University of Nottingham
Department of Mathematical Sciences
MATH 4068- Multivariate Analysis

Coursework, Spring 2024

by

Raonak Shukla
Student Id-20601487

"If I can't picture it, I can't understand it." ***- Albert Einstein***

I have read and understood the School and University guidelines on plagiarism. I confirm that this work is our own, apart from the acknowledged references.

```
# Importing the file and major Libraries for our analysis
setwd('C:\\Users\\raona\\OneDrive\\Documents\\Multivariate')
install.packages("factoextra",repos = "https://cran.r-project.org/")
install.packages('glmnet',repos = "https://cran.r-project.org/")
install.packages('pls',repos = "https://cran.r-project.org/")
install.packages('pls',repos = "https://cran.r-project.org/")
install.packages('rgl',repos = "https://cran.r-project.org/")
install.packages("ggordiplots",repos = "https://cran.r-project.org/")
install.packages('caTools',repos = "https://cran.r-project.org/")
install.packages('klaR',repos = "https://cran.r-project.org/")
install.packages('ggfortify',repos = "https://cran.r-project.org/")
```

```
library(ggfortify)
library(ggrepel)
library(caTools)
library(klaR)
library(ggordiplots)
library(factoextra)
library(glmnet)
library(pls)
library(ggplot2)
library(tidyr)
library(dplyr)
library(glmnet)
library(CCA)
```

```
#Importing the dataset
UN <- read.csv('UN.csv')
continent <- UN$continent

# Slicing the columns with GDP
gdp <- UN[,c(1,3:14)] # The GDP per capita.
years <- seq(1952, 2007,5)
con <- "continent"
colnames(gdp) <- append(con,as.character(years))
rownames(gdp) <- UN[,2]

# Slicing the columns with Life Expectancy
lifeExp <- UN[,c(1,15:26)] # the life expectancy
colnames(lifeExp) <- append(con,as.character(years))
rownames(lifeExp) <- UN[,2]

# Slicing the columns with Population Size
popn <- UN[,c(1,27:38)] # the population size
colnames(popn) <- append(con,as.character(years))
rownames(popn) <- UN[,2]
```

Exploratory Data Analysis(EDA)

The UN dataset consists of the Gross Domestic Product(GDP), Life Expectancy, and Population data of 142 countries from 1952 to 2007 collected every 5 years. In this section, the summary statistics and relation between GDP, Life Expectancy, and Population will be discussed for the years 1952 and 2007 with the help of plots. As shown in the table below it can be inferred that over all for the whole world in 55 years the Gross Domestic Product(GDP) per capita, Life Expectancy, and Population increased by 283.08%, 36.54%, and 159.62% respectively.

```
UN %>% group_by(continent) %>%
summarise(pc_gdp =
 (mean(gdpPercap_2007)-mean(gdpPercap_1952))*100/mean(gdpPercap_1952),
pc_life=(mean(lifeExp_2007)-mean(lifeExp_1952))*100/mean(lifeExp_1952),
pc_pop=(mean(pop_2007)-mean(pop_1952))*100/mean(pop_1952))
```

Table 1: Percentage change from year 1952 to 2007

Continent	GDP(per capita)	Life Expectancy	Population
Africa	146.51	40.04	291.15
America	169.74	38.15	160.43
Asia	477.63	53.21	173.04
Europe	342.58	20.56	40.17
Oceanian	189.47	16.55	129.74
World	283.08	36.54	159.62

Continent-wise, the picture is a bit different, the growth in GDP and population was highest for America and lowest for Africa. Population growth was minimal in Europe. This shows a growing disparity between the least developed and developed nations. It is also evident from the figure below.

```
#EDA Life expectancy Vs GDP for 2007
ggplot(UN, aes(x=gdpPercap_2007,y=lifeExp_2007,col=continent,size=pop_2007))+
    geom_point()+geom_text_repel(label=UN$country,size=1)+
        labs(x= 'GDP for the year 2007',y='Life Expectancy for year 2007')
```

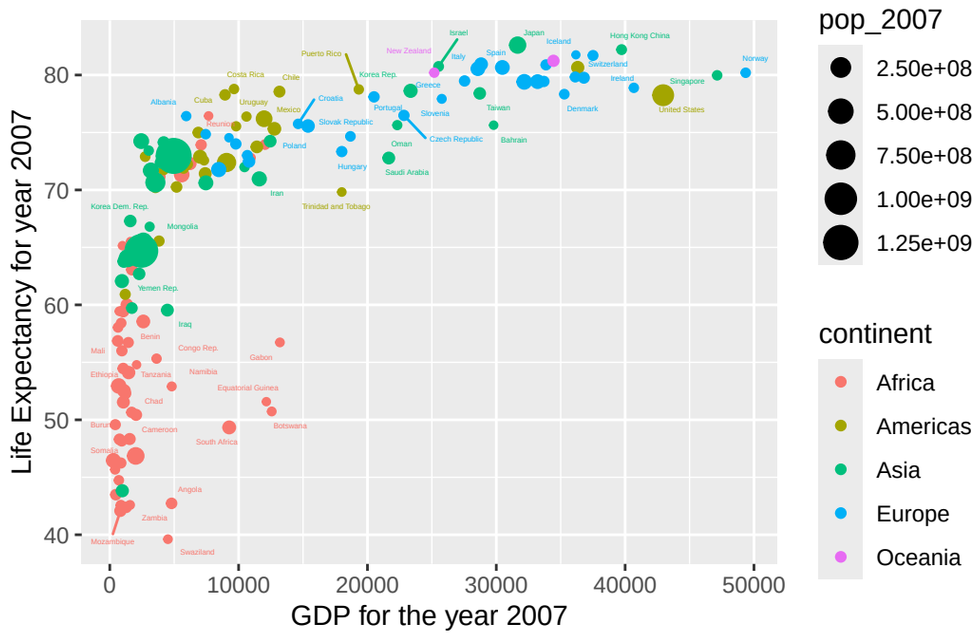


Figure 1: GDP Vs. Life Expectancy for 2007

```
#EDA Life expectancy Vs GDP for 1952
ggplot(UN, aes(x=gdpPercap_1952,y=lifeExp_1952,col=continent,size=pop_1952))+
    geom_point()+geom_text_repel(label=UN$country)+
        labs(x= 'GDP for the year 1952',y='Life Expectancy for year 1952')
```

The two plots(Figure 1 and 2) try to compare GDP and Life Expectancy for the years 1957 and 2007. It can be construed that almost all countries have developed in the span of 50 years. This shows improvement in the standard of living and medical research. However, countries with geo-political instability like Afghanistan and countries in West and Central Africa have slipped in growth indicators.

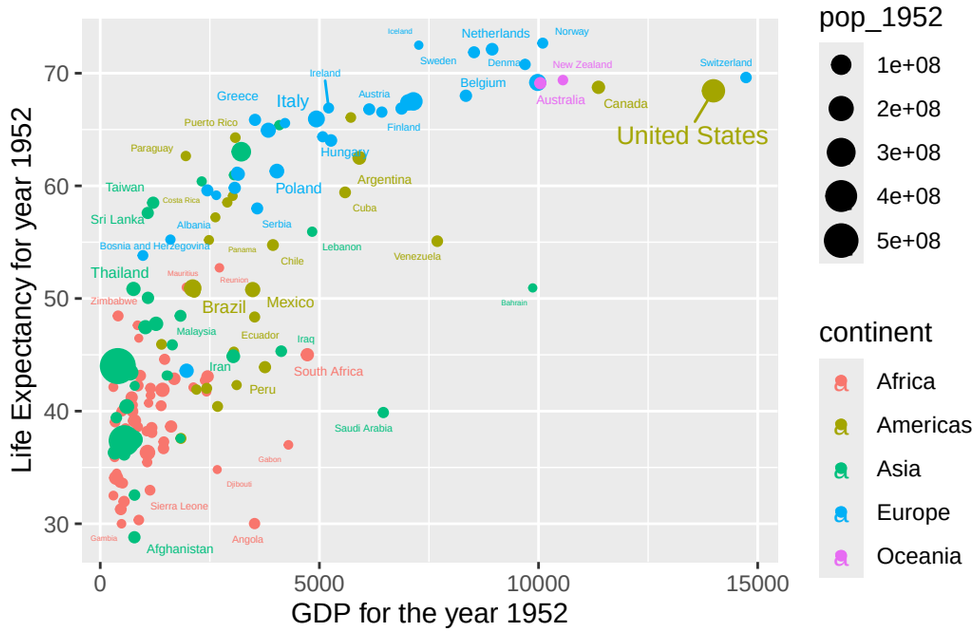


Figure 2: GDP Vs. Life Expectancy for 1952

The next table documents the mean GDP, Life Expectancy, and Population for the year 2007. It is evident that development is not inclusive and we are very far away from UN Sustainable Development Goals(SDGs).

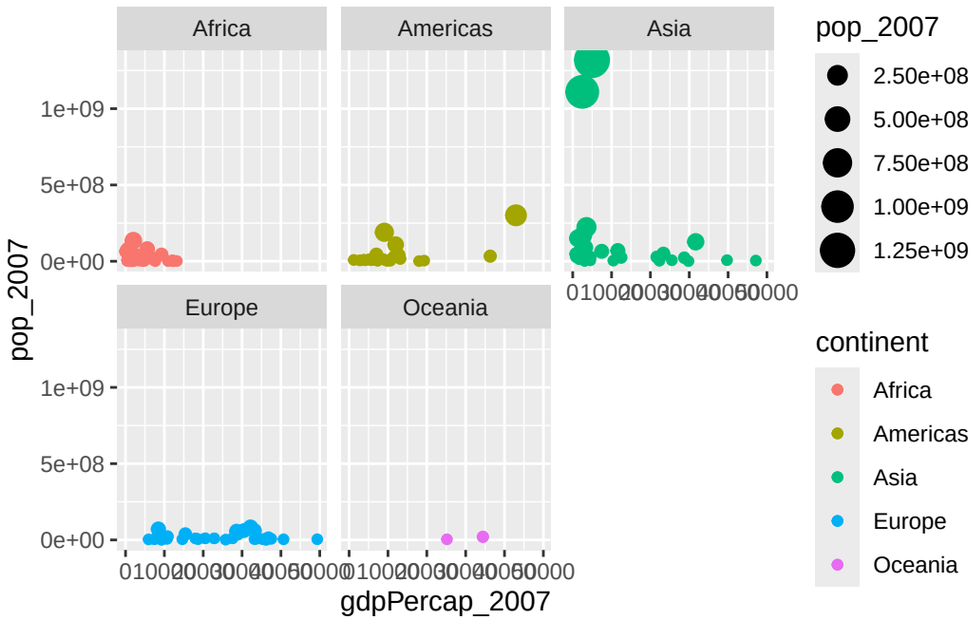
```
UN %>% group_by(continent) %>%
    summarise(mean_gdpPercap_2007=mean(gdpPercap_2007),mean_lifeExp_2007=
        mean(lifeExp_2007),mean_pop_2007=mean(pop_2007)/10**6)
```

Table 2: Mean GDP, Life Expectancy and Population for 2007

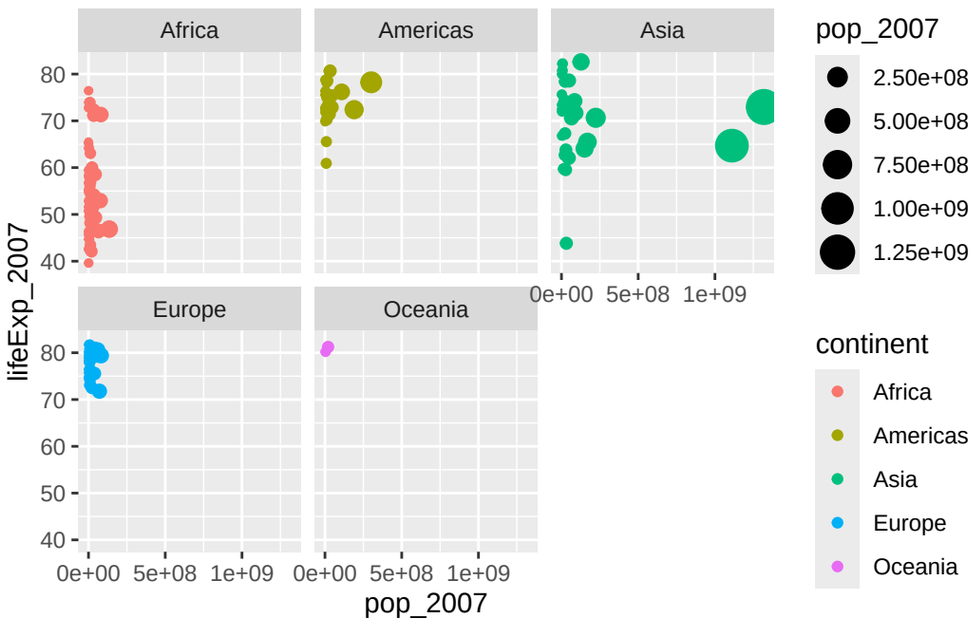
Continent	GDP(per capita)	Life Expectancy	Population(in millions)
Africa	3089.033	54.80	17.87
America	11003.032	73.60	35.95
Asia	11384.466	70.51	119.04
Europe	25054.482	77.65	19.53
Oceania	29810.188	80.72	12.27
World	11427.4	66.93	44.31

```
#EDA Population Vs GDP for 2007
ggplot(UN, aes(x=gdpPercap_2007,y=pop_2007,col=continent))+
    geom_point()+geom_point(aes(color=continent,size=pop_2007))+
        facet_wrap(~continent)
```

The plots below shows GDP vs. Life Expectancy for 2007 for all five continents separately. It shows two countries in Asia India and China have a huge population and comparatively low GDP putting huge burden on the resources. Countries in Europe and Oceania have low populations and high GDP making room to allocate resources for the development of social and human capital. African countries have low populations and GDP making the development process difficult without any support from the United Nation. If we plot Life expectancy vs. Population it can be inferred that Oceania has the highest average Life Expectancy as a continent followed by Europe and Africa has the lowest Life Expectancy. This can be attributed to the fact that African countries have low GDPs (percapita) making it difficult to allocate resources for social and human capital.




```
#EDA Life Expectancy Vs Population for 2007
ggplot(UN, aes(y=lifeExp_2007,x=pop_2007,col=continent,size=pop_2007))+
    geom_point(aes(color=continent,size=pop_2007))+
        facet_wrap(~continent)
```



There is a positive correlation of 70% between GDP and Life Expectancy making it a good pair for regression later on where GDP can act as the independent variables and Life Expectancy as the dependent variable.

Principal Component Analysis

```
#PCA on GDP
X <- gdp[,2:13]
gdp.pca<- prcomp(X,scale=TRUE)
summary(gdp.pca)
```

```
#Visualisation of Principal Components
autoplot(gdp.pca, data = gdp, colour='continent')
```

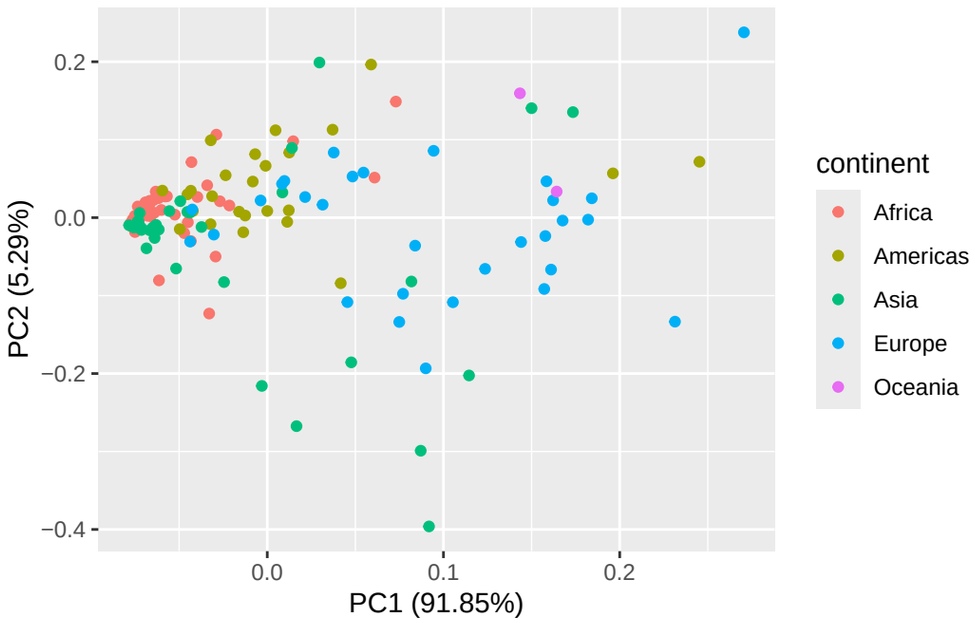


Figure 3: PCA plot based on Continent for GDP

```
fviz_eig(gdp.pca, addlabels = TRUE, ylim = c(0, 100))
```

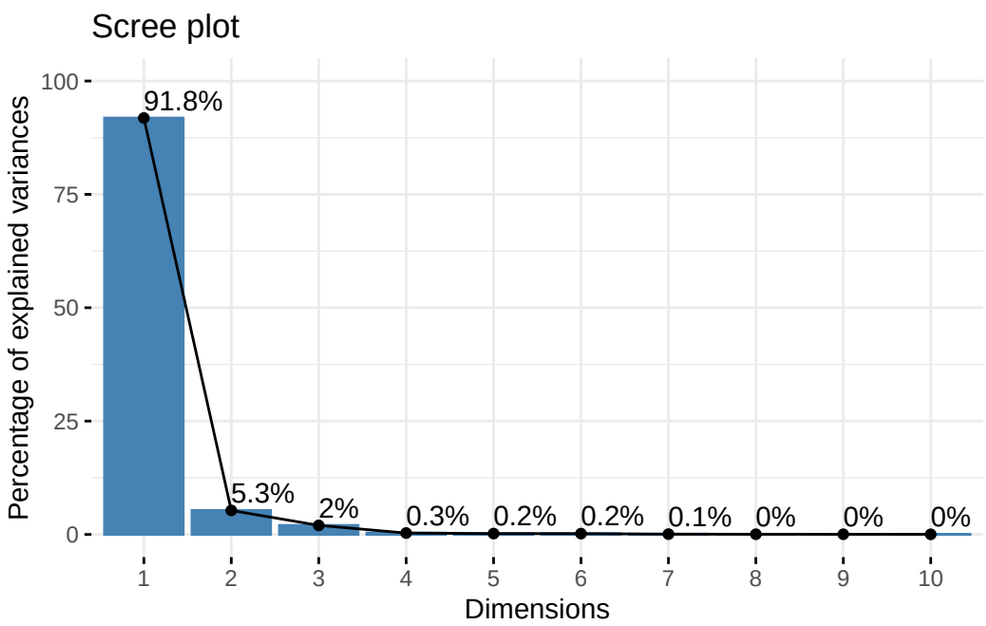


Figure 4: Scree Plot for GDP

```
fviz_ellipses(gdp.pca, habillage=gdp$continent,repel = TRUE)
```

```
fviz_pca_biplot(gdp.pca, label = "var", habillage=gdp$continent,
                addEllipses=TRUE, ellipse.level=0.95,
                ggtheme = theme_minimal())
```

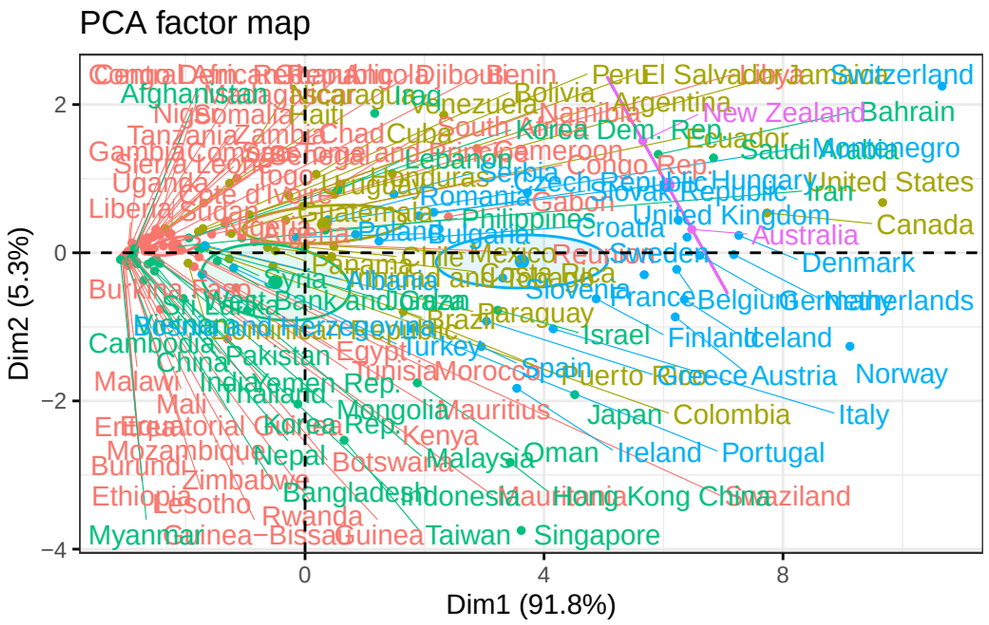


Figure 5: PCA Plot for GDP with country labels

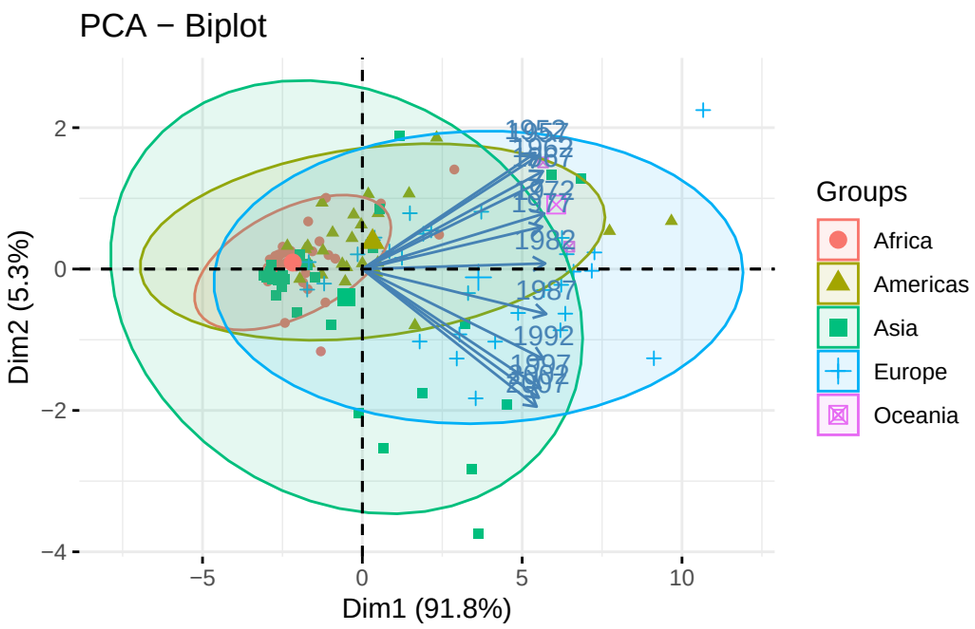


Figure 6: PCA-Biplot for GDP

Principal Component Analysis(PCA) is a statistical technique used to project the dataset's features on a vector that maximizes the variance between the points. It looks for a few linear combinations that can be used to summarise the data without losing much information. The first projection vector is known as the First Principal Component Vector. For the GDP dataset, the first principal component captures 91.8% of the variance(Figure 4). Together with the second principal vector they capture 97.1% of the variance in the dataset. This technique can reduce the dimension in terms of features in the data set. From the circle plot(Figure 6), we can infer that all the years are positively correlated to the first principal vector. The angle between the feature and principal component tells the amount of contribution to the principal component. The lesser the angle between the feature and the first component, the more the contribution to the component in terms of linear combinations. All the countries(Figure 5) that align in the direction of the first principal component are the countries with high GDP(per capita) e.g. USA, Norway, and Switzerland on the other hand countries lying opposite to the direction of the first principal vector are the countries with low GDP. These countries are developing nations from Asia and Africa e.g. Rwanda, Mozambique, Syria etc.

```
#PCA on Life Expectancy
Y<-lifeExp[,2:13]
life.pca<- prcomp(Y,scale=TRUE)
summary(life.pca)
```

```
#Visualisation of Principal Components
autoplot(life.pca, data = lifeExp, colour='continent')
```

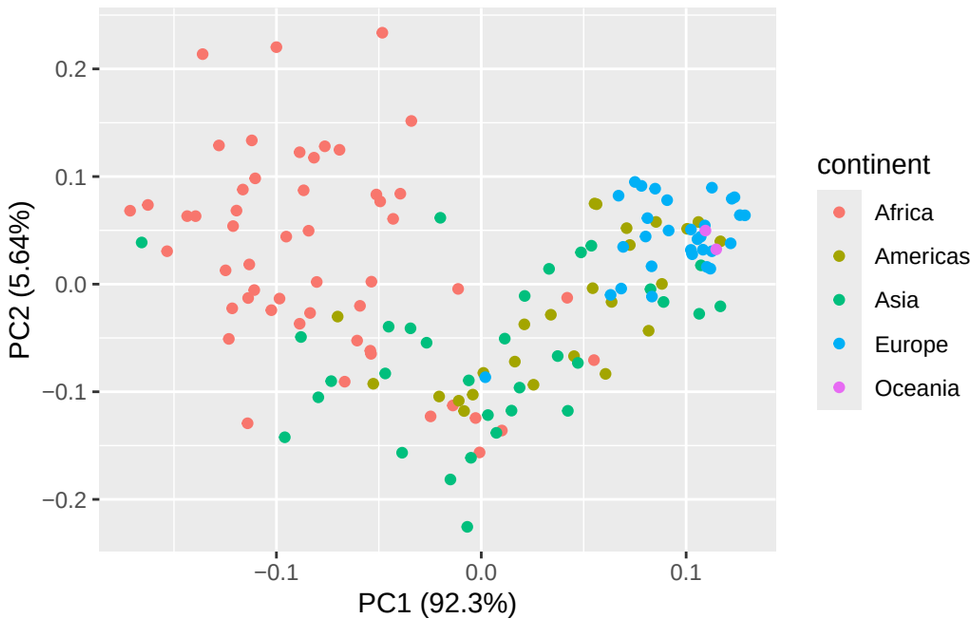


Figure 7: PCA plot for Life Expectancy

```
fviz_eig(life.pca, addlabels = TRUE, ylim = c(0, 100))
```

```
fviz_ellipses(life.pca, habillage=lifeExp$continent,repel = TRUE)
```

```
fviz_pca_biplot(life.pca, label = "var", habillage=lifeExp$continent,
                addEllipses=TRUE, ellipse.level=0.95,
                ggtheme = theme_minimal())
```

From the scree plot(Figure 8) for life expectancy, the first principal component captures 92.3% of the variance in the dataset. Together with the second component it captures 97.94% of the variance in the dataset. Countries that align in the direction of the first principal component are the countries with high life expectancy e.g. Netherlands, Denmark, and Germany(Figure 9). However, those lying in the direction opposite to the first principal vector are the countries with low life expectancy e.g. Ethiopia, Myanmar, Nigeria etc. On plotting the principal component it is very easy to segregate the countries of Africa from that of Europe and America.

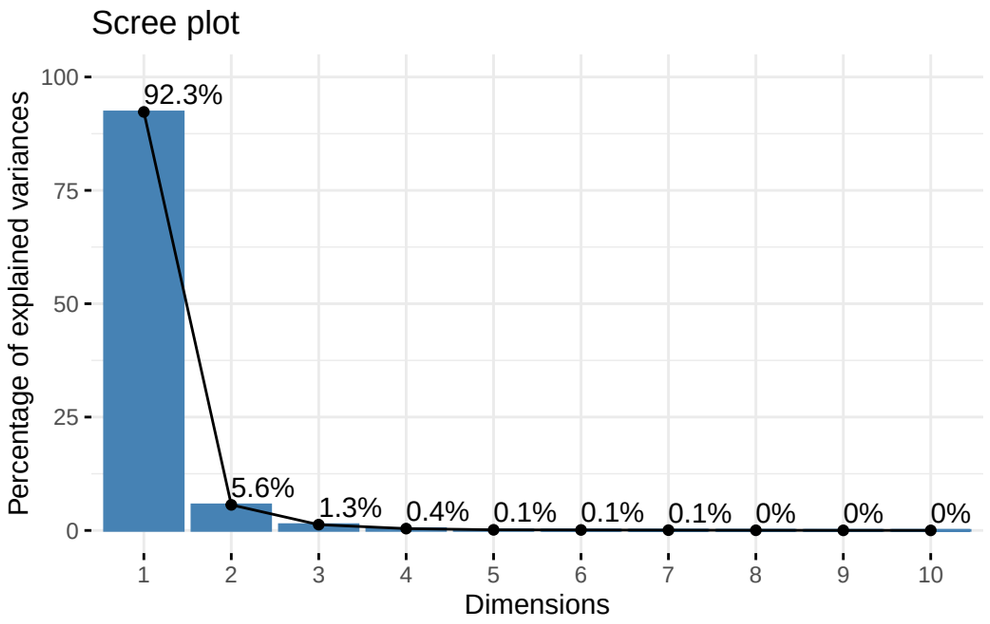


Figure 8: Screeplot for Life Expectancy

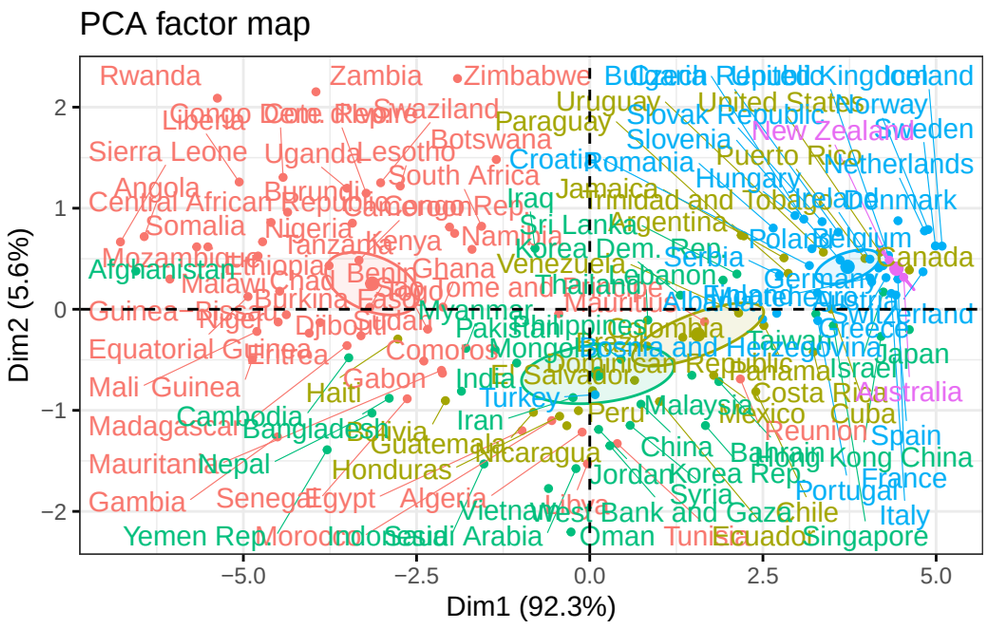


Figure 9: PCA plot for Life Expectancy with country label

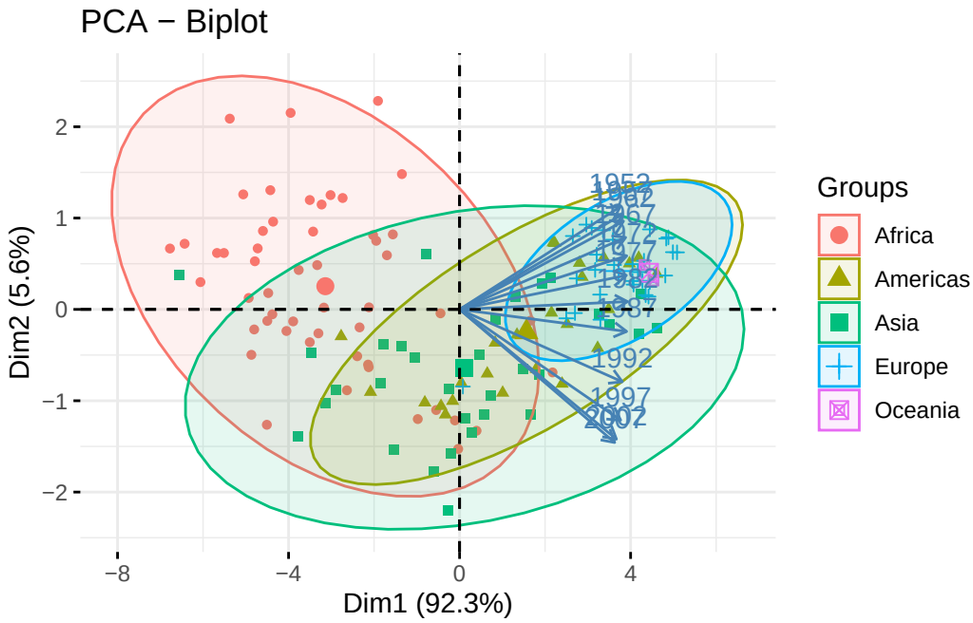


Figure 10: PCA Biplot for Life Expectancy

Since PCA is performed separately on GDP and Life expectancy data sample covariance matrix is used as all the features in a particular dataset are measured on same scale.

```
# GDP vs Life Expectancy
plot(x=gdp.pca$x[,1],y=life.pca$x[,1],col=as.factor(gdp$continent),
    pch=20, main="First Principal Component GDP Vs LifeExp",
        ylab = "Life Exp",xlab = "GDP")
legend('bottomright',inset=0.05,legend=unique(as.factor(gdp$continent)),
    pch=20,col=unique(as.factor(gdp$continent)),title="Continents",
        pt.bg = as.factor(gdp$continent))
```

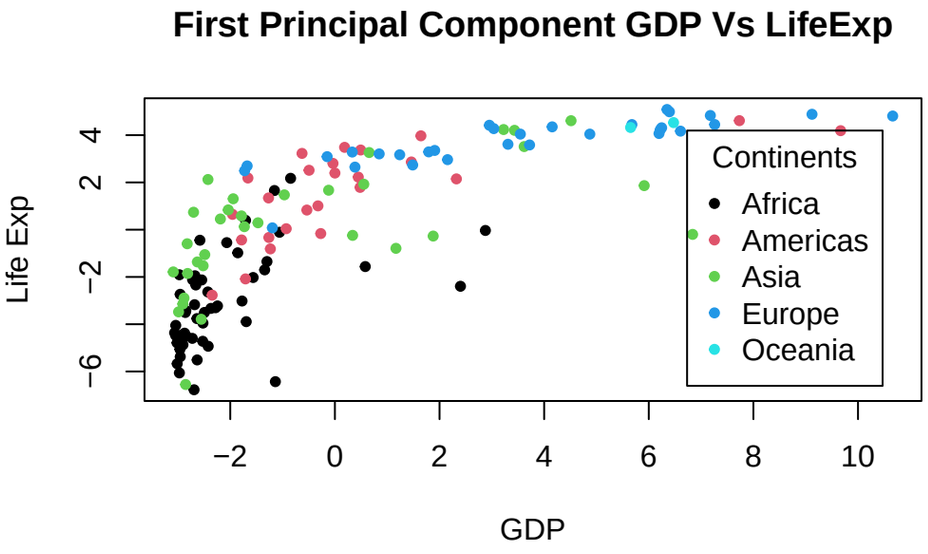


Figure 11: GDP vs. Life Expectancy for Principal Components

If we compare the above plot(Figure 11) with the one from the EDA for Life Expectancy Vs. GDP(Figure 1). One can see that both of them are almost similar. The reason for this)is that the first principal component for GDP(per capita) and life expectancy seeks the standard linear combination of the variables that have maximal variance. In this case, it makes sense as it separates out the countries, thereby easing consideration of differences between them.

Canonical Correlation Analysis

```
prem.cca <- cc(log(X),Y)
plt.cc(prem.cca, ind.names = rownames(gdp), type='b',var.label=FALSE)
```

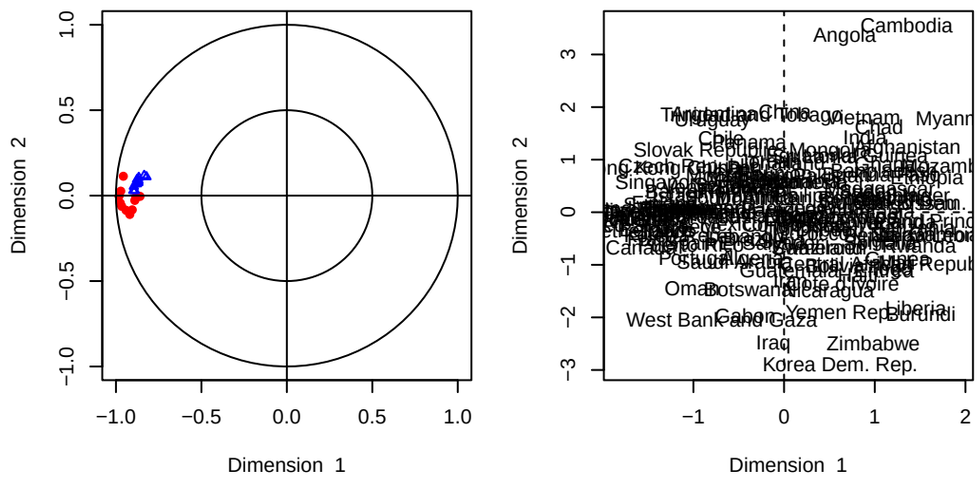


Figure 12: CCA Plots

```
plot(x=prem.cca$scores$xscores[,1],y=prem.cca$scores$yscores[,1],
    pch=20,col=as.factor(gdp$continent),main="CC_1 Score GDP VS Life Exp",
        xlab="GDP_CC_1",ylab="LifeExp_CC_1")
legend('bottomright',inset=0.0005,y.intersp=0.5,
    legend=unique(as.factor(gdp$continent)),pch=20,
        col=unique(as.factor(gdp$continent)),title="Continents",
            pt.bg = as.factor(gdp$continent))
```

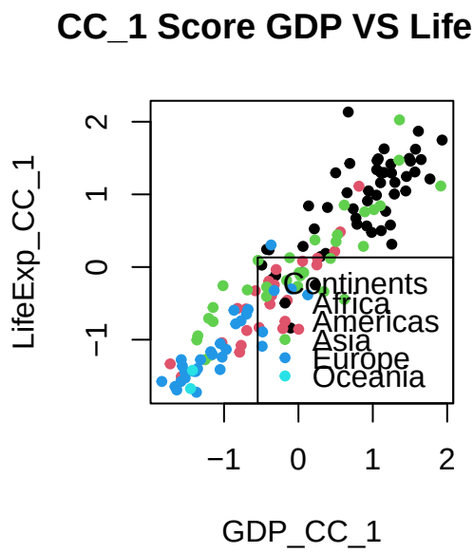


Figure 13: GDP vs. Life Expectancy for CCA 1

```
par(mfrow=c(1,2))
hist(UN$gdpPercap_2007,main="Without Log")
hist(log(UN$gdpPercap_2007),main="With Log transformation")
```

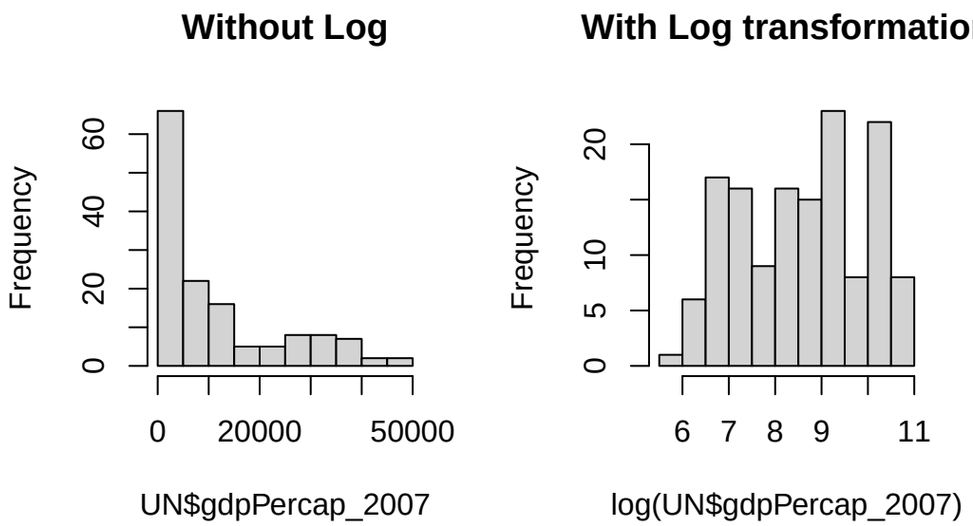


Figure 14: Log Transformation

Canonical Correlation Analysis(CCA) relies on partitioning the variables into two sets and the prime objective of the analysis is to find linear combinations to have the largest possible correlations between two sets. Unlike PCA which considers interrelationships within the sets of variables, CCA focuses on the relationship between two groups of variables. From the above plots(Figure 12), it can be inferred that the angle between Life Expectancy(Blue) and GDP per capita(Red) is very small which means there is a strong correlation between the two variables especially for the year 2007 the angle is almost zero. The distance of dots from the center of the circle tells the strength of the variable or contribution to CCA1 and CCA2. Both GDP (per capita) and life expectancy contribute more to CCA 1 compared to CCA2. From (Figure 13) it can be inferred that there is a strong correlation between Life Expectancy and GDP.

From the second plot(Figure 12), it can be said that countries with high GDP and High Life expectancy are clustered in the directions of the CCA1 e.g. USA, Singapore, Norway, etc. However, countries like Angola, Mozambique, and Afghanistan which are low on both GDP per capita and Life expectancy lie opposite to the vectors these can be categorized as least developed countries. Countries lying in the direction perpendicular to the vector are middle-income countries with mid-range GDPs with average life expectancies like India, China, and Saudi Arabia.

Finally, from the histogram(Figure 14)one can tell that GDP data is right skewed. CCA assumes normality for optimal performance, so taking the logarithm can compress the larger values and stretch the smaller ones, leading to a more normal distribution. Using logarithms improves the suitability of the data for CCA by aligning the feature with CCA1 and CCA2.

Multidimensional Scaling

```
# Multidimensional Scaling
UN.transformed <- cbind(log(UN[,3:14]), UN[,15:26], log(UN[,27:38]))
un.mds <- cmdscale(dist(UN.transformed,method= "euclidean"),k=3)
coords <- data.frame(x=un.mds[,1], y=un.mds[,2])
ggplot(coords, aes(x=x, y=y, col= UN$continent)) +  geom_point() +
    labs(x= 'Dimension 1',y='Dimension 2',color='Continent')+
    geom_text_repel(aes(label = UN$country), size = 1.5)
```

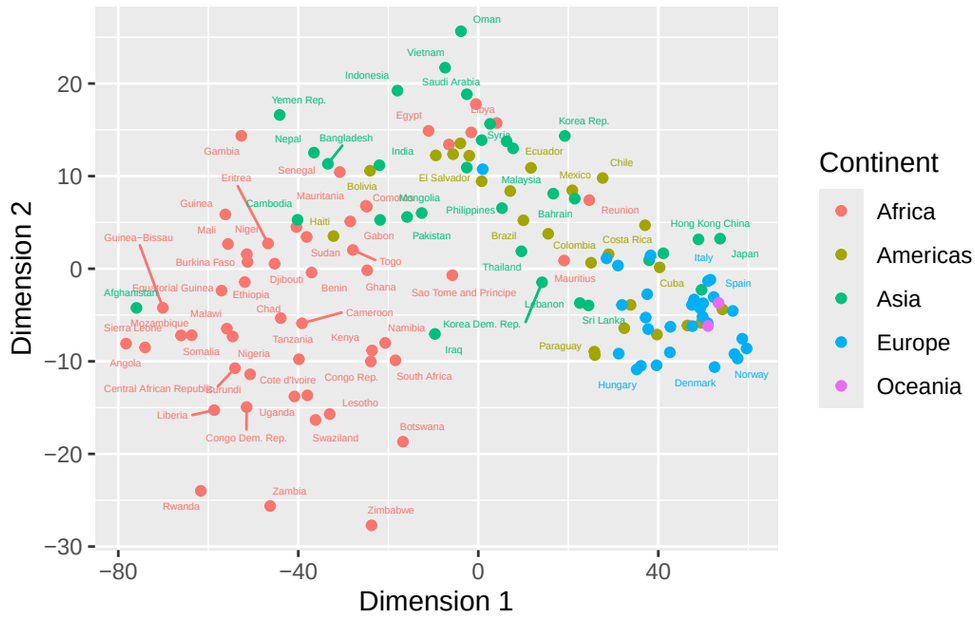


Figure 15: 2-dimensional representation of the data

Multidimensional Scaling (MDS) is a multivariate statistical technique used for visualizing the relationships between objects (data points) based on their dissimilarities (distances or similarities). The combined data has 36 features, 12 each from GDP, Life Expectancy, and Population making it a high-dimensional space. MDS creates a lower-dimensional representation for easier visualization and understanding of the relationships between the objects. Here, we are using Euclidean distance to generate a pairwise distance between countries based on 36 features.

From the plot(Figure 15), it can interpreted that countries of Europe that have high GDP, life expectancy, and low populations are clustered together (as they are very similar or have low Euclidean distance between them). However, African countries which have low GDP, life expectancy, and high populations are completely separate from European countries(as they are very dissimilar or have high Euclidean distance relative to European countries). The plot can also be used to interpret how far a country is from most developed nations in the world.

Linear discriminant analysis

```
UN_lda <- UN[,c(1,3:38)]
UN_lda$continent <- as.factor(UN_lda$continent)

set.seed(787)
sample <- sample.split(UN_lda$continent, SplitRatio = 0.7)
train  <- subset(UN_lda, sample == TRUE)
test   <- subset(UN_lda, sample == FALSE)
un.lda<-lda(continent~., train)
un.pred <- predict(un.lda, test)
print(paste("The predictive accuracy is ",
    sum(un.pred$class== test$continent)/dim(test)[1]*100, "%"))
```

```
[1] "The predictive accuracy is  79.069 %"
```

```
table(un.pred$class, test$continent)
```

```
          Africa Americas Asia Europe Oceania
  Africa      16        0    4      0       0
  Americas     0        6    2      0       1
  Asia         0        0    3      0       0
  Europe       0        1    1      9       0
  Oceania      0        0    0      0       0
```

Linear Discriminant Analysis(LDA) aims to identify a linear transformation that maximizes the separation between different classes while minimizing the variation within each class. From the plot below(Figure 16) linear transformation has segregated data into different classes with 79.069% accuracy.

```
un.lda1<-lda(continent~., UN_lda)
un.pred1 <- predict(un.lda1, UN_lda)
gg_ordiplot(un.lda1, UN_lda$continent,ellipse = TRUE,label = TRUE,
    hull = FALSE,spiders = FALSE,kind = c("ehull"),pt.size = 2)
```

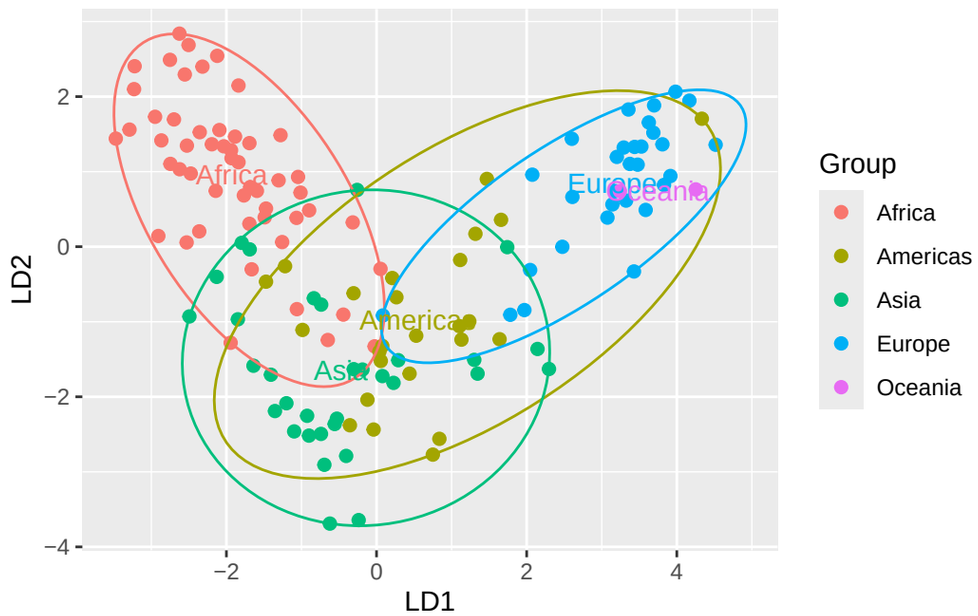


Figure 16: Plot for LDA based on continents

Clustering

```
UN.scaled <- UN[,1:26]
UN.scaled[,3:26] <- scale(UN[,3:26])
h_clust<-hclust(dist(UN.scaled[,3:26],method="euclidean"),method="complete")
p<-plot(h_clust, labels=UN.scaled$country,cex=0.5)
cut<- cutree(h_clust, k=6)
table(cut, UN.scaled$continent)
```

cut	Africa	Americas	Asia	Europe	Oceania
1	7	13	16	3	0
2	42	2	7	0	0
3	3	1	3	0	0
4	0	7	4	16	0
5	0	2	1	11	2
6	0	0	1	0	0

```
rect.hclust(h_clust, k = 6, border = "red")
```

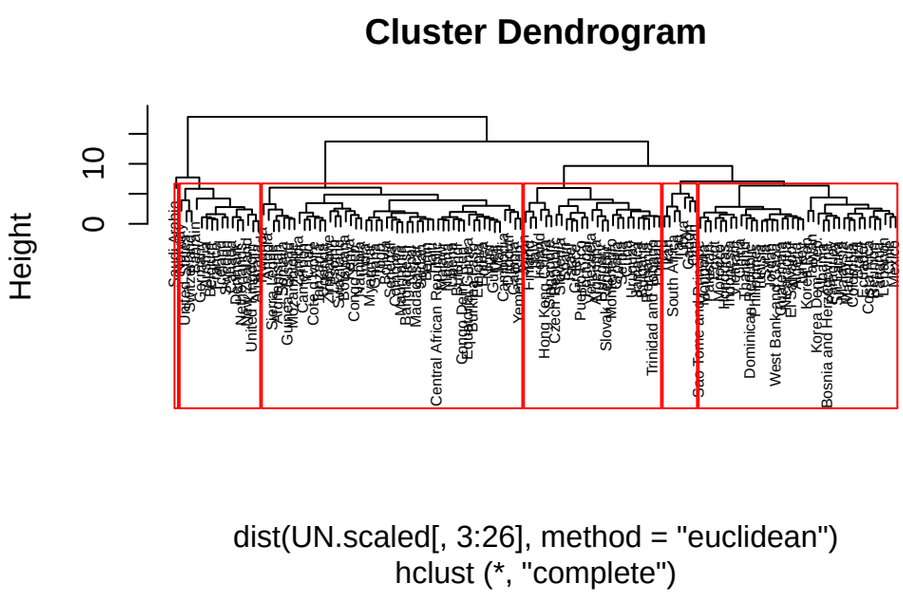


Figure 17: Dendrogram for hierarchical clustering

```
clusters <- as.data.frame(cut)
UN_hclust <- cbind(UN,clusters)
ggplot(UN_hclust, aes(x=gdpPercap_2007,y=lifeExp_2007,col=as.factor(cut)))+
  geom_point()+ geom_text(label=UN_hclust$country)+
  labs(x= 'GDP for the year 2007',y='Life Expectancy for year 2007',
    color='Clusters') +ggtitle("Clustering by Hierarchial Clustering")
```

Clustering is a statistical technique used to group data points into clusters based on their similarities. It's an unsupervised learning method, meaning it doesn't involve predefined categories or labels for the data.

In this clustering problem, different methods like hierarchical clustering and K-means are used to cluster countries based on continent. Also, for hierarchical clustering different methods like "single", and "complete" are discussed. Since GDP and Life expectancy are indicators for growth using complete instead of single as a criteria for clustering provides a better picture as it considers the maximum distance from the cluster. It helps group countries into highly developed, middle-developed, and least developed countries. Six clusters are found adequate to segregate developed countries like the USA, UK, Germany, France(Cluster 1 and 2) from least developed countries of

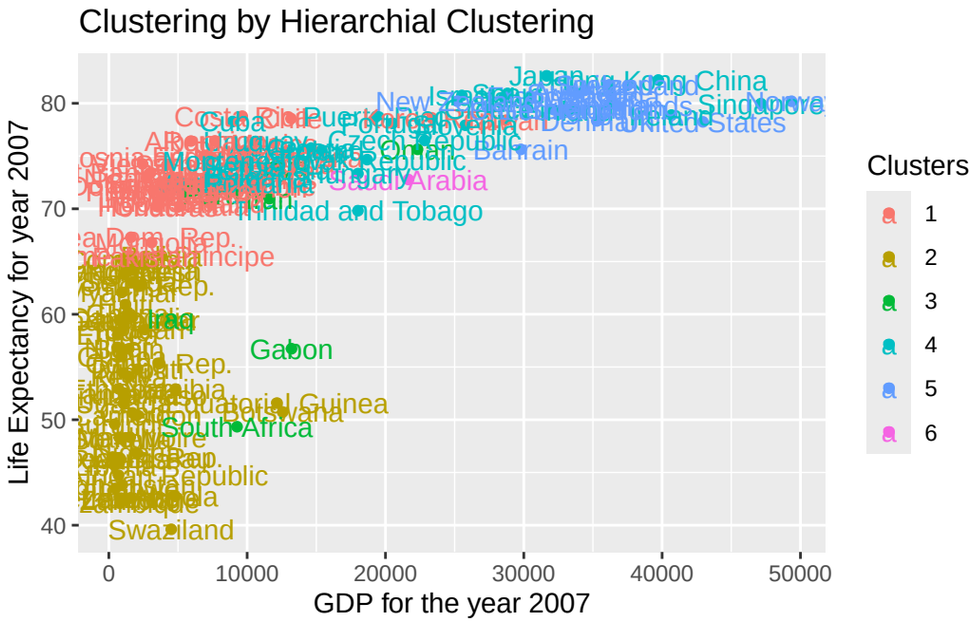


Figure 18: GDP vs. Life Expectancy clustered by hierarchical clustering

Asia and Africa(Cluster 3 and 5), Cluster 4 and 6 groups countries which countries have mid-level of development(Figure 17).

```
# K-means for clustering
un.k <- kmeans(UN.scaled[,3:26], centers = 5, nstart=25)
table(un.k$cluster, UN.scaled$continent)
```

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

```
fviz_cluster(un.k,UN.scaled[,3:26], ellipse.type = "norm")
```

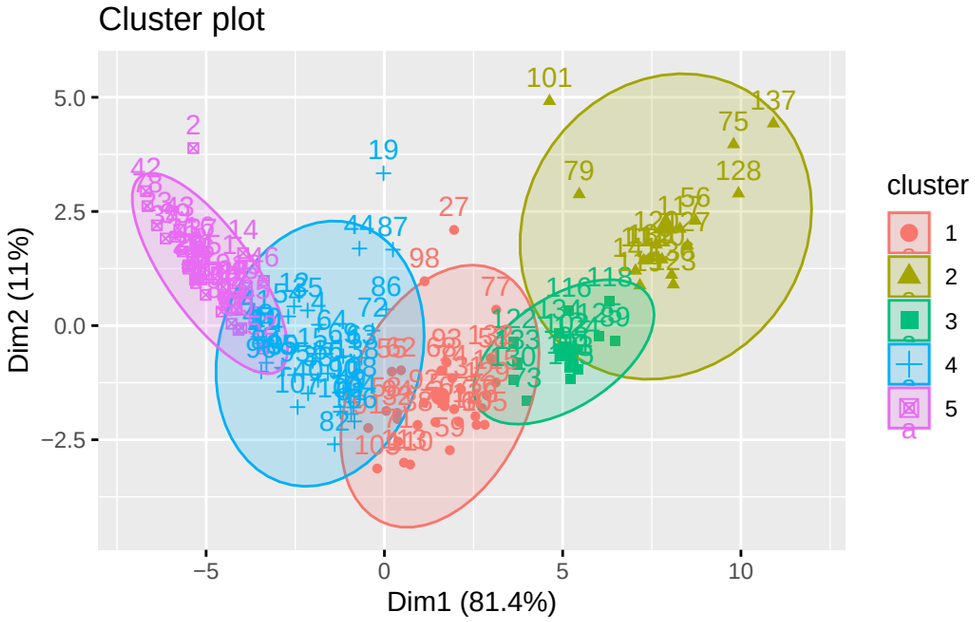


Figure 19: Cluster Plot using K-means

```
fviz_nbclust(UN.scaled[,3:26], kmeans, method = "wss")
```

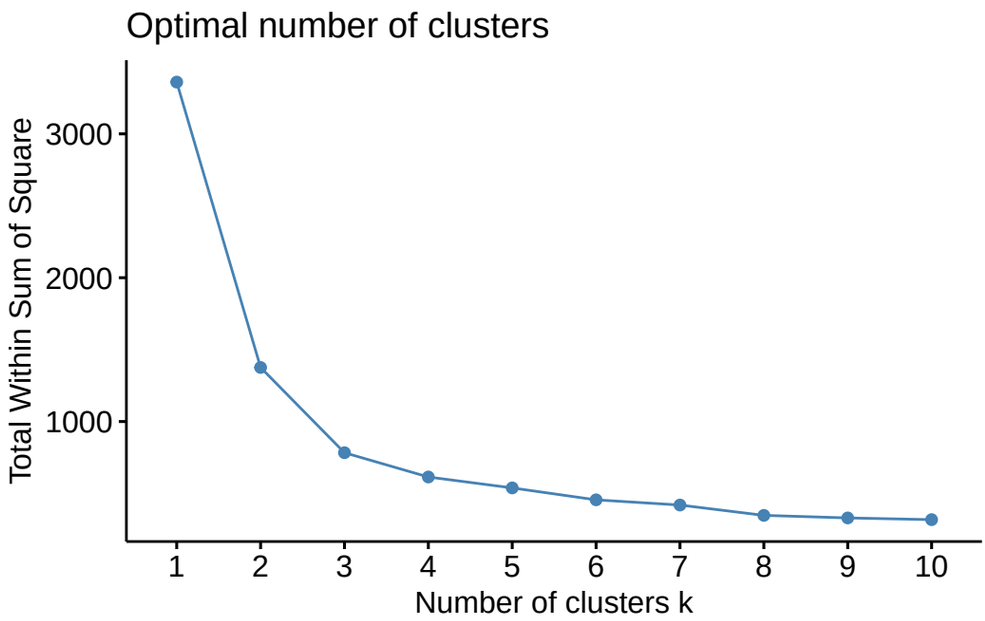


Figure 20: Optimum number of cluster

```
clusters <- as.data.frame(un.k$cluster)
UN_kmeans <- cbind(UN,clusters)
ggplot(UN_kmeans, aes(x=gdpPercap_2007,y=lifeExp_2007,
    col=as.factor(un.k$cluster)))+
geom_point()+ geom_text(label=UN_kmeans$country)+
labs(x= 'GDP for the year 2007',y='Life Expectancy for year 2007',
    color='Clusters')+ggtitle("Clustering by K-means")
```

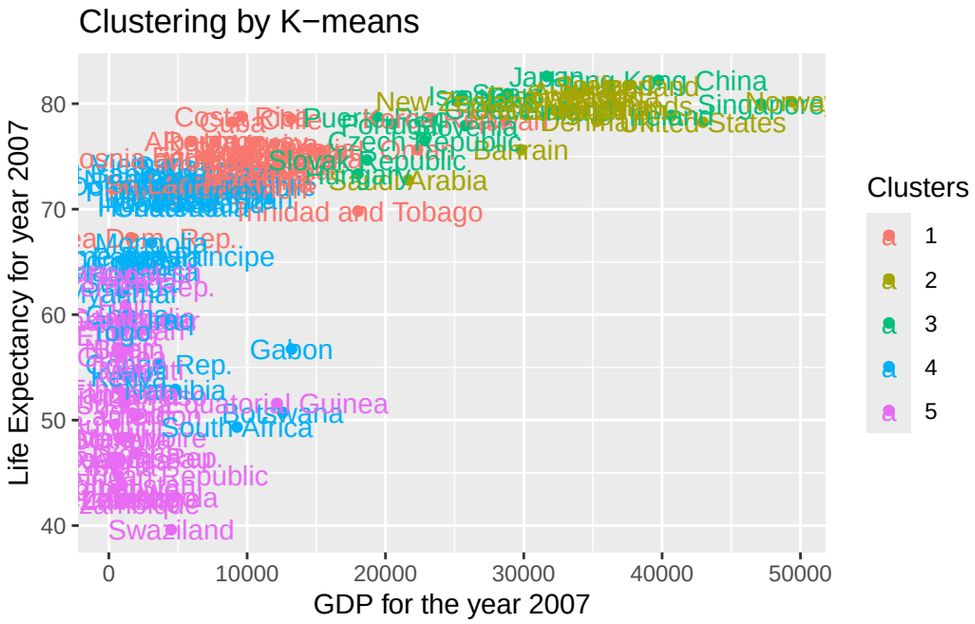


Figure 21: GDP vs. Life Expectancy clustered by K-means clustering

Next K-means algorithm is used for clustering. In order to compare hierarchical clustering with K-means the number of clusters is kept same i.e. six for K-means and Euclidean distance is used to create a distance matrix(Figure 19). Also from the Elbow plot(Figure 20) one can tell that there are six clusters are optimum to reduce the within-cluster sum of squared error. To compare the clustering methods a plot is created for GDP Vs Life Expectancy and grouping is done based on the cluster generated by both methods. The k-mean algorithm rely on reducing the sum of of

squared error within the cluster(WCSS) to do so it chooses points that are close to the centroid. By minimizing WCSS, K-means tries to create clusters where the data points within a cluster are close to each other (small distances) and far from points in other clusters (large distances).

Both methods(Figure 18 and 21) have created almost similar clusters. All the least developing countries of Africa and Asia and developed nations of Europe and America are grouped in the same clusters in both methods. This can be attributed to the fact that both these cluster represents the extreme on-distance matrix. However, the result are not exactly the same for middle income countries.

Linear Regression

```
# Data Preparation for Linear Regression
UN_reg <-cbind(gdp[,2:13], lifeExp[,13])
colnames(UN_reg)[13]<- "Life.Exp"
UN_reg_log <- cbind(log(gdp[,2:13]), lifeExp[,13])
colnames(UN_reg_log)[13]<- "Life.Exp"
independent<- as.matrix(log(UN_reg[,1:12]))
dependent<- as.matrix(UN_reg[13])
```

```
#Fitting a linear model(Ordinary Least Square Method)
model1 <- lm(Life.Exp~.,data = UN_reg)
summary(model1)
```

```
Ordinary Least Square Regression without Log Transformation
Call:
lm(formula = Life.Exp ~ ., data = UN_reg)

Residuals:
    Min      1Q  Median      3Q     Max
-22.054  -5.872   1.131   6.745  14.941

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 58.5547252  1.1248985  52.053   <2e-16 ***
`1952`      -0.0014463  0.0024503  -0.590   0.5561
`1957`       0.0045240  0.0032510   1.392   0.1665
`1962`      -0.0038932  0.0026852  -1.450   0.1495
`1967`      -0.0001022  0.0016317  -0.063   0.9502
`1972`       0.0013214  0.0015864   0.833   0.4064
`1977`      -0.0010691  0.0010150  -1.053   0.2942
`1982`       0.0011905  0.0012654   0.941   0.3486
`1987`       0.0003911  0.0008256   0.474   0.6365
`1992`      -0.0017830  0.0012528  -1.423   0.1571
`1997`       0.0022685  0.0013548   1.674   0.0965 .
`2002`      -0.0014386  0.0010788  -1.334   0.1847
`2007`       0.0007789  0.0006784   1.148   0.2530
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.877 on 128 degrees of freedom
Multiple R-squared:  0.5066,    Adjusted R-squared:  0.4603
F-statistic: 10.95 on 12 and 128 DF,  p-value: 9.424e-15
```

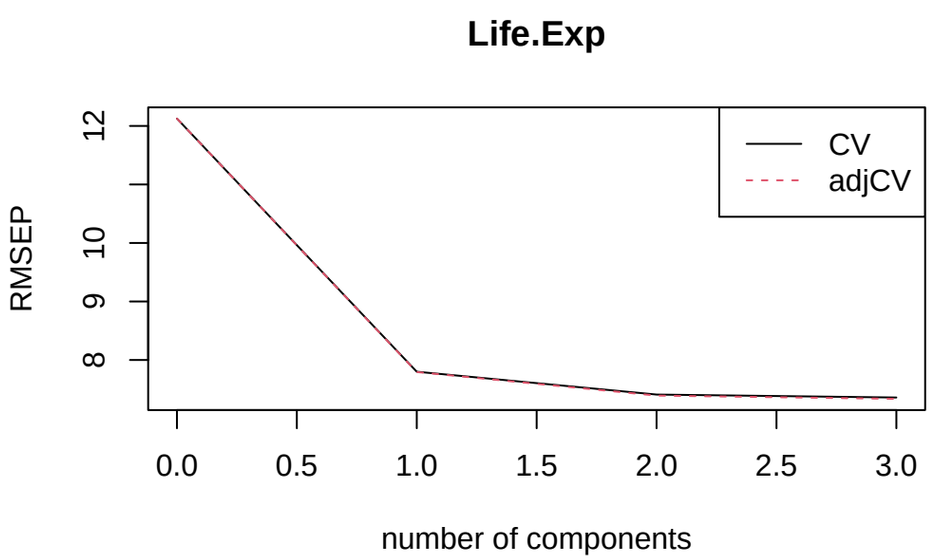
```
#Fitting a linear model with log transformation (Ordinary Least Square Method)
model1_log <- lm(Life.Exp~.,data = UN_reg_log)
summary(model1_log)
```

```
Ordinary Least Square Regression with Log Transformation
Call:
lm(formula = Life.Exp ~ ., data = UN_reg_log)

Residuals:
     Min       1Q   Median       3Q      Max
-23.1669  -2.4054   0.5789   3.8217  13.2822

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)    4.393      4.679   0.939  0.34966
`1952`        -5.405      6.416  -0.842  0.40117
`1957`        13.302      9.535   1.395  0.16540
`1962`        -5.630      9.928  -0.567  0.57167
`1967`         1.655      7.080   0.234  0.81553
`1972`        -4.552      6.441  -0.707  0.48106
`1977`        -2.883      6.033  -0.478  0.63361
`1982`        -3.357      7.647  -0.439  0.66141
`1987`         9.838      6.427   1.531  0.12830
`1992`        -8.542      5.757  -1.484  0.14034
`1997`        19.130      6.801   2.813  0.00569 **
`2002`       -13.121      7.945  -1.652  0.10108
`2007`         7.003      4.721   1.483  0.14043
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.949 on 128 degrees of freedom
Multiple R-squared:  0.6976,    Adjusted R-squared:  0.6692
F-statistic:  24.6 on 12 and 128 DF,  p-value: < 2.2e-16
```

Linear regression is a fundamental statistical method used for predicting a continuous outcome variable based on the value of one or more predictor variables. For the base model, ordinary Least Square is used. The first modeling is done on original features and the second model performs regression with the log of GDP. Taking a log of GDP parameters has increased the R-squared from 0.5066 to 0.6976 which means that the new model can better explain the variability in dependent features (Life Expectancy) based on independent features (GDP per capita). Simultaneously, the residual error has decreased from 8.87 to 6.95 which means the new model is better. However, the normality plot for both models it violates the normality assumption of residuals leading to poor fit. Taking a log of the GDP features reduces skewness, leading to a better fit. Since taking logs on GDP is performing better it's better to perform future regression modeling with log of GDP.

```
# Fitting PCR Regression
model2 <- pcr(Life.Exp~.,data = UN_reg_log,,ncomp=3, validation =
"CV",scale = TRUE)
summary(model2)
plot(RMSEP(model2), legendpos = "topright")
rmse <- sqrt(mean((model2$residuals)^2))
print(paste("The mean Square Error for PCR is",rmse))
```




```
[1] "The mean Square Error for PCR is 7.24643988862126"
```

From Principal Component Analysis(PCA), it can be recalled that the first three components(Figure 4) can explain 99% of the variability in the dataset. Regressing on these components leads to the simplification of the regression model. However, the Root Mean Square Error(RMSE) has increased as taking only the first three component has led to a loss of information.

```
#Fitting a Lasso Regression
set.seed(164)
model3 <- glmnet(independent, dependent, alpha=1)
plot(model3, xvar='lambda')
lambdas <- 10^seq(3,-2,by=-0.1)
cv_fit <- cv.glmnet(independent, dependent, alpha = 1, lambda = lambdas)
print(cv_fit)
plot(cv_fit, main="Lasso Regression")
coef(glmnet(independent, dependent, alpha=1, lambda=cv_fit$lambda.1se))
```

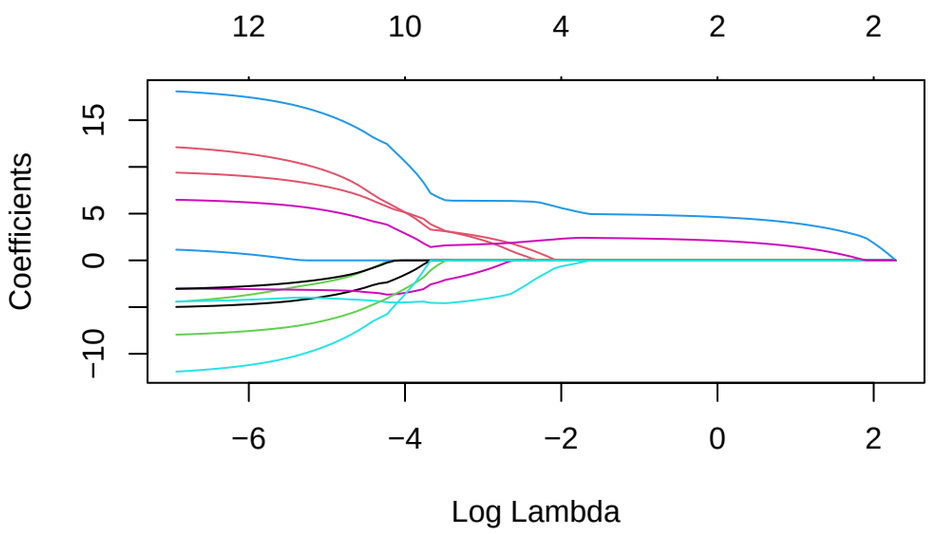


Figure 22: Coefficients Vs Log Lambda

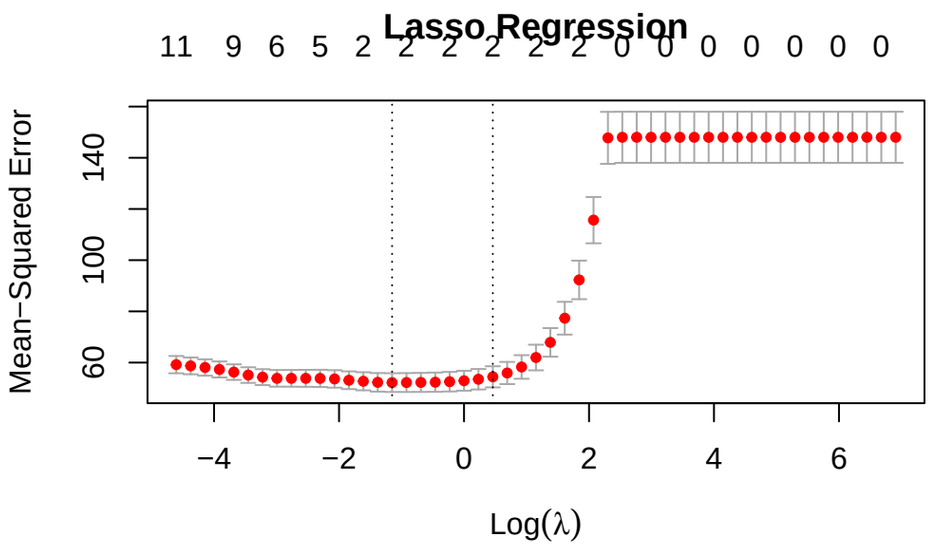


Figure 23: MSE vs. Log(λ)

```
Coefficients for Lasso Regression
                    s0
(Intercept) 13.836162
1952         .
1957         .
1962         .
1967         .
1972         .
1977         .
1982         .
1987         .
1992         .
1997         4.410438
2002         .
2007         1.881697
```

As seen above the shrinkage method has reduced the Mean Squared Errors significantly from 6.949 for OLS to 3.668. If the correlation matrix for the given dataset is analyzed it can be deduced that a lot of covariates are highly correlated which leads to poor determination of coefficients by OLS. From the summary of the model1 (OLS), it can be seen that some features have very high positive coefficients and some negative which lead to the cancellation of terms and high variance. From the plots(Figure 22), it can be visualized that coefficients are shrunk depending on their singular values. For Lasso, only two features survived 1997 and 2007 other features have small singular values. Similarly, for Ridge Regression, only the last four features have high coefficients others are shrunken to near-zero values(Figure 24). Generally speaking, shrinkage is more in the least important direction. The value of Lambda is calculated through cross-validation which is 0.3162 and 1 for lasso (Figure 23) and Ridge Regression(Figure 25) respectively.

```r
#Fitting a Ridge Regression
set.seed(164)
model4 <- glmnet(independent, dependent, alpha=0)
plot(model4, xvar='lambda')
lambdas <- 10^seq(3,-2,by=-0.1)
cv_fit1 <- cv.glmnet(independent, dependent, alpha = 0, lambda = lambdas)
print(cv_fit1)
plot(cv_fit1, main= "Ridge Regression")
```

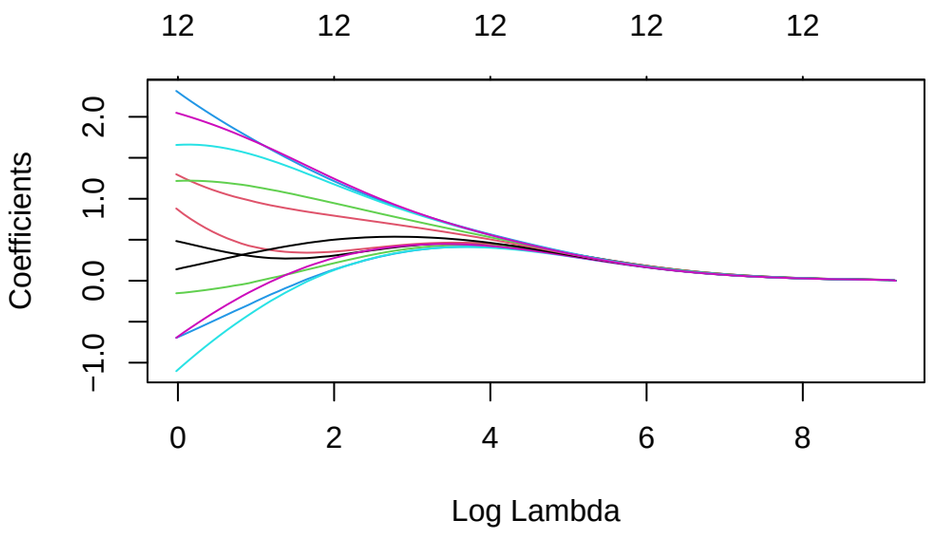


Figure 24: Coefficients vs. Log(λ)

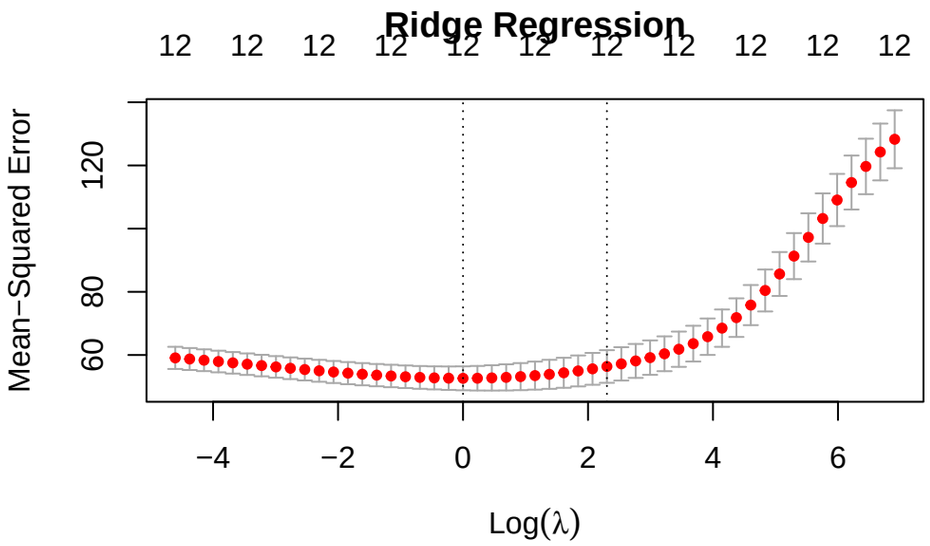


Figure 25: MSE vs. $\log(\lambda)$

```
coef(glmnet(independent, dependent, alpha=0, lambda=cv_fit$lambda.1se))
```

```
Coefficients for Ridge Regression
                     s0
(Intercept)  5.07715650
1952         0.34171011
1957         0.60050190
1962        -0.06021555
1967        -0.44865852
1972        -0.70020120
1977        -0.39444742
1982         0.21601652
1987         1.07867786
1992         1.19198042
1997         2.00114899
2002         1.64233827
2007         1.90742862
```