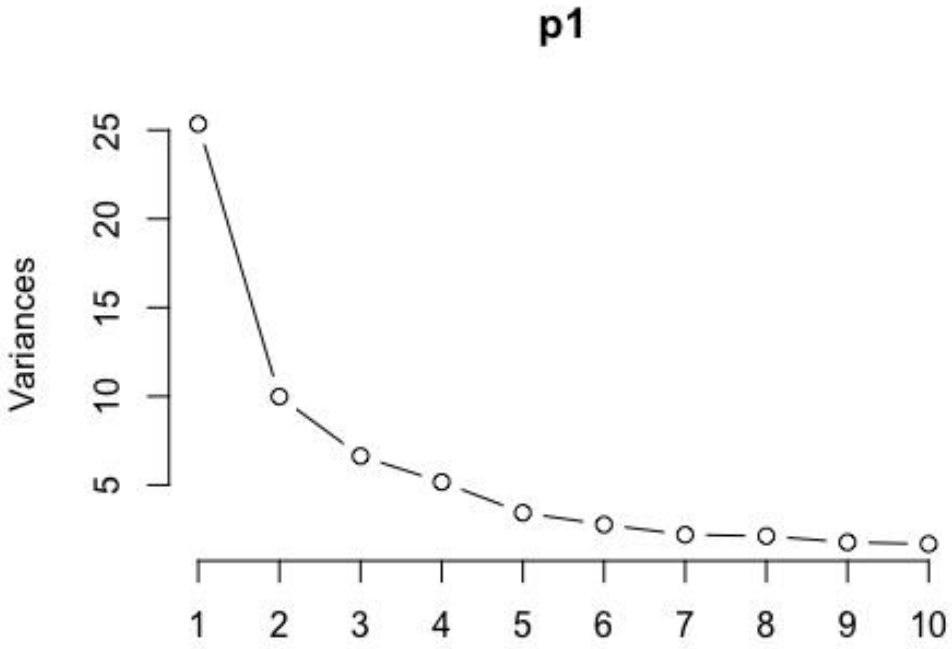
Result from R code sample

LINGYI TAN

Clustering

1. Read in the dataset and conduct PCA to reduce dimension

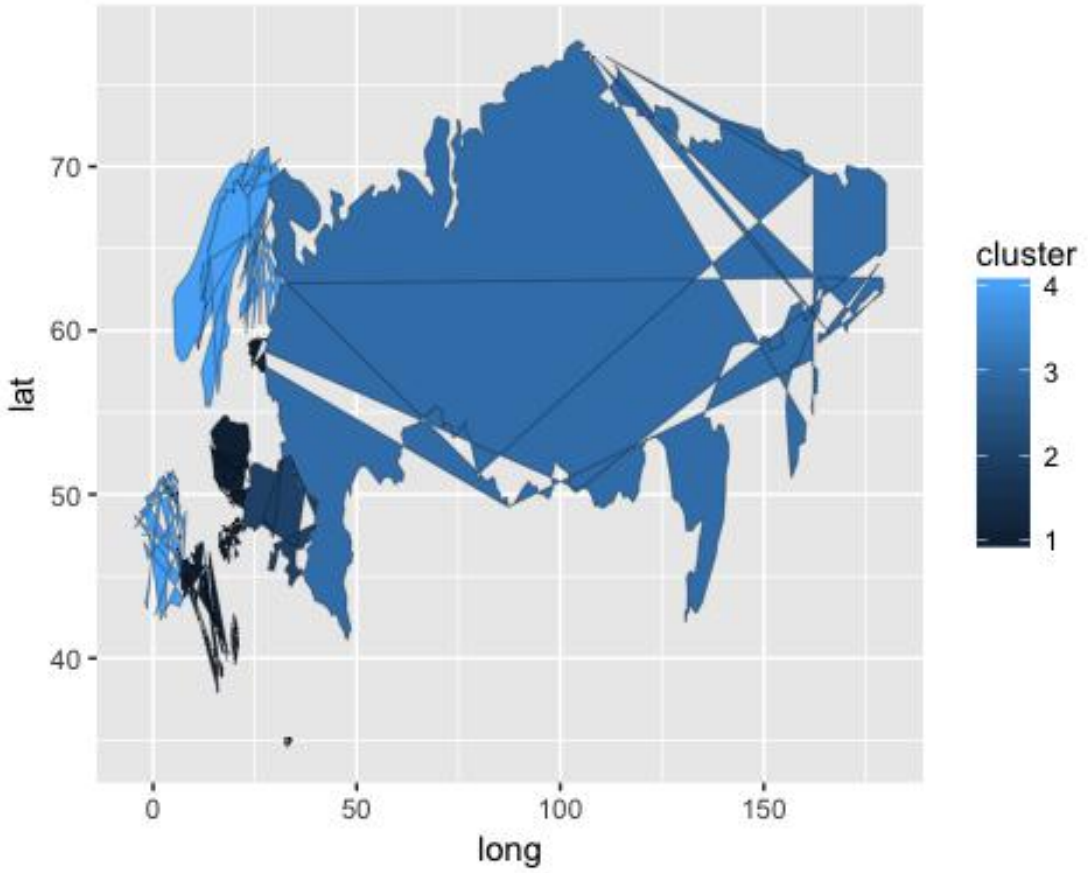
```
#Screenplot for PCA
screeplot(p1, type="lines")
```




```
#According to screenplot, 5 principal components should be selected to summarize the data since the slope between 5 and 6 is less steep then previous slopes.
```

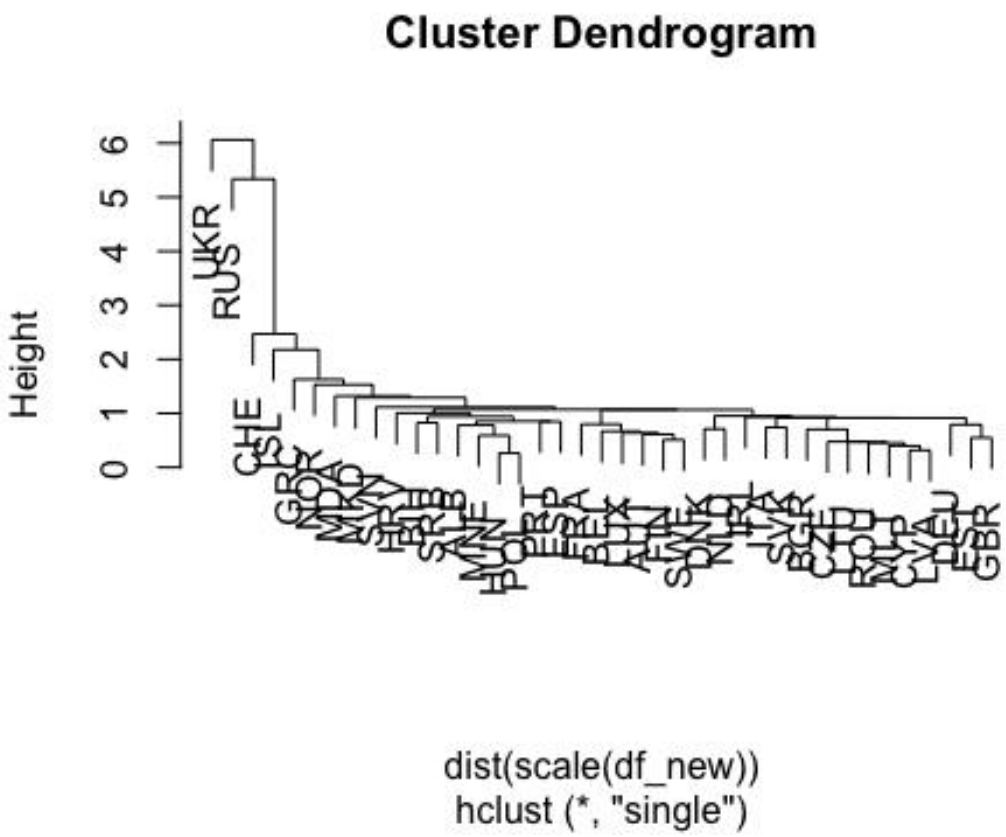
2. Run k-means clustering with multiple initializations

```
ggplot() +
geom_polygon(data = europeCoords,
aes(x = long, y = lat, group = region, fill = cluster),color = "black", size = 0.1)
```

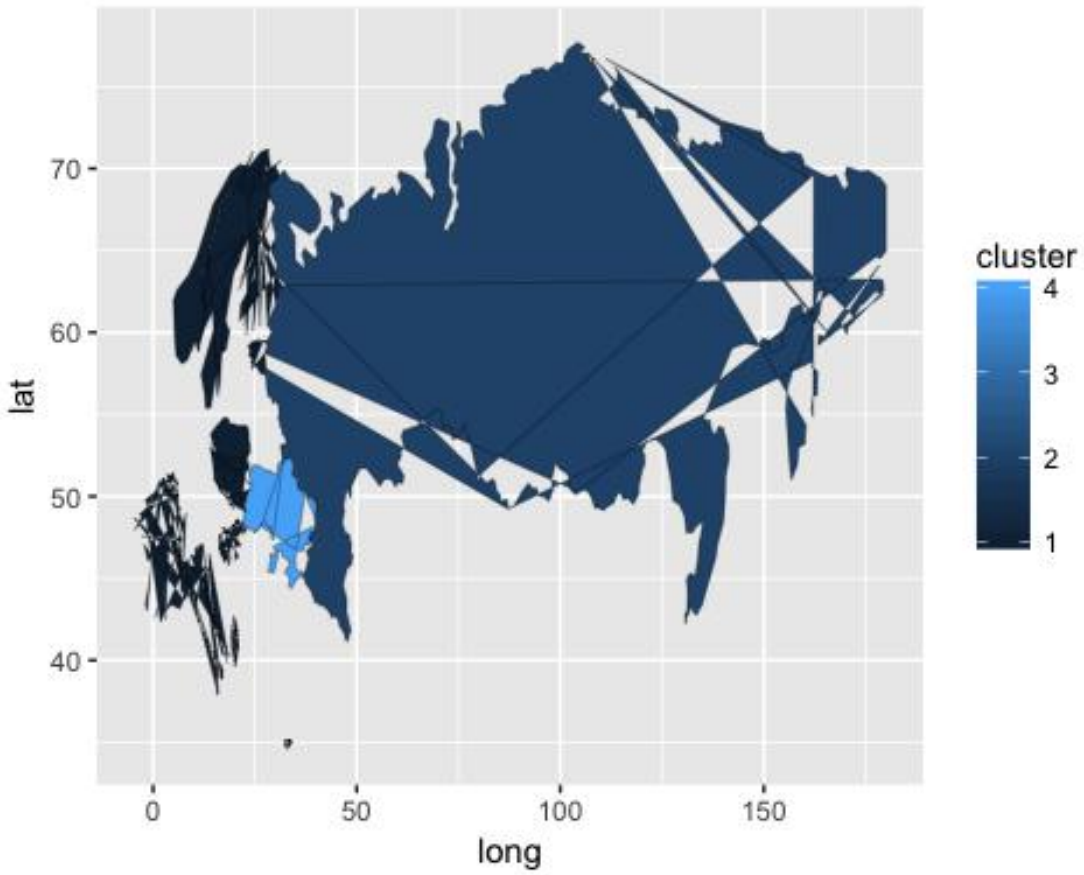


3. Apply hierarchical clustering to the projected data and compare the dendrograms obtained by single linkage and complete linkage.

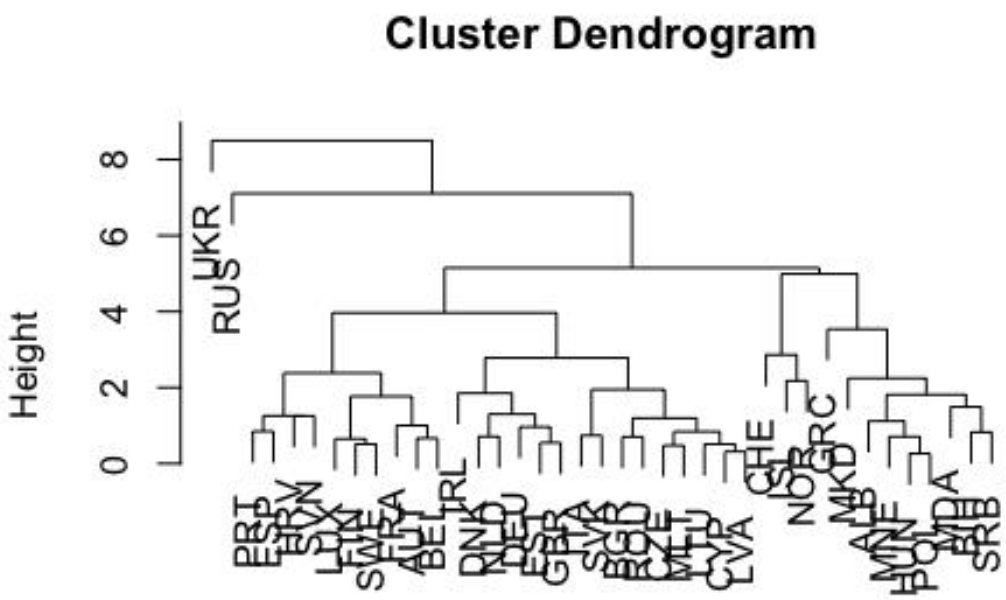
```
hc1 <- hclust(dist(scale(df_new)), method="single")
#unbalanced cluster
plot(hc1)
```




```
ggplot() +
geom_polygon(data = europeCoords,
aes(x = long, y = lat, group = region, fill = cluster),color = "black", size = 0.1)
```

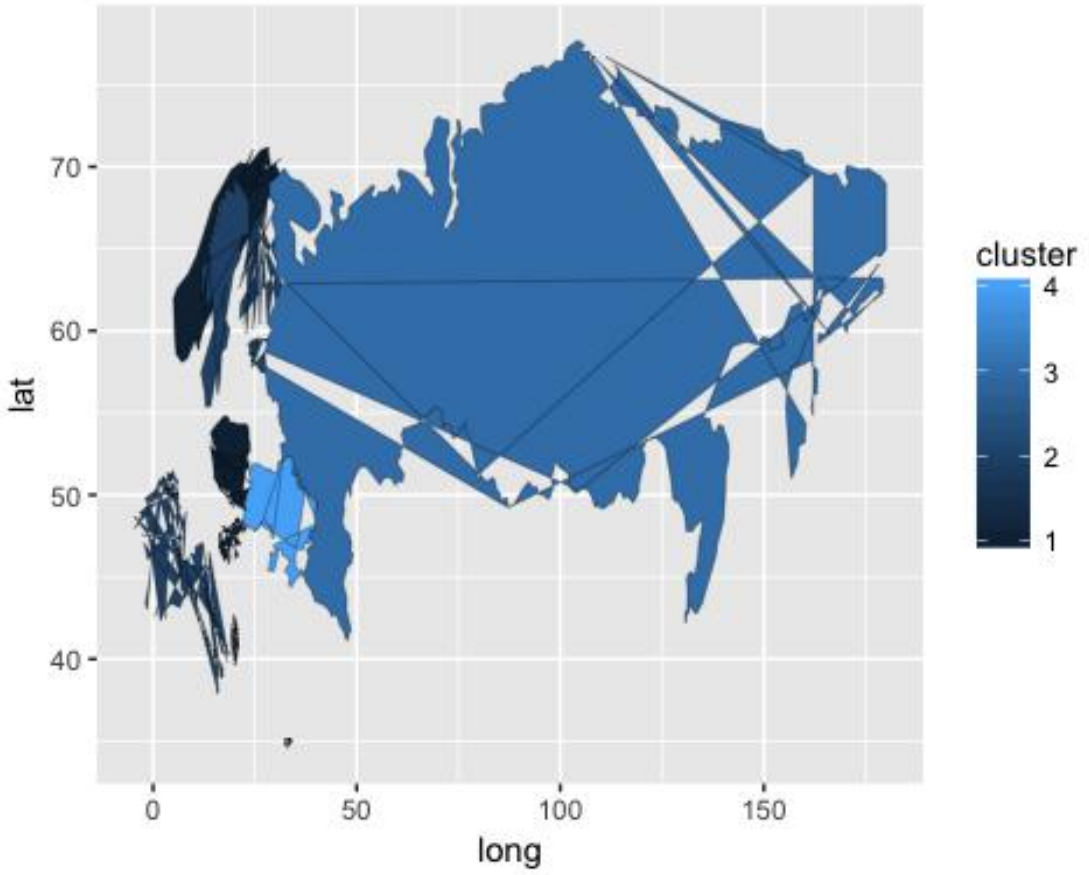



```
hc2 <- hclust(dist(scale(df_new)), method="complete")
plot(hc2)
```



#Cluster in single linkage are extremely unbalanced and spread out with only one county respectively in the last three clusters and all of the rest 36 countries classified into the first cluster. With clusters fused together, single linkage has a lower dissimilarity score (height). Cluster in complete linkage is much more balanced, with two clusters have over 10 counties and a greater height.

```
ggplot() +
geom_polygon(data = europeCoords,
aes(x = long, y = lat, group = region, fill = cluster),color = "black", size = 0.1)
```



Cluster memberships obtain from these two approaches are significantly different from that from k-means. Clusters in k-means are much more balanced than that in clustering. Most countries are classified into one or cluster and fewer countries are classified into the following clusters, and this situation is extremely pronounced at the single linkage in which the last three clusters have only one country respectively.

Implement of latent Dirichlet allocation (LDA) topic modeling

3. Run LDA on the reduced document term matrix with 3 topics

```
tweet<-read.csv("https://www.dropbox.com/s/wbrnfj3tt86ra24/tweets.csv?raw=1")
library(tm)



mod1 <- LDA(dtm.common, k, control=list(seed = 1221))
terms(mod1,10)
```

```
##       Topic 1   Topic 2   Topic 3
##  [1,] "trump"   "trump"   "trump"
##  [2,] "hillari" "hillari" "clinton"
##  [3,] "will"    "get"     "amp"
##  [4,] "dont"    "peopl"   "obama"
##  [5,] "amp"     "clinton" "new"
##  [6,] "just"    "obama"   "make"
##  [7,] "obama"   "right"   "call"
##  [8,] "want"    "one"     "like"
##  [9,] "say"     "think"   "will"
## [10,] "clinton" "can"     "now"
```

4. Remove the words that are too common and again run LDA to regain 3 topics

```
mod2 <- LDA(dtm.reduced, k, control=list(seed = 1221))
terms(mod2,10)
```

```
##       Topic 1  Topic 2   Topic 3
##  [1,] "polic"  "voter"   "republican"
##  [2,] "free"   "anoth"   "liber"
##  [3,] "bill"   "hes"     "email"
##  [4,] "wont"   "everi"   "even"
##  [5,] "hous"   "isi"     "happen"
##  [6,] "report" "protect" "islam"
##  [7,] "ask"    "poll"    "famili"
```

```
##  [8,] "well"    "attack"  "turn"
##  [9,] "woman"   "immigr"  "girl"
## [10,] "law"     "ccot"    "read"
```

#Words in mod 1 are more about names and frequently-used verb such as get, make, want or say. We can't get lots of useful information from mod1 and we can't conclude what each topic is about. While words in mod2 show us lots of useful information and we can roughly know what each topic is about. The first topic may be about policy and law, the second is about terrorism and the third is about political party.