Multivariate Analysis of Population and GDP growth

Lokeshwaran Arunachalam (University of Nottingham)

Data Loading and Manipulation

```
# Load the data
gap = read.csv("gap.csv")
```

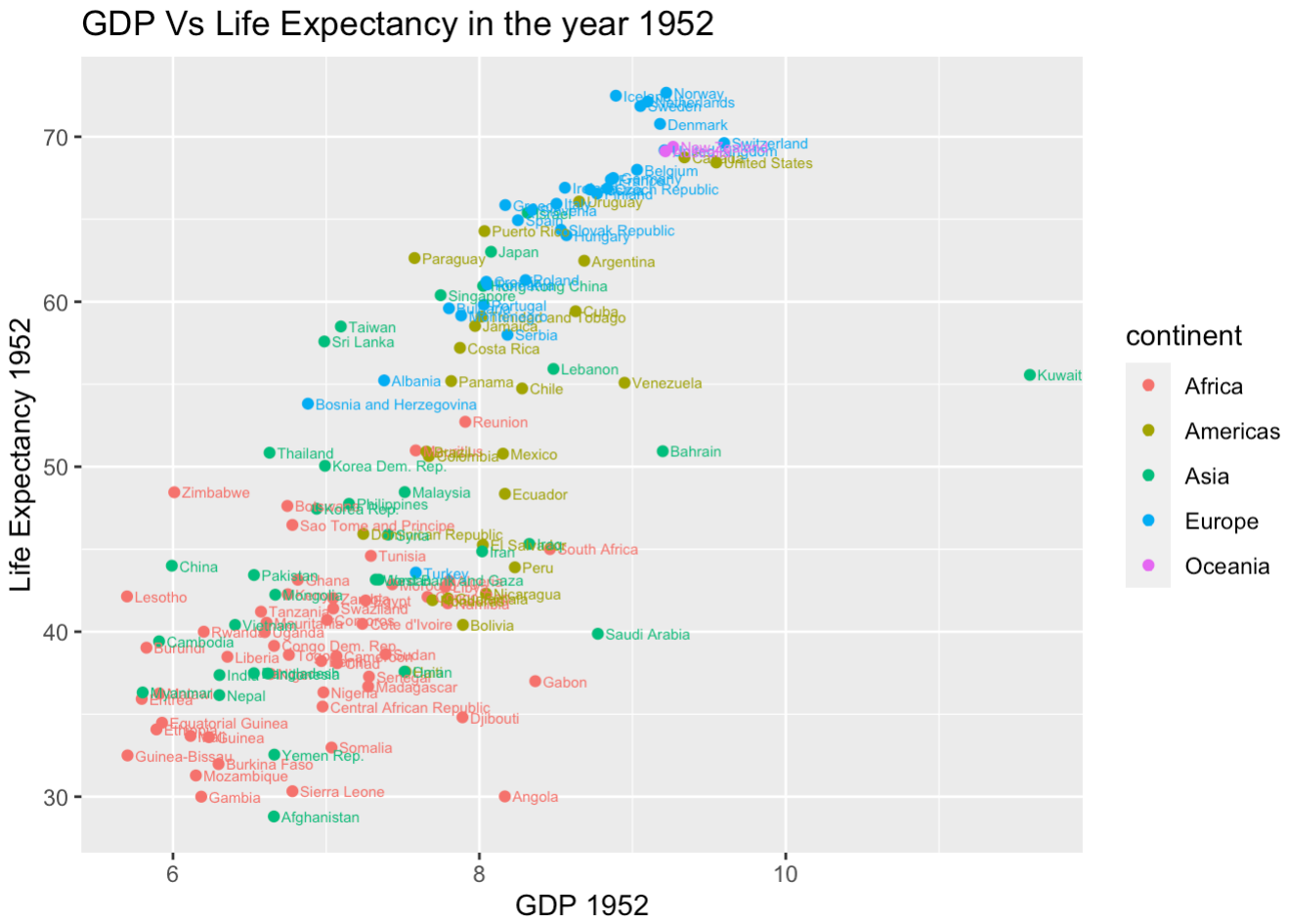
```
# apply log to GDP values
log_gap = data.frame(gap)
log_gap[, 3:14] = log(gap[, 3:14])
```

```
#GDP values
gdp = gap[, 1:14]
years = seq(1952, 2007, 5)
heading = c("continent", "country")
col_names_gdp = c(heading, years)
colnames(gdp) = col_names_gdp
```

```
#Life Expectancy values
life_exp_year = gap[, 15:26]
attr_row = gap[, 1:2]
life_exp = cbind(attr_row, life_exp_year)
colnames(life_exp) = col_names_gdp
```

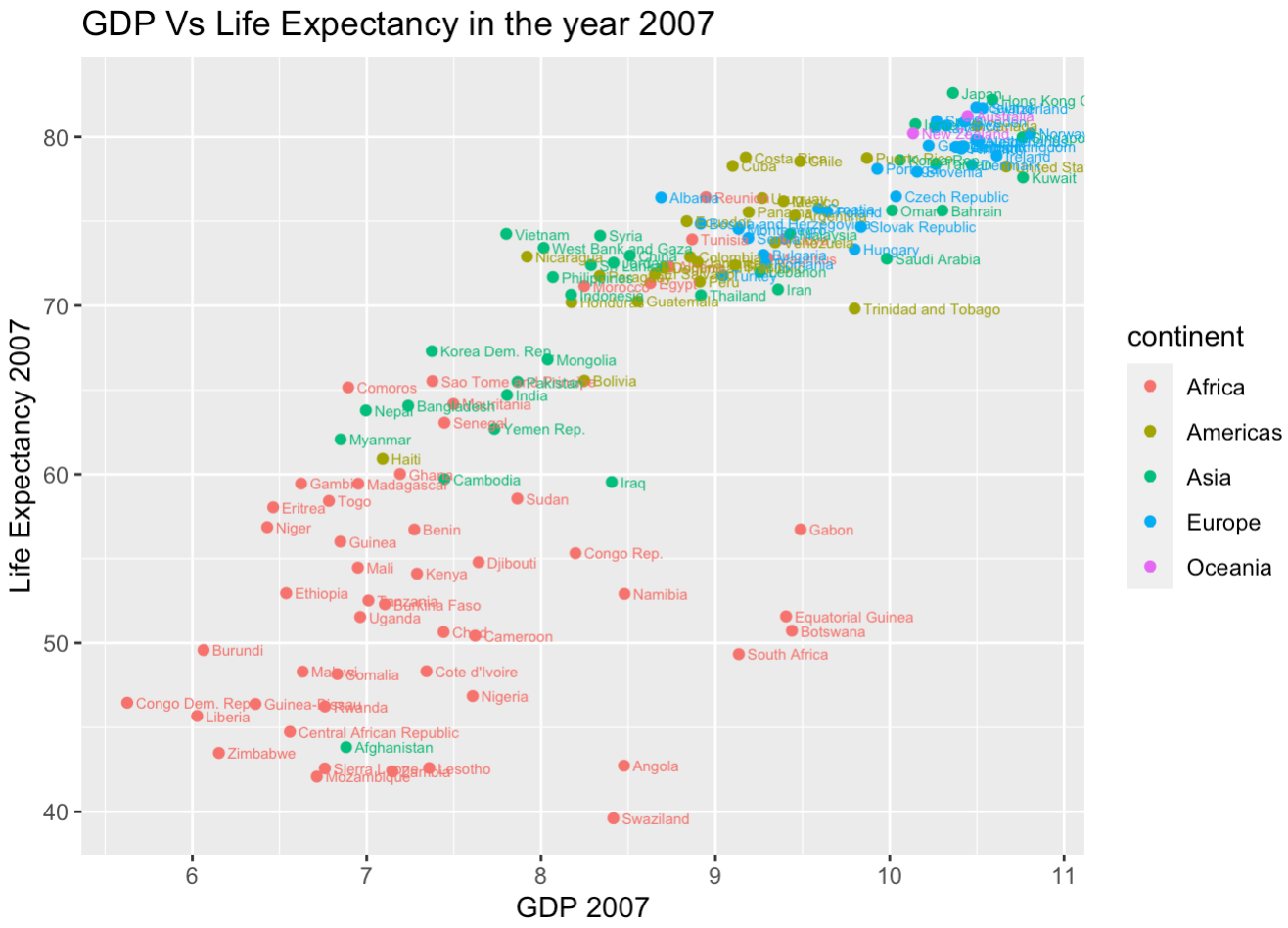
Exploratory Data Analysis

```
library(ggplot2)
ggplot(log_gap, aes(x = gdpPercap_1952, y = lifeExp_1952, colour = continent)) + geom
_point() + geom_text(aes(x = gdpPercap_1952+0.05, label = country), size = 2, hjust =
0) + labs(x = "GDP 1952", y = "Life Expectancy 1952") + ggtitle("GDP Vs Life Expectan
cy in the year 1952")
```



In the year 1952 the European countries have Higher GDP and Life expectancy compared to other continents. Whereas, the African and Asian countries has lowest GDP and life expectancy. The Kuwait is one of the Asian country which has the highest GDP in the year 1952. The American countries GDP and life span is little bit lower than the European countries.

```
ggplot(log_gap, aes(x = gdpPercap_2007, y = lifeExp_2007, colour = continent)) + geom
_point() + geom_text(aes(x = gdpPercap_2007+0.05, label = country), size = 2, hjust =
0) + labs(x = "GDP 2007", y = "Life Expectancy 2007") + ggtitle("GDP Vs Life Expectan
cy in the year 2007")
```



In the year 2007, the European, Oceania and some of the Asian countries achieved average life expectancy nearly equal to 80. The growth rate of Asian countries is staggering in both GDP and lifespan when compared to the year 1952. The Japan has the largest life expectancy when compared to other countries. Afghanistan is the only Asian country whose life expectancy is below 45 years of age.

Principal Component Analysis

Principal Component Analysis for GDP

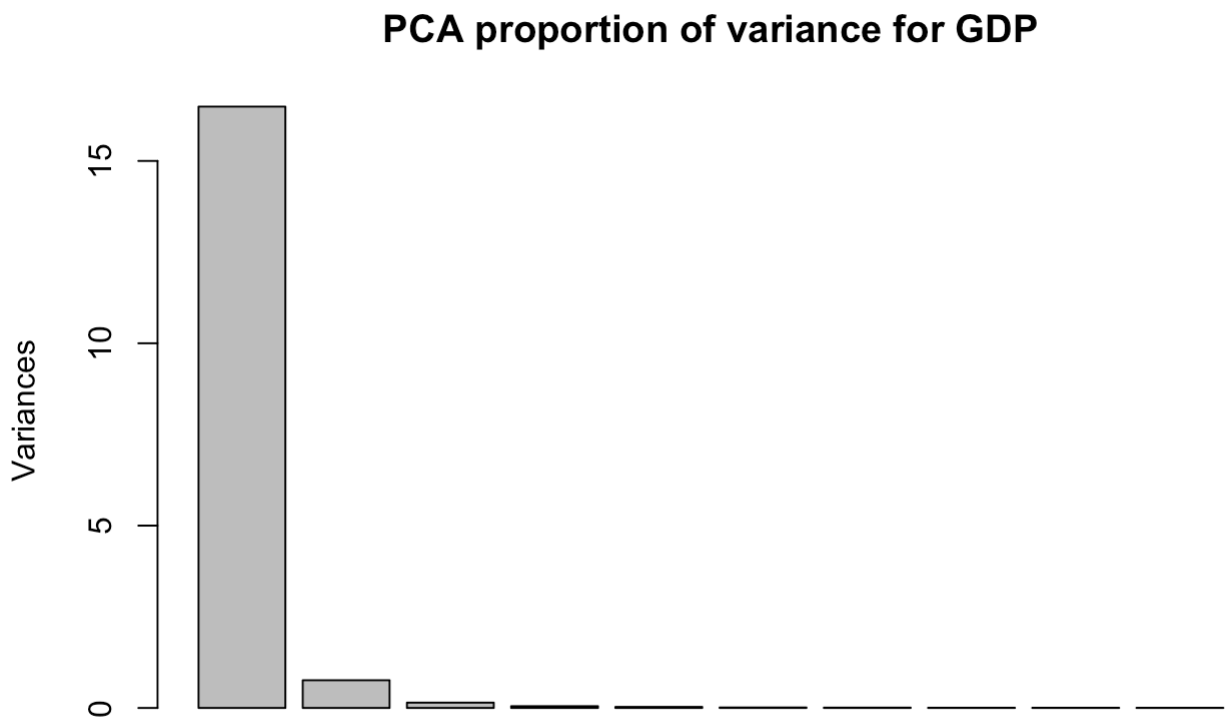
```
gdp_X = as.matrix(log(gdp[, 3:14]))
#PCA using covariance matrix
gdp_pca_cov = prcomp(gdp_X, scale = FALSE)
summary(gdp_pca_cov)
```

```
## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5     PC6     PC7
## Standard deviation     4.0608 0.8718 0.38139 0.2215 0.16930 0.11366 0.09388
## Proportion of Variance 0.9416 0.0434 0.00831 0.0028 0.00164 0.00074 0.00050
## Cumulative Proportion  0.9416 0.9850 0.99328 0.9961 0.99771 0.99845 0.99895
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.07325 0.06644 0.05813 0.05649 0.04419
## Proportion of Variance 0.00031 0.00025 0.00019 0.00018 0.00011
## Cumulative Proportion  0.99926 0.99951 0.99971 0.99989 1.00000
```

PCA of GDP using covariance matrix :

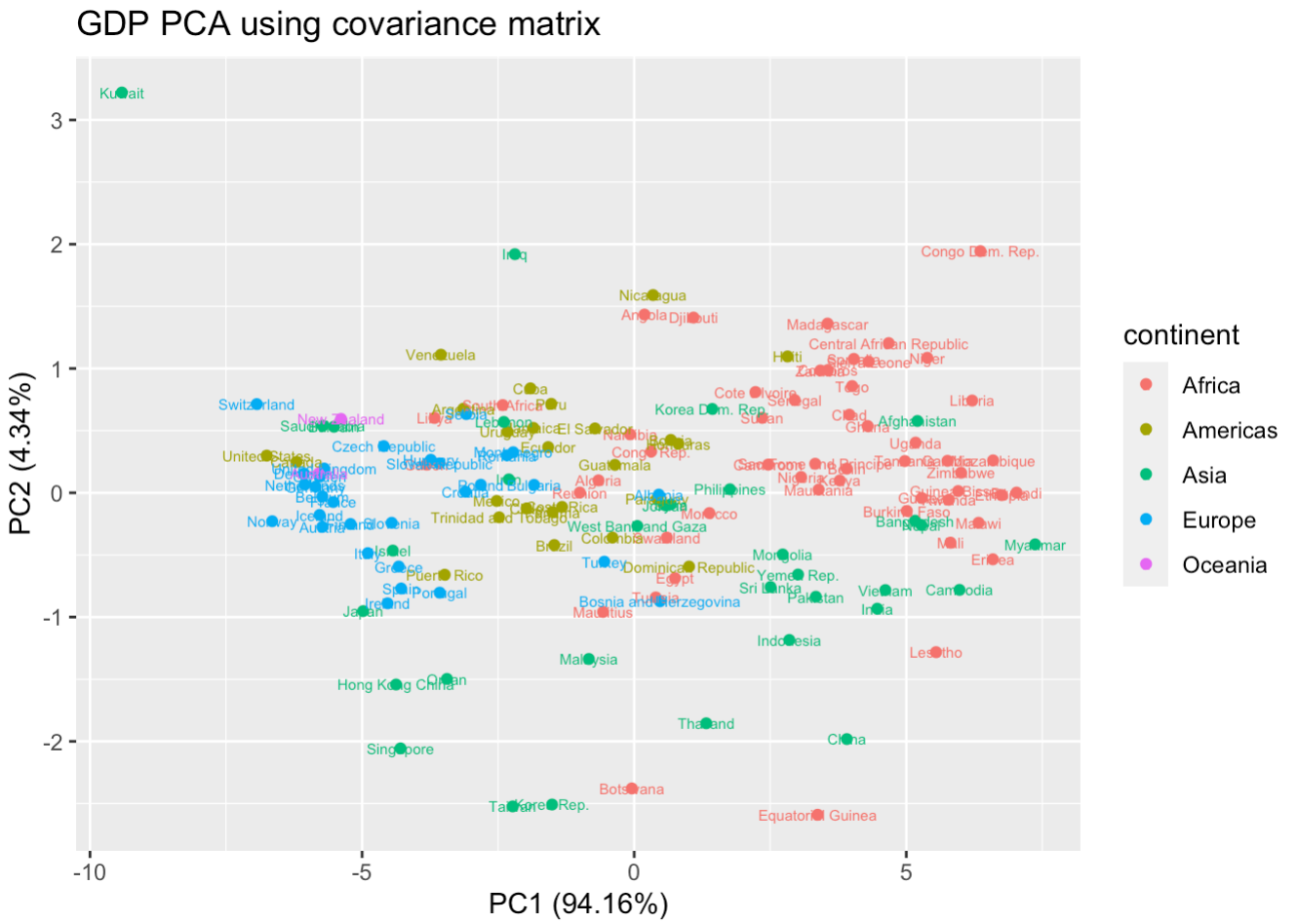
The proportion of variation for PC1 and PC2 is 94% and 4%. The combined proportion of variance explained by PC3 to PC4 vectors were only 2%.

```
#scree plot
plot(gdp_pca_cov, main = "PCA proportion of variance for GDP")
```



From the scree plot we can clearly see that the variance of the first two principal component cover most of the variance in the data. Hence we can retain first two principal components.

```
library(ggfortify)
autoplot(gdp_pca_cov, data = gap, colour="continent", scale=FALSE, label.label = "cou
ntry", label = TRUE, label.size = 2, main="GDP PCA using covariance matrix")
```



The PC1 demonstrates rate of growth trend in GDP from 1952 to 2007. Whereas, the PC2 depicts the initial GDP values of the countries. For example Kuwait, switzerland and United States had the largest GDP values in 1952. The countries from Africa seems to have highest rate of growth in GDP and the European countries have larger GDP values. However, rate of growth of the GDP for the European countries is very low when compared to countries from Asia.

```
#PCA using correlation matrix
gdp_pca_cor = prcomp(gdp_X, scale = TRUE)
summary(gdp_pca_cor)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.3592 0.73033 0.32047 0.18149 0.13622 0.09627 0.07515
## Proportion of Variance 0.9404 0.04445 0.00856 0.00274 0.00155 0.00077 0.00047
## Cumulative Proportion  0.9404 0.98481 0.99337 0.99612 0.99766 0.99844 0.99891
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.06154 0.05953 0.04728 0.04439 0.03990
## Proportion of Variance 0.00032 0.00030 0.00019 0.00016 0.00013
## Cumulative Proportion  0.99922 0.99952 0.99970 0.99987 1.00000
```

PCA of GDP using correlation matrix : we can see that there isn't much difference in proportion of variance and standard deviation of PCA using covariance or correlation. However, we would combine the PC score result of GDP and life expectancy in further analysis. Hence, if we use scaled data it will be helpful for analysis.

Principal Component Analysis for Life Expectancy

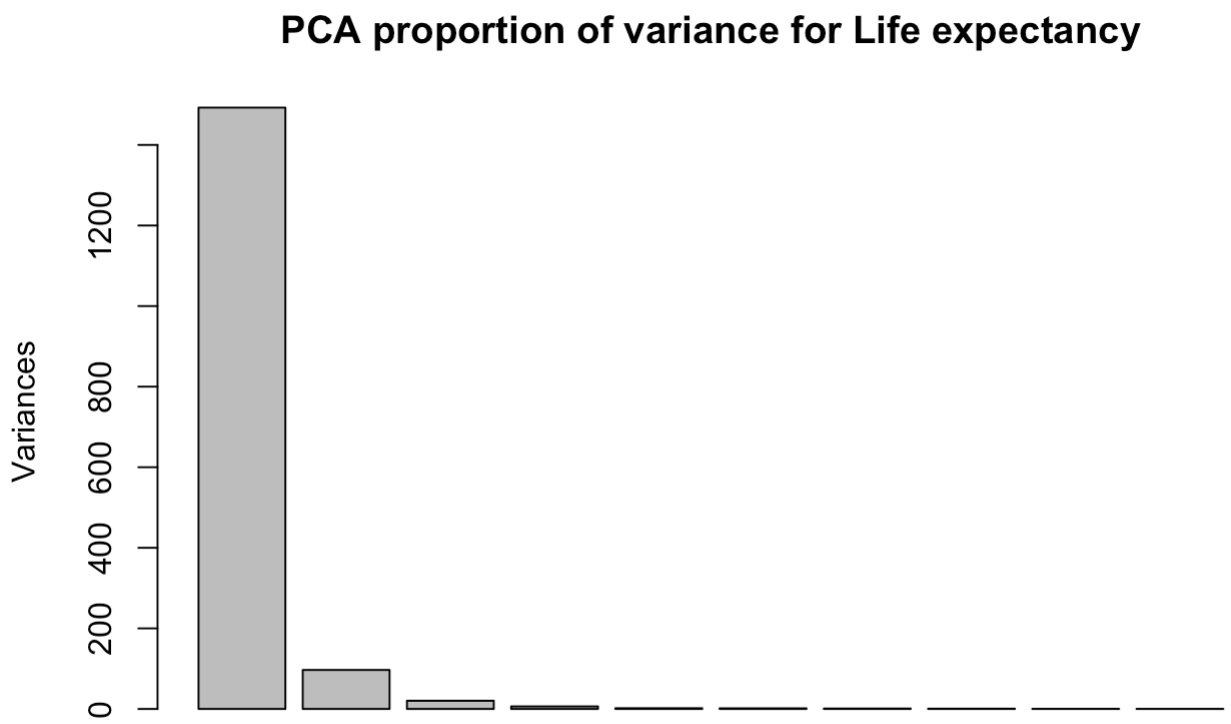
```
life_X = as.matrix(life_exp[3:14])
#PCA using covariance matrix
life_exp_cov = prcomp(life_X, scale = FALSE)
summary(life_exp_cov)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     38.6350 9.83955 4.50873 2.52549 1.39085 1.27733 1.01896
## Proportion of Variance  0.9203 0.05969 0.01253 0.00393 0.00119 0.00101 0.00064
## Cumulative Proportion   0.9203 0.97994 0.99247 0.99641 0.99760 0.99860 0.99924
##                             PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.79691 0.48377 0.43319 0.33836 0.23486
## Proportion of Variance 0.00039 0.00014 0.00012 0.00007 0.00003
## Cumulative Proportion  0.99964 0.99978 0.99990 0.99997 1.00000
```

PCA of Life Expectancy using covariance matrix :

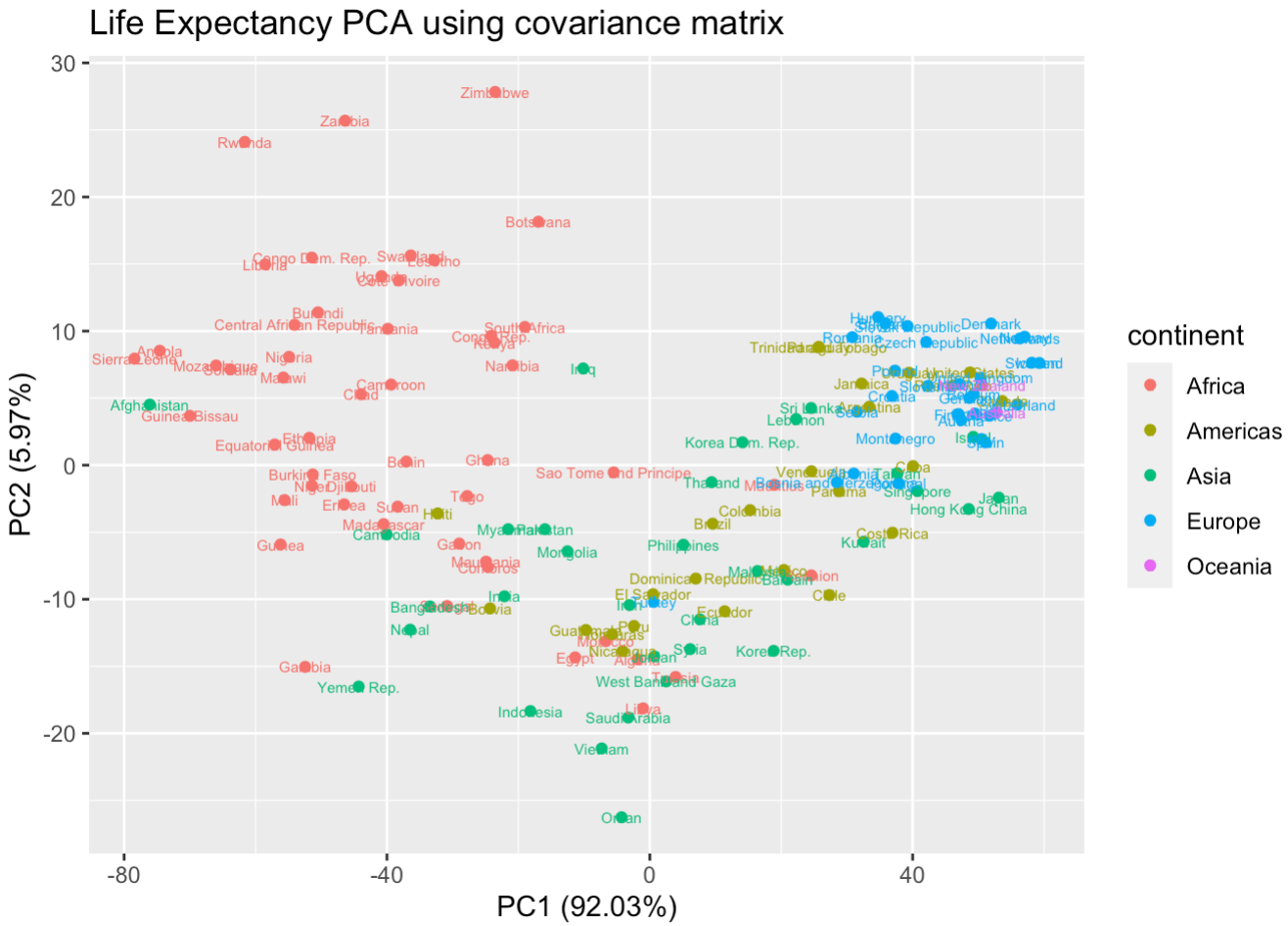
The proportion of variation for PC1 and PC2 is 92% and 5%. The combined proportion of variance explained by PC3 to PC4 vectors were only 3%.

```
#scree plot
plot(life_exp_cov, main = "PCA proportion of variance for Life expectancy")
```



From the scree plot for life expectancy we can depict that the variance of the first two principal component cover most of the variance in the data. Hence we can retain first two principal components.

```
autoplot(life_exp_cov, data = gap, colour="continent", scale=FALSE, label.label = "co
untry", label = TRUE, label.size = 2, main="Life Expectancy PCA using covariance matr
ix")
```



The PC1 demonstrates rate of growth trend in life expectancy from 1952 to 2007. Whereas, the PC2 depicts the initial life span of the countries. For example, most of the PC values for African countries were on Top left indicates that they had low average life expectancy in 1952 and there is no evident progress over the years in terms of life expectancy.

```
#PCA using correlation matrix
life_exp_cor = prcomp(life_X, scale = TRUE)
summary(life_exp_cor)
```

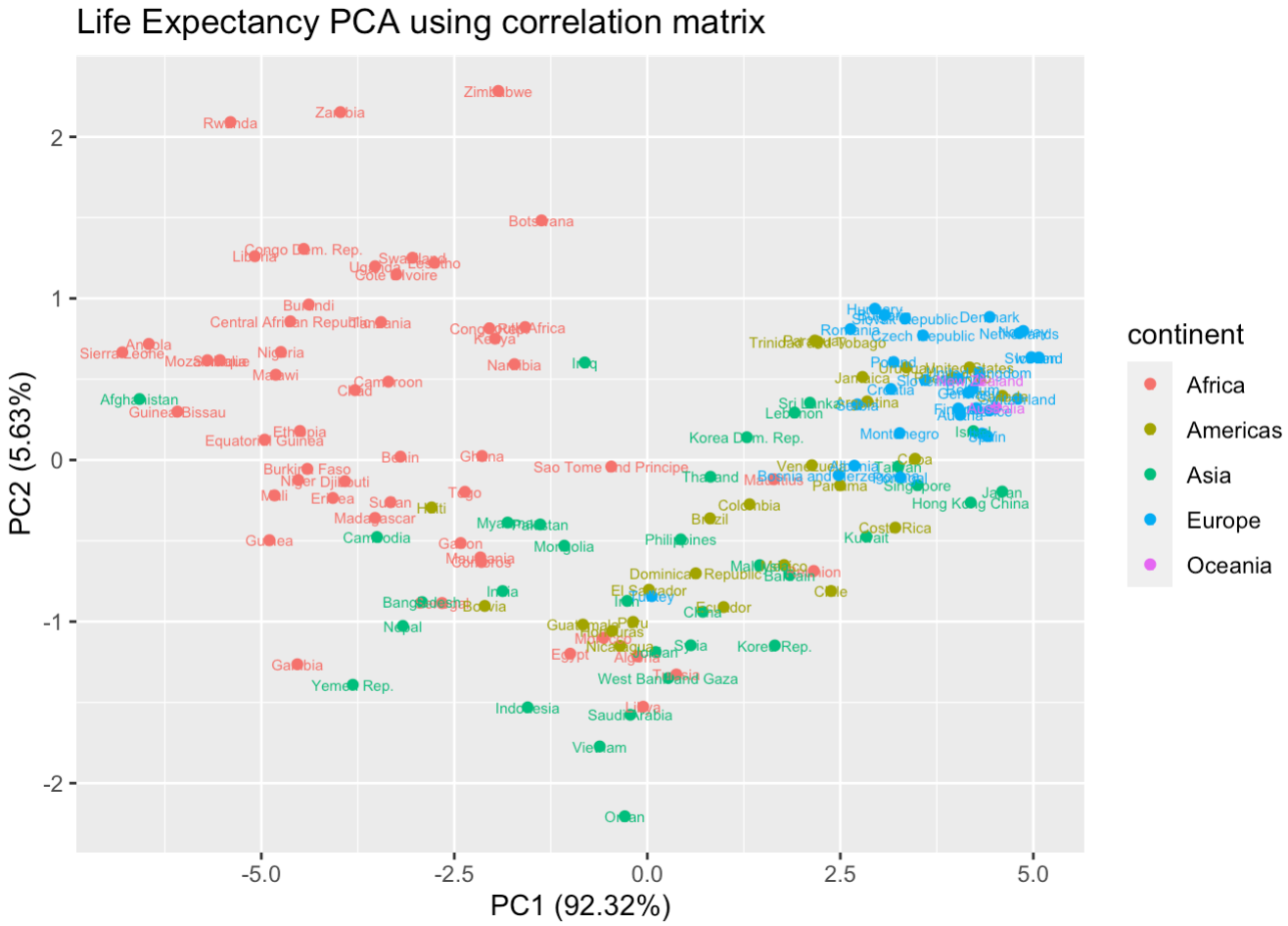
```
## Importance of components:
##                           PC1    PC2     PC3     PC4    PC5    PC6     PC7
## Standard deviation     3.3284 0.8219 0.39169 0.21950 0.1250 0.1098 0.08778
## Proportion of Variance 0.9232 0.0563 0.01279 0.00402 0.0013 0.0010 0.00064
## Cumulative Proportion  0.9232 0.9795 0.99229 0.99630 0.9976 0.9986 0.99925
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.06709 0.04413 0.03673 0.02815 0.02024
## Proportion of Variance 0.00038 0.00016 0.00011 0.00007 0.00003
## Cumulative Proportion  0.99963 0.99979 0.99990 0.99997 1.00000
```

PCA of life expectancy using correlation matrix :

we can see that there isn't much difference in proportion of variance but there is huge difference in standard deviation for PC1 score. The standard deviation of PC1 using covariance matrix it is 38.63 but using correlation matrix is 3.3. Most of the machine learning algorithm will work better if they have scaled data.

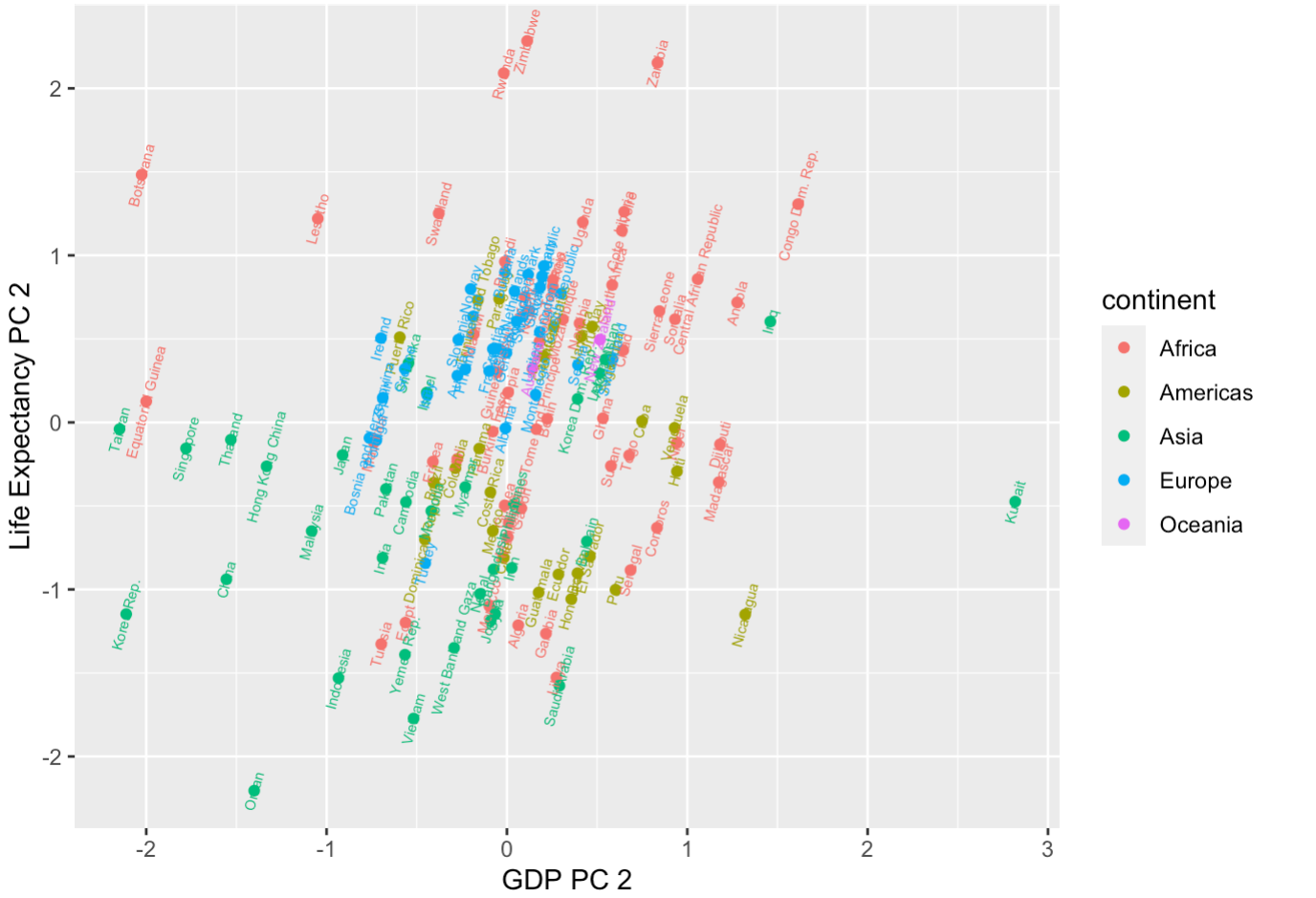
Hence we will use PCA using correlation matrix for further analysis.

```
autoplot(life_exp_cor, data = gap, colour="continent", scale=FALSE, label.label = "co
untry", label = TRUE, label.size = 2, main="Life Expectancy PCA using correlation mat
rix")
```

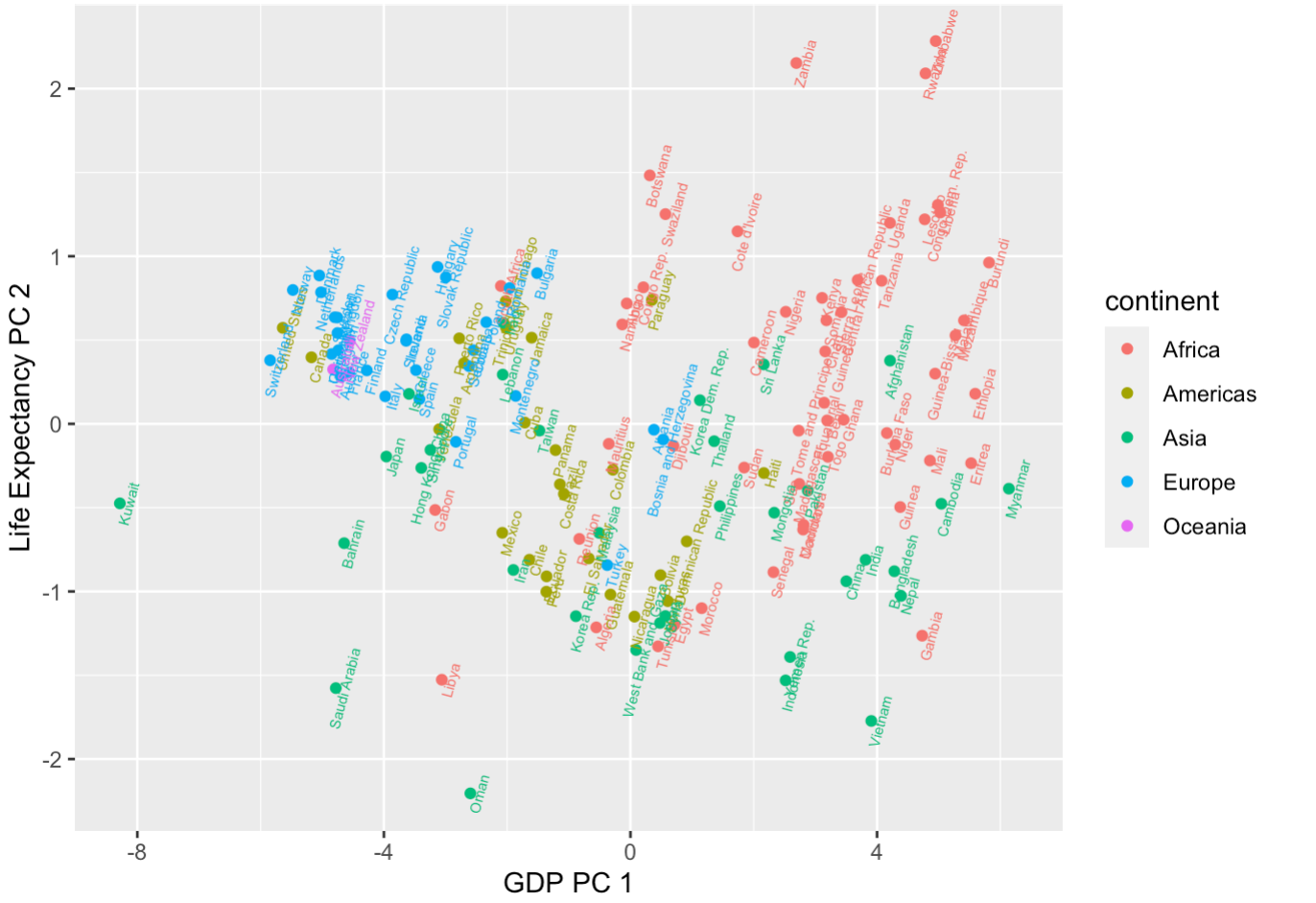


Combination of PCA vectors of GDP and Life Expectancy

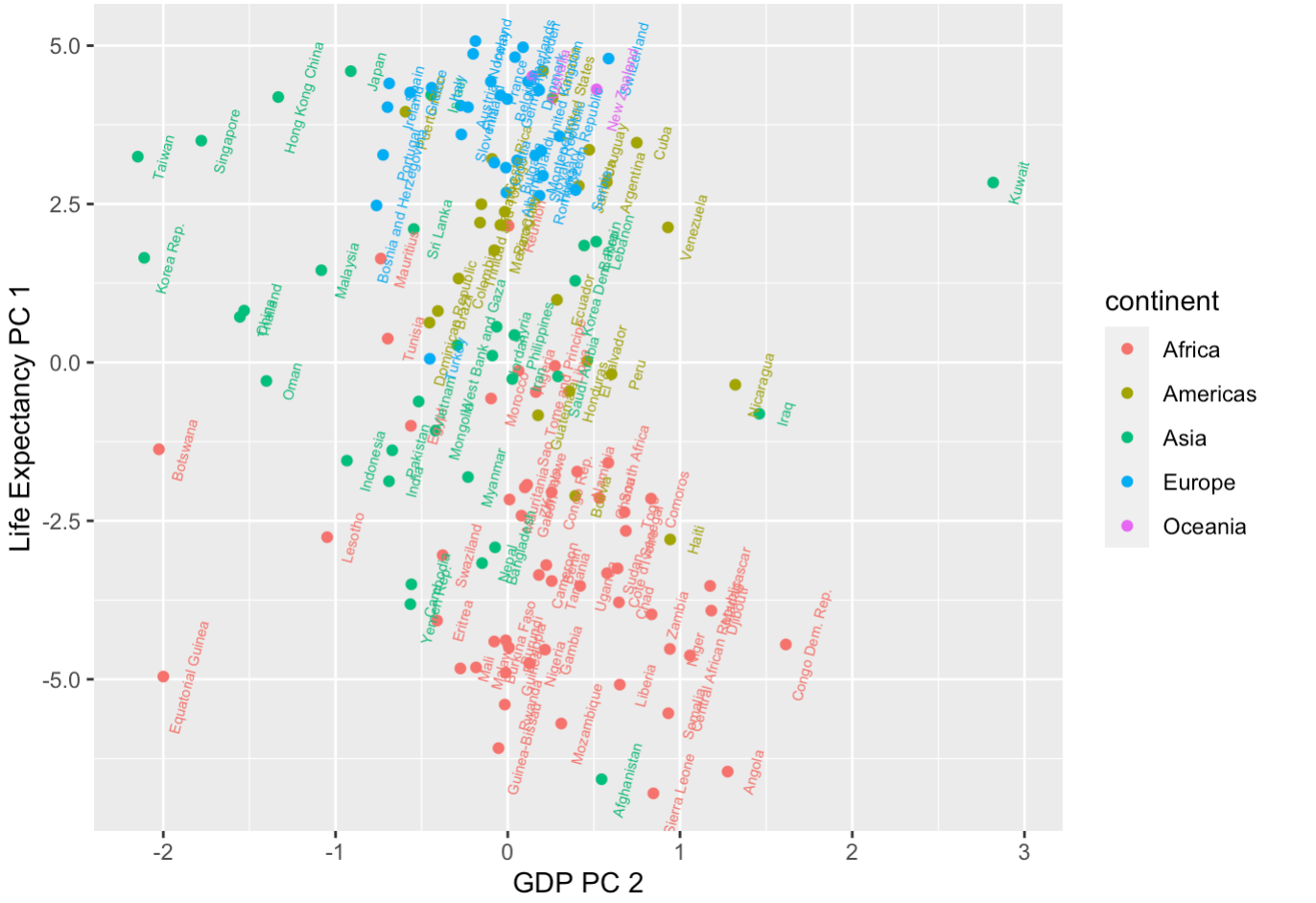
```
ggplot(gap, aes(x = gdp_pca_cor$x[, 2], y = life_exp_cor$x[, 2] , colour = continen
t))+geom_point() +geom_text(aes(label = country), size = 2, angle=75) + labs(x = "GDP
PC 2", y = "Life Expectancy PC 2")
```



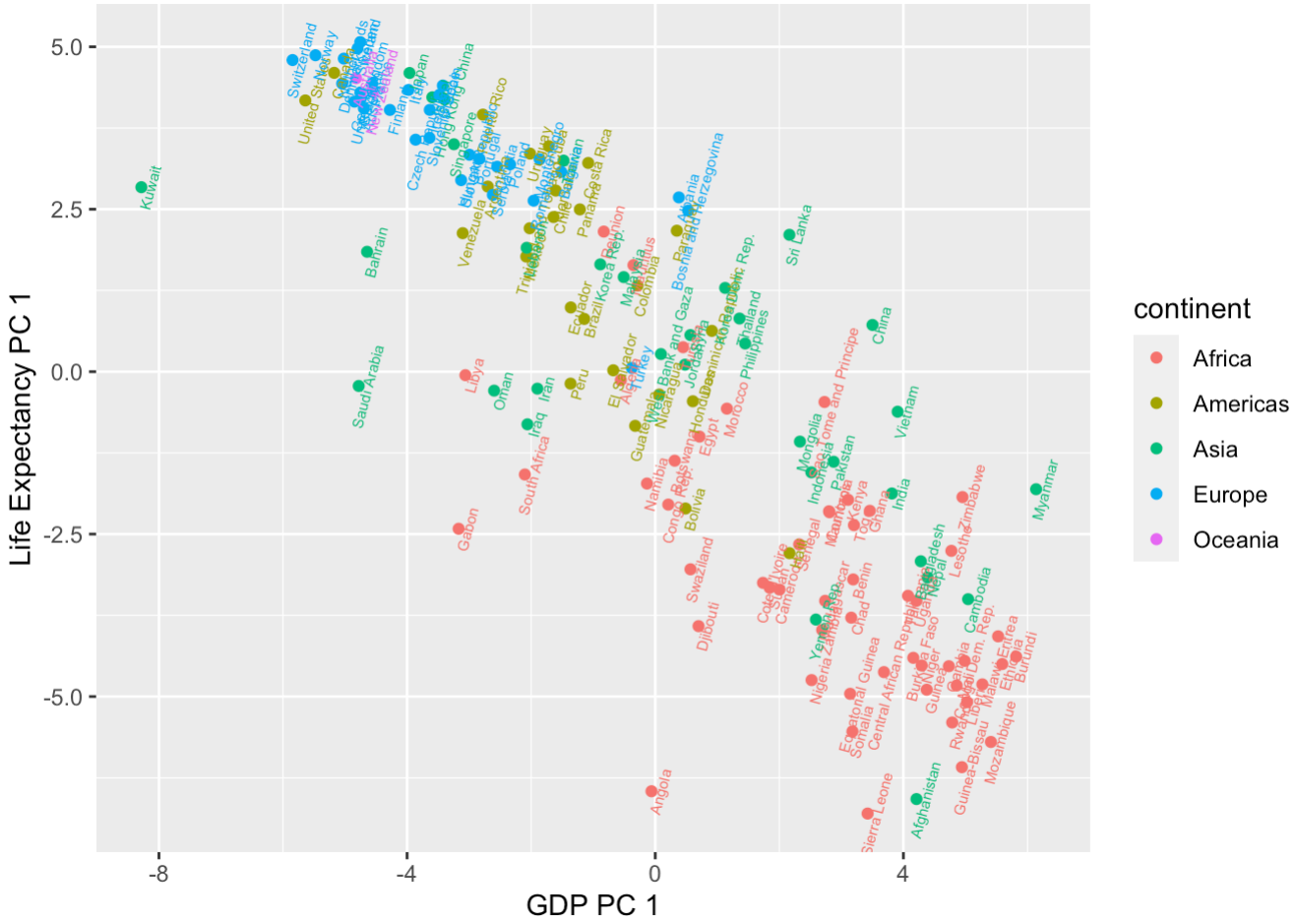

```
ggplot(gap, aes(x = gdp_pca_cor$x[, 1], y = life_exp_cor$x[, 2] , colour = continen
t))+geom_point() +geom_text(aes(x = gdp_pca_cor$x[, 1]+0.15, label = country), size =
2, angle=75) + labs(x = "GDP PC 1", y = "Life Expectancy PC 2")
```




```
ggplot(gap, aes(x = gdp_pca_cor$x[, 2], y = life_exp_cor$x[, 1] , colour = continen
t))+geom_point() +geom_text(aes(x = gdp_pca_cor$x[, 2]+0.15, label = country), size =
2, angle=75) + labs(x = "GDP PC 2", y = "Life Expectancy PC 1")
```




```
ggplot(gap, aes(x = gdp_pca_cor$x[, 1], y = life_exp_cor$x[, 1] , colour = continen
t))+geom_point() +geom_text(aes(x = gdp_pca_cor$x[, 1]+0.15, label = country), size =
2, angle=75) + labs(x = "GDP PC 1", y = "Life Expectancy PC 1")
```



GDP PC1 Vs Life Expectancy PC1 chart :

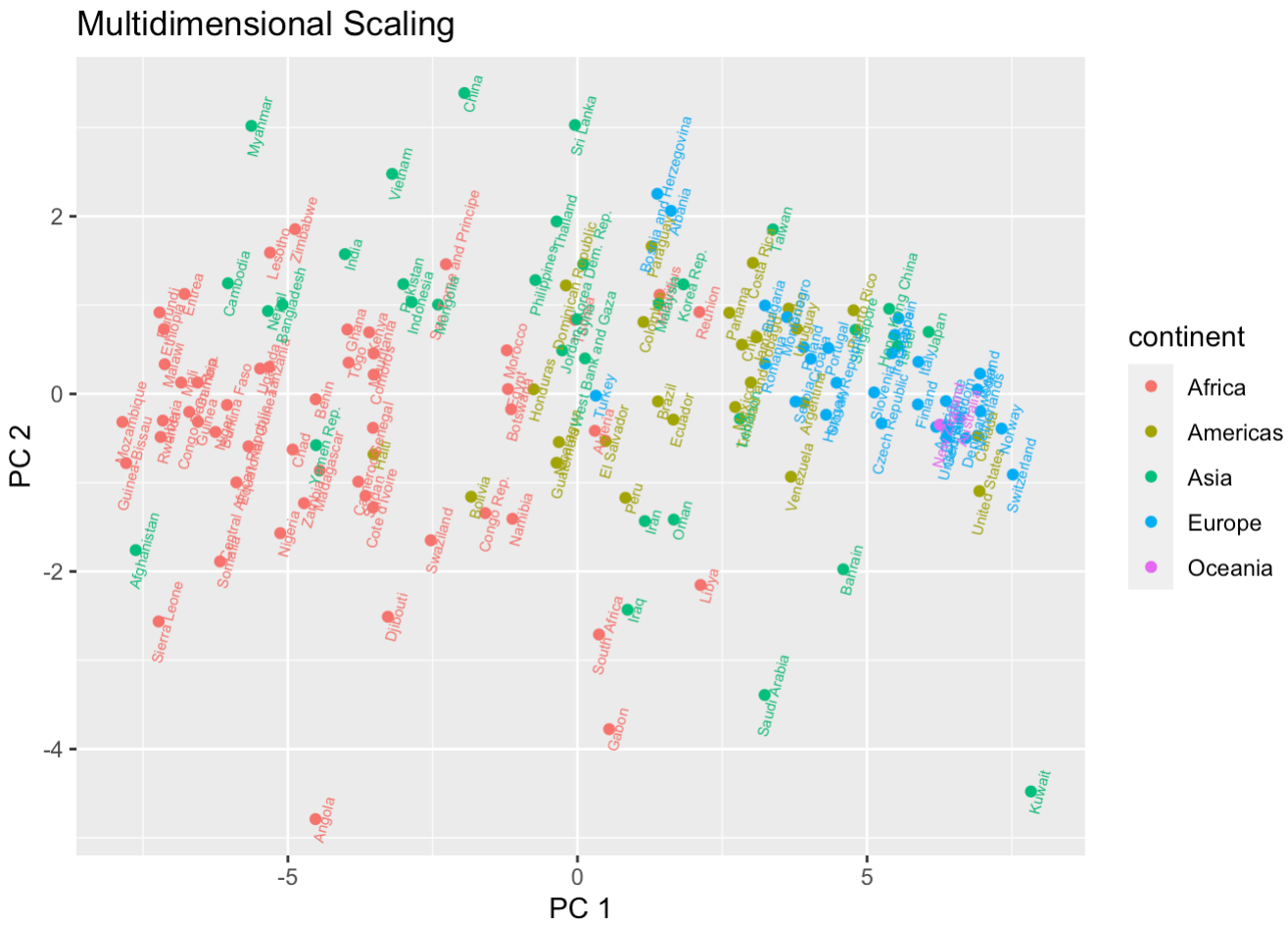
From the data we can observe rate of GDP growth of African countries is very high but the life expectancy rate is very low. The European countries showed more interest in improving living standard than GDP growth over the period of years. American countries gave equal importance to both GDP and living standard.

Multidimensional Scaling :

```
dist_data = dist(as.matrix(scale(log_gap[, 3:dim(gap)[2]])))
mul_dim_s = cmdscale(dist_data, eig = TRUE, k = 2)
```

Scaling the data ensures all the variables relationship have equal weight in the calculation of distance.

```
ggplot(gap, aes(x = mul_dim_s$points[,1], y = mul_dim_s$points[,2] , colour = contine
nt))+geom_point()+ labs(x = "PC 1", y = "PC 2") +geom_text(aes(x = mul_dim_s$points[,
1]+0.15, label = country), size = 2, angle=75) + ggtitle("Multidimensional Scaling")
```



The PC1 represents life span of the countries and the y axis represent rate of the GDP growth from 1952 to 2007.The X axis graph values are similar to PCA of life expectancy and the Y axis values can be related to the PCA values of GDP.

Hypothesis Test

Hypothesis Test for the year 2007

```
gdp_lif_2007 = gap[c("gdpPercap_2007", "lifeExp_2007", "continent")]
gdp_lif_2007["gdpPercap_2007"]  = log(gdp_lif_2007["gdpPercap_2007"])
gdp_lif_2007_asian = subset(gdp_lif_2007, continent == "Asia")
gdp_lif_2007_europe = subset(gdp_lif_2007, continent == "Europe")
```

Null Hypothesis H_0 : The mean GDP and average life span of European and Asian were same in the year 2007.

Alternate Hypothesis H_1 : The mean GDP and average life span of European and Asian countries were different in the year 2007.

```
library(ICSNP)
HotellingsT2(gdp_lif_2007_asian[c("gdpPercap_2007","lifeExp_2007")], gdp_lif_2007_eur
ope[c("gdpPercap_2007","lifeExp_2007")])
```

```
##
##  Hotelling's two sample T2-test
##
## data:  gdp_lif_2007_asian[c("gdpPercap_2007", "lifeExp_2007")] and gdp_lif_2007_eu
rope[c("gdpPercap_2007", "lifeExp_2007")]
## T.2 = 12.681, df1 = 2, df2 = 60, p-value = 2.55e-05
## alternative hypothesis: true location difference is not equal to c(0,0)
```

The P-value is $p = 2.55e^{-5}$ and $\delta^2 = 12.681$

```
qf(0.95,2, 60)
```

```
## [1] 3.150411
```

The critical value for $\alpha = 0.05$ is $F_{2,60,0.005} = 3.150411$

The probability value is $2.55e^{-5}$ which is less than 0.05 and $\delta^2 > F_{2,60,0.005}$. Hence we will reject the Null Hypothesis. Therefore the mean value of GDP and life expectancy of European and Asian countries were different in the year 2007.

Hypothesis Test for the year 1952

```
gdp_lif_1952 = gap[c("gdpPercap_1952", "lifeExp_1952", "continent")]
gdp_lif_1952["gdpPercap_1952"]  = log(gdp_lif_1952["gdpPercap_1952"])
gdp_lif_1952_asian = subset(gdp_lif_1952, continent == "Asia")
gdp_lif_1952_europe = subset(gdp_lif_1952, continent == "Europe")
```

Null Hypothesis H_0 : The mean GDP and average life span of European and Asian were same in the year 1952.

Alternate Hypothesis H_1 : The mean GDP and average life span of European and Asian countries were different in the year 1952.

```
HotellingsT2(gdp_lif_1952_asian[c("gdpPercap_1952","lifeExp_1952")], gdp_lif_1952_eur
ope[c("gdpPercap_1952","lifeExp_1952")])
```

```
##
##  Hotelling's two sample T2-test
##
## data:  gdp_lif_1952_asian[c("gdpPercap_1952", "lifeExp_1952")] and gdp_lif_1952_eu
rope[c("gdpPercap_1952", "lifeExp_1952")]
## T.2 = 39.347, df1 = 2, df2 = 60, p-value = 1.21e-11
## alternative hypothesis: true location difference is not equal to c(0,0)
```

The P-value is $p = 1.21e^{-11}$ and $\delta^2 = 39.347$

The critical value for $\alpha = 0.05$ is $F_{2,60,0.005} = 3.150411$

The probability value is $1.21e^{-11}$ which is less than 0.05 and $\delta^2 > F_{2,60,0.005}$. Hence we will reject the Null Hypothesis. Therefore the mean value of GDP and life expectancy of European and Asian countries were different in the year 1952.

There is an increase in the probability value in 2007 when compared to the year 1952. Hence, the mean value of GDP and life expectancy of European and Asian countries are moving closer over the period.

Linear Discriminant Analysis

```
library(MASS)
library(caTools)


copy_gap = data.frame(gap)
copy_gap[, 3:14] = log(copy_gap[, 3:14])
copy_gap = copy_gap[-2]

train_index = sample.split(Y = copy_gap, SplitRatio = 0.8)
train = copy_gap[train_index, ]
test = copy_gap[!train_index, ]

lda_model = lda(train[, 2:length(train)], train[,1])
```

```
lda_pred = predict(lda_model, test[, 2:length(test)])
accuracy_lda = sum(lda_pred$class== test$continent)/dim(test)[1]*100
accuracy_lda = round(accuracy_lda)
```

The accuracy of the LDA model is 59%

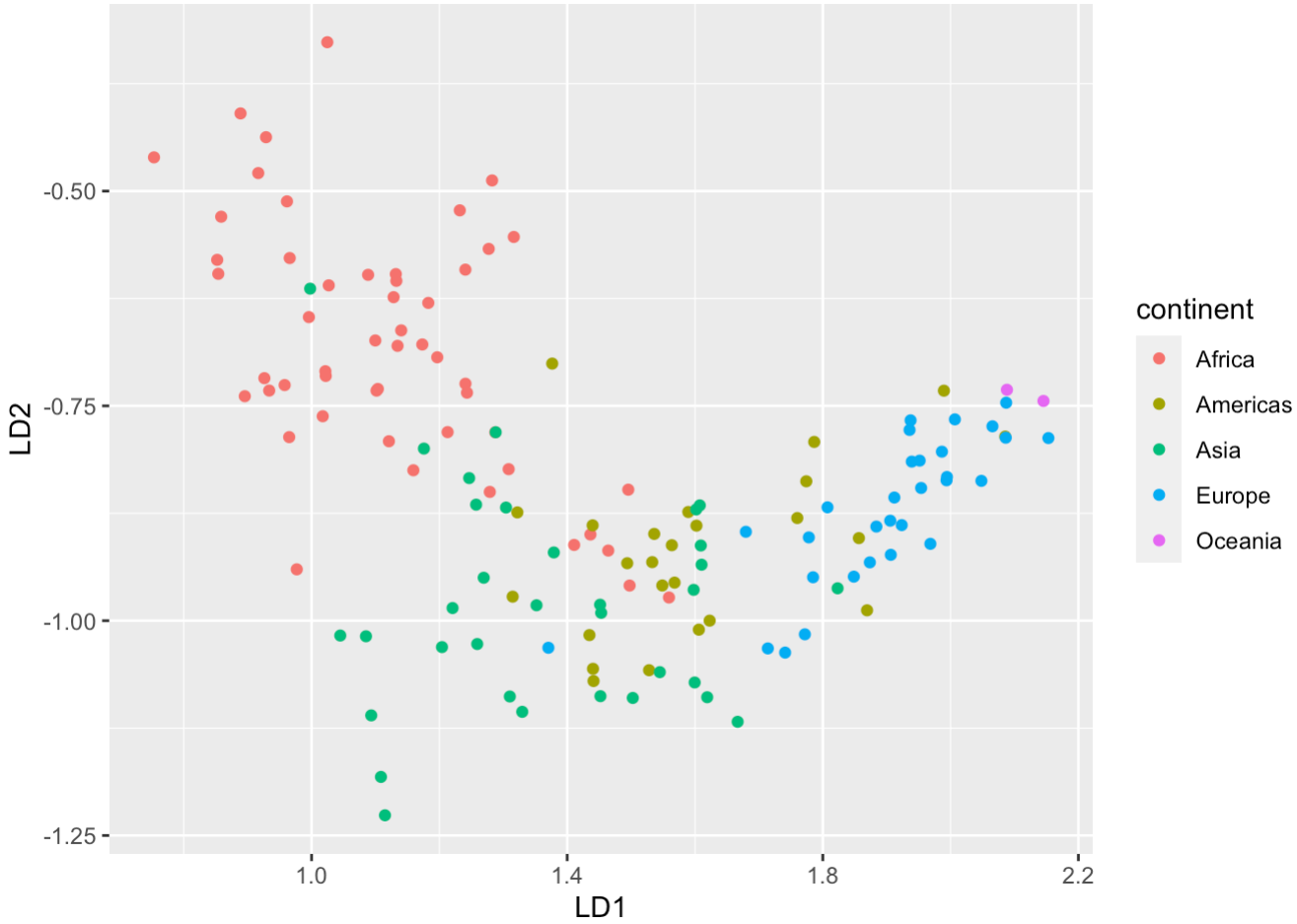
```
table(lda_pred$class, test$continent)
```

```
##
##            Africa Americas Asia Europe Oceania
##   Africa        8        1    2      0       0
##   Americas      1        2    0      1       0
##   Asia          1        0    2      0       0
##   Europe        1        2    1      5       2
```

The Rows represent the number of predicted countries and the columns represent the number of actual countries in a continent.

Dimensionality Reduction using Fisher's Linear Discriminant Rule

```
library(vcvComp)
B=cov.B(copy_gap[,2:length(train)], copy_gap[,1])
W=cov.W(copy_gap[,2:length(train)], copy_gap[,1])
gap_eig <- eigen(solve(W)%*% B)
V <- gap_eig$vectors[,1:2]
Z <- as.matrix(copy_gap[,2:length(copy_gap)])%*% V
ggplot2::qplot(as.numeric(Z[,1]), as.numeric(Z[,2]), colour=copy_gap$continent, xlab
='LD1',
               ylab='LD2')+ labs(colour = "continent")
```



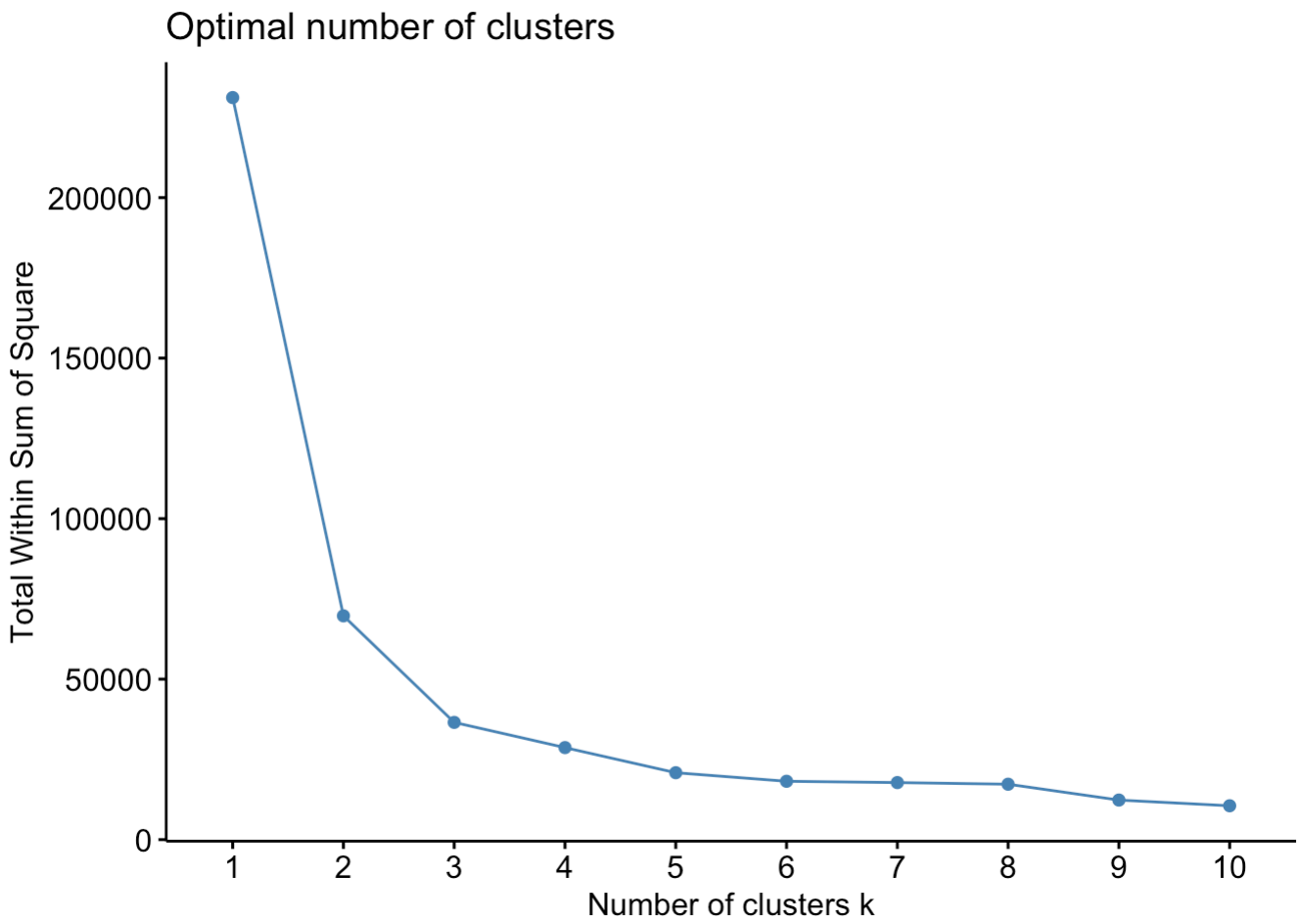
The plot shows that LDA projected vectors can be used to separate continents easily when compared to vectors using Principal Component Scores. The LDA vectors main objective is to maximize the between-class variance and increase the within-class variance.

Clustering

K means Clustering

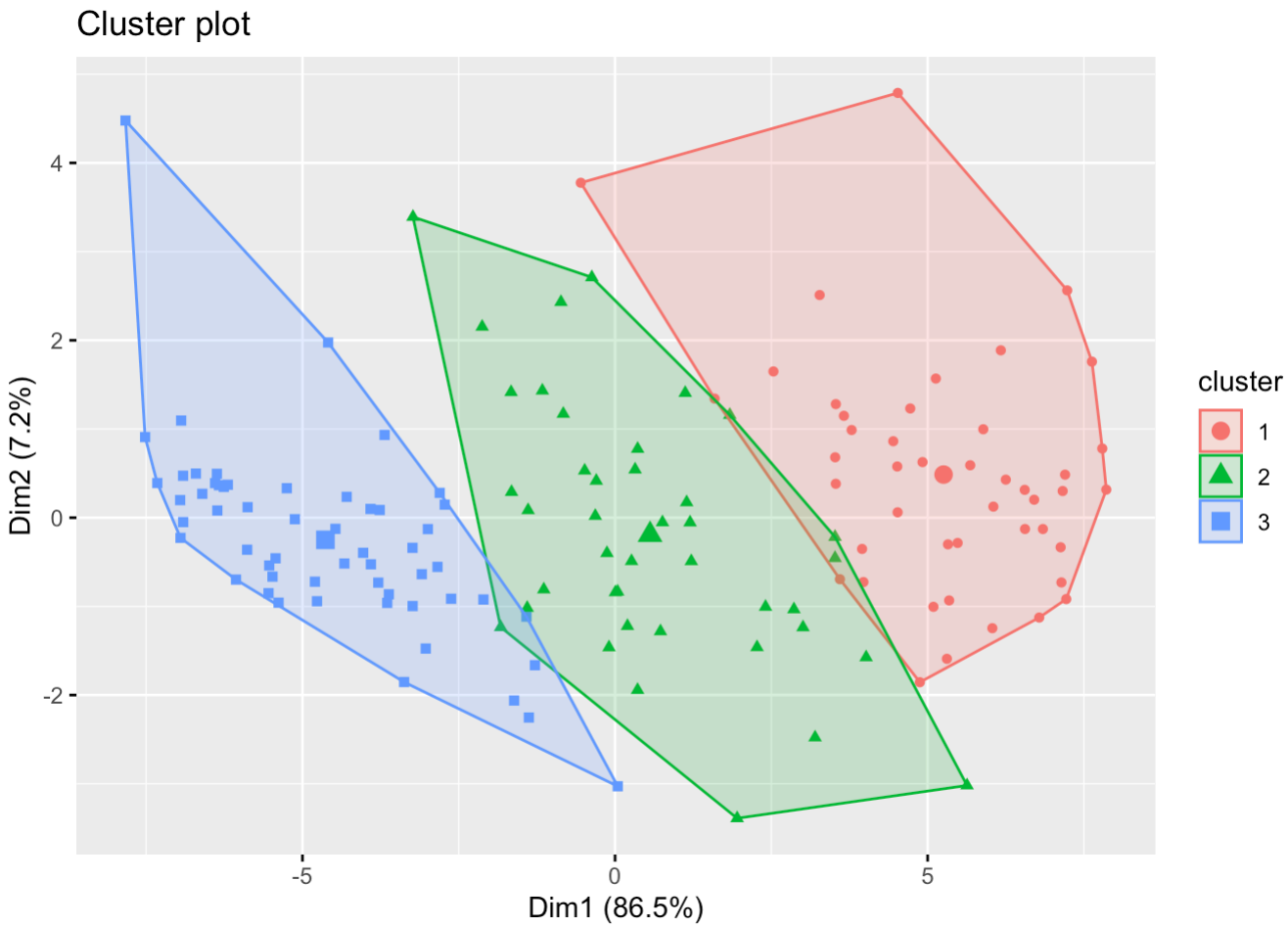
```
library(factoextra)

fviz_nbclust(copy_gap[, 2:length(copy_gap)], kmeans, method = "wss")
```



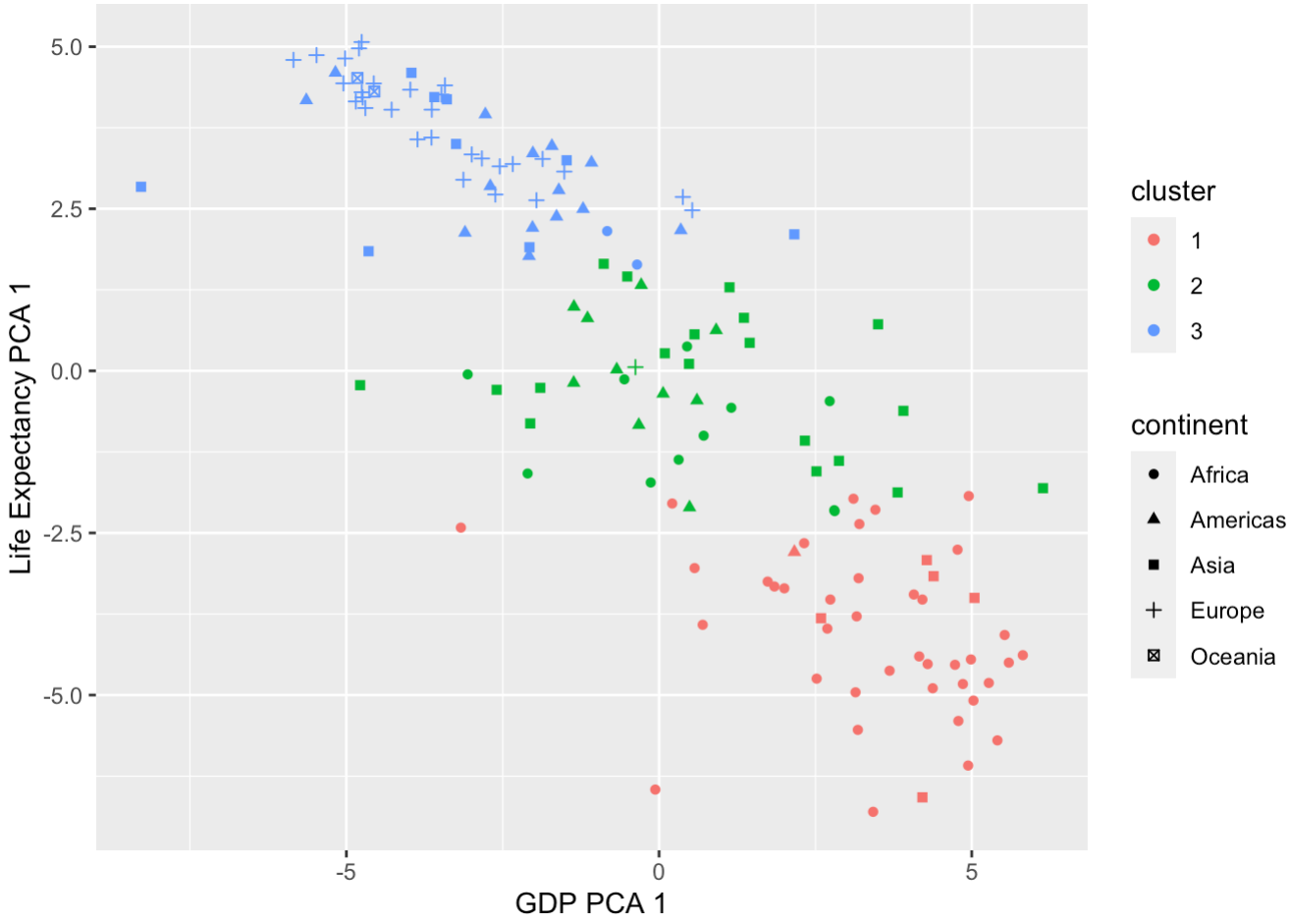
From the plot we can decide there are 3 clusters based on Total within sum of squares in clusters. There is only a minor improvement when we move from cluster 3 to 4. However, there is a slight improvement when we move from cluster 4 to 10 we can neglect those clusters because there is no major improvement in total sum of squares within clusters.

```
kmeanss = kmeans(copy_gap[, 2:length(copy_gap)], centers = 3, nstart = 20)
fviz_cluster(kmeanss, data = copy_gap[, 2:length(copy_gap)], geom = "point")
```



The Log of GDP values were taken to remove outliers.

```
gap_clone = data.frame(copy_gap)
gap_clone$cluster <- factor(kmeanss$cluster)
ggplot(gap_clone, aes(x = gdp_pca_cor$x[, 1], y = life_exp_cor$x[, 1] , colour = clus
ter, shape = continent))+geom_point() + labs(x = "GDP PCA 1", y = "Life Expectancy PC
A 1")
```

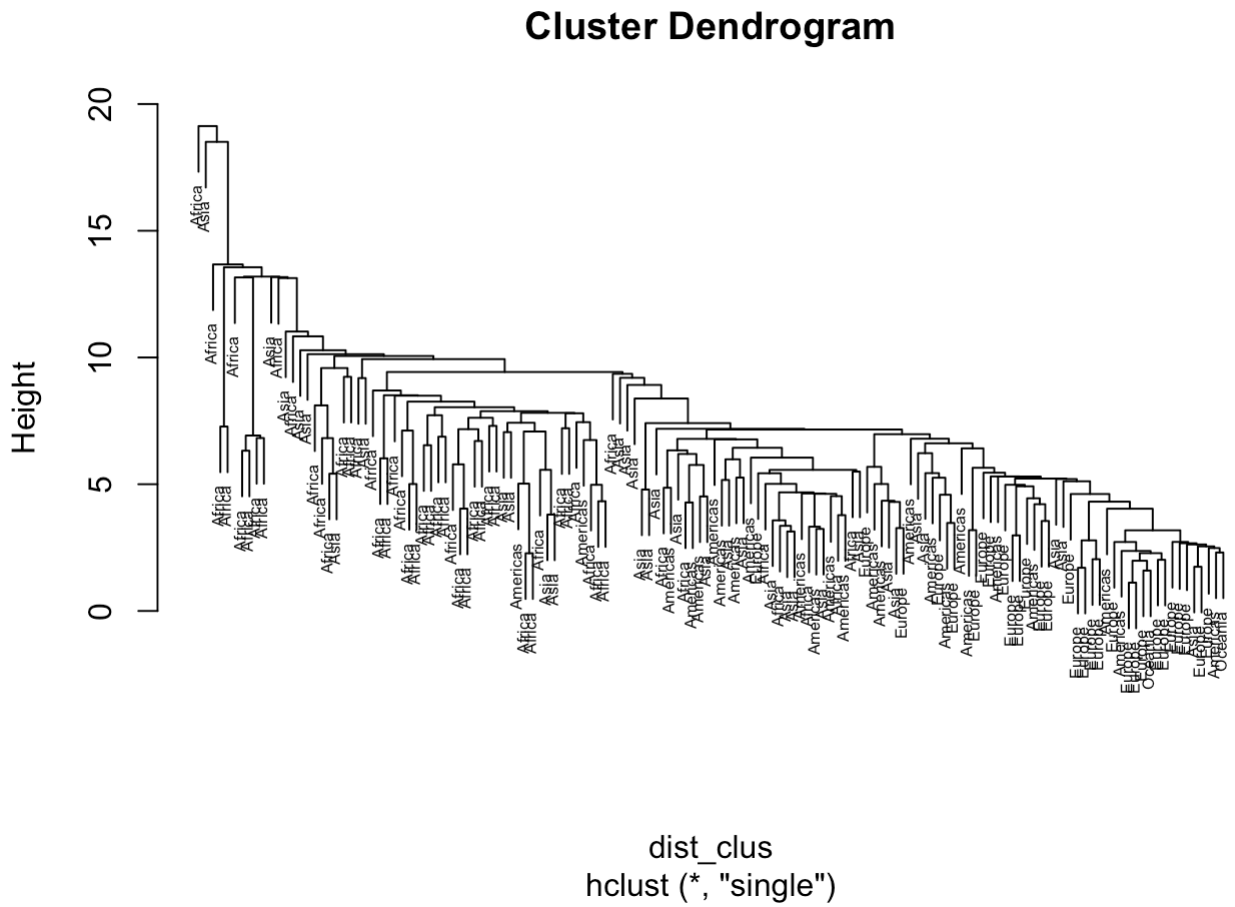


Agglomerative Hierarchical Clustering

```
data_clus = copy_gap[, 2:length(copy_gap)]
mat_clus = as.matrix(data_clus)
dist_clus = dist(mat_clus, method = "euclidean")
```

Hierarchical clustering using Single linkage

```
hier_clus_sing = hclust(dist_clus, method="single")
plot(hier_clus_sing, labels = gap$continent, cex = 0.5)
```



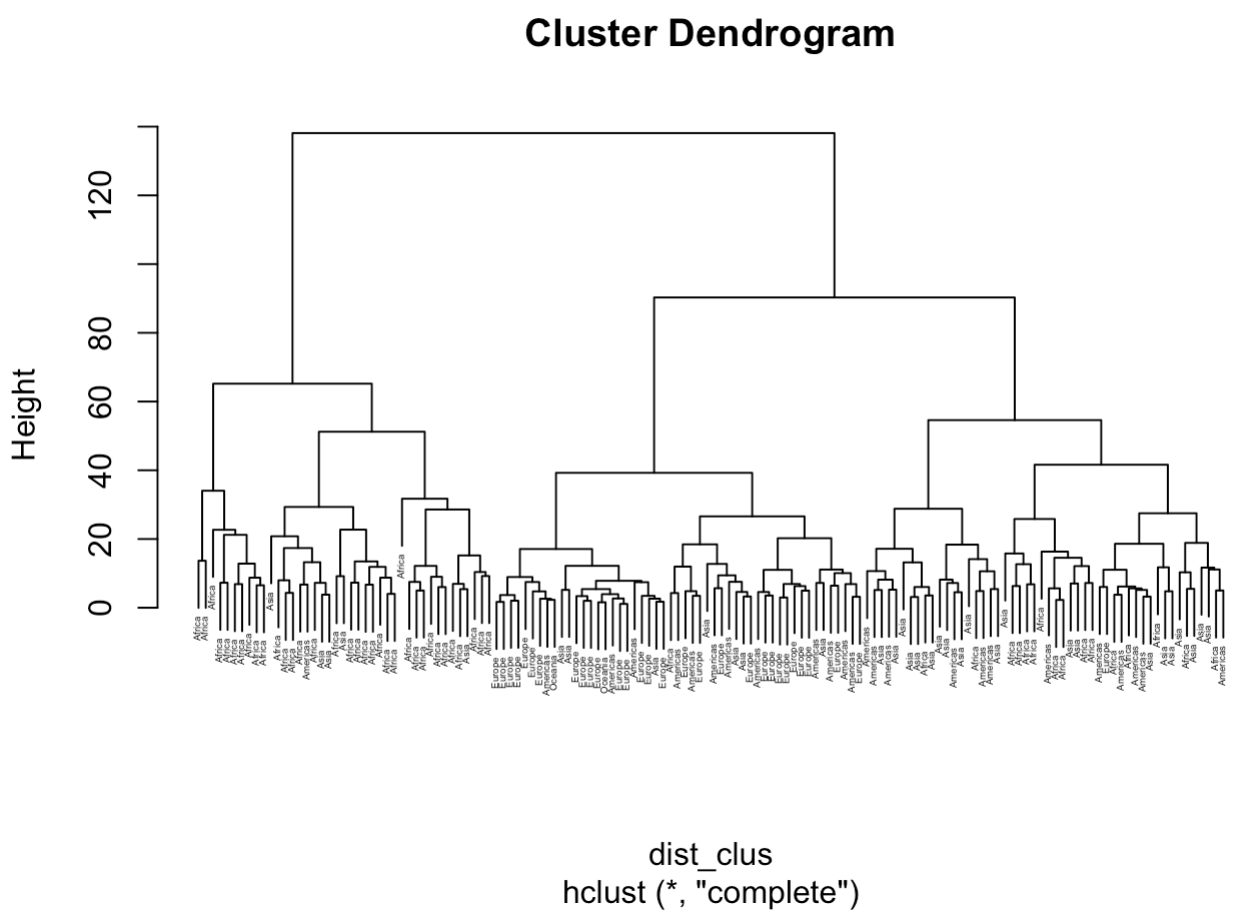
Single Linkage Confusion matrix

```
table(cutree(hier_clus_sing, k=3), gap$continent)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     51       25   32     30       2
##   2      1        0    0      0       0
##   3      0        0    1      0       0
```

Hierarchical clustering using complete linkage

```
hier_clus_comp = hclust(dist_clus, method="complete")
plot(hier_clus_comp, labels = gap$continent, cex=0.3)
```



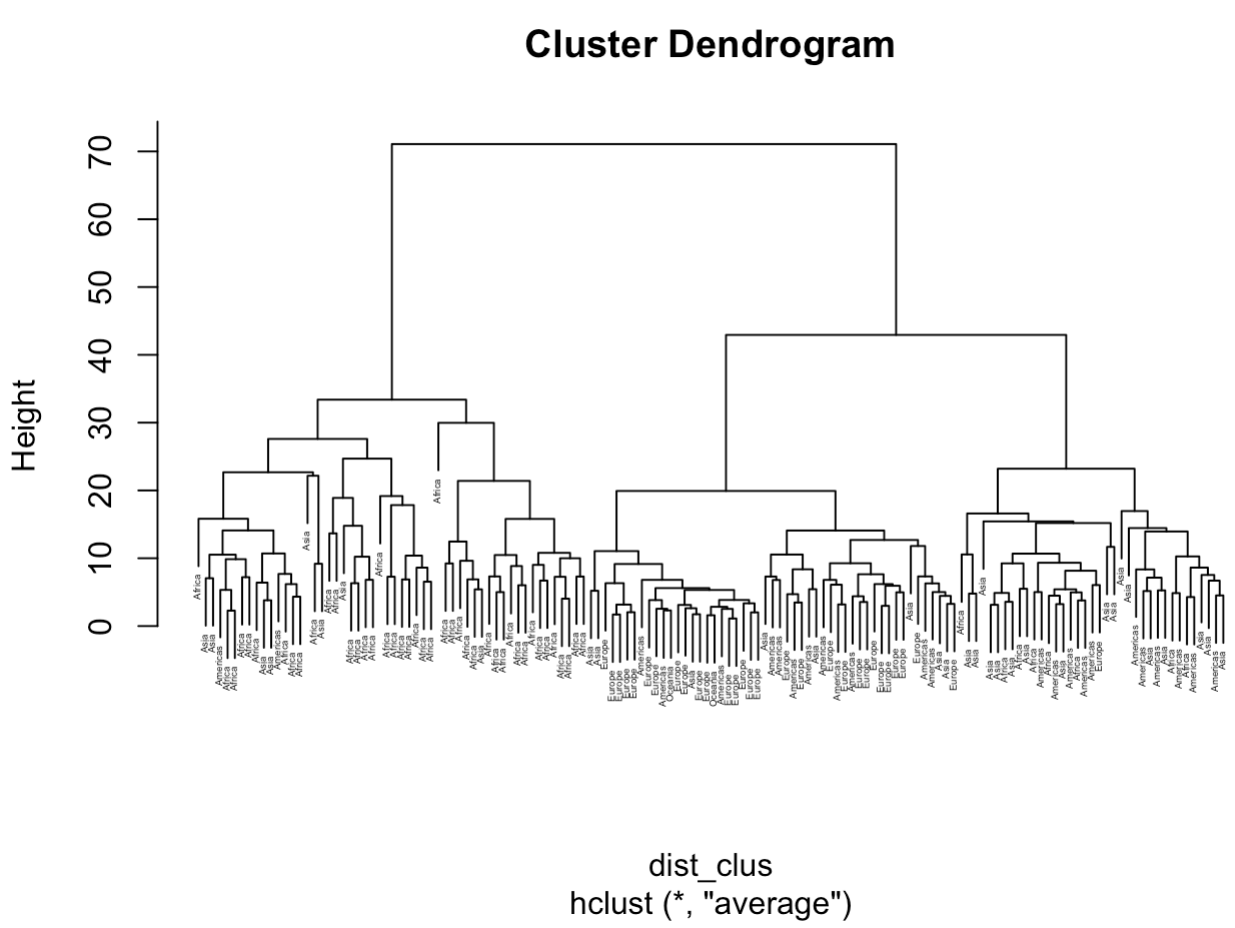
Complete Linkage confusion matrix

```
table(cutree(hier_clus_comp, k=3), gap$continent)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     16       12   21      1       0
##   2     35        1    5      0       0
##   3      1       12    7     29       2
```

Hierarchical clustering using Group Average

```
hier_clus_avg = hclust(dist_clus, method="average")
plot(hier_clus_avg, labels = gap$continent, cex=0.3)
```



Group average confusion matrix

```
table(cutree(hier_clus_avg, k=3), gap$continent)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1      8       11   17      1       0
##   2     44        2    8      0       0
##   3      0       12    8     29       2
```

The dendorgrams shows that complete linkage and group average method able to cluster similar countries that belongs to same continent. However, the confusion matrix shows that group average method performs well compared to complete linkage method.

Compare K means with Hierarchial clustering

Kmeans clustering confusion matrix

```
table(gap_clone$cluster, gap$continent)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     39        1    5      0       0
##   2     11       10   19      1       0
##   3      2       14    9     29       2
```

The hierarchical clustering using average method performs better than K-means clustering. The log values of GDP values was taken to remove outlier. This method gives high accuracy of clustering when compared to scaled data. The countries of Africa, Europe and Asia continents mostly cluster together naturally. However, countries of America and Oceania continents don't cluster naturally

Linear Regression

```
library(pls)
gdp_data  = gap[, 3:14]
lifeExp_2007 = gap["lifeExp_2007"]
merg_data = cbind(gdp_data,lifeExp_2007 )
```

```
train_index = sample(1:142, size = 30, replace = FALSE)
lin_mod = lm(lifeExp_2007 ~ ., data = merg_data[train_index,])
test_pred = predict(lin_mod, merg_data[-train_index, 1:length(merg_data)-1])
summary(lin_mod)
```

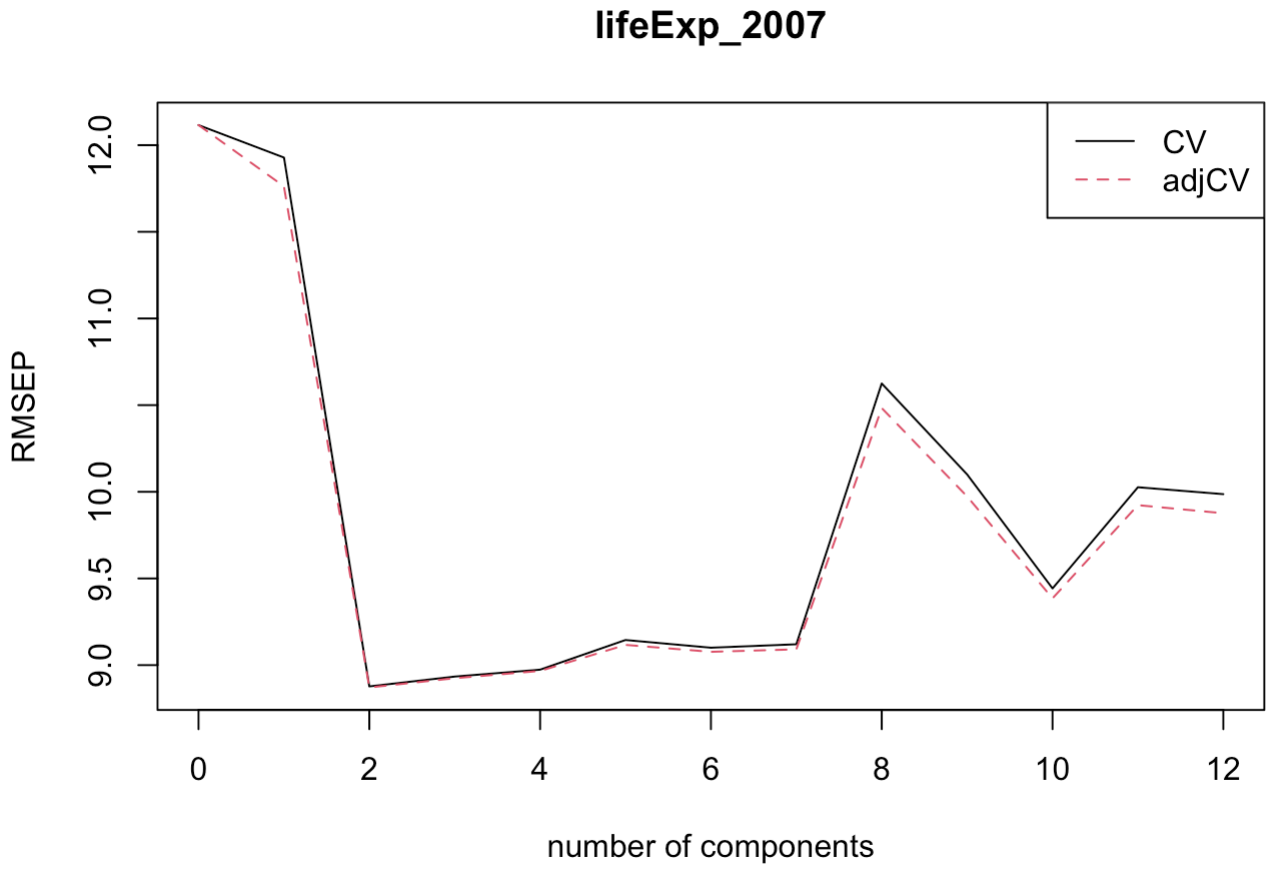
```
##
## Call:
## lm(formula = lifeExp_2007 ~ ., data = merg_data[train_index,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.2511  -4.3252   0.3541   4.9977  14.3350
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    53.371699   3.574291  14.932 3.33e-11 ***
## gdpPercap_1952  0.018959   0.010417   1.820   0.0864 .
## gdpPercap_1957 -0.015792   0.015565  -1.015   0.3245
## gdpPercap_1962  0.011877   0.012389   0.959   0.3512
## gdpPercap_1967 -0.019442   0.009449  -2.057   0.0553 .
## gdpPercap_1972 -0.003579   0.005451  -0.657   0.5202
## gdpPercap_1977  0.004671   0.005097   0.916   0.3723
## gdpPercap_1982  0.002829   0.007738   0.366   0.7192
## gdpPercap_1987  0.002812   0.010350   0.272   0.7891
## gdpPercap_1992 -0.002618   0.005607  -0.467   0.6465
## gdpPercap_1997  0.009405   0.005707   1.648   0.1177
## gdpPercap_2002 -0.012281   0.006359  -1.931   0.0703 .
## gdpPercap_2007  0.006166   0.003559   1.732   0.1013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.267 on 17 degrees of freedom
## Multiple R-squared:  0.6468, Adjusted R-squared:  0.3975
## F-statistic: 2.594 on 12 and 17 DF,  p-value: 0.03564
```

```
errors <- test_pred- merg_data[-train_index, length(merg_data)]
sq_error_lin = sqrt(mean((errors)^2))
```

The Root mean square error using Linear regression model is 31.8707029

Principal Component Regression

```
pcr_mod_2 = pcr(lifeExp_2007 ~ ., data=merg_data, validation='CV', scale = TRUE)
plot(RMSEP(pcr_mod_2), legendpos = "topright")
```



As we can observe that using 2 principal component will yield better accuracy for Principal Component Regression model

```
pca_life_exp = prcomp(merg_data[ , 1:length(merg_data)-1])
pc_scores = pca_life_exp$x[, 1:2]

train = pc_scores[train_index, ]
test = pc_scores[-train_index, ]
pca_train_data = data.frame(y = merg_data[train_index, ]$lifeExp_2007, x= train)
pca_lm_mod = lm(y ~ ., data = pca_train_data)

test_data = data.frame(x = test)
test_pred = predict(pca_lm_mod, test_data)
errors <- test_pred- merg_data[-train_index, ]$lifeExp_2007
sq_error_pc = sqrt(mean((errors)^2))
```

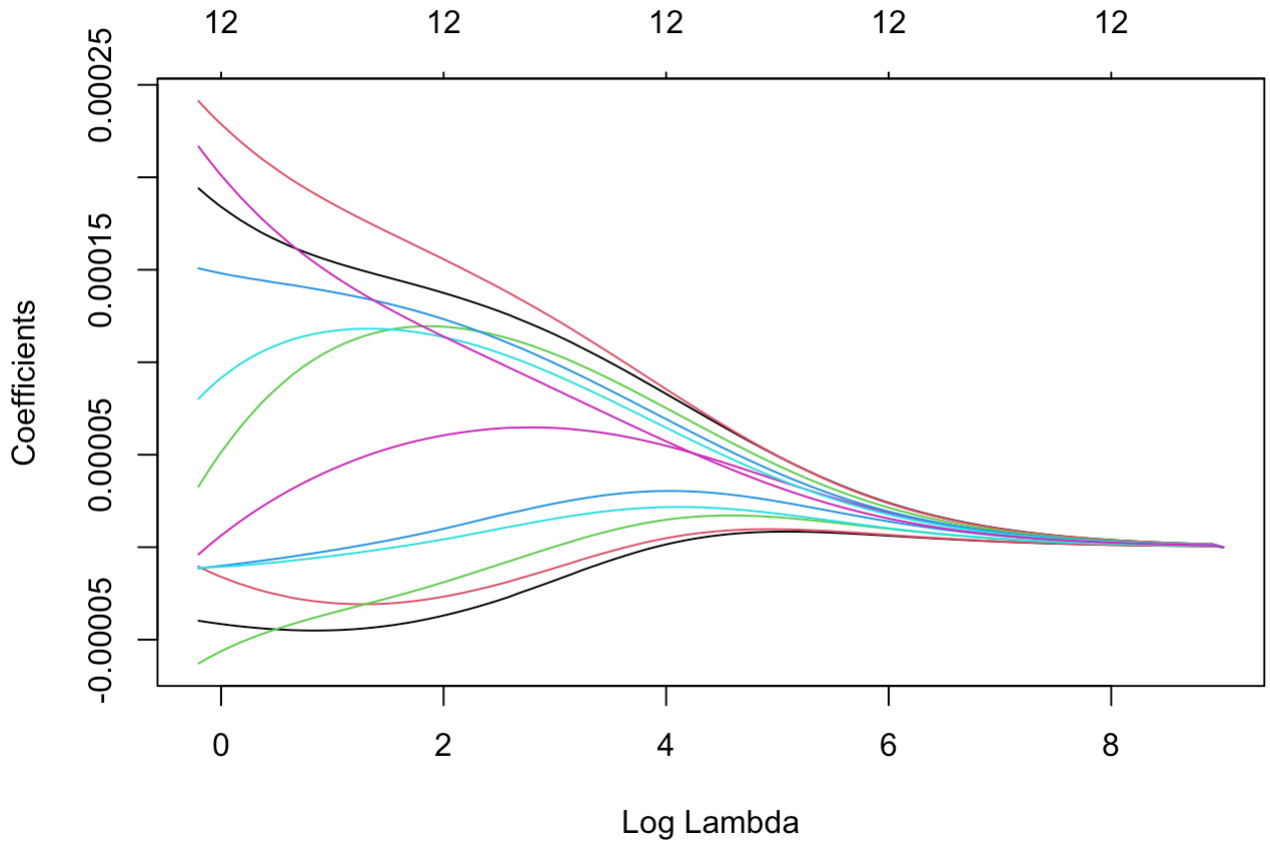
The principal Component Regression with two principal component yielded better accuracy than using other combination of principal component vectors

The Root mean square error using Principal component regression model is 8.8094987

Ridge Regression

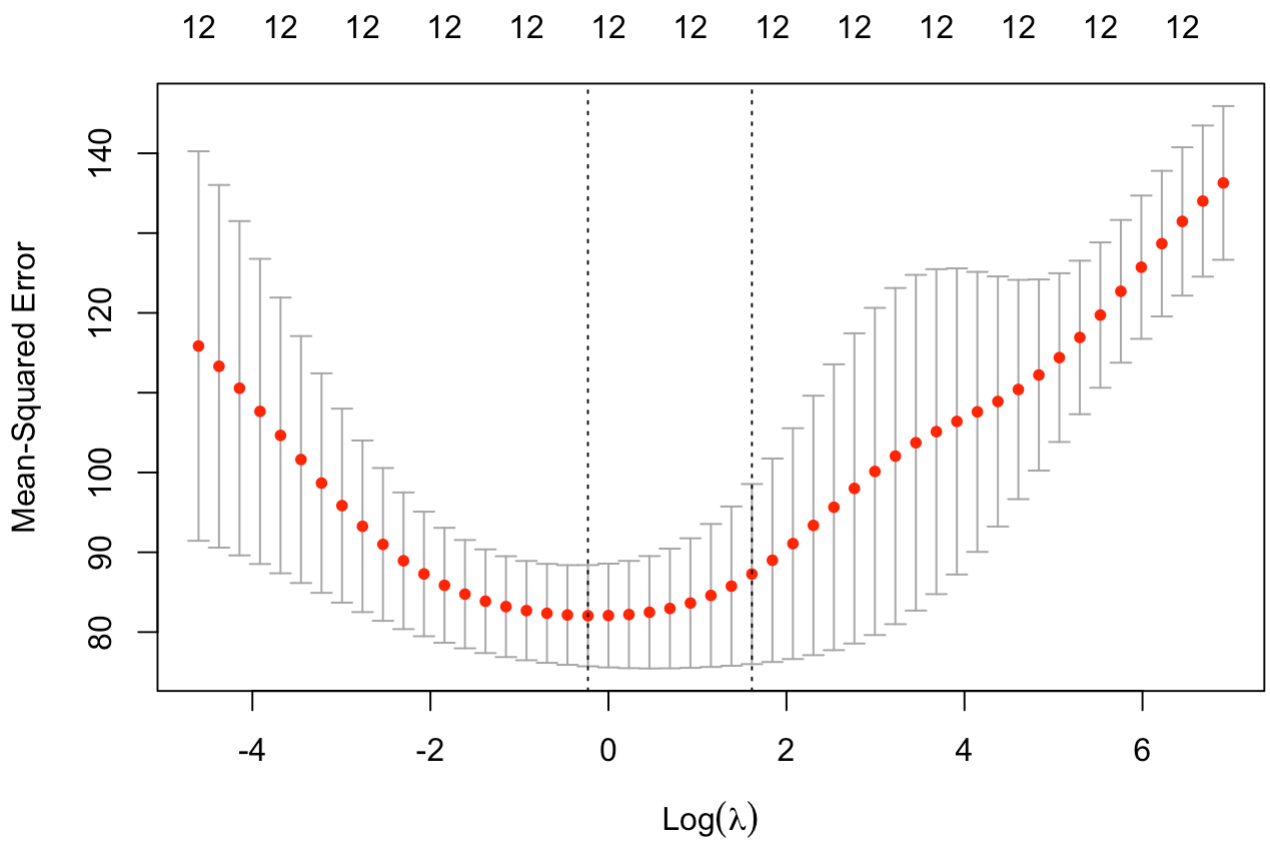
```
library(glmnet)

ridge_mod = glmnet(merg_data[, 1:length(merg_data)-1], merg_data[, length(merg_dat
a)], alpha = 0)
plot(ridge_mod, xvar = "lambda")
```



The increase in lambda values leads to constrain the attribute values nearly equal to zero

```
lambdas <- 10^seq(3,-2,by=-0.1)
cv_fit <- cv.glmnet(as.matrix(merg_data[, 1:length(merg_data)-1]), as.matrix(merg_dat
a[, length(merg_data)]), alpha = 0, lambda = lambdas)
plot(cv_fit)
```




```
min_lambda = cv_fit$lambda.min
```

The lambda value 0.7943282 yields minimum mean squared error in Ridge regression model

```
ridge_mod = glmnet(merg_data[train_index, 1:length(merg_data)-1], merg_data[train_ind
ex, length(merg_data)], alpha = 0, lambda = 0.794)
```

```
test_data = data.frame(x = test)
test_pred = predict(ridge_mod, as.matrix(merg_data[-train_index, 1:length(merg_data)-
1]))
errors <- test_pred- merg_data[-train_index, length(merg_data)]
sq_error_rid = sqrt(mean((errors)^2))
```

The Root mean square error using Ridge regression model is 8.8802142

The Ridge regression RMSE score is similar to Principal component regression model.

The Ridge regression and Principal Component Regression model accuracy is better than linear Regression. The principal component regression model uses only 2 principal component vectors to train the model. Therefore, we select Principal Component model as an optimal model.