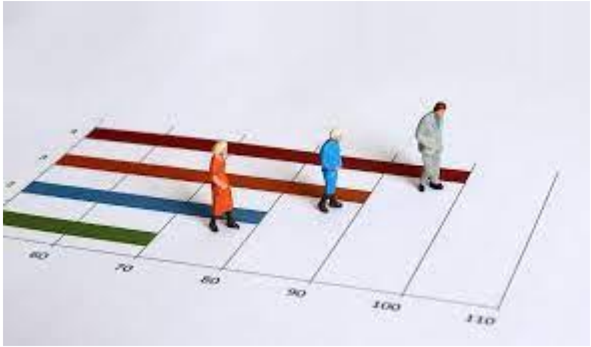
University of Essex

Group Project: MA317 -Experimental modelling:

Descriptive Statistical Study on Global Life Expectancy

Data, Model, and outcomes-Dr. Stella Hadjiantoni



Group Members/:

1.	Zia ullah khan	2101142	zk21867@essex.ac.uk	-Question 1 2 4 5 + Report + Presentation
2.	Aeman Hassan	2100681	az21206@essex.ac.uk	-Question 3 2 5 1 + Report + Presentation
3.	Akash Kumar	2105425	ak21434@essex.ac.uk	-Question 1 2 3 4 + Report + Presentation
4.	Wahyu Maiwa	2107460	wm21615@essex.ac.uk	-Question 1 2 3 4 + Report + Presentation
5.	Prateek Tiwary	2100920	pt21775@essex.ac.uk	-Question 1 2 3 5 + Report + Presentation

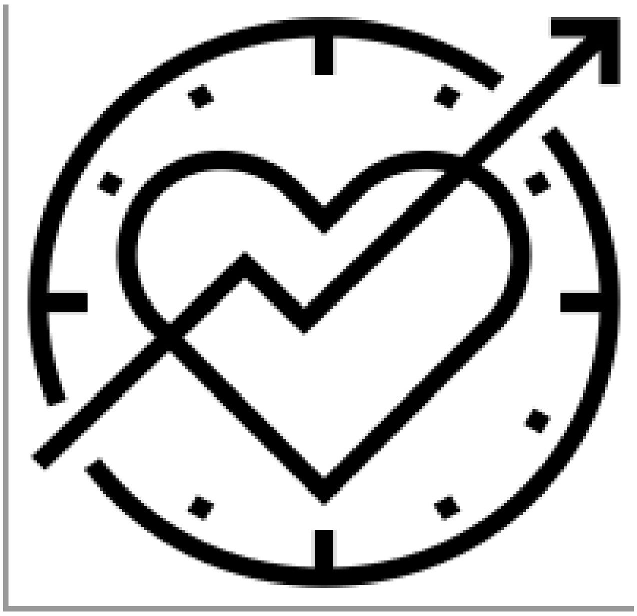
ABSTRACT

Group projects not only give us practical experience and allow them to put what has been taught into practise, but they also accustomed creative construction and group dynamics. Group initiatives have proven to be extremely valuable. They are infrequently utilised in online education courses, as they are in traditional learning. The underpinnings of this study are examined. It shows how to use group projects efficiently and then how to include them into online learning courses.

Content

Introduction

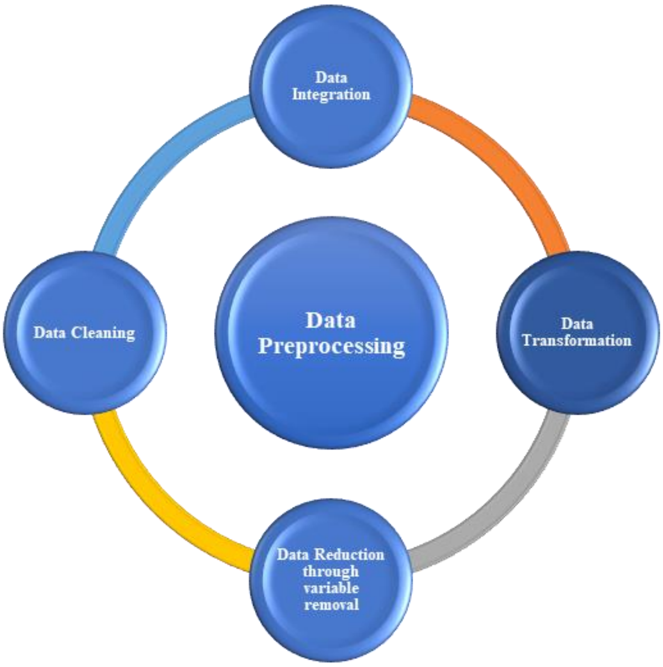
"Life expectancy would have grown leaps and bounds if green vehicle smelled as good as bacon. – Doug Larson"



Life expectancy is a demographical mapping of a living being's typical life span based on his date of birth, present age, sex, and other factors. In this research paper, we want to apply the techniques and concepts learned in the "Modelling Experimental Data" course to a real-world collection of observations on factors influencing life expectancy.

It must be conducted both graphically and numerically. The report focuses on two area of findings. Firstly, to develop a model for predicting the life expectancy for countries in 2019 and secondly to provide a similar prediction for the nations whose data for the evaluation were missing. This can be done by considering the other factors affecting the such

We'll look at demographic variables, income composition, and death rates for 216 countries in a cross-sectional dataset for 2019. The study will look at how factors including birth rate, mortality, GDP, health spending, the HIV-affected population, and drinking water sanitation can affect life expectancy. Life expectancy will be the dependent variable in a multivariate regression model with five independent factors. The information comes from the World Development Indicators of the World Bank. The World Bank undertook various mortality statistics studies. As a result, this year has been chosen for our study. The data has been subjected to a number of adjustments and simplifications in order for us to be able to answer a complex phenomenon using the fundamental statistic, Microsoft Excel, and R.



Pre-processing of Data:

Analyzing data that hasn't been fully scrutinised for these flaws can lead to erroneous results. As a result, the representation and quality of data must come first before any analysis. In computational statistics, data preprocessing is frequently the most important stage of a machine learning project.It is the most initial but critical part of Model designing. It has four main steps to be conducted.:

- Loading the data set in R by using ‘read.csv()’ function.
- By looking at the data set and using the ‘summary()’ and ‘str()’ functions to acquire a summary of the data set, you can perform some visual descriptive analysis.
- Factoring the class label, Continent, to make it a categorical variable.
- Some extraneous columns will be deleted as well, as will those that will not be used in the analysis. Rank column is similar to the first.

Points of note:

• Unlike Python, which uses Numpy arrays to store data and conduct operations, R allows us to do operations directly on the dataset, which is a list. We don't need to categorise the dependent and independent components directly because R uses an attribute called formula to distinguish dependent and independent parts from a dataset. Organizing Categorical Data:

Categorical variables are data types that can be separated into categories. Race, sex, age group, educational level, and other category characteristics are examples. We have two categorical characteristics in our dataset: nation and purchased item. The factor method in R can be used to transform text into numerical codes.

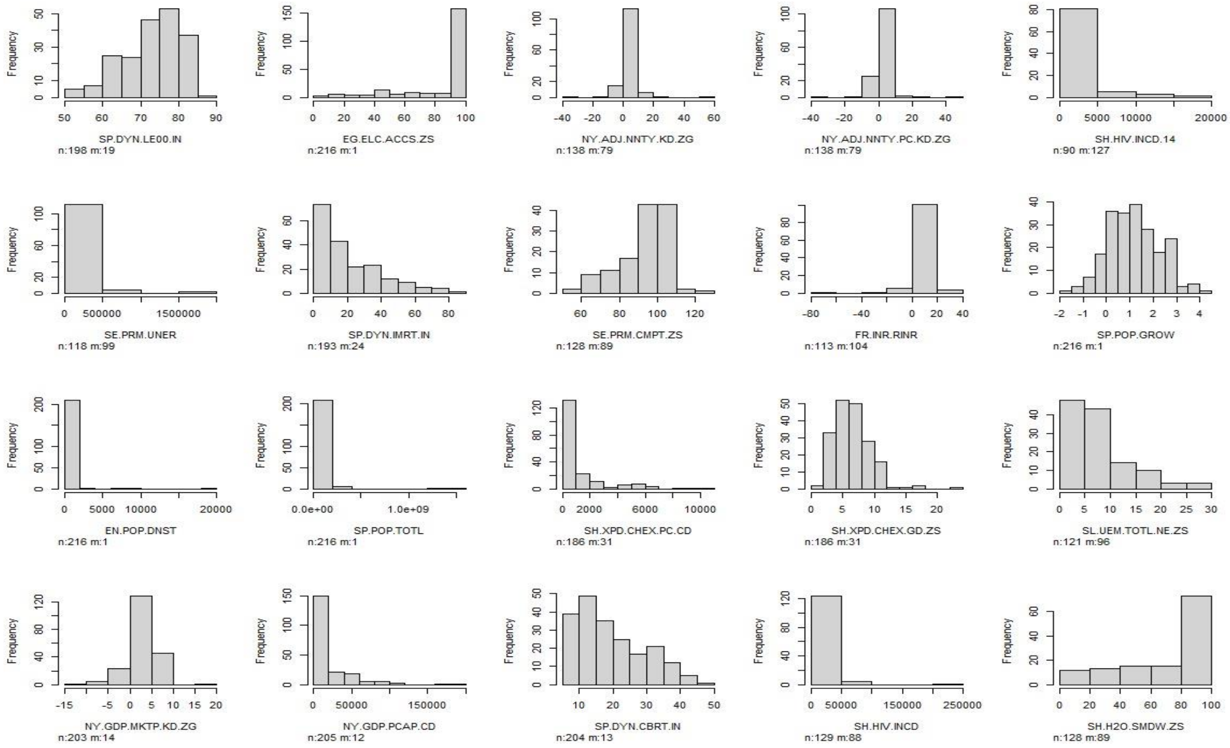
Analysis, Question 1 to 2:

Descriptive analysis is defined as one of the blocks in data science which helps in defining or interpreting the data at a very initial level. In simpler words, it defines the data by working on the statistical distribution, the anomalies in the data, the level of dispersion or other factors related to our area of interest.

In data the concept of central tendency refers to that one number that acts as a benchmark defining the entire data with respect to it. Mean, Median and Mode are the terms that are defined as those numbers around which the whole data revolves. Mean or Average is the same term and is a number which estimates the whole dataset. Median however is a value that divides the data in two halves. Other than these few other conditions are also to be observed such as Standard deviation, percentile and Interquartile range Lastly, there is mode which is the most frequently repeated value of the dataset. From the given dataset, column such as life expectancy and population its growth and density possess as very significant difference which was later curtailed down with the help of logarithmic and square root approach to make the data points normally distributed.

Mean and Median often connects the data to the understanding of Skewness. Skewness is a measure of probability distribution. It can be positive, negative or undefined. If the data represents a perfect normal distribution, then each side of the skewness curve will be a mirror image. The Skewness of data is considered negative if the median is higher than the mean and positive if the mean value of the data is more. After further mathematical transformation of variables such as access to electricity, population density and GDP per capita stilled contained left skewness which means the mean-median difference value were not affected.

Figure: Histogram showing Variable summary.



Locating of Missing Value:

The following are the five steps to ensuring that missing data is accurately recognised and dealt with:

1------**M**ake sure your data is properly coded.

2---------------**W**ithin each variable, look for missing values.

3---------------------**L**ook for missing-person patterns.

4--------------------------------**E**xamine the data that is missing and the data that is present.

5--**M**ake a decision about how to deal with missing data.

Missing data are a common occurrence and can have a significant effect on the inference that can be drawn from the data.

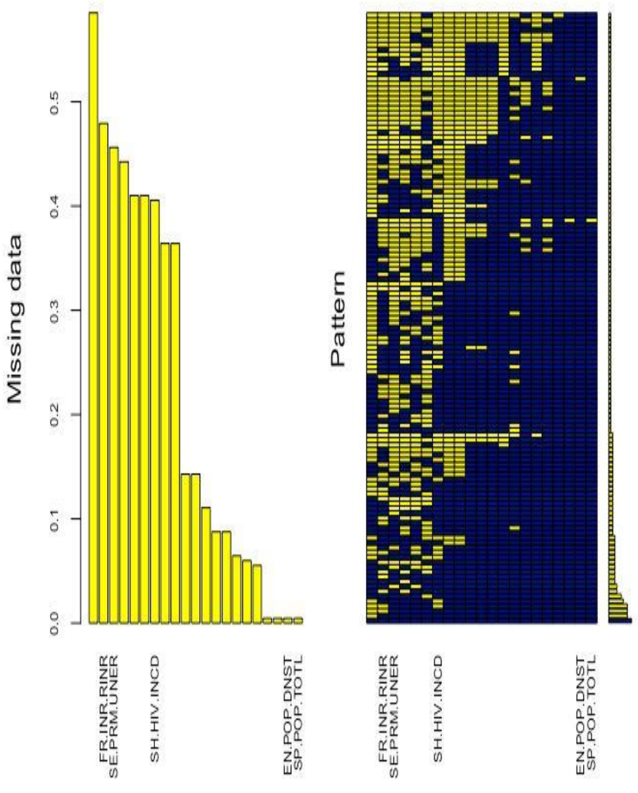
Types of Missing Data

- Missing Completely at Random (MCAR): We consider data to be totally missing at random when missing values are dispersed at random across all observations.
- Missing at Random (MAR): The key difference between MCAR and MAR is that with MAR, data is missing in sub-samples of data rather than across all observations.
- Not Missing at Random (NMAR): We can't consider missing data as if it's missing at random when it has a structure. If data was absent for all students from specific schools in the preceding scenario, the data could not be considered MAR.

Method of resolving missing value.:

- **Dropping columns:** variable which have more than 60% of data missing must be removed because it doesn't depict any meaningful information also it creates bias.
  .
- **Multivariate Imputation by Chained Equations (MICE):** Using predictive mean matching (imputation method) dependent variables are imputed though an iterative series of predictive model in each iteration a specific variable in the dependent variable is imputed using other filled variables. It iterates until all the null values are imputed.
  Where m=5 implies that no more than 5 consecutive iteration usually performed.

  Linear Regression: we have performed prediction method which tries to fit a linear regression model using the dependent variable and predict the independent variable and the model was tested by available target variable in the dataset.5 models were fit and there respective F-statics, R-squared, P-values and residual standard error was analysed and the model 5 came out to be the best model for imputation of missing values which have
  F-statics =106.5
  R-square = 0.9113
  P-value = 2.2e16
  Residual standard error 2.289
  Best imputed model is used to predict the missing values in independent variable.



Discussion, Question 3 4 5

Including unessential variables makes the model bias/overfitting and complex. It can cause great difficulty in identifying statistically significant model. Feature selection helps in improving performance and reducing training time.

Feature collinearity is a measure of linear relationship between features and target. The reason behind feature collinearity is to find the features which are highly correlated with each other and target variables. The features which are highly correlated with others can be dropped.

In this dataset we will be using two methods to find the features for further analysis

Variance Inflation Factor.

It determines how much the variance of an independent variable is inflated (increased) by its interaction/correlation with other independent variables. It also gives a quick assessment of how much a variable contributes to the standard error of the regression.

In the best imputed model, a Variance Inflation was analysed and VIF of over 10 indicates that the variables have high correlation among each other. Features having value more than 10 was dropped have high multicollinearity with other dependent variable which helps the model to have more generalisation. Features such as adjusted national income & per capita have very hight inflation value and health expenditure, GDP per capita, population growth and birth rate have significant linear relation with independent variable.

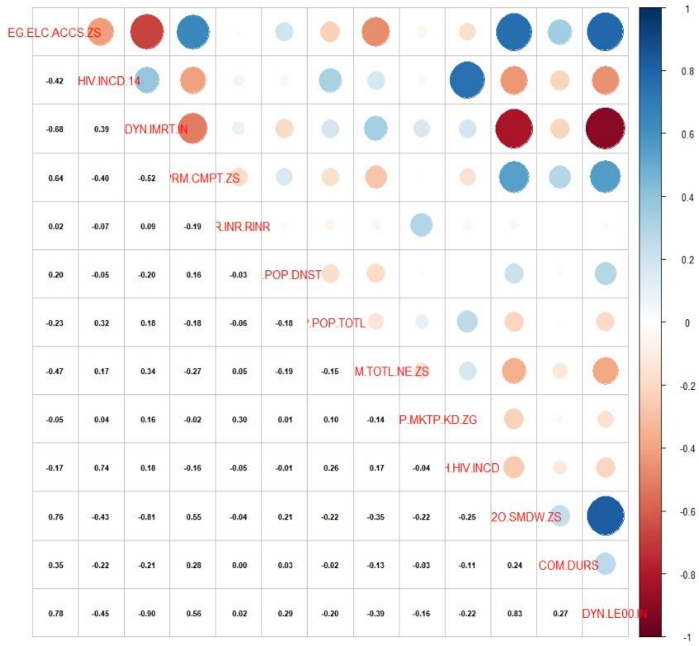
Correlation Matrix.

Correlation metrics are used to determine whether or not two variables are related with each other. A correlation can be either positive or negative.

- Positive correlation: both variables increase together linearly proportional
- Negative correlation: one rises makes other falls inverse proportional.

Correlation value greater than 0.70 indicated highly correlated and can be dropped

Dependent variable such as Children infected with HIV and people using safely manged drinking water have very high correlation with the independent variable.



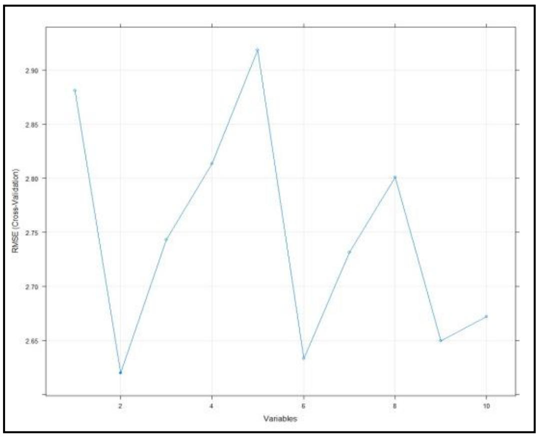
Factors affecting Life Expectancy

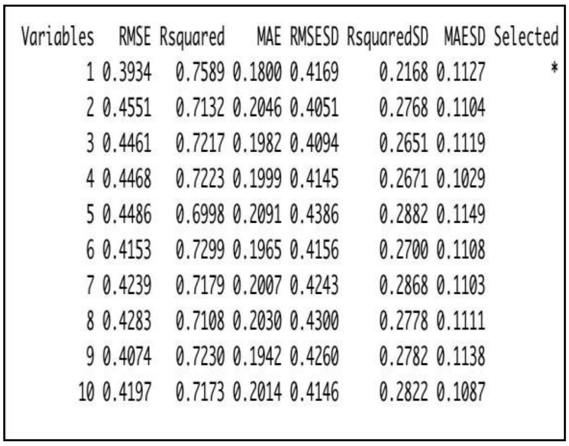
In the life Expectancy dataset, we have a total 29 features and after pre-processing the dataset we have 14 features for our analysis. Life expectancy at birth (SP.DYN.LE00.IN) is response variable and other 14 are predictor variables.

Ranking feature by importance

Recursive feature elimination: In the recursion method we recursively fit the model with every predictor variable and analyze residual, coefficient, adjusted-R square and P-value. A small p-value (less than 5%) is a significant criterion to select the null hypothesis and consequently convey the strength of the relationship with the response variable. R-squared value helps in finding the significance of the selected model. There are three techniques to find the optimal number of features to fit the model significantly.

On performing recursive feature on elimination on variance influenced dataset with 10-fold cross validation, we got newly HIV infected children as a important feature.





```
Variables   RMSE Rsquared    MAE RMSESD RsquaredSD  MAESD Selected
        1 0.3934   0.7589 0.1800 0.4169     0.2168 0.1127        *
        2 0.4551   0.7132 0.2046 0.4051     0.2768 0.1104
        3 0.4461   0.7217 0.1982 0.4094     0.2651 0.1119
        4 0.4468   0.7223 0.1999 0.4145     0.2671 0.1029
        5 0.4486   0.6998 0.2091 0.4386     0.2882 0.1149
        6 0.4153   0.7299 0.1965 0.4156     0.2700 0.1108
        7 0.4239   0.7179 0.2007 0.4243     0.2868 0.1103
        8 0.4283   0.7108 0.2030 0.4300     0.2778 0.1111
        9 0.4074   0.7230 0.1942 0.4260     0.2782 0.1138
       10 0.4197   0.7173 0.2014 0.4146     0.2822 0.1087
```

Stepwise regression: Fitting the model with every pair of predictor variable and response variable and selecting the least p-valued variable and recursively doing so with the remaining variable until we get a desired combination(p-value< 5%).

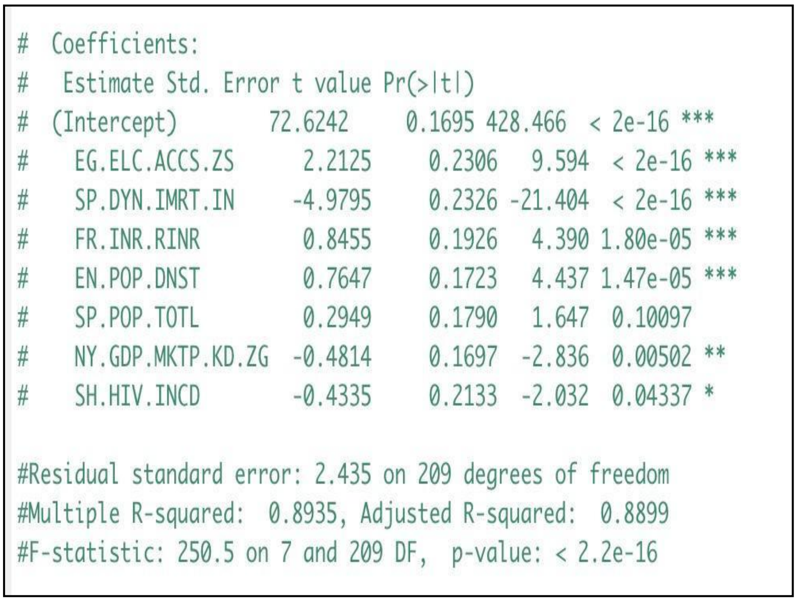
Forward selection: Fitting the model by adding features and keeping only the features which increases the overall model fit.

Backward elimination: Fitting all prediction variables at initial and subsequently removing the variable which has highest p-value until desired accuracy.

AIC: AIC is designed to locate the variation in data, **while penalizing for models that use an excessive number of parameters**.

BIC: The Bayesian Information Criterion (BIC) is a statistic for comparing the goodness of fit of various regression models. In practise, we fit numerous regression models to the same dataset and select the model with the lowest BIC value as the best fit. Feature given below suggest the best suitable model.

Summary of life expectancy best model:best AIC



```
#  Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
#  (Intercept)       72.6242    0.1695 428.466  < 2e-16 ***
#    EG.ELC.ACCS.ZS      2.2125     0.2306   9.594  < 2e-16 ***
#    SP.DYN.IMRT.IN     -4.9795     0.2326 -21.404  < 2e-16 ***
#    FR.INR.RINR         0.8455     0.1926   4.390 1.80e-05 ***
#    EN.POP.DNST         0.7647     0.1723   4.437 1.47e-05 ***
#    SP.POP.TOTL         0.2949     0.1790   1.647  0.10097
#    NY.GDP.MKTP.KD.ZG  -0.4814     0.1697  -2.836  0.00502 **
#    SH.HIV.INCD        -0.4335     0.2133  -2.032  0.04337 *

#Residual standard error: 2.435 on 209 degrees of freedom
#Multiple R-squared:  0.8935, Adjusted R-squared:  0.8899
#F-statistic: 250.5 on 7 and 209 DF,  p-value: < 2.2e-16
```

Question 5: At last, with the second data set shared to us we shall be using one way ANOVA. ANOVA (Analysis of Variance) is a statistical test that is used to examine the differences between the means of many groups. It uses one independent variable for analysis. One-way Anova is conducted on one dependent numeric and one independent categorical variable.

It depicts the effect of change in dependent variable changes with respect to independent variable.

By determining whether the means of the treatment levels differ from the overall mean of the dependent variable, ANOVA establishes if the groups formed by the levels of the independent variable are statistically different.

The null hypothesis is rejected if any of the group means deviate considerably from the overall mean.

The F-test compares the variation in each group's mean to the variance in the entire group. The F-test will find a higher F-value if the variance within groups is smaller than the variance between groups, indicating a higher possibility that the difference seen is real and not due to chance.

The null hypothesis in ANOVA is that there is no difference between group means. The ANOVA will show a statistically significant result if any group differs considerably from the overall group mean.

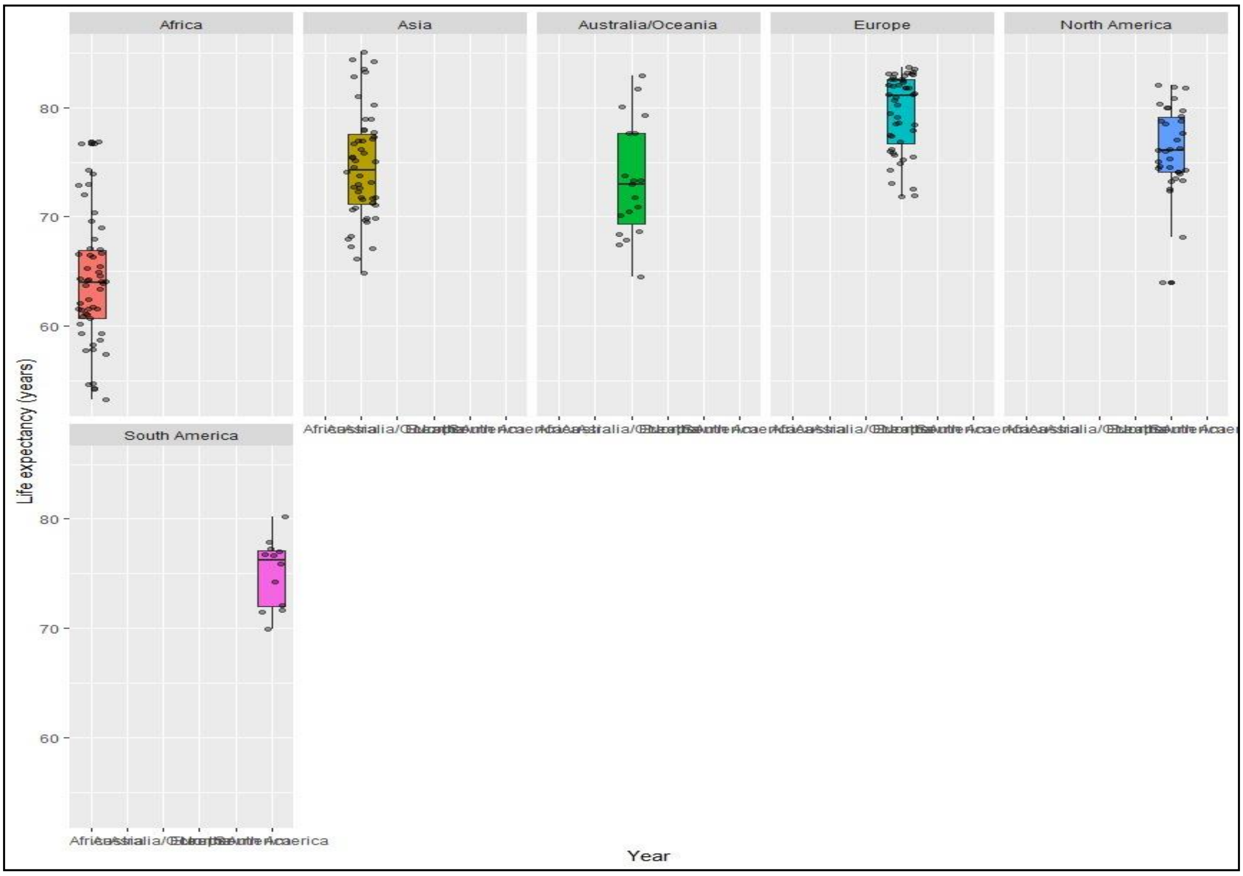
The F statistic, which is the ratio of the mean sum of squares (the variance explained by the independent variable) to the mean square error, is used to assess significant differences between group averages (the variance left over).

The difference between groups is considered statistically significant if the F statistic is greater than the critical value (the value of F that corresponds to your alpha value, usually 0.05). because we have one factor continent and we will check the Average life expectancies against the different treatment of groups.

Following are the assumptions that oneway ANOVA follows:

- Observational independence;
- normally distributed response variable;
- variance homogeneity

Figure:



It is quite clear from the graph and from the Anova test there is high Fscore which is the ratio of variance between and within the group and Pvalue is less than 0.05 threshold,so it is statiscally significant and we can there is strong association between life expectancy and continent variable.

Conclusion:

To summarise although the project was quite enriching, however, the data could have been in detail for a better understanding of variation among the variables in the data set.

References:

1)fmwww.bc.edu

2)sthda.com

3)bookdown.org

4)University of Essex

5)World Bank

6) https://databank.worldbank.org/source/world-development-indicators

Appendix:

```
#Packages required
  install.packages('gapminder')
  install.packages('finalfit')
  install.packages('Hmisc')
  install.packages('ggpubr')
  install.packages('psych')
  install.packages("mice")
  install.packages('faraway')
  install.packages('corrplot')
  install.packages('mlbench')
  install.packages('caret')



#loading libraries
  library(ggplot2)
  library(finalfit)  #package for finishing tabulation %  reference: https://finalfit.org/
  library(gapminder) #package for finding missing values %  reference:
https://cran.r-project.org/web/packages/gapminder/README.html
  library(Hmisc)
  library("ggpubr") # package must be installed first
  library(psych)
  library(mice)    #reference https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4701517/
  library(VIM)     #reference
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4701517/
  library(faraway)
  library(corrplot)
  library(mlbench)
  library(caret)

#load in data
life_expectancy <- read.csv("Life_Expectancy_Data1 _1.csv")
```

```
## List characteristics of the dataframe
head(life_expectancy)
str(life_expectancy)
colnames(life_expectancy)


#1) we will remove country name and country code columns    1) Country Code    2) Country Name

#saving continent in another variable for reusing in question5
var_continent <- life_expectancy$Continent;

#Three columns have been eliminated as country and country codes are unique
#Removed EG.FEC.RNEW.ZS as all the values are null/NA's
life_expectancy <- life_expectancy[, -c(1,2,3, 25)]

missing_glimpse(life_expectancy) # missing data for each variable, we will remove any variable in which missing values are more than 60%


# Following columns are to be removed as they contain more than 80% missing data points
# "SE.PRM.CUAT.ZS" 83.4
# "SE.TER.CUAT.BA.ZS" 82.5
# "SE.ADT.LITR.ZS" 88.5
# "SI.POV.LMIC"89.9

life_expectancy <- life_expectancy[, !(colnames(life_expectancy) %in% c("SE.PRM.CUAT.ZS","SE.TER.CUAT.BA.ZS","SE.ADT.LITR.ZS", "SI.POV.LMIC"))]



#histogram
#Histograms are the  exploratory plots
#because they show densities of data and can assist in providing better distributional information.

hist.data.frame(life_expectancy)

#Normal distribution : SP.DYN.LE00.IN , SP.POP.GROW , NY.ADJ.NNTY.KD.ZG ,FR.INR.RINR ,SH.XPD.CHEX.GD.ZS ,

#Positively skewed :   SH.HIV.INCD.14 , SE.PRM.UNER, SP.DYN.IMRT.IN, EN.POP.DNST , SP.POP.TOTL, SH.XPD.CHEX.PC.CD, SL.UEM.TOTL.NE.ZS ,
                        #NY.GDP.MKTP.KD.ZG, NY.GDP.PCAP.CD, SP.DYN.CBRT.IN, SH.HIV.INCD

#Negatively skewed : EG.ELC.ACCS.ZS ,NY.ADJ.NNTY.PC.KD.ZG , SE.PRM.CMPT.ZS , SH.H2O.SMDW.ZS,

# After analyzing features, following could be converted to normal distribution by either applying log or taking sqrt(square root)


#examining distribution of dependent variable i.e. SP.DYN.LE00.IN
```

```
        ggplot(life_expectancy, aes(SP.DYN.LE00.IN)) + geom_density(fill="blue")  #
dependent variable distribution
        ggplot(life_expectancy, aes(log(SP.DYN.LE00.IN))) + geom_density(fill="blue")
        ggplot(life_expectancy, aes(sqrt(SP.DYN.LE00.IN))) + geom_density(fill="blue")


        # Following lines of code is used to check the distribution of features
        ggplot(life_expectancy, aes(SH.H2O.SMDW.ZS)) + geom_density(fill="green")  #
dependent variable distribution
        ggplot(life_expectancy, aes(log(SH.H2O.SMDW.ZS))) + geom_density(fill="green")
        ggplot(life_expectancy, aes(sqrt(SH.H2O.SMDW.ZS))) + geom_density(fill="green")


        #Following features have been transformed by applying either log or square root
(sqrt)
        life_expectancy$SH.XPD.CHEX.GD.ZS <-
log(life_expectancy$SH.XPD.CHEX.GD.ZS)   # for normal distribution
        life_expectancy$SP.DYN.IMRT.IN <- log(life_expectancy$SP.DYN.IMRT.IN)
        life_expectancy$EN.POP.DNST <- log(life_expectancy$EN.POP.DNST)
        life_expectancy$SP.POP.TOTL <- log(life_expectancy$SP.POP.TOTL)
        life_expectancy$SH.XPD.CHEX.PC.CD <-
log(life_expectancy$SH.XPD.CHEX.PC.CD)
        life_expectancy$SL.UEM.TOTL.NE.ZS <-
sqrt(life_expectancy$SL.UEM.TOTL.NE.ZS)
        life_expectancy$NY.GDP.PCAP.CD <- log(life_expectancy$NY.GDP.PCAP.CD)
        life_expectancy$SP.DYN.CBRT.IN <- log(life_expectancy$SP.DYN.CBRT.IN)

        #Now we are checking distribution again, we can see now that most of the data is
normally distributed
        hist.data.frame(life_expectancy)


        #density plot  allow to analyze the spread and the shape of the distribution
        plot(density(life_expectancy$SH.XPD.CHEX.PC.CD, na.rm = TRUE))
        plot(density(life_expectancy$SP.DYN.IMRT.IN, na.rm = TRUE))
        plot(density(life_expectancy$SL.UEM.TOTL.NE.ZS, na.rm = TRUE))
        plot(density(life_expectancy$NY.GDP.PCAP.CD, na.rm = TRUE))
        plot(density(life_expectancy$SP.DYN.CBRT.IN, na.rm = TRUE))


        ggdensity(life_expectancy$NY.GDP.PCAP.CD,
                main = "Density plot of POP.Grow",
                xlab = "POP.Grow"
        )
        ggdensity(life_expectancy$SL.UEM.TOTL.NE.ZS,
                main = "Density plot of POP.Grow",
                xlab = "POP.Grow"
        )

        #Description of qq plots
        ggqqplot(life_expectancy$SH.XPD.CHEX.PC.CD)
        qqPlot(life_expectancy$SL.UEM.TOTL.NE.ZS)

        qqnorm(life_expectancy$SP.DYN.CBRT.IN, pch = 1, frame = FALSE)
        qqline(life_expectancy$SP.DYN.CBRT.IN, col = "red", lwd = 2)
```

```
		qqnorm(life_expectancy$SH.XPD.CHEX.PC.CD, pch = 1, frame = FALSE)
		qqline(life_expectancy$SH.XPD.CHEX.PC.CD, col = "red", lwd = 2)

		#scatterplot
		#Often you will want to see how to numeric variables relate to each other, and 
scatterplot (simply plot())
		#From scatter analyzes we can see that most of the variables show positive or 
negative correlation

		#examining correlation between indendepndt and dependent variables

		plot(life_expectancy$SP.DYN.LE00.IN ~ life_expectancy$SH.XPD.CHEX.GD.ZS)

		plot(life_expectancy$SP.DYN.LE00.IN ~ life_expectancy$SP.POP.GROW)

		plot(life_expectancy$SP.DYN.LE00.IN ~ life_expectancy$SP.DYN.IMRT.IN)

		ggplot(data = life_expectancy) +
		  geom_point(mapping = aes(x = EG.ELC.ACCS.ZS, y = SP.DYN.LE00.IN))

		ggplot(data = life_expectancy) +
		  geom_point(mapping = aes(x = EN.POP.DNST, y = SP.DYN.LE00.IN))

		ggplot(data = life_expectancy) +
		  geom_point(mapping = aes(x = NY.GDP.PCAP.CD, y = SP.DYN.LE00.IN))


		#Detecting outliers

		 df1 <- life_expectancy[,1:7]
		 df2 <- life_expectancy[,8:14]
		 df3 <- life_expectancy[,15:21]

		 par(mar=c(1,1,1,1))

		# From below box plot, it is clear depicting outliers in different variables,
		# SH.HIV.INCD.14, Fr.INR.RINR and some other variables are showing maximum 
outliers along with few others.
		boxplot(df1, col = rainbow(ncol(df1)))
		boxplot(df2, col = rainbow(ncol(df2)))
		boxplot(df3, col = rainbow(ncol(df3)))



		#Descriptive statistics

		#summary: Results of summary will be in the report section, summary is showing the 
null values,  mean, median
		summary(life_expectancy)  # done with r summary function

		#Descriptive statistics with describeBy()   reference: 
https://statsandr.com/blog/descriptive-statistics-in-r/
```

```
            #The describeBy() function from the {psych} package allows to report several
            #summary statistics (i.e., number of valid cases, mean, standard deviation, median,
trimmed mean, mad:
            #median absolute deviation (from the median), minimum, maximum, range,
skewness and kurtosis) by a grouping variable.

            describeBy(
              life_expectancy
            )


            #------------------------------------------------------------------------------------#


            #-------------------------Question_2----------------------------------------------#

            #life expectancy in another variable

            life_exp_q1 <- life_expectancy

            str(life_expectancy)

            dim(life_expectancy)

            missing_glimpse(life_expectancy)  # percentage of missing values

            # Below is the table showing missing values % in each column

            #  Features                                  Missing count     Missing values percentage

            #SP.DYN.LE00.IN            SP.DYN.LE00.IN    <dbl> 198       19           8.8
            #EG.ELC.ACCS.ZS            EG.ELC.ACCS.ZS    <dbl> 216        1            0.5
            #NY.ADJ.NNTY.KD.ZG       NY.ADJ.NNTY.KD.ZG    <dbl> 138       79          36.4
            #NY.ADJ.NNTY.PC.KD.ZG NY.ADJ.NNTY.PC.KD.ZG    <dbl> 138       79
36.4
            #SH.HIV.INCD.14            SH.HIV.INCD.14    <int>  90      127          58.5
            #SE.PRM.UNER                SE.PRM.UNER    <dbl> 118       99          45.6
            #SP.DYN.IMRT.IN            SP.DYN.IMRT.IN    <dbl> 193       24          11.1
            #SE.PRM.CMPT.ZS            SE.PRM.CMPT.ZS    <dbl> 128       89          41.0
            #FR.INR.RINR               FR.INR.RINR    <dbl> 113      104          47.9
            #SH.XPD.CHEX.PC.CD      SH.XPD.CHEX.PC.CD    <dbl> 186       31          14.3
            #SH.XPD.CHEX.GD.ZS      SH.XPD.CHEX.GD.ZS    <dbl> 186       31          14.3
            #SL.UEM.TOTL.NE.ZS      SL.UEM.TOTL.NE.ZS    <dbl> 121       96          44.2
            #SH.HIV.INCD               SH.HIV.INCD    <int> 129       88          40.6
            #SH.H2O.SMDW.ZS            SH.H2O.SMDW.ZS    <dbl> 128       89          41.0


            # The md.pattern() function along with Multivariate Imputation by Chained
Equations (MICE) package
            # helps in producing a table displaying the missing pattern
            md.pattern(life_expectancy)

            # The below pattern is displaying that there are 42 rows with no missing values, 54
rows in which there is one column
            # data missing
```

```
md.pattern(life_expectancy[,c(1:7)],rotate.names = TRUE)
md.pattern(life_expectancy[,c(8:14)])
md.pattern(life_expectancy[,c(15:21)])

md.pairs(life_expectancy)


par(mar=c(1,1,1,1))
marginplot(life_expectancy[,c('SP.DYN.LE00.IN', 'EG.ELC.ACCS.ZS')])

#Nonmissing values are displayed in blue color and missing values are in red color. There are 19 missing values on SP.DYN.LE00.IN
marginplot(life_expectancy[,c('EG.ELC.ACCS.ZS', 'SP.DYN.LE00.IN')])

#Nonmissing values are displayed in blue color and missing values are in red color. There are 99 missing values on SP.DYN.LE00.IN
# and 19 missing values in another column SE.PRM.UNER
marginplot(life_expectancy[,c('SE.COM.DURS', 'SE.PRM.UNER')])

life_expect_impute <- life_expectancy #store data in another variable to preserve life expectancy variable


#More than 59% values in the data set with no missing value.
#There are 36% missing values in NY.ADJ.NNTY.PC.KD.ZG, 14% missing values in SH.XPD.CHEX.PC.CD and SH.XPD.CHEX.GD.ZS and so on.
mice_plot <- aggr(life_expect_impute, col=c('navyblue','yellow'),
          numbers=TRUE, sortVars=TRUE,
          labels=names(life_expect_impute), cex.axis=.7,
          gap=3, ylab=c("Missing data","Pattern"))


colnames(life_expect_impute)

#method Applying multiple imputation

# storing dependent variable life expectancy and will scale others -- applying normallization, scaling as some values are bigger
sp.leoo.in <- life_expect_impute$SP.DYN.LE00.IN
subset_life_expect = life_expect_impute
subset_life_expect <- subset_life_expect[, !(colnames(subset_life_expect) %in% c("SP.DYN.LE00.IN"))]

subset_life_expect.scaled = scale(subset_life_expect, center= TRUE, scale=TRUE)

imputed_Data <- mice(subset_life_expect.scaled, m=5, maxit = 50, method = 'pmm', seed = 500)
summary(imputed_Data)

#check imputed values
imputed_Data$imp$EG.ELC.ACCS.ZS

# we are applying two different methods to impute data i.e m = 1,2

#Question 2 part 2, imputed dependent variable
```

```
        # https://bookdown.org/mwheymans/bookmi/single-missing-data-imputation.html
#reference code has been done here
        #using regression to impute missing values in dependent variable
        #The life expectancy variables are used to predict the missing dependent variable
values

        # The method “norm.predict” in the mice package fits a linear regression model in
the dataset and generates the imputed values
        # for the variable by using the regression coefficients of the linear regression model.
        # The completed dataset can be extracted by using the complete function in the mice
package.
        # Complete data


        imputed_Data <- mice(subset_life_expect.scaled, m=5, maxit = 50, method = 'pmm',
seed = 500)
        summary(imputed_Data)

        dataset1 <- complete(imputed_Data,1)
        dataset2 <- complete(imputed_Data,2)
        dataset3 <- complete(imputed_Data,3)
        dataset4 <- complete(imputed_Data,4)
        dataset5 <- complete(imputed_Data,5)

        dataset1$SP.DYN.LE00.IN <- sp.leoo.in
        dataset2$SP.DYN.LE00.IN <- sp.leoo.in
        dataset3$SP.DYN.LE00.IN <- sp.leoo.in
        dataset4$SP.DYN.LE00.IN <- sp.leoo.in
        dataset5$SP.DYN.LE00.IN <- sp.leoo.in

        imp.regress1 <- mice(dataset1, method="norm.predict", m=1, maxit=1)
        imp.regress2 <- mice(dataset2, method="norm.predict", m=1, maxit=1)
        imp.regress3 <- mice(dataset3, method="norm.predict", m=1, maxit=1)
        imp.regress4 <- mice(dataset4, method="norm.predict", m=1, maxit=1)
        imp.regress5 <- mice(dataset5, method="norm.predict", m=1, maxit=1)

        completeData_life_expectancy1 <- complete(imp.regress1,1)
        completeData_life_expectancy2 <- complete(imp.regress2,1)
        completeData_life_expectancy3 <- complete(imp.regress3,1)
        completeData_life_expectancy4 <- complete(imp.regress4,1)
        completeData_life_expectancy5 <- complete(imp.regress5,1)


        fit1 <- lm(SP.DYN.LE00.IN ~ ., data = completeData_life_expectancy1)
        fit2 <- lm(SP.DYN.LE00.IN ~ ., data = completeData_life_expectancy2)
        fit3 <- lm(SP.DYN.LE00.IN ~ ., data = completeData_life_expectancy3)
        fit4 <- lm(SP.DYN.LE00.IN ~ ., data = completeData_life_expectancy4)
        fit5 <- lm(SP.DYN.LE00.IN ~ ., data = completeData_life_expectancy5)

        summary(fit1)   # F-statistic: 0.907  , R-Squared :  0.9074  ,  p-value: < 2.2e-16   ,
Residual standard error: 2.335
        summary(fit2)   # F-statistic: 105.8  , R-Squared :  0.9108  ,  p-value: < 2.2e-16   ,
Residual standard error: 2.309
        summary(fit3)   # F-statistic: 97.46  , R-Squared :  0.9038  ,  p-value: < 2.2e-16   ,
Residual standard error: 2.391
```

```
summary(fit4)   # F-statistic: 102.1  , R-Squared :  0.9078  ,  p-value: < 2.2e-16   , Residual standard error: 2.325
summary(fit5)   # F-statistic: 106.5  , R-Squared :  0.9113  ,  p-value: < 2.2e-16   , Residual standard error: 2.289

# From the above, we can conclude that Imputation m = 5 is giving the best imputated data values.

summary(fit5)   # F-statistic: 106.5  , R-Squared :  0.9113  ,  p-value: < 2.2e-16   , Residual standard error: 2.289

best_imputed_model <- completeData_life_expectancy5


#-----------------------------------------------------------------------------------#

#-------------------------Question_3----------------------------------------------#



#handling outliers  -> outliers will remove rows, so we will not apply it to our original variable/dataset

#method -- IQR
life_exp_outliers <- completeData_life_expectancy5

colnames(life_exp_outliers)

df <- life_exp_outliers

#find absolute value of z-score for each value in each column
z_scores <- as.data.frame(sapply(df, function(df) (abs(df-mean(df))/sd(df))))

#view first six rows of z_scores data frame
head(z_scores)

#only keep rows in dataframe with all z-scores less than absolute value of 3
no_outliers <- z_scores[!rowSums(z_scores>3), ]

#view row and column count of new data frame
dim(no_outliers)



#Question 3 model collinearity starts from here:

print(best_imputed_model)

pairs(best_imputed_model, col = "dodgerblue")

summary(fit5)

#method 1 through VIF factor

vif(fit5)
```

```
#Variation inflation factor for variables

# EG.ELC.ACCS.ZS    NY.ADJ.NNTY.KD.ZG NY.ADJ.NNTY.PC.KD.ZG SH.HIV.INCD.14       SP.DYN.IMRT.IN       SE.PRM.CMPT.ZS         FR.INR.RINR SP.POP.GROW
#   4.491983         2693.112913         2639.304321            3.278377            6.863412 2.458868            1.418490          114.194711
# EN.POP.DNST         SP.POP.TOTL    SH.XPD.CHEX.PC.CD SH.XPD.CHEX.GD.ZS    SL.UEM.TOTL.NE.ZS    NY.GDP.MKTP.KD.ZG NY.GDP.PCAP.CD       SP.DYN.CBRT.IN
#   1.333766            1.932735           204.285173           15.336280            1.975925 1.888550          183.148762            11.227590
# SH.HIV.INCD       SH.H2O.SMDW.ZS         SE.COM.DURS
#   3.002139            6.741018            1.321556


# We will drop those columns whose VIF is greater than 10, vif greater than 10 means there variable has high multicollinearity and
# should be removed from the dataset

#following columns should be dropped because VIF is greater than 10

# NY.ADJ.NNTY.KD.ZG       2693.112913
# NY.ADJ.NNTY.PC.KD.ZG    2639.304321
# SH.XPD.CHEX.PC.CD  204.285173
# SH.XPD.CHEX.GD.ZS  15.336280
# NY.GDP.PCAP.CD   183.148762
# SP.POP.GROW   114.194711
# SP.DYN.CBRT.IN  11.227590


#Following columns are dropped

best_imputed_model <- best_imputed_model[, !(colnames(best_imputed_model) %in% c("NY.ADJ.NNTY.KD.ZG","NY.ADJ.NNTY.PC.KD.ZG","SH.XPD.CHEX.PC.CD", "SH.XPD.CHEX.GD.ZS"))]
model_VIF_less_than_10 <- best_imputed_model[, !(colnames(best_imputed_model) %in% c("NY.GDP.PCAP.CD","SP.POP.GROW","SP.DYN.CBRT.IN"))]

colnames(model_VIF_less_than_10)


#Method 2  -> we find correlation of variables and high correlation more than 70% will be removed
cor1 = cor(model_VIF_less_than_10)
corrplot.mixed(cor1, lower.col = 'black', number.cex = .7)

# reference :  https://stackoverflow.com/questions/35095638/caret-package-findcorrelation-function
set.seed(7)

correlationMatrix <- cor(model_VIF_less_than_10)
# summarize the correlation matrix
print(correlationMatrix)
```

```
            # find attributes that are highly corrected (ideally >0.70)
            highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.70)
            # print indexes of highly correlated attributes
            print(highlyCorrelated)


            # from the correlation it is clear that these features/variables have high correlation
            # SH.HIV.INCD.14  SH.H2O.SMDW.ZS

            model_VIF_less_than_10 <- model_VIF_less_than_10[,
!(colnames(model_VIF_less_than_10) %in% c("SH.HIV.INCD.14","SH.H2O.SMDW.ZS"))]

            #Final model after collinearity variables/columns

            colnames(model_VIF_less_than_10)

            # "EG.ELC.ACCS.ZS"    "SP.DYN.IMRT.IN"    "SE.PRM.CMPT.ZS"
            # "FR.INR.RINR"   "EN.POP.DNST"  "SP.POP.TOTL"   "SL.UEM.TOTL.NE.ZS"
            # "NY.GDP.MKTP.KD.ZG" "SH.HIV.INCD" "SE.COM.DURS"
            # "SP.DYN.LE00.IN"

            #------------------------------------------------------------#

            #-------------------------Question4----------------------------------------------#


            #suggest the best model
            # 1) Remove Redundant Features
            # 2) Rank Features By Importance
            # 3) Feature Selection -- Recursive Feature Elimination or RFE


            lifeexpectancy_model1 = lm(SP.DYN.LE00.IN ~ ., data = model_VIF_less_than_10)
            summary(lifeexpectancy_model1)



            lifeexpectancy_model2 = lm(SP.DYN.LE00.IN ~ -SE.PRM.CMPT.ZS -
SL.UEM.TOTL.NE.ZS - SE.COM.DURS, data = model_VIF_less_than_10)
            summary(lifeexpectancy_model2)


            lifeexpectancy_model3 = lm(SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS  +
SP.DYN.IMRT.IN  + FR.INR.RINR + EN.POP.DNST + SP.POP.TOTL  +
NY.GDP.MKTP.KD.ZG + NY.GDP.MKTP.KD.ZG +  SH.HIV.INCD, data =
model_VIF_less_than_10)
            summary(lifeexpectancy_model3)
            #Residual standard error: 2.435 on 209 degrees of freedom
            #Multiple R-squared:  0.8935,        Adjusted R-squared:  0.8899
            #F-statistic: 250.5 on 7 and 209 DF,  p-value: < 2.2e-16
```

```
#reference: https://machinelearningmastery.com/feature-selection-with-the-caret-r-package/
# Recursive feature elimination
#-> from the graph it is clear that with 2,6 features, we can get a model with root mean squared error less than 2.65
# A random forest algorithm is used to evaulate features
# ensure the results are repeatable
set.seed(7)
# define the control using a random forest selection function
control <- rfeControl(functions=rfFuncs, method="cv", number=10)
# run the RFE algorithm
results <- rfe(model_VIF_less_than_10[,1:10], model_VIF_less_than_10[,11], sizes=c(1:12), rfeControl=control)
# summarize the results
print(results)
# list the chosen features
predictors(results)
# plot the results
plot(results, type=c("g", "o"))





# AIC BIC and Adjusted  R squared   criteria for model selection


set.seed(1234)

#Selection Procedures

#Backward Search

life_expect_mod1 = lm(SP.DYN.LE00.IN ~ ., data = model_VIF_less_than_10[,1:11])
coef(life_expect_mod1)
extractAIC(life_expect_mod1) # returns both p and AIC  ->  11.0000 399.3832


life_expect_mod1_back_aic = step(life_expect_mod1, direction = "backward")
# least AIC=394.16

n = length(resid(life_expect_mod1))
(p = length(coef(life_expect_mod1)))

aic_factor <- n * log(mean(resid(life_expect_mod1) ^ 2)) + 2 * p
aic_factor  # 399.3832

coef(life_expect_mod1_back_aic)


#using BIC

n = length(resid(life_expect_mod1))
life_expect_mod1_back_bic = step(life_expect_mod1, direction = "backward", k = log(n))
```

```
coef(life_expect_mod1_back_bic)

#From the below it can be observed that adjusted r squared is almost same for model
1, backward aic and backward bic with k = log(n)
#AIC R squared
summary(life_expect_mod1)$adj.r.squared     # 0.8887421

summary(life_expect_mod1_back_aic)$adj.r.squared   # 0.8899456

#BIC R squared
summary(life_expect_mod1_back_bic)$adj.r.squared   # 0.8880772


#functions

# From the text: http://daviddalpiaz.github.io/appliedstats/variable-selection-and-
model-building.html
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
calc_rmse = function(actual, predicted) {
  sqrt(sum((actual - predicted)^2) / length(actual))
}
calc_avg_per_error = function(actual, predicted) {
  inter_abs = abs(predicted - actual)
  100 * (sum(inter_abs / actual)) / length(actual)
}


calc_loocv_rmse(life_expect_mod1)           # 2.542271

calc_loocv_rmse(life_expect_mod1_back_aic)   # 2.510068
calc_loocv_rmse(life_expect_mod1_back_bic)   # 2.513588

#We see that we would prefer the model chosen via AIC if using LOOCV RMSE as
our metric.


#Forward Search

colnames(model_VIF_less_than_10)
#"EG.ELC.ACCS.ZS"    "SP.DYN.IMRT.IN"     "SE.PRM.CMPT.ZS"
"FR.INR.RINR"       "EN.POP.DNST"        "SP.POP.TOTL"        "SL.UEM.TOTL.NE.ZS"
"NY.GDP.MKTP.KD.ZG" "SH.HIV.INCD"        "SE.COM.DURS"        "SP.DYN.LE00.IN"

life_expect_mod1_start = lm(SP.DYN.LE00.IN ~ 1, data =
model_VIF_less_than_10[,1:11])
life_expect_mod1_forw_aic = step(life_expect_mod1_start, scope = SP.DYN.LE00.IN
~ EG.ELC.ACCS.ZS  + SP.DYN.IMRT.IN  +
                                    EN.POP.DNST + SP.POP.TOTL  + NY.GDP.MKTP.KD.ZG +
SE.COM.DURS + SH.HIV.INCD + SL.UEM.TOTL.NE.ZS + FR.INR.RINR +
SE.PRM.CMPT.ZS,  direction = "forward")

#Step:  AIC=394.16
```

```
            #SP.DYN.LE00.IN ~ SP.DYN.IMRT.IN + EG.ELC.ACCS.ZS + EN.POP.DNST +
            #  FR.INR.RINR + NY.GDP.MKTP.KD.ZG + SH.HIV.INCD + SP.POP.TOTL

            # Df Sum of Sq    RSS    AIC
            # <none>                      1239.6 394.16
            # + SE.PRM.CMPT.ZS     1    3.4876 1236.2 395.55
            # + SE.COM.DURS        1    1.2270 1238.4 395.95
            # + SL.UEM.TOTL.NE.ZS  1    0.0012 1239.6 396.16   least AIC value


            life_expect_mod1_forw_bic = step(
              life_expect_mod1_start,
              scope = SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS  + SP.DYN.IMRT.IN  +
                EN.POP.DNST + SP.POP.TOTL  + NY.GDP.MKTP.KD.ZG +  SE.COM.DURS +
SH.HIV.INCD + SL.UEM.TOTL.NE.ZS + FR.INR.RINR + SE.PRM.CMPT.ZS,  direction =
"forward", k = log(n))
            # Step:  AIC=416.16
            # SP.DYN.LE00.IN ~ SP.DYN.IMRT.IN + EG.ELC.ACCS.ZS + EN.POP.DNST +
FR.INR.RINR + NY.GDP.MKTP.KD.ZG   This is the best model against forward AIc 416.16


            summary(life_expect_mod1)$adj.r.squared             # 0.8887421

            summary(life_expect_mod1_forw_aic)$adj.r.squared     # 0.8899456

            summary(life_expect_mod1_forw_bic)$adj.r.squared     # 0.8880772


            calc_loocv_rmse(life_expect_mod1)           # 2.542271

            calc_loocv_rmse(life_expect_mod1_forw_aic)   # 2.510068

            calc_loocv_rmse(life_expect_mod1_forw_bic)   # 2.513588


            #We can compare the two selected models' Adjusted R2 as well as their LOOCV
RMSE
            #The results are very similar to those using backwards selection, although the models
are not exactly the same.


            #Stepwise Search

            life_expect_mod1_both_aic = step(
              life_expect_mod1_start,
              scope =  SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS  + SP.DYN.IMRT.IN  +
                EN.POP.DNST + SP.POP.TOTL  + NY.GDP.MKTP.KD.ZG +  SE.COM.DURS +
SH.HIV.INCD + SL.UEM.TOTL.NE.ZS + FR.INR.RINR + SE.PRM.CMPT.ZS,
              direction = "both")


            life_expect_mod1_both_bic = step(
              life_expect_mod1_start,
              scope =  SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS  + SP.DYN.IMRT.IN  +
                EN.POP.DNST + SP.POP.TOTL  + NY.GDP.MKTP.KD.ZG +  SE.COM.DURS +
SH.HIV.INCD + SL.UEM.TOTL.NE.ZS + FR.INR.RINR + SE.PRM.CMPT.ZS,
               direction = "both", k = log(n))


            summary(life_expect_mod1)$adj.r.squared           # 0.8887421

            summary(life_expect_mod1_both_aic)$adj.r.squared   # 0.8899456

            summary(life_expect_mod1_both_bic)$adj.r.squared   # 0.8880772

            calc_loocv_rmse(life_expect_mod1)          # 2.542271

            calc_loocv_rmse(life_expect_mod1_both_aic)  # 2.510068

            calc_loocv_rmse(life_expect_mod1_both_bic)  # 2.513588



            #Exhaustive Search

            install.packages('leaps')
            library(leaps)

            all_life_expectancy_mod = summary(regsubsets(SP.DYN.LE00.IN ~ ., data =
model_VIF_less_than_10[,1:11]))

            all_life_expectancy_mod$which


            all_life_expectancy_mod$rss

            all_life_expectancy_mod$adjr2

            #  find which model has the highest Adjusted R2 we can use the which.max()
function.

            (best_r2_ind = which.max(all_life_expectancy_mod$adjr2))


            all_life_expectancy_mod$which[best_r2_ind, ]


            p = length(coef(life_expect_mod1))
            n = length(resid(life_expect_mod1))

            life_expect_model1_aic = n * log(all_life_expectancy_mod$rss / n) + 2 * (2:p)

            best_aic_ind = which.min(life_expect_model1_aic)
            all_life_expectancy_mod$which[best_aic_ind,]

            life_expect_mod1_best_aic = lm(SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS +
SP.DYN.IMRT.IN + FR.INR.RINR + EN.POP.DNST + SP.POP.TOTL +
NY.GDP.MKTP.KD.ZG + SH.HIV.INCD  , data = model_VIF_less_than_10[,1:11])
```

```
            extractAIC(life_expect_mod1_best_aic)

            extractAIC(life_expect_mod1_back_aic)

            extractAIC(life_expect_mod1_forw_aic)

            extractAIC(life_expect_mod1_both_aic)


            plot(life_expect_model1_aic ~ I(2:p), ylab = "AIC", xlab = "p, number of
parameters",
                pch = 20, col = "dodgerblue", type = "b", cex = 2,
                main = "AIC vs Model Complexity")

            # We could easily repeat this process for   BIC


            life_expect_mod1_bic = n * log(all_life_expectancy_mod$rss / n) + log(n) * (2:p)

            which.min(life_expect_mod1_bic)

            all_life_expectancy_mod$which[5,]

            life_expectancy_mod1_best_bic = lm(SP.DYN.LE00.IN ~ EG.ELC.ACCS.ZS +
SP.DYN.IMRT.IN + FR.INR.RINR + EN.POP.DNST + SP.POP.TOTL +
NY.GDP.MKTP.KD.ZG + SH.HIV.INCD  , data = model_VIF_less_than_10[,1:11])

            extractAIC(life_expectancy_mod1_best_bic, k = log(n))

            extractAIC(life_expect_mod1_back_bic, k = log(n))

            extractAIC(life_expect_mod1_forw_bic, k = log(n))

            extractAIC(life_expect_mod1_both_bic, k = log(n))

            # best models
            summary(life_expect_mod1_best_aic)

            summary(life_expectancy_mod1_best_bic)

            #  Coefficients:
            #   Estimate Std. Error t value Pr(>|t|)
            #  (Intercept)       72.6242     0.1695 428.466  < 2e-16 ***
            #    EG.ELC.ACCS.ZS      2.2125     0.2306   9.594  < 2e-16 ***
            #    SP.DYN.IMRT.IN     -4.9795     0.2326 -21.404  < 2e-16 ***
            #    FR.INR.RINR         0.8455     0.1926   4.390 1.80e-05 ***
            #    EN.POP.DNST         0.7647     0.1723   4.437 1.47e-05 ***
            #    SP.POP.TOTL         0.2949     0.1790   1.647  0.10097
            #    NY.GDP.MKTP.KD.ZG  -0.4814     0.1697  -2.836  0.00502 **
            #    SH.HIV.INCD        -0.4335     0.2133  -2.032  0.04337 *

            #Residual standard error: 2.435 on 209 degrees of freedom
            #Multiple R-squared:  0.8935,        Adjusted R-squared:  0.8899
            #F-statistic: 250.5 on 7 and 209 DF,  p-value: < 2.2e-16
```

```
#----------------------------------------------------------------------#



#-------------------------Question 5 -----------------------------------#

# var_continent -> variable we store continents in question 1

#dataset from question 4
summary(model_VIF_less_than_10)

factor_Continent <- var_continent

model_VIF_less_than_10$Continent <- factor_Continent


one_way_Anova_model <- model_VIF_less_than_10[,c(11,12)]

factor_level <- as.factor(one_way_Anova_model$Continent)
factor_level

table(one_way_Anova_model$Continent)


group_mean <- group_by(one_way_Anova_model, Continent) %>%
  summarise(
    mean = mean(SP.DYN.LE00.IN, na.rm = TRUE),
    sd = sd(SP.DYN.LE00.IN, na.rm = TRUE)
  )

group_mean

#A tibble: 6 x 3
#Continent         mean    sd
#<chr>            <dbl> <dbl>
#  1 Africa           64.1  5.93
#2 Asia             74.6  5.07
#3 Australia/Oceania  73.3  5.25
#4 Europe           79.5  3.54
#5 North America      76.1  3.90
#6 South America      75.1  3.15

one_way_Anova_model %>%
  ggplot(aes(x = factor(Continent), y = SP.DYN.LE00.IN)) +
  geom_boxplot(aes(fill = Continent)) +     # add colour to boxplots
  geom_jitter(alpha = 0.4) +               # alpha = transparency
  facet_wrap(~ Continent, ncol = 5) +       # spread by continent
  theme(legend.position = "none") +         # remove legend
  xlab("") +                          # label x-axis
  ylab("Life expectancy (years)")       # label y-axis



# Compute the analysis of variance
```

```
aov.model = aov(SP.DYN.LE00.IN ~ Continent,data=one_way_Anova_model)  #do the analysis of variance

# Summary of the analysis
summary(aov.model) #show the summary table

#The output includes the columns F value (59.68) and Pr(>F) -> <2e-16 corresponding to the p-value of the test.
# P value is less than the threshold value 0.05, so there is a statistical difference between the groups/continents.

#The above Anova test has a significant F-test score (59.68) and a small P-value,
#we can conclude that there is a strong association between a life expectancy and continent variables.

#Null Hypothesis
# The null hypothesis says the mean is same for all groups mean1=mean2=mean3=mean4=mean5=mean6
# The alternate hypthosis means the mean is not same or at least one of the mean is different


#The null hypothesis is rejected as the alternative hypothesis has a p-value <0.05 which is <2e-16
# The details are below

# Df Sum Sq Mean Sq F value Pr(>F)
# Continent     5   6819  1363.9   59.68 <2e-16 ***
#  Residuals   211   4822    22.9


#reference: http://www.sthda.com/english/wiki/one-way-anova-test-in-r
# reference: https://www.guru99.com/r-anova-tutorial.html

# Multiple pairwise-comparison between the means of groups
#In one-way ANOVA test, a significant p-value indicates that some means value of group is different, but we don’t know which pairs of groups are different.
#It’s possible to perform multiple pairwise-comparison, to determine if the mean difference between specific pairs of group are statistically significant.

#Tukey multiple pairwise-comparisons
# Tukey HSD (Tukey Honest Significant Differences, R function: TukeyHSD()) used for performing pairwise comparison between means of different groups
TukeyHSD(aov.model)


#It can be seen from the output, that only the difference between
# Europe-Asia, Europe-Australia/Oceania, North America-Europe,  are significant with an
#adjusted p-value of 0.0000119, 0.0000425, 0.0210795 respectively.

print(model.tables(aov.model,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(SP.DYN.LE00.IN~Continent,data=one_way_Anova_model)        #graphical summary
```

```
#------------------------------------------------------------------------------#
```