Cluster Anaylsis of European Languages

Samuel Dummer

10/17/2021

Introduction

There are many languages throughout Europe, of which, many are shared by different countries throughout the continent. There are many countries that have similar dialects. In this data set, it analyzes the percent of people that speak a variety of European languages. These languages include: Finnish, Swedish, Danish, Norwegian, English, German, Dutch, Flemish, French, Italian, Spanish, and Portuguese. To help model the these grouping and similarities we will be using Clustering Analysis. While performing clustering analysis, there are 4 big ideas we will be using distance measures, partition clustering, hierarchical clustering, and k-means clustering.

Clustering is a method in which we can group many objects in data sets that are similar. There are many different fields of work that use this clustering analysis. There are many ways in which one can perform a cluster anaysis, all of which depend on the data set you have. This means that for each unique data set, there will be completely different methods used. The main ways in which clustering analysis can be done are Centroid-based clustering, Distribution-based clustering, Density-based clustering, and Grid-based clustering. In this specific data set, we will be using centroid-based clustering, in this case, specifically k-means clustering.

K-means clustering is described as an unsupervised form of machine learning algorithm that helps group the datapoints into clusters. It uses specific math to help create the smallest possible centroids (a.k.a. clusters) and groups data points into them. In the data set on European Languages, we did some math and found that two clusters would be optimal. This means that when we use the k-means clustering we must take into account that we will have 2 clusters.

Once we have used the k-means method, there are many ways of analyzing the data. One of these methods includes distance measures are methods in which we can measure the distance between clusters. In this sense, the distance helps to show the similarity between two points. There are many different forms of taking this distance. These methods include the Euclidean, Manhattan, Pearson and Spearman distance. The Euclidean and Manhattan method are forms of taking distance that measure the direct distance between point, while, the Pearson and Spearman methods measure the correlation. For example, Pearson's method measures the degree in which there is a linear correlation between two points, and the computers the correlation between the rank of x variables and the rank of y variables.

Moving along, another method of vizulization of the k-means clustering is partition clustering. This method helps to graph the points and show the groupings of the different points. Each data point is colored based on the group that it is in and a circle of polygon (depending on the method of graphing) is drawn around the group

Additionally, hierarchical clustering is another form of visualization of the clustering. This helps to show how similar different objects are. This method goes step by step to find the two closest values then grouping them and continues this until it is one group. Using this it can create a sort of tree diagram that shows the relationship and similarity of two points.

All in all, there are many ways in which the data can be analyzed when using clustering analysis with different methods being more helpful in different data sets.

Methodology

In this specific data set, we started off by loading in many useful libraries such as tidyverse, cluster, and facttoextra. Then we load in the data from a group of data sets given to us. In this case we are analyzing European languages. Next, we quickly looked at the structure and summary of the data set to get a general sense of it. Then we used the fviz_nbclust method to calculate the number of clusters the is optimal. Following this we performed a k-means clustering on the data set to group the points into clusters. Then we performed some partition clustering, distance measures and hierarchical clustering on the data to help visualize the clusters and similarities.

Results

Cleaning Environment, Setting Up Directory, and Loading Libraries

To help get us ready for the script, we need to clean up the environment from any previous scripts. Then we set the directory and load in any necessary librries. Some of these libraries include, tidyverse, cluster, and factoextra. The last two help with the the cluster analysis.

```
rm(list=ls())
setwd("C:/Users/isabe/Desktop/RFLoder")
library(tidyverse)
library(cluster)
library(factoextra)
```

Loading in and Cleaning Data Set

Now that everything is set up and ready for the data set, we can read in the data. In this case we use the read.delim2() function. We also skip the first 18 lines to skip the useless information in the data that might cause an error. Then we need to re-title the columns since it isn't titled. We title it to all the different languages so: Finnish, Swedish, Danish, Norwegian, English, German, Dutch, Flemish, French, Italian, Spanish, and Portuguese.

```
language <- read.delim2(url("https://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file46.txt"),  sep =
colnames(language) <- c("Finnish", "Swedish", "Danish", "Norwegian", "English", "German", "Dutch", "Flem
```

Summary and Structure

Looking at the summary and structure of the data is the most important thing to do once the data has been loaded. Doing so helps to find any problems when reading in the data and helps to get a general sense of the data set. The functions we used to observe the structure and summary were head(), tail(), str(), glimpse(), and summary().

```
head(language)
```

```
##              Finnish Swedish Danish Norwegian English German Dutch Flemish
## West Germany       0       0      0         0      21    100     2       1
## Italy              0       0      0         0       5      3     0       0
## France             0       2      3         0      10      7     1       1
## Netherlands        0       0      0         0      41     47   100     100
```

```
## Belgium            0       0      0         0      14     15     0      59
## Luxemburg          0       0      0         0      31    100     4       1
##              French Italian Spanish Portuguese
## West Germany     10       2       1          0
## Italy            11     100       1          0
## France          100      12       7          1
## Netherlands      16       2       2          0
## Belgium          44       2       1          0
## Luxemburg        92      10       0          0
```

```
tail(language)
```

```
##         Finnish Swedish Danish Norwegian English German Dutch Flemish French
## Sweden        5     100     10        11      43     25     0       0      6
## Denmark       0      22    100        20      38     36     1       1     10
## Norway        0      25     19       100      34     19     0       0      4
## Finland     100      23      0         0      12     11     0       0      2
## Spain         0       0      0         0       5      1     0       0     11
## Ireland       0       0      0         0     100      1     0       0      2
##         Italian Spanish Portuguese
## Sweden        1       1          0
## Denmark       3       1          0
## Norway        1       0          1
## Finland       1       0          0
## Spain         2     100          0
## Ireland       0       0          0
```

```
str(language)
```

```
## 'data.frame':    16 obs. of  12 variables:
##  $ Finnish   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Swedish   : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ Danish    : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ Norwegian : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ English   : int  21 5 10 41 14 31 100 9 18 21 ...
##  $ German    : int  100 3 7 47 15 100 7 0 100 83 ...
##  $ Dutch     : int  2 0 1 100 0 4 0 0 1 1 ...
##  $ Flemish   : int  1 0 1 100 59 1 0 0 1 2 ...
##  $ French    : int  10 11 100 16 44 92 15 10 4 64 ...
##  $ Italian   : int  2 100 12 2 2 10 3 1 2 23 ...
##  $ Spanish   : int  1 1 7 2 1 0 2 2 1 3 ...
##  $ Portuguese: int  0 0 1 0 0 0 0 100 0 1 ...
```

```
glimpse(language)
```

```
## Rows: 16
## Columns: 12
## $ Finnish    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 100, 0, 0
## $ Swedish    <int> 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 100, 22, 25, 23, 0, 0
## $ Danish     <int> 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 10, 100, 19, 0, 0, 0
## $ Norwegian  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 20, 100, 0, 0, 0
## $ English    <int> 21, 5, 10, 41, 14, 31, 100, 9, 18, 21, 43, 38, 34, 12, 5, 1~
```

```
## $ German     <int> 100, 3, 7, 47, 15, 100, 7, 0, 100, 83, 25, 36, 19, 11, 1, 1
## $ Dutch      <int> 2, 0, 1, 100, 0, 4, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0
## $ Flemish    <int> 1, 0, 1, 100, 59, 1, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0
## $ French     <int> 10, 11, 100, 16, 44, 92, 15, 10, 4, 64, 6, 10, 4, 2, 11, 2
## $ Italian    <int> 2, 100, 12, 2, 2, 10, 3, 1, 2, 23, 1, 3, 1, 1, 2, 0
## $ Spanish    <int> 1, 1, 7, 2, 1, 0, 2, 2, 1, 3, 1, 1, 0, 0, 100, 0
## $ Portuguese <int> 0, 0, 1, 0, 0, 0, 0, 100, 0, 1, 0, 0, 1, 0, 0, 0
```

```
summary(language)
```

```
##     Finnish          Swedish          Danish         Norwegian
##  Min.   :  0.000   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :  0.000   Median :  0.00   Median :  0.00   Median :  0.000
##  Mean   :  6.562   Mean   : 10.75   Mean   :  8.25   Mean   :  8.188
##  3rd Qu.:  0.000   3rd Qu.:  7.00   3rd Qu.:  0.75   3rd Qu.:  0.000
##  Max.   :100.000   Max.   :100.00   Max.   :100.00   Max.   :100.000
##     English          German           Dutch           Flemish
##  Min.   :  5.00   Min.   :  0.00   Min.   :  0.000   Min.   :  0.00
##  1st Qu.: 11.50   1st Qu.:  6.00   1st Qu.:  0.000   1st Qu.:  0.00
##  Median : 21.00   Median : 17.00   Median :  0.000   Median :  0.50
##  Mean   : 31.38   Mean   : 34.69   Mean   :  6.875   Mean   : 10.38
##  3rd Qu.: 38.75   3rd Qu.: 56.00   3rd Qu.:  1.000   3rd Qu.:  1.00
##  Max.   :100.00   Max.   :100.00   Max.   :100.000   Max.   :100.00
##      French          Italian          Spanish         Portuguese
##  Min.   :  2.00   Min.   :  0.00   Min.   :  0.000   Min.   :  0.000
##  1st Qu.:  5.50   1st Qu.:  1.00   1st Qu.:  0.750   1st Qu.:  0.000
##  Median : 10.50   Median :  2.00   Median :  1.000   Median :  0.000
##  Mean   : 25.06   Mean   : 10.31   Mean   :  7.625   Mean   :  6.438
##  3rd Qu.: 23.00   3rd Qu.:  4.75   3rd Qu.:  2.000   3rd Qu.:  0.250
##  Max.   :100.00   Max.   :100.00   Max.   :100.000   Max.   :100.000
```

Setting the Seed

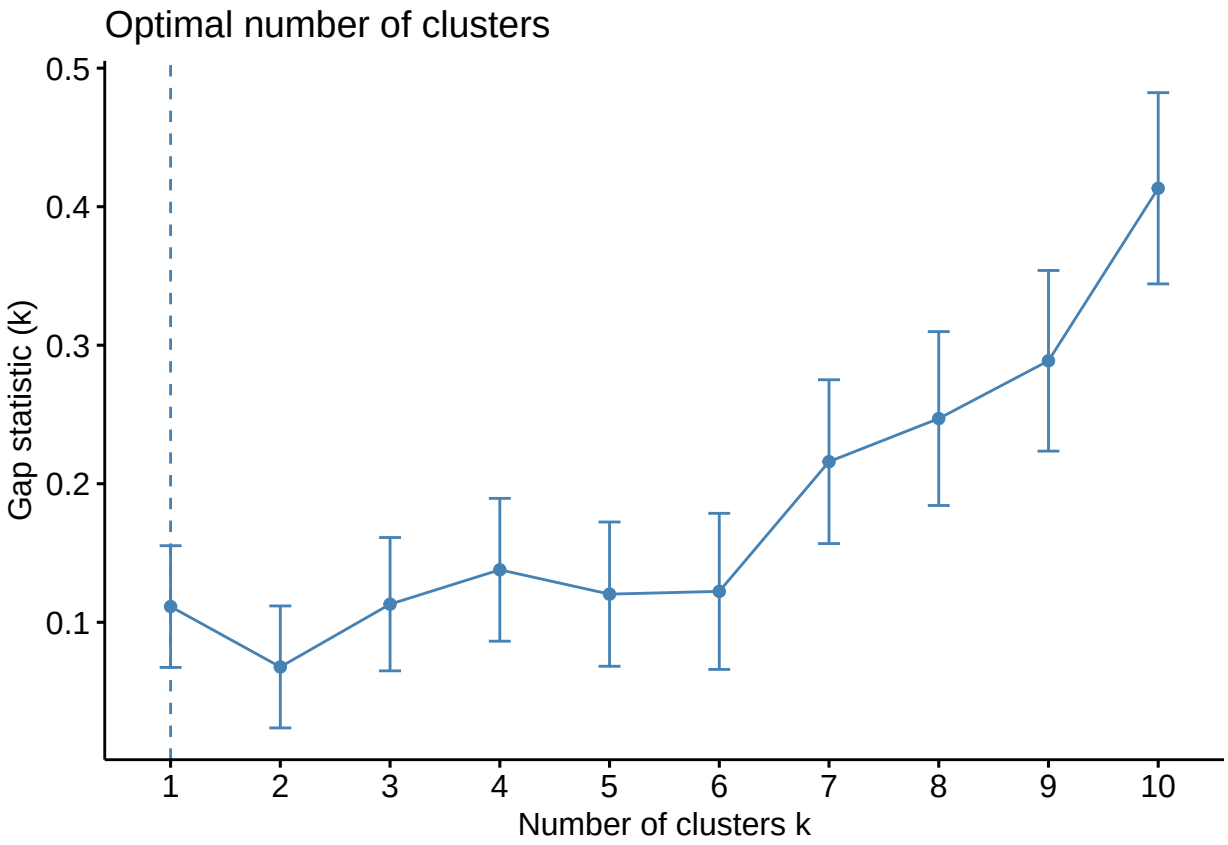
Next we set the seed using set.seed() since the kmean() function needs a seed to run. Any number works as a seed so we just picked a random one.

```
set.seed(14987345)
```

Finding Optimal Number of Clusters

It is a little difficult to find the best number of clusters for the analysis. To help speed up this process, we used the fviz_nbclust() function which gives us the optimal number of clusters we need. In this case we were given 1 cluster, but since that isn't as interesting, we decided to go with 2 clusters.

```
fviz_nbclust(language, kmeans,
             method = "gap_stat")
```



Performing K-Means Clustering

Next, we perform the k-mean method for cluster analysis. We do this by using the kmeans() function. Here, the function is run and then we are given many values, but we mainly focus on the clustering. The different values are placed into each cluster and then we are able to see which countries are in the different clusters.

```
group <- kmeans(language, 2, nstart = 25)
names(group)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

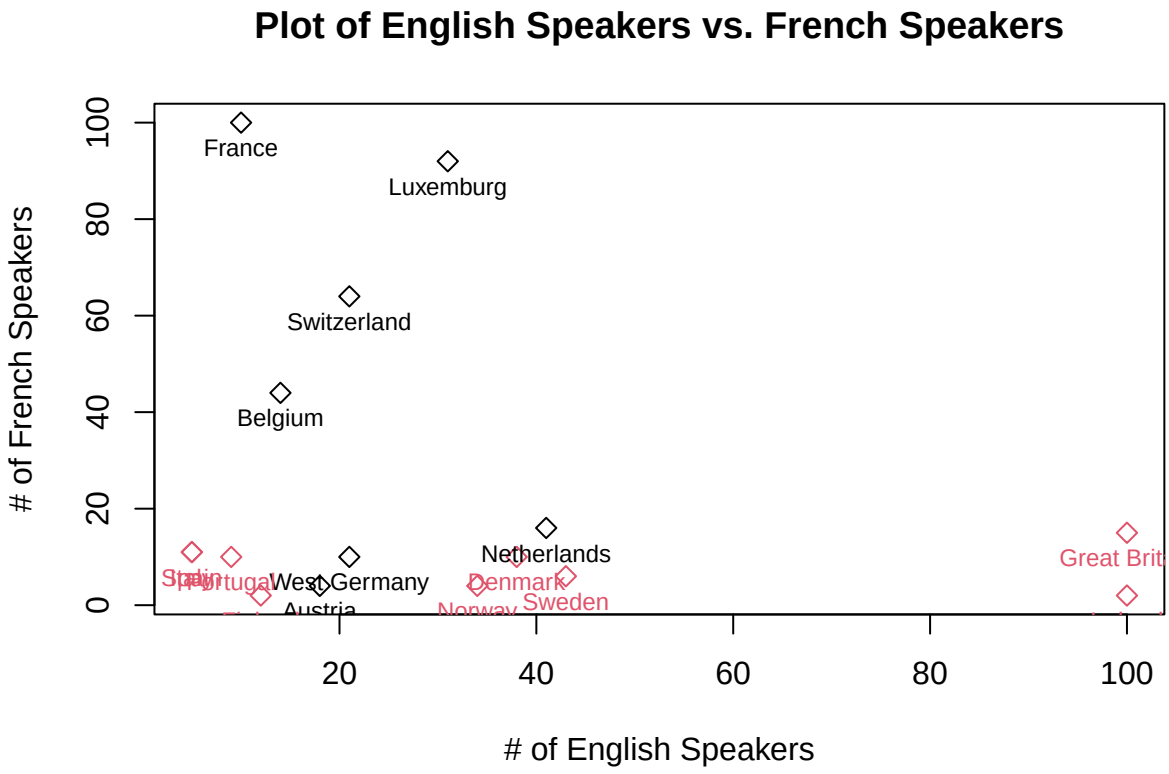
```
group$cluster
```

```
##  West Germany         Italy        France   Netherlands       Belgium
##             1             2             1             1             1
##    Luxemburg Great Britain      Portugal       Austria   Switzerland
##             1             2             2             1             1
##        Sweden       Denmark        Norway       Finland         Spain
##             2             2             2             2             2
##       Ireland
##             2
```

Simple Plot

Here we simply graphed the Countries based on English and French then placed the text for each country. This plot helps to show the relationships between each country based on English and French. For example, we can see the Ireland and Great Britain have high English levels, but everything else is low while France and Belgium has a high French level and low English. Overall, there is a negative exponential also known as an inverse correlation. This means that the higher the English levels, the lower the French.

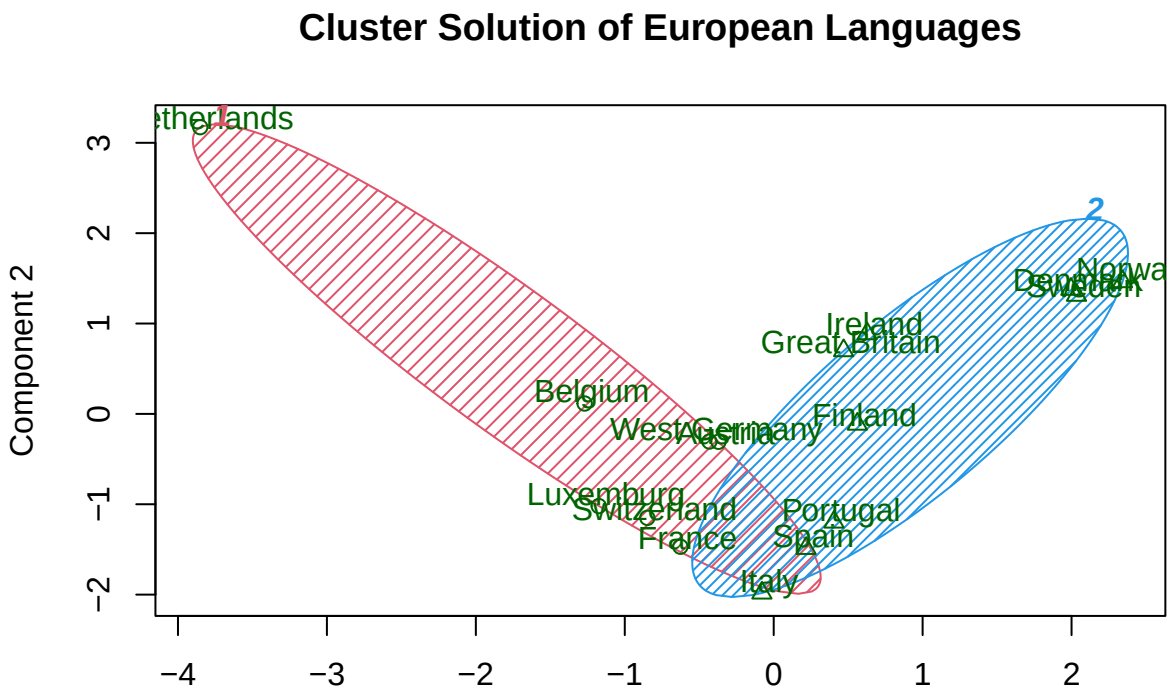
```
plot(language$English,language$French, main = "Plot of English Speakers vs. French Speakers", xlab = "#
text(language[,"English"], language[,"French"], pos = 1, cex = .75, label = row.names(language), col =
```



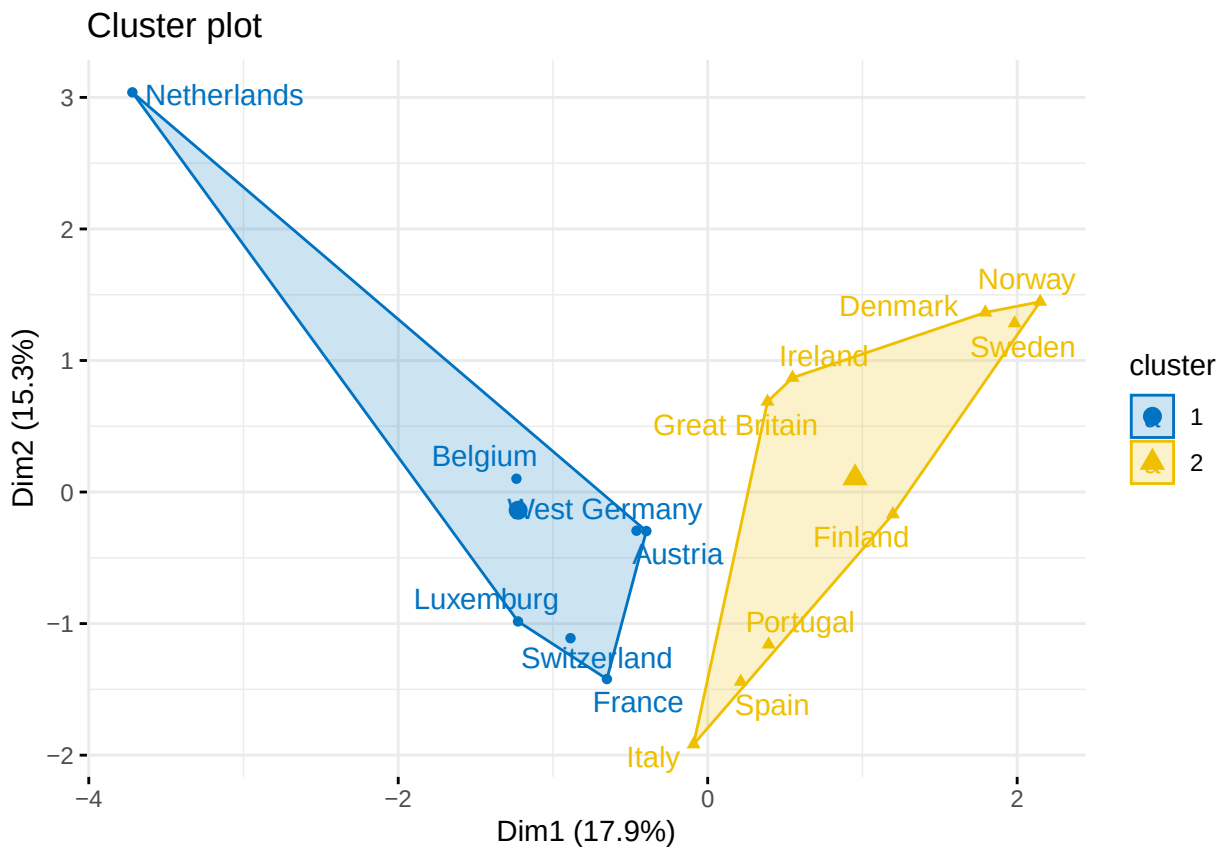
Partition Clustering

Now that the countries have been placed into the clusters, we are able to create some plots. These plots help to show the similarities between different objects and the clusters themselves. Both plots we created were fairly similar in most ways just the format was slightly different. For the first plot, we simply used the clusplot() function and twe are given a plot with the different countries along the plot and a large oval that outlines the cluster. It is also colored based on the cluster. The other graph does a similar plot, except, the cluster is outlined by a ploygon and the clusters are labeled by both color and a shape. These plots are important in showing the relationships between countries. For example we are able to see the major similarities between Germany and Austria, Belgium and France, and Great Britain and Ireland. This makes sense since each of these pairs of countries speaks similar language and is geographically close to each other.

```
clusplot(language[,-1], group$cluster, main = "Cluster Solution of European Languages", color = T, shade
```



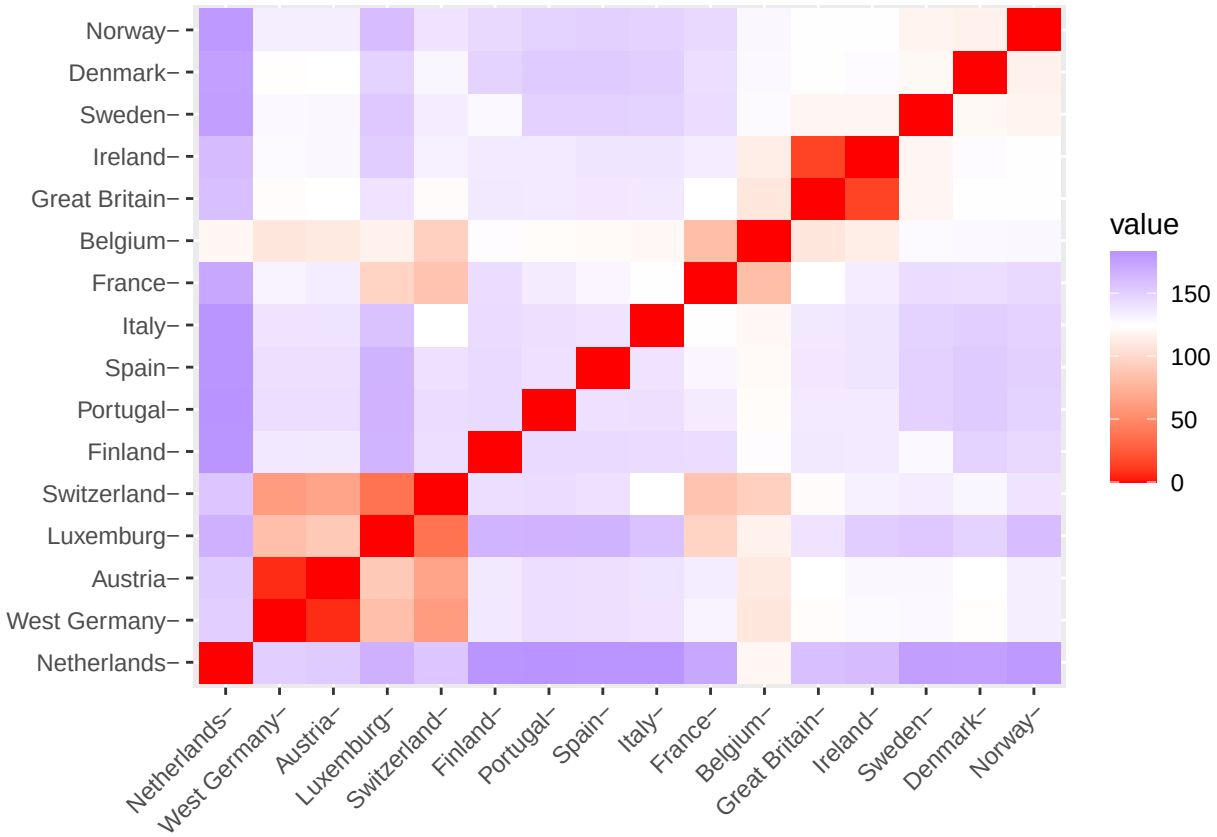

```
fviz_cluster(group, data = language,
             ellipse.type = "convex",
             palette = "jco",
             repel = TRUE,
             ggtheme = theme_minimal())
```



Distance Measure: Euclidean Method

Next, we created a plot that helps to show the similar data. We created a sort of heat plot that helps to show the distances between different countries. In this case, the more red it is, the closer it is and the more purple it is the further it is. We are able to observe similar data as the previous graph, but this time, is is also easy to observe the countries which is the furthest away. For example, Portugal is the furthest away from the Netherlands in terms of language since most Portuguese people speak Portuguese which Dutch people speak Dutch. There is also no real overlap since the countries are fairly far away from each other.

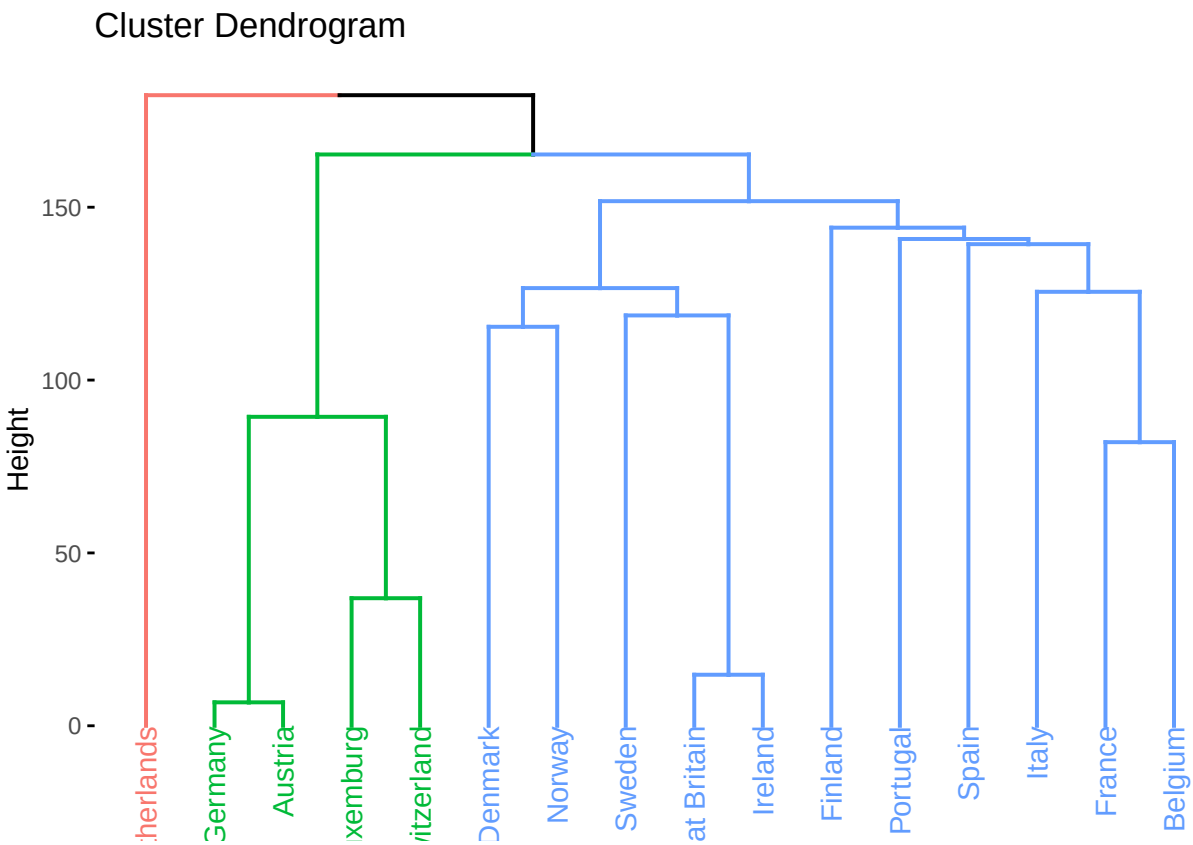
```
dist.euc.lang <- dist(language, method = "euclidean")
fviz_dist(dist.euc.lang)
```



Hierarcical Clustering

Now that we know some of the relationship, we can continue and see all the relationships between countries. This can be done through hierarchical clustering which takes each of the next closest points one at a time to form a sort of tree diagram of the points. This is a much simpler method to observe the relationships and similarities. In this data set, we can see many of the similarties found earlier, plus we can observe that Denmark, Norway, and Sweden are very similar, as are Luxembourg and Switzerland.

```
fviz_dend(hclust(dist(language)), k=3,
          as.ggplot = T,
          show_labels = T)
```



Conclusion

This script performs a cluster analysis on data of European Languages in different countries. This cluster analysis helped us find many countries that had similar language speaking levels. This means that countries such as Ireland and Britain were close to each other. Additionally it grouped the countries into 2 clusters that were the most similar. In general, this showed many surprises such as how the Netherlands was nowhere close to any other countries in terms of language which very much surprised me since I believed that it would hat least have some in common with countries such as Belgium and Switzerland. This analysis was very helpful overall in finding how much countries varied from each other.

References

No References