Statistical Learning

# Statistical Modeling of Life Expectancy Data

**Francesco Ansuini - 2085702,**
**Mikhail Kolobov - 2072001,**
**Riccardo Russo - 2087618**

Department of Mathematics, Università degli Studi di Padova

Date: May 25, 2023

# Project Presentation

This project aims at performing a regression analysis pointing at understanding what are the factors that mostly influence **Life Expectancy** in a certain country. This investigation will consider several variables, each related to a number of countries, each considered in different years. Life expectancy is intended as the average number of years a person is expected to live based on the statistical analysis of a population within a specific geographic area. Life expectancy can vary significantly between different countries and is known to be influenced by numerous factors, including healthcare access, quality of healthcare, nutrition, lifestyle factors, socioeconomic conditions, education, and public health measures. Life expectancy is often used as an indicator of a country's overall health status and the effectiveness of its healthcare system. In this analysis, some of these factors will be taken into account, with the aim of performing a **linear regression analysis** to predict the life expectancy of a given country showing certain levels of such factors. As far as the methods are concerned, the multiple linear regression will be investigated, together with its regularized variants, namely Ridge regression and Lasso regression.

# Data Presentation

The dataset chosen for this analysis contains information about the life expectancy and other variables for almost all countries around the world, from the year 2000 to 2015. The dataset contains the following variables:

| Variable | Description |
|---|---|
| Country | The country of reference |
| Region | Categorical variable: macro-area where the country belongs |
| Year | The year of reference |
| Status | Categorical variable: Developed or Developing country |
| Life Expectancy | Life expectancy in years |
| Adult Mortality | Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population) |
| Infant Deaths | Number of Infant Deaths per 1000 population |
| Alcohol Consumption | Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol) |
| Hepatitis B | Hepatitis B (HepB) immunization coverage among 1-year-olds (%) |
| Polio | Polio (Pol3) immunization coverage among 1-year-olds (%) |
| Diphtheria | Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%) |
| Measles | Number of reported cases |
| BMI | Average Body Mass Index of entire population |
| <5 Deaths | Number of under-five deaths per 1000 population |
| Incidents of HIV | Deaths per 1 000 live births HIV/AIDS (0-4 years) |

| Variable | Description |
|---|---|
| GDP per Capita | Gross Domestic Product per capita (in USD) |
| Population | Population of the country in millions |
| Thinness 5-9 years | Prevalence of thinness among children for Age 5 to 9(%) |
| Thinness 10-19 years | Prevalence of thinness among children and adolescents for Age 10 to 19 (%) |
| Schooling | Number of years of School (years) |

In our analysis, **Life Expectancy** will be kept as the dependent variable, while the other variables will be considered as regressors. Notice however that not every single variable described above is useful in our analysis, and some of them have to be discarded a-priori during the pre-processing phase. There are also some variables that are intuitively highly correlated, like the three regressors related to immunization coverage (Hepatitis B, Polio, Diphteria). This correlation can create some issues during the modeling phase, so we might need to get rid of them. However, instead of discarding them straight away, an appropriate analysis will be carried out first. In terms of code, we start off by loading the necessary libraries for the subsequent analysis, and then load the dataset from the file *New_dataset_LifeExpectancy.csv*:

```r
library(dplyr)
library(tidyr)
library(readxl)
library(psych)
library(corrplot)
library(car)
library(leaps)
library(glmnet)
library(readr)
library(ggplot2)
library(ggcorrplot)
library(cowplot)
library(gridExtra)
library(grid)

data <- read_csv("New_dataset_LifeExpectancy.csv",col_types = cols(X1 = col_skip()))
```

# Data Preprocessing

In the initial dataset, named “data” some variables had missing values, namely Adult_mortality, GDP_per_capita, Hepatitis_B, and Schooling. To address these missing values, median imputation was performed. This method involved replacing the missing values with the median values of the corresponding country across different years. By using the median, we aimed to minimize the potential influence of outliers on the imputed values.

```r
nna <- sum(is.na(data$Adult_mortality))
cat("Adult Mortality presents", nna, "missing values", sep = " ")
```

```
Adult Mortality presents 236 missing values
```

```
nna <- sum(is.na(data$GDP_per_capita))
cat("GDP per Capita presents", nna, "missing values", sep = " ")
```

GDP per Capita presents 182 missing values

```
nna <- sum(is.na(data$Hepatitis_B))
cat("Hepatitis B presents", nna, "missing values", sep = " ")
```

Hepatitis B presents 311 missing values

```
nna <- sum(is.na(data$Schooling))
cat("Schooling presents", nna, "missing values", sep = " ")
```

Schooling presents 285 missing values

```
countries <- unique(data$Country[is.na(data$Adult_mortality)])
for (c in countries) {
  x <- data$Adult_mortality[data$Country == c]
  x[is.na(x)] <- median(x, na.rm = T)
  data$Adult_mortality[data$Country == c] <- x
}

countries <- unique(data$Country[is.na(data$GDP_per_capita)])
for (c in countries) {
  x <- data$GDP_per_capita[data$Country == c]
  x[is.na(x)] <- median(x, na.rm = T)
  data$GDP_per_capita[data$Country == c] <- x
}

countries <- unique(data$Country[is.na(data$Hepatitis_B)])
for (c in countries) {
  x <- data$Hepatitis_B[data$Country == c]
  x[is.na(x)] <- median(x, na.rm = T)
  data$Hepatitis_B[data$Country == c] <- x
}

countries <- unique(data$Country[is.na(data$Schooling)])
for (c in countries) {
  x <- data$Schooling[data$Country == c]
  x[is.na(x)] <- median(x, na.rm = T)
  data$Schooling[data$Country == c] <- x
}
```

The initial dataset was merged with the “measles” dataset based on the common variables “Year” and “Country.” As the previous values in the “measles” column seemed relatively constant, which is unlikely considering the nature of the variable itself, they were replaced with the “real values” obtained from the World Health Organization website (contained in the file *Measles.csv*) . This merged dataset, referred to as “life” served as the foundation for subsequent analysis.

```
measles <- read_delim("Measles.csv", ";", escape_double = FALSE, trim_ws = TRUE)
colnames(measles)[1] <- "Country"
life <- merge(data, measles, by = c("Year", "Country"))
```

To streamline the analysis, certain columns were removed from the dataset, including the old measles column, which has been replaced, year, and country. The focus was primarily on the geographical region of each country. Additionally, a new column named “Status” was created in order to get rid of the original one-hot encoding of the “Status” variable, which required two columns named “Economy_status_Developed” and “Economy_status_Developing”

```
life <- life[,-9] # Remove old Measles
life2 <- life[,-c(1,2)] # Remove country and year
life2$Region <- as.factor(life2$Region)
temp <- life2$Economy_status_Developed - life2$Economy_status_Developing
temp[temp == 1] = "Developed"
temp[temp == -1] = "Developing"
life2$Status <- as.factor(temp)
life2 <- life2[,-c(16,17)]
```

Lastly, it was observed that the “MeaslesCases” column represented the count of cases, while all other variables were expressed as ratios. To ensure consistency, the “MeaslesCases” values were adjusted by dividing them by the corresponding “Population_mln” column, resulting in measles cases per million people.

Finally, before stepping into the exploratory data analysis, we print a summary of our dataset, in order to get a hint of the scale and location of each single variable.

```
life2$MeaslesCases <- life2$MeaslesCases/life2$Population_mln

summary(life2)
```

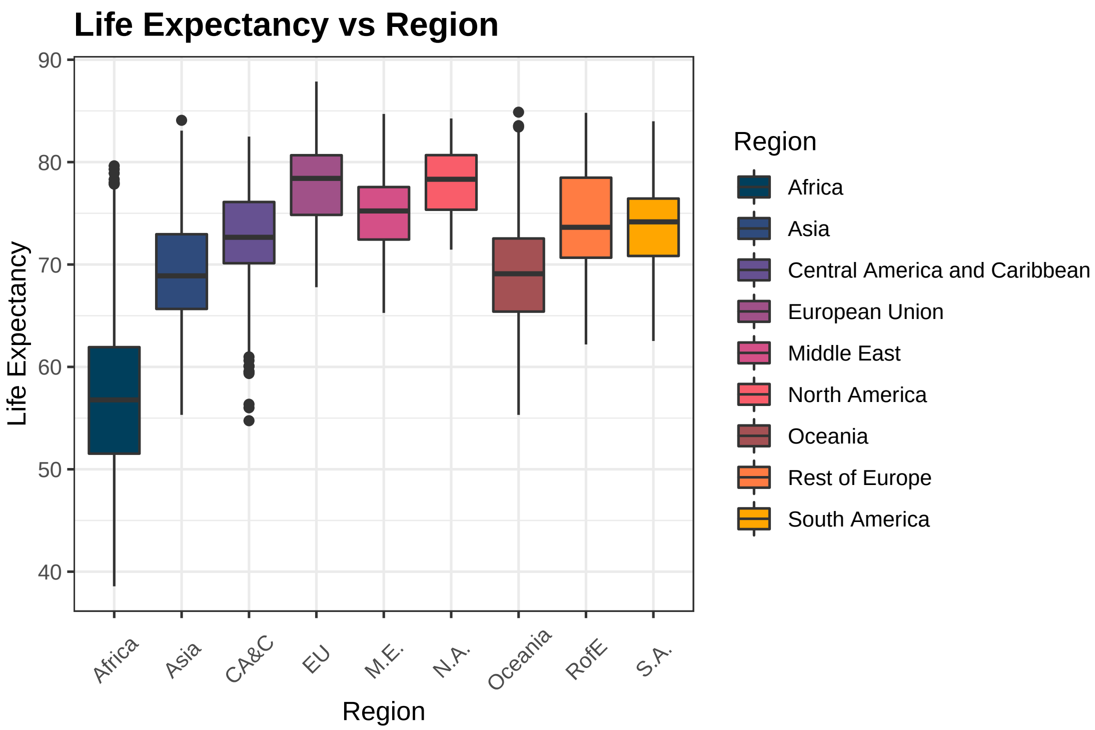
```
##                           Region    Infant_deaths    Under_five_deaths
##  Africa                       :697   Min.   :  1.80   Min.   :  2.30  
##  European Union               :379   1st Qu.:  7.90   1st Qu.:  9.40  
##  Asia                         :372   Median : 20.00   Median : 23.35  
##  Central America and Caribbean:253   Mean   : 30.72   Mean   : 43.82  
##  Rest of Europe               :187   3rd Qu.: 48.80   3rd Qu.: 68.90  
##  South America                :160   Max.   :138.10   Max.   :224.90  
##  (Other)                      :324                                    
##  Adult_mortality  Alcohol_consumption  Hepatitis_B         BMI       
##  Min.   : 49.38   Min.   : 0.000      Min.   :14.00   Min.   :19.80  
##  1st Qu.:103.76   1st Qu.: 1.170      1st Qu.:79.00   1st Qu.:23.20  
##  Median :163.46   Median : 4.015      Median :89.00   Median :25.50  
##  Mean   :193.83   Mean   : 4.781      Mean   :84.12   Mean   :24.93  
##  3rd Qu.:251.58   3rd Qu.: 7.660      3rd Qu.:96.00   3rd Qu.:26.30  
##  Max.   :719.36   Max.   :17.870      Max.   :99.00   Max.   :32.00  
##                                                                      
##      Polio        Diphtheria    Incidents_HIV     GDP_per_capita  
##  Min.   : 8.0   Min.   :16.0   Min.   : 0.0100   Min.   :   148  
##  1st Qu.:81.0   1st Qu.:81.0   1st Qu.: 0.0800   1st Qu.:  1328  
##  Median :93.0   Median :93.0   Median : 0.1500   Median :  4275  
##  Mean   :86.3   Mean   :86.1   Mean   : 0.9723   Mean   : 11352  
##  3rd Qu.:97.0   3rd Qu.:97.0   3rd Qu.: 0.5100   3rd Qu.: 12389  
##  Max.   :99.0   Max.   :99.0   Max.   :21.6800   Max.   :112418  
##                                                                   
##  Population_mln     Thinness_ten_nineteen_years Thinness_five_nine_years
##  Min.   :   0.080   Min.   : 0.100              Min.   : 0.100          
##  1st Qu.:   2.270   1st Qu.: 1.700              1st Qu.: 1.700          
##  Median :   8.045   Median : 3.400              Median : 3.400          
##  Mean   :  37.636   Mean   : 4.938              Mean   : 4.986
##  3rd Qu.:  20.925   3rd Qu.: 7.200              3rd Qu.: 7.300
##  Max.   :1379.860   Max.   :27.700              Max.   :28.600
##
##    Schooling      Life_expectancy  MeaslesCases           Status
##  Min.   : 1.100   Min.   :38.57   Min.   :   0.000   Developed : 498
##  1st Qu.: 4.800   1st Qu.:62.22   1st Qu.:   0.000   Developing:1874
##  Median : 7.800   Median :71.09   Median :   3.043
##  Mean   : 7.584   Mean   :68.79   Mean   : 101.531
##  3rd Qu.:10.300   3rd Qu.:76.11   3rd Qu.:  31.272
##  Max.   :14.100   Max.   :87.87   Max.   :8164.512
##
```

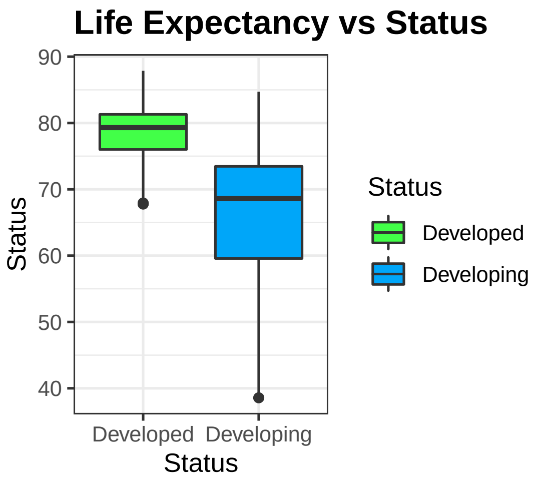
# Exploratory Data Analysis

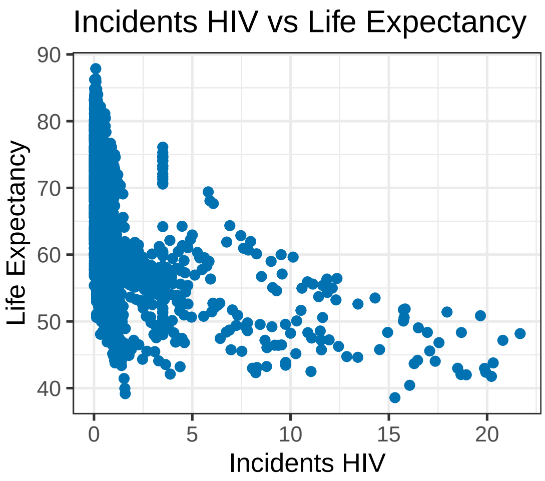
In this preliminary part of the analysis, we seek to explore the features of each variable in our dataset, and also their mutual relationship. This exploratory data analysis can be useful to understand what are the variables that might play a crucial role during the analysis, as well as the ones that could be unnecessary. Moreover, it can help identifying whether there are some strong correlations between variables, which could generate multicollinearity between regressors in our following models and leading to all the issues related to it, like biased coefficient estimators.

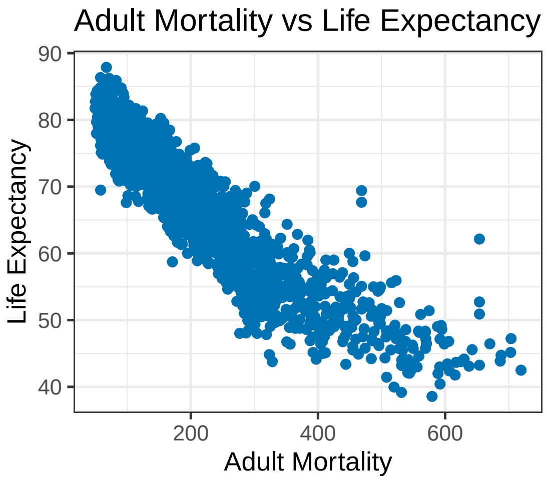
In particular, we first investigate the interaction between Life_expectancy and some variables that intuitively might be correlated with it.

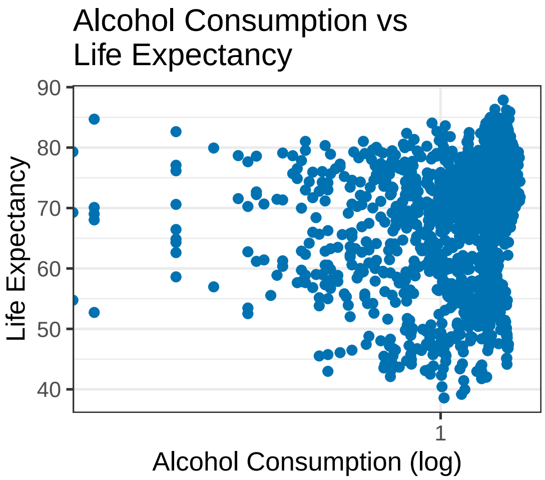
```r
my_colors <- c("#003f5c","#2f4b7c","#665191", "#a05188",
                        "#d45087", "#f95d6a","#a45154" , "#ff7c43",  "#ffa600")
v <- levels(life2$Region)
life2.1  <- life2
life2.1$Region <- factor(life2.1$Region, levels = v,
                         labels = c("Africa", "Asia", "CA&C", "EU", "M.E.",
                                    "N.A.", "Oceania", "RofE", "S.A."))
reg <- ggplot(life2.1, aes(x = Region, y = Life_expectancy, fill = Region)) +
  geom_boxplot() +
  ggtitle("Life Expectancy vs Region") +
  scale_fill_manual(values = my_colors, labels = v) + # use custom colors
  theme_bw() +
  theme(plot.title = element_text(size = 14, face = "bold"),
        axis.text.x = element_text(angle = 45, vjust = 0.5)) +
  labs(y = "Life Expectancy")


my_colors <- c("#42ff48","#00a6f9")
stat <- ggplot(life2, aes(x = Status, y = Life_expectancy, fill = Status)) +
  geom_boxplot() +
  ggtitle("Life Expectancy vs Status") +
  scale_fill_manual(values = my_colors) + # use custom colors
  theme_bw() +
  theme(plot.title = element_text(size = 14, face = "bold")) +
  labs(y = "Status")


ad_m <- ggplot(life2, aes(x = (Adult_mortality), y = Life_expectancy)) +
  ggtitle("Adult Mortality vs Life Expectancy") +
  geom_point(color = "#0072B2") +
  labs(x = "Adult Mortality", y = "Life Expectancy") +
  theme_bw()


alc <- ggplot(life2, aes(x = log(Alcohol_consumption), y = Life_expectancy)) +
  ggtitle("Alcohol Consumption vs \nLife Expectancy") +
  geom_point(color = "#0072B2") +
  scale_x_continuous(trans = "log10", breaks = c(1, 10, 100)) +
  labs(x = "Alcohol Consumption (log)", y = "Life Expectancy") +
  theme_bw()


hiv <- ggplot(life2, aes(x = (Incidents_HIV), y = Life_expectancy)) +
  ggtitle("Incidents HIV vs Life Expectancy") +
  geom_point(color = "#0072B2") +
  labs(x = "Incidents HIV", y = "Life Expectancy") +
  theme_bw()

lay <- rbind(c(1,1),
             c(1,1),
             c(1,1),
             c(2,3),
             c(2,3),
             c(4,5),
             c(4,5))
grid.arrange(reg,stat,hiv,ad_m,alc, layout_matrix = lay)
```


Life Expectancy vs Region
Life Expectancy
90
80
70
60
50
40
Africa
Asia
CA&C
EU
M.E.
N.A.
Oceania
RofE
S.A.
Region
Region
Africa
Asia
Central America and Caribbean
European Union
Middle East
North America
Oceania
Rest of Europe
South America


Life Expectancy vs Status
Status
90
80
70
60
50
40
Developed
Developing
Status
Status
Developed
Developing


Incidents HIV vs Life Expectancy
Life Expectancy
90
80
70
60
50
40
0
5
10
15
20
Incidents HIV


Adult Mortality vs Life Expectancy
Life Expectancy
90
80
70
60
50
40
200
400
600
Adult Mortality


Alcohol Consumption vs
Life Expectancy
Life Expectancy
90
80
70
60
50
40
1
Alcohol Consumption (log)

Starting from Life Expectancy vs Region, we can clearly see that the regions are all similar in terms of Life Expectancy, with some of them showing a larger variance. The region that is the most different from the others is Africa, which presents a boxplot that is centered in a much lower value compared to the others. This is due to the health conditions that the majority of African countries experience. However this is also the region that displays the largest variance, hinting at the fact that health conditions in the whole region of Africa can vary greatly from one country to another. In the time span from 2000 to 2015, the country in the Africa region showing the highest Life Expectancy is Tunisia in 2015, while the one showing the lowest life expectancy is Lesotho in 2007.

```
i <- which.max(life$Life_expectancy[life$Region == "Africa"])
c <- life$Country[life$Region == "Africa"][i]
y <- life$Year[life$Region == "Africa"][i]
l <- life$Life_expectancy[life$Region == "Africa"][i]
cat("African country with highest Life Expectancy
    between 2000 and 2015 is \n", c,"in",y,":", l)
```

```
## African country with highest Life Expectancy
##     between 2000 and 2015 is
##  Tunisia in 2015 : 79.63802
```

```
i <- which.min(life$Life_expectancy[life$Region == "Africa"])
c <- life$Country[life$Region == "Africa"][i]
y <- life$Year[life$Region == "Africa"][i]
l <- life$Life_expectancy[life$Region == "Africa"][i]
cat("African country with lowest Life Expectancy
    between 2000 and 2015 is \n", c,"in",y,":", l)
```

```
## African country with lowest Life Expectancy
##     between 2000 and 2015 is
##  Lesotho in 2007 : 38.57091
```

The relationship between Life Expectancy and the classification of countries as Developed or Developing is evident when examining the boxplot. The plot clearly depicts a noticeable distinction in the distribution of life expectancies between these two categories. Developed countries are represented by a smaller number of data points (498), while developing countries exhibit a larger number (1874). Additionally, the lower variance observed in the life expectancy of developed countries compared to developing ones can be attributed to several factors. Developed countries typically have well-established healthcare systems, higher standards of living, and greater access to quality education and resources. These factors contribute to more consistent and predictable life expectancy outcomes, resulting in a narrower range of values and lower variance.

As far as the comparison between numerical variables is concerned, lets analyse how Life Expectancy behaves with respect to some of the variables in our dataset that might intuitively be influencing it.

In terms of Incidents HIV against Life Expectancy, there appears to be a moderate linear correlation between them, but we will need to analyze the data further to determine if this variable should be kept. This examination will be done during modeling, more specifically in the variable selection phase.

```
c <- cor(life2$Incidents_HIV, life2$Life_expectancy)
cat("Correlation between Life Expectancy and Incidents HIV is:", c)
```

```
## Correlation between Life Expectancy and Incidents HIV is: -0.5464225
```

The analysis of the relationship between Adult Mortality and Life Expectancy reveals a strong linear correlation between these two variables. This observation suggests that Adult Mortality holds significant informational value for our research endeavors. Consequently, we will include Adult Mortality as a crucial feature in all of our models.

```
c <- cor(life2$Adult_mortality, life2$Life_expectancy)
cat("Correlation between Life Expectancy and Adult Mortality is:", c)
```

```
## Correlation between Life Expectancy and Adult Mortality is: -0.9181206
```

As we can see in the last plot, there is no clear linear relationship between life expectancy and alcohol consumption. We originally believed that higher alcohol consumption would lead to lower life expectancy, but this hypothesis is not supported by the data. It is possible that the regions with higher alcohol consumption also have higher life expectancy. One plausible explanation for this is that the regions where there is a greater alcohol consumption are also the more developed regions. This leads both to an improved healthcare system, leading to a higher life expectancy, but also to an easier access and consumption of alcoholic beverages.

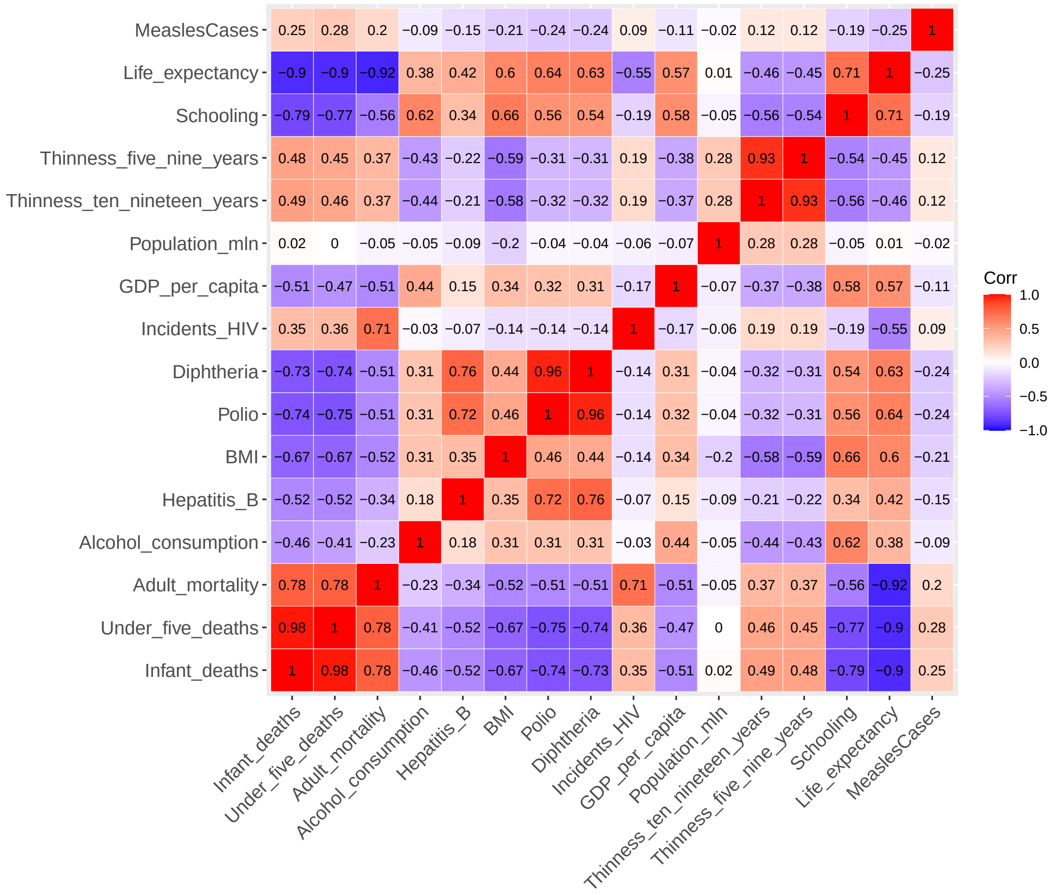
```
c <- cor(life2$Alcohol_consumption, life2$Life_expectancy)
cat("Correlation between Life Expectancy and Alcohol Consumption is:", c)
```

```
## Correlation between Life Expectancy and Alcohol Consumption is: 0.3804203
```

Upon examining the correlation matrix, it becomes apparent that several variables exhibit strong correlations with each other. The presence of such high correlations raises the question of the likelihood of multicollinearity among these variables.

Multicollinearity refers to the condition where two or more independent variables in a regression model are highly correlated with each other. This can pose challenges in statistical analysis, as it makes it difficult to ascertain the individual contributions of each variable and may affect the stability and interpretability of the model.

```
corr_mat <- round(cor(subset(life2, select = -c(Region, Status) )), 3)
ggcorrplot(corr_mat,
           outline.color = "white", lab = TRUE,
           lab_size = 3, ggtheme = ggplot2::theme_gray)
```



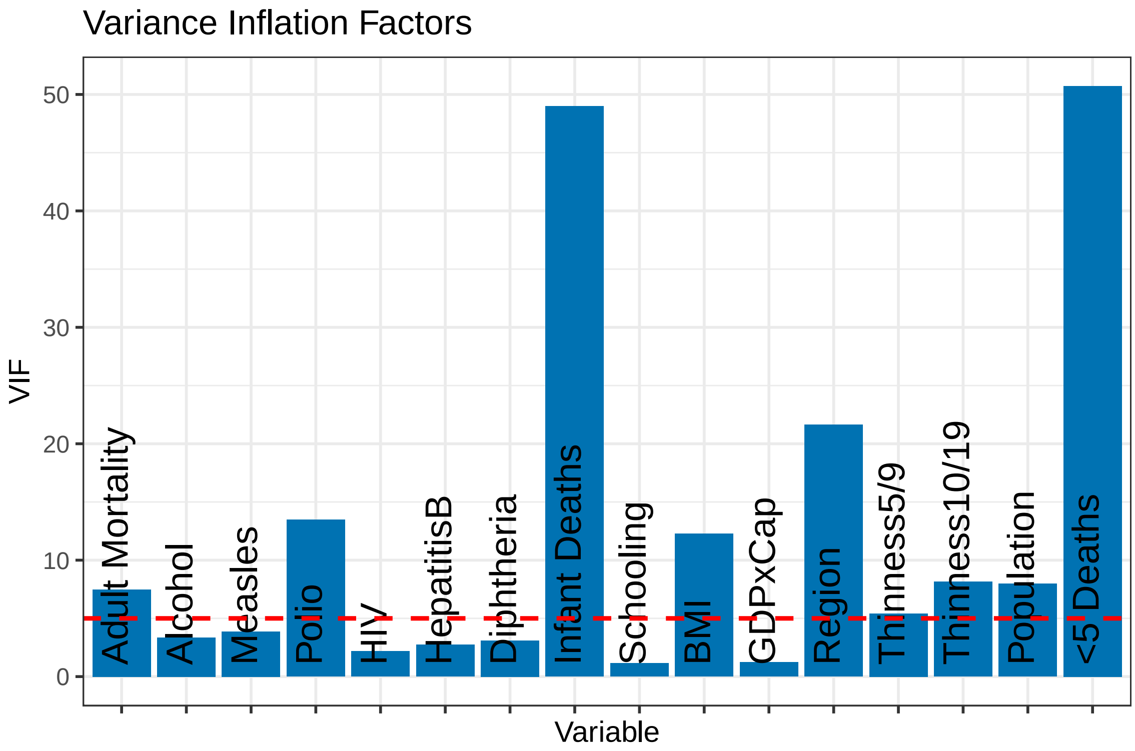
The high correlation observed between Polio and Diphtheria can be attributed to the common practice of administering vaccines against these diseases together. It is common for individuals to receive vaccinations for both Polio and Diphtheria simultaneously, leading to a strong association between the two variables in the correlation analysis.

## Variance Inflation Factors Analysis

Variance Inflation Factor (VIF) is a measure used to assess multicollinearity in regression analysis. The VIF quantifies the extent to which the variance of the estimated regression coefficient of a particular predictor variable is inflated due to multicollinearity. In particular, it assesses the impact of multicollinearity by examining how much the variance of the estimated coefficient increases when including all the predictor variables compared to when the predictor variable is considered alone.

Researchers often consider a threshold value for VIF, such as 5 or 10, to determine if multicollinearity is a concern. If the VIF exceeds the chosen threshold, it suggests a high degree of multicollinearity, and it may be necessary to address the issue by directly acting on the regressor itself, either by transforming it or removing it.

```r
v_temp <- c("Region","Infant Deaths","<5 Deaths","Adult Mortality",
          "Alcohol","HepatitisB","Measles","BMI","Polio","Diphtheria",
          "HIV","GDPxCap","Population","Thinness10/19","Thinness5/9","Schooling")

mod1 <- lm(Life_expectancy~ ., data = subset(life2, select =-c(Status)))
vif1 <- data.frame("Variable" =  names(vif(mod1)[,1]), "Vif" = vif(mod1)[,1],
                   "VarNames" = v_temp,row.names = NULL)

ggplot(vif1, aes(x = Variable, y = Vif)) +
 geom_bar(stat = "identity", fill = "#0072B2") +
 geom_text(aes(label = VarNames, y = 1),
           vjust = -0.5, angle = 90, hjust = 0, size = 5, nudge_x = 0.3) +
 labs(x = "Variable", y = "VIF", title = "Variance Inflation Factors") +
 theme_bw()+
 theme(axis.text.x = element_blank())+
 geom_hline(yintercept = 5, linetype = "dashed", color = "red", size = 0.8)
```



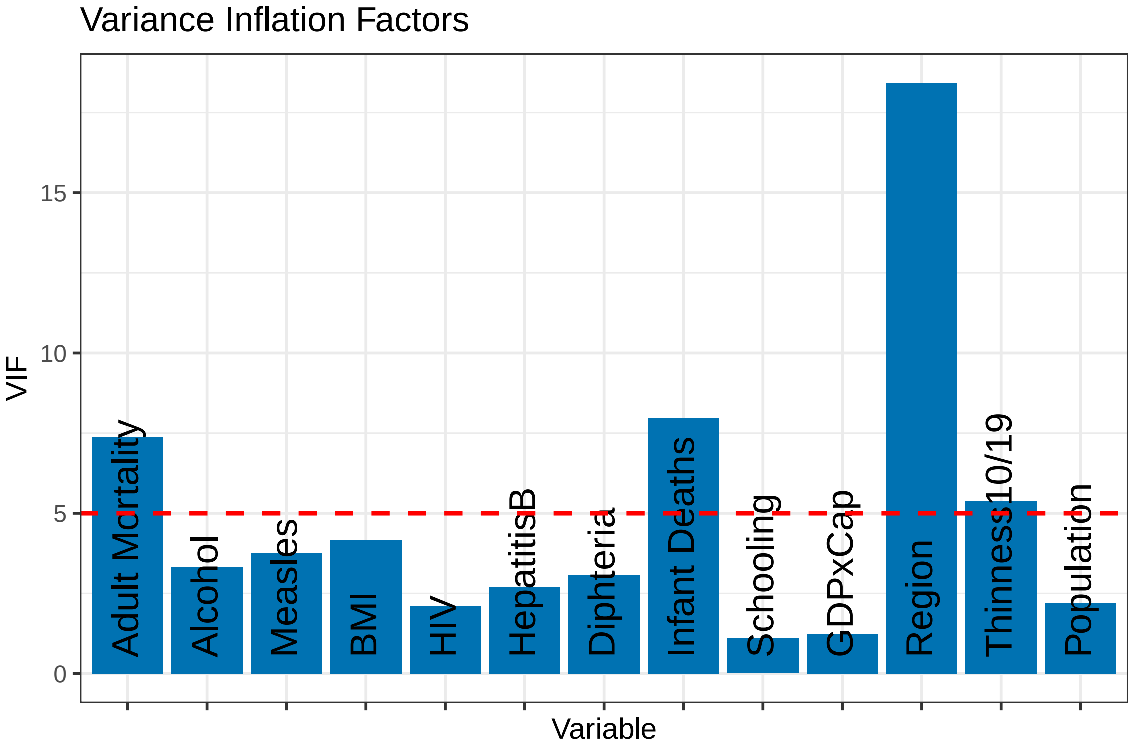
After carefully analysing the Variance Inflation Factor (VIF) plots, we have made the decision to remove some of the features of our dataset.

In particular we started by dropping the Under_five_death column, due to its high correlation with Infant_deaths. This is because they are clearly correlated, as they convey basically the same information.

Despite the higher Variance Inflation Factor (VIF) value, we have decided to retain the BMI variable in our analysis to investigate its potential impact on life expectancy. While the higher VIF suggests some degree of multicollinearity with other variables, we believe that BMI might still provide informative insights into the relationship between a measure of obesity with life expectancy.

After careful evaluation, we have decided to remove thinness_5_9 from our analysis, due to its great correlation with thinness_10_19. These variables, which represent the prevalence of thinness among children aged 5-9 and 10-19 respectively, exhibit a high degree of similarity, and therefore one should be dropped from the model to avoid having regressors that are too correlated.

```
v_temp <- c("Region","Infant Deaths","Adult Mortality",
            "Alcohol","HepatitisB","Measles","BMI","Diphteria",
            "HIV","GDPxCap","Population","Thinness10/19","Schooling")

mod2 <- lm(Life_expectancy~ .,
           data = subset(life2, select =-c(Status, Under_five_deaths,
                                            Polio,Thinness_five_nine_years)))
vif2 <- data.frame("Variable" =  names(vif(mod2)[,1]), "Vif" = vif(mod2)[,1],
                   "VarNames" = v_temp,row.names = NULL)

ggplot(vif2, aes(x = Variable, y = Vif)) +
  geom_bar(stat = "identity", fill = "#0072B2") +
  geom_text(aes(label = VarNames, y = 0.5),
            vjust = -0.5, angle = 90, hjust = 0, size = 5, nudge_x = 0.3) +
  labs(x = "Variable", y = "VIF", title = "Variance Inflation Factors") +
  theme_bw()+
  theme(axis.text.x = element_blank())+
  geom_hline(yintercept = 5, linetype = "dashed", color = "red", size = 0.8)
```



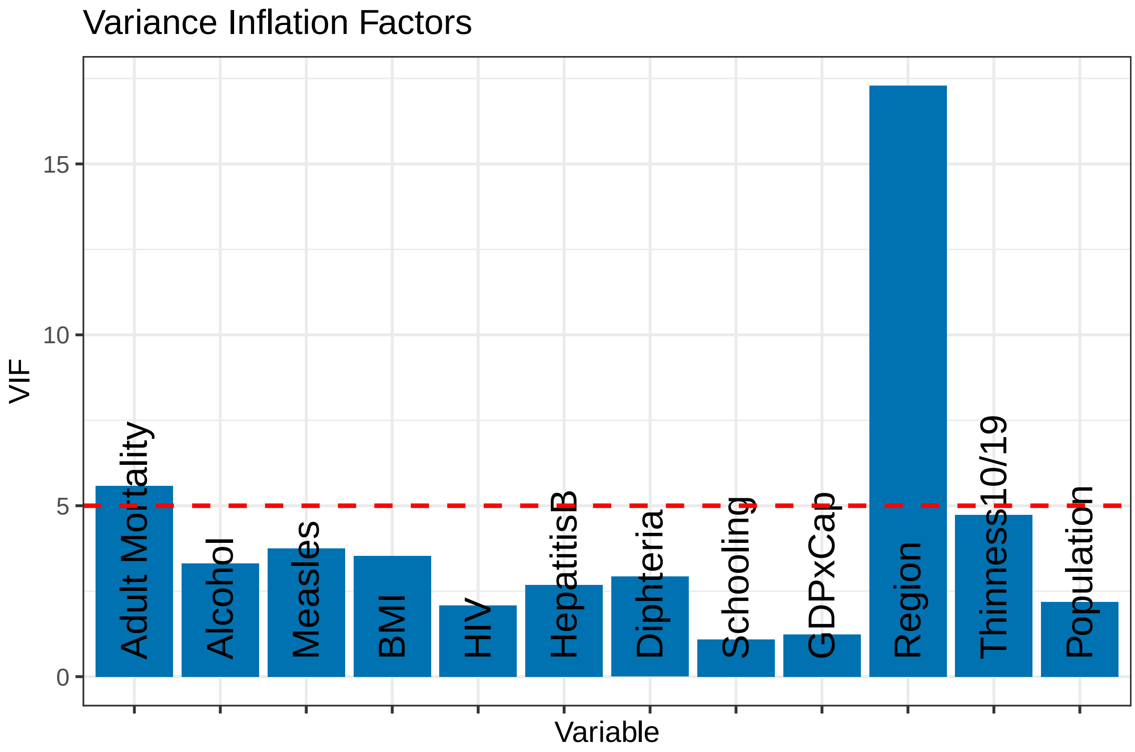
```
c <- cor(life2$Adult_mortality, life2$Infant_deaths)
cat("Correlation between Adult Mortality and Infant Deaths is:", c)
```

```
## Correlation between Adult Mortality and Infant Deaths is: 0.7781972
```

Given the high correlation coefficient of 0.78 between infant_deaths and adult_mortality, we have made the decision to remove the infant_deaths variable from our analysis as well. Such a strong correlation suggests a substantial relationship between these two variables.

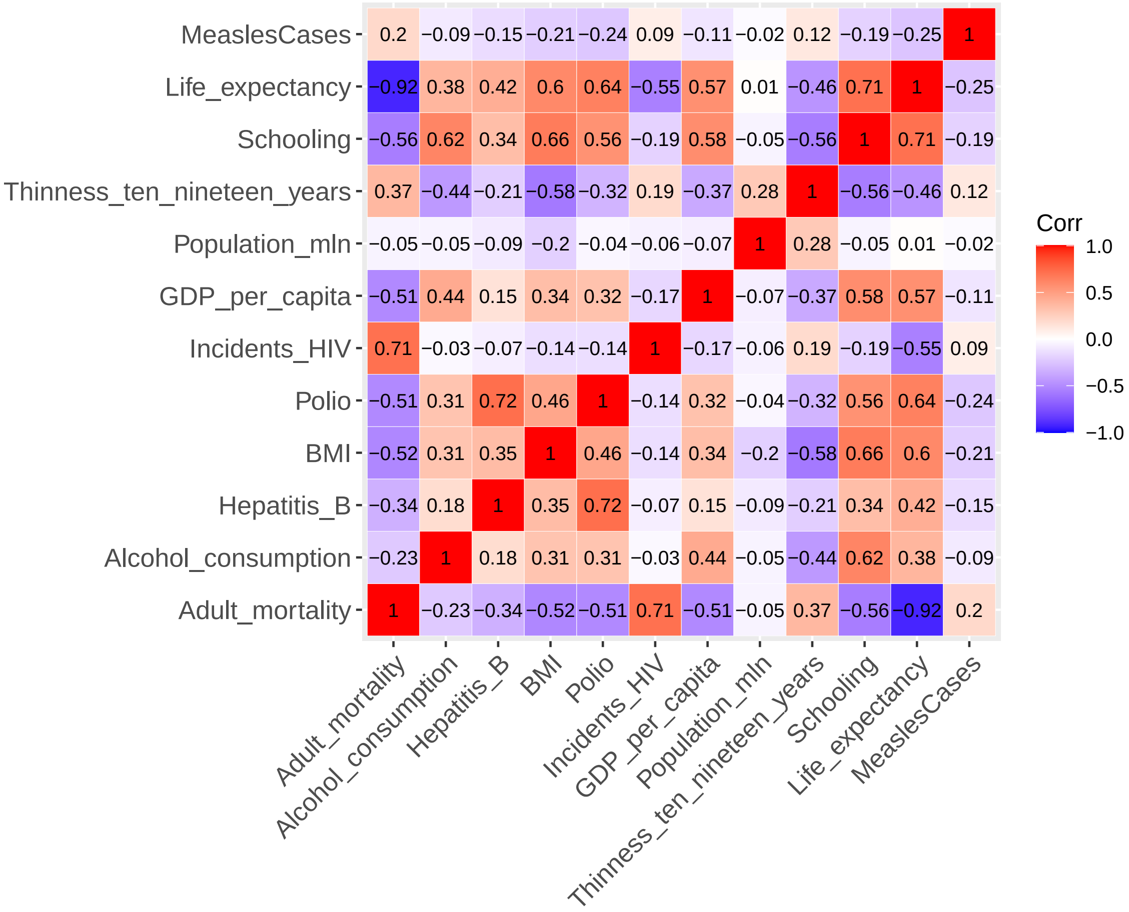
These are the final VIF's for our linear model, after analysing and discarding the variables that convey similar information. Although the "region" variable has a high variance inflation factor, we decided to keep it because we believe that it is interesting to compare life expectancy among different regions. However, in the following section we will investigate whether every level of the "Region" categorical variable should be kept, or if some of them could be merged together.

```
v_temp <- c("Region","Adult Mortality",
            "Alcohol","HepatitisB","Measles","BMI","Diphteria",
            "HIV","GDPxCap","Population","Thinness10/19","Schooling")
mod3 <- lm(Life_expectancy~ .,
           data = subset(life2, select =-c(Infant_deaths, Status,
                                           Under_five_deaths, Polio,
                                           Thinness_five_nine_years)))
vif3 <- data.frame("Variable" =  names(vif(mod3)[,1]), "Vif" = vif(mod3)[,1],
                   "VarNames" = v_temp,row.names = NULL)

ggplot(vif3, aes(x = Variable, y = Vif)) +
  geom_bar(stat = "identity", fill = "#0072B2") +
  geom_text(aes(label = VarNames, y = 0.5),
            vjust = -0.5, angle = 90, hjust = 0, size = 5, nudge_x = 0.3) +
  labs(x = "Variable", y = "VIF", title = "Variance Inflation Factors") +
  theme_bw()+
  theme(axis.text.x = element_blank())+
  geom_hline(yintercept = 5, linetype = "dashed", color = "red", size = 0.8)
```



The final correlation matrix is presented as follows. This shows no signs of deep correlations in our regressors, so we should have eliminated similar variables from our dataset.

```
corr_mat2 <- round(cor(subset(life2,
                              select = -c(Infant_deaths, Region,
                                          Status, Under_five_deaths, Diphtheria,
                                          Thinness_five_nine_years) )), 3)
ggcorrplot(corr_mat2,
           outline.color = "white", lab = TRUE,
           lab_size = 3, ggtheme = ggplot2::theme_gray)
```



# Data Modeling

The next step involves conducting a regression analysis on the provided data.

To begin, our initial approach involved fitting a multiple linear regression model to assess the compatibility of linearity with our dataset. By doing so, we can investigate the linear relationships and potential associations between these variables.

After fitting the multiple linear regression model, we proceeded to evaluate its performance and validity. This evaluation included examining various statistical metrics such as the coefficient of determination (R-squared), significance of individual regression coefficients, and the overall goodness of fit of the model. These analyses provided insights into the extent to which a linear relationship exists between the selected independent variables and the dependent variable.

Following the assessment of the multiple linear regression model, we expanded our analysis to explore regularized variants of regression models. Regularization is a technique that can enhance the generalization ability of a model by addressing issues such as overfitting and high variance. Regularized regression models add a penalty term to the standard regression objective function, which helps to control the complexity of the model and prevent it from fitting noise or irrelevant patterns in the data.

The regularized variants we considered are Ridge regression and Lasso regression. Ridge regression introduces a penalty term to the least squares objective function, which shrinks the regression coefficients towards zero and reduces their variance. Lasso regression, on the other hand, not only shrinks the coefficients but also performs variable selection by driving some coefficients exactly to zero, effectively eliminating those variables from the model.

By exploring these regularized variants, we aimed to determine whether regularization could improve the generalization capability of our regression model. We evaluated the performance of the regularized models using appropriate metrics and compared them to the results obtained from the multiple linear regression model. This analysis provided valuable insights into the potential benefits of incorporating regularization techniques in our regression analysis.

## Multiple Linear Regression

We initially created several linear models, one for each possible reference category, to determine whether we could combine two or more regions and simplify the model. After careful consideration, we decided to merge Central America and the Caribbean with South America, as well as Asia and Oceania. This is because they showed no significant difference, and because they are both geographically close and have similar healthcare systems. We used Africa as the reference category in the end.

```
life3 <- subset(life2, select = -c(Infant_deaths, Under_five_deaths,
                                   Polio, Thinness_five_nine_years))

# Let's put together Asia + Oceania and Central America and Caribbean + South America

life3$Region <- as.character(life3$Region)

i <- life3$Region == "Asia" | life3$Region == "Oceania"
life3$Region[i] <- "AsiaOceania"
i <- life3$Region == "Central America and Caribbean" | life3$Region == "South America"
life3$Region[i] <- "South America"

life3$Region <- as.factor(life3$Region)

life3$Region <- relevel(life3$Region, ref= "Africa")

full_model <- lm(Life_expectancy~ ., data = life3)
s <- summary(full_model)
```

Now let's check the summary for this full model.

```
print(s)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ ., data = life3)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9.5974 -1.8676  0.0464  1.7302 21.3549 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  7.079e+01  1.325e+00  53.424  < 2e-16 ***
## RegionAsiaOceania            7.271e-01  2.200e-01   3.306 0.000962 ***
## RegionEuropean Union         5.816e+00  3.944e-01  14.749  < 2e-16 ***
## RegionMiddle East            1.320e+00  3.336e-01   3.958 7.78e-05 ***
## RegionNorth America          4.837e+00  5.808e-01   8.328 3.24e-05 ***
## RegionRest of Europe         1.417e+00  3.115e-01   4.548 5.69e-06 ***
## RegionSouth America          3.232e+00  2.408e-01  13.421  < 2e-16 ***
## Adult_mortality             -5.979e-02  1.147e-03 -52.118  < 2e-16 ***
## Alcohol_consumption          1.498e-02  2.657e-02   0.564 0.572944    
## Hepatitis_B                 -4.165e-03  5.876e-03  -0.709 0.478518    
## BMI                         -3.810e-02  4.888e-02  -0.779 0.435805    
## Diphtheria                   7.944e-02  6.855e-03  11.589  < 2e-16 ***
## Incidents_HIV               -2.289e-01  3.839e-02  -5.963 2.85e-09 ***
## GDP_per_capita               1.932e-05  5.523e-06   3.499 0.000476 ***
## Population_mln               1.988e-05  4.326e-04   0.046 0.963343    
## Thinness_ten_nineteen_years  1.427e-02  1.827e-02   0.781 0.434898    
## Schooling                    4.807e-01  3.790e-02  12.683  < 2e-16 ***
## MeaslesCases                -4.926e-04  1.377e-04  -3.578 0.000353 ***
## StatusDeveloping            -1.540e+00  3.753e-01  -4.103 4.21e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 2.8 on 2353 degrees of freedom
## Multiple R-squared:   0.92,  Adjusted R-squared:  0.9194 
## F-statistic:  1504 on 18 and 2353 DF,  p-value: < 2.2e-16
```

Through the construction of the complete model, we are able to discern the variables with the least significance by examining their corresponding p-values. Subsequently, we can eliminate these variables and iterate this process until only the statistically significant variables remain. The threshold for the p-value that was chosen is the usual 5%. The sequence of the least significant variables, along with their respective p-values, is as follows: Population_mln (p-value 0.96), alcohol_consumption (p-value 0.57), Hepatitis_B (p-value 0.49), Thinness_ten_nineteen_years (p-value 0.41), and BMI (p-value 0.2).

```
life4 <- subset(life3, select = -c(Population_mln))
model3 <- lm(Life_expectancy~ ., data = life4)
# summary(model3)


life5 <- subset(life4, select = -c(Alcohol_consumption))
model4 <- lm(Life_expectancy~ ., data = life5)
# summary(model4)


life6 <- subset(life5, select = -c(Hepatitis_B))
model5 <- lm(Life_expectancy~ ., data = life6)
# summary(model5)
```

```
life7 <- subset(life6, select = -c(Thinness_ten_nineteen_years))
model6 <- lm(Life_expectancy~ ., data = life7)
# summary(model6)


life8 <- subset(life7, select = -c(BMI))
model7 <- lm(Life_expectancy~ ., data = life8)
s <- summary(model7)
```

Now let's check the final results of our summary.

```
print(s)
```

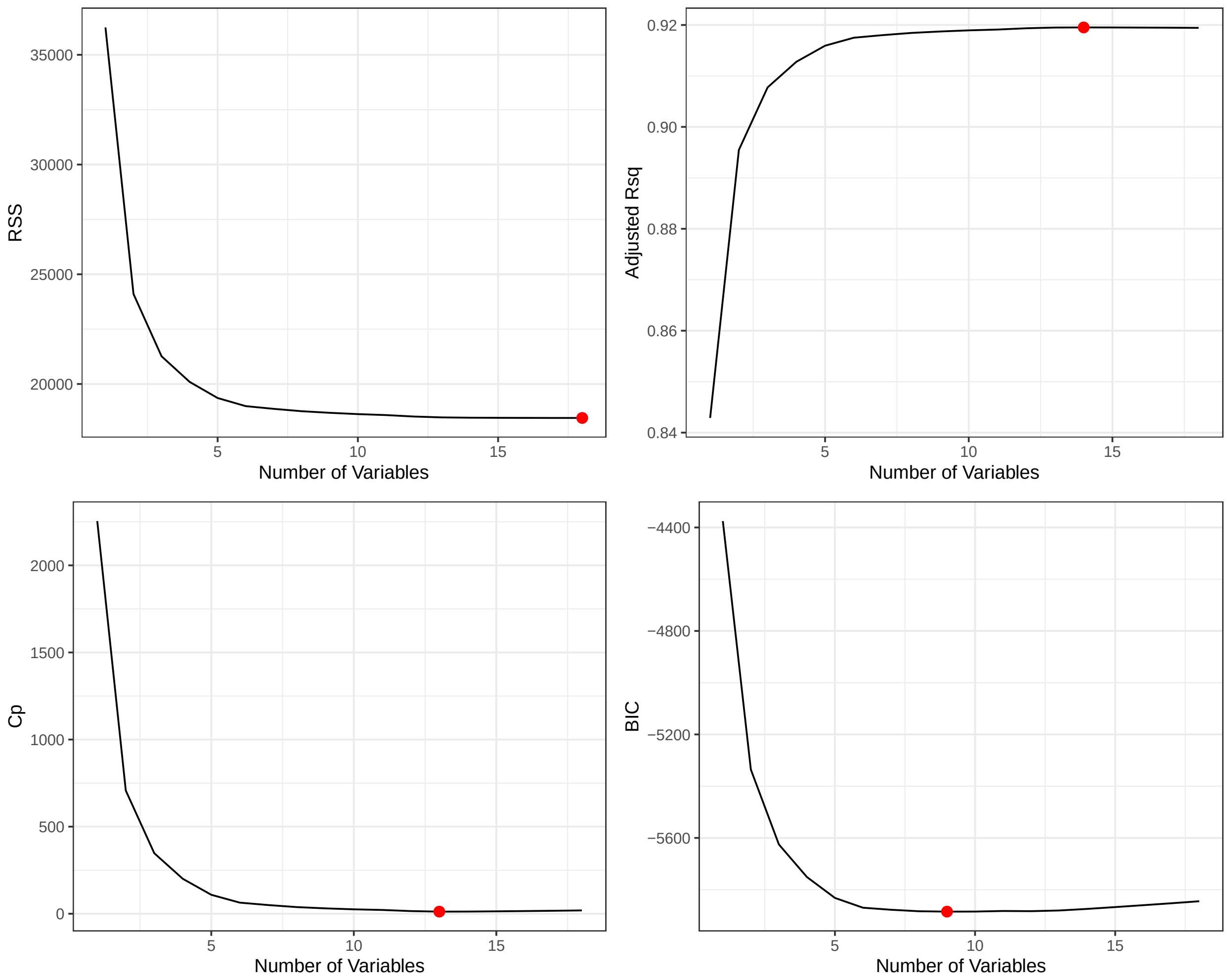
```
##
## Call:
## lm(formula = Life_expectancy ~ ., data = life8)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5612 -1.8685  0.0657  1.7281 21.2515
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            7.000e+01  6.498e-01 107.728  < 2e-16 ***
## RegionAsiaOceania      7.621e-01  2.102e-01   3.626 0.000294 ***
## RegionEuropean Union   6.014e+00  3.789e-01  15.874  < 2e-16 ***
## RegionMiddle East      1.205e+00  3.060e-01   3.938 8.44e-05 ***
## RegionNorth America    4.906e+00  5.518e-01   8.891 9.18e-06 ***
## RegionRest of Europe   1.441e+00  3.054e-01   4.719 2.51e-06 ***
## RegionSouth America    3.187e+00  2.273e-01  14.022  < 2e-16 ***
## Adult_mortality       -5.940e-02  1.077e-03 -55.128  < 2e-16 ***
## Diphtheria             7.631e-02  4.858e-03  15.707  < 2e-16 ***
## Incidents_HIV         -2.225e-01  3.662e-02  -6.076 1.43e-09 ***
## GDP_per_capita         2.000e-05  5.476e-06   3.653 0.000265 ***
## Schooling              4.643e-01  3.129e-02  14.837  < 2e-16 ***
## MeaslesCases          -4.897e-04  1.370e-04  -3.575 0.000358 ***
## StatusDeveloping      -1.606e+00  3.611e-01  -4.449 9.05e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.799 on 2358 degrees of freedom
## Multiple R-squared:  0.9199, Adjusted R-squared:  0.9195
## F-statistic:  2084 on 13 and 2358 DF,  p-value: < 2.2e-16
```

The preceding summary yields important insights. Firstly, all variables, except for the varying levels of "Region," exhibit statistically significant results with remarkably low p-values. Moreover, the linear model demonstrates a strong fit to the data, as indicated by the considerably high coefficient of determination (R-squared) of nearly 92%. This implies that our set of variables collectively accounts for over 90% of the observed variability in "Life Expectancy."

Through variable selection, we have eliminated 5 variables, resulting in a reduction from 18 variables (inclusive of dummy variables representing "Region" levels) to 13 regressors in our model.

Subsequently, we employed an internal function (regsubsets) in R to perform an additional variable selection procedure in order to take into account other metrics, comparing models with varying numbers of variables using multiple criteria such as residual sum of squares (RSS), adjusted R-squared, Mallow's Cp, and Bayesian Information Criterion (BIC). From this analysis, we identified the "simplest model" that exhibited optimal performance across these measures. Remarkably, we observed that the variables selected remained consistent with our previous approach, except for the regions we specifically intended to include in our analysis. This consistency reinforced our confidence in the correctness of our model selection process.

```r
regfit.best <- regsubsets(Life_expectancy~., data= life3, nvmax = 30, method = "forward")
reg.summary <- summary(regfit.best)

#- residual sum of squares:
rss_df <-
  data.frame(Num_Variables = 1:length(reg.summary$rss),
             RSS = reg.summary$rss)
rss_plot <- ggplot(data = rss_df, aes(x = Num_Variables, y = RSS)) +
  geom_line() +
  labs(x = "Number of Variables", y = "RSS") +
  geom_point(
    data = rss_df[which.min(rss_df$RSS),],
    aes(x = Num_Variables, y = RSS),
    col = "red",
    cex = 4,
    pch = 20
  ) +
  theme_bw()

# Create a data frame with the adjusted R^2 values
adjr2_df <-
  data.frame(Num_Variables = 1:length(reg.summary$adjr2),
             AdjR2 = reg.summary$adjr2)
adjr2_plot <-
  ggplot(data = adjr2_df, aes(x = Num_Variables, y = AdjR2)) +
  geom_line() +
  labs(x = "Number of Variables", y = "Adjusted Rsq") +
  geom_point(
    data = adjr2_df[which.max(adjr2_df$AdjR2),],
    aes(x = Num_Variables, y = AdjR2),
    col = "red",
    cex = 4,
    pch = 20
  ) +
  theme_bw()


# Create a data frame with the Mallow's Cp values
cp_df <-
  data.frame(Num_Variables = 1:length(reg.summary$cp),
             Cp = reg.summary$cp)
cp_plot <- ggplot(data = cp_df, aes(x = Num_Variables, y = Cp)) +
  geom_line() +
  labs(x = "Number of Variables", y = "Cp") +
  geom_point(
    data = cp_df[which.min(cp_df$Cp),],
    aes(x = Num_Variables, y = Cp),
    col = "red",
    cex = 4,
    pch = 20
  ) +
  theme_bw()

# Create a data frame with the BIC values
bic_df <-
  data.frame(Num_Variables = 1:length(reg.summary$bic),
             BIC = reg.summary$bic)
bic_plot <- ggplot(data = bic_df, aes(x = Num_Variables, y = BIC)) +
  geom_line() +
  labs(x = "Number of Variables", y = "BIC") +
  geom_point(
    data = bic_df[which.min(bic_df$BIC),],
    aes(x = Num_Variables, y = BIC),
    col = "red",
    cex = 4,
    pch = 20
  ) +
  theme_bw()

plot_grid(rss_plot, adjr2_plot, cp_plot, bic_plot, nrow = 2)
```



The remaining variables after the automatic variable selection procedure, according to the 4 metrics are:

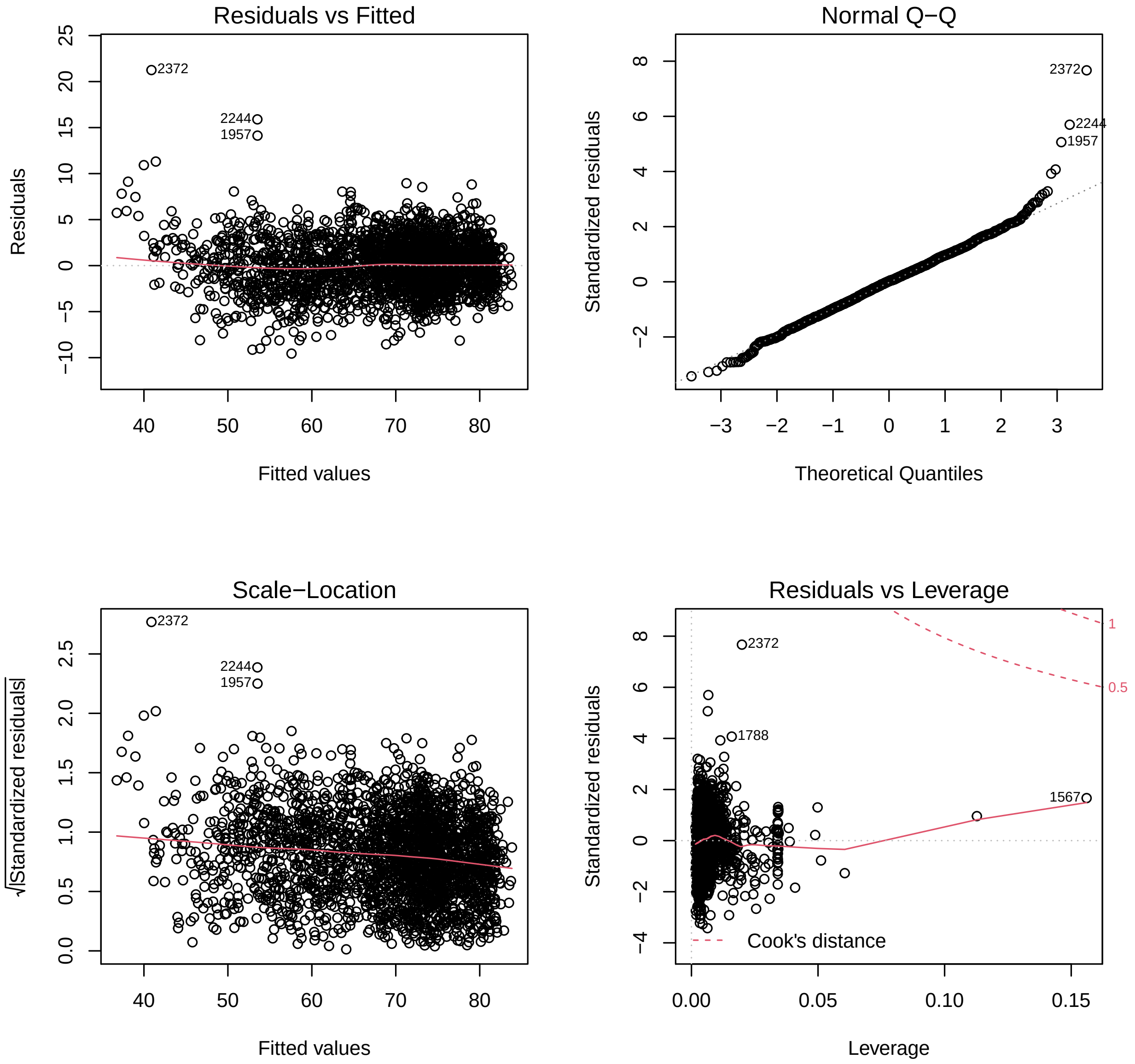
```
vars <- names(summary(regfit.best)$which[which.min(summary(regfit.best)$bic), ])

vars_rss <- as.vector(summary(regfit.best)$which[which.min(summary(regfit.best)$rss),])
vars_adjr2 <- as.vector(summary(regfit.best)$which[which.max(summary(regfit.best)$adjr2),])
vars_cp <- as.vector(summary(regfit.best)$which[which.min(summary(regfit.best)$cp),])
vars_bic <- as.vector(summary(regfit.best)$which[which.min(summary(regfit.best)$bic),])

my_table <- data.frame(
  "Variable" = vars,
  "RSS" = vars_rss,
  "Adjusted R2" = vars_adjr2,
  "Mallows's Cp" = vars_cp,
  "BIC" = vars_bic
)
colnames(my_table) <- c("Variable","RSS","Adjusted R2", "Mallows's Cp", "BIC")
knitr::kable(my_table)
```

| Variable | RSS | Adjusted R2 | Mallows's Cp | BIC |
|---|---|---|---|---|
| (Intercept) | TRUE | TRUE | TRUE | TRUE |
| RegionAsiaOceania | TRUE | TRUE | TRUE | FALSE |
| RegionEuropean Union | TRUE | TRUE | TRUE | FALSE |
| RegionMiddle East | TRUE | TRUE | TRUE | FALSE |
| RegionNorth America | TRUE | TRUE | TRUE | TRUE |
| RegionRest of Europe | TRUE | TRUE | TRUE | FALSE |
| RegionSouth America | TRUE | TRUE | TRUE | TRUE |
| Adult_mortality | TRUE | TRUE | TRUE | TRUE |
| Alcohol_consumption | TRUE | FALSE | FALSE | FALSE |
| Hepatitis_B | TRUE | FALSE | FALSE | FALSE |
| BMI | TRUE | TRUE | FALSE | FALSE |
| Diphtheria | TRUE | TRUE | TRUE | TRUE |
| Incidents_HIV | TRUE | TRUE | TRUE | TRUE |
| GDP_per_capita | TRUE | TRUE | TRUE | TRUE |
| Population_mln | TRUE | FALSE | FALSE | FALSE |
| Thinness_ten_nineteen_years | TRUE | FALSE | FALSE | FALSE |
| Schooling | TRUE | TRUE | TRUE | TRUE |
| MeaslesCases | TRUE | TRUE | TRUE | TRUE |
| StatusDeveloping | TRUE | TRUE | TRUE | TRUE |

As we can see from the plots, we have that the RSS selects the most complex model. This is explained by the fact that the RSS can not decrease as we remove variables, so it will achieve its lowest value when we include all the variables. The adjsuted R-squared is a measure that tries to modify the R-squared by balancing model completeness and simplicity. This is however still a bit biased towards more complex models. The Mallows's Cp provides a measure of how well the model fits the data, taking into account the number of predictors included in the model. The interpretation of Mallows's Cp involves comparing its value to a reference value given by the number of regressors in our initial model. Finally, we have the BIC, which achieves its lowest value when having 9 regressors. Out of the 4 measures, this is the one we compare our model with, as it strikes a good balance between model completeness and simplicity.

Finally, after having set up a linear model, it is of fundamental importance to check that the basic assumption of the model are met. For this purpose, we can exploit the internal R function *plot()* in order to visualize the diagostic plots and get an idea of whether the assumptions are met.

```
par(mfrow = c(2,2))
plot(model7)
```



From these diagnostic plots, some considerations can be done regarding the assumptions of our model.

- The Residual VS Fitted Values plot demonstrates that the linearity and the homoschedasticity assumptions of our linear model are met. As a matter of fact, all the points are evenly scattered around the 0 line, and there is no evident trend among them;
- The normal QQ plot shows that the normality assumption is also met. The points lie across the diagonal line, with a slight though not significant departure from the line for larger quantiles;
- The Scale-Location plot shows again that the homoschedasticity assumption is met, as the points are evenly scattered around 1;
- The Residual vs Leverage plot instead shows that there are not units that should be classified as “influential units”, therefore no concerning outliers were detected.

## Ridge Regression

Now we will explore two regularized versions of linear regression, starting from Ridge Regression. This model is structured in the same way as the linear regression model, but the optimization criterion is now changed. While linear regression aims at finding the parameters so that the Mean Squared Error is minimised, Ridge Regression still aims at minimising the MSE, but with an additional penalty term. This penalty is the so called L2 regularization norm, and it is defined as:

$$L2 = \lambda * ||\beta||_2^2 = \lambda * \sum_{i=1}^{n} \beta_i^2$$

$$\min \left( \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda * \sum_{i=1}^{p} \beta_i^2 \right)$$

Here, $\lambda$ represents the regularization parameter, and $||\beta||_2^2$ represents the squared L2 norm of the coefficient vector ($\beta$) of the model. By adding this penalty term to the objective function of a model we aim at discouraging large coefficient values, shrinking all parameters towards zero. This is done in the hope of improving the prediction capabilities of our regression model.

For our analysis, we have split the data into two sets: the training set and the test set. The standard threshold we used for this division is 80% for the training set and 20% for the test set. This division allows us to use the training set to build and train our models, while the test set serves as an independent dataset to evaluate the model's performance and generalization ability. In addition to the data split, we have also defined a grid range for the lambda parameters that we will test to tune this hyperparameter. In this case, we have set the grid range for lambda parameters as powers of 10, with exponents ranging from -10 to 10, with a total number of 100 values. This range allows us to explore a wide range of regularization strengths, from very small values to large values. By trying multiple lambda values within this range, we can determine the optimal level of regularization that strikes the right balance between bias and variance in our models. Finally, before stepping into Ridge Modeling, it is best to rescale our features using the formula:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}.$$

This is because in the penalty term we have the sum of the squared coefficient, so Ridge regression coefficient might change greatly when multiplying a single predictor by a constant.

```r
set.seed(17)

X <- model.matrix(Life_expectancy~., data=life3)
X <- X[,-1]
y <- life3$Life_expectancy


training_size <- round(0.8*nrow(life3))
test_size <- nrow(life3) - training_size

indices <- sample.int(nrow(life3), size = training_size, replace = F)

life3_train <- life3[indices,]
life3_test <- life3[-indices,]


X_train <- model.matrix(Life_expectancy~., data=life3_train)
X_train <- X_train[,-1]

X_test <- model.matrix(Life_expectancy~., data=life3_test)
X_test <- X_test[,-1]

y_train <- life3_train$Life_expectancy
y_test <- life3_test$Life_expectancy

grid <- 10^seq(10, -10, length=100)

scaling <- function(c){
  den <- sqrt((1/length(c))*sum((c - mean(c))^2))
  out <- c/den
  return(out)
}

X_train_ridge <- X_train
X_train_ridge[,c(7:17)] <- apply(X_train_ridge[,c(7:17)], 2, scaling)
X_test_ridge <- X_test
X_test_ridge[,c(7:17)] <- apply(X_test_ridge[,c(7:17)], 2, scaling)
```

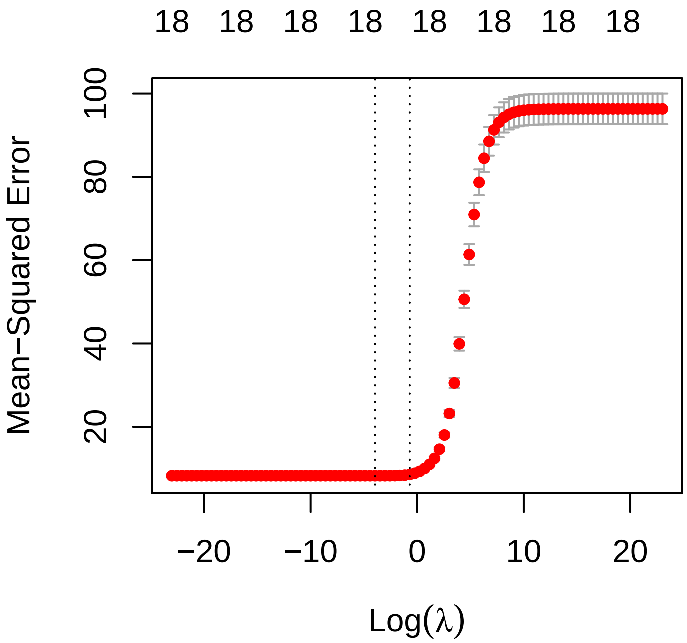
The dataset that is split in training and test set is the the one that was produced after the exploratory data analysis (namely *life3*), and not the one obtained after the variable selection procedure. This is because regularization will take care of shrinking towards zero irrelevant parameters.

The most suitable $\lambda$ is chosen according to 10-fold cross validation, which is a technique exploited for tuning hyperparameters with the aim of enhancing predictive abilities of our model.

```
set.seed(123) # For Reproducibility
cv_model <- cv.glmnet(X_train_ridge, y_train, alpha = 0, nfolds = 10,
                      type.measure = "mse", lambda = grid)
best_lambda_ridge <- cv_model$lambda.min
ridge_model <- glmnet(X_train_ridge, y_train, alpha = 0, lambda = best_lambda_ridge)
ridge.pred <- predict(ridge_model, s = best_lambda_ridge, newx = X_test_ridge,
                      type="response")
```

The type.measure argument was set to "mse" (mean squared error), indicating that the mean squared error was used as the evaluation metric for model selection during cross-validation. The cv.glmnet function returned the best lambda value, denoted as lambda.min, which corresponds to the lambda value that yielded the lowest mean squared error during cross-validation. We stored this optimal lambda value in the variable best_lambda_ridge. As it can be seen from the plot below, even though cross validation selected a value of lambda of 0.01917, the MSE doesn't really increase for smaller values of lambda. This hints at the fact that our model might not need regularization in order to improve its generalization abilities.

```
plot(cv_model)
```



To make predictions on the test set, denoted by X_test, we used the predict function with the ridge_model object. We specified s = best_lambda_ridge to indicate that we want to use the optimal lambda value for prediction. By following this code sequence, we were able to perform cross-validated ridge regression, determine the optimal lambda value, train the ridge model, and make predictions on the test set. This approach helps us find an appropriate balance between model complexity and generalization performance, ultimately aiding in accurate predictions for new data.

The performance of our Ridge model in terms of MSE on the test set is as follows:

```
cat("MSE on test for Ridge Model on life3:",
    mean((ridge.pred - y_test)^2),"with lambda = ",best_lambda_ridge,  "\n")
```

```
## MSE on test for Ridge Model on life3: 6.850547 with lambda =  0.0191791
```

### Lasso Regression

Lasso Regression is another regularized version of linear regression that also aims to minimize the mean squared error (MSE) while incorporating a penalty term. However, unlike Ridge Regression, Lasso Regression uses a different penalty term known as L1 regularization.

The objective function of Lasso Regression is given by:

$$L1 = \lambda * \sum_{j=1}^{p} |\beta_j|$$

$$\min \left[ \sum_{i=1}^{n} \left( y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

where the penalty term denotes the sum of the absolute values of all the coefficient values.

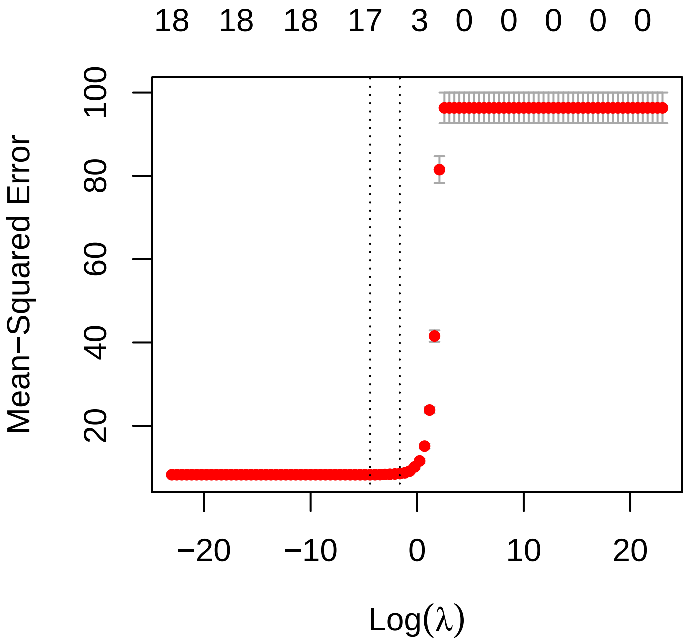
By introducing L1 regularization, Lasso Regression encourages sparse solutions by promoting some coefficients to be exactly zero. This property makes Lasso Regression useful for feature selection, as it tends to automatically identify and discard irrelevant or redundant features. In contrast to Ridge Regression, where the coefficient values gradually shrink towards zero, Lasso Regression can directly force some coefficients to become exactly zero. Lasso Regression is particularly beneficial when dealing with high-dimensional datasets, as it not only provides prediction accuracy but also performs feature selection by eliminating less relevant predictors.

The most suitable $\lambda$ is again chosen according to 10-fold cross validation. It is a common choice to standardize the features of our dataset before applying Lasso regression. This is done in order to bring all the features to a common scale, ensuring that no feature dominates the Lasso regularization process. This leads to Lasso penalizing features equally.

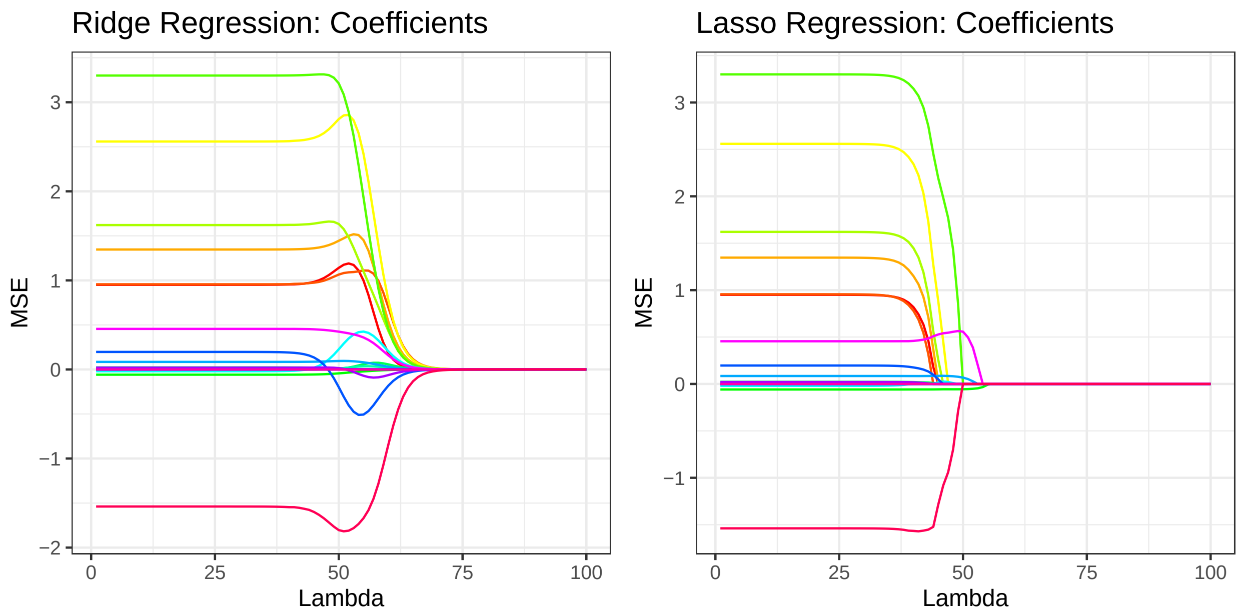
```
X_train_lasso <- X_train
X_train_lasso[,c(7:17)] <- scale(X_train_lasso[,c(7:17)])
X_test_lasso <- X_test
X_test_lasso[,c(7:17)] <- scale(X_test_lasso[,c(7:17)])

set.seed(123) # For Reproducibility
cv_model <- cv.glmnet(X_train, y_train, alpha = 1, nfolds = 10,
                      type.measure = "mse", lambda = grid)
best_lambda_lasso <- cv_model$lambda.min
lasso_model <- glmnet(X_train, y_train, alpha = 0, lambda = best_lambda_lasso)
lasso.pred <- predict(lasso_model, s = best_lambda_lasso, newx = X_test, type="response")
```

The $\lambda$ selected according to CV for our Lasso model is 0.012045. This shows the same conditions as in the Ridge regression, hinting at the fact that regularization is not really needed as our data can be already well explained using a simple linear model.

```
plot(cv_model)
```



```r
results <- as.data.frame(matrix(nrow = 100, ncol = 18))
for (i in 1:100) {
  ridge_model <- glmnet(X_train, y_train, alpha = 0, lambda = grid[i])
  results[i,] <- as.vector(ridge_model$beta)
}

plot_data <- data.frame(Lambda = 100:1, (results))
plot_data <- reshape2::melt(plot_data, id.vars = "Lambda")

num_variables <- ncol(results)
color_palette <- rainbow(num_variables)
ridge_coef_plot <-
  ggplot(plot_data, aes(x = Lambda, y = value, color = variable)) +
  geom_line() +
  scale_color_manual(values = color_palette)  +
  xlab("Lambda") +
  ylab("MSE") +
  ggtitle("Ridge Regression: Coefficients") +
  theme_bw() +
  theme(plot.title = element_text(size = 14), legend.position = "right") +
  guides(color = "none")




results <- as.data.frame(matrix(nrow = 100, ncol = 18))
for (i in 1:100) {
  lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = grid[i])
  results[i,] <- as.vector(lasso_model$beta)
}

plot_data <- data.frame(Lambda = 100:1, (results))
plot_data <- reshape2::melt(plot_data, id.vars = "Lambda")

num_variables <- ncol(results)
color_palette <- rainbow(num_variables)
lasso_coef_plot <-
  ggplot(plot_data, aes(x = Lambda, y = value, color = variable)) +
  geom_line() +
  scale_color_manual(values = color_palette)  +
  xlab("Lambda") +
  ylab("MSE") +
  ggtitle("Lasso Regression: Coefficients") +
  theme_bw() +
  theme(plot.title = element_text(size = 14), legend.position = "right") +
  guides(color = "none")

lay <- rbind(c(1,2))
grid.arrange(ridge_coef_plot,lasso_coef_plot, layout_matrix = lay)
```



The disparities between the Lasso and Ridge methods become even more pronounced when examining these two plots. In ridge regression, the coefficients of the variables converge collectively, reaching a value close to 0 at approximately the same time. However, in lasso regression, the coefficients converge towards zero, one by one, exhibiting a sequential diminishing effect, just like in a classic variable selection procedure.

Again the performance of Lasso regression was evaluated on the test set, producing the following MSE:

```
cat("MSE on test for Lasso Model on life3:", mean((lasso.pred - y_test)^2),
    "with lambda = ",best_lambda_lasso,  "\n")
```

```
## MSE on test for Lasso Model on life3: 6.737747 with lambda =  0.01204504
```

### Model Comparison

In this section we are going to analyse the difference in performance on the test set for the three models: linear regression (on the full dataset), Ridge and Lasso regression. Before proceding however, we have to train a linear model on the training set and evaluate its performance in terms of MSE on the test set.

```
set.seed(123)
linear_model <- lm(Life_expectancy~ ., data = life3_train)
lm.pred <- predict(linear_model, life3_test[,-12], type = "response")
```

Then we can compare the results of the three models in terms of Mean Squared Error on the test set:

```
cat(" MSE on test for Linear Model on life3:", mean((lm.pred - y_test)^2), "\n",
    "MSE on test for Ridge Model on life3:", mean((ridge.pred - y_test)^2),
    "with lambda = ",best_lambda_ridge,  "\n",
    "MSE on test for Lasso Model on life3:", mean((lasso.pred - y_test)^2),
    "with lambda = ",best_lambda_lasso,  "\n")
```

```
##  MSE on test for Linear Model on life3: 6.730594
##  MSE on test for Ridge Model on life3: 6.850547 with lambda =  0.0191791
##  MSE on test for Lasso Model on life3: 6.737747 with lambda =  0.01204504
```

Upon comparing the ridge model, the lasso model, and the full model, we have observed that the selected lambda values are significantly small, and the disparities among the three mean squared error (MSE) measurements computed on the test set are negligible. Consequently, we can confidently deduce that our model does not necessitate any form of regularization. Furthermore, the outcomes regarding generalizability remain unchanged across the board.

## Prediction on New Data: Life Expectancy in 2016

After building several models for our dataset, we recognized the importance of assessing their performance on new data to evaluate their predictive capabilities. To achieve this, we obtained a separate dataset from the year 2016, containing the same columns as our original dataset, and attempted to predict the life_expectancy variable using our trained models.

By testing our models on this new dataset, we can assess how well they generalize to unseen data and gain insights into their predictive power. This evaluation allows us to understand if the models have learned meaningful patterns and relationships that can be applied to different time periods or populations.

### Data Collection

The data collection for new data required a discrete amount of work. This is because a more recent version of this dataset was not available, and so each variable has to be looked for on reliable websites. The websites where we found the majority of our data is the official website of the World Health Organization (https://www.who.int/data), but also from other reliable sources such as "Our World in Data" (https://ourworldindata.org/) and "The World Bank" (https://www.worldbank.org/en/home). The next step was to merge all the variables together into one unique datasets and check whether the new values were coherent with the previous ones.

## Predictions on 2016

```
new_data_2016 <- read_csv("New2016Data_LifeExp.csv")

new_data_2016$MeaslesCases <- new_data_2016$MeaslesCases/new_data_2016$Population_mln

new_data_2016$Status <- as.factor(new_data_2016$Status)

summary(new_data_2016)
```

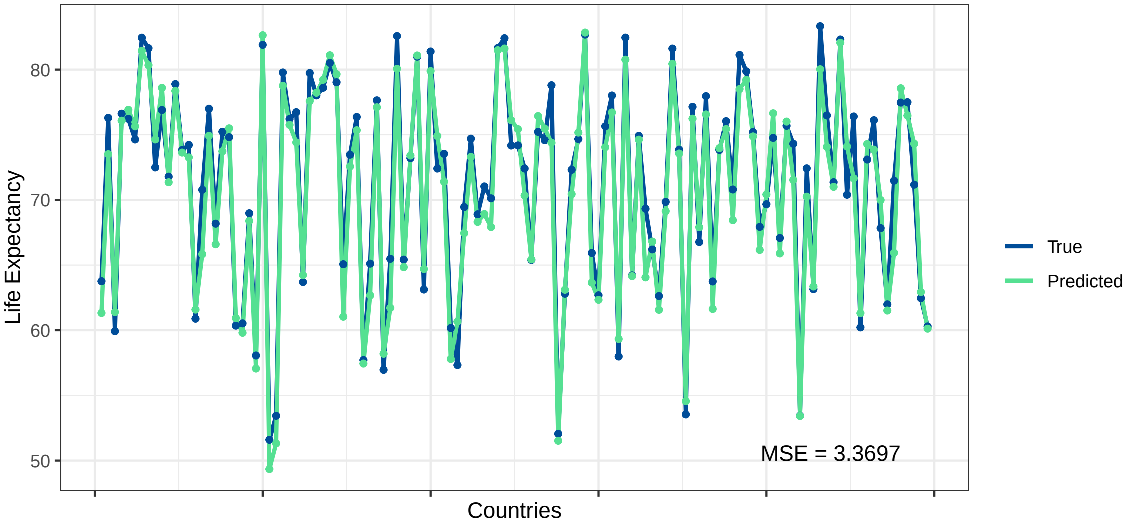
```
##    Country               Year      Adult_mortality       BMI
##  Length:124         Min.   :2016   Min.   : 52.00   Min.   :20.50
##  Class :character   1st Qu.:2016   1st Qu.: 96.75   1st Qu.:23.70
##  Mode  :character   Median :2016   Median :150.50   Median :26.20
##                     Mean   :2016   Mean   :168.46   Mean   :25.51
##                     3rd Qu.:2016   3rd Qu.:221.75   3rd Qu.:27.12
##                     Max.   :2016   Max.   :483.00   Max.   :29.50
##      Polio       Incidents_HIV     GDP_per_capita     Schooling
##  Min.   :44.00   Min.   : 0.0100   Min.   :   306   Min.   : 1.500
##  1st Qu.:82.75   1st Qu.: 0.0800   1st Qu.:  1570   1st Qu.: 5.925
##  Median :92.00   Median : 0.1450   Median :  5538   Median : 8.550
##  Mean   :87.73   Mean   : 0.7446   Mean   : 11008   Mean   : 8.206
##  3rd Qu.:97.00   3rd Qu.: 0.4000   3rd Qu.: 13669   3rd Qu.:10.725
##  Max.   :99.00   Max.   :14.3000   Max.   :105462   Max.   :14.100
##  Life_expectancy Alcohol_consumption  MeaslesCases      Population_mln
##  Min.   :51.59   Min.   : 0.000      Min.   :   0.000   Min.   :   0.0906
##  1st Qu.:65.41   1st Qu.: 1.290      1st Qu.:   0.000   1st Qu.:   3.0056
##  Median :73.34   Median : 3.890      Median :   1.159   Median :   9.8307
##  Mean   :71.28   Mean   : 4.668      Mean   : 106.559   Mean   :  46.3052
##  3rd Qu.:76.77   3rd Qu.: 7.178      3rd Qu.:   7.993   3rd Qu.:  28.1846
##  Max.   :83.33   Max.   :15.610      Max.   :9992.543   Max.   :1401.8897
##  Thinness_ten_nineteen_years  Hepatitis_B      Diphtheria           Status
##  Min.   : 0.300              Min.   :26.00   Min.   :19.00   Developed : 22
##  1st Qu.: 1.900              1st Qu.:84.00   1st Qu.:84.75   Developing:102
##  Median : 4.200              Median :92.00   Median :93.00
##  Mean   : 5.247              Mean   :87.22   Mean   :87.90
##  3rd Qu.: 7.150              3rd Qu.:97.00   3rd Qu.:97.00
##  Max.   :26.700              Max.   :99.00   Max.   :99.00
##     Region
##  Length:124
##  Class :character
##  Mode  :character
##
##
##
```

By comparing the scales and the summary measures of the different variables, we are able to state that the variables are compatible with out previous dataset, with no feature showing a completely different scale or location.

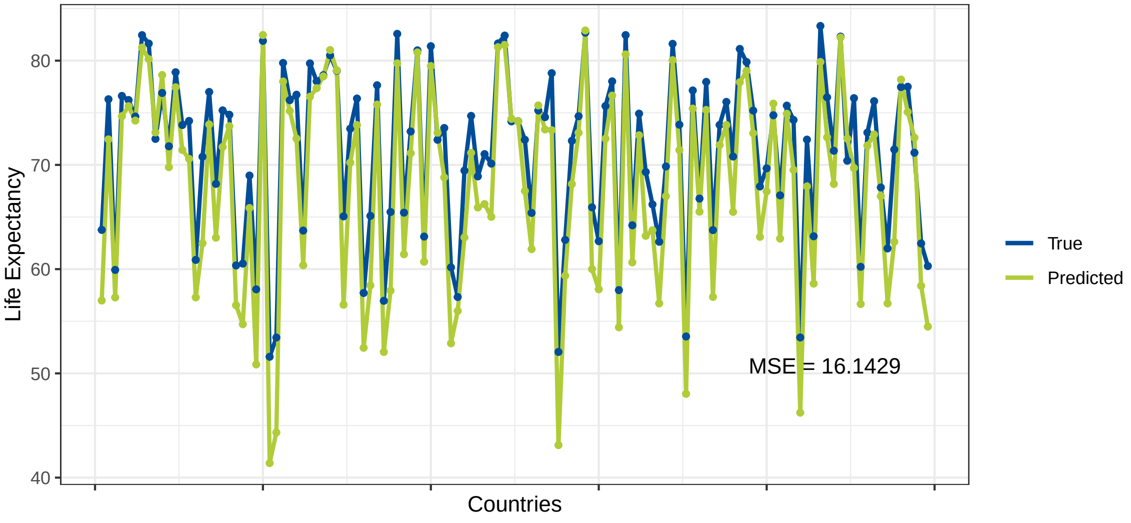
```
X_predict <- (new_data_2016[,-c(1,2, 9)]) # Remove country, Year, Life_expectancy
```

```r
y_true <- new_data_2016$Life_expectancy

y_pred_linearmodel <- predict(model7 ,(X_predict))

mse_linear <- (sum((y_pred_linearmodel - y_true)^2))/length(y_true)

X <- model.matrix(Life_expectancy~., data=life3)
X <- X[,-1]
y <- life3$Life_expectancy

X_test_2016 <- model.matrix(Life_expectancy~., data=new_data_2016[,-c(1,2)])
X_test_2016 <- X_test_2016[,-1]
X_test_2016 <- X_test_2016[,match(colnames(X), colnames(X_test_2016))]

X_ridge <- X
# Scale without considering the categorical variables:
X_ridge[,c(7:17)] <- apply(X_ridge[,c(7:17)], 2, scaling)
X_test_2016_ridge <- X_test_2016
X_test_2016_ridge[,c(7:17)] <- (apply(X_test_2016_ridge[,c(7:17)], 2, scaling))

X_lasso <- X
X_lasso[,c(7:17)] <- scale(X_lasso[,c(7:17)])
X_test_2016_lasso <- X_test_2016
X_test_2016_lasso[,c(7:17)] <- scale(X_test_2016_lasso[,c(7:17)])

grid <- 10^seq(10, -10, length=100)

set.seed(123)
cv_model <- cv.glmnet(X_ridge, y, alpha = 0, nfolds = 10,
                      type.measure = "mse", lambda = grid)
best_lambda_ridge <- cv_model$lambda.min
ridge_model <- glmnet(X_ridge, y, alpha = 0, lambda = best_lambda_ridge)
y_pred_ridge <- predict(ridge_model, s = best_lambda_ridge,
                        newx = X_test_2016_ridge, type="response")
mse_ridge <- (sum((y_pred_ridge - y_true)^2))/length(y_true)

# set.seed(123)
cv_model <- cv.glmnet(X_lasso, y, alpha = 1, nfolds = 10,
                      type.measure = "mse", lambda = grid)
best_lambda_lasso <- cv_model$lambda.min
lasso_model <- glmnet(X_lasso, y, alpha = 1, lambda = best_lambda_lasso)
y_pred_lasso <- predict(lasso_model,  s = best_lambda_lasso,
                        newx = X_test_2016_lasso, type="response")
mse_lasso <- (sum((y_pred_lasso - y_true)^2))/length(y_true)


linear_plot <- ggplot(data = data.frame(y_true = y_true,
                                        y_pred_linearmodel = y_pred_linearmodel),
                      aes(x = 1:length(y_true))) +
  geom_line(aes(y = y_true, color = "#004d99"), size = 1) +
  geom_line(aes(y = y_pred_linearmodel, color = "#55e092"), size = 1) +
  geom_point(aes(y = y_true), color = "#004d99", size = 2, shape = 20) +
  geom_point(aes(y = y_pred_linearmodel), color = "#55e092", size = 2, shape = 20) +
  scale_color_manual(values = c("#004d99", "#55e092"), labels = c("True", "Predicted")) +
  labs(title = "Linear Regression", y = "Life Expectancy", color = " ", x = "Countries") +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5))+
  annotate("text", x = 120, y = 50, label = paste("MSE =", round(mse_linear,4)),
           hjust = 1, vjust = 0, size = 4)

ridge_plot <- ggplot(data = data.frame(y_true = y_true, y_pred_ridge = y_pred_ridge),
                     aes(x = 1:length(y_true))) +
  geom_line(aes(y = y_true, color = "#004d99"), size = 1) +
  geom_line(aes(y = y_pred_ridge, color = "#b0cc39"), size = 1) +
  geom_point(aes(y = y_true), color = "#004d99", size = 2, shape = 20) +
  geom_point(aes(y = y_pred_ridge), color = "#b0cc39", size = 2, shape = 20) +
  scale_color_manual(values = c("#004d99", "#b0cc39"), labels = c("True", "Predicted")) +
  labs(y = "Life Expectancy", color = " ", x = "Countries") +
  theme_bw()+
  theme(axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5))+
  annotate("text", x = 120, y = 50, label = paste("MSE =", round(mse_ridge,4)),
           hjust = 1, vjust = 0, size = 4)+
  ggtitle(bquote("Ridge Regression with " * lambda ~ "=" ~ .(round(best_lambda_ridge, 4))))

lasso_plot <- ggplot(data = data.frame(y_true = y_true, y_pred_lasso = y_pred_lasso),
                     aes(x = 1:length(y_true))) +
  geom_line(aes(y = y_true, color = "#004d99"), size = 1) +
  geom_line(aes(y = y_pred_lasso, color = "#ffa600"), size = 1) +
  geom_point(aes(y = y_true), color = "#004d99", size = 2, shape = 20) +
  geom_point(aes(y = y_pred_lasso), color = "#ffa600", size = 2, shape = 20) +
  scale_color_manual(values = c("#004d99", "#ffa600"), labels = c("True", "Predicted")) +
  labs(y = "Life Expectancy", color = " ", x = "Countries") +
  theme_bw()+
  theme(axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5))+
  annotate("text", x = 120, y = 50, label = paste("MSE =", round(mse_lasso,4)),
           hjust = 1, vjust = 0, size = 4)+
  ggtitle(bquote("Lasso Regression with " * lambda ~ "=" ~ .(round(best_lambda_lasso, 4))))


plot_grid(linear_plot, ridge_plot, lasso_plot, nrow = 3)
```

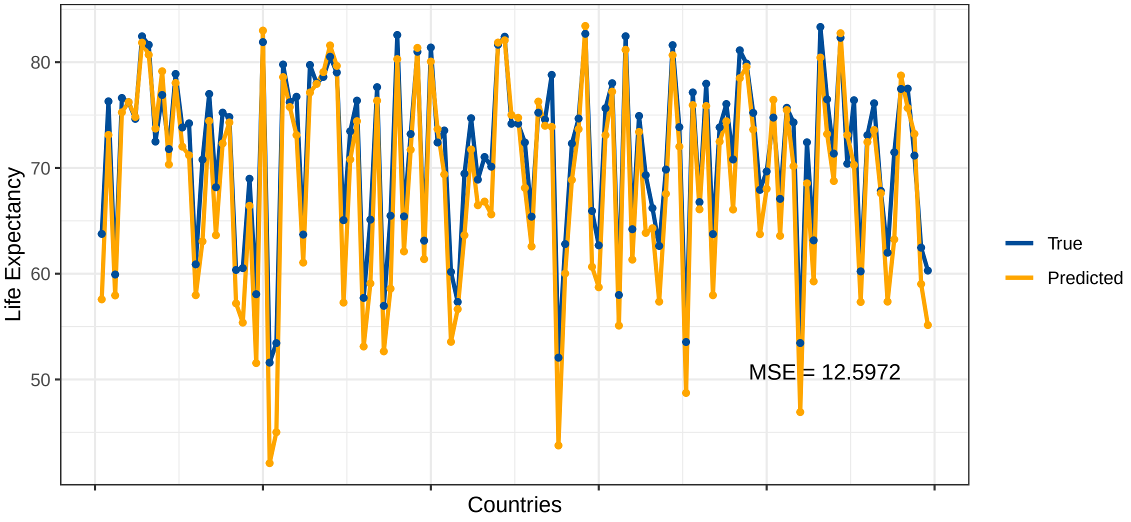
Linear Regression



Ridge Regression with $\lambda$ = 0.0076



Lasso Regression with $\lambda$ = 0.0048



Based on the plots and the absence of significant errors in the predictions of our models for the 2016 dataset, it appears that our models are performing quite well on new data. The fact that the predicted values are close to each other indicates a consistent and reliable performance across the different models. However, the linear regression model seems to generalize better than its regularized variants, according to MSE. This might happen when the underlying relationship in our data is approximately linear and there are no significant violations of the assumptions of linear regression (such as linearity, independence of errors, constant variance, and normality of errors). This makes linear regression a good fit to the data, producing satisfactory results in terms of MSE. On the other hand, ridge regression and lasso regression are regularization techniques that are often used when the number of input variables is large and we want to prevent overfitting and improve generalization performance. However, in cases where the data has a simple linear structure and the number of predictors is not too large, the additional regularization introduced by ridge regression and lasso regression may not be necessary. In such scenarios, a simple linear regression model can outperform the regularized variants in terms of MSE because it can provide a closer fit to the data without introducing unnecessary regularization.

The consistency in the predictions suggests that our models have effectively captured the underlying patterns and relationships in the training data, enabling them to make accurate predictions for the life_expectancy variable. This outcome provides confidence in the models' ability to generalize and be applied to other years for prediction purposes.

# Conclusions & Further Works

From our analysis of the data, we have gained valuable insights into some of the factors influencing life expectancy. We observed a strong correlation between certain variables, such as adult mortality and life expectancy, which show a negative correlation, or again Schooling and Life Expectancy, showing instead a positive impact. These findings highlight the importance of these variables in understanding and predicting life expectancy.

In our modeling process, we utilized various techniques such as ridge regression, lasso regression, and linear regression. Through cross-validation and evaluation of different lambda values, we identified the optimal lambda values for each model. We found that despite some variations in the models, they generally performed well and produced accurate predictions.

Additionally, we assessed the performance of our models on new data from the year 2016. The models demonstrated consistent and reliable performance, indicating their potential applicability for predicting life expectancy in different time periods, especially for the linear model.

Overall, our analysis suggests that variables such as adult mortality, vaccination rates, and other factors have a significant impact on life expectancy. The models we developed provide a useful framework for predicting life expectancy and understanding the underlying factors contributing to it.

Among the variables we examined, Schooling emerged as a significant predictor of Life Expectancy. Our model indicates that each additional year of education is associated with an increase in Life Expectancy of approximately 0.46 years, while other factors are held constant.

The relationship between Schooling and Life Expectancy is not direct, but rather operates through various interconnected mechanisms. Countries with a higher average number of years in school often have better healthcare systems. This implies that not only does increased schooling contribute to improved health knowledge and behaviors, but it also indicates better access to healthcare resources, leading to higher life expectancy.

In summary, Schooling plays a crucial role in predicting Life Expectancy. The positive association suggests that higher levels of education are linked to longer life spans, reflecting the combined impact of improved health knowledge and access to healthcare resources in countries with a more advanced education system.

Another significant predictor we found is the variable 'Status' Transitioning from a 'developed' country to a 'developing' country is associated with an average decrease in life expectancy of approximately 1.6 years.

The ‘Status’ variable represents the level of development and socioeconomic factors within a country. ‘Developed’ countries generally have higher levels of economic prosperity, advanced healthcare systems, and better living conditions, contributing to longer life expectancies. In contrast, ‘developing’ countries face socio-economic challenges such as limited healthcare access, higher poverty rates, and lower educational attainment, leading to poorer health outcomes and shorter life spans. This underscores the impact of socio-economic factors on health outcomes and highlights the need to address disparities and improve healthcare access in developing countries for better life expectancy.

It is important to note that while the models showed promising results, further evaluation and refinement may be necessary. Additional evaluation metrics and testing on diverse datasets can help ensure the robustness and generalizability of our models. Moreover, further research and analysis can build upon these findings to enhance our understanding of life expectancy and potentially inform public health policies and interventions aimed at improving population health outcomes. Possible expansions of our analysis could include the examination of larger datasets, including more specific regressors. This could potentially uncover the undirect relationships we have in our model between some of the regressors and life expectancy, for instance Schooling, and explain in a more detailed manner how multiple specific factors affect our dependent variable. Finally, with larger datasets, shrinkage methods like ridge and lasso could become fundamental, as they are designed to simplify our model by applying regularization terms.