# STA141 Final Project

STA 141A FQ24

## Introduction

Life expectancy is an important indicator The purpose of this project is to analyze the relationship certain factors have with an individual's life expectancy. Life expectancy is defined here as a statistical measure of the average time that an individual will live in a given country. Life expectancy is an important

### Questions of Interest

In our analysis, we focus on two particular questions:

## Dataset

The dataset that we used for this project, found on Kaggle, is Life Expectancy Data collected by the World Health Organisation (WHO). The dataset contains information from 193 countries spanning from the years 2000-2015. In addition to life expectancy (our dependent variable in this analysis), the dataset contains the other following variables:

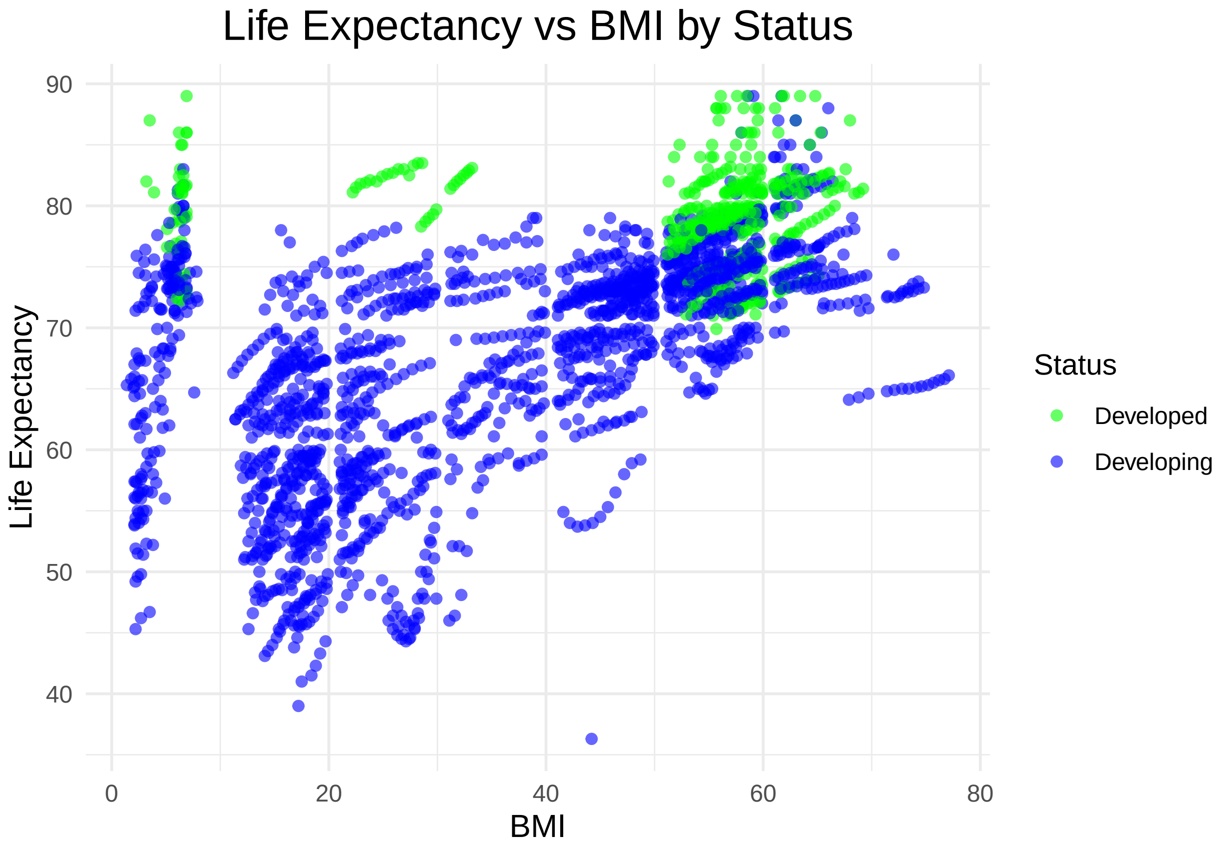
- Year: The year the data was reported
- Status: Developed or Developing status
- Adult Mortality: probability of dying between 15 and 60 years per 1000 population
- infant deaths: Number of Infant Deaths per 1000 population
- Alcohol: Alcohol recorded per capita (15+) consumption (in litres of pure alcohol)
- percentage expenditure: Expenditure on health as a percentage of GDP per capita(%)
- Hepatitis B: HepB immunization coverage among 1-year-olds (%)
- Measles: number of reported cases of measles per 1000 population
- BMI: Average Body Mass Index of entire population
- under-five deaths: Number of under-five deaths per 1000 population
- Polio: Polio (Pol3) immunization coverage among 1-year-olds (%)
- Total expenditure: General government expenditure on health as a percentage of total government expenditure (%)
- Diphtheria: DTP3 immunization coverage among 1-year-olds (%)
- HIV/AIDS: Deaths per 1000 live births HIV/AIDS (0-4 years)
- GDP: Gross Domestic Product per capita (in USD)
- Population: Population of the country
- thinness 1-19 years: Prevalence of thinness among children and adolescents for Age 10 to 19 (% )
- thinness 5-9 years: Prevalence of thinness among children for Age 5 to 9(%)
- Income composition of resources: Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
- Schooling: Number of years of Schooling(years)

### Data Cleaning

An initial exploration of the data revealed that the variables

jdjdj

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



### g

After an initial

### h

Before creating our four models to compare, we removed any remaining data points containing NA values in non-predictor columns. Our data was split into training and testing sets (80% train, 20% test). We tested four different regression models: * Linear Regression: * * *

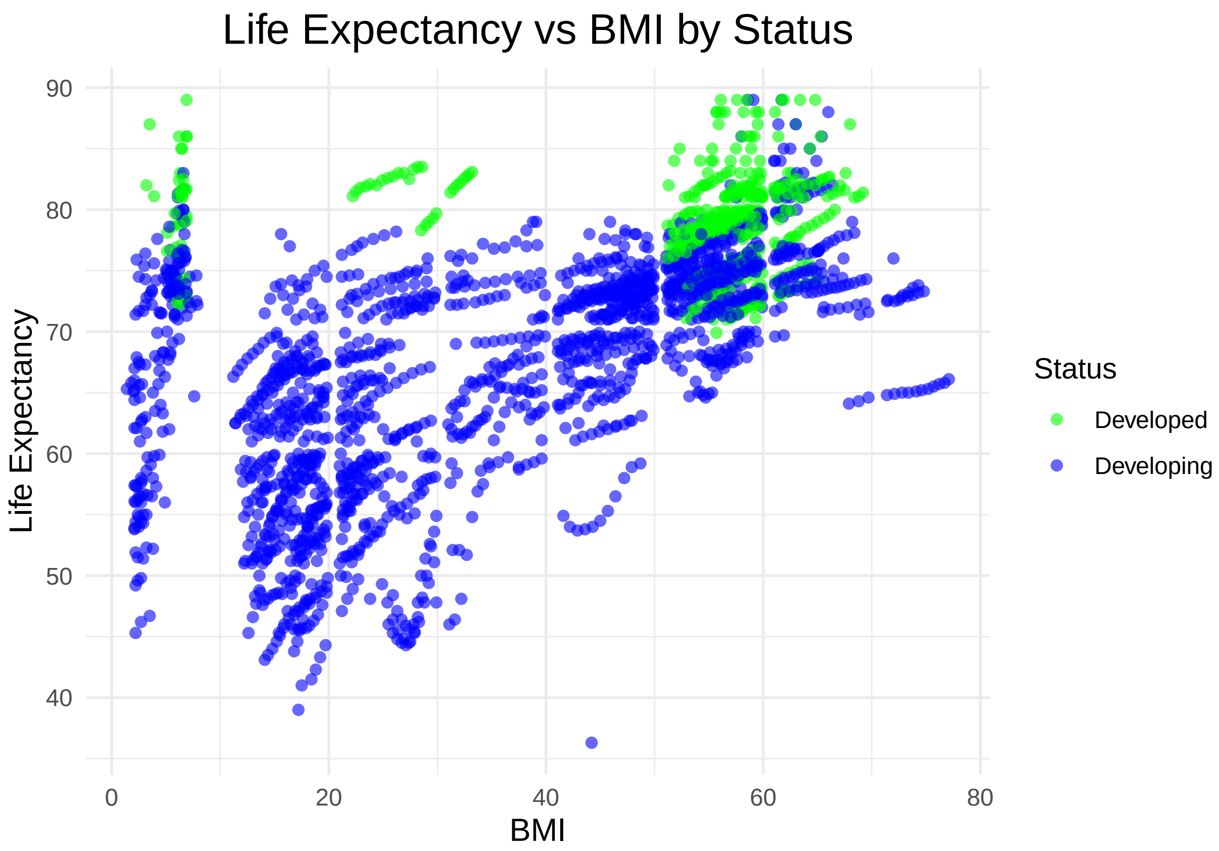
Each model was compared based on their Root Mean Squared Error (RMSE), with a lower score indicating the model to be better at predicting life expectancy.

```r
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(randomForest)
library(caret)
library(glmnet)
library(xgboost)
library(ranger)

# Load the dataset
data <- read.csv("Life Expectancy Data.csv")

# Check for missing values in the dataset
missing_values <- sapply(data, function(x) sum(is.na(x)))
print(missing_values)
```

```
##                         Country                            Year 
##                               0                               0 
##                          Status                 Life.expectancy 
##                               0                              10 
##                 Adult.Mortality                   infant.deaths 
##                              10                               0 
##                         Alcohol          percentage.expenditure 
##                             194                               0 
##                     Hepatitis.B                         Measles 
##                             553                               0 
##                             BMI               under.five.deaths 
##                              34                               0 
##                           Polio               Total.expenditure 
##                              19                             226 
##                      Diphtheria                        HIV.AIDS 
##                              19                               0 
##                             GDP                      Population 
##                             448                             652 
##            thinness..1.19.years              thinness.5.9.years 
##                              34                              34 
## Income.composition.of.resources                       Schooling 
##                             167                             163
```

```r
# Data Cleaning: Remove rows with missing Life.expectancy, Alcohol, GDP, HIV.AIDS
data_clean <- data %>%
  filter(!is.na(Life.expectancy), !is.na(Alcohol), !is.na(GDP), !is.na(HIV.AIDS))

# Relationship between Life Expectancy and BMI
ggplot(data_clean, aes(x = BMI, y = Life.expectancy, color = Status)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Life Expectancy vs BMI by Status",
    x = "BMI",
    y = "Life Expectancy",
    color = "Status"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12)
  ) +
  scale_color_manual(values = c("Developing" = "blue", "Developed" = "green"))
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```r
# Check if there are any missing values left
missing_values_clean <- sapply(data_clean, function(x) sum(is.na(x)))
print("Missing values after cleaning:")
```

```
## [1] "Missing values after cleaning:"
```

```r
print(missing_values_clean)
```

```
##                        Country                           Year
##                              0                              0
##                         Status                Life.expectancy
##                              0                              0
##                Adult.Mortality                  infant.deaths
##                              0                              0
##                        Alcohol         percentage.expenditure
##                              0                              0
##                    Hepatitis.B                        Measles
##                            461                              0
##                            BMI              under.five.deaths
##                             15                              0
##                           Polio               Total.expenditure
##                               7                               5
##                      Diphtheria                        HIV.AIDS
##                               7                               0
##                             GDP                      Population
##                               0                             211
##            thinness..1.19.years              thinness.5.9.years
##                              15                              15
## Income.composition.of.resources                       Schooling
##                               2                               2
```

```r
# If still any NAs in non-predictor columns, we'll remove them.
data_clean <- na.omit(data_clean)

# Ensure 'Status' is a factor
data_clean$Status <- as.factor(data_clean$Status)

# Split the data into train and test sets (80% train, 20% test)
set.seed(42)
trainIndex <- createDataPartition(data_clean$Life.expectancy, p = 0.8, list = FALSE)
train <- data_clean[trainIndex, ]
test <- data_clean[-trainIndex, ]

# Remove 'Country' column from both training and testing datasets
train <- train %>% select(-Country)
test <- test %>% select(-Country)

# --- Convert All Predictors to Numeric ---
# Convert factor columns to numeric (one-hot encoding or label encoding)
train[] <- lapply(train, function(x) if(is.factor(x)) as.numeric(as.factor(x)) else x)
test[] <- lapply(test, function(x) if(is.factor(x)) as.numeric(as.factor(x)) else x)

# --- Model 1: Linear Regression ---
lm_model <- lm(Life.expectancy ~ ., data = train)
lm_preds <- predict(lm_model, test)
lm_rmse <- RMSE(lm_preds, test$Life.expectancy)

# --- Model 2: Random Forest ---
rf_model <- randomForest(Life.expectancy ~ ., data = train, ntree = 100)
rf_preds <- predict(rf_model, test)
rf_rmse <- RMSE(rf_preds, test$Life.expectancy)

# --- Model 3: Ridge Regression (using glmnet) ---
x_train <- as.matrix(train %>% select(-Life.expectancy))
y_train <- train$Life.expectancy
```

```r
ridge_model <- cv.glmnet(x_train, y_train, alpha = 0)
ridge_preds <- predict(ridge_model, s = ridge_model$lambda.min, newx = as.matrix(test %>
ridge_rmse <- RMSE(ridge_preds, test$Life.expectancy)

# --- Model 4: XGBoost ---
x_train <- as.matrix(train %>% select(-Life.expectancy))  # Convert to numeric matrix
y_train <- train$Life.expectancy
x_test <- as.matrix(test %>% select(-Life.expectancy))    # Convert to numeric matrix

dtrain <- xgb.DMatrix(data = x_train, label = y_train)
dtest <- xgb.DMatrix(data = x_test)

xgb_model <- xgboost(data = dtrain, nrounds = 100, objective = "reg:squarederror", verbo
xgb_preds <- predict(xgb_model, dtest)
xgb_rmse <- RMSE(xgb_preds, test$Life.expectancy)

# --- Model Comparison Results ---
results <- data.frame(Model = c("Linear Regression", "Random Forest", "Ridge Regression"
                      RMSE = c(lm_rmse, rf_rmse, ridge_rmse, xgb_rmse))








# --- Correlation Analysis ---
# Correlation between Life Expectancy and key variables
cor_test_alcohol <- cor.test(data_clean$Alcohol, data_clean$Life.expectancy, use = "comp
cor_test_gdp <- cor.test(data_clean$GDP, data_clean$Life.expectancy, use = "complete.obs
cor_test_hiv <- cor.test(data_clean$HIV.AIDS, data_clean$Life.expectancy, use = "complet

# Print correlation results
print(paste("Correlation between Life Expectancy and Alcohol: ", cor_test_alcohol$estima
```

```
## [1] "Correlation between Life Expectancy and Alcohol:  0.402718321727353"
```

```r
print(paste("Correlation between Life Expectancy and GDP: ", cor_test_gdp$estimate))
```

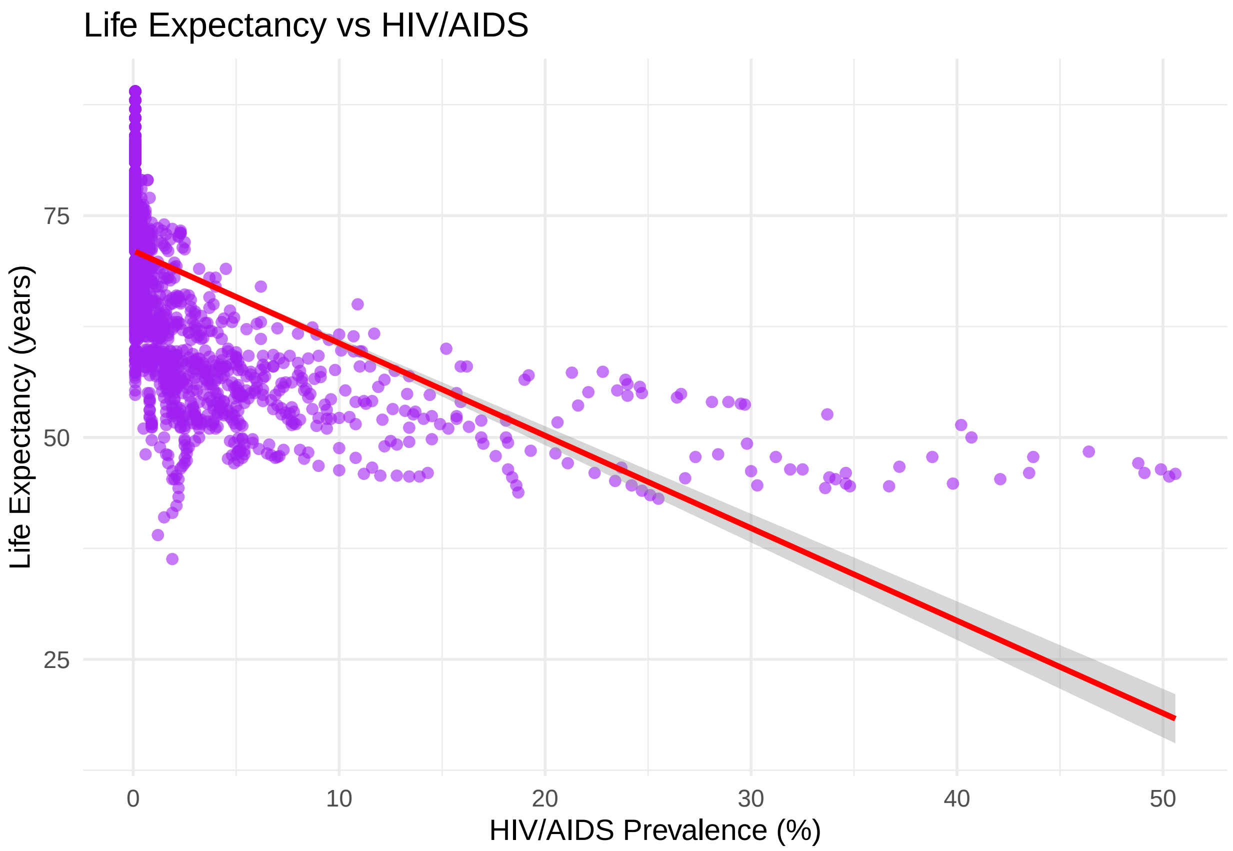
```
## [1] "Correlation between Life Expectancy and GDP:  0.441321809913566"
```

```r
print(paste("Correlation between Life Expectancy and HIV/AIDS: ", cor_test_hiv$estimate)
```

```
## [1] "Correlation between Life Expectancy and HIV/AIDS:  -0.59223629259264"
```

```r
# 3. Relationship between Life Expectancy and Diseases (e.g., HIV/AIDS)
ggplot(data, aes(x = HIV.AIDS, y = Life.expectancy)) +
  geom_point(color = "purple", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Life Expectancy vs HIV/AIDS",
       x = "HIV/AIDS Prevalence (%)",
       y = "Life Expectancy (years)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 10 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

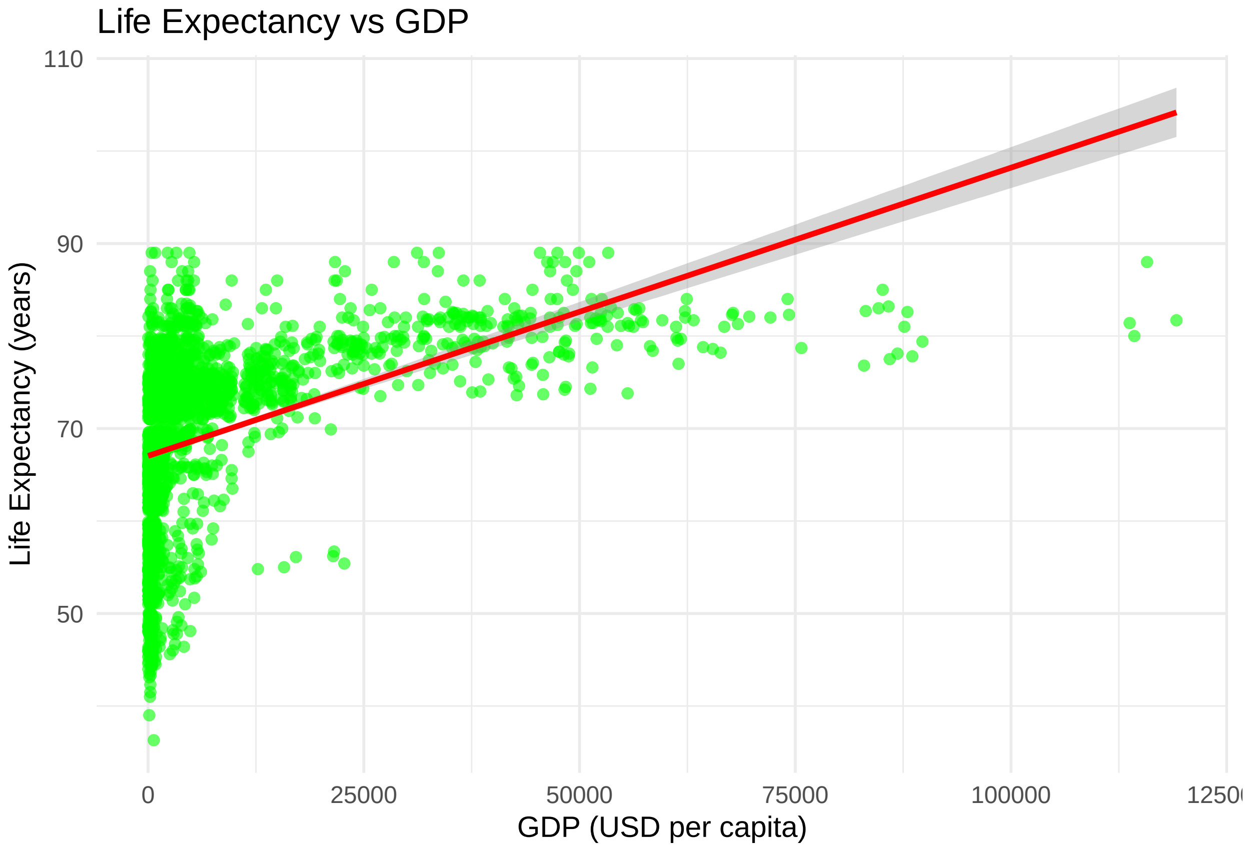


```
# 2. Relationship between Life Expectancy and GDP
ggplot(data, aes(x = GDP, y = Life.expectancy)) +
  geom_point(color = "green", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Life Expectancy vs GDP",
       x = "GDP (USD per capita)",
       y = "Life Expectancy (years)") +
  theme_minimal()
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

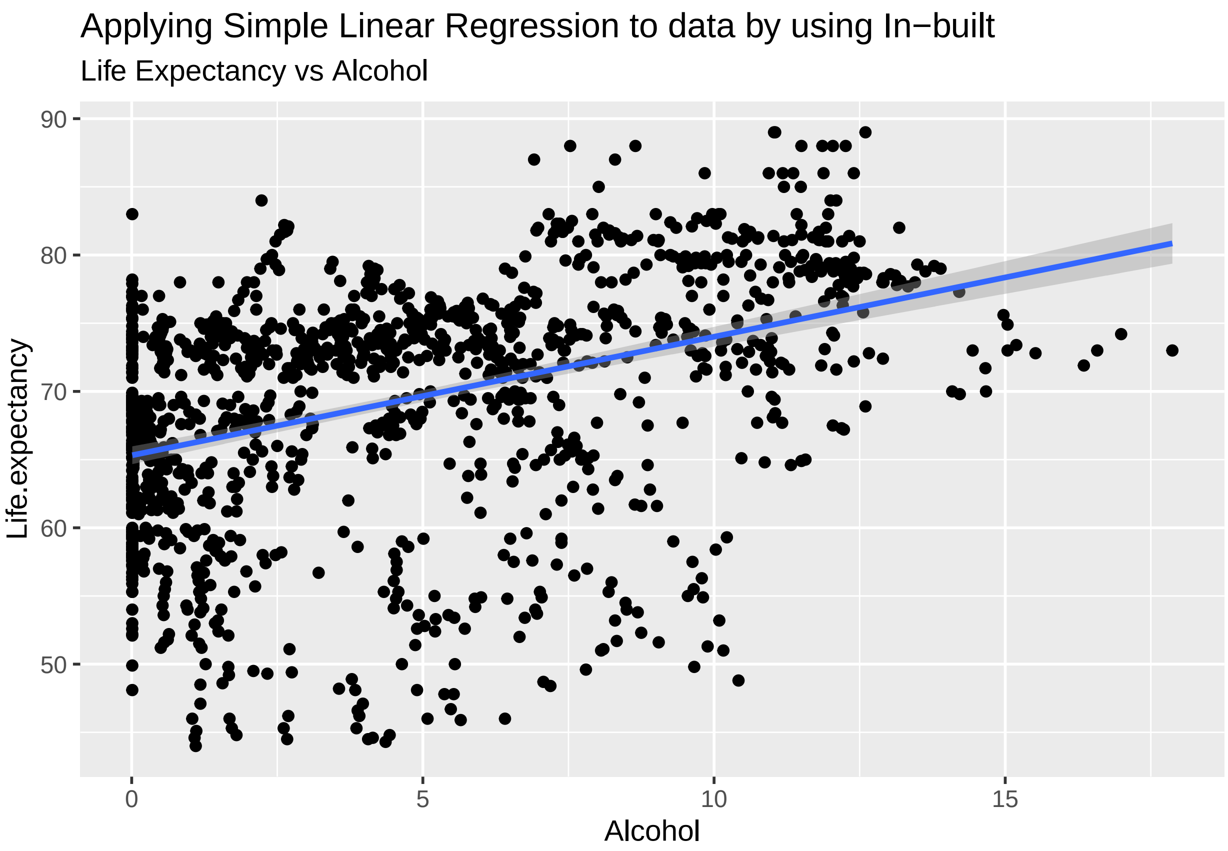
```
## Warning: Removed 453 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 453 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
#USING IN-BUILT
# Visualizing best fit line using In-built
```

```r
ggplot(train, aes(Alcohol, Life.expectancy) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ x) +
  labs(
    title = "Applying Simple Linear Regression to data by using In-built",
    subtitle = "Life Expectancy vs Alcohol"
  )
```



```r
# Inspect the data structure
str(data)
```

```
## 'data.frame':    2938 obs. of  22 variables:
##  $ Country                        : chr  "Afghanistan" "Afghanistan" "Afghanistan" "A
##  $ Year                           : int  2015 2014 2013 2012 2011 2010 2009 2008 2007
##  $ Status                         : chr  "Developing" "Developing" "Developing" "Deve
##  $ Life.expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 5
##  $ Adult.Mortality                : int  263 271 268 272 275 279 281 287 295 295 ...
##  $ infant.deaths                  : int  62 64 66 69 71 74 77 80 82 84 ...
##  $ Alcohol                        : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.02
##  $ percentage.expenditure         : num  71.3 73.5 73.2 78.2 7.1 ...
##  $ Hepatitis.B                    : int  65 62 64 67 68 66 63 64 63 64 ...
##  $ Measles                        : int  1154 492 430 2787 3013 1989 2861 1599 1141 1
##  $ BMI                            : num  19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2
##  $ under.five.deaths              : int  83 86 89 93 97 102 106 110 113 116 ...
##  $ Polio                          : int  6 58 62 67 68 66 63 64 63 58 ...
##  $ Total.expenditure              : num  8.16 8.18 8.13 8.52 7.87 9.2 9.42 8.33 6.73
##  $ Diphtheria                     : int  65 62 64 67 68 66 63 64 63 58 ...
##  $ HIV.AIDS                       : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ GDP                            : num  584.3 612.7 631.7 670 63.5 ...
##  $ Population                     : num  33736494 327582 31731688 3696958 2978599 ...
##  $ thinness..1.19.years           : num  17.2 17.5 17.7 17.9 18.2 18.4 18.6 18.8 19 1
##  $ thinness.5.9.years             : num  17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 1
##  $ Income.composition.of.resources: num  0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.4
##  $ Schooling                      : num  10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
```

```
# Ensure 'Status' is a factor variable
data$Status <- as.factor(data$Status)

# Check the number of categories in each factor variable
cat_levels <- sapply(data, function(x) if (is.factor(x)) length(levels(x)))
print(cat_levels)
```

```
## $Country
## NULL
##
## $Year
## NULL
##
## $Status
## [1] 2
##
## $Life.expectancy
## NULL
##
## $Adult.Mortality
## NULL
##
## $infant.deaths
## NULL
##
## $Alcohol
## NULL
##
## $percentage.expenditure
## NULL
##
## $Hepatitis.B
## NULL
##
## $Measles
## NULL
##
## $BMI
## NULL
##
## $under.five.deaths
## NULL
##
## $Polio
## NULL
##
## $Total.expenditure
## NULL
##
## $Diphtheria
## NULL
##
## $HIV.AIDS
## NULL
##
## $GDP
## NULL
##
## $Population
## NULL
##
## $thinness..1.19.years
## NULL
##
## $thinness.5.9.years
## NULL
##
## $Income.composition.of.resources
## NULL
##
## $Schooling
## NULL
```

```
# Identify and handle categorical predictors with too many levels
# Assuming 'Country' has too many levels, we'll exclude it from the analysis
data <- select(data, -Country)

# Handle missing values by removing rows with NAs
data <- na.omit(data)

# Simplify the regression formula to include all numeric variables and relevant factor
predictors <- paste(names(data)[!names(data) %in% c("Life.expectancy", "Status", "Year"
formula <- as.formula(paste("Life.expectancy ~", predictors))

# Subset data into Developed and Developing countries
developed <- filter(data, Status == "Developed")
developing <- filter(data, Status == "Developing")

# Linear regression model for Developed countries
model_lm_developed <- lm(formula, data = developed)
summary_lm_developed <- summary(model_lm_developed)
print("Linear Regression Summary for Developed Countries:")
```

```
## [1] "Linear Regression Summary for Developed Countries:"
```

```
print(summary_lm_developed)
```

```
## 
## Call:
## lm(formula = formula, data = developed)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.2005 -1.5012 -0.6834  0.6349  9.0709 
## 
## Coefficients: (1 not defined because of singularities)
##                                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                       5.420e+01  5.281e+00  10.263  < 2e-16 ***
## Adult.Mortality                  -1.069e-03  3.684e-03  -0.290  0.77192    
## infant.deaths                    -3.177e-01  5.400e-01  -0.588  0.55688    
## Alcohol                          -1.342e-01  7.579e-02  -1.770  0.07808 .  
## percentage.expenditure            1.429e-04  1.711e-04   0.835  0.40472    
## Hepatitis.B                       1.799e-02  8.804e-03   2.043  0.04220 *  
## Measles                          -1.245e-04  8.840e-05  -1.408  0.16053    
## BMI                              -6.278e-03  9.454e-03  -0.664  0.50730    
## under.five.deaths                 5.082e-01  4.503e-01   1.129  0.26024    
## Polio                           -2.366e-02  2.631e-02  -0.899  0.36946
## Total.expenditure                1.737e-02  7.031e-02   0.247  0.80505
## Diphtheria                      -3.057e-03  2.580e-02  -0.118  0.90580
## HIV.AIDS                                NA         NA      NA       NA
## GDP                             -1.946e-05  2.735e-05  -0.712  0.47747
## Population                       2.939e-09  1.148e-08   0.256  0.79814
## thinness..1.19.years            -3.309e+00  2.075e+00  -1.595  0.11214
## thinness.5.9.years               1.025e+00  1.869e+00   0.548  0.58408
## Income.composition.of.resources  4.349e+01  6.398e+00   6.797 9.54e-11 ***
## Schooling                       -3.998e-01  1.462e-01  -2.735  0.00674 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.516 on 224 degrees of freedom
## Multiple R-squared:  0.6779, Adjusted R-squared:  0.6534
## F-statistic: 27.73 on 17 and 224 DF,  p-value: < 2.2e-16
```

```
# Linear regression model for Developing countries
model_lm_developing <- lm(formula, data = developing)
summary_lm_developing <- summary(model_lm_developing)
print("Linear Regression Summary for Developing Countries:")
```

```
## [1] "Linear Regression Summary for Developing Countries:"
```

```
print(summary_lm_developing)
```

```
##
## Call:
## lm(formula = formula, data = developing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7350  -2.0930   0.0149   2.3857  12.0373
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.296e+01  7.792e-01  67.968  < 2e-16 ***
## Adult.Mortality                 -1.605e-02  9.743e-04 -16.469  < 2e-16 ***
## infant.deaths                    9.393e-02  1.070e-02   8.777  < 2e-16 ***
## Alcohol                         -1.154e-01  3.589e-02  -3.214  0.00134 **
## percentage.expenditure           9.775e-04  2.981e-04   3.279  0.00107 **
## Hepatitis.B                     -5.338e-03  4.916e-03  -1.086  0.27775
## Measles                         -3.761e-06  1.086e-05  -0.346  0.72911
## BMI                                4.001e-02  6.834e-03   5.855 5.95e-09 ***
## under.five.deaths                 -7.060e-02  7.747e-03  -9.114  < 2e-16 ***
## Polio                              6.665e-03  5.224e-03   1.276  0.20222
## Total.expenditure                  7.322e-02  4.596e-02   1.593  0.11141
## Diphtheria                         1.488e-02  6.128e-03   2.427  0.01534 *
## HIV.AIDS                          -4.405e-01  1.797e-02 -24.509  < 2e-16 ***
## GDP                               -8.430e-06  4.508e-05  -0.187  0.85168
## Population                        -1.254e-09  1.765e-09  -0.711  0.47746
## thinness..1.19.years              -4.149e-04  5.311e-02  -0.008  0.99377
## thinness.5.9.years                -2.929e-02  5.256e-02  -0.557  0.57736
## Income.composition.of.resources  8.501e+00  8.468e-01  10.039  < 2e-16 ***
## Schooling                          9.323e-01  6.630e-02  14.063  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.588 on 1388 degrees of freedom
## Multiple R-squared:  0.8179, Adjusted R-squared:  0.8155
## F-statistic: 346.3 on 18 and 1388 DF,  p-value: < 2.2e-16
```

```
# Random Forest regression using ranger
# Developed countries
rf_developed <- ranger(formula, data = developed, num.trees = 500, mtry = 3, importance
print("Random Forest Summary for Developed Countries:")
```

```
## [1] "Random Forest Summary for Developed Countries:"
```

```
print(rf_developed)
```

```
## Ranger result
##
## Call:
##  ranger(formula, data = developed, num.trees = 500, mtry = 3,      importance = "impu
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      242
## Number of independent variables:  18
## Mtry:                             3
## Target node size:                 5
## Variable importance mode:         impurity
## Splitrule:                        variance
## OOB prediction error (MSE):       3.564952
## R squared (OOB):                  0.8047957
```

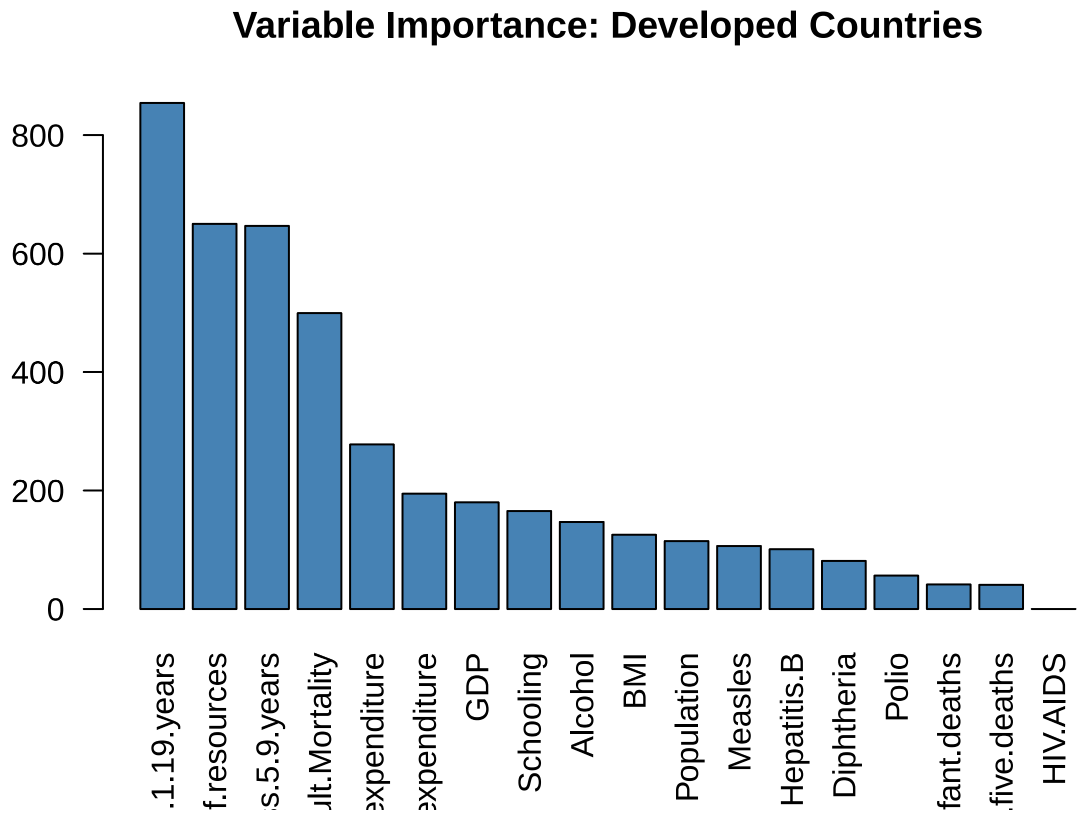
```
# Developing countries
rf_developing <- ranger(formula, data = developing, num.trees = 500, mtry = 3, importanc
print("Random Forest Summary for Developing Countries:")
```

```
## [1] "Random Forest Summary for Developing Countries:"
```

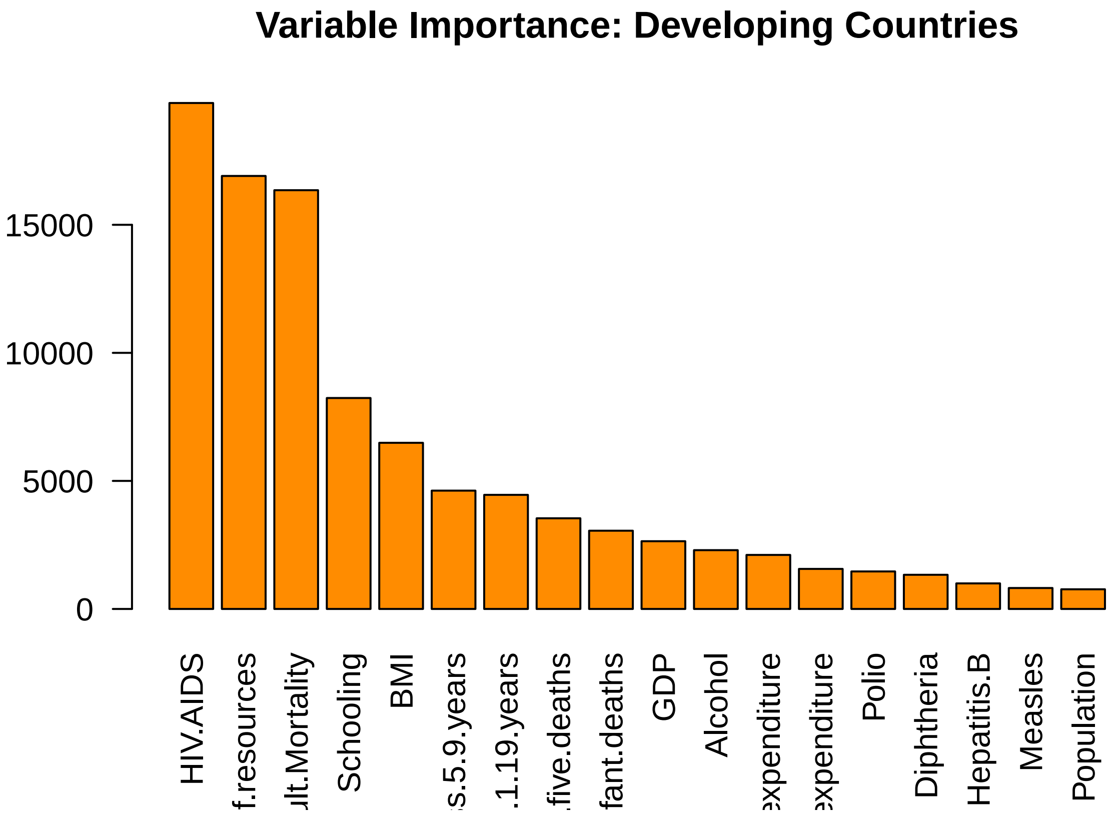
```
print(rf_developing)
```

```
## Ranger result
##
## Call:
##  ranger(formula, data = developing, num.trees = 500, mtry = 3,      importance = "imp
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      1407
## Number of independent variables:  18
## Mtry:                             3
## Target node size:                 5
## Variable importance mode:         impurity
## Splitrule:                        variance
## OOB prediction error (MSE):       3.360522
## R squared (OOB):                  0.9518474
```

```
# Visualize Variable Importance for Random Forest Models
importance_developed <- rf_developed$variable.importance
importance_developing <- rf_developing$variable.importance

# Plotting variable importance
barplot(sort(importance_developed, decreasing = TRUE), main = "Variable Importance: Deve
```



```r
barplot(sort(importance_developing, decreasing = TRUE), main = "Variable Importance: Dev
```

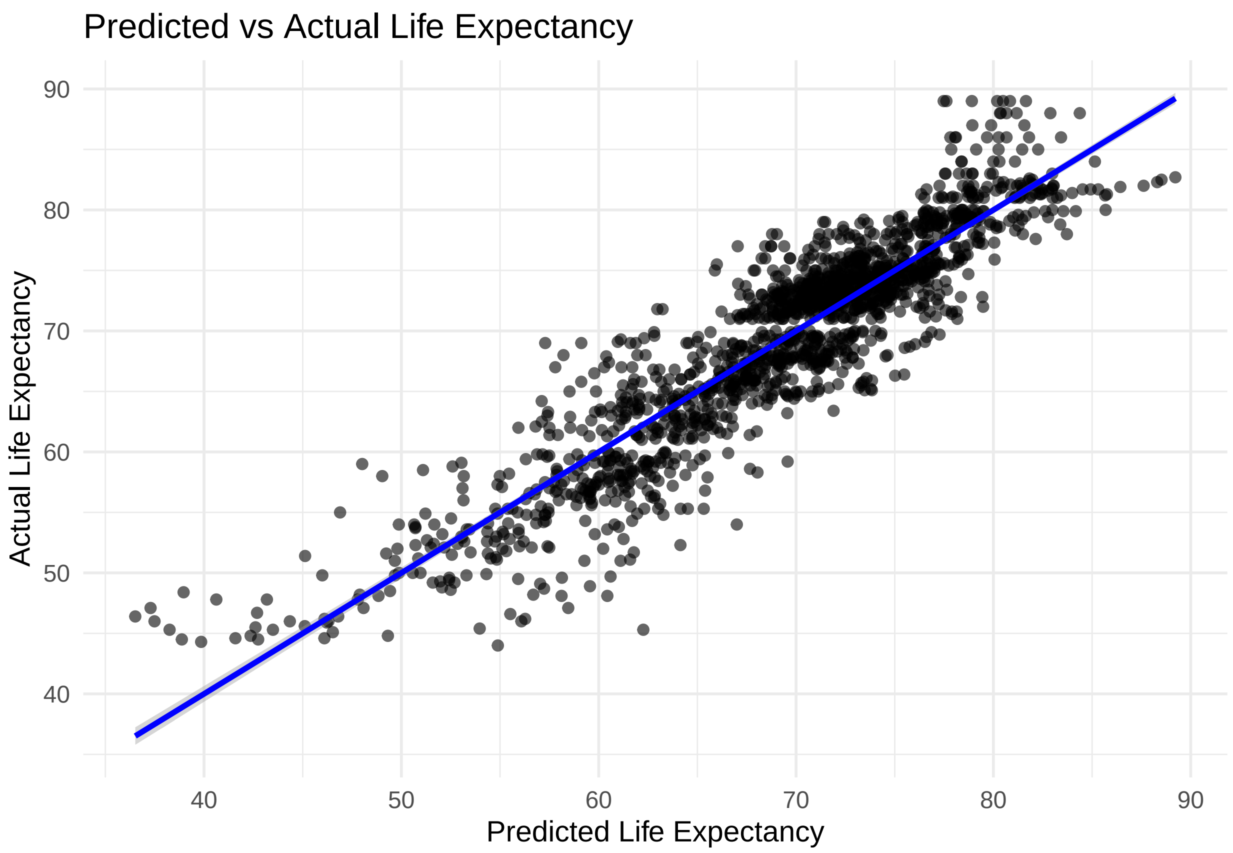


```r
# Inspect the data structure
str(data)
```

```
## 'data.frame':    1649 obs. of  21 variables:
##  $ Year                           : int  2015 2014 2013 2012 2011 2010 2009 2008 2007
##  $ Status                         : Factor w/ 2 levels "Developed","Developing": 2 2
##  $ Life.expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 5
##  $ Adult.Mortality                : int  263 271 268 272 275 279 281 287 295 295 ...
##  $ infant.deaths                  : int  62 64 66 69 71 74 77 80 82 84 ...
##  $ Alcohol                        : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.02
##  $ percentage.expenditure         : num  71.3 73.5 73.2 78.2 7.1 ...
##  $ Hepatitis.B                    : int  65 62 64 67 68 66 63 64 63 64 ...
##  $ Measles                        : int  1154 492 430 2787 3013 1989 2861 1599 1141 1
##  $ BMI                            : num  19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2
##  $ under.five.deaths              : int  83 86 89 93 97 102 106 110 113 116 ...
##  $ Polio                          : int  6 58 62 67 68 66 63 64 63 58 ...
##  $ Total.expenditure              : num  8.16 8.18 8.13 8.52 7.87 9.2 9.42 8.33 6.73
##  $ Diphtheria                     : int  65 62 64 67 68 66 63 64 63 58 ...
##  $ HIV.AIDS                       : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ GDP                            : num  584.3 612.7 631.7 670 63.5 ...
##  $ Population                     : num  33736494 327582 31731688 3696958 2978599 ...
##  $ thinness..1.19.years           : num  17.2 17.5 17.7 17.9 18.2 18.4 18.6 18.8 19 1
##  $ thinness.5.9.years             : num  17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 1
##  $ Income.composition.of.resources: num  0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.4
##  $ Schooling                      : num  10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
##  - attr(*, "na.action")= ’omit’ Named int [1:1289] 33 45 46 47 48 49 58 59 60 61 ...
##   ..- attr(*, "names")= chr [1:1289] "33" "45" "46" "47" ...
```

```r
# Handle missing values xby removing rows with NAs
data <- na.omit(data)

# Ensure 'Status' is a factor variable
data$Status <- as.factor(data$Status)

# Define the regression formula: Life expectancy predicted by all other variables
predictors <- paste(names(data)[!names(data) %in% c("Life.expectancy", "Country", "Year'
formula <- as.formula(paste("Life.expectancy ~", predictors))

# Fit the regression model
model <- lm(formula, data = data)
data$predicted <- predict(model, data)
ggplot(data, aes(x = predicted, y = Life.expectancy)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Predicted vs Actual Life Expectancy",
       x = "Predicted Life Expectancy",
       y = "Actual Life Expectancy") +
  theme_minimal()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```r
# Alternatively, you can plot residuals to check for patterns:


# Summary of the regression model
summary_model <- summary(model)
print("Linear Regression Summary:")
```

```
## [1] "Linear Regression Summary:"
```

```r
print(summary_model)
```

```
##
## Call:
## lm(formula = formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9597  -2.0621  -0.0147   2.2751  11.7115
##
## Coefficients:
```

```
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.445e+01  8.400e-01  64.822  < 2e-16 ***
## StatusDeveloping               -9.684e-01  3.379e-01  -2.865  0.00422 **
## Adult.Mortality                -1.663e-02  9.494e-04 -17.517  < 2e-16 ***
## infant.deaths                   9.350e-02  1.065e-02   8.777  < 2e-16 ***
## Alcohol                        -9.140e-02  3.316e-02  -2.756  0.00592 **
## percentage.expenditure          3.673e-04  1.801e-04   2.040  0.04156 *
## Hepatitis.B                    -6.525e-03  4.449e-03  -1.467  0.14265
## Measles                        -7.865e-06  1.079e-05  -0.729  0.46597
## BMI                             3.376e-02  5.998e-03   5.628 2.15e-08 ***
## under.five.deaths              -7.035e-02  7.711e-03  -9.123  < 2e-16 ***
## Polio                           7.935e-03  5.152e-03   1.540  0.12370
## Total.expenditure               7.586e-02  4.067e-02   1.865  0.06236 .
## Diphtheria                      1.490e-02  5.928e-03   2.513  0.01205 *
## HIV.AIDS                       -4.370e-01  1.784e-02 -24.490  < 2e-16 ***
## GDP                             8.738e-06  2.837e-05   0.308  0.75813
## Population                     -6.425e-10  1.749e-09  -0.367  0.71337
## thinness..1.19.years           -1.238e-02  5.300e-02  -0.234  0.81527
## thinness.5.9.years             -4.798e-02  5.231e-02  -0.917  0.35917
## Income.composition.of.resources  9.817e+00  8.321e-01  11.797  < 2e-16 ***
## Schooling                       8.665e-01  5.940e-02  14.587  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.588 on 1629 degrees of freedom
## Multiple R-squared:  0.8356, Adjusted R-squared:  0.8336
## F-statistic: 435.7 on 19 and 1629 DF,  p-value: < 2.2e-16
```

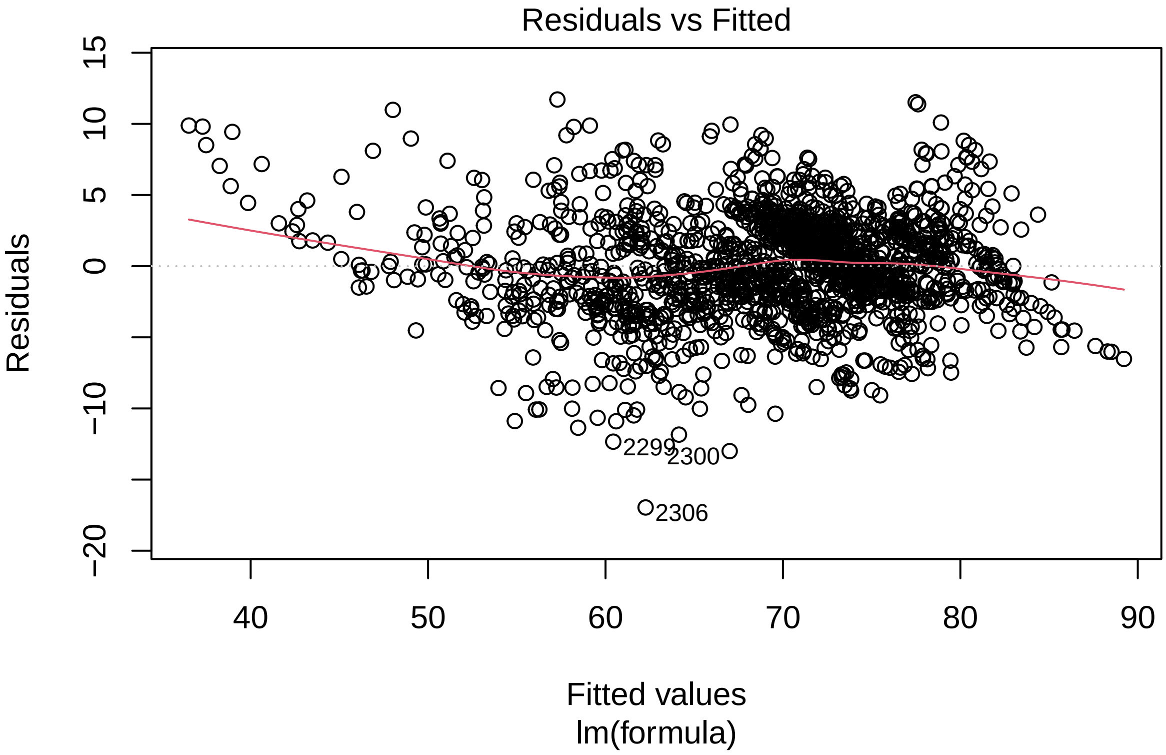
```r
# Extract coefficients and p-values
coefficients <- summary_model$coefficients
print("Coefficients and p-values:")
```

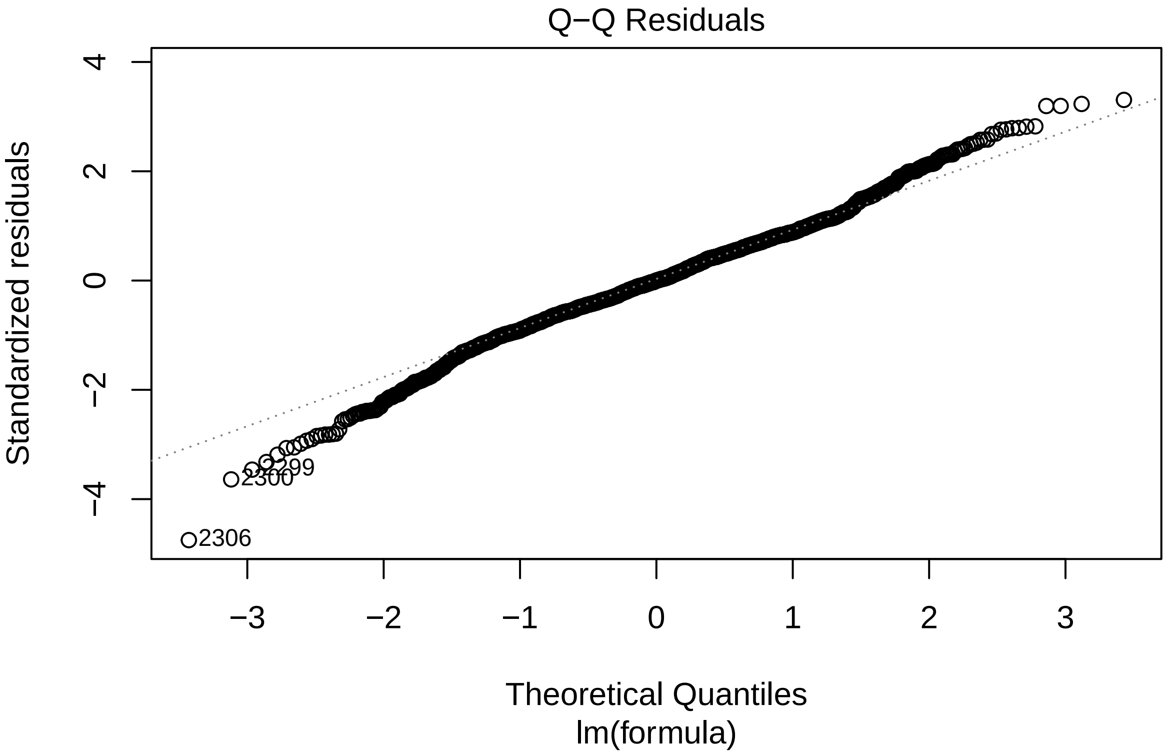
```
## [1] "Coefficients and p-values:"
```

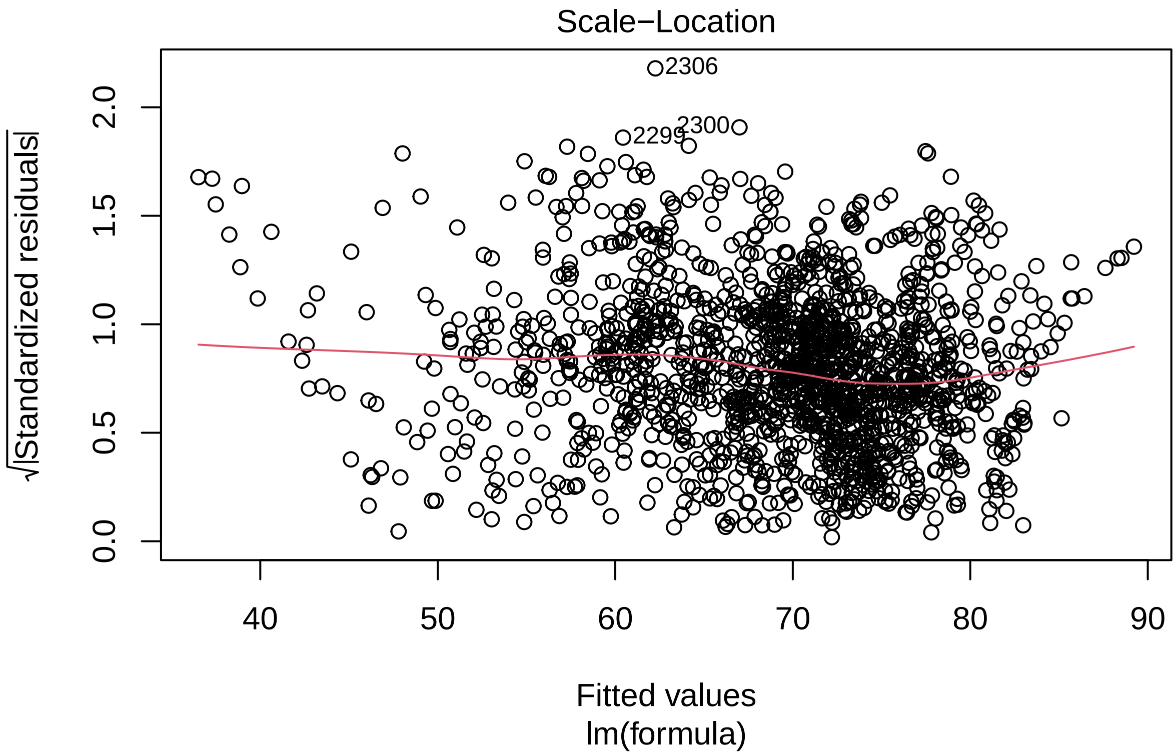
```r
print(coefficients)
```

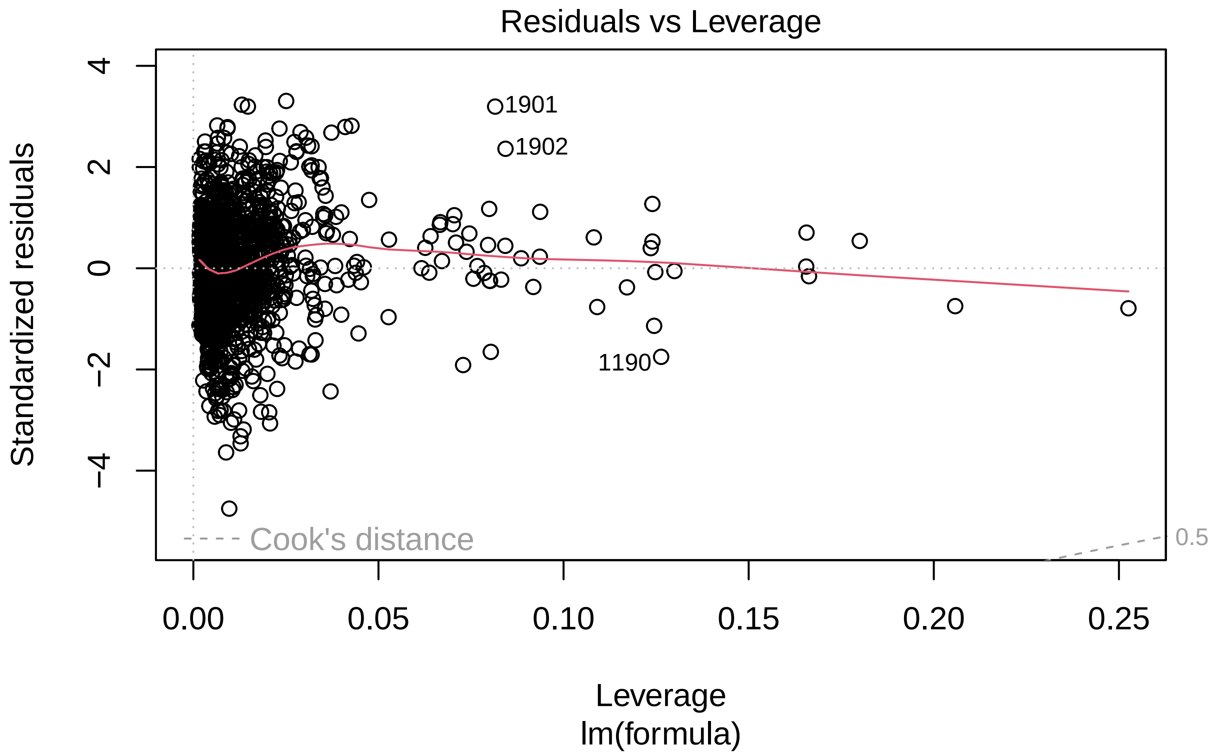
```
##                                      Estimate   Std. Error     t value
## (Intercept)                      5.445111e+01 8.400125e-01  64.8217869
## StatusDeveloping                -9.683668e-01 3.379401e-01  -2.8654984
## Adult.Mortality                 -1.663174e-02 9.494415e-04 -17.5173908
## infant.deaths                    9.349971e-02 1.065327e-02   8.7766178
## Alcohol                         -9.139501e-02 3.316341e-02  -2.7558989
## percentage.expenditure           3.673363e-04 1.801074e-04   2.0395401
## Hepatitis.B                     -6.524647e-03 4.448550e-03  -1.4666908
## Measles                         -7.865434e-06 1.078595e-05  -0.7292296
## BMI                              3.375565e-02 5.998098e-03   5.6277248
## under.five.deaths               -7.034836e-02 7.711285e-03  -9.1227806
## Polio                            7.935254e-03 5.151979e-03   1.5402342
## Total.expenditure                7.585822e-02 4.067445e-02   1.8650089
## Diphtheria                       1.489933e-02 5.927886e-03   2.5134300
## HIV.AIDS                        -4.369640e-01 1.784256e-02 -24.4899830
## GDP                              8.737938e-06 2.837049e-05   0.3079939
## Population                      -6.424645e-10 1.748706e-09  -0.3673942
## thinness..1.19.years            -1.238499e-02 5.300123e-02  -0.2336736
## thinness.5.9.years              -4.798356e-02 5.231461e-02  -0.9172113
## Income.composition.of.resources  9.816570e+00 8.321251e-01  11.7969891
## Schooling                        8.665032e-01 5.940312e-02  14.5868300
##                                      Pr(>|t|)
## (Intercept)                      0.000000e+00
## StatusDeveloping                 4.216969e-03
## Adult.Mortality                  4.421388e-63
## infant.deaths                    4.167811e-18
## Alcohol                          5.918326e-03
## percentage.expenditure           4.155722e-02
## Hepatitis.B                      1.426532e-01
## Measles                          4.659661e-01
## BMI                              2.146026e-08
## under.five.deaths                2.098456e-19
## Polio                            1.236975e-01
## Total.expenditure                6.235983e-02
## Diphtheria                       1.205217e-02
## HIV.AIDS                        4.926216e-113
## GDP                              7.581264e-01
## Population                       7.133727e-01
## thinness..1.19.years             8.152678e-01
## thinness.5.9.years               3.591677e-01
## Income.composition.of.resources  7.022005e-31
## Schooling                        2.172561e-45
```

```r
plot(model)
```


Residuals vs Fitted
Residuals
15
10
5
0
−10
−20
40
50
60
70
80
90
2299
2300
2306
Fitted values
lm(formula)

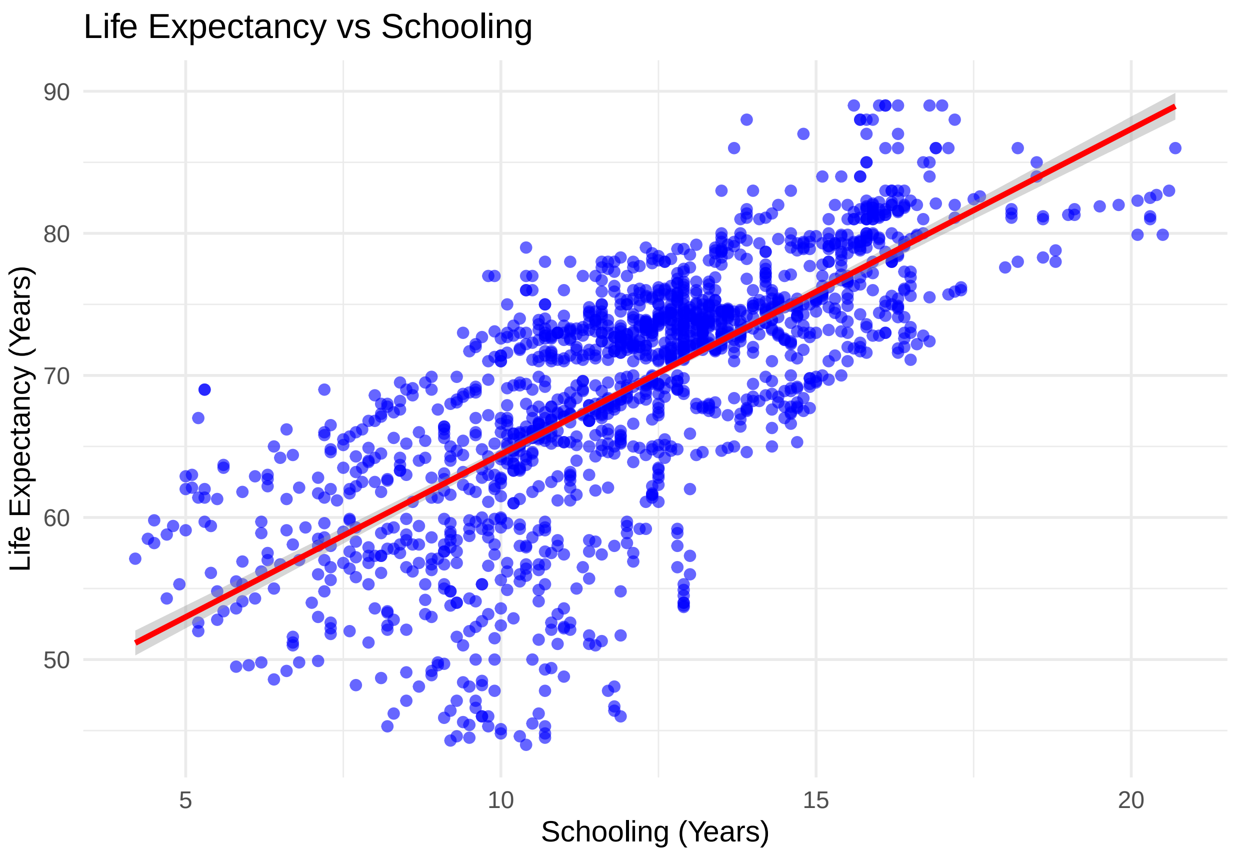

Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
2299
2300
2306
Theoretical Quantiles
lm(formula)


Scale−Location
2306
2299
2300
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
40
50
60
70
80
90
Fitted values
lm(formula)



```
# Scatterplot with regression line
ggplot(data_clean, aes(x = Schooling, y = Life.expectancy)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Life Expectancy vs Schooling",
       x = "Schooling (Years)",
       y = "Life Expectancy (Years)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# --- Save Cleaned Dataset for Future Use ---
write.csv(data_clean, "cleaned_life_expectancy_data.csv", row.names = FALSE)
```