Multivariate CW
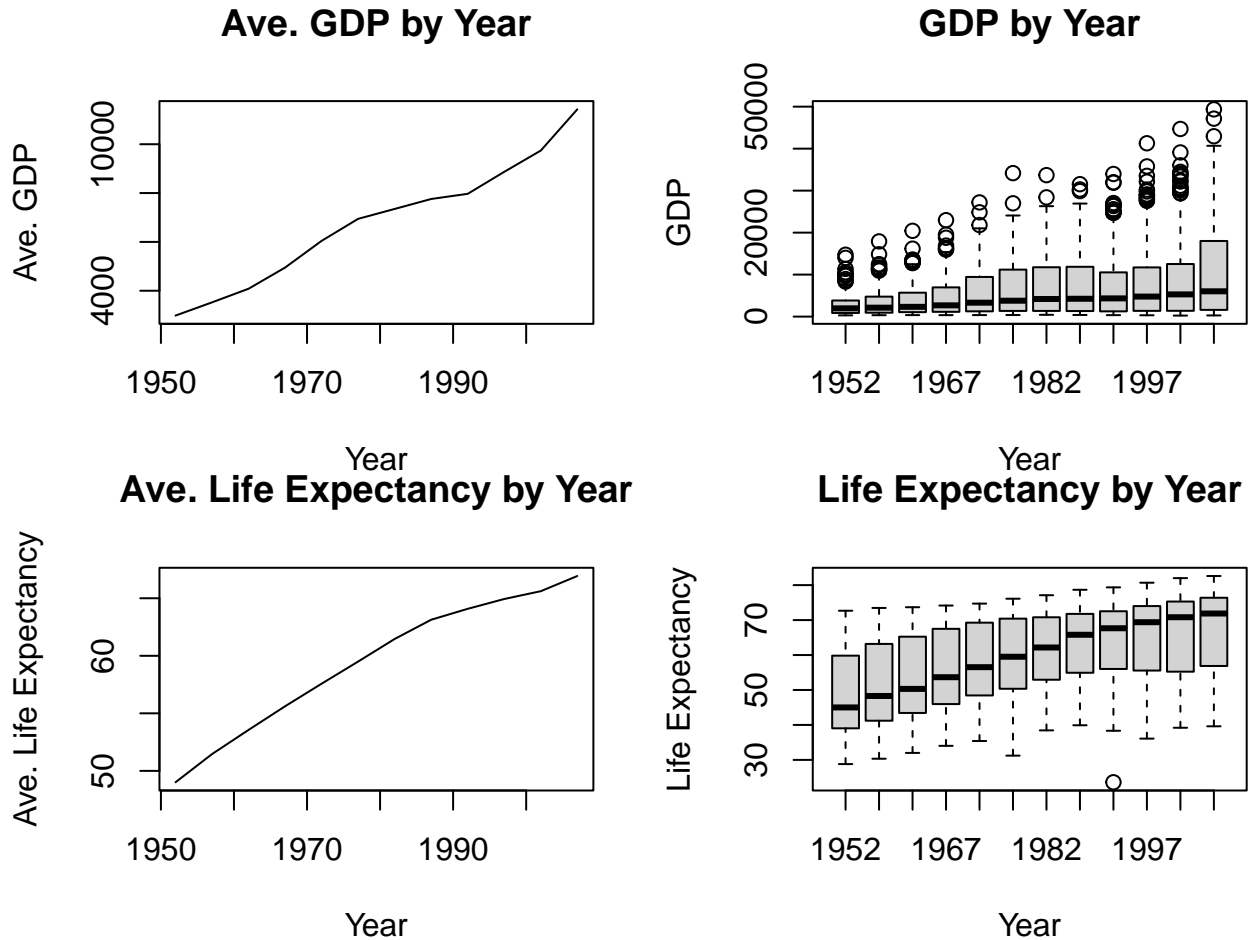
Liangxiao LI
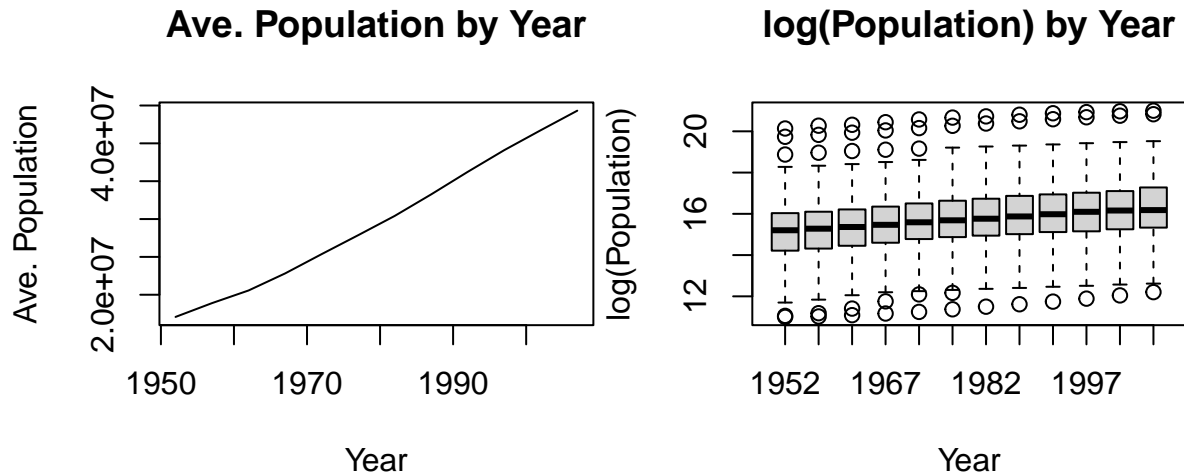
2024-04-13

Part1: Exploratory Data Analysis

First we load the data

Since we have 141 rows of different countries, therefore visulizing individual line plots for each country would result in a cluttered figure. For such a large number of states, I'll focus on aggregate plots by calculating the average GDP, life expectancy, population accross the United State.

Ave. GDP by Year

GDP by Year

Ave. Life Expectancy by Year

Life Expectancy by Year

Ave. Population by Year   **log(Population) by Year**

From the above line plots, it can be concluded that the average GDP, life expectancy and population are growing steadily across the globe as years goes by.

In the following section we plot the box plots for each dataframe, since the population value exceed the R integer boundary, we'll plot 'years x log(population)' for the population box plot

Part2: Principal component analysis

Here for all three different datasets, I perform PCA horizontally, treating each country as a data point and each year as a feature.

Since the columns are 12 different years, this means the features are measuring similar entities, therefore we should perform PCA based on *S(sample covariance matrix)* for gdp, life expectancy and population.

## Part2.1: Number of PCs to retain

First we plot the scree plot and the corresponding biplots to decide how many PCs we should retain.
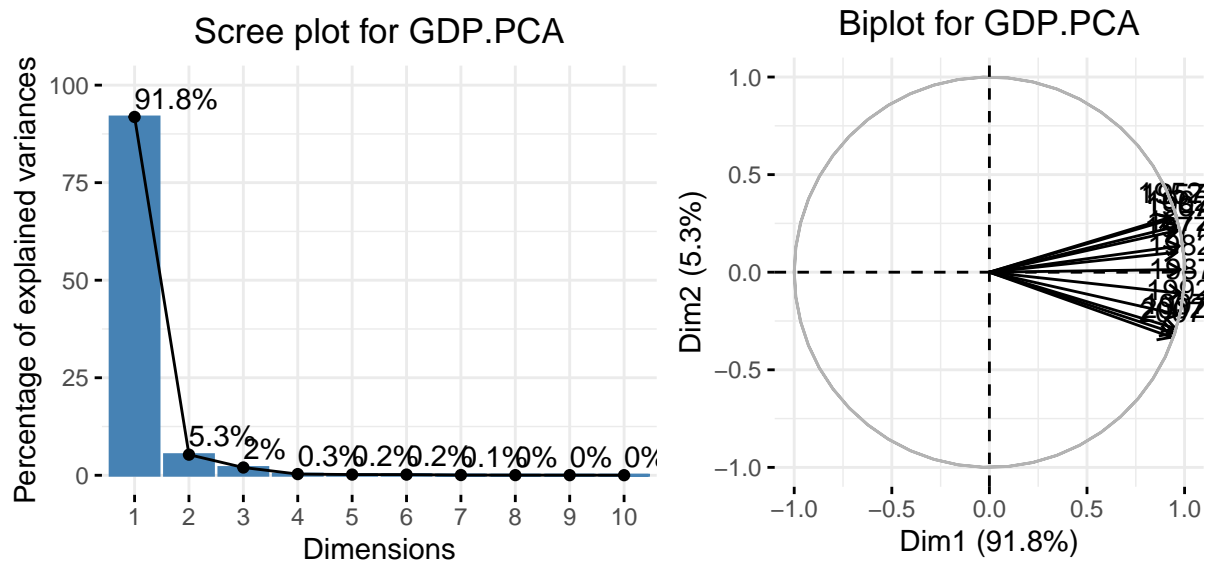
From the scree plot, we'd retain PC1 and PC2 as they explained 91.8% and 5.3% of the variance within the data.

Scree plot for LE.PCA

92.3%

5.6%
1.3% 0.4% 0.1% 0.1% 0.1% 0% 0% 0%

Variables – PCA
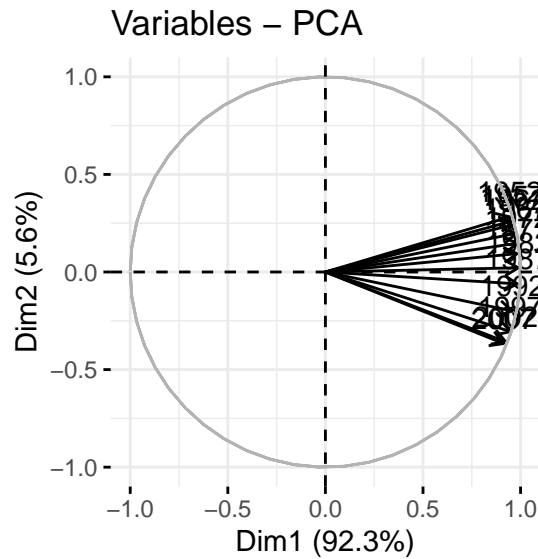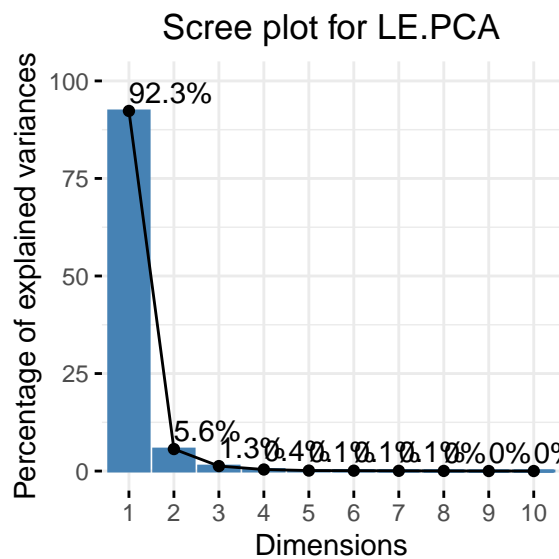
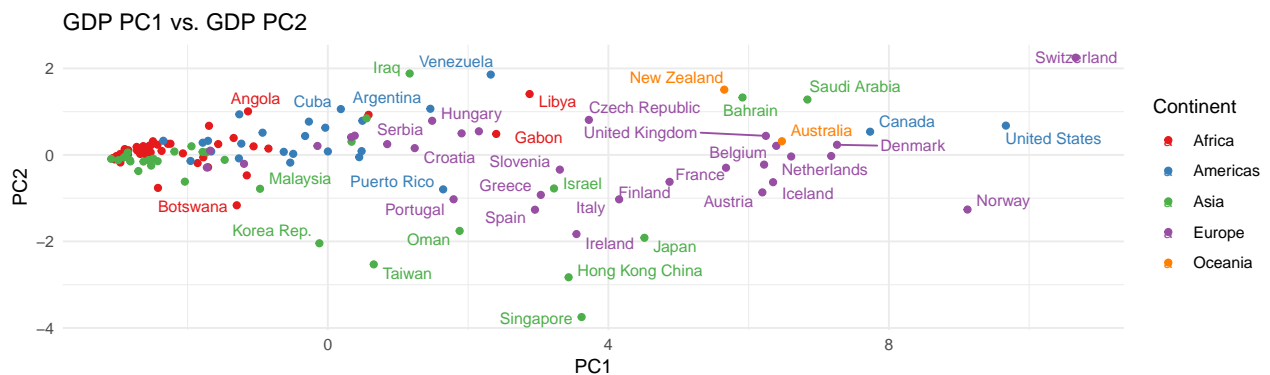From the scree plot, we'd retain PC1 and PC2 for life expectancy as they explained 92.3% and 5.6% of the variance within the data.

Scree plot for Popn.PCA

99.6%

0.3% 0% 0% 0% 0% 0% 0% 0% 0%

Variables – PCA
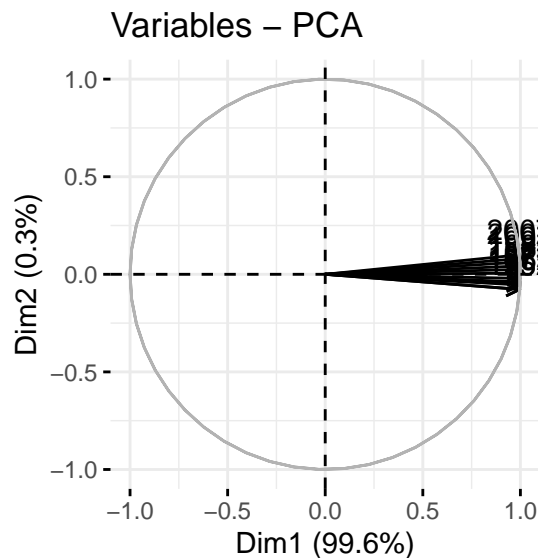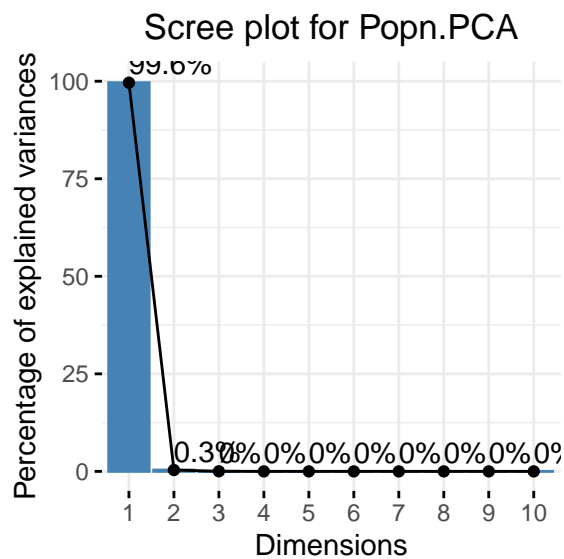
From the scree plot, we'd retain PC1 for population as it explained 99.6% of the variance within the data.

Part2.2: Scatter plots for PCs and interpretations

GDP PC1 vs. GDP PC2

```
##      1952 1957 1962 1967 1972 1977 1982  1987  1992  1997  2002  2007
## PC1 0.28 0.28 0.29 0.29 0.29 0.29 0.29  0.30  0.29  0.29  0.28  0.28
## PC2 0.35 0.34 0.30 0.27 0.17 0.13 0.02 -0.14 -0.27 -0.36 -0.39 -0.42
```
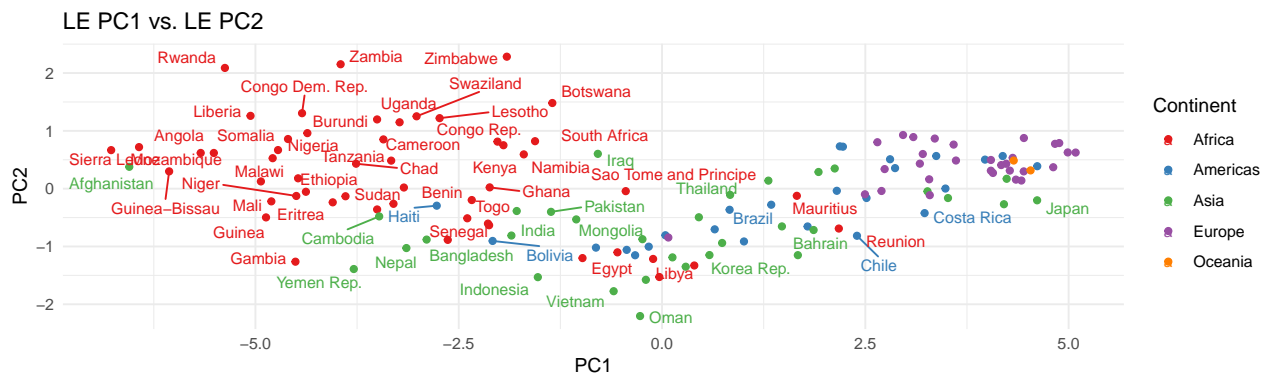
Above is the scatter plot of PC scores and the loadings for GDP PC1/2:

a) PC1 seems to measure a general trend for GDP, which can be seen from the loadings above (All loadings are approximately +0.28 ~ +0.3). Higher PC1 indicate higher overall GDP.

Therefore **European** countries have generally higher GDP while **African** countries have generally lower GDP.

b) PC2 Might represent a cyclic variation which such as economic fluctuations. As higher PC2 indicate higher GDP between 1952-1982, while lower PC2 indicate higher GDP between 1987-2007.

Therefore **Asia** countries tend to have higher GDP after 1987, especially Singapore.

LE PC1 vs. LE PC2

```
##      1952 1957 1962 1967 1972 1977 1982  1987  1992  1997  2002  2007
## PC1 0.28 0.29 0.29 0.29 0.30 0.29 0.30  0.30  0.29  0.28  0.28  0.27
## PC2 0.34 0.32 0.30 0.24 0.18 0.11 0.03 -0.07 -0.24 -0.37 -0.43 -0.45
```
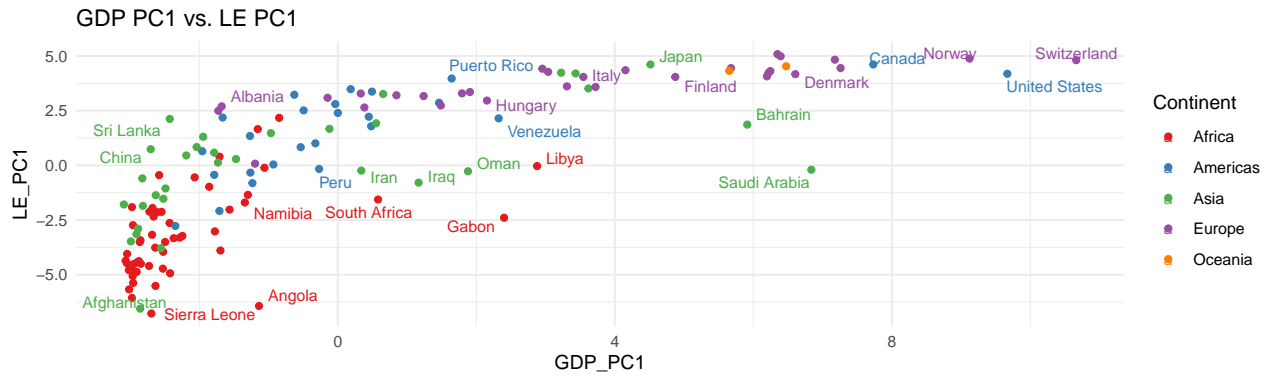
Above is the scatter plot of PC scores and the loadings for life expectancy PC1/2:

a) PC1 seems to measure a general trend for life expectancy, which can be seen from the loadings above (All loadings are approximately +0.28 ~ +0.3). Higher PC1 indicate higher overall life expectancy.

Therefore **European** countries have the highest overall life expectancy while **African** countries have the lowest overall life expectancy.

b) PC2 Might represent a cyclic variation which such as . As higher PC2 indicate higher life expectancy between 1952-1982, while lower PC2 indicate higher life expectancy after 1987.

Therefore **Asian** countries tend to have higher life expectancy after 1987, especially Oman. This means that Asian countries started developing quickly after 1987.

GDP PC1 vs. LE PC1

```
##              1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
## LE_PC1       0.28 0.28 0.29 0.29 0.29 0.29 0.29  0.3 0.29 0.29 0.28 0.28
## GDP_PC1      0.28 0.29 0.29 0.29 0.30 0.29 0.30  0.3 0.29 0.28 0.28 0.27
```

Above is the scatter plot of PC scores and the loadings for the first PC score for life expectancy against first PC score for GDP:

As mentioned in previous sections, GDP_PC1 and LE_PC1 both measure general trends for GDP and LE. Higher GDP_PC1 indicate higher overall GDP, higher LE_PC1 indicate higher overall life expectancy.

Therefore **African** tend to have both the lowest overall GDP/life expectancy. **European** and **American** have the higher overall GDP/life expectancy.

For individual countries, **United States**, **Switzerland** and **Norway** have both high GDP_PC1 and LE_PC1, indicating best economic situation and citizen health condition. **Sierra Leone** and **Afgharistan** have low GDP_PC1 and LE_PC1, indicating worst economic situation and citizen health condition.

## Canonical correlation analysis

Scatter Plot of Canonical Scores
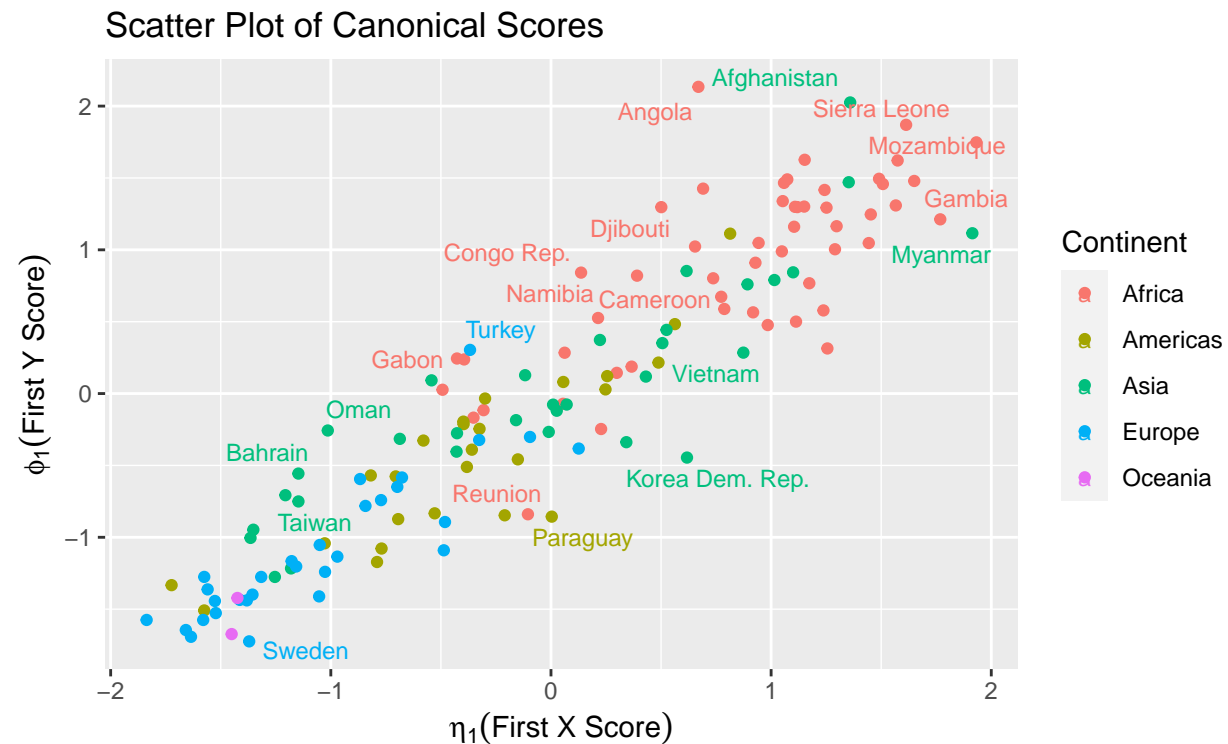
Above is the scatter plot of the first pair of CC variables. From this plot we can conclude that there are strong correlation between the first pair of CC variables η_1(First X score) and ϕ_1(First Y score).

Interpretation

To help interpret the first pair of canonical variables, I ploted the their loadings(x/y coefficients) in the following graphs.
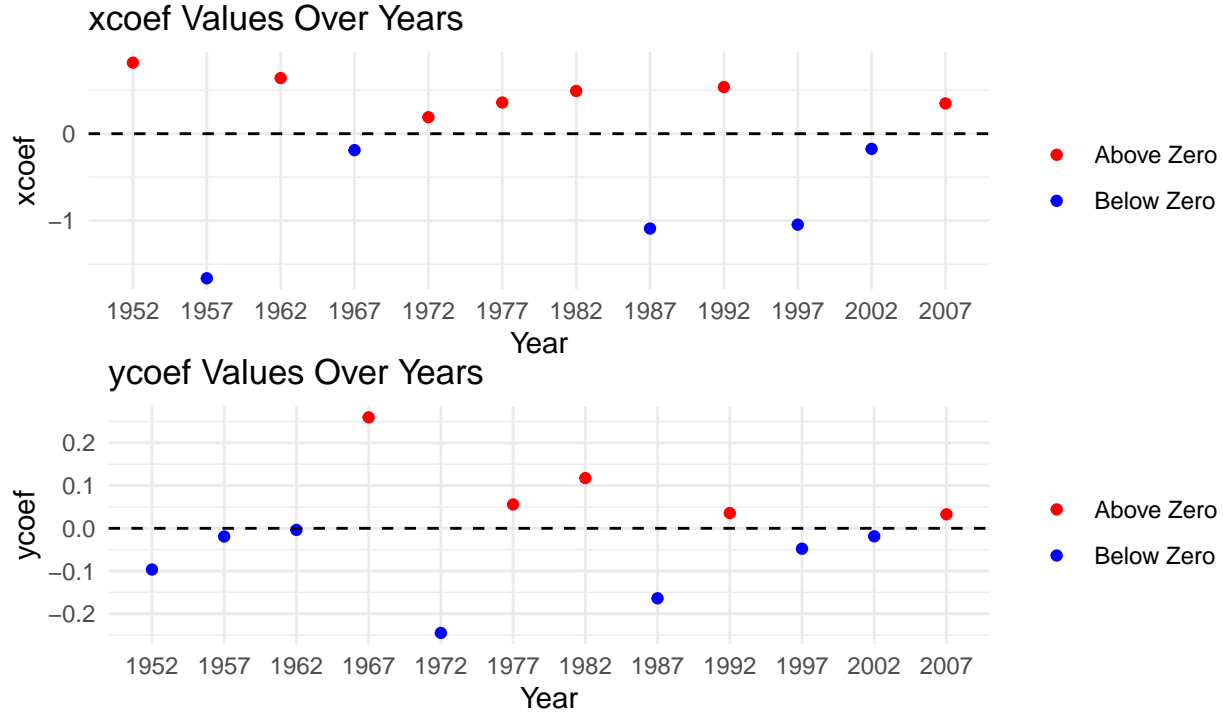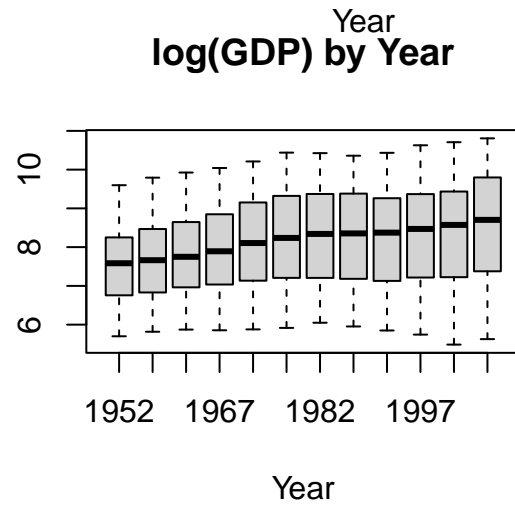

We see that for log(gdp), the higher the value in the red dots mean higher value in the η_1 and ϕ_1.

1) η_1: The higher the first X scores, the higher the values of log(gdp) in 1952, 1962, 1972-1982, 1992, 2007.

2) ϕ_1: The higher the first Y score, the higher the life expectancy in 1967, 1977,1982,1992 and 2007.

We see that Europe countries generally has lower η_1, meaning that it yields high gdp value overall since the x coefficient has a mean of -0.06(less than 0). Europe countries also have lower ϕ_1, meaning that it yields higher life expectancy since the y coefficient has a mean of -0.008(less than 0). The African countries yields the opposite conclusion.

Why log(gdp)?

To explain why we apply log(gdp) instead of gpd here, we need to look back at the exploratory analysis.

Ave. GDP by Year

GDP by Year

Ave. log(GDP) by Year

log(GDP) by Year
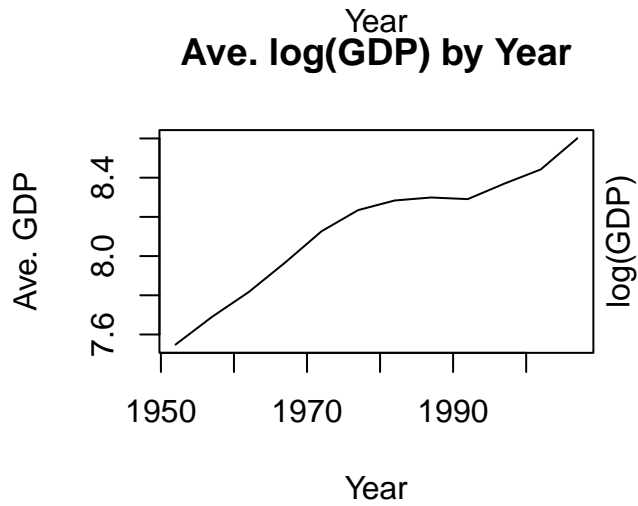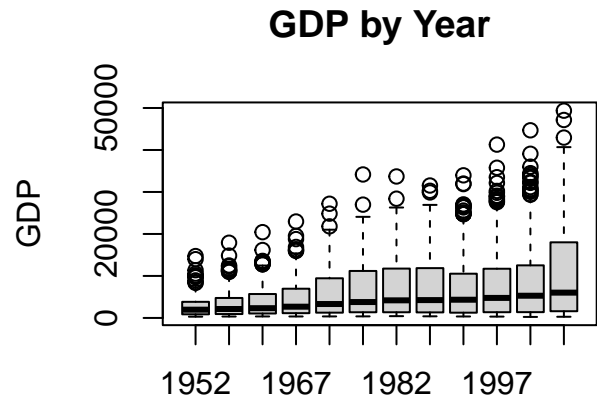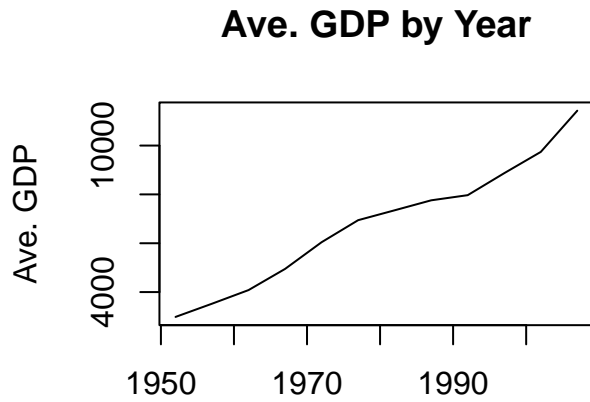
The above boxplots and lineplots show that log(gdp) removes the extreme values and the spread of data doesn't increase too dramatically. This helps make reduce the distribution less skewed, which is helpful for the CCA.

Multidimensional scaling
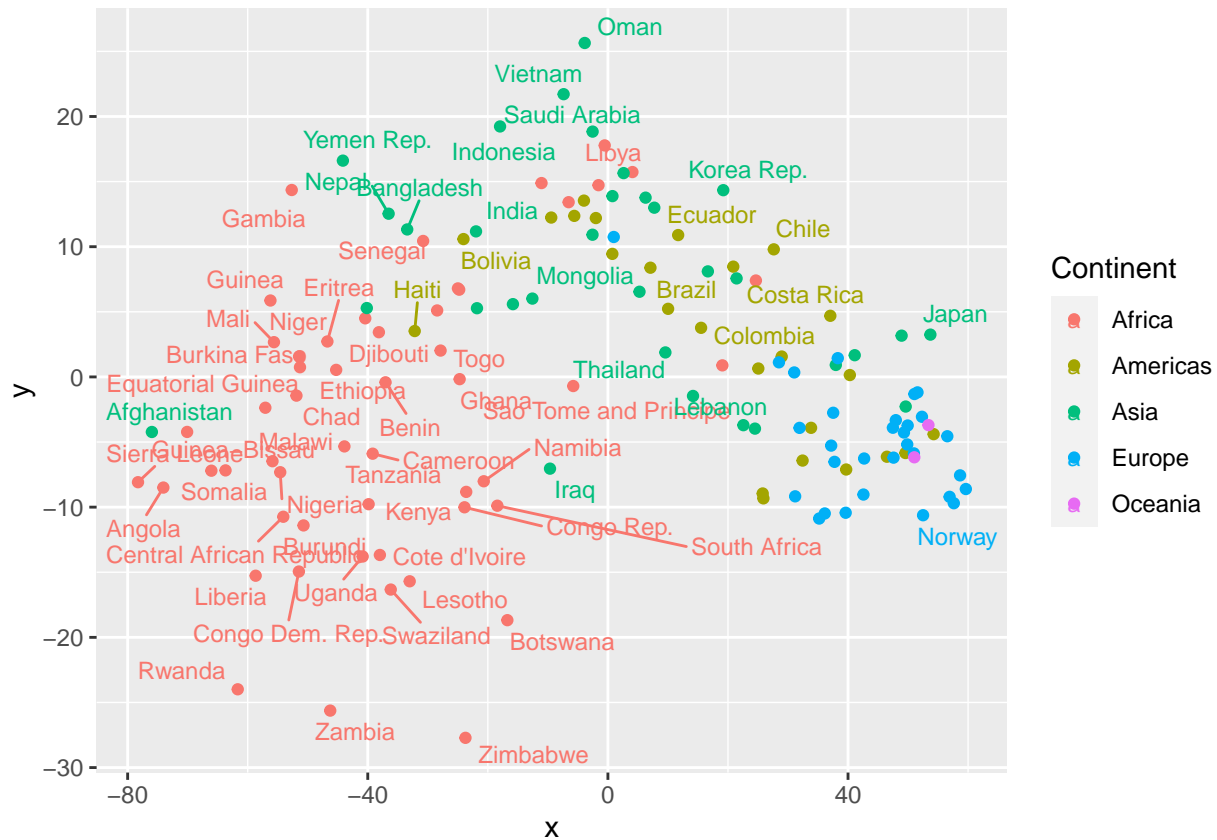
It can be seen that data points are clustered based on continents, African countries are mostly located at the left bottom while the European countries are located at the right center.

This shows that countries from the same continent tends to share similar properties as they clustered together.

Linear Discriminant Analysis

Since we are predicting the continent of each country, let's first visualize out sample data.
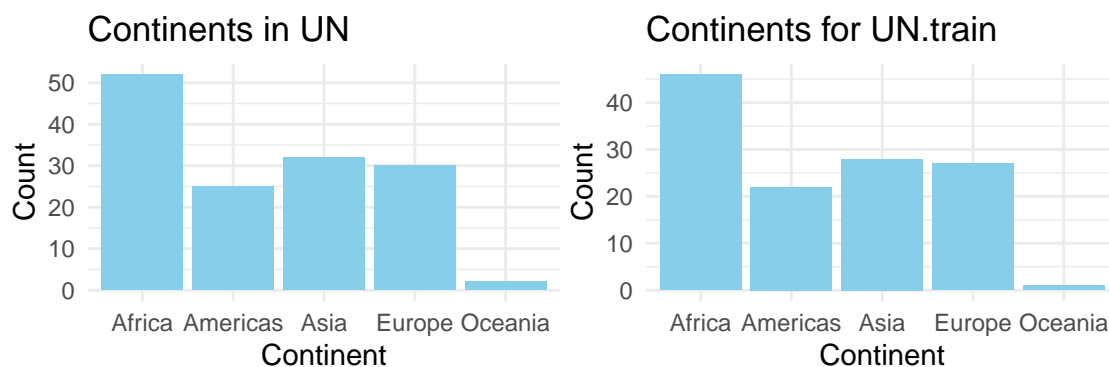

```
## [1] "The number of occurrences of Oceania in UN is 2"
```

```
## [1] "The number of occurrences of Oceania in UN.train is 1"
```

This plot shows that we have only 2 observation of Oceania. Therefore I implement stratified sampling technique from the caret package to ensure that the train-test split is well partitioned and guarantees at least

one observation of Oceania is included into the training set

Now we fit the lda predictor and the result is given as follows:

```
## [1] "The predictive accuracy is  64.7058823529412 %"
```

```
##
##           Africa Americas Asia Europe Oceania
##   Africa       6        0    1      0       0
##   Americas     0        1    1      0       0
##   Asia         0        2    2      0       0
##   Europe       0        0    0      2       1
##   Oceania      0        0    0      1       0
```

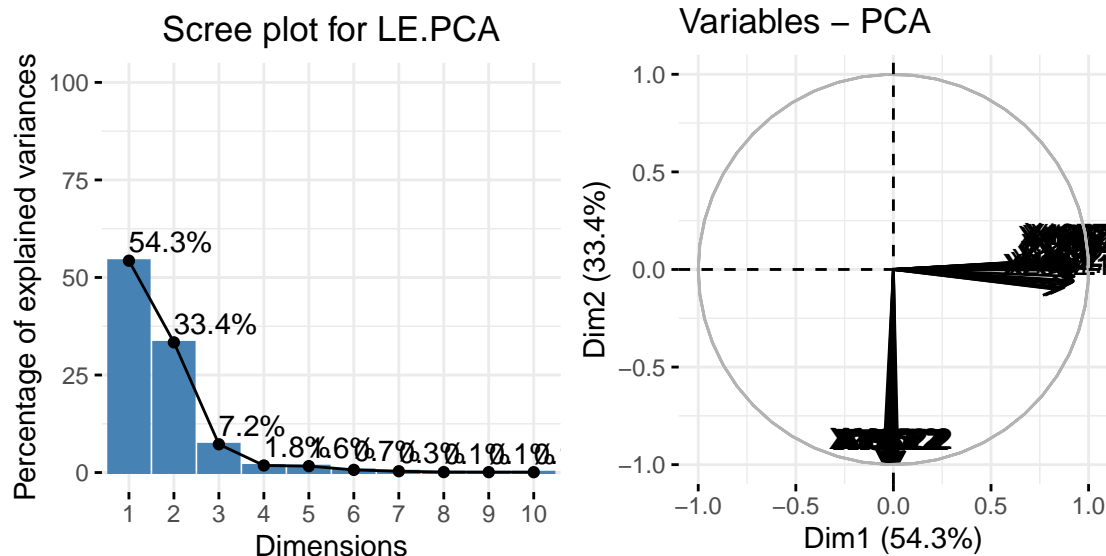The above table shows that 1 Oceania country has been mis-classified as Europe, 6 Asia countries has been mis-classified as Africa/America and 1 American country is mis-classified as Asia.

64% prediction accuracy is unacceptable for a predictive model. Therefore I decide to perform PCA on the dataset before performing LDA.

Based on the scree plot above, I decided to train the LDA predictor based on the first 3 principle components. Using the same train_test split obtained earlier, the PCA based lda predictor shows an increased accuracy of 76%, which is improved compared with the previous non-PCA measure.

```
## [1] "The predictive accuracy is  76.4705882352941 %"
```

```
##
##           Africa Americas Asia Europe Oceania
##   Africa       5        0    0      0       0
##   Americas     0        2    1      0       0
##   Asia         1        1    3      0       0
##   Europe       0        0    0      3       1
##   Oceania      0        0    0      0       0
```

Clustering

Since we have five different continents, therefore we cut off the tree at depth=5 and check which clustering technique can distinguish the continents the best.

9

Optimal number of clusters
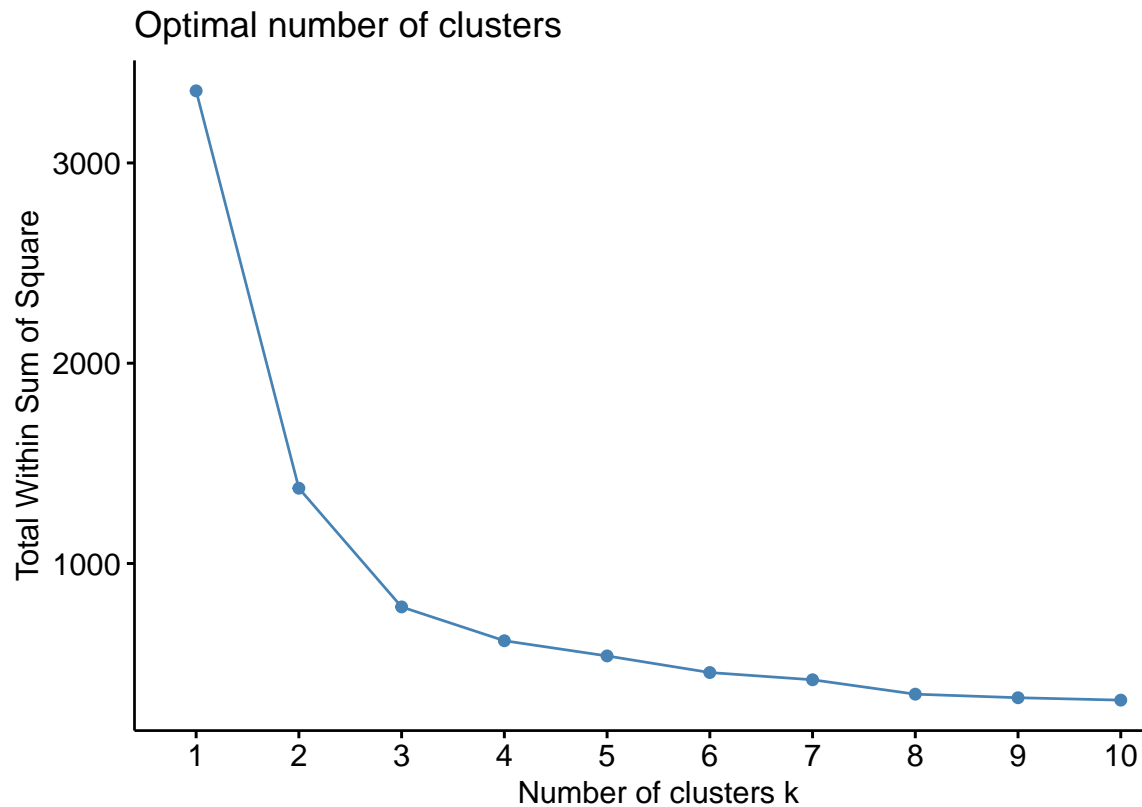
1) K-means clustering

Based on the elbow method, it seems to have three natural clusters in the data, because the Wss decrease rapidly from 2 to 3, and later parts only yields minor improvements.

```
## 
##      Africa Americas Asia Europe Oceania
##    1     10       21   16     12       0
##    2     42        2   10      0       0
##    3      0        2    6     18       2
```

```
## 
##      Africa Americas Asia Europe Oceania
##    1      3       14    7      8       0
##    2     16        7   14      1       0
##    3     33        1    5      0       0
##    4      0        1    4     10       0
##    5      0        2    2     11       2
```

2) Model-based clustering : Gaussian clusters

Since how to choose K for model-based clustering is beyond the scope of this module, I randomly choose G = 5 ^_^ (Because we got five different continents and hopefully it can cluster the sets into different continents)

```r
table(UN.m$classification, UN$continent)
```

```
## 
##      Africa Americas Asia Europe Oceania
##    1      8       14   16      3       0
##    2     35        1    5      0       0
##    3      9        0    4      0       0
##    4      0        8    7     15       0
##    5      0        2    0     12       2
```

Once again it successfully clustered out a group which is Africa.

1) Hierarchical clustering with single linkage

```
##
## Group_num Africa Americas Asia Europe Oceania
##         1     50       25   30     30       2
##         2      1        0    0      0       0
##         3      1        0    0      0       0
##         4      0        0    1      0       0
##         5      0        0    1      0       0
```

2) Hierarchical clustering with complete linkage

```
##
## Group_num Africa Americas Asia Europe Oceania
##         1     10       14   19      3       0
##         2     42        2    7      0       0
##         3      0        7    4     16       0
##         4      0        2    1     11       2
##         5      0        0    1      0       0
```

3) Hierarchical clustering with Ward's method

```
##
## Group_num Africa Americas Asia Europe Oceania
##         1      5        6   14      1       0
##         2     42        2    5      0       0
##         3      3        0    4      0       0
##         4      2       15    5     11       0
##         5      0        2    4     18       2
```

4) Hierarchical clustering with average linkage

```
##
## Group_num Africa Americas Asia Europe Oceania
##         1     10       21   19     12       0
##         2     42        2    7      0       0
##         3      0        1    5     16       2
##         4      0        1    0      2       0
##         5      0        0    1      0       0
```
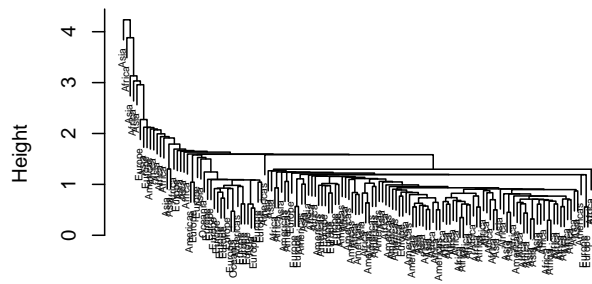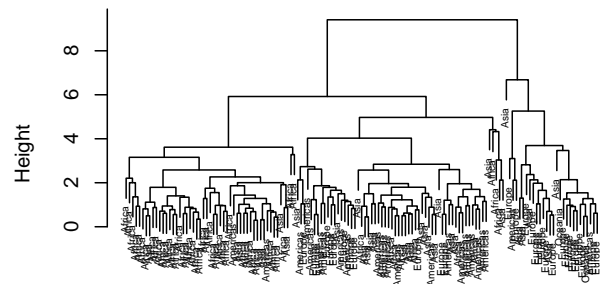
Cluster Dendrogram

dist(UN.scaled[, 3:26], method = "euclidean")
hclust (*, "single")

Cluster Dendrogram

dist(UN.scaled[, 3:26], method = "euclidean")
hclust (*, "average")

Cluster Dendrogram
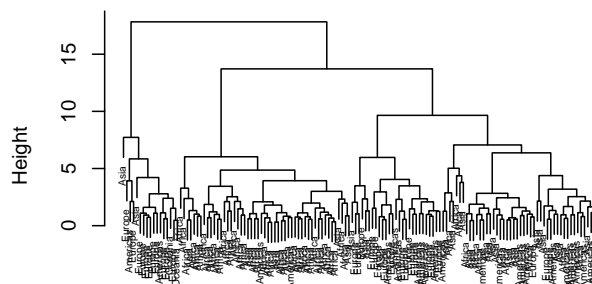
dist(UN.scaled[, 3:26], method = "euclidean")
hclust (*, "complete")

Cluster Dendrogram
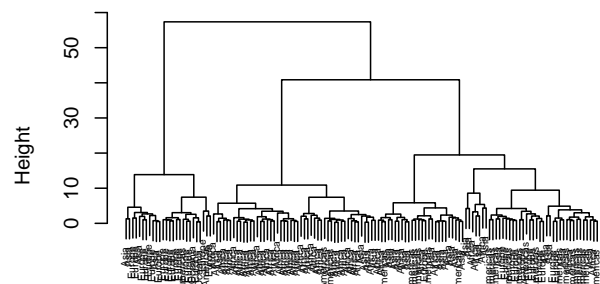
dist(UN.scaled[, 3:26], method = "euclidean")
hclust (*, "ward.D2")

From the above tables, it can be checked that the Hierarchical clustering with Ward's method is the best, as group_2 is dominanted with Africa countries, while group_5 is dominated by Europe countries.

This means that Africa and Europe contries tend to be closer to each other under the Ward's method.

Linear regression

1) OLS

```
x <- UN[,3:14]
y <- UN[,26]
data <- cbind(x,y)
le.lm <- lm(y~., data=data)
le.lm
```

```
##
## Call:
## lm(formula = y ~ ., data = data)
##
## Coefficients:
##     (Intercept)  gdpPercap_1952  gdpPercap_1957  gdpPercap_1962  gdpPercap_1967
##      58.5547252      -0.0014463       0.0045240      -0.0038932      -0.0001022
## gdpPercap_1972  gdpPercap_1977  gdpPercap_1982  gdpPercap_1987  gdpPercap_1992
```

```
##      0.0013214      -0.0010691       0.0011905       0.0003911      -0.0017830
## gdpPercap_1997  gdpPercap_2002  gdpPercap_2007
##      0.0022685      -0.0014386       0.0007789
```

2) PCR (principle component regression)

```
x.pca <- prcomp(x)
PCscores <- x.pca$x
le.pcr <- lm(y~PCscores,data = data.frame(cbind(y,PCscores)))
coef(le.pcr)
```

```
##   (Intercept)    PCscoresPC1    PCscoresPC2    PCscoresPC3    PCscoresPC4
##  6.693238e+01   3.152536e-04  -1.107914e-04   3.406986e-05  -2.775251e-04
##   PCscoresPC5    PCscoresPC6    PCscoresPC7    PCscoresPC8    PCscoresPC9
## -7.372128e-04   1.290469e-04  -1.971468e-03   1.449776e-03   1.597800e-03
##  PCscoresPC10   PCscoresPC11   PCscoresPC12
##  2.379535e-03  -2.013464e-03   5.846231e-03
```

```
le.pcr.coef <- x.pca$rotation %*% coef(le.pcr)[-1]
head(le.pcr.coef)
```

```
##                          [,1]
## gdpPercap_1952 -0.0014463346
## gdpPercap_1957  0.0045239681
## gdpPercap_1962 -0.0038931800
## gdpPercap_1967 -0.0001022091
## gdpPercap_1972  0.0013213713
## gdpPercap_1977 -0.0010690524
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
##
##     R2

## The following object is masked from 'package:stats':
##
##     loadings

## Data:    X dimension: 141 12
##  Y dimension: 141 1
## Fit method: svdpc
## Number of components considered: 12
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           12.13    8.901    8.890    8.945    8.999    8.973    9.064
## adjCV        12.13    8.896    8.885    8.937    8.988    8.961    9.049
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps
## CV       8.983    9.398    9.420     9.567     9.739     9.538
## adjCV    8.960    9.355    9.372     9.502     9.660     9.469
##
## TRAINING: % variance explained
##     1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X     93.96    98.44    99.16    99.52    99.74    99.85    99.90    99.94
## y     46.44    46.72    46.72    46.86    47.44    47.45    48.52    48.94
```

```
##     9 comps  10 comps  11 comps  12 comps
## X    99.97     99.98    100.00    100.00
## y    49.26     49.67     49.93     50.66
```

y
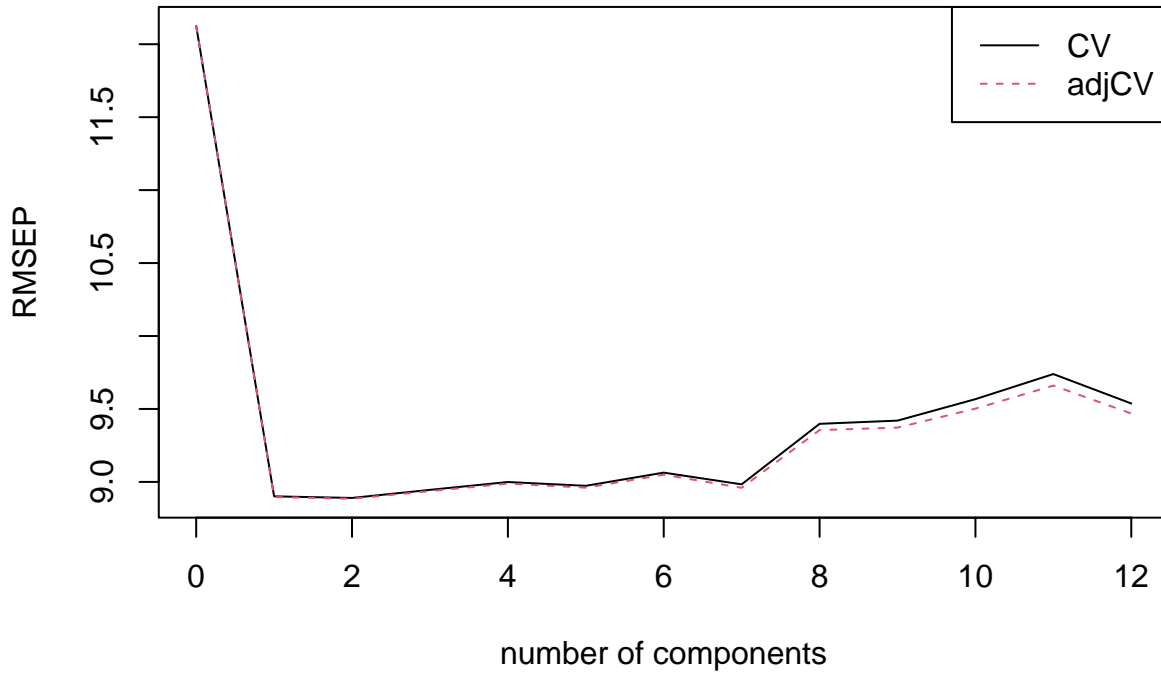
From the above figure, it looks like 1 component is sufficient. Using more than 1 component doesn't lead to big decreases in the cross-validation error and even lead to larger prediction errors.

   3) Ridge regression

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:spam':
##
##     det
```

```
## Loaded glmnet 4.1-7
```
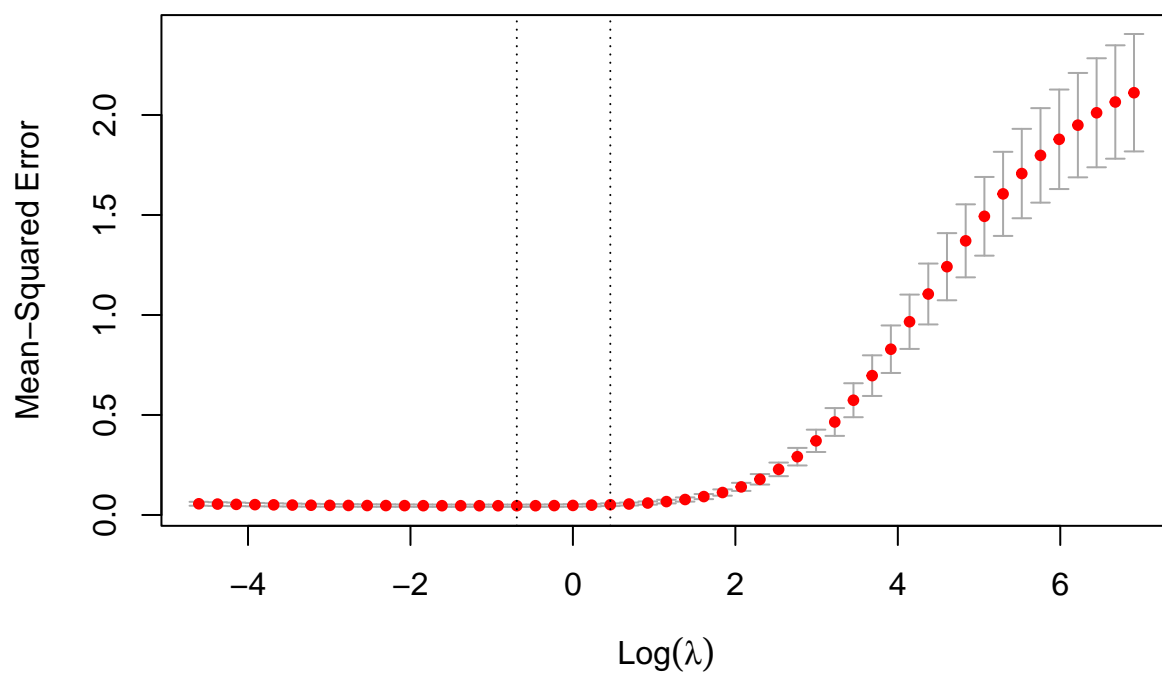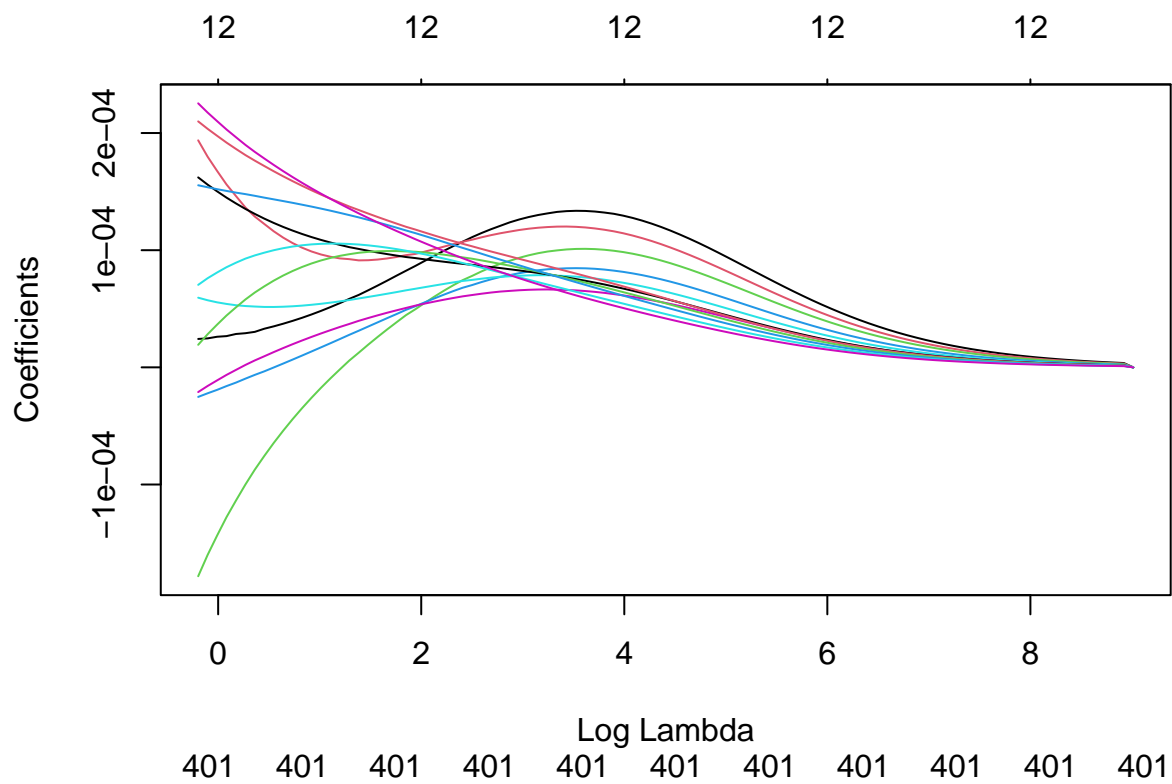
```
## [1] 0.5011872
```

```
## [1] 1.584893
```

```
coef(glmnet(x, y, alpha=0, lambda=cv_fit$lambda.1se))
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                    s0
```

```
## (Intercept)      5.925656e+01
## gdpPercap_1952  3.131674e-05
## gdpPercap_1957  1.288819e-04
## gdpPercap_1962 -7.105767e-05
## gdpPercap_1967 -3.017010e-06
## gdpPercap_1972  4.981351e-05
## gdpPercap_1977  7.622660e-06
## gdpPercap_1982  1.251719e-04
## gdpPercap_1987  1.720270e-04
## gdpPercap_1992  7.059357e-05
## gdpPercap_1997  1.450368e-04
## gdpPercap_2002  9.681959e-05
## gdpPercap_2007  1.778978e-04
```

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
UN <- read.csv('UN.csv')

gdp <- UN[,3:14] # The GDP per capita.
years <- seq(1952, 2007,5)
colnames(gdp) <- years
rownames(gdp) <- UN[,2]

lifeExp <- UN[,15:26] # the life expectancy
colnames(lifeExp) <- years
rownames(lifeExp) <- UN[,2]

popn <- UN[,27:38] # the population size
colnames(popn) <- years
rownames(popn) <- UN[,2]

library(ggplot2)
library(reshape2)
average_gdp <- apply(gdp, 2, mean)
plot(years, average_gdp, type = "l", xlab = "Year", ylab = "Ave. GDP", main = "Ave. GDP by Year")

boxplot(gdp, names = years, main = "GDP by Year", xlab = "Year", ylab = "GDP")

average_le <- apply(lifeExp, 2, mean)
plot(years, average_le, type = "l", xlab = "Year", ylab = "Ave. Life Expectancy", main = "Ave. Life Exp

boxplot(lifeExp, names = years, main = "Life Expectancy by Year", xlab = "Year", ylab = "Life Expectancy


average_popn <- apply(popn, 2, mean)
plot(years, average_popn, type = "l", xlab = "Year", ylab = "Ave. Population", main = "Ave. Population

boxplot(log(popn), names = years, main = "log(Population) by Year", xlab = "Year", ylab = "log(Populatic
library(ggplot2)  # Make sure ggplot2 is loaded
gdp.pca <- prcomp(gdp, scale=TRUE)
le.pca <- prcomp(lifeExp, scale=TRUE)
popn.pca <- prcomp(popn, scale=TRUE)
```

```r
#summary(gdp.pca)
#gdp.pca$rotation # the loadings/eigenvectors
#gdp.pca$center  # the sample mean
library(factoextra)
library(ggrepel)
#fviz_eig(gdp.pca, addlabels = TRUE, ylim = c(0, 100)) #Scree plot

plot <- fviz_eig(gdp.pca, addlabels = TRUE, ylim = c(0, 100)) + ggtitle("Scree plot for GDP.PCA")+ theme
print(plot)

plot <- fviz(gdp.pca, element = "var") + ggtitle("Biplot for GDP.PCA")+ theme(plot.title = element_text
print(plot)

#fviz_eig(le.pca, addlabels = TRUE, ylim = c(0, 100)) #Scree plot

fviz_eig(le.pca, addlabels = TRUE, ylim = c(0, 100)) + ggtitle("Scree plot for LE.PCA")+ theme(plot.tit

fviz(le.pca, element='var') #Interpretation of leading PC

#fviz_eig(le.pca, addlabels = TRUE, ylim = c(0, 100)) #Scree plot

fviz_eig(popn.pca, addlabels = TRUE, ylim = c(0, 100)) + ggtitle("Scree plot for Popn.PCA")+ theme(plot

fviz(popn.pca, element='var') #Interpretation of leading PC
#fviz(gdp.pca, element='var')#Interpretation of leading PC
pca_data <- data.frame(PC1 = gdp.pca$x[,1], PC2 = gdp.pca$x[,2], Continent = UN$continent)

ggplot(pca_data, aes(x = PC1, y = PC2, color = Continent)) +
  geom_point() + # This adds the scatter plot points
  geom_text_repel(aes(label = UN[,2]), size = 3)+
  labs(title = "GDP PC1 vs. GDP PC2",
       x = "PC1", y = "PC2") +
  theme_minimal()+ # Use a minimal theme for the plot
  scale_color_brewer(palette = "Set1")
#scale(gdp, center = TRUE) %*% gdp.pca$rotation[,1]
pc_loadings <- data.frame(PC1 = round(gdp.pca$rotation[,1],digits = 2), PC2 = round(gdp.pca$rotation[,2]
# If you want to include the variable names as a row names in the table
rownames(pc_loadings) <- rownames(gdp.pca$rotation)

# Display the table
t(pc_loadings)

pca_data <- data.frame(PC1 = le.pca$x[,1], PC2 = le.pca$x[,2], Continent = UN$continent)

ggplot(pca_data, aes(x = PC1, y = PC2, color = Continent)) +
  geom_point() + # This adds the scatter plot points
  geom_text_repel(aes(label = UN[,2]), size = 3)+
  labs(title = "LE PC1 vs. LE PC2",
       x = "PC1", y = "PC2") +
  theme_minimal()+ # Use a minimal theme for the plot
  scale_color_brewer(palette = "Set1")

pc_loadings <- data.frame(PC1 = round(le.pca$rotation[,1],digits = 2), PC2 = round(le.pca$rotation[,2],
```

```r
# If you want to include the variable names as a row names in the table
rownames(pc_loadings) <- rownames(le.pca$rotation)

# Display the table
t(pc_loadings)

pca_data <- data.frame(PC1 = gdp.pca$x[,1], PC2 = le.pca$x[,1], Continent = UN$continent)

ggplot(pca_data, aes(x = PC1, y = PC2, color = Continent)) +
  geom_point() + # This adds the scatter plot points
  geom_text_repel(aes(label = UN[,2]), size = 3)+
  labs(title = "GDP PC1 vs. LE PC1",
       x = "GDP_PC1", y = "LE_PC1") +
  theme_minimal()+ # Use a minimal theme for the plot
  scale_color_brewer(palette = "Set1")

pc_loadings <- data.frame(LE_PC1 = round(gdp.pca$rotation[,1],digits = 2), GDP_PC1 = round(le.pca$rotat:
# If you want to include the variable names as a row names in the table
rownames(pc_loadings) <- rownames(popn.pca$rotation)

# Display the table
t(pc_loadings)
library(CCA)
#How the scores are calculated
#temp <-  scale(log(gdp), center = TRUE, scale =FALSE) #Centering the matrix
#print(temp %*% cca$xcoef[,1])

cca<-cc(log(gdp),lifeExp)
#plt.cc(cca, var.label=FALSE)

# Convert cca scores to a dataframe
scores_df <- data.frame(xscores = cca$scores$xscores[,1],
                        yscores = cca$scores$yscores[,1],
                        row.names = rownames(cca$scores$xscores))
# Assuming you have a dataframe `UN` with a column `continent` that matches the rows of your CCA analys
scores_df$continent <- UN$continent

ggplot(scores_df, aes(x = xscores, y = yscores, color = continent)) +
  geom_point() +
  geom_text_repel(aes(label = rownames(scores_df)), size = 3) +
  labs(x = expression( eta[1] ("First X Score")),
       y = expression( phi[1] ("First Y Score")),
       title = "Scatter Plot of Canonical Scores",
       color = "Continent")
# Plot the data using ggplot2
data <- data.frame(cca$xcoef[,1])
ggplot(data, aes(x = rownames(cca$xcoef), y = cca$xcoef[,1])) +
  geom_point(aes(color = ifelse(cca$xcoef[,1] > 0, "Above Zero", "Below Zero"))) + # Color points based
  scale_color_manual(values = c("Above Zero" = "red", "Below Zero" = "blue")) + # Assign colors
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") + # Add horizontal line at y=0

  labs(title = "xcoef Values Over Years",
       x = "Year",
```

```r
      y = "xcoef") +
  theme_minimal()+ # Use a minimal theme+ # Use a minimal theme
  theme(legend.title = element_blank())

data <- data.frame(cca$ycoef[,1])
ggplot(data, aes(x = rownames(cca$ycoef), y = cca$ycoef[,1])) +
  geom_point(aes(color = ifelse(cca$ycoef[,1] > 0, "Above Zero", "Below Zero"))) + # Color points based
  scale_color_manual(values = c("Above Zero" = "red", "Below Zero" = "blue")) + # Assign colors
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") + # Add horizontal line at y=0
  labs(title = "ycoef Values Over Years",
       x = "Year",
       y = "ycoef") +
  theme_minimal()+ # Use a minimal theme+ # Use a minimal theme
  theme(legend.title = element_blank())
average_gdp <- apply(gdp, 2, mean)
plot(years, average_gdp, type = "l", xlab = "Year", ylab = "Ave. GDP", main = "Ave. GDP by Year")

boxplot(gdp, names = years, main = "GDP by Year", xlab = "Year", ylab = "GDP")

average_gdp <- apply(log(gdp), 2, mean)
plot(years, average_gdp, type = "l", xlab = "Year", ylab = "Ave. GDP", main = "Ave. log(GDP) by Year")

boxplot(log(gdp), names = years, main = "log(GDP) by Year", xlab = "Year", ylab = "log(GDP)")

head(cca$scores$xscores[,1]) # the canonical correlation variables
library(dplyr)
library(ggpubr) # repels figure labels
UN.transformed <- cbind(log(UN[,3:14]), UN[,15:26], log(UN[,27:38]))
UN.transformed <- dist(UN.transformed)
UN.transformed <- cmdscale(UN.transformed)
UN.transformed <- data.frame(UN.transformed,
                       row.names = rownames(cca$scores$xscores))
colnames(UN.transformed) <- c("x", "y")

UN.transformed$continent <- UN$continent

ggplot(UN.transformed, aes(x = x, y = y, color = continent)) +
    geom_point() +  # This will color the points based on continent
    geom_text_repel(aes(label = row.names(UN.transformed)), size = 3) +
    labs(color = "Continent")  # Labeling the color legend as "Continent"
library(caret)
# If 'continent' is not a factor, convert it to factor
temp <- UN
temp$continent <- as.factor(temp$continent)

set.seed(123)  # for reproducibility
# Creating indices for a stratified sample
test.index <- createDataPartition(UN$continent, p = 0.1, list = FALSE)
#set.seed(123) # so that I get the same results each time.
#test.index <- sample(1:141, size=20)
UN.test <- UN[test.index,]
UN.train <- UN[-test.index,]
# Plotting the data
```

```r
ggplot(UN, aes(x = continent)) +  # Specify where the data comes from and which variable to plot
  geom_bar(fill = "skyblue") +  # This automatically counts the number of each unique value in the cont
  labs(title = "Continents in UN",
       x = "Continent",
       y = "Count") +  # Adding labels
  theme_minimal()  # Using a minimalistic theme


# Plotting the data
ggplot(UN.train, aes(x = continent)) +  # Specify where the data comes from and which variable to plot
  geom_bar(fill = "skyblue") +  # This automatically counts the number of each unique value in the cont
  labs(title = "Continents for UN.train",
       x = "Continent",
       y = "Count") +  # Adding labels
  theme_minimal()  # Using a minimalistic theme

print(paste("The number of occurrences of Oceania in UN is", table(UN$continent)["Oceania"]))


print(paste("The number of occurrences of Oceania in UN.train is", table(UN.train$continent)["Oceania"]
UN.lda<-lda(continent ~ gdpPercap_1952+gdpPercap_1957+gdpPercap_1962+gdpPercap_1967+gdpPercap_1972+gdpP
UN.pred <- predict(UN.lda, UN.test)
print(paste("The predictive accuracy is ",
sum(UN.pred$class== UN.test$continent)/dim(UN.test)[1]*100, "%"))
table(UN.pred$class, UN.test$continent)
library(ggplot2)  # Make sure ggplot2 is loaded
all <- data.frame(gdp, lifeExp, popn)

# Perform Principal Component Analysis on the combined dataset
all.pca <- prcomp(all, scale = TRUE)

fviz_eig(all.pca, addlabels = TRUE, ylim = c(0, 100)) + ggtitle("Scree plot for LE.PCA")+ theme(plot.ti

fviz(all.pca, element='var') #Interpretation of leading PC
all.test <- data.frame(all.pca$x[test.index,1:6])
all.train <- data.frame(all.pca$x[-test.index,1:6])

all.train$continent <- UN$continent[-test.index]
all.test$continent <- UN$continent[test.index]

all.lda<-lda(continent ~ PC1+PC2+PC3+PC4+PC5+PC6, all.train)
all.pred <- predict(all.lda,all.test)
print(paste("The predictive accuracy is ",
sum(all.pred$class== all.test$continent)/dim(all.test)[1]*100, "%"))

table(all.pred$class, all.test$continent)
UN.scaled <- UN[,1:26]
UN.scaled[,3:26] <- scale(UN[,3:26])
library(factoextra)
fviz_nbclust(UN.scaled[,3:26], kmeans, method = "wss")
set.seed(123)
UN.k <- kmeans(UN.scaled[,3:26], centers = 3, nstart=25)
table(UN.k$cluster, UN$continent)
```

```r
set.seed(123)
UN.k <- kmeans(UN.scaled[,3:26], centers = 5, nstart=25)
table(UN.k$cluster, UN$continent)
library(mclust)
UN.m <- Mclust(UN.scaled[,3:26],G=5)
#plot(UN.m, what = c("classification"))
table(UN.m$classification, UN$continent)
UN.single <- hclust(dist(UN.scaled[,3:26],method="euclidean"),method="single")
#plot(UN.single, labels=UN$continent,cex=0.2)
Group_num <- cutree(UN.single, k=5)
table(Group_num, UN$continent)
UN.complete <- hclust(dist(UN.scaled[,3:26],method="euclidean"),method="complete")
#plot(UN.complete, labels=UN$continent,cex=0.2)
Group_num <- cutree(UN.complete, k=5)
table(Group_num, UN$continent)
UN.ward <- hclust(dist(UN.scaled[,3:26],method="euclidean"),method="ward.D2")
#plot(UN.ward, labels=UN$continent,cex=0.2)
Group_num <- cutree(UN.ward, k=5)
table(Group_num, UN$continent)
UN.average <- hclust(dist(UN.scaled[,3:26],method="euclidean"),method="average")
#plot(UN.average, labels=UN$continent,cex=0.2)
Group_num <- cutree(UN.average, k=5)
table(Group_num, UN$continent)
par(mfrow=c(2,2))
plot(UN.single, labels = UN$continent, cex=0.5)
plot(UN.average, labels = UN$continent, cex=0.5)
plot(UN.complete, labels = UN$continent, cex=0.5)
plot(UN.ward, labels = UN$continent, cex=0.5)
x <- UN[,3:14]
y <- UN[,26]
data <- cbind(x,y)
le.lm <- lm(y~., data=data)
le.lm
x.pca <- prcomp(x)
PCscores <- x.pca$x
le.pcr <- lm(y~PCscores,data = data.frame(cbind(y,PCscores)))
coef(le.pcr)
le.pcr.coef <- x.pca$rotation %*% coef(le.pcr)[-1]
head(le.pcr.coef)
library(pls)
le.pcr_pls <- pcr(y~., data=data, validation = 'CV')
summary(le.pcr_pls)

plot(RMSEP(le.pcr_pls), legendpos = "topright")
library(glmnet)
le.ridge <- glmnet(x,y,alpha = 0)
plot(le.ridge,xvar='lambda')
lambdas <- 10^seq(3,-2,by=-0.1)
cv_fit <- cv.glmnet(gasoline$NIR, gasoline$octane, alpha = 0, lambda = lambdas)
plot(cv_fit)
cv_fit$lambda.min
cv_fit$lambda.1se
coef(glmnet(x, y, alpha=0, lambda=cv_fit$lambda.1se))
```