Life Expectancy Prediction using Multiple Linear Regression

MA317 Group Coursework

Department of Mathematical Sciences, University of Essex, CO4 3SQ Colchester, United Kingdom

Group 10

Smruti Das - sd20396

Jhansi Rani Choutapalem - jc20168

Venkata Naga Sai Pooja Kommasani - vk20242

Mamatha Sai Yarabarla - my20474

Raja Sumanth Dulam - rd20618

Kotla Madhav Srinivas -mk20955

Abstract

Life expectancy is one of the most important factors and helps to anticipate the procurement of health care services, pensions, and other facilities. It is also crucial to determine the course of treatment and facilitates Advance Care Planning. Physicians tend to overestimate life expectancy and miss the window of opportunity to initiate Advance Care Planning [1]. Hence, it is much important to determine the correct life expectancy of people. This research tests the potential of using machine learning techniques for predicting life expectancy. We approached the task of predicting life expectancy as a supervised machine learning task. We trained and tested a multiple linear regression model on the dataset provided. We developed the model with the help of the backward selection method and evaluated its performance on a test dataset. We performed ANOVA testing by bucketing the countries into the continent.

Keywords: Life expectancy prediction, R programming, Linear Regression, Anova Testing

Contents

Word Count = 2890

1. Introduction

Life expectancy is an important factor to decide the status of the countries. More developed countries have higher life expectancy than the under-developed ones due to lack of many facilities and other factors. Adequate research and modelling will help us to understand factors that affect human longevity and ways to improve quality of life. Whilst the calculation of life expectancy is a complicated process and requires many variables and circumstances to consider. For this project work, The Dataset which was provided to us had life expectancy values for all the countries in a year. It has 21 independent variables which will be used to determine life expectancy.

1.1. Exploratory data analysis

We started with the analysis of the dataset to understand the column values. Exploratory data analysis is a much-needed step to be performed before we start treating the data and preparing the model.

1.2. Imputing Missing Values

Missing Values reduces the statistical power of the data. Secondly, the lost data can cause bias in the estimation of parameters.

1.3. Collinearity

Collinearity increases the variance of the estimators. hence, reduces the adequacy of the Model. Hence, we performed the co-linearity check.

1.4. Modelling

Once the dataset is pre-processed, we use it to train linear regression models to make the prediction. Various techniques like forward/backward selection are used to select the best model.

1.5. Prediction

Once the model is trained, we use the model to make a prediction for the test or any unseen data.

1.6. One Way ANOVA test

We Use Anova Model for conducting statistical hypothesis testing.

2. Exploratory Data Analysis

Here we are going to use data derived from World Bank database. We main aim is to predict the Life expectancy in the year 2018. To do that here we have two datasets "LifeExpectancyData1.csv" and "LifeExpectancyData2.csv" which should be used to make predictions.

2.1 Analysis

Firstly, we must work on the first dataset i.e., LifeExpectancyData1.csv. To work on the dataset, we need to load the data and it can be done using read.csv () function. Using the inbuilt function dim() we can see the dimensions of the dataset i.e., 232 observations and 23 variables. We can see the structure of the data using str(). After that as the column names seems a bit confusing, we are renaming them with the help of rename function.

Here we are going to use the function summary () to display the statistical or numerical summary of the dataset. To summarize the numerical data, we are going to present them using plots. There are many types of plots which can be used to display the relation between the variables in the dataset.

Here we are going visualize Life expectancy by GDP per capita, population and Continent and Employment ratio vs Unemployment by population

Fig 1. Life expectancy Scatter Plots

Here we are visualizing how unemployment stayed the same even after increase in literacy rate.

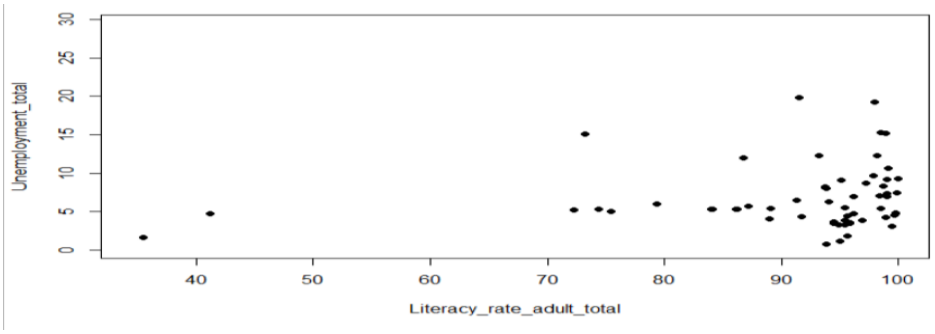


Fig 2. Literacy vs Unemployment

Here in the below boxplot, we can visualize the Mortality rate of male adults by continent and analyse that Sub – Saharan Africa has the highest.

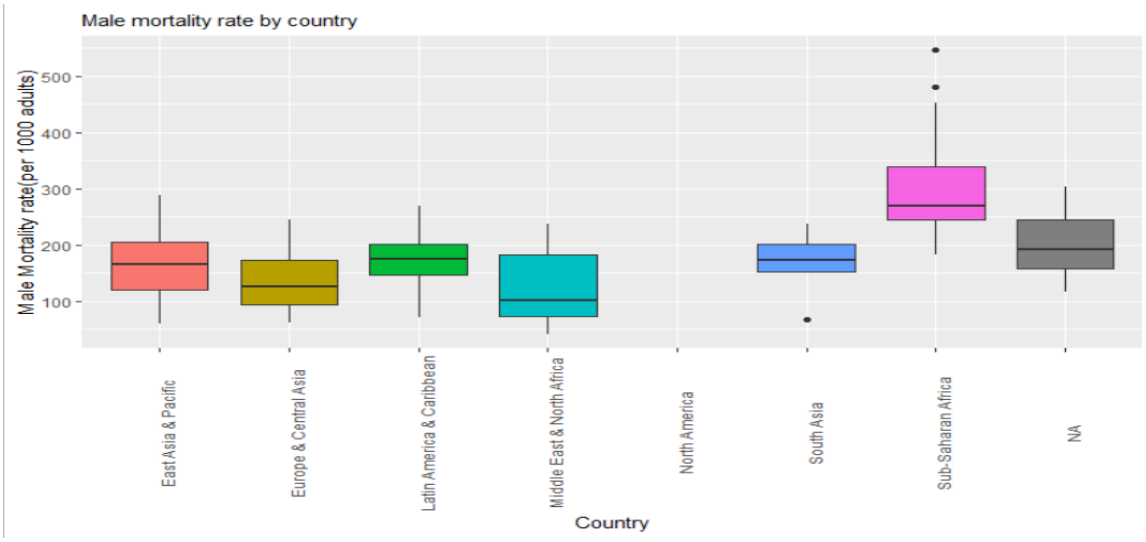


Fig 3. Male Mortality rate by country

3. Missing Value imputation

We check for the missing values using sapply() function and is.na(). Then display the number of missing values in each column using colSums(). To deal with the missing values in a dataset there are many types of imputation methods such as Mean imputation, Median imputation, mode imputation and Multiple imputations method.

Mean imputation: Mean imputation used by taking the average of the variables and replace the missing values. Mean imputation does not work when the data has any extreme value.

Median imputation: where the data will be sorted and the middle values which is an average value impute in the missing values.

Mode imputation: The most repeating values from the dataset is planning to impute the missing values.

Multiple Imputation: We know that there are few things which might not be advantageous to us while imputing the missing values using the above methods such as output being biased. In Multiple imputation the distribution replaces the missing values by predicting the possible values based on other observations. This can be done using a package named "mice" which means Multiple Imputation with Chained Equations. We can start imputing using mice () function and then do a complete case analysis to iterate the imputing values and adjust according to the data. Using only complete case analysis with the above methods may not be good as the output predicted will most likely be biased but using it with multiple imputation method adds strength to the dataset after replacing the missing values. The mice function usually uses the method "pmm" to calculate the missing values.

We can do any imputation mentioned above but while trying to use the multiple imputation method we are seeing null values which will affect our data, so we are using mean and median imputations. We can directly use mean imputation but for few variables there are way too many outliers found while using mean imputation, so we are using median imputation for those variables.

4. Collinearity

Linear combination is nothing but using one regressor by multiplying the other regressors by constants and adding the results. When one regressor is highly correlated with another regressor (or) When one regressor is highly correlated with a linear combination of other regressor, it is called as collinearity. To calculate collinearity, we need to install "corrplot" package and load it. To find the collinearity between the variables, we check the correlation in a pairwise manner and to visualize it we can plot using corrplot.mixed () function which calculates the correlation value between each variable and display it in the map.

Each regressor in the data tries to tell a story about the dependent variable using p-value, effect size etc., which reflects the overlap of the story they tell. As we know that there is no perfect solution for collinearity but for high collinearity, we can consider removing the regressors with high variance inflation factor which can be found using vif (). To find the variance inflation factor values, we are building a full model with all the variables which are having numerical

values and then we are using vif (full_model) which displays the vif value of each variable. Now we can remove the variables with high vif value while building our selected model.

5. Modelling

There are 21 independent features that determine the life expectancy value. Hence, we would be using multiple linear regression for the prediction. Multiple linear regression is an extension of simple linear regression used to predict an outcome variable (y) based on multiple distinct predictor variables. For Instance, with n number predictor variables (x), the prediction of y is expressed by the equation (1)

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + \ldots + b_n * x_n \qquad (1)$$

In R, We can perform linear regression using lm(). Initially we will start with full model including all the predictor variables. The coefficients show how much we can expect life expectancy to increase if the predictor variables increase. The intercept is what life expectancy would be if all the coefficients of the predictor variables were 0. Some of the variables which have negative coefficients of estimates indicate that these variables are inversely related to life expectancy. As per Adjusted R2 value, approximately 0.926 of the observed variation can be explained by the model's inputs, we will try to improve the results by using various feature selection.

5.1 Feature selection

When we're building a machine learning model, it is very important that we select only those features or predictors which are necessary. Firstly, we need to ensure that our model is simple. Secondly, including insignificant variables can hamper model performance.

5.1.1 Backward Feature Selection

Stepwise regression is a way of selecting important variables to get a simple model. We must select a significance level or select the P-value. All the features having p values less than this significance level are retained to the model. In Backward selection we start with all the variables in the model and then keep on deleting the worst features one by one. We have used wrapper function step AIC in MASS library to perform backward selection and produce a final model which has only 8 important features which has better adjusted R-square value as 0.928 than the full mode.

5.1.1 Mallow's Cp

Mallow's C_p is a technique for model selection in regression. Model with different numbers of parameters are compared based in C_p value. C_p considers ratio of SSE for p – 1 variable model to MSE for full model; then penalizes for the number of variables as shown in equation (\ref{eq:function2}), where n is the number of observations and p is the number of parameters. A model having $C_p < p$ is considered as good model. Choosing the smallest model for which this is true is the best method.

$$C_p = SSE_p/(MSE_{fullmodel}) - (n - 2p) \qquad (2)$$

In the figure below, we have plotted C_p value based on number of parameters which is calculated based on wrapper function leaps. It performs an exhaustive search for the best model. The Best model generated by Mallow's Cp technique is same as the one selected by backward selection method.

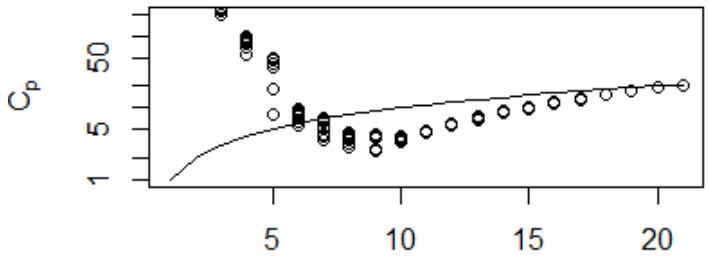


Fig 4. C_p vs Number of Features

Figure shown is appendix .1 shows the plot for standardised residuals against fitted values and qq-plot for selected model. There is not much difference in the model which is justified with the adjusted r-square value which is nearly same.

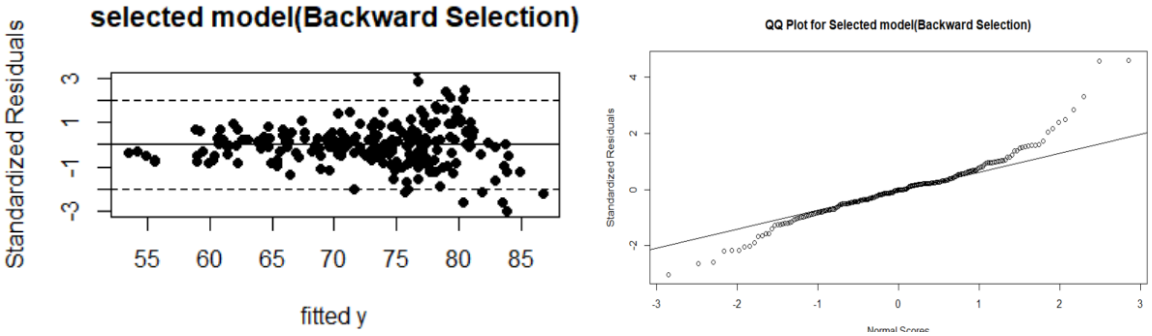


Fig 5. Residual & QQ plot of Selected Model

5.2 Model Evaluation

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and tells us how well the model will perform on test dataset. There are Many methods used to evaluate a model and select the best one.

5.2.1. R^2 Coefficient

R^2 is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of multiple determination for multiple regression. R^2 explains how much variation in dependent variable is explained by the independent variables. we can calculate value by using below equation. where RSS is residual sum of squares which is calculated by summation of square of (actual Y - predicted Y). TSS (total sum of squares) is calculated by summation of square of (actual Y - average Y).

$$R^2 = 1 - (RSS/TSS) \tag{3}$$

It ranges between 0 and 1. The higher the R^2 value better is the model

- 0 indicates that model explains none of the variability of the response data around its mean.
- 1 indicates that the model explains all the variability of the response data around its mean.

5.2.2. AIC Score

AIC lets us test how well the model fits the dataset without over-fitting it. AIC score rewards models that achieve a high goodness-of-fit score and penalizes them if they become overly complex. It is used to compare two models. It estimates models relatively, meaning that AIC scores are only useful in comparison with other AIC scores for the same dataset. The desired result is to find the lowest possible AIC, which indicates the best balance of model fit with generalizability.

AIC uses a model's maximum likelihood estimation (log-likelihood) as a measure of fit as shown in equation below. Log-likelihood is a measure of how likely one is to see their observed data in model. The model with the maximum likelihood is the one that fits the data the best. It adds a penalty term for models with higher parameter complexity, since more parameters means a model is more likely to overfit to the training data.

$$AIC = 2\ln(L) - 2p \tag{4}$$

5.2.2. Mallows' C_p

Mallows' Cp statistic estimates the bias that is introduced into the predicted responses by having a simple model. It helps you strike an important balance with the number of predictors in the model. Mallows' Cp compares the precision and bias of the full model to models with a subset of the predictors. A small Mallows' Cp value indicates that the model is relatively precise (has small variance) in estimating the true regression coefficients and predicting future responses.

	Adjusted R2 Value	AIC	Mallows' CP
Full Model	0.9259	989.728	
Selected Model_vic	0.9059	988.84	12.6 (19 Parameters)
Selected_Model	0.9256	971.70	2.502095 (8 Parameters)

Table1.1 Model evaluation

From the table above we can see that the **Selected_Model** performs better in all the evaluation methods. The Adjusted R2 Value value is highest, AIC score is lowest and mallow's cp value comparison is less than the number of parameters. Hence, we would be using **Selected_Model** to make predictions.

6. Prediction

The trained model is used to make prediction on a test dataset provided to us. Before we can make a prediction, we need to pre-process the test dataset as done for the training dataset.

Below are the steps taken for making prediction:

1. Load the test data into dataframe
2. Rename the columns as done in training set
3. Drop the columns Country Name, Country Code
4. Impute the missing values based on the training values : It is very important to impute the values based on training set not on the test set as that might impute some values which might not fit the model well and would give incorrect predictions
5. Make prediction: we have used predict function on **Selected_Model** to make prediction for test set
6. Export the prediction into a csv file.

7. One Way ANOVA

One-way ANOVA examines at least two different groups to determine if there is a evidence suggesting that the relevant population means are significantly different. One Factor ANOVA is a parametric test. For conducting One-way ANOVA, we require one independent variable and a dependent variable. The most common usage of one-way ANOVA is to test the differences of means among two or more groups. In this project the purpose of one-way

ANOVA is to obtain the statistical differences among the means of life expectancies across continents.

One-way ANOVA is the best method to obtain the mean differences of the life expectancies across continents. Benefits of this method are it can control overall Type 1 error rate and provide overall test of equality of group means.

Type1 error means rejection of true null hypothesis. (for statistical hypothesis testing)

Null Hypothesis: There is no significance difference between the means of Life Expectancies.

Alternative Hypothesis: There is a Significance difference between the means of Life Expectancies.

Results of One-way ANOVA test:

The below image shows the summary of the One-way ANOVA test. From the test we can reject null hypothesis and accept alternative hypothesis at 5% level of significance.

```
                     Df Sum Sq Mean Sq F value Pr(>F)
as.factor(continent)   4   5845  1461.3   61.57 <2e-16 ***
Residuals            180   4272    23.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
47 observations deleted due to missingness
```

Post-hoc Tests to calculate mean differences of life expectancies across continents:

The below graph shows the differences of average life expectancies across continents.There is a significant difference in life expectancies from the other continents in the continent groups Africa- Americas, Africa-Asia, Africa-Europe, Africa-Oceania, Americas-Europe, Asia-Europe and Europe Oceania.

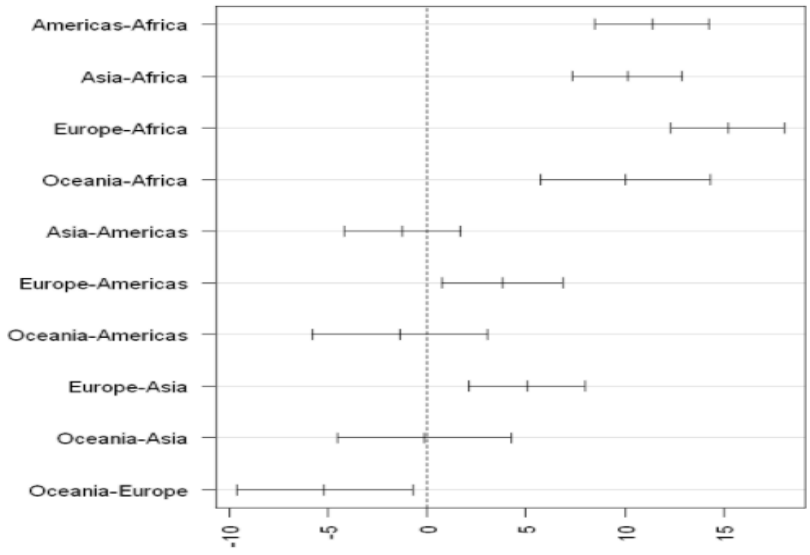


Fig 6. 95% family-wise confidence level

8. Conclusion

The linear model seems to fit perfectly for predicting the life expectancy based on selected independent variables. The Adj. R-Squared value of our final model was 0.9289, which shows that the model built using multiple linear regression is a good model for accurately predicting life expectancy. However, proper data pre-processing such as missing value imputation, collinearity check, should be done before training a model to get best results.

9. References

1. A. A. Bhosale and K. K. Sundaram, "Life prediction equation for human beings," *2010 International Conference on Bioinformatics and Biomedical Technology*, Chengdu, China, 2010, pp. 266-268, doi: 10.1109/ICBBT.2010.5478965.
2. J. J. Kang, "Prediction of Personalised Life Expectancy using Personal Health Devices in mHealth Networks," *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, Australia, 2018, pp. 1-5, doi: 10.1109/ATNAC.2018.8615428.
3. Medium. 2021. *The Akaike Information Criterion*. [online] Available at: https://towardsdatascience.com/the-akaike-information-criterion-c20c8fd832f2
4. Editor, M., 2021. *Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?*. [online] Blog.minitab.com. Available at: https://blog.minitab.com/en/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit
5. Libguides.library.kent.edu. 2021. *LibGuides: SPSS Tutorials: One-Way ANOVA*. [online] Available at: https://libguides.library.kent.edu/spss/onewayanova

10. Individual contributions

Name	Part done as per question	Individual Contribution	Additional Contribution
Smruti Das	**4 (b) & 4 (c)**	Code, Report, and presentation Part for 4 (b) & 4 (c)	Have collated whole report, Have collated the presentation, Written Abstract and conclusion, Worked on re-naming the columns
Jhansi Rani Choutapalem	**4 (a)**	Code, Report, and presentation Part for 4 (a)	Have worked on some Data Analysis, Have written the introduction part, Have worked on calculating VIF in collinearity.
Venkata Naga Sai Pooja Kommasani	**5**	Code, Report, and presentation Part for 5	Have Done Missing value imputation, have worked in collated the presentation.
Mamatha Sai Yarabarla	**1**	Code and Report, and presentation Part for 1	
Raja Sumanth Dulam	**2**	Code and Report, and presentation Part for 2	

Kotla Madhav Srinivas	**3**	Code and Report, and presentation Part for 3	Have built a full model to help calculate the VIF values for variables, so that variables with high VIF and we removed for building selected model.

11. Appendix

11.1 R Code

```
#Loading libraries
library(ggplot2)
library(countrycode)
library(dplyr)
library(car)
library(corrplot)
```

```
> #Loading libraries
> library(ggplot2)
> library(countrycode)
> library(dplyr)
> library(car)
> library(corrplot)
>
```

```
#loading the data
df <- read.csv("D:\\New folder\\Modelling Experimental data\\LifeExpectancyData1.csv",
header = T)

#numerical summary statistics
summary(df)
```

```
> #loading the data
> df <- read.csv("D:\\New folder\\Modelling Experimental data\\LifeExpectancyData1.csv", header = T)
> #numerical summary statistics
> summary(df)
 Country.Name       Country.Code       SP.DYN.LE00.IN  EG.ELC.ACCS.ZS   NY.ADJ.NNTY.KD.ZG
 Length:232         Length:232         Min.   :52.80   Min.   : 11.02   Min.   :-28.100
 Class :character   Class :character   1st Qu.:67.91   1st Qu.: 80.58   1st Qu.:  1.016
 Mode  :character   Mode  :character   Median :73.60   Median : 99.92   Median :  2.741
                                       Mean   :72.59   Mean   : 85.43   Mean   :  2.777
                                       3rd Qu.:77.53   3rd Qu.:100.00   3rd Qu.:  5.160
                                       Max.   :84.93   Max.   :100.00   Max.   : 22.083
                                       NA's   :1                        NA's   :59
 NY.ADJ.NNTY.KD      SE.PRM.UNER.ZS     SE.XPD.PRIM.ZS    SP.DYN.IMRT.IN  SE.ADT.LITR.ZS   SP.POP.GROW
 Min.   :4.011e+08   Min.   : 0.00047   Min.   : 0.6578   Min.   : 1.60   Min.   :34.52   Min.   :-4.0484
 1st Qu.:1.801e+10   1st Qu.: 0.76587   1st Qu.:29.2652   1st Qu.: 6.20   1st Qu.:74.14   1st Qu.: 0.4727
 Median :1.735e+11   Median : 2.84067   Median :34.0356   Median :14.80   Median :91.29   Median : 1.1598
 Mean   :3.308e+12   Mean   : 6.03466   Mean   :34.7588   Mean   :22.23   Mean   :83.34   Mean   : 1.2379
 3rd Qu.:1.290e+12   3rd Qu.: 7.83525   3rd Qu.:39.0394   3rd Qu.:34.49   3rd Qu.:96.10   3rd Qu.: 2.0306
 Max.   :6.770e+13   Max.   :47.34818   Max.   :61.6475   Max.   :83.40   Max.   :99.99   Max.   : 4.9212
 NA's   :66          NA's   :74         NA's   :209       NA's   :12      NA's   :121     NA's   :1
  SP.POP.TOTL        SE.PRM.CMPT.ZS   SH.XPD.CHEX.GD.ZS SH.XPD.CHEX.PC.CD  SL.UEM.TOTL.NE.ZS
 Min.   :3.791e+04   Min.   : 40.56   Min.   : 2.138    Min.   :   18.51   Min.   : 0.110
 1st Qu.:2.901e+06   1st Qu.: 88.00   1st Qu.: 4.474    1st Qu.:   85.72   1st Qu.: 4.000
 Median :1.149e+07   Median : 95.79   Median : 5.993    Median :  350.07   Median : 5.327
 Mean   :3.485e+08   Mean   : 92.12   Mean   : 6.480    Mean   : 1134.97   Mean   : 7.019
 3rd Qu.:8.349e+07   3rd Qu.: 99.90   3rd Qu.: 7.952    3rd Qu.: 1024.33   3rd Qu.: 8.385
 Max.   :7.592e+09   Max.   :123.00   Max.   :16.885    Max.   :10623.85   Max.   :29.420
 NA's   :1           NA's   :75       NA's   :16        NA's   :16         NA's   :100
 SP.DYN.AMRT.FE   SP.DYN.AMRT.MA  NY.GDP.MKTP.KD.ZG NY.GDP.PCAP.PP.CD  SP.DYN.CBRT.IN  NY.GNP.PCAP.PP.CD
 Min.   : 31.87   Min.   : 41.5   Min.   :-6.356    Min.   :   780.1   Min.   : 6.40   Min.   :  780
 1st Qu.: 77.44   1st Qu.:139.2   1st Qu.: 1.941    1st Qu.:  5280.6   1st Qu.:11.64   1st Qu.: 5140
 Median :115.17   Median :188.4   Median : 3.116    Median : 14208.1   Median :17.70   Median :13960
 Mean   :134.05   Mean   :197.2   Mean   : 3.195    Mean   : 21259.2   Mean   :19.88   Mean   :20617
 3rd Qu.:185.02   3rd Qu.:246.3   3rd Qu.: 4.475    3rd Qu.: 30174.1   3rd Qu.:26.88   3rd Qu.:29020
 Max.   :419.36   Max.   :545.7   Max.   :15.133    Max.   :120325.9   Max.   :46.08   Max.   :91280
 NA's   :45       NA's   :45      NA's   :11        NA's   :15                         NA's   :15
 SL.EMP.TOTL.SP.ZS
 Min.   :33.17
 1st Qu.:52.32
 Median :58.42
 Mean   :58.03
 3rd Qu.:63.69
 Max.   :86.61
 NA's   :11
>
```

```
#renaming the column names of dataset
df <- df %>%
  rename('life_expectancy_at_birth'='SP.DYN.LE00.IN',
         'per_pop_access_to_electricity'='EG.ELC.ACCS.ZS',
         'per_annual_growth_national_income'='NY.ADJ.NNTY.KD.ZG',
         'net_national_income'='NY.ADJ.NNTY.KD',
         'per_children_out_of_school'='SE.PRM.UNER.ZS',
         'per_expenditure_primary_education'='SE.XPD.PRIM.ZS',
         'infant_mortality_rate_per_1000'='SP.DYN.IMRT.IN',
         'per_adult_literacy_rate'='SE.ADT.LITR.ZS',
         'per_annual_population_growth'='SP.POP.GROW',
         'total_population'='SP.POP.TOTL',
         'per_primary_completion'='SE.PRM.CMPT.ZS',
         'per_gdp_health_expenditure'='SH.XPD.CHEX.GD.ZS',
         'health_expenditure_per_capita'='SH.XPD.CHEX.PC.CD',
         'per_unemployment'='SL.UEM.TOTL.NE.ZS',
         'per_adult_female_mortality_rate_per_1000'='SP.DYN.AMRT.FE',
         'per_adult_male_mortality_rate_per_1000'='SP.DYN.AMRT.MA',
         'per_annual_gdp_growth'='NY.GDP.MKTP.KD.ZG',
         'gdp_per_capita'='NY.GDP.PCAP.PP.CD',
         'crude_birth_rate_per_1000'='SP.DYN.CBRT.IN',
         'gni_per_capita'='NY.GNP.PCAP.PP.CD',
         'employment_to_population_ratio'='SL.EMP.TOTL.SP.ZS'
          )
```

```
> #renaming the column names of dataset
> df <- df %>%
+   rename('life_expectancy_at_birth'='SP.DYN.LE00.IN',
+          'per_pop_access_to_electricity'='EG.ELC.ACCS.ZS',
+          'per_annual_growth_national_income'='NY.ADJ.NNTY.KD.ZG',
+          'net_national_income'='NY.ADJ.NNTY.KD',
+          'per_children_out_of_school'='SE.PRM.UNER.ZS',
+          'per_expenditure_primary_education'='SE.XPD.PRIM.ZS',
+          'infant_mortality_rate_per_1000'='SP.DYN.IMRT.IN',
+          'per_adult_literacy_rate'='SE.ADT.LITR.ZS',
+          'per_annual_population_growth'='SP.POP.GROW',
+          'total_population'='SP.POP.TOTL',
+          'per_primary_completion'='SE.PRM.CMPT.ZS',
+          'per_gdp_health_expenditure'='SH.XPD.CHEX.GD.ZS',
+          'health_expenditure_per_capita'='SH.XPD.CHEX.PC.CD',
+          'per_unemployment'='SL.UEM.TOTL.NE.ZS',
+          'per_adult_female_mortality_rate_per_1000'='SP.DYN.AMRT.FE',
+          'per_adult_male_mortality_rate_per_1000'='SP.DYN.AMRT.MA',
+          'per_annual_gdp_growth'='NY.GDP.MKTP.KD.ZG',
+          'gdp_per_capita'='NY.GDP.PCAP.PP.CD',
+          'crude_birth_rate_per_1000'='SP.DYN.CBRT.IN',
+          'gni_per_capita'='NY.GNP.PCAP.PP.CD',
+          'employment_to_population_ratio'='SL.EMP.TOTL.SP.ZS'
+
+   )
>
```

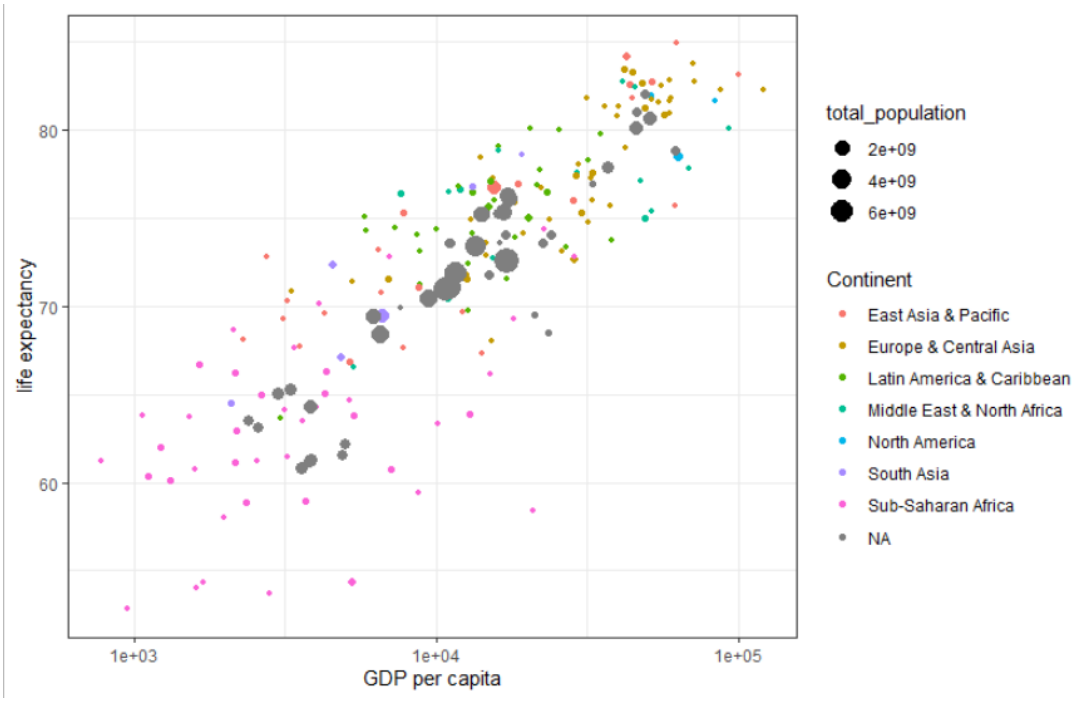
#Creating a column with name continent and categorizing the country according to continent using countrycode()

```
df$Continent = countrycode(sourcevar = df[, "Country.Name"],
                           origin = "country.name",
                           destination = "region")
```

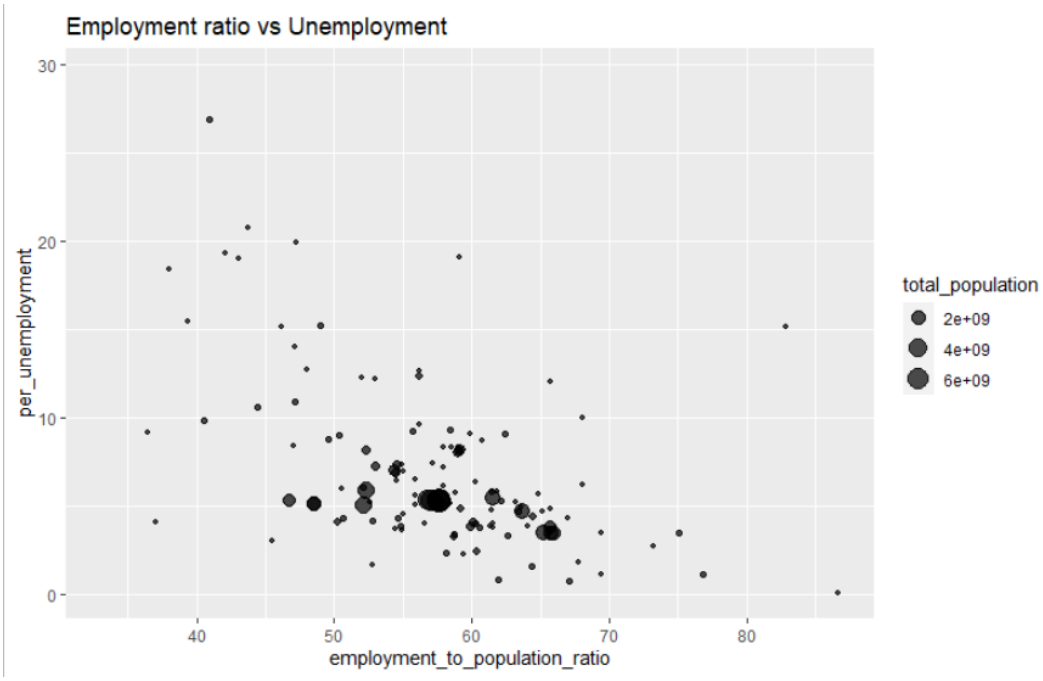
```
> #Creating a column with name continent and categorizing the country according to continent using countrycode()
> df$Continent = countrycode(sourcevar = df[, "Country.Name"],
+                            origin = "country.name",
+                            destination = "region")
```

#Plotting GDP per capita vs Life Expectancy with population by Continents

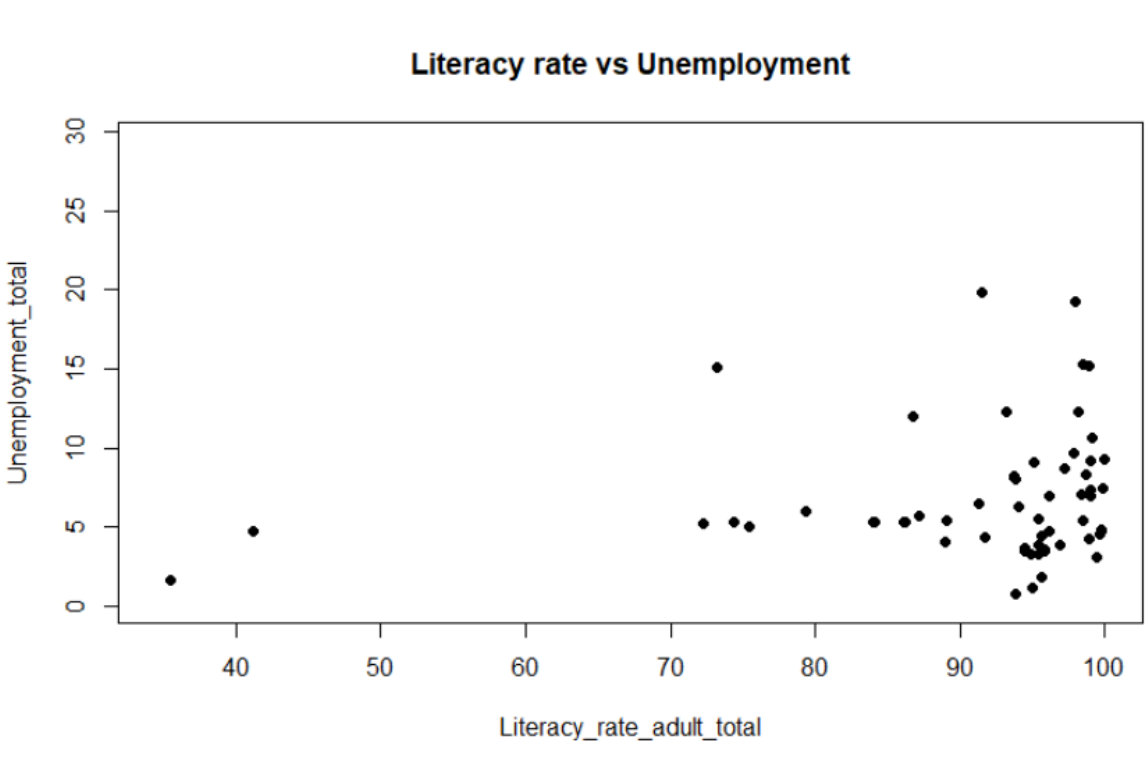
```
ggplot(df, aes(x = gdp_per_capita, y = life_expectancy_at_birth, size = total_population,
color = Continent)) +
  geom_point() +
  scale_x_log10() +
  theme_bw() +labs( x = 'GDP per capita', y = 'life expectancy')
```



#Plotting Employment to population ratio vs Unemployment
ggplot(df, aes(x=employment_to_population_ratio, y=per_unemployment, size = total_population)) +
geom_point(alpha=0.7) + labs(title = "Employment ratio vs Unemployment", xlab = "Employment Ratio", ylab = "Unemployment")

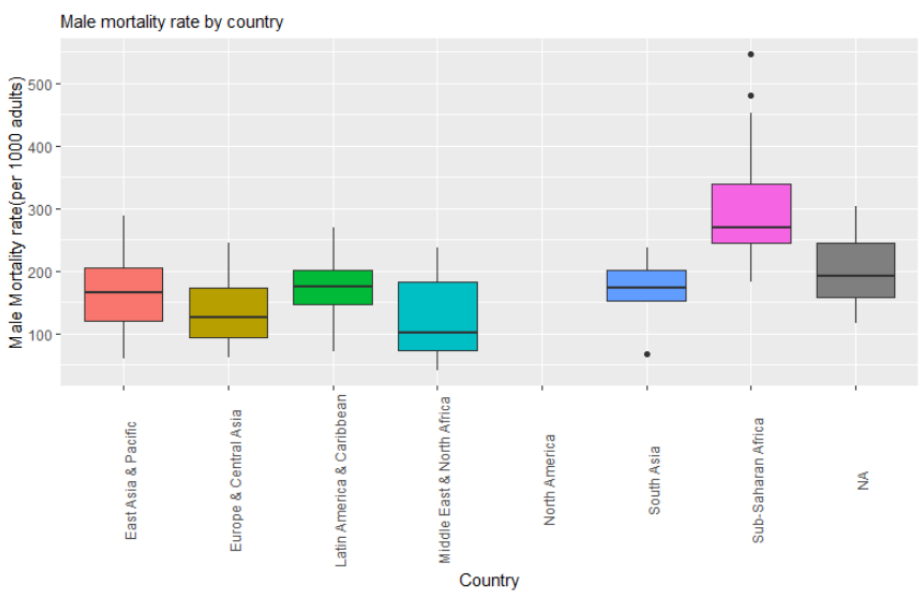


#Plotting Literacy rate vs Unemployment
Literacy_rate_adult_total = df[,10]
Unemployment_total = df[,16]
plot(Literacy_rate_adult_total, Unemployment_total, pch=16, main = "Literacy rate vs Unemployment")



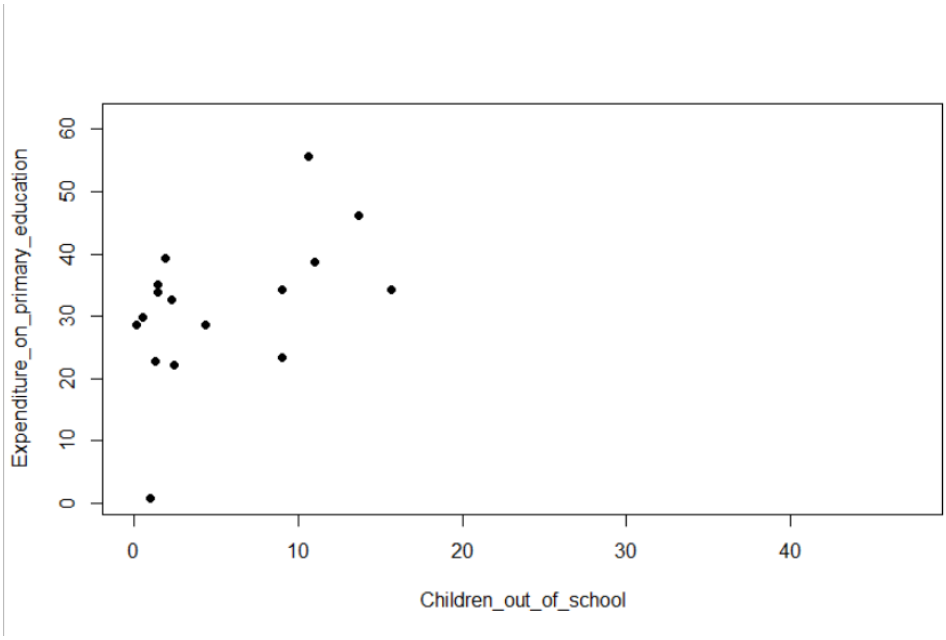
#Plotting Male Mortatlity rate according to continent

```
ggplot(df, aes(x=Continent, y=per_adult_male_mortality_rate_per_1000, fill = Continent))
+
  geom_boxplot() +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("Male mortality rate by country") +
  xlab("Country") + ylab("Male Mortality rate(per 1000 adults)") +   theme(axis.text.x =
element_text(angle = 90))
```



#Plotting Children out of school vs Expenditure on Primary Educaation

Children_out_of_school = df[,7]
Expenditure_on_primary_education = df[,8]
plot(Children_out_of_school, Expenditure_on_primary_education, pch=16)



#Removing the columns not needed for training
df <- subset(df,select=-c(Country.Name,Country.Code,Continent))
#checking for missing values
colSums(sapply(df, is.na))

```
> #checking for missing values
> colSums(sapply(df, is.na))
                life_expectancy_at_birth          per_pop_access_to_electricity
                                       1                                      0
       per_annual_growth_national_income                    net_national_income
                                      59                                     66
             per_children_out_of_school       per_expenditure_primary_education
                                      74                                    209
          infant_mortality_rate_per_1000                per_adult_literacy_rate
                                      12                                    121
            per_annual_population_growth                      total_population
                                       1                                      1
                  per_primary_completion           per_gdp_health_expenditure
                                      75                                     16
           health_expenditure_per_capita                       per_unemployment
                                      16                                    100
per_adult_female_mortality_rate_per_1000   per_adult_male_mortality_rate_per_1000
                                      45                                     45
                  per_annual_gdp_growth                         gdp_per_capita
                                      11                                     15
              crude_birth_rate_per_1000                         gni_per_capita
                                       0                                     15
          employment_to_population_ratio
                                      11
> |
```

#Missing value imputation using mean median
df$life_expectancy_at_birth[is.na(df$life_expectancy_at_birth)] <-
mean(df$life_expectancy_at_birth,na.rm = TRUE)
df$per_annual_growth_national_income[is.na(df$per_annual_growth_national_income)] <-
mean(df$per_annual_growth_national_income,na.rm = TRUE)

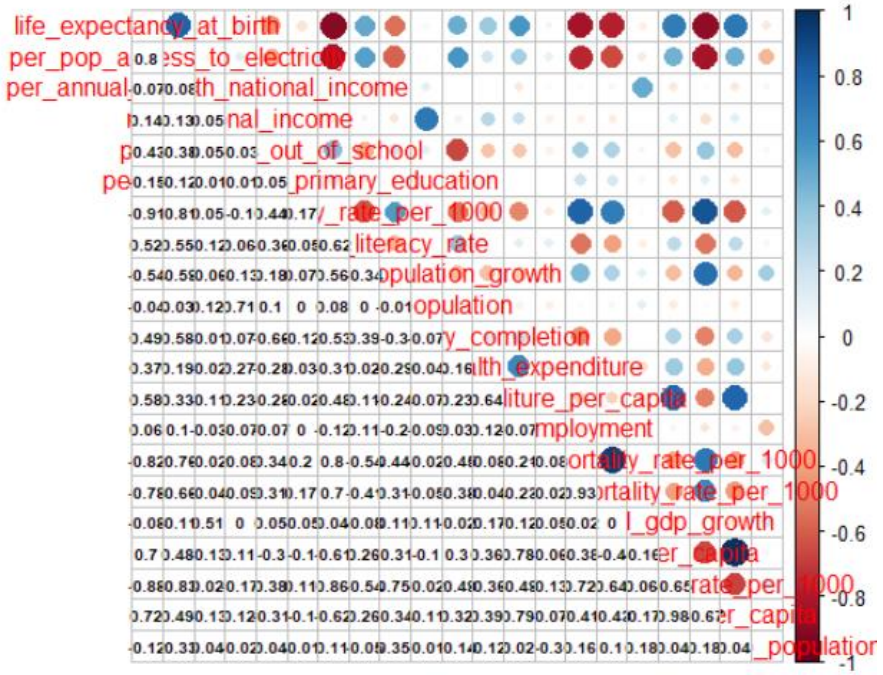
```
df$per_children_out_of_school[is.na(df$per_children_out_of_school)] <-
median(df$per_children_out_of_school,na.rm = TRUE)
df$per_expenditure_primary_education [is.na(df$per_expenditure_primary_education )] <-
mean(df$per_expenditure_primary_education ,na.rm = TRUE)
df$infant_mortality_rate_per_1000[is.na(df$infant_mortality_rate_per_1000)] <-
mean(df$infant_mortality_rate_per_1000,na.rm = TRUE)
df$per_adult_literacy_rate[is.na(df$per_adult_literacy_rate)] <-
mean(df$per_adult_literacy_rate,na.rm = TRUE)
df$per_annual_population_growth[is.na(df$per_annual_population_growth)] <-
median(df$per_annual_population_growth,na.rm = TRUE)
df$total_population[is.na(df$total_population)] <- median(df$total_population,na.rm =
TRUE)
df$per_primary_completion[is.na(df$per_primary_completion)] <-
mean(df$per_primary_completion,na.rm = TRUE)
df$per_gdp_health_expenditure[is.na(df$per_gdp_health_expenditure)] <-
mean(df$per_gdp_health_expenditure,na.rm = TRUE)
df$health_expenditure_per_capita[is.na(df$health_expenditure_per_capita)] <-
mean(df$health_expenditure_per_capita,na.rm = TRUE)
df$per_unemployment[is.na(df$per_unemployment)] <- median(df$per_unemployment,na.rm
= TRUE)
df$per_adult_female_mortality_rate_per_1000[is.na(df$per_adult_female_mortality_rate_p
er_1000)] <- mean(df$per_adult_female_mortality_rate_per_1000,na.rm = TRUE)
df$per_adult_male_mortality_rate_per_1000[is.na(df$per_adult_male_mortality_rate_per_1
000)] <- mean(df$per_adult_male_mortality_rate_per_1000,na.rm = TRUE)
df$per_annual_gdp_growth[is.na(df$per_annual_gdp_growth)] <-
mean(df$per_annual_gdp_growth,na.rm = TRUE)
df$gdp_per_capita[is.na(df$gdp_per_capita)] <- median(df$gdp_per_capita,na.rm = TRUE)
df$gni_per_capita[is.na(df$gni_per_capita)] <- mean(df$gni_per_capita,na.rm = TRUE)
df$net_national_income[is.na(df$net_national_income )] <- mean(df$net_national_income
,na.rm = TRUE)
df$employment_to_population_ratio[is.na(df$employment_to_population_ratio)] <-
mean(df$employment_to_population_ratio,na.rm = TRUE)

# checking if the imputation is done for all the columns
colSums(sapply(df, is.na))
```

```
> # checking if the imputation is done for all the columns
> colSums(sapply(df, is.na))
                life_expectancy_at_birth          per_pop_access_to_electricity
                                       0                                       0
       per_annual_growth_national_income                     net_national_income
                                       0                                       0
             per_children_out_of_school       per_expenditure_primary_education
                                       0                                       0
          infant_mortality_rate_per_1000                per_adult_literacy_rate
                                       0                                       0
           per_annual_population_growth                      total_population
                                       0                                       0
                 per_primary_completion           per_gdp_health_expenditure
                                       0                                       0
          health_expenditure_per_capita                       per_unemployment
                                       0                                       0
per_adult_female_mortality_rate_per_1000   per_adult_male_mortality_rate_per_1000
                                       0                                       0
                 per_annual_gdp_growth                            gdp_per_capita
                                       0                                       0
             crude_birth_rate_per_1000                            gni_per_capita
                                       0                                       0
         employment_to_population_ratio
                                       0
```

#Collinearity

df_new.corr<-cor(df)

corrplot.mixed(df_new.corr, lower.col = "black", number.cex = .6)



#building full model

full_model <- glm(life_expectancy_at_birth ~.,data=df)

#calculating VIF values of variables in current model

vif_values <- vif(full_model)

#displaying vif values

vif_values

```
> vif_values
            per_pop_access_to_electricity        per_annual_growth_national_income
                                 5.858202                                 1.496312
                      net_national_income               per_children_out_of_school
                                 2.653762                                 2.060831
       per_expenditure_primary_education           infant_mortality_rate_per_1000
                                 1.085453                                 7.997653
                  per_adult_literacy_rate             per_annual_population_growth
                                 2.167880                                 4.081017
                           total_population                   per_primary_completion
                                 2.585600                                 2.457072
              per_gdp_health_expenditure            health_expenditure_per_capita
                                 2.753956                                 4.900613
                         per_unemployment per_adult_female_mortality_rate_per_1000
                                 1.230909                                16.918009
  per_adult_male_mortality_rate_per_1000                   per_annual_gdp_growth
                                12.038178                                 1.695009
                            gdp_per_capita               crude_birth_rate_per_1000
                                36.027479                                12.737166
                            gni_per_capita           employment_to_population_ratio
                                39.001321                                 1.597252
> |
```

#Plotting VIF values

barplot(vif_values, main = "VIF Values", horiz = TRUE, col = "steelblue")

abline(v =5, lwd =3, lty =2)

11.1.4. Modelling

 # Load the libraries

library(ggplot2)

library(tidyverse)

library(ggthemes)

library(scales)

library(MASS)

library(olsrr)

library(leaps)

library(mice)

library(car)

library(naniar)

```
> library(ggplot2)
> library(tidyverse)
> library(ggthemes)
> library(scales)
> library(MASS)
> library(olsrr)
> library(leaps)
> library(mice)
> library(car)
> library(naniar)
```

 # Counting the number of columns that consists of numerical data

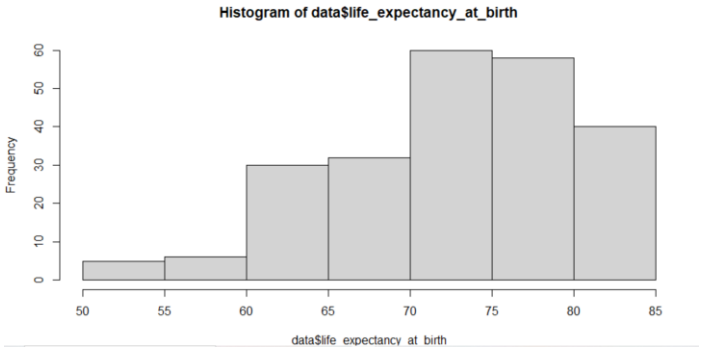
```
num_cols <- unlist(lapply(data, is.numeric))
cat("Total number of numeric columns: ", ncol(data[num_cols]))
```

```
> # Counting the number of columns that consists of numerical data
> num_cols <- unlist(lapply(data, is.numeric))
> cat("Total number of numeric columns: ", ncol(data[num_cols]))
Total number of numeric columns:  21
```

```
#statistics of target variable
summary(data$life_expectancy_at_birth)
```

```
> summary(data$life_expectancy_at_birth)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  52.80   67.91   73.60   72.59   77.53   84.93       1
> 
```

```
#plotting histogram for the target variable
hist(data$life_expectancy_at_birth)
```




```
# looking at some basic statistics
data %>%
    summarize(count = n(),
          avg_life_expectancy_at_birth = mean(life_expectancy_at_birth, na.rm=TRUE),
          avg_national_income = mean(net_national_income, na.rm=TRUE),
          avg_infant_mortality_rate_per_1000 = mean(infant_mortality_rate_per_1000,
na.rm=TRUE),
          avg_gdp_per_capita = mean(gdp_per_capita, na.rm=TRUE))
```

```
> data %>%
+       summarize(count = n(),
+             avg_life_expectancy_at_birth = mean(life_expectancy_at_birth, na.rm=TRUE),
+             avg_national_income = mean(net_national_income, na.rm=TRUE),
+             avg_infant_mortality_rate_per_1000 = mean(infant_mortality_rate_per_1000, na.rm=TRUE),
+            avg_gdp_per_capita = mean(gdp_per_capita, na.rm=TRUE))
  count avg_life_expectancy_at_birth avg_national_income avg_infant_mortality_rate_per_1000 avg_gdp_per_capita
1   232                     72.5895        3.307617e+12                           22.23478            21259.2
```

```
#1. Full Model
full_model <- lm(formula=life_expectancy_at_birth ~.,data=data)
summary(full_model)
```

```
> #1. Full Model
> full_model <- lm(formula=life_expectancy_at_birth ~.,data=data)
> summary(full_model)

Call:
lm(formula = life_expectancy_at_birth ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1208 -0.9854  0.0327  0.8173  8.4936

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                              8.441e+01  3.604e+00  23.420  < 2e-16 ***
per_pop_access_to_electricity            1.262e-02  1.308e-02   0.965 0.335741
per_annual_growth_national_income       -3.875e-02  3.635e-02  -1.066 0.287590
net_national_income                      2.121e-15  2.748e-14   0.077 0.938538
per_children_out_of_school              -2.804e-02  2.682e-02  -1.046 0.296960
per_expenditure_primary_education        1.019e-02  3.400e-02   0.300 0.764791
infant_mortality_rate_per_1000          -1.098e-01  1.928e-02  -5.694 4.12e-08 ***
per_adult_literacy_rate                 -1.341e-02  1.587e-02  -0.845 0.399088
per_annual_population_growth             2.419e-01  2.247e-01   1.077 0.282867
total_population                        -1.225e-10  2.010e-10  -0.610 0.542812
per_primary_completion                  -1.738e-02  1.937e-02  -0.897 0.370712
per_gdp_health_expenditure               9.960e-02  8.234e-02   1.210 0.227734
health_expenditure_per_capita            5.499e-04  1.528e-04   3.599 0.000397 ***
per_unemployment                        -3.113e-02  3.717e-02  -0.837 0.403308
per_adult_female_mortality_rate_per_1000 -1.931e-02  7.675e-03  -2.516 0.012617 *
per_adult_male_mortality_rate_per_1000   -1.453e-02  5.814e-03  -2.500 0.013190 *
per_annual_gdp_growth                   -2.780e-02  6.711e-02  -0.414 0.679143
gdp_per_capita                           3.715e-06  3.752e-05   0.099 0.921218
crude_birth_rate_per_1000               -2.144e-01  4.822e-02  -4.446 1.41e-05 ***
gni_per_capita                           2.163e-05  4.125e-05   0.524 0.600597
employment_to_population_ratio           1.660e-03  1.588e-02   0.105 0.916845
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.948 on 211 degrees of freedom
Multiple R-squared:  0.9323,    Adjusted R-squared:  0.9259
F-statistic: 145.3 on 20 and 211 DF,  p-value: < 2.2e-16
```

#2. Reduced Model using backward selection method
#Perform Backward selection

stepAIC(full_model, direction = "backward", trace = FALSE)

```
> #Perform Backward selection
> stepAIC(full_model, direction = "backward", trace = FALSE)

Call:
lm(formula = life_expectancy_at_birth ~ per_annual_growth_national_income +
    infant_mortality_rate_per_1000 + per_gdp_health_expenditure +
    health_expenditure_per_capita + per_adult_female_mortality_rate_per_1000 +
    per_adult_male_mortality_rate_per_1000 + crude_birth_rate_per_1000 +
    gni_per_capita, data = data)

Coefficients:
                             (Intercept)         per_annual_growth_national_income
                              82.7211588                                -0.0534561
          infant_mortality_rate_per_1000               per_gdp_health_expenditure
                              -0.1138146                                 0.1064425
           health_expenditure_per_capita  per_adult_female_mortality_rate_per_1000
                               0.0005691                                -0.0171925
  per_adult_male_mortality_rate_per_1000                crude_birth_rate_per_1000
                              -0.0176529                                -0.1832587
                        gni_per_capita
                               0.0000311
```

```
# build selected model
selected_model = lm(formula = life_expectancy_at_birth ~
                      per_annual_growth_national_income   +
                      infant_mortality_rate_per_1000 +
                      per_gdp_health_expenditure   +
                      health_expenditure_per_capita   +
                      per_adult_female_mortality_rate_per_1000   +
                      per_adult_male_mortality_rate_per_1000 +
                     crude_birth_rate_per_1000   + gni_per_capita ,
                 data = data)
summary(selected_model)
```

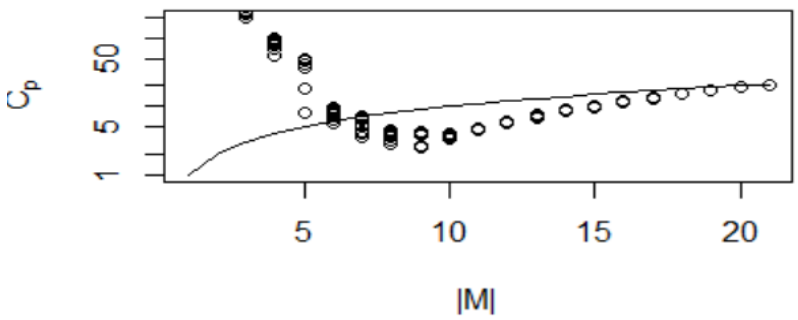
```
> selected_model = lm(formula = life_expectancy_at_birth ~ per_annual_growth_national_income   +
+                       infant_mortality_rate_per_1000 + per_gdp_health_expenditure   +
+                       health_expenditure_per_capita   + per_adult_female_mortality_rate_per_1000   +
+                       per_adult_male_mortality_rate_per_1000 + crude_birth_rate_per_1000   + gni_per_capita   ,
+                     data = data)
>
> summary(selected_model)

Call:
lm(formula = life_expectancy_at_birth ~ per_annual_growth_national_income +
    infant_mortality_rate_per_1000 + per_gdp_health_expenditure +
    health_expenditure_per_capita + per_adult_female_mortality_rate_per_1000 +
    per_adult_male_mortality_rate_per_1000 + crude_birth_rate_per_1000 +
    gni_per_capita, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3307 -0.9537 -0.0305  0.7581  8.7719

Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                          8.272e+01  8.484e-01  97.507  < 2e-16 ***
per_annual_growth_national_income   -5.346e-02  2.985e-02  -1.791 0.074716 .
infant_mortality_rate_per_1000      -1.138e-01  1.683e-02  -6.761 1.19e-10 ***
per_gdp_health_expenditure           1.064e-01  6.968e-02   1.528 0.128025
health_expenditure_per_capita        5.691e-04  1.443e-04   3.945 0.000107 ***
per_adult_female_mortality_rate_per_1000 -1.719e-02  6.962e-03  -2.469 0.014287 *
per_adult_male_mortality_rate_per_1000   -1.765e-02  5.046e-03  -3.498 0.000566 ***
crude_birth_rate_per_1000            -1.833e-01  2.903e-02  -6.313 1.46e-09 ***
gni_per_capita                       3.110e-05  1.374e-05   2.264 0.024531 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.919 on 223 degrees of freedom
Multiple R-squared:  0.9306,    Adjusted R-squared:  0.9281
F-statistic: 373.6 on 8 and 223 DF,  p-value: < 2.2e-16
```

#3. Reduced Model using leaps (Mallows' Cp value)

```
full_model.cp <- lm(formula=life_expectancy_at_birth ~.,data=data,x=TRUE)
X <- full_model.cp$x
y <-data$life_expectancy_at_birth
all.models <- leaps(X, y, int = FALSE, strictly.compatible = FALSE, method="Cp")
```

```
> full_model.cp <- lm(formula=life_expectancy_at_birth ~.,data=data,x=TRUE)
> X <- full_model.cp$x
> y <-data$life_expectancy_at_birth
> all.models <- leaps(X, y, int = FALSE, strictly.compatible = FALSE, method="Cp")
```

```
#plot the cp value for each model
plot(all.models$size, all.models$Cp, log="y", xlab="|M|",
ylab=expression(C[p]),ylim=c(1,200))
lines(all.models$size, all.models$size) # this plots the line C_p=|M|
```




```
#find the model with lowest cp value
#this finds the smallest C_p value
min.cp <- all.models$Cp == min(all.models$Cp)
#this finds the corresponding model with the smallest C_p
min.cp <- all.models$which[min.cp, ]
min.cp
```

```
> #find the model with lowest cp value
> min.cp <- all.models$Cp == min(all.models$Cp) #this finds the smallest C_p value
> min.cp <- all.models$which[min.cp, ] #this finds the corresponding model with the smallest C_p
> min.cp #
                                (Intercept)                 per_pop_access_to_electricity
                                       TRUE                                         FALSE
       per_annual_growth_national_income                           net_national_income
                                       TRUE                                         FALSE
              per_children_out_of_school                 per_expenditure_primary_education
                                      FALSE                                         FALSE
          infant_mortality_rate_per_1000                        per_adult_literacy_rate
                                       TRUE                                         FALSE
            per_annual_population_growth                              total_population
                                      FALSE                                         FALSE
                  per_primary_completion                      per_gdp_health_expenditure
                                      FALSE                                          TRUE
           health_expenditure_per_capita                                per_unemployment
                                       TRUE                                         FALSE
per_adult_female_mortality_rate_per_1000          per_adult_male_mortality_rate_per_1000
                                       TRUE                                          TRUE
                    per_annual_gdp_growth                                gdp_per_capita
                                      FALSE                                         FALSE
               crude_birth_rate_per_1000                                gni_per_capita
                                       TRUE                                          TRUE
          employment_to_population_ratio
                                      FALSE
```

```
#Selected model based on Cp value
selected_model_cp = lm(formula = life_expectancy_at_birth ~
                         per_annual_growth_national_income +
                       infant_mortality_rate_per_1000 + health_expenditure_per_capita +
                       per_gdp_health_expenditure + per_adult_female_mortality_rate_per_1000 +
                       per_adult_male_mortality_rate_per_1000 + crude_birth_rate_per_1000 +
                       gni_per_capita,
                     data = data)
summary(selected_model_cp)
```

Selected Model using Mallows' Cp and selected model using backward selection give the model with same features as the best model

```
> vif(full_model)
            per_pop_access_to_electricity        per_annual_growth_national_income
                                 5.858202                                 1.496312
                      net_national_income                per_children_out_of_school
                                 2.653762                                 2.060831
        per_expenditure_primary_education           infant_mortality_rate_per_1000
                                 1.085453                                 7.997653
                  per_adult_literacy_rate             per_annual_population_growth
                                 2.167880                                 4.081017
                         total_population                    per_primary_completion
                                 2.585600                                 2.457072
               per_gdp_health_expenditure            health_expenditure_per_capita
                                 2.753956                                 4.900613
                       per_unemployment per_adult_female_mortality_rate_per_1000
                                 1.230909                                16.918009
   per_adult_male_mortality_rate_per_1000                   per_annual_gdp_growth
                                12.038178                                 1.695009
                           gdp_per_capita                crude_birth_rate_per_1000
                                36.027479                                12.737166
                           gni_per_capita           employment_to_population_ratio
                                39.001321                                 1.597252
```

 # selected the columns having vif score less than 5

```
selected_model_vif = lm(formula = life_expectancy_at_birth ~
per_annual_growth_national_income +
                        net_national_income + per_children_out_of_school +
                        per_expenditure_primary_education + per_adult_literacy_rate +
                        per_annual_population_growth + total_population +
                        per_primary_completion + per_gdp_health_expenditure +
                        health_expenditure_per_capita + per_unemployment +
                        per_annual_gdp_growth + employment_to_population_ratio +
                        crude_birth_rate_per_1000 + per_pop_access_to_electricity +
                        infant_mortality_rate_per_1000 + crude_birth_rate_per_1000 +
                        per_adult_male_mortality_rate_per_1000+
per_adult_female_mortality_rate_per_1000,
                    data = data)
summary(selected_model_vif)
```

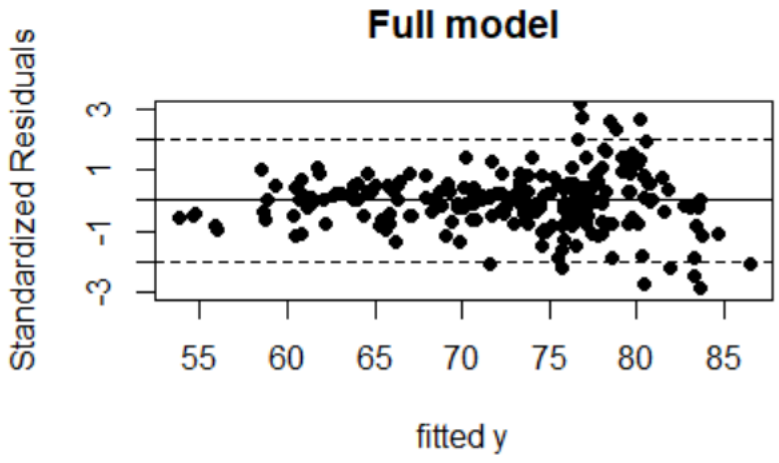
```
> # selected the columns having vif score less than 5
> selected_model_vif = lm(formula = life_expectancy_at_birth ~ per_annual_growth_national_income +
+                           net_national_income + per_children_out_of_school +
+                           per_expenditure_primary_education + per_adult_literacy_rate +
+                           per_annual_population_growth + total_population +
+                           per_primary_completion + per_gdp_health_expenditure +
+                           health_expenditure_per_capita + per_unemployment +
+                           per_annual_gdp_growth + employment_to_population_ratio +
+                           crude_birth_rate_per_1000 + per_pop_access_to_electricity +
+                           infant_mortality_rate_per_1000 + crude_birth_rate_per_1000 +
+                           per_adult_male_mortality_rate_per_1000 + per_adult_female_mortality_rate_per_1000,
+                         data = data)
> summary(selected_model_vif)

Call:
lm(formula = life_expectancy_at_birth ~ per_annual_growth_national_income +
    net_national_income + per_children_out_of_school + per_expenditure_primary_education +
    per_adult_literacy_rate + per_annual_population_growth +
    total_population + per_primary_completion + per_gdp_health_expenditure +
    health_expenditure_per_capita + per_unemployment + per_annual_gdp_growth +
    employment_to_population_ratio + crude_birth_rate_per_1000 +
    per_pop_access_to_electricity + infant_mortality_rate_per_1000 +
    crude_birth_rate_per_1000 + per_adult_male_mortality_rate_per_1000 +
    per_adult_female_mortality_rate_per_1000, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4444 -0.9424 -0.0045  0.8091  8.3286
```

```
Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                              8.648e+01  3.384e+00  25.558  < 2e-16 ***
per_annual_growth_national_income       -3.513e-02  3.635e-02  -0.966   0.3349
net_national_income                      3.174e-15  2.746e-14   0.116   0.9081
per_children_out_of_school              -2.938e-02  2.678e-02  -1.097   0.2738
per_expenditure_primary_education        4.610e-03  3.391e-02   0.136   0.8920
per_adult_literacy_rate                 -1.620e-02  1.582e-02  -1.024   0.3068
per_annual_population_growth             3.236e-01  2.158e-01   1.500   0.1352
total_population                        -1.595e-10  1.994e-10  -0.800   0.4247
per_primary_completion                  -1.757e-02  1.920e-02  -0.915   0.3612
per_gdp_health_expenditure               5.237e-02  7.599e-02   0.689   0.4914
health_expenditure_per_capita            7.221e-04  1.136e-04   6.356 1.24e-09 ***
per_unemployment                        -3.587e-02  3.704e-02  -0.968   0.3340
per_annual_gdp_growth                   -5.627e-02  6.507e-02  -0.865   0.3881
employment_to_population_ratio           2.451e-03  1.581e-02   0.155   0.8770
crude_birth_rate_per_1000               -2.510e-01  4.308e-02  -5.827 2.06e-08 ***
per_pop_access_to_electricity            9.596e-03  1.298e-02   0.739   0.4604
infant_mortality_rate_per_1000          -1.118e-01  1.913e-02  -5.844 1.89e-08 ***
per_adult_male_mortality_rate_per_1000  -1.499e-02  5.788e-03  -2.590   0.0103 *
per_adult_female_mortality_rate_per_1000 -1.845e-02  7.673e-03  -2.405   0.0170 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.952 on 213 degrees of freedom
Multiple R-squared:  0.9314,    Adjusted R-squared:  0.9256
F-statistic: 160.7 on 18 and 213 DF,  p-value: < 2.2e-16
```

Plotting

#plots the standardised residuals against fitted values for FULL model

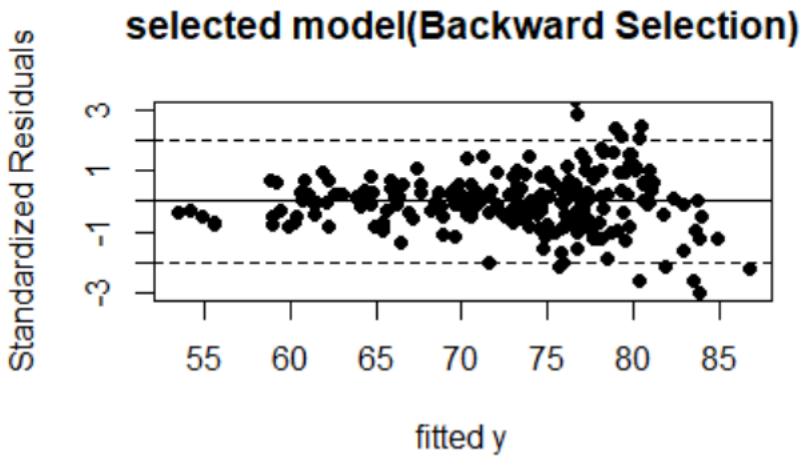
```
stdres_fullmodel<-rstandard(full_model)
plot(full_model$fitted.values,stdres_fullmodel,pch=16,
    ylab="Standardized Residuals",xlab="fitted y",ylim=c(-3,3),main="Full model")
abline(h=0)
abline(h=2,lty=2)
abline(h=-2,lty=2)
```



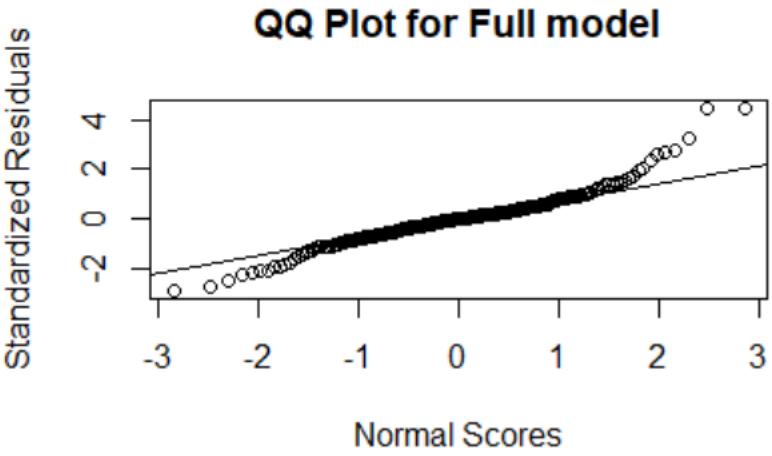
#plots the standardised residuals against fitted values for selected model

```
stdres_selected_model<-rstandard(selected_model)
plot(selected_model$fitted.values,stdres_selected_model,pch=16,
    ylab="Standardized Residuals",xlab="fitted y",ylim=c(-3,3),main="selected
model(Backward Selection)")
abline(h=0)
abline(h=2,lty=2)
```

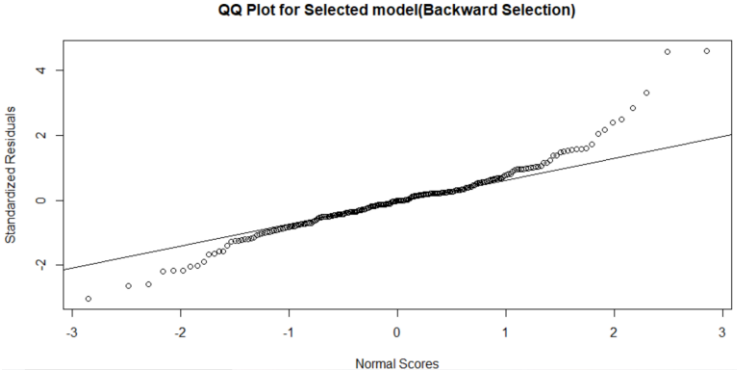
```
abline(h=-2,lty=2)
```




```
#plots the QQ-plot for the FULL model
qqnorm(stdres_fullmodel, ylab="Standardized Residuals",
      xlab="Normal Scores", main="QQ Plot for Full model" )
qqline(stdres_fullmodel)
```




```
#plots the QQ-plot for the selected model
qqnorm(stdres_selected_model, ylab="Standardized Residuals",
      xlab="Normal Scores", main="QQ Plot for Selected model(Backward Selection)" )
qqline(stdres_selected_model)
```



```
########################################
#Model Evaluation

### In order for us to chose between the reduced and the full model we need to use one of the model selection criteria

# Adjusted r-square value
print(paste0("adj. r-square value of Full Model is : ",summary(full_model)$adj.r.squared))
print(paste0("adj. r-square value of selected Model is :
",summary(selected_model)$adj.r.squared))
print(paste0("adj. r-square value of selected Model - VIF is :
",summary(selected_model_vif)$adj.r.squared))
```

```
> # Adjusted r-sqare value
> print(paste0("adj. r-square value of Full Model is : ",summary(full_model)$adj.r.squared))
[1] "adj. r-square value of Full Model is : 0.925913749071697"
> print(paste0("adj. r-square value of selected Model is : ",summary(selected_model)$adj.r.squared))
[1] "adj. r-square value of selected Model is : 0.928072517818729"
> print(paste0("adj. r-square value of selected Model - VIF is : ",summary(selected_model_vif)$adj.r.squared))
[1] "adj. r-square value of selected Model - VIF is : 0.925616310481739"
```

```
#We will consider calculating the AIC
print(paste0("AIC Score of Full Model is : ",AIC(full_model)))
print(paste0("AIC Score of selected Model-BS is : ",AIC(selected_model)))
print(paste0("AIC Score of selected Model-VIF is : ",AIC(selected_model_vif)))
```

```
> ### We will consider calculating the AIC and Mallow's Cp. The AIC can be easily found using the command 'AIC(. . . )'
> print(paste0("AIC Score of Full Model is : ",AIC(full_model)))
[1] "AIC Score of Full Model is : 989.728292098849"
> print(paste0("AIC Score of selected Model-BS is : ",AIC(selected_model)))
[1] "AIC Score of selected Model-BS is : 971.700456968409"
> print(paste0("AIC Score of selected Model-VIF is : ",AIC(selected_model_vif)))
[1] "AIC Score of selected Model-VIF is : 988.846547391993"
>
```

```
#The better fitting model is the one with the lowest value of AIC
#which in this case is the Selected model with AIC = 971.7 compared with Full Model with AIC = 989.7

### We can determine if the reduced model is a viable fit by calculating Mallow's Cp,
#using the 'ols_mallows_cp' function from the 'olsrr' package.
ols_mallows_cp(selected_model,full_model)
ols_mallows_cp(selected_model_vif,full_model)
```

```
> ols_mallows_cp(selected_model,full_model)  #2.502095
[1] 2.502095
> ols_mallows_cp(selected_model_vif,full_model)  #825.5202
[1] 19.85514
```

```
# As this value (2.5)  is much less than the number of features (8). Our Selected model is an acceptable model

#prediction

# load the testdata into a dataframe
test_data <- read.csv("LifeExpectancyData2.csv",header=TRUE)

# check the dimension of test dataset
dim(test_data)
```

```
> # load the testdata into a dataframe
> test_data <- read.csv("LifeExpectancyData2.csv",header=TRUE)
>
> # check the dimension of test dataset
> dim(test_data)
[1] 11 22
>
```

```
test_data <- test_data %>%
  rename( 'per_pop_access_to_electricity'='EG.ELC.ACCS.ZS',
        'per_annual_growth_national_income'='NY.ADJ.NNTY.KD.ZG',
        'net_national_income'='NY.ADJ.NNTY.KD',
        'per_children_out_of_school'='SE.PRM.UNER.ZS',
        'per_expenditure_primary_education'='SE.XPD.PRIM.ZS',
        'infant_mortality_rate_per_1000'='SP.DYN.IMRT.IN',
        'per_adult_literacy_rate'='SE.ADT.LITR.ZS',
        'per_annual_population_growth'='SP.POP.GROW',
        'total_population'='SP.POP.TOTL',
        'per_primary_completion'='SE.PRM.CMPT.ZS',
        'per_gdp_health_expenditure'='SH.XPD.CHEX.GD.ZS',
        'health_expenditure_per_capita'='SH.XPD.CHEX.PC.CD',
        'per_unemployment'='SL.UEM.TOTL.NE.ZS',
        'per_adult_female_mortality_rate_per_1000'='SP.DYN.AMRT.FE',
        'per_adult_male_mortality_rate_per_1000'='SP.DYN.AMRT.MA',
        'per_annual_gdp_growth'='NY.GDP.MKTP.KD.ZG',
        'gdp_per_capita'='NY.GDP.PCAP.PP.CD',
        'crude_birth_rate_per_1000'='SP.DYN.CBRT.IN',
        'gni_per_capita'='NY.GNP.PCAP.PP.CD',
        'employment_to_population_ratio'='SL.EMP.TOTL.SP.ZS'
  )
```

```
> test_data <- test_data %>%
+   rename( 'per_pop_access_to_electricity'='EG.ELC.ACCS.ZS',
+           'per_annual_growth_national_income'='NY.ADJ.NNTY.KD.ZG',
+           'net_national_income'='NY.ADJ.NNTY.KD',
+           'per_children_out_of_school'='SE.PRM.UNER.ZS',
+           'per_expenditure_primary_education'='SE.XPD.PRIM.ZS',
+           'infant_mortality_rate_per_1000'='SP.DYN.IMRT.IN',
+           'per_adult_literacy_rate'='SE.ADT.LITR.ZS',
+           'per_annual_population_growth'='SP.POP.GROW',
+           'total_population'='SP.POP.TOTL',
+           'per_primary_completion'='SE.PRM.CMPT.ZS',
+           'per_gdp_health_expenditure'='SH.XPD.CHEX.GD.ZS',
+           'health_expenditure_per_capita'='SH.XPD.CHEX.PC.CD',
+           'per_unemployment'='SL.UEM.TOTL.NE.ZS',
+           'per_adult_female_mortality_rate_per_1000'='SP.DYN.AMRT.FE',
+           'per_adult_male_mortality_rate_per_1000'='SP.DYN.AMRT.MA',
+           'per_annual_gdp_growth'='NY.GDP.MKTP.KD.ZG',
+           'gdp_per_capita'='NY.GDP.PCAP.PP.CD',
+           'crude_birth_rate_per_1000'='SP.DYN.CBRT.IN',
+           'gni_per_capita'='NY.GNP.PCAP.PP.CD',
+           'employment_to_population_ratio'='SL.EMP.TOTL.SP.ZS'
+   )
>
```

 # Dataframe to write the prediction into

```
Test_final_data <- subset(test_data,select=c(Country.Name,Country.Code))
```

#preprocessing as in the training dataset

```
test_data <- subset(test_data,select=-c(Country.Name,Country.Code))
```

```
> # Dataframe to write the prediction into
> Test_final_data <- subset(test_data,select=c(Country.Name,Country.Code))
>
> #preprocessing as in the training dataset
> test_data <- subset(test_data,select=-c(Country.Name,Country.Code))
```

 # checking if the imputation is done for all the columns

colSums(sapply(test_data, is.na))

```
> colSums(sapply(test_data, is.na))
           per_pop_access_to_electricity        per_annual_growth_national_income
                                       0                                        6
                     net_national_income                per_children_out_of_school
                                       6                                        5
       per_expenditure_primary_education           infant_mortality_rate_per_1000
                                       8                                        1
                 per_adult_literacy_rate             per_annual_population_growth
                                       7                                        0
                        total_population                   per_primary_completion
                                       0                                        6
             per_gdp_health_expenditure            health_expenditure_per_capita
                                       3                                        3
                        per_unemployment per_adult_female_mortality_rate_per_1000
                                       5                                        1
  per_adult_male_mortality_rate_per_1000                    per_annual_gdp_growth
                                       1                                        1
                          gdp_per_capita                crude_birth_rate_per_1000
                                       1                                        0
                          gni_per_capita           employment_to_population_ratio
                                       1                                        0
>
```

#Missing value imputation using mean median of training Dataset

test_data$per_annual_growth_national_income[is.na(test_data$per_annual_growth_national_income)] <- mean(data$per_annual_growth_national_income,na.rm = TRUE)
test_data$per_children_out_of_school[is.na(test_data$per_children_out_of_school)] <- median(data$per_children_out_of_school,na.rm = TRUE)
test_data$per_expenditure_primary_education [is.na(test_data$per_expenditure_primary_education)] <- mean(data$per_expenditure_primary_education ,na.rm = TRUE)
test_data$infant_mortality_rate_per_1000[is.na(test_data$infant_mortality_rate_per_1000)] <- mean(data$infant_mortality_rate_per_1000,na.rm = TRUE)
test_data$per_adult_literacy_rate[is.na(test_data$per_adult_literacy_rate)] <- mean(data$per_adult_literacy_rate,na.rm = TRUE)
test_data$per_annual_population_growth[is.na(test_data$per_annual_population_growth)] <- median(data$per_annual_population_growth,na.rm = TRUE)
test_data$total_population[is.na(test_data$total_population)] <- median(data$total_population,na.rm = TRUE)
test_data$per_primary_completion[is.na(test_data$per_primary_completion)] <- mean(data$per_primary_completion,na.rm = TRUE)
test_data$per_gdp_health_expenditure[is.na(test_data$per_gdp_health_expenditure)] <- mean(data$per_gdp_health_expenditure,na.rm = TRUE)
test_data$health_expenditure_per_capita[is.na(test_data$health_expenditure_per_capita)] <- mean(data$health_expenditure_per_capita,na.rm = TRUE)
test_data$per_unemployment[is.na(test_data$per_unemployment)] <- median(data$per_unemployment,na.rm = TRUE)
test_data$per_adult_female_mortality_rate_per_1000[is.na(test_data$per_adult_female_mortality_rate_per_1000)] <- mean(data$per_adult_female_mortality_rate_per_1000,na.rm = TRUE)

```
test_data$per_adult_male_mortality_rate_per_1000[is.na(test_data$per_adult_male_mortalit
y_rate_per_1000)] <- mean(data$per_adult_male_mortality_rate_per_1000,na.rm = TRUE)
test_data$per_annual_gdp_growth[is.na(test_data$per_annual_gdp_growth)] <-
mean(data$per_annual_gdp_growth,na.rm = TRUE)
test_data$gdp_per_capita[is.na(test_data$gdp_per_capita)] <-
median(data$gdp_per_capita,na.rm = TRUE)
test_data$gni_per_capita[is.na(test_data$gni_per_capita)] <-
mean(data$gni_per_capita,na.rm = TRUE)
test_data$net_national_income[is.na(test_data$net_national_income )] <-
mean(data$net_national_income ,na.rm = TRUE)
test_data$employment_to_population_ratio[is.na(test_data$employment_to_population_ratio
)] <- mean(data$employment_to_population_ratio,na.rm = TRUE)
```

```
> #Missing value imputation using mean median of training Dataset
> test_data$per_annual_growth_national_income[is.na(test_data$per_annual_growth_national_income)] <- mean(data$per_annual_g
rowth_national_income,na.rm = TRUE)
> test_data$per_children_out_of_school[is.na(test_data$per_children_out_of_school)] <- median(data$per_children_out_of_scho
ol,na.rm = TRUE)
> test_data$per_expenditure_primary_education [is.na(test_data$per_expenditure_primary_education )] <- mean(data$per_expend
iture_primary_education ,na.rm = TRUE)
> test_data$infant_mortality_rate_per_1000[is.na(test_data$infant_mortality_rate_per_1000)] <- mean(data$infant_mortality_r
ate_per_1000,na.rm = TRUE)
> test_data$per_adult_literacy_rate[is.na(test_data$per_adult_literacy_rate)] <- mean(data$per_adult_literacy_rate,na.rm =
 TRUE)
> test_data$per_annual_population_growth[is.na(test_data$per_annual_population_growth)] <- median(data$per_annual_populatio
n_growth,na.rm = TRUE)
> test_data$total_population[is.na(test_data$total_population)] <- median(data$total_population,na.rm = TRUE)
> test_data$per_primary_completion[is.na(test_data$per_primary_completion)] <- mean(data$per_primary_completion,na.rm = TRU
E)
> test_data$per_gdp_health_expenditure[is.na(test_data$per_gdp_health_expenditure)] <- mean(data$per_gdp_health_expenditur
e,na.rm = TRUE)
> test_data$health_expenditure_per_capita[is.na(test_data$health_expenditure_per_capita)] <- mean(data$health_expenditure_p
er_capita,na.rm = TRUE)
> test_data$per_unemployment[is.na(test_data$per_unemployment)] <- median(data$per_unemployment,na.rm = TRUE)
> test_data$per_adult_female_mortality_rate_per_1000[is.na(test_data$per_adult_female_mortality_rate_per_1000)] <- mean(dat
a$per_adult_female_mortality_rate_per_1000,na.rm = TRUE)
> test_data$per_adult_male_mortality_rate_per_1000[is.na(test_data$per_adult_male_mortality_rate_per_1000)] <- mean(data$pe
r_adult_male_mortality_rate_per_1000,na.rm = TRUE)
> test_data$per_annual_gdp_growth[is.na(test_data$per_annual_gdp_growth)] <- mean(data$per_annual_gdp_growth,na.rm = TRUE)
> test_data$gdp_per_capita[is.na(test_data$gdp_per_capita)] <- median(data$gdp_per_capita,na.rm = TRUE)
> test_data$gni_per_capita[is.na(test_data$gni_per_capita)] <- mean(data$gni_per_capita,na.rm = TRUE)
> test_data$net_national_income[is.na(test_data$net_national_income )] <- mean(data$net_national_income ,na.rm = TRUE)
> test_data$employment_to_population_ratio[is.na(test_data$employment_to_population_ratio)] <- mean(data$employment_to_popu
lation_ratio,na.rm = TRUE)
```

 # checking if the imputation is done for all the columns

```
colSums(sapply(test_data, is.na)
```

```
> # checking if the imputation is done for all the columns
> colSums(sapply(test_data, is.na))
           per_pop_access_to_electricity        per_annual_growth_national_income
                                       0                                        0
                     net_national_income                per_children_out_of_school
                                       0                                        0
       per_expenditure_primary_education            infant_mortality_rate_per_1000
                                       0                                        0
                  per_adult_literacy_rate             per_annual_population_growth
                                       0                                        0
                         total_population                    per_primary_completion
                                       0                                        0
             per_gdp_health_expenditure            health_expenditure_per_capita
                                       0                                        0
                         per_unemployment per_adult_female_mortality_rate_per_1000
                                       0                                        0
  per_adult_male_mortality_rate_per_1000                     per_annual_gdp_growth
                                       0                                        0
                           gdp_per_capita                crude_birth_rate_per_1000
                                       0                                        0
                           gni_per_capita            employment_to_population_ratio
                                       0                                        0
>
```

 # prediction of value

```
y_predicted <- predict(selected_model, test_data)
y_predicted
```

```
> # prediction of value
> y_predicted <- predict(selected_model, test_data)
> y_predicted
       1        2        3        4        5        6        7        8        9       10       11
57.81303 77.13978 62.48102 77.02421 74.56215 81.73870 71.30715 66.16971 57.89312 73.01140 60.77381
```

#store the details into the dataframe having country name and code
Test_final_data <- cbind(Test_final_data,y_predicted)

#write to a csv file
write.csv(Test_final_data,'Predicted_life_expectancy.csv')

```
> #store the details into the dataframe having country name and code
> Test_final_data <- cbind(Test_final_data,y_predicted)
>
> #write to a csv file
> write.csv(Test_final_data,'Predicted_life_expectancy.csv')
>
```

#One Way ANOVA:

library(countrycode)

 # Grouping countries into continents

data['continent'] <- countrycode(sourcevar = data[,"Country.Name"],

origin = "country.name",

destination = "continent")

```
> library(countrycode)
> # Grouping countries into continents
> data['continent'] <- countrycode(sourcevar = data[,"Country.Name"],
+                                  origin = "country.name",
+                                  destination = "continent")
Warning message:
In countrycode(sourcevar = data[, "Country.Name"], origin = "country.name",  :
  Some values were not matched unambiguously: Arab World, Caribbean small states, Central Europe and the Baltics, Early-demographic dividend, East Asia & Pacific, East Asia & Pacific (excluding high income), East Asia & Pacific (IDA & IBRD countries), Euro area, Europe & Central Asia, Europe & Central Asia (excluding high income), Europe & Central Asia (IDA & IBRD countries), European Union, Fragile and conflict affected situations, Heavily indebted poor countries (HIPC), High income, IBRD only, IDA & IBRD total, IDA blend, IDA only, IDA total, Kosovo, Late-demographic dividend, Latin America & Caribbean, Latin America & Caribbean (excluding high income), Latin America & the Caribbean (IDA & IBRD countries), Least developed countries: UN classification, Low & middle income, Low income, Lower middle income, Middle East & North Africa, Middle East & North Africa (excluding high income), Middle East & North Africa (IDA & IBRD countries), Middle income, North America, OECD members, Other sma [... truncated]
```

The warning is because there are few unidentified country names. And while building anova model, all those null continents i.e data for those 47 country name is being omitted.

 # Calculating Means of group means

group.means<-tapply(data$life_expectancy_at_birth,data$continent,mean)

group.means

```
> group.means<-tapply(data$life_expectancy_at_birth,data$continent,mean)
> group.means
  Africa Americas     Asia   Europe  Oceania
64.11582 75.48770 74.25291 79.28906 74.11959
```

#box plot for comparing life expectancies

boxplot(data$life_expectancy_at_birth~data$continent,

main = "Comparing Life Expectancy of Continents",

xlab = "Continents", ylab = "Life Expectancy")

```
> boxplot(data$life_expectancy_at_birth~data$continent,
+         main = "Comparing Life Expectancy of Continents",
+         xlab = "Continents", ylab = "Life Expectancy")
```

 # Conducting One Way ANOVA

anova1way<-aov(data$life_expectancy_at_birth~as.factor(continent),data=data)

 # Summary of the ANOVA model

summary(anova1way)

```
> # Conducting One Way ANOVA
> anova1way<-aov(data$life_expectancy_at_birth~as.factor(continent),data=data)
> # Summary of the ANOVA model
> summary(anova1way)
                      Df Sum Sq Mean Sq F value Pr(>F)
as.factor(continent)   4   5845  1461.3   61.57 <2e-16 ***
Residuals            180   4272    23.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
47 observations deleted due to missingness
```

#pairwise t test

pairwise.t.test(data$life_expectancy_at_birth, data$continent, p.adj = "bonferroni")

```
> #pairwise ttest
> pairwise.t.test(data$life_expectancy_at_birth, data$continent, p.adj = "bonferroni")

        Pairwise comparisons using t tests with pooled SD

data:  data$life_expectancy_at_birth and data$continent

         Africa  Americas Asia    Europe
Americas < 2e-16 -        -       -
Asia     < 2e-16 1.0000   -       -
Europe   < 2e-16 0.0071   4.0e-05 -
Oceania  1.4e-08 1.0000   1.0000  0.0150

P value adjustment method: bonferroni
```

 # Plotting Differences in mean levels of life expectancy across continents

tukey.Exp<-TukeyHSD(anova1way)

 # Set the margin on all sides

par(mar = c(3, 8, 4, 5))

plot(tukey.Exp,las = 2) ’’’

```
> # Plotting Differences in mean levels of life expectancy across continents
> tukey.Exp<-TukeyHSD(anova1way)
> par(mar = c(3, 8, 4, 5)) # Set the margin on all sides to 2
> plot(tukey.Exp,las = 2)
```

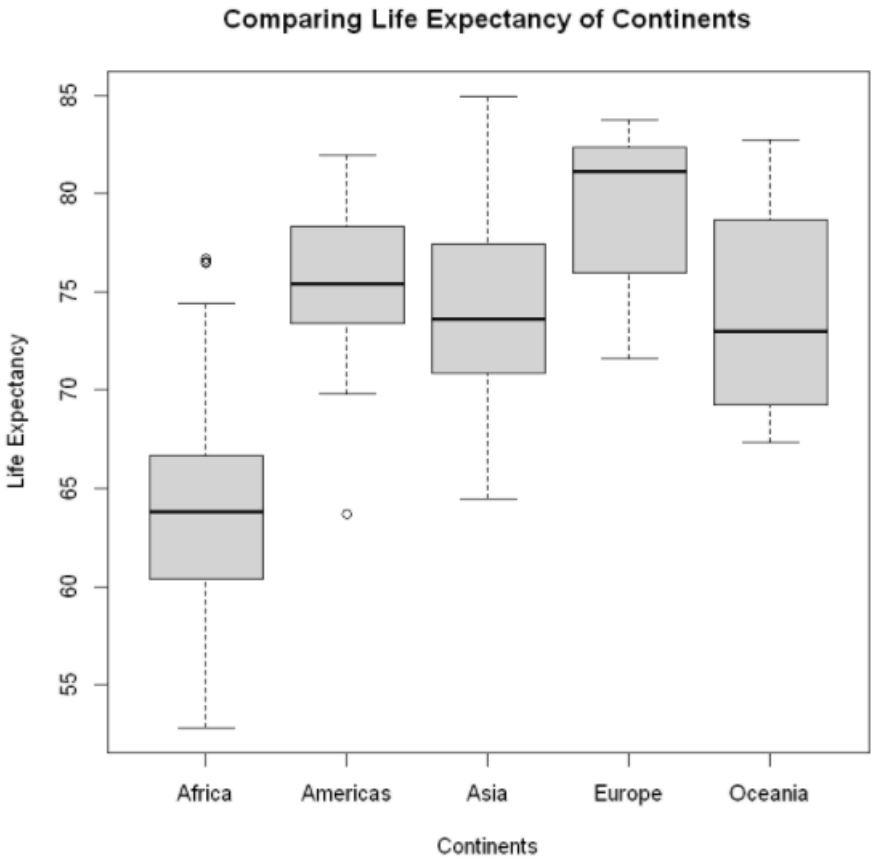
<u>Explanation of code and outputs:</u>

For grouping countries into continents library(countrycode) is used. After grouping continents one-way ANOVA test is conducted.

Results of Mean Life Expectancies of continents:

Africa: 64.1158160974 **Americas:** 75.4876973340009 **Asia:** 74.2529051488889 **Europe:** 79.28906463425 **Oceania:** 74.1195914641667

Box plot for Life expectancies Across Continents:



Results of pairwise t test:

```
         Africa  Americas Asia    Europe
Americas < 2e-16 -        -       -
Asia     < 2e-16 1.0000   -       -
Europe   < 2e-16 0.0071   4.0e-05 -
Oceania  1.4e-08 1.0000   1.0000  0.0150

P value adjustment method: bonferroni
```