,MARGIN=2))/sum(apply(pca.t3[1:3,],FUN=sum,MARGIN=2));

pca.6 <- cutree(pc1.pc2,k=6);
pca.6df <- as.data.frame(list(Country=my.pca[,1],Group=my.pca[,2],Cluster=pca.6))
pca.t6 <- table(pca.6df$Group,pca.6df$Cluster)
pca.comp6 <- t(pca.t6[1:3,])*(1/apply(pca.t6[1:3,],FUN=sum,MARGIN=2))
pca.accuracy6 <- sum(apply(pca.t6[1:3,],FUN=max,MARGIN=2))/sum(apply(pca.t6[1:3,],FUN=sum,MARGIN=2));
```

Of these four 'cluster models' which one is the most accurate? Make a table to display their accuracy for easy comparison.

Part 6: k-Means Clustering Analysis

Now let's apply k-means clustering to the same data. Do we need to know multiple methods for clustering? Yes. Since hierarchical clustering computes a full cluster tree for k=2 to k=N, it is a computationally expensive clustering technique that cannot be used on larger data sets. Clustering methods that partition the data into k clusters for a specified k are more applicable to larger data sets since they are more computationally efficient. One of, if not THE, most popular clustering technique of the partitioning type is the k-means algorithm.

Let's perform the analogous cluster analysis using k-means for k=3 and k=6. This will allow us to compare the classification accuracy of our different cluster models.

```
# Cluster in FIN*SER view;
# Specify 3 Clusters;
finser.k3 <- kmeans(x=label.data[,c('FIN','SER')],centers=3);
names(finser.k3)

finser.k3df <- as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=finser.k3$cluster,
FIN=label.data$FIN,SER=label.data$SER));
finser.k3tab <- table(finser.k3df$Group,finser.k3df$Cluster);
finser.k3ac <-
sum(apply(finser.k3tab[1:3,],FUN=max,MARGIN=2))/sum(apply(finser.k3tab[1:3,],FUN=sum,MARGIN=2));


# Plot the cluster centers;
plot(label.data$SER,label.data$FIN,xlab='Services',ylab='Finance',xlim=c(0,27),ylim=c(0,17),col='white')
text(eu.df$SER,eu.df$FIN,labels=eu.df$Country,cex=0.75,pos=4,col='green')
text(efta.df$SER,efta.df$FIN,labels=efta.df$Country,cex=0.75,pos=4,col='blue')
text(eastern.df$SER,eastern.df$FIN,labels=eastern.df$Country,cex=0.75,pos=4,col='red')
text(other.df$SER,other.df$FIN,labels=other.df$Country,cex=0.75,pos=4,col='grey')
text(finser.k3$centers[,2],finser.k3$centers[,1],labels=seq(1,3,1),col='black',cex=1)
points(finser.k3$centers[,2],finser.k3$centers[,1],col='black',cex=2.5)
text(finser.k3df$SER,finser.k3df$FIN,labels=finser.k3df$Cluster,col='grey',cex=0.75);
title('k-Means with 3 Clusters')
```

For k-means we can plot the original labels, their assigned clusters, and the cluster centers. Let's take a look at this plot. Here we can see how outliers affect clustering algorithms (they get assigned their own cluster), how our data is split (Eastern Block versus the rest of Europe), and where our four 'Other' countries would belong if we assigned them to a political group based on their economies.

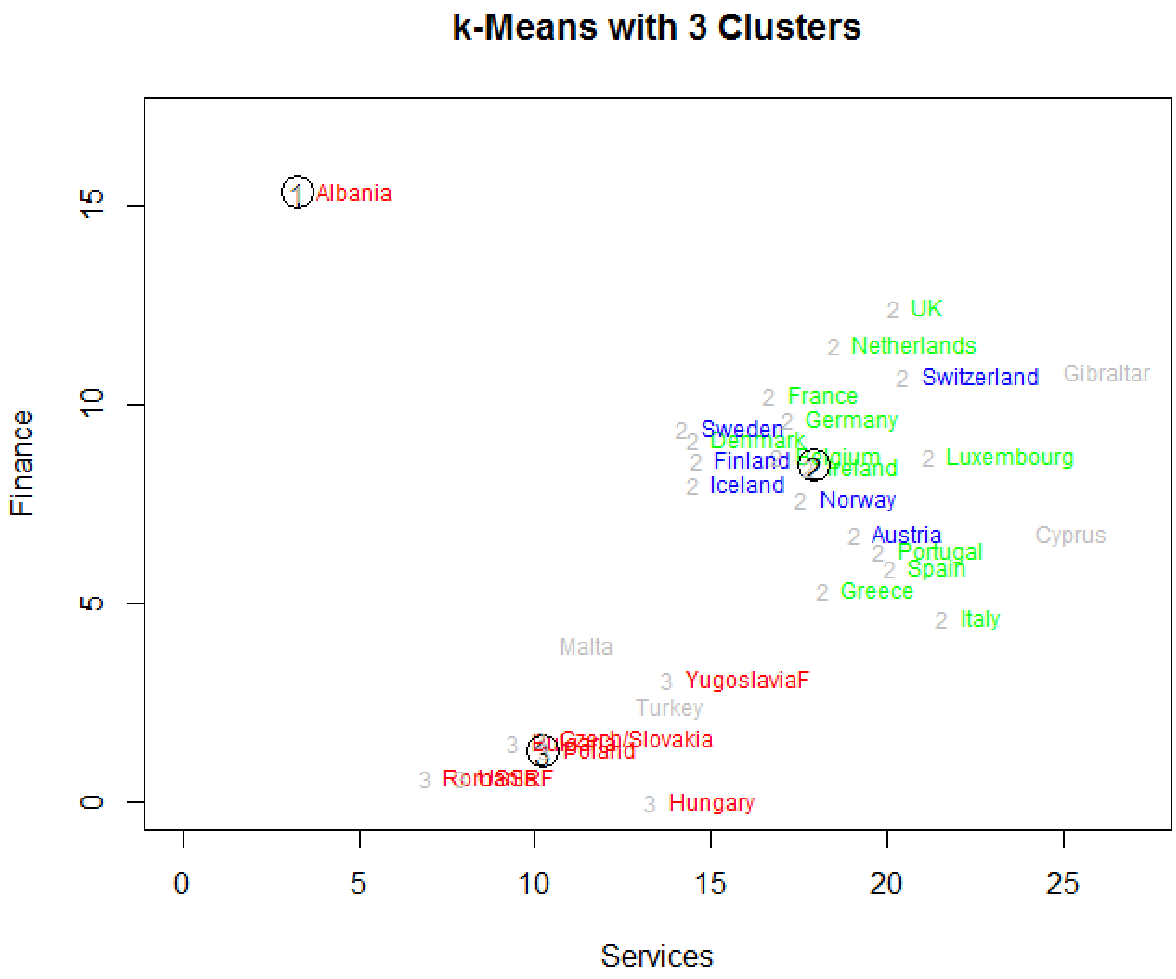


Figure 6.1 k-Means with k=3 for FIN vs SER

Now let’s perform the k-means analysis with k=6.

```
# Specify 6 Clusters;
finser.k6 <- kmeans(x=label.data[,c('FIN','SER')],centers=6);
names(finser.k6)

finser.k6df <- as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=finser.k6$cluster,
FIN=label.data$FIN,SER=label.data$SER));
finser.k6tab <- table(finser.k6df$Group,finser.k6df$Cluster);
finser.k6ac <-
sum(apply(finser.k6tab[1:3,],FUN=max,MARGIN=2))/sum(apply(finser.k6tab[1:3,],FUN=sum,MARGIN=2));

# Plot the cluster centers;
plot(label.data$SER,label.data$FIN,xlab='Services',ylab='Finance',xlim=c(0,27),ylim=c(0,17),col='white')
text(eu.df$SER,eu.df$FIN,labels=eu.df$Country,cex=0.75,pos=4,col='green')
text(efta.df$SER,efta.df$FIN,labels=efta.df$Country,cex=0.75,pos=4,col='blue')
text(eastern.df$SER,eastern.df$FIN,labels=eastern.df$Country,cex=0.75,pos=4,col='red')
text(other.df$SER,other.df$FIN,labels=other.df$Country,cex=0.75,pos=4,col='grey')
text(finser.k6$centers[,2],finser.k6$centers[,1],labels=seq(1,6,1),col='black',cex=1)
points(finser.k6$centers[,2],finser.k6$centers[,1],col='black',cex=2.5)
text(finser.k6df$SER,finser.k6df$FIN,labels=finser.k6df$Cluster,col='grey',cex=0.75);
title('k-Means with 6 Clusters')
```

And now our final set of clusters. Keeping in the mindset of model comparisons let’s perform the same analysis in the principal components space using k=3 and k=6.

```
# Cluster in PC1*PC2 view;
# Specify 3 Clusters;
pca.k3 <- kmeans(x=my.pca[,-c(1,2)],centers=3);
names(pca.k3)

pca.k3df <-
as.data.frame(list(Country=my.pca[,1],Group=my.pca[,2],Cluster=pca.k3$cluster,pc1=my.pca$pc1,pc2=my.pca$
pc2));
pca.k3tab <- table(pca.k3df$Group,pca.k3df$Cluster);
pca.k3ac <- sum(apply(pca.k3tab[1:3,],FUN=max,MARGIN=2))/sum(apply(pca.k3tab[1:3,],FUN=sum,MARGIN=2));

# Plot the cluster centers;
plot(my.pca$pc1,my.pca$pc2,xlab='Principal Component 1',ylab='Principal Component 2',
xlim=c(-60,20),ylim=c(-22,25),col='white')
text(eu.pca$pc1,eu.pca$pc2,labels=eu.pca$Country,cex=0.75,pos=4,col='green')
text(efta.pca$pc1,efta.pca$pc2,labels=efta.pca$Country,cex=0.75,pos=4,col='blue')
text(eastern.pca$pc1,eastern.pca$pc2,labels=eastern.pca$Country,cex=0.75,pos=4,col='red')
#text(other.pca$pc1,other.pca$pc2,labels=other.df$Country,cex=0.75,pos=4,col='grey')
text(pca.k3$centers[,1],pca.k3$centers[,2],labels=seq(1,3,1),col='black',cex=1)
points(pca.k3$centers[,1],pca.k3$centers[,2],col='black',cex=2.5)
text(pca.k3df$pc1,pca.k3df$pc2,labels=pca.k3df$Cluster,col='grey',cex=0.75);
title('k-Means with 3 Clusters')

# Specify 6 Clusters;
pca.k6 <- kmeans(x=my.pca[,-c(1,2)],centers=6);
names(pca.k6)

pca.k6df <-
as.data.frame(list(Country=my.pca[,1],Group=my.pca[,2],Cluster=pca.k6$cluster,pc1=my.pca$pc1,pc2=my.pca$
pc2));
pca.k6tab <- table(pca.k6df$Group,pca.k6df$Cluster);
pca.k6ac <- sum(apply(pca.k6tab[1:3,],FUN=max,MARGIN=2))/sum(apply(pca.k6tab[1:3,],FUN=sum,MARGIN=2));
```

```
# Plot the cluster centers;
plot(my.pca$pc1,my.pca$pc2,xlab='Principal Component 1',ylab='Principal Component 2',
xlim=c(-60,25),ylim=c(-25,30),col='white')
text(eu.pca$pc1,eu.pca$pc2,labels=eu.pca$Country,cex=0.75,pos=4,col='green')
text(efta.pca$pc1,efta.pca$pc2,labels=efta.pca$Country,cex=0.75,pos=4,col='blue')
text(eastern.pca$pc1,eastern.pca$pc2,labels=eastern.pca$Country,cex=0.75,pos=4,col='red')
#text(other.pca$pc1,other.pca$pc2,labels=other.df$Country,cex=0.75,pos=4,col='grey')
text(pca.k6$centers[,1],pca.k6$centers[,2],labels=seq(1,6,1),col='black',cex=1)
points(pca.k6$centers[,1],pca.k6$centers[,2],col='black',cex=2.5)
text(pca.k6df$pc1,pca.k6df$pc2,labels=pca.k6df$Cluster,col='grey',cex=0.75);
title('k-Means with 6 Clusters')
```

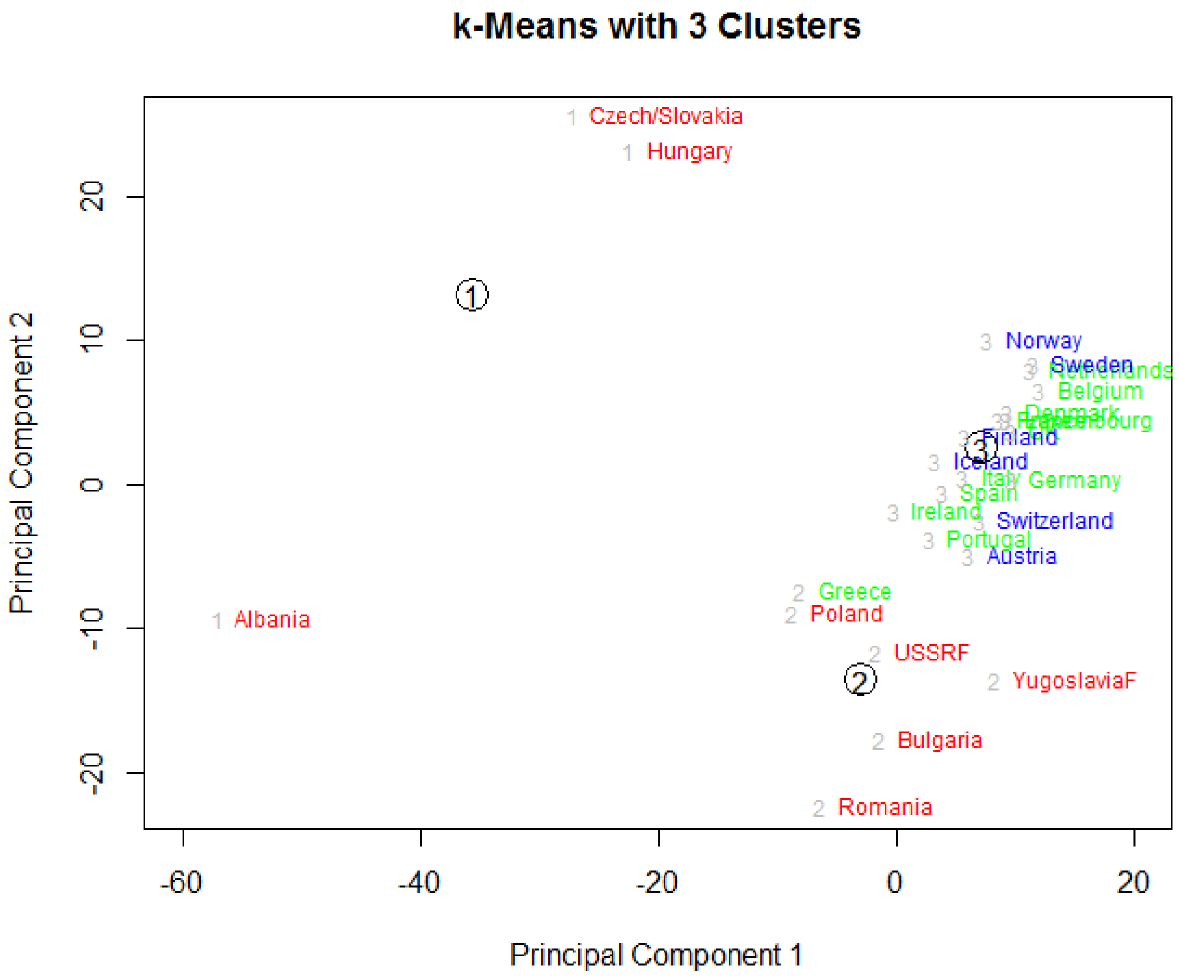


Figure 6.2 k-Means with k=3 for PC1 vs PC2

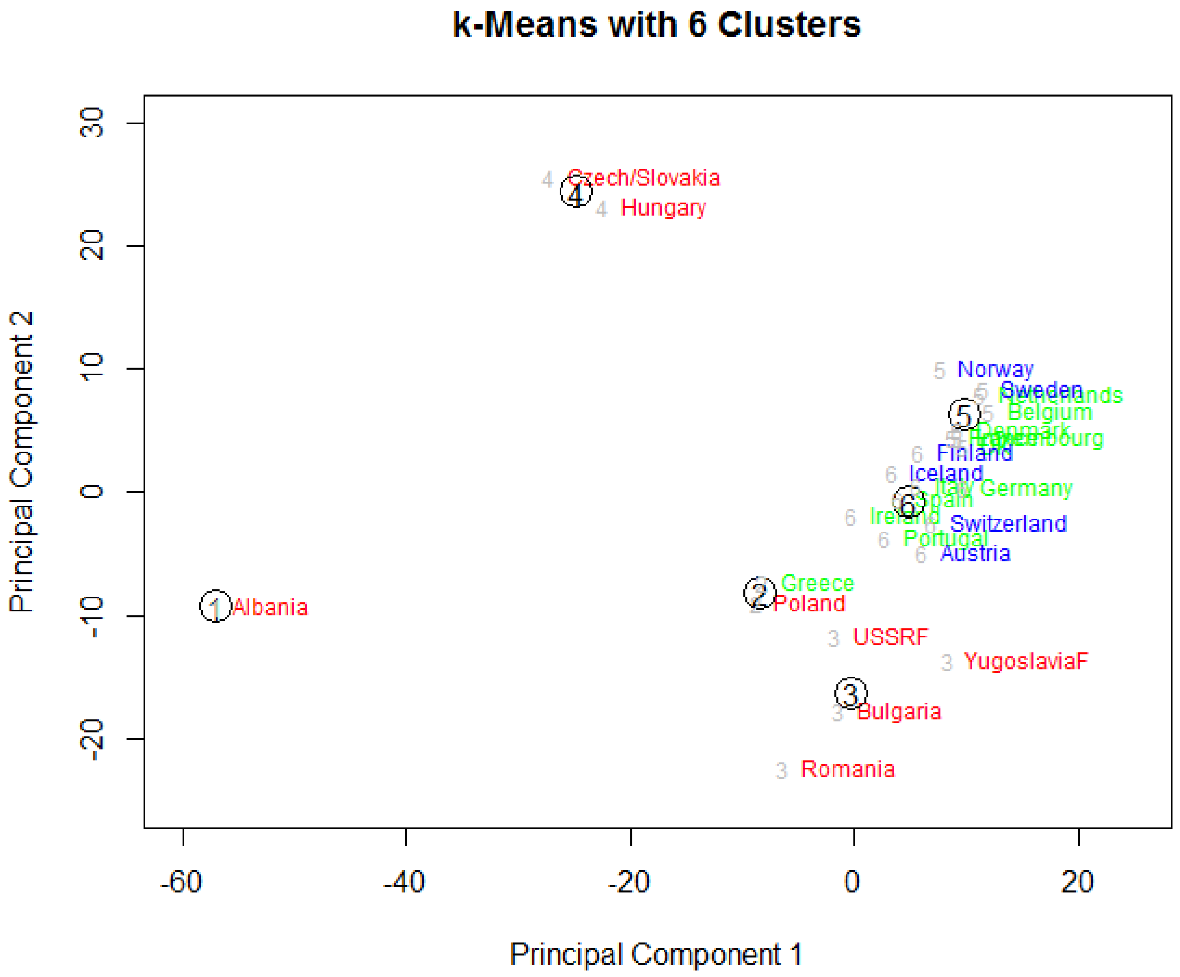


Figure 6.3 k-Means with k=6 for PC1 vs PC2

What do we notice as we increase the number of clusters from k=3 to k=6?

Of these eight cluster models which is the most accurate? Make a table summarizing the eight models and their accuracy.

Part 7: Computing the 'Optimal' Number of Clusters by Brute Force

After completing our initial cluster analyses we should begin to wonder how many clusters would be the correct number of clusters, and how would we determine the correct number of clusters. Unfortunately, the answer to that question is not as simple as the question. One idea that should be apparent is that we would need to be able to evaluate a large number of clusters bases on some criterion that allows an objective comparison. In our problem we can use the classification accuracy rate of our clusters.

Here we have plotted out the classification accuracy for both the hierarchical and k-means clustering algorithms for k=1 to k=20. Overall we can see that the classification accuracy tends to increase as the number of clusters increase, but the classification accuracy is not strictly monotone. Maybe the best cluster model is the k-means cluster model with k=14. What do you think?

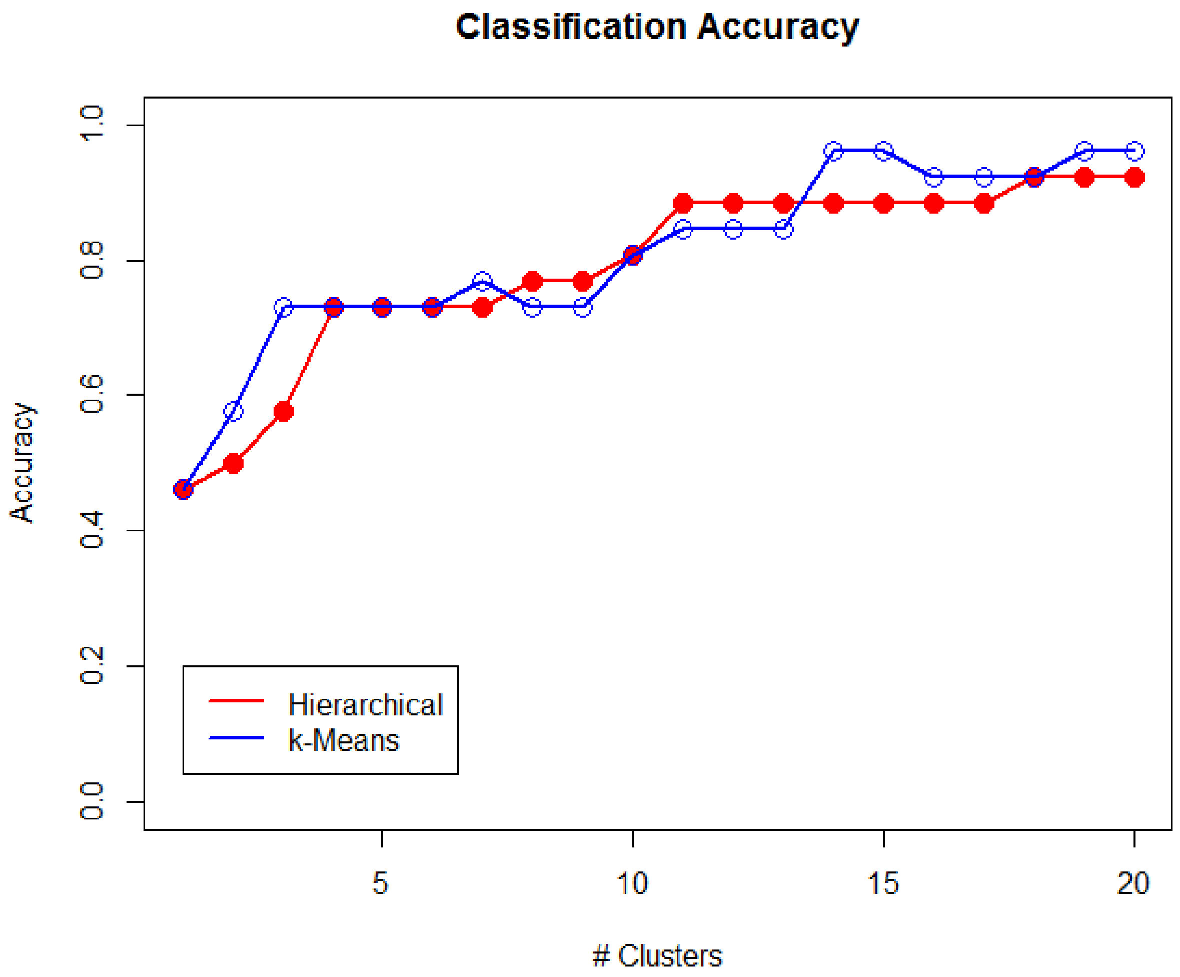


Figure 7.1 Classification Accuracy by Number of Clusters

Run the following code to produce the plot. Spend some time to understand what the code is doing.

```
# Loop through 1-20 clusters using all dimensions;
# Compute the accuracy for each cluster, store, and plot;
# Set the maximum number of clusters to consider;
k.max <- 20;

# Initialize the accuracy arrays for storage;
accuracy.hier <- rep(NA,k.max);
accuracy.kmeans <- rep(NA,k.max);

# Fit the hierarchical clustering model outside of the loop for efficiency;
all.h <- hclust(d=dist(label.data[,-c(1,2)]),method='complete');

# Loop through different cluster sizes and compute classification accuracy;
for (j in 1:k.max){

# Fit hierarchical cluster model of size j;
hier.j <- cutree(all.h,k=j);
hier.df <- as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=hier.j));
hier.table <- table(hier.df$Group,hier.df$Cluster);

# Cannot use apply() on a vector;
if (j==1){
        accuracy.hier[j] <- max(hier.table[1:3,])/sum(hier.table[1:3,]);
}else{
        accuracy.hier[j] <-
sum(apply(hier.table[1:3,],FUN=max,MARGIN=2))/sum(apply(hier.table[1:3,],FUN=sum,MARGIN=2));
}#end if-else;

# Fit k-means clustering model of size j;
kmeans.j <- kmeans(x=label.data[,-c(1,2)],centers=j);
kmeans.df <-
as.data.frame(list(Country=label.data[,1],Group=label.data[,2],Cluster=kmeans.j$cluster));
kmeans.table <- table(kmeans.df$Group,kmeans.df$Cluster);

# Cannot use apply() on a vector;
if (j==1){
        accuracy.kmeans[j] <- max(kmeans.table[1:3,])/sum(kmeans.table[1:3,]);
}else{
        accuracy.kmeans[j] <-
sum(apply(kmeans.table[1:3,],FUN=max,MARGIN=2))/sum(apply(kmeans.table[1:3,],FUN=sum,MARGIN=2));
}#end if-else;

} #end j loop;



plot(seq(1,k.max,1),accuracy.hier,ylim=c(0,1),xlab='#
Clusters',ylab='Accuracy',cex.axis=1,type='l',lwd=2,col='red')
points(seq(1,k.max,1),accuracy.hier,ylim=c(0,1),cex=1.5,type='p',col='red',pch=19)
points(seq(1,k.max,1),accuracy.kmeans,ylim=c(0,1),type='l',lwd=2,col='blue')
points(seq(1,k.max,1),accuracy.kmeans,ylim=c(0,1),cex=1.5,type='p',col='blue')
title('Classification Accuracy')
legend(1,0.2,legend=c('Hierarchical','k-Means'),col=c('red','blue'),lwd=2)
```

Assignment Document:

All assignment reports should conform to the standards and style of the report template provided to you. Results should be presented and discussed in an organized manner with the discussion in close proximity of the results. The report should not contain unnecessary results or information. The section headers in the assignment are the section headers that should be present in the report. The document should be submitted in pdf format. Name your file Assignment8_LastName.pdf.