# INC 491: Data Science and Intelligent Techniques

# Mini Project: Life Expectancy (WHO) Analysis

Nontawit Markjan 59070504003 AE

To

Dr. Issarapong Khuankrue

This report is part of the subject

INC 491 - Data Science and Intelligent Techniques

Developed by Automation Engineering Students

King Mongkut's University of Technology Thonburi

Semester 1/2019

# Index

Page

**1. Background information** **4**

**2. Data manipulation** **6**

2.1 Import 6

2.2 Tidy 7

2.3 Transform 9

2.4 Visualize 10

2.4.1 Percentage of country status 10

2.4.2 Comparison between developing and developed countries 10

2.4.2.1 Life expectancy 11

2.4.2.2 Adult mortality 11

2.4.2.3 Infant deaths 12

2.4.2.4 Alcohol 12

2.4.2.5 Percentage expenditure 13

2.4.2.6 Hepatitis B 13

2.4.2.7 Measles 14

2.4.2.8 BMI 14

2.4.2.9 Under-5 deaths 15

2.4.2.10 Polio 15

2.4.2.11 Total expenditure 16

2.4.2.12 Diphtheria 16

2.4.2.13 HIV/AIDS 17

2.4.2.14 GDP 17

2.4.2.15 Population 18

2.4.2.16 Thinness 1-19 years 18

2.4.2.17 Thinness 5-9 years 19

2.4.2.18 Income composition of resources 19

2.4.2.19 Schooling 20

Page

2.4.3 Effect of some factors to life expectancy 20

2.4.3.1 Adult mortality 21

2.4.3.2 Infant Deaths 21

2.4.3.3 Alcohol 22

2.4.3.4 Population 22

2.4.3.5 Schooling 23

2.4.4 Correlation 23

2.5 Modelling 25

2.5.1 Create train and test dataset 25

2.5.2 Multiple linear regression 25

2.5.3 Random forest 26

2.5.4 Gradient boosting 27

2.5.5 Supported vector machine (SVM) 28

2.6 Accuracy 28

2.6.1 Mean-squared error (MSE) 29

2.6.2 Root-mean-squared error (RMSE) 29

2.6.3 Min/max accuracy 30

**3. Summary** **30**

**4. Appendix** **31**

## 1. Background information

This mini project studies the factors affecting life expectancy by using dataset from the Global Health Observatory (GHO), which is World Health Organization's (WHO) gateway to health-related statistics repository. It keeps track of the health status as well as many other related factors for all countries. The data-sets are available to public for health data analysis. The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website.

The dataset consists of the factors from year 2000 to 2015. There are some missing data from less known countries such as Vanuatu, Tonga, Togo, Cabo Verde etc. This dataset is the final version that was merged from the different source. It consists of 22 columns and 2,938 rows. The details of each column are shown below.

| **Columns** | **Definition** |
|---|---|
| 'Country' | Country |
| 'Year' | Year |
| 'Status' | Developed or Developing status |
| 'Life expectancy' | Life Expectancy in age |
| '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' | Alcohol, recorded per capita (15+) consumption (in liters of pure alcohol) |
| 'percentage expenditure' | Expenditure on health as a percentage of Gross Domestic Product per capita (%) |
| 'Hepatitis B' | Hepatitis B (Hep B) immunization coverage among 1-year-olds (%) |
| 'Measles' | Measles, number of reported cases 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’ | Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%) |
| ‘HIV/AIDS’ | Deaths per 1 000 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) |

Table 1: columns definition

## 2. Data manipulation

Data manipulation in this project is based on the below process where the red part represents exploratory Data Analysis (EDA) - the process for data investigation which discovers the pattern, structure, details, and maximize the insight of the given data. It is the philosophy on how the data should be manipulated. Modelling and analysis are the blue part.

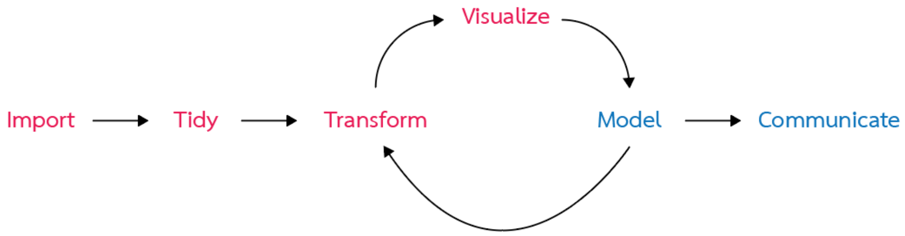


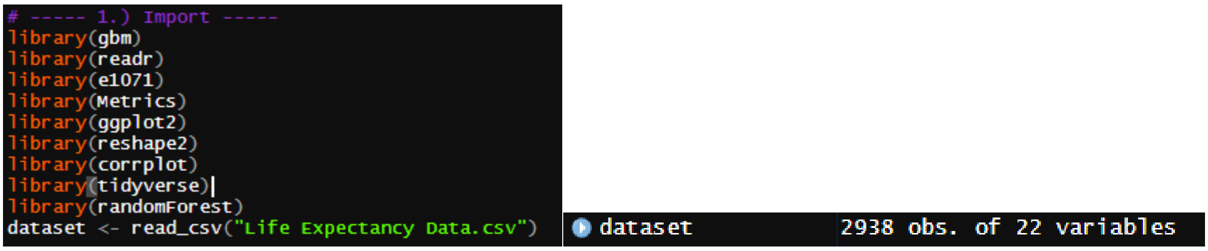
Figure 1: Data exploration process

### 2.1 Import

Life Expectancy Data.csv is the dataset for this mini project

Figure 2: Dataset

The dataset was imported as data frame in the project with this code by using 'read_csv' library to read comma separated value file. The libraries for data manipulation are also called.



```
# ----- 1.) Import -----
library(gbm)
library(readr)
library(e1071)
library(Metrics)
library(ggplot2)
library(reshape2)
library(corrplot)
library(tidyverse)
library(randomForest)
dataset <- read_csv("Life Expectancy Data.csv")
```

```
dataset          2938 obs. of 22 variables
```

Figure 3: Code for import comma separated value files (.csv) with libraries at the left and data frame as results at the right

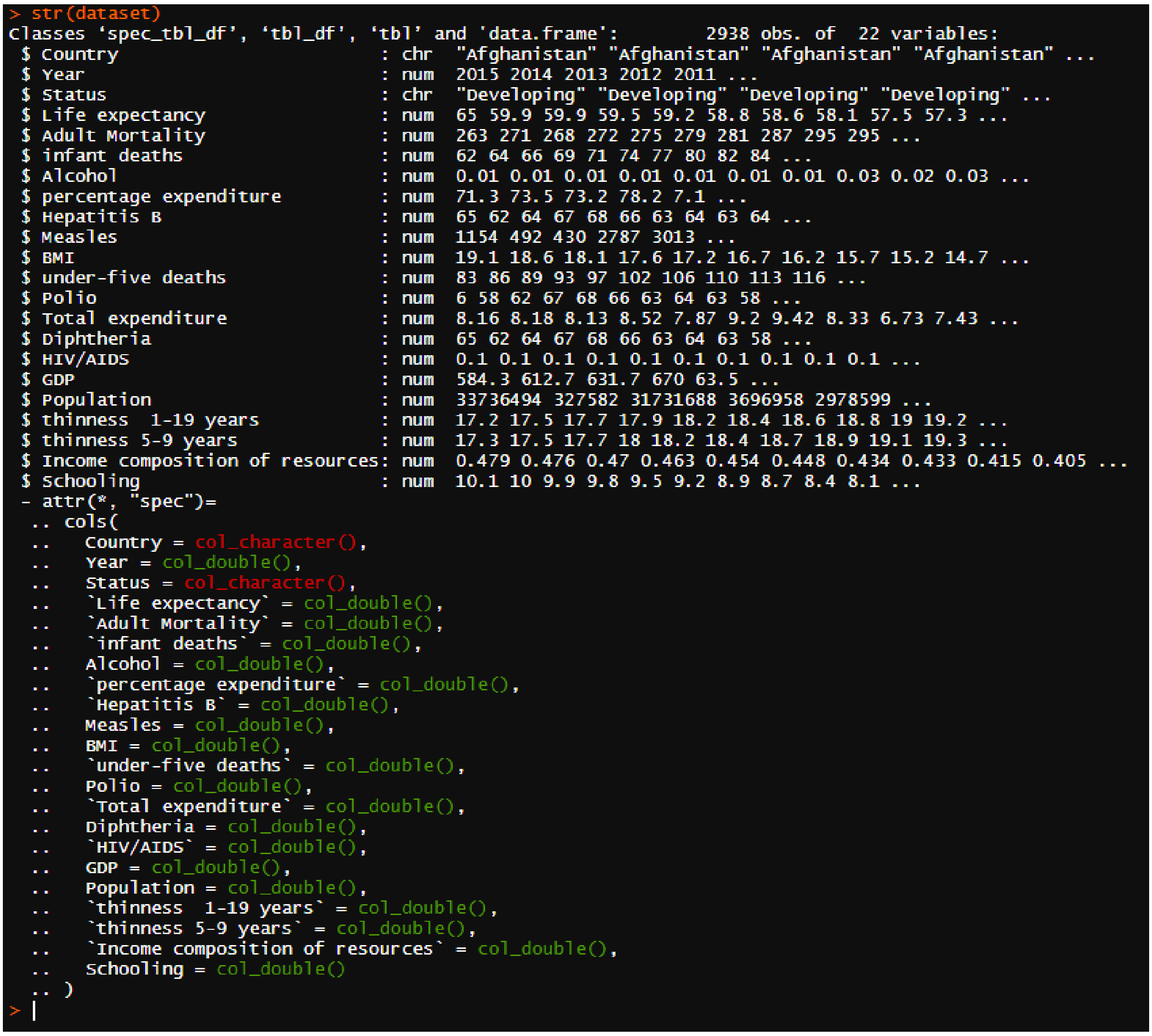
2.2 Tidy

The datasets were observed by using these codes.

‘str(dataset)’ – observe the structure of R object, which is ‘dataset’ in this case.

‘summary(dataset)’ – produce the summary details of each attribute in dataset

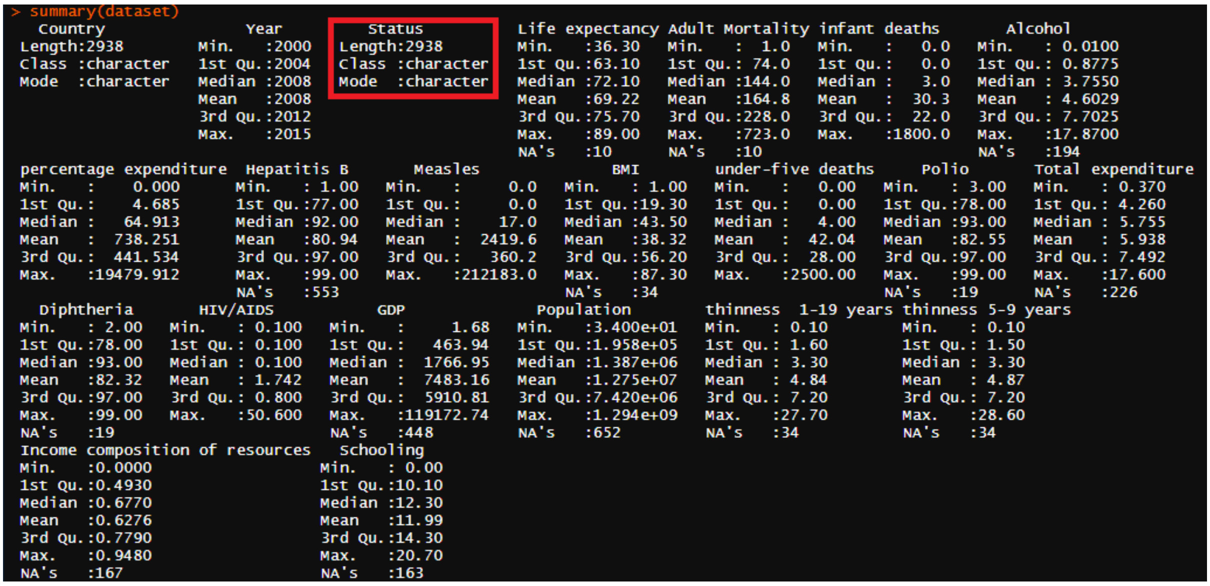
‘View(dataset)’ – show all entries of dataframe



```
> str(dataset)
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':	2938 obs. of  22 variables:
 $ Country                        : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ Year                           : num  2015 2014 2013 2012 2011 ...
 $ Status                         : chr  "Developing" "Developing" "Developing" "Developing" ...
 $ Life expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 57.3 ...
 $ Adult Mortality                : num  263 271 268 272 275 279 281 287 295 295 ...
 $ infant deaths                  : num  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 0.03 ...
 $ percentage expenditure         : num  71.3 73.5 73.2 78.2 7.1 ...
 $ Hepatitis B                    : num  65 62 64 67 68 66 63 64 63 64 ...
 $ Measles                        : num  1154 492 430 2787 3013 ...
 $ BMI                            : num  19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2 14.7 ...
 $ under-five deaths              : num  83 86 89 93 97 102 106 110 113 116 ...
 $ Polio                          : num  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 7.43 ...
 $ Diphtheria                     : num  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 19.2 ...
 $ thinness 5-9 years             : num  17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 19.3 ...
 $ Income composition of resources: num  0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.433 0.415 0.405 ...
 $ Schooling                      : num  10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
 - attr(*, "spec")=
  .. cols(
  ..   Country = col_character(),
  ..   Year = col_double(),
  ..   Status = col_character(),
  ..   `Life expectancy` = col_double(),
  ..   `Adult Mortality` = col_double(),
  ..   `infant deaths` = col_double(),
  ..   Alcohol = col_double(),
  ..   `percentage expenditure` = col_double(),
  ..   `Hepatitis B` = col_double(),
  ..   Measles = col_double(),
  ..   BMI = col_double(),
  ..   `under-five deaths` = col_double(),
  ..   Polio = col_double(),
  ..   `Total expenditure` = col_double(),
  ..   Diphtheria = col_double(),
  ..   `HIV/AIDS` = col_double(),
  ..   GDP = col_double(),
  ..   Population = col_double(),
  ..   `thinness  1-19 years` = col_double(),
  ..   `thinness 5-9 years` = col_double(),
  ..   `Income composition of resources` = col_double(),
  ..   Schooling = col_double()
  .. )
> |
```

Figure 4: Code for showing ‘dataset’ data frame structure

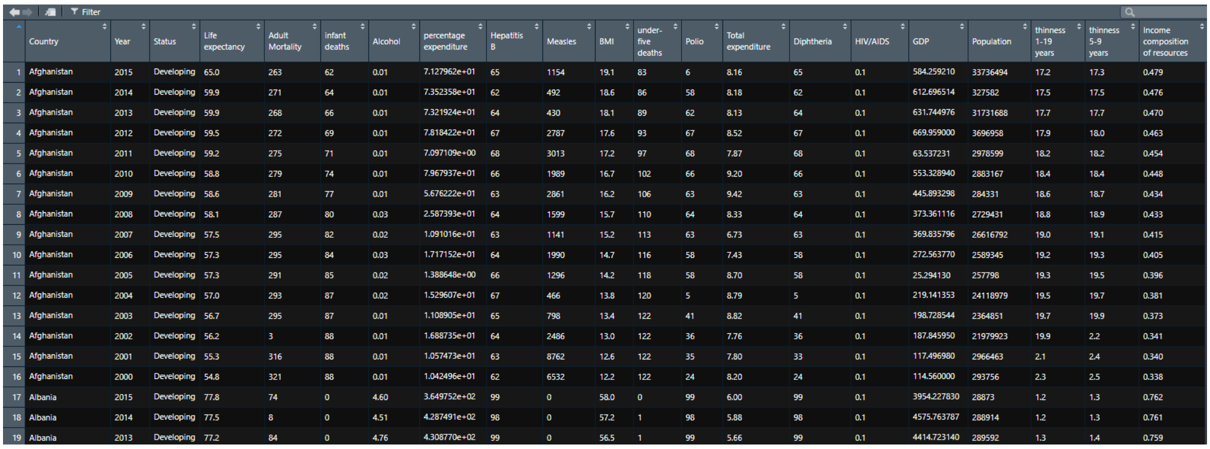
In addition, we can observe that some variables have inappropriate data type, which is ‘Status’. It should be set as the factor.



```
> summary(dataset)
   Country               Year         Status          Life expectancy Adult Mortality infant deaths       Alcohol       
 Length:2938        Min.   :2000   Length:2938        Min.   :36.30   Min.   :  1.0   Min.   :   0.0   Min.   : 0.0100  
 Class :character   1st Qu.:2004   Class :character   1st Qu.:63.10   1st Qu.: 74.0   1st Qu.:   0.0   1st Qu.: 0.8775  
 Mode  :character   Median :2008   Mode  :character   Median :72.10   Median :144.0   Median :   3.0   Median : 3.7550  
                    Mean   :2008                      Mean   :69.22   Mean   :164.8   Mean   :  30.3   Mean   : 4.6029  
                    3rd Qu.:2012                      3rd Qu.:75.70   3rd Qu.:228.0   3rd Qu.:  22.0   3rd Qu.: 7.7025  
                    Max.   :2015                      Max.   :89.00   Max.   :723.0   Max.   :1800.0   Max.   :17.8700  
                                                      NA's   :10      NA's   :10                       NA's   :194      
 percentage expenditure  Hepatitis B       Measles               BMI        under-five deaths     Polio       Total expenditure
 Min.   :    0.000      Min.   : 1.00   Min.   :     0.0   Min.   : 1.00   Min.   :   0.00   Min.   : 3.00   Min.   : 0.370   
 1st Qu.:    4.685      1st Qu.:77.00   1st Qu.:     0.0   1st Qu.:19.30   1st Qu.:   0.00   1st Qu.:78.00   1st Qu.: 4.260   
 Median :   64.913      Median :92.00   Median :    17.0   Median :43.50   Median :   4.00   Median :93.00   Median : 5.755   
 Mean   :  738.251      Mean   :80.94   Mean   :  2419.6   Mean   :38.32   Mean   :  42.04   Mean   :82.55   Mean   : 5.938   
 3rd Qu.:  441.534      3rd Qu.:97.00   3rd Qu.:   360.2   3rd Qu.:56.20   3rd Qu.:  28.00   3rd Qu.:97.00   3rd Qu.: 7.492   
 Max.   :19479.912      Max.   :99.00   Max.   :212183.0   Max.   :87.30   Max.   :2500.00   Max.   :99.00   Max.   :17.600   
                        NA's   :553                        NA's   :34                        NA's   :19      NA's   :226      
   Diphtheria       HIV/AIDS            GDP              Population        thinness  1-19 years thinness 5-9 years
 Min.   : 2.00   Min.   : 0.100   Min.   :     1.68   Min.   :3.400e+01   Min.   : 0.10        Min.   : 0.10       
 1st Qu.:78.00   1st Qu.: 0.100   1st Qu.:   463.94   1st Qu.:1.958e+05   1st Qu.: 1.60        1st Qu.: 1.50       
 Median :93.00   Median : 0.100   Median :  1766.95   Median :1.387e+06   Median : 3.30        Median : 3.30       
 Mean   :82.32   Mean   : 1.742   Mean   :  7483.16   Mean   :1.275e+07   Mean   : 4.84        Mean   : 4.87       
 3rd Qu.:97.00   3rd Qu.: 0.800   3rd Qu.:  5910.81   3rd Qu.:7.420e+06   3rd Qu.: 7.20        3rd Qu.: 7.20       
 Max.   :99.00   Max.   :50.600   Max.   :119172.74   Max.   :1.294e+09   Max.   :27.70        Max.   :28.60       
 NA's   :19                       NA's   :448         NA's   :652         NA's   :34           NA's   :34          
 Income composition of resources   Schooling    
 Min.   :0.0000                  Min.   : 0.00  
 1st Qu.:0.4930                  1st Qu.:10.10  
 Median :0.6770                  Median :12.30  
 Mean   :0.6276                  Mean   :11.99  
 3rd Qu.:0.7790                  3rd Qu.:14.30  
 Max.   :0.9480                  Max.   :20.70  
 NA's   :167                     NA's   :163    
```

Figure 5: Code for showing 'dataset' data frame summary

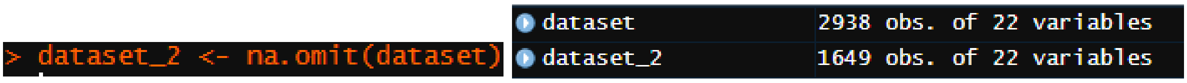
For 'view' function, we can clearly see the data inside the data frame. The structure is the same as we observed by using 'str' function but 'view' does not tell the data type of each variable.



| | Country | Year | Status | Life expectancy | Adult Mortality | infant deaths | Alcohol | percentage expenditure | Hepatitis B | Measles | BMI | under-five deaths | Polio | Total expenditure | Diphtheria | HIV/AIDS | GDP | Population | thinness 1-19 years | thinness 5-9 years | Income composition of resources |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Afghanistan | 2015 | Developing | 65.0 | 263 | 62 | 0.01 | 7.127962e+01 | 65 | 1154 | 19.1 | 83 | 6 | 8.16 | 65 | 0.1 | 584.259210 | 33736494 | 17.2 | 17.3 | 0.479 |
| 2 | Afghanistan | 2014 | Developing | 59.9 | 271 | 64 | 0.01 | 7.352358e+01 | 62 | 492 | 18.6 | 86 | 58 | 8.18 | 62 | 0.1 | 612.696514 | 327582 | 17.5 | 17.5 | 0.476 |
| 3 | Afghanistan | 2013 | Developing | 59.9 | 268 | 66 | 0.01 | 7.321924e+01 | 64 | 430 | 18.1 | 89 | 62 | 8.13 | 64 | 0.1 | 631.744976 | 31731688 | 17.7 | 17.7 | 0.470 |
| 4 | Afghanistan | 2012 | Developing | 59.5 | 272 | 69 | 0.01 | 7.818422e+01 | 67 | 2787 | 17.6 | 93 | 67 | 8.52 | 67 | 0.1 | 669.959000 | 3696958 | 17.9 | 18.0 | 0.463 |
| 5 | Afghanistan | 2011 | Developing | 59.2 | 275 | 71 | 0.01 | 7.097109e+00 | 68 | 3013 | 17.2 | 97 | 68 | 7.87 | 68 | 0.1 | 63.537231 | 2978599 | 18.2 | 18.2 | 0.454 |
| 6 | Afghanistan | 2010 | Developing | 58.8 | 279 | 74 | 0.01 | 7.967937e+01 | 66 | 1989 | 16.7 | 102 | 66 | 9.20 | 66 | 0.1 | 553.328940 | 2883167 | 18.4 | 18.4 | 0.448 |
| 7 | Afghanistan | 2009 | Developing | 58.6 | 281 | 77 | 0.01 | 5.676222e+01 | 63 | 2861 | 16.2 | 106 | 63 | 9.42 | 63 | 0.1 | 445.893298 | 284331 | 18.6 | 18.7 | 0.434 |
| 8 | Afghanistan | 2008 | Developing | 58.1 | 287 | 80 | 0.03 | 2.587393e+01 | 64 | 1599 | 15.7 | 110 | 64 | 8.33 | 64 | 0.1 | 373.361116 | 2729431 | 18.8 | 18.9 | 0.433 |
| 9 | Afghanistan | 2007 | Developing | 57.5 | 295 | 82 | 0.02 | 1.091016e+01 | 63 | 1141 | 15.2 | 113 | 63 | 6.73 | 63 | 0.1 | 369.835796 | 26616792 | 19.0 | 19.1 | 0.415 |
| 10 | Afghanistan | 2006 | Developing | 57.3 | 295 | 84 | 0.03 | 1.717152e+01 | 64 | 1990 | 14.7 | 116 | 58 | 7.43 | 58 | 0.1 | 272.563770 | 2589345 | 19.2 | 19.3 | 0.405 |
| 11 | Afghanistan | 2005 | Developing | 57.3 | 291 | 85 | 0.02 | 1.388648e+00 | 66 | 1296 | 14.2 | 118 | 58 | 8.70 | 58 | 0.1 | 25.294130 | 257798 | 19.3 | 19.5 | 0.396 |
| 12 | Afghanistan | 2004 | Developing | 57.0 | 293 | 87 | 0.02 | 1.529607e+01 | 67 | 466 | 13.8 | 120 | 5 | 8.79 | 5 | 0.1 | 219.141353 | 24118979 | 19.5 | 19.7 | 0.381 |
| 13 | Afghanistan | 2003 | Developing | 56.7 | 295 | 87 | 0.01 | 1.108905e+01 | 65 | 798 | 13.4 | 122 | 41 | 8.82 | 41 | 0.1 | 198.728544 | 2364851 | 19.7 | 19.9 | 0.373 |
| 14 | Afghanistan | 2002 | Developing | 56.2 | 3 | 88 | 0.01 | 1.688735e+01 | 64 | 2486 | 13.0 | 122 | 36 | 7.76 | 36 | 0.1 | 187.845950 | 21979923 | 19.9 | 2.2 | 0.341 |
| 15 | Afghanistan | 2001 | Developing | 55.3 | 316 | 88 | 0.01 | 1.057473e+01 | 63 | 8762 | 12.6 | 122 | 35 | 7.80 | 33 | 0.1 | 117.496980 | 2966463 | 2.1 | 2.4 | 0.340 |
| 16 | Afghanistan | 2000 | Developing | 54.8 | 321 | 88 | 0.01 | 1.042496e+01 | 62 | 6532 | 12.2 | 122 | 24 | 8.20 | 24 | 0.1 | 114.560000 | 293756 | 2.3 | 2.5 | 0.338 |
| 17 | Albania | 2015 | Developing | 77.8 | 74 | 0 | 4.60 | 3.649752e+02 | 99 | 0 | 58.0 | 0 | 99 | 6.00 | 99 | 0.1 | 3954.227830 | 28873 | 1.2 | 1.3 | 0.762 |
| 18 | Albania | 2014 | Developing | 77.5 | 8 | 0 | 4.51 | 4.287491e+02 | 98 | 0 | 57.2 | 1 | 98 | 5.88 | 98 | 0.1 | 4575.763787 | 288914 | 1.2 | 1.3 | 0.761 |
| 19 | Albania | 2013 | Developing | 77.2 | 84 | 0 | 4.76 | 4.308770e+02 | 99 | 0 | 56.5 | 1 | 99 | 5.66 | 99 | 0.1 | 4414.723140 | 289592 | 1.3 | 1.4 | 0.759 |

Figure 6: Code for showing 'dataset' data frame whole entries

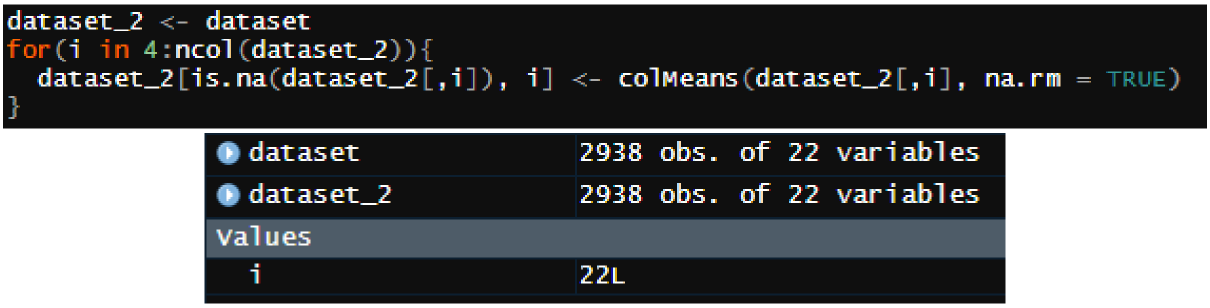
I also observed that there are about a half of dataset that have null value after I omitted the row with the null value. Therefore, I replace the null value with the mean in each column.



```
> dataset_2 <- na.omit(dataset)
```

```
dataset      2938 obs. of 22 variables
dataset_2    1649 obs. of 22 variables
```

Figure 7: Code for omitting the row with null value and create new data frame and the result



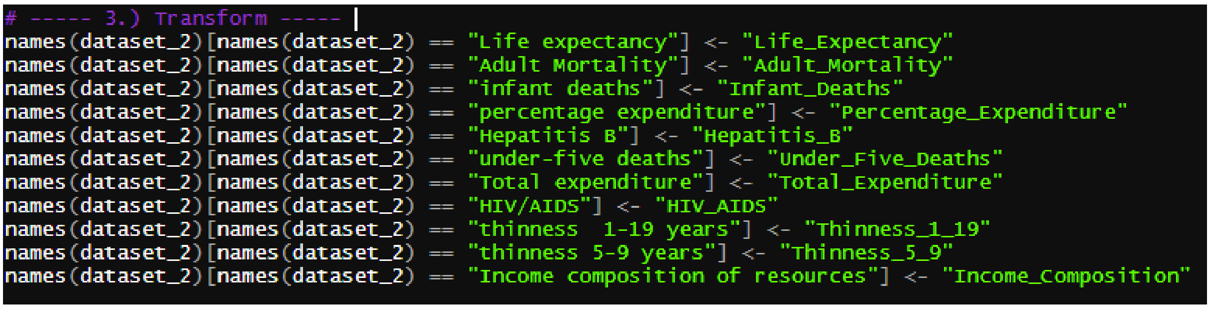
```
dataset_2 <- dataset
for(i in 4:ncol(dataset_2)){
  dataset_2[is.na(dataset_2[,i]), i] <- colMeans(dataset_2[,i], na.rm = TRUE)
}
```

```
dataset      2938 obs. of 22 variables
dataset_2    2938 obs. of 22 variables
Values
  i          22L
```

Figure 8: Replace null value with the mean in each column and its result

### 2.3 Transform

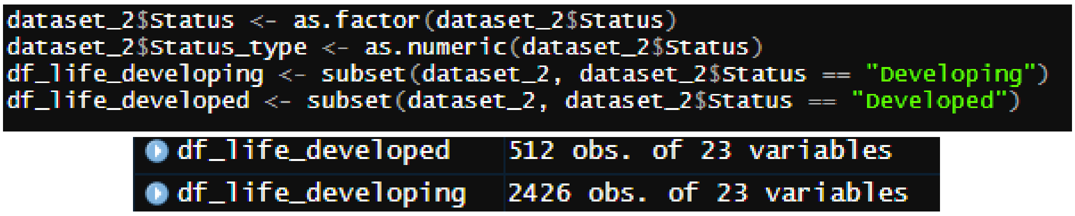
From the current data frame structure, I change the name of some columns that has the space in the name to prevent the error from using any library.



```
# ----- 3.) Transform -----
names(dataset_2)[names(dataset_2) == "Life expectancy"] <- "Life_Expectancy"
names(dataset_2)[names(dataset_2) == "Adult Mortality"] <- "Adult_Mortality"
names(dataset_2)[names(dataset_2) == "infant deaths"] <- "Infant_Deaths"
names(dataset_2)[names(dataset_2) == "percentage expenditure"] <- "Percentage_Expenditure"
names(dataset_2)[names(dataset_2) == "Hepatitis B"] <- "Hepatitis_B"
names(dataset_2)[names(dataset_2) == "under-five deaths"] <- "Under_Five_Deaths"
names(dataset_2)[names(dataset_2) == "Total expenditure"] <- "Total_Expenditure"
names(dataset_2)[names(dataset_2) == "HIV/AIDS"] <- "HIV_AIDS"
names(dataset_2)[names(dataset_2) == "thinness  1-19 years"] <- "Thinness_1_19"
names(dataset_2)[names(dataset_2) == "thinness 5-9 years"] <- "Thinness_5_9"
names(dataset_2)[names(dataset_2) == "Income composition of resources"] <- "Income_Composition"
```

Figure 9: code for rename the columns

The 'Status' column is converted as factor, the new column name 'Status_type', which represents the 'Status' as numeric, and two new data frames are created as the data for 'Developing' countries and 'Developed' countries because we are going to see the differences of the factors for both types.



```
dataset_2$Status <- as.factor(dataset_2$Status)
dataset_2$Status_type <- as.numeric(dataset_2$Status)
df_life_developing <- subset(dataset_2, dataset_2$Status == "Developing")
df_life_developed <- subset(dataset_2, dataset_2$Status == "Developed")
```

```
df_life_developed     512 obs. of 23 variables
df_life_developing    2426 obs. of 23 variables
```

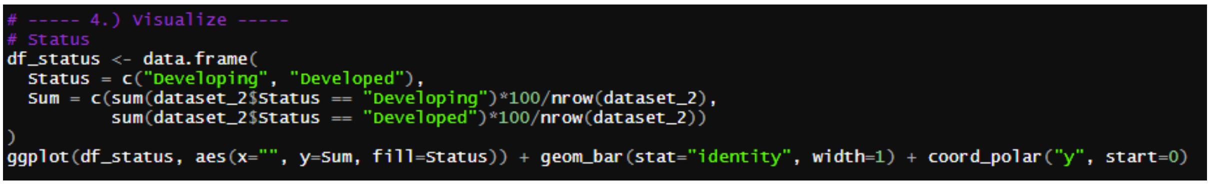
Figure 10: Code for data transformation and the result

## 2.4 Visualize

From my observation, there are many aspects and relationships between the data as represents below.

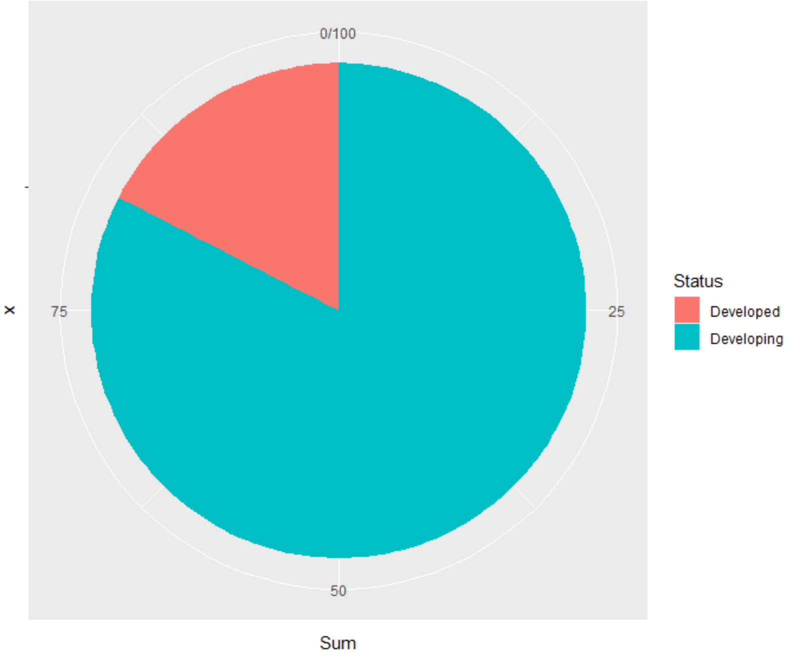
### 2.4.1 Percentage of country status

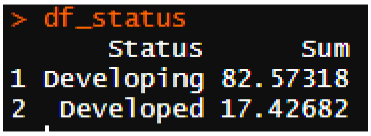
There are 82.57% as developing countries and 17.42% as developed countries.



```
# ----- 4.) Visualize -----
# Status
df_status <- data.frame(
  Status = c("Developing", "Developed"),
  Sum = c(sum(dataset_2$Status == "Developing")*100/nrow(dataset_2),
          sum(dataset_2$Status == "Developed")*100/nrow(dataset_2))
)
ggplot(df_status, aes(x="", y=Sum, fill=Status)) + geom_bar(stat="identity", width=1) + coord_polar("y", start=0)
```

Figure 11: Code for percentage of country status calculation





```
> df_status
      Status      Sum
1 Developing 82.57318
2  Developed 17.42682
```

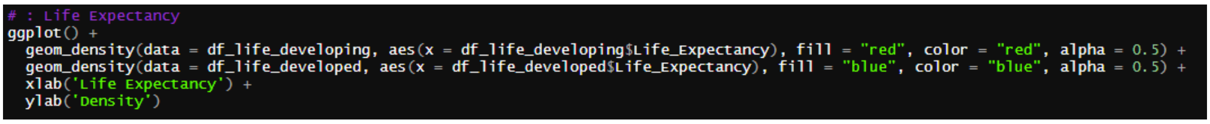
Figure 12: Visualization for percentage of country status calculation

### 2.4.2 Comparison between developing and developed countries

There are the differences distribution between the developing and developed countries as follows.

### 2.4.2.1 Life expectancy

People in developed countries rather have a higher life expectancy than developing country



```
# : Life Expectancy
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Life_Expectancy), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Life_Expectancy), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Life Expectancy') +
  ylab('Density')
```

Figure 13: Code for plotting life expectancy between developing and developed countries

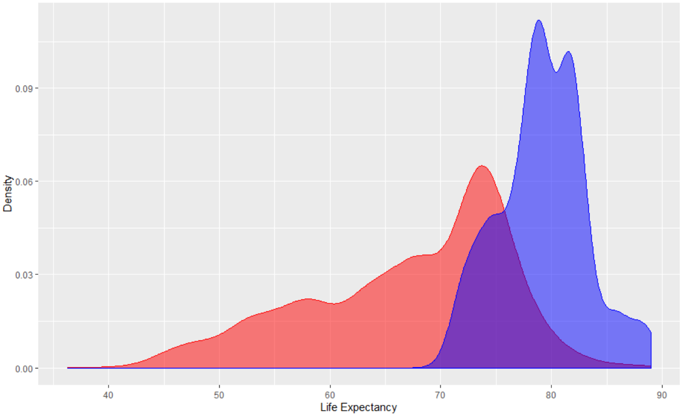
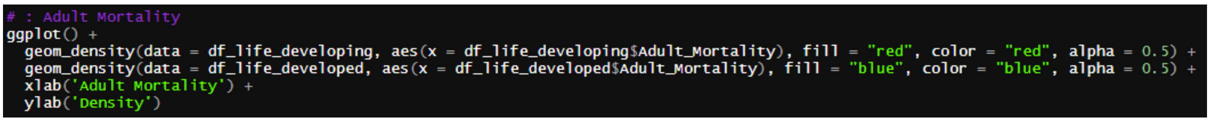


Figure 14: Visualization for plotting life expectancy between developing and developed countries

### 2.4.2.2 Adult mortality

People in developed countries rather have a lower adult mortality than developing country



```
# : Adult Mortality
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Adult_Mortality), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Adult_Mortality), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Adult Mortality') +
  ylab('Density')
```

Figure 15: Code for plotting adult mortality between developing and developed countries

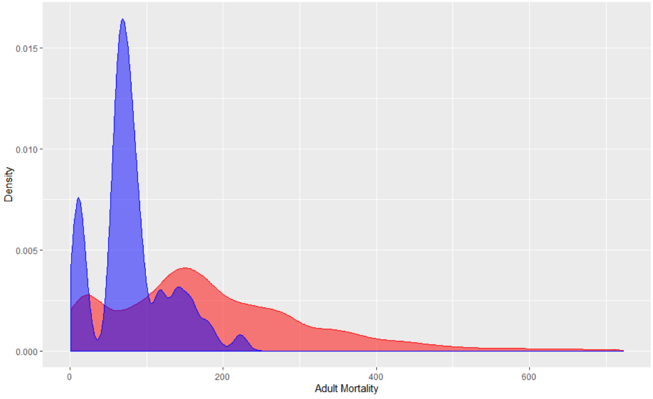
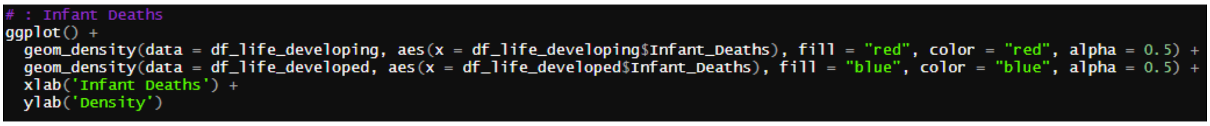


Figure 16: Visualization for plotting adult mortality between developing and developed countries

### 2.4.2.3 Infant deaths

People in developed countries rather have a lower infant death than developing country. There are some countries that have the high infant deaths which becomes the outlier as shown below.



```
# : Infant Deaths
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Infant_Deaths), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Infant_Deaths), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Infant Deaths') +
  ylab('Density')
```

Figure 17: Code for plotting infant deaths between developing and developed countries

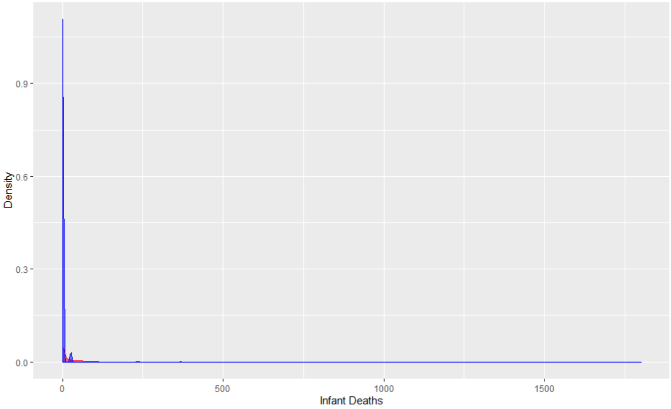
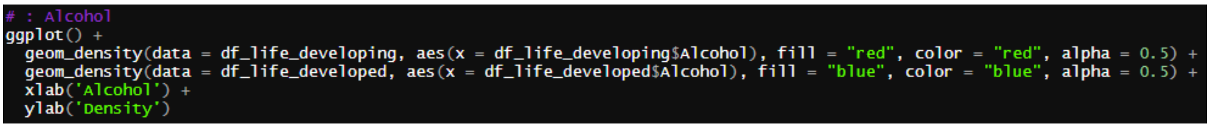


Figure 18: Visualization for plotting infant deaths between developing and developed countries

### 2.4.2.4 Alcohol

People in developed countries rather have a higher alcohol consumption per capita (liter) than developing country.



```
# : Alcohol
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Alcohol), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Alcohol), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Alcohol') +
  ylab('Density')
```

Figure 19: Code for plotting alcohol consumption (liter) between developing and developed countries

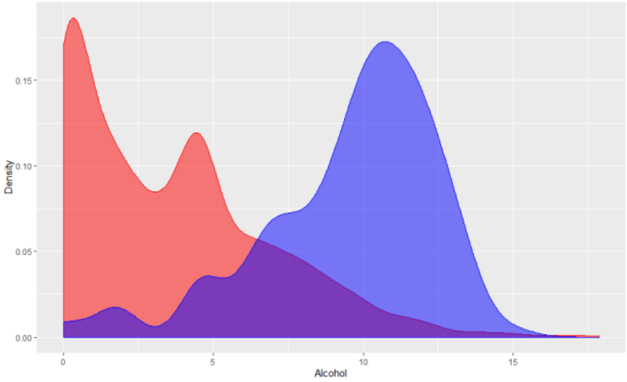
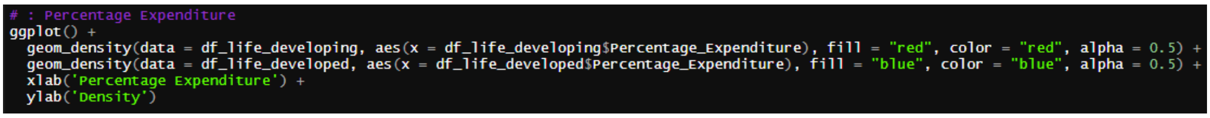


Figure 20: Visualization for plotting alcohol consumption (liter) between developing and developed countries

`

### 2.4.2.5 Percentage expenditure

People in developed countries rather have a higher percentage expenditure than developing country



```
# : Percentage Expenditure
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Percentage_Expenditure), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Percentage_Expenditure), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Percentage Expenditure') +
  ylab('Density')
```

Figure 21: Code for plotting percentage expenditure between developing and developed countries

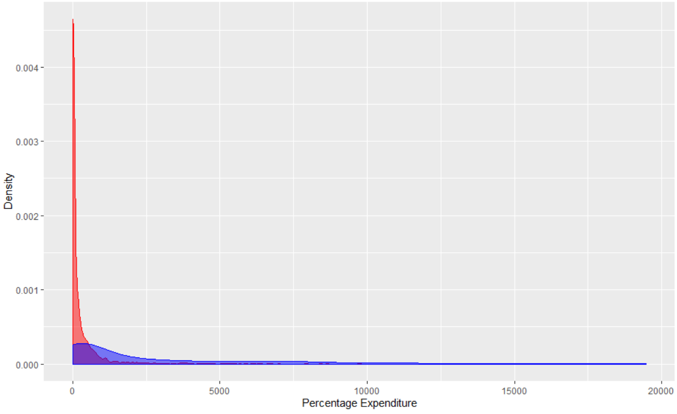
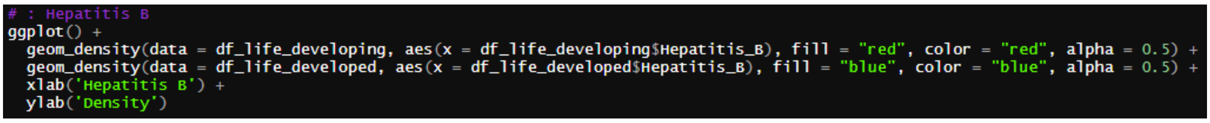


Figure 22: Visualization for plotting percentage expenditure between developing and developed countries

### 2.4.2.6 Hepatitis B

People in developed countries and developing countries quite have the same distribution of hepatitis B, developed countries tend to have a higher rate.



```
# : Hepatitis B
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Hepatitis_B), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Hepatitis_B), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Hepatitis B') +
  ylab('Density')
```

Figure 23: Code for plotting hepatitis B between developing and developed countries

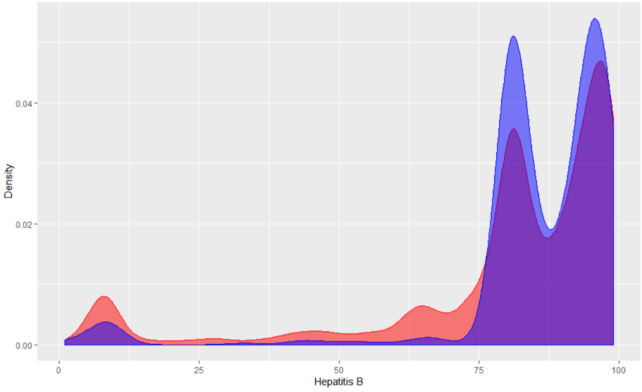
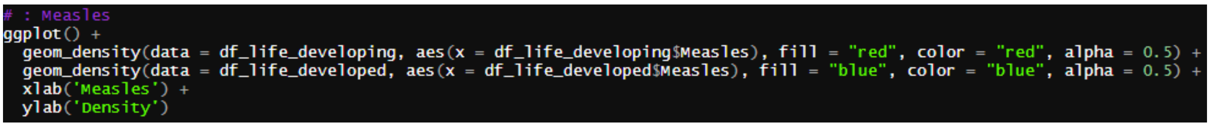


Figure 24: Visualization for plotting hepatitis B between developing and developed countries

### 2.4.2.7 Measles

People in developed countries and developing country have the same distribution of measles. However, developed countries have higher chance to has less measles due to the higher density compare to developing countries.



```
# : Measles
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Measles), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Measles), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Measles') +
  ylab('Density')
```

Figure 25: Code for plotting Measles between developing and developed countries

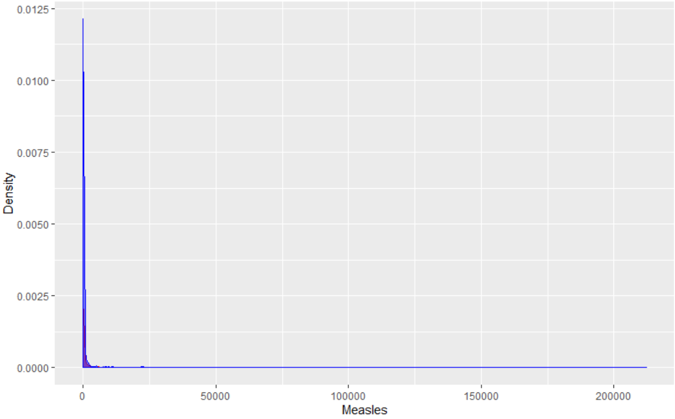
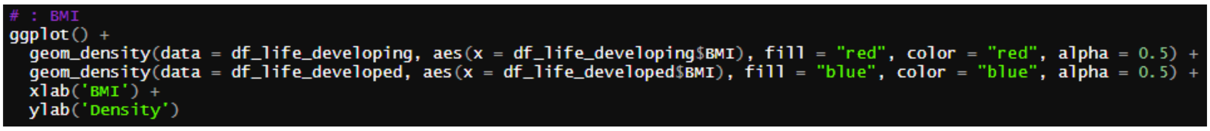


Figure 26: Visualization for plotting measles between developing and developed countries

### 2.4.2.8 BMI

Developed countries have gap distribution of BMI. Most people have higher BMI than appropriate value (18.5 – 22.9)



```
# : BMI
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$BMI), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$BMI), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('BMI') +
  ylab('Density')
```

Figure 27: Code for plotting BMI between developing and developed countries

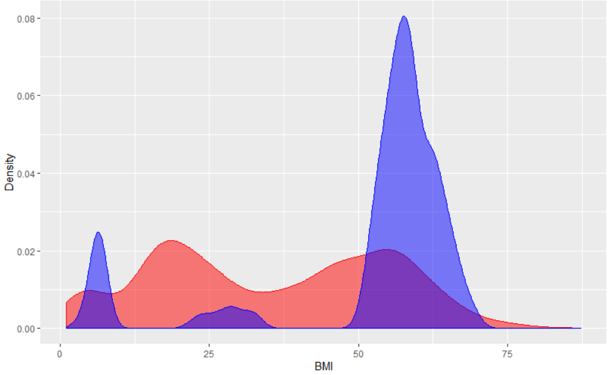
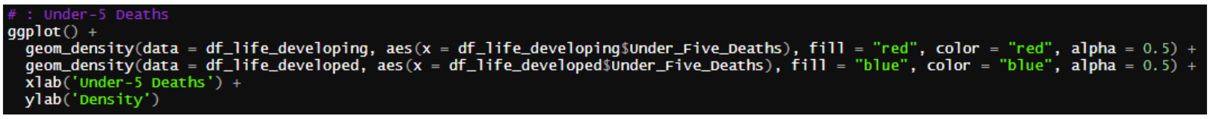


Figure 28: Visualization for plotting BMI between developing and developed countries

2.4.2.9 Under-5 deaths

Developed countries have less under 5 deaths compares to developing countries.



```
# : Under-5 Deaths
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Under_Five_Deaths), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Under_Five_Deaths), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Under-5 Deaths') +
  ylab('Density')
```

Figure 29: Code for plotting under-5 deaths between developing and developed countries

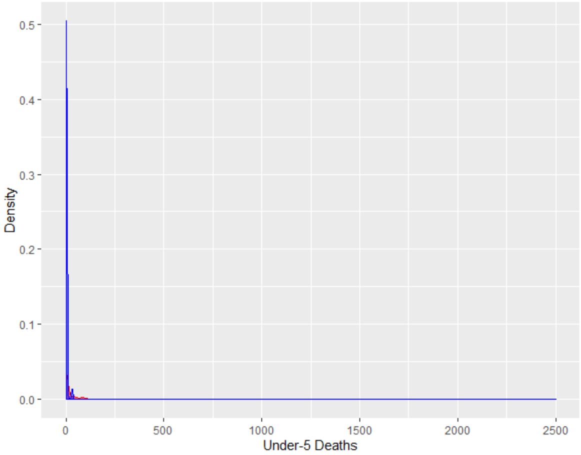
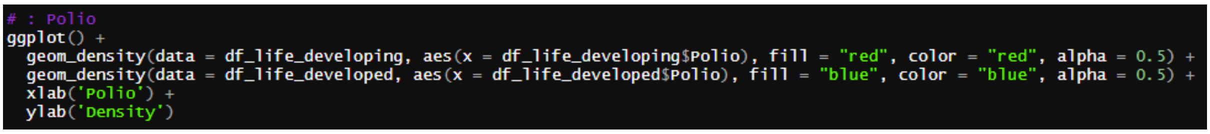


Figure 30: Visualization for plotting under-5 deaths between developing and developed countries

2.4.2.10 Polio

Developed countries have higher polio rates compares to developing countries.



```
# : Polio
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Polio), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Polio), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Polio') +
  ylab('Density')
```

Figure 31: Code for plotting polio between developing and developed countries

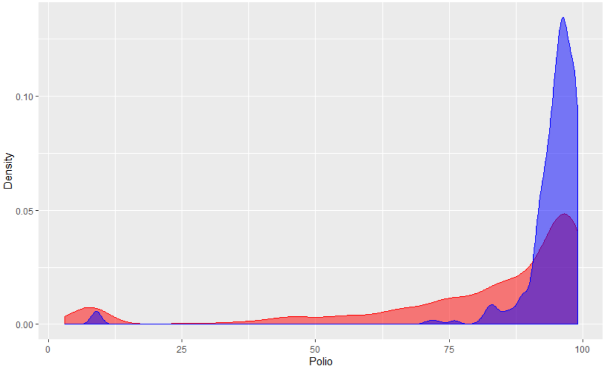
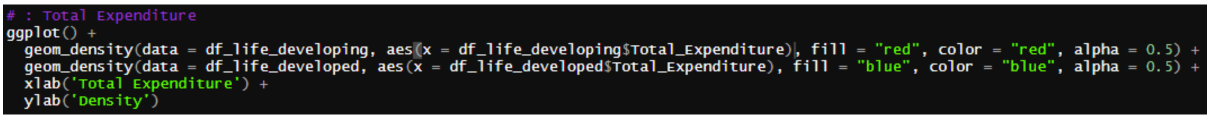


Figure 32: Visualization for plotting polio between developing and developed countries

2.4.2.11 Total expenditure

Government in developed countries expense on health more than developing countries.



```
# : Total Expenditure
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Total_Expenditure), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Total_Expenditure), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Total Expenditure') +
  ylab('Density')
```

Figure 33: Code for plotting total expenditure between developing and developed countries

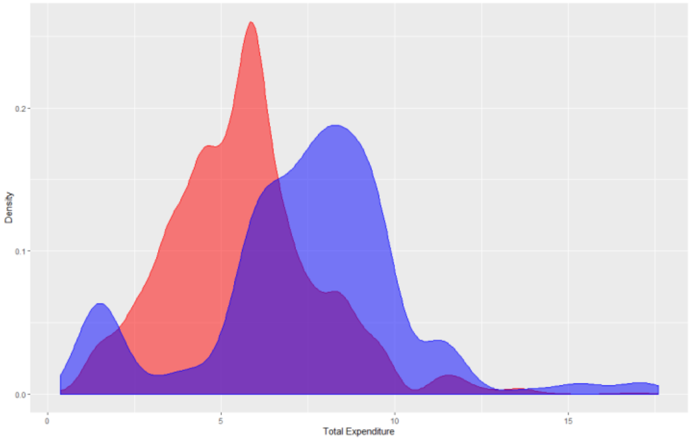
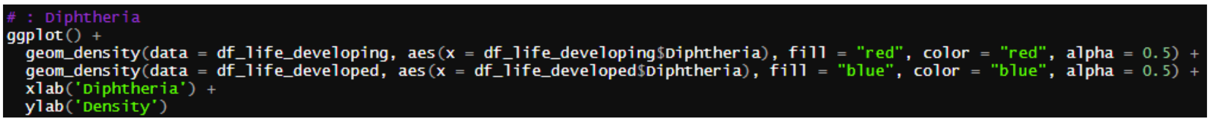


Figure 34: Visualization for plotting total expenditure between developing and developed countries

2.4.2.12 Diphtheria

Developed countries have higher diphtheria rate than developing countries.



```
# : Diphtheria
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Diphtheria), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Diphtheria), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Diphtheria') +
  ylab('Density')
```

Figure 35: Code for plotting diphtheria between developing and developed countries

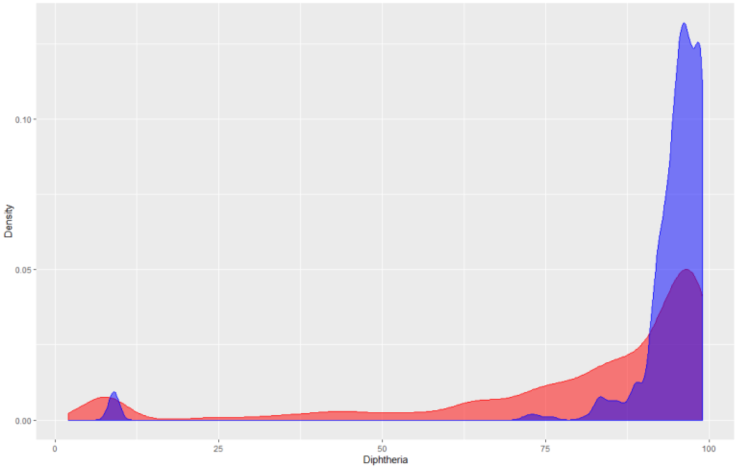
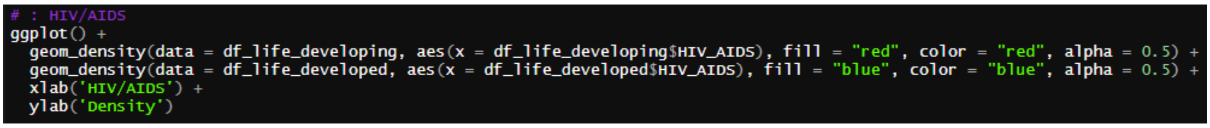


Figure 36: Visualization for plotting polio between developing and developed countries

2.4.2.13 HIV/AIDS

Developed countries have lower death per 1,000 live births HIV/AIDS than developing countries.



```
# : HIV/AIDS
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$HIV_AIDS), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$HIV_AIDS), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('HIV/AIDS') +
  ylab('Density')
```

Figure 37: Code for plotting diphtheria between developing and developed countries

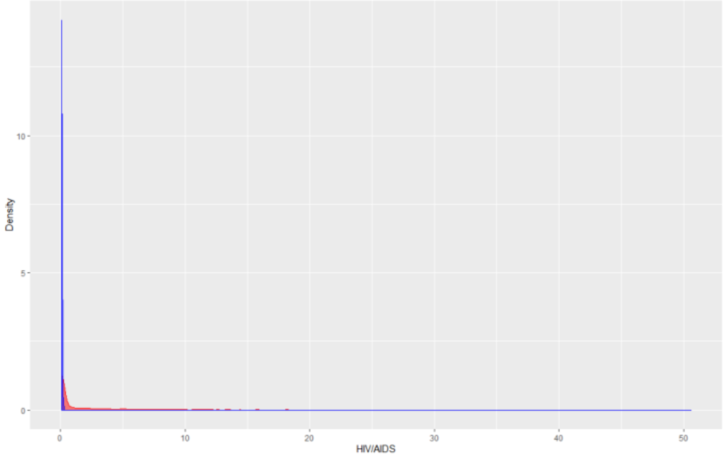
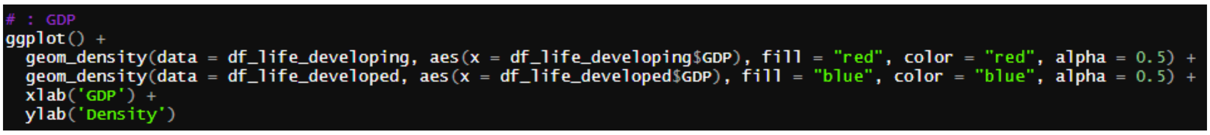


Figure 38: Visualization for plotting HIV/AIDS between developing and developed countries

2.4.2.14 GDP

Developed countries have higher GDP than developing countries.



```
# : GDP
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$GDP), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$GDP), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('GDP') +
  ylab('Density')
```

Figure 39: Code for plotting GDP between developing and developed countries

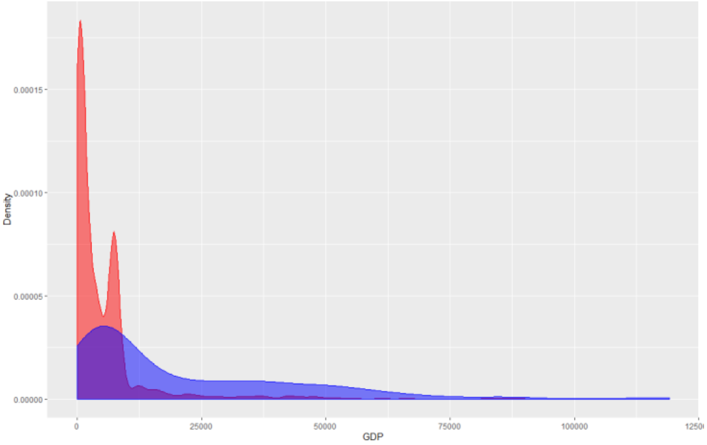
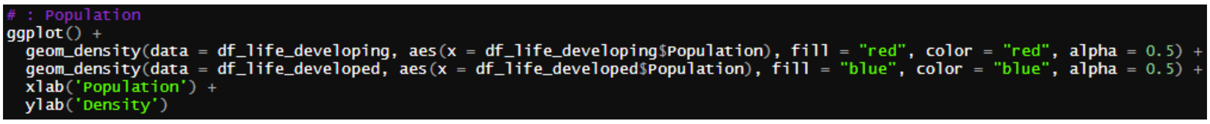


Figure 40: Visualization for plotting GDP between developing and developed countries

2.4.2.15 Population

Developed and developing countries have quite the same distribution of population



```
# : Population
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Population), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Population), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Population') +
  ylab('Density')
```

Figure 41: Code for plotting population between developing and developed countries

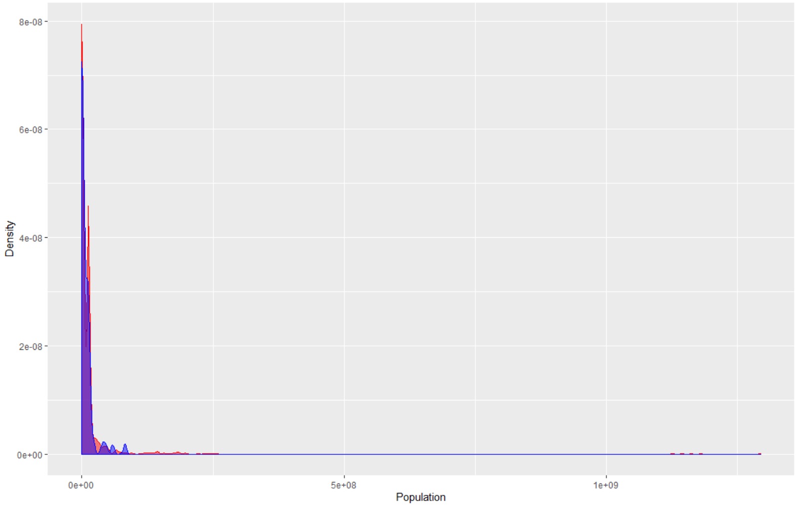
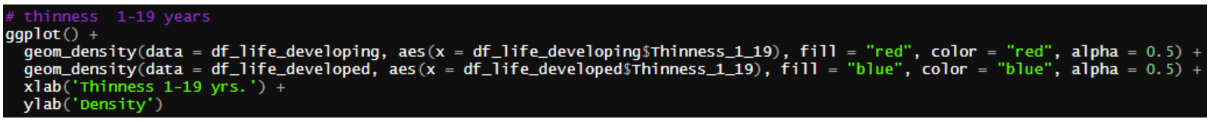


Figure 42: Visualization for plotting population between developing and developed countrie

2.4.2.16 Thinness 10-19 years

Developed countries tend to have less thin adolescents between age 10 – 19 than developing countries.



```
# thinness  1-19 years
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Thinness_1_19), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Thinness_1_19), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Thinness 1-19 yrs.') +
  ylab('Density')
```

Figure 43: Code for plotting thinness 10-19 years between developing and developed countries

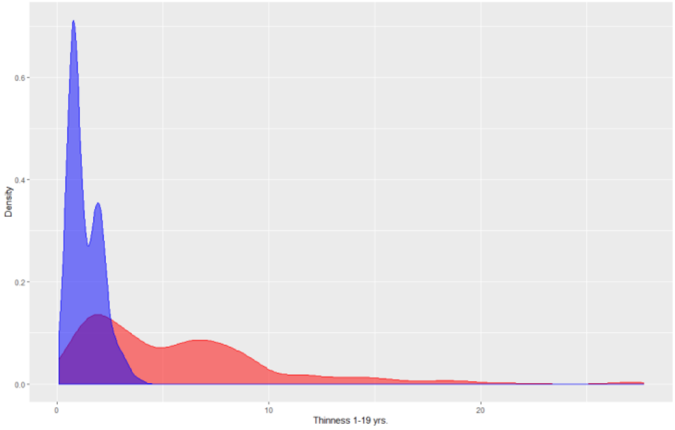
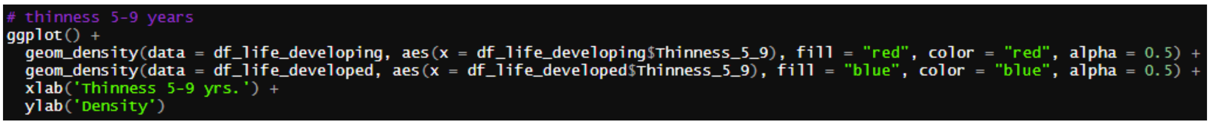


Figure 44: Visualization for plotting Thinness 10-19 years between developing and developed countries

2.4.2.17 Thinness 5-19 years

Developed countries tend to have less thin adolescents between age 5 – 9 than developing countries.



```
# thinness 5-9 years
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Thinness_5_9), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Thinness_5_9), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Thinness 5-9 yrs.') +
  ylab('Density')
```

Figure 45: Code for plotting thinness 5-9 years between developing and developed countries

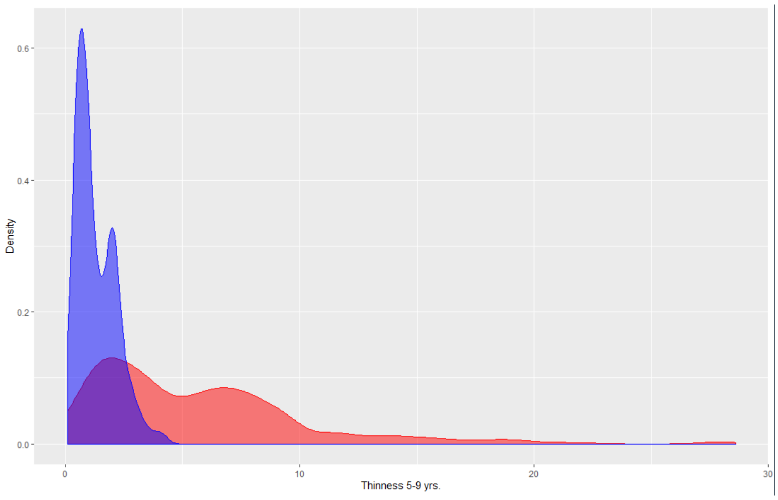
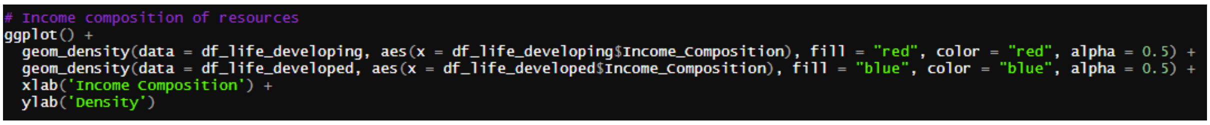


Figure 46: Visualization for plotting thinness 5-9 years between developing and developed countries

2.4.2.18 Income composition of resources

Developed countries tend to have more income composition of resources than developing countries.



```
# Income composition of resources
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Income_Composition), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Income_Composition), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Income Composition') +
  ylab('Density')
```

Figure 47: Plotting income composition of resources between developing and developed countries

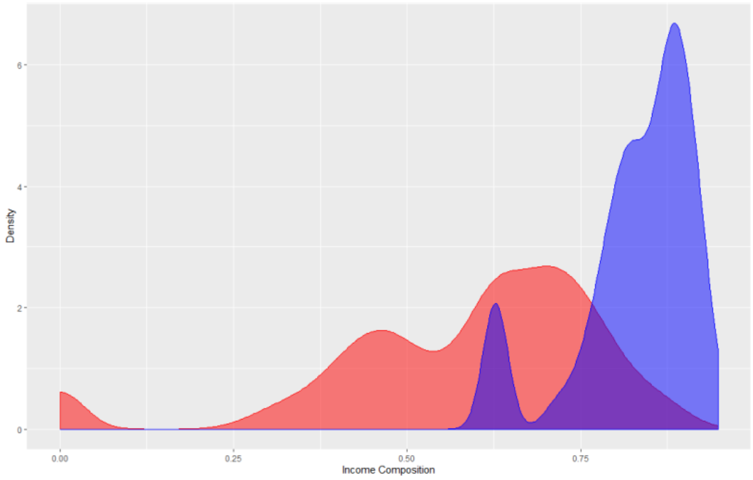
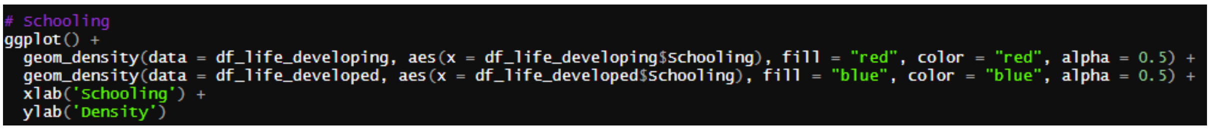


Figure 48: Visualization for plotting income composition of resources between developing and developed countries

### 2.4.2.19 Schooling

Developed countries tend to have more years of schooling than developing countries.



```
# Schooling
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$Schooling), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Schooling), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Schooling') +
  ylab('Density')
```

Figure 49: Code for plotting schooling between developing and developed countries

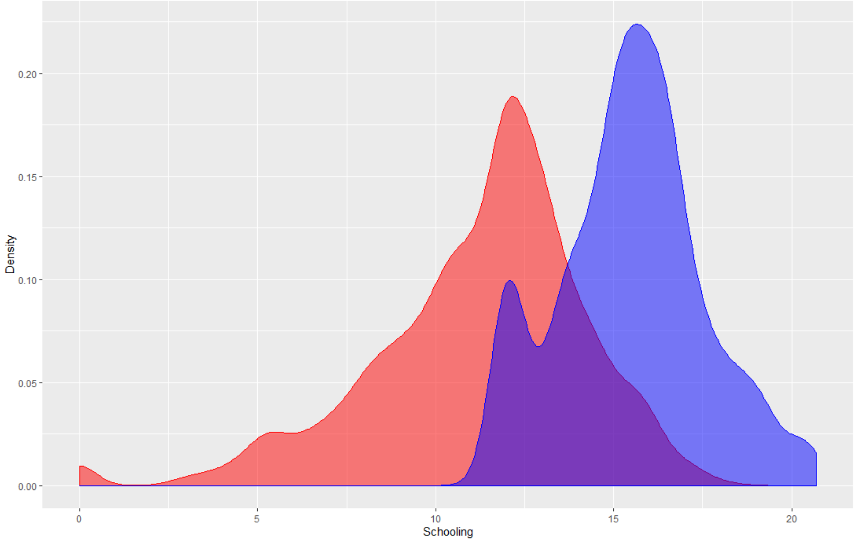
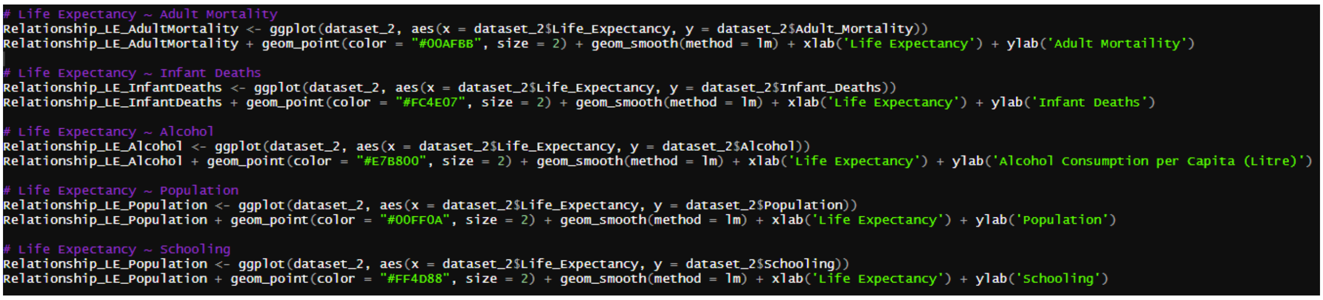


Figure 50: Visualization for plotting schooling between developing and developed countries

### 2.4.3 Effect of some factors to life expectancy

This section shows the relationship between some factors to life expectancy, consists of adult mortality, infant deaths, alcohol, population, and schooling. The code for implementation is shown below.



```
# Life Expectancy ~ Adult Mortality
Relationship_LE_AdultMortality <- ggplot(dataset_2, aes(x = dataset_2$Life_Expectancy, y = dataset_2$Adult_Mortality))
Relationship_LE_AdultMortality + geom_point(color = "#00AFBB", size = 2) + geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Adult Mortaility')

# Life Expectancy ~ Infant Deaths
Relationship_LE_InfantDeaths <- ggplot(dataset_2, aes(x = dataset_2$Life_Expectancy, y = dataset_2$Infant_Deaths))
Relationship_LE_InfantDeaths + geom_point(color = "#FC4E07", size = 2) + geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Infant Deaths')

# Life Expectancy ~ Alcohol
Relationship_LE_Alcohol <- ggplot(dataset_2, aes(x = dataset_2$Life_Expectancy, y = dataset_2$Alcohol))
Relationship_LE_Alcohol + geom_point(color = "#E7B800", size = 2) + geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Alcohol Consumption per Capita (Litre)')

# Life Expectancy ~ Population
Relationship_LE_Population <- ggplot(dataset_2, aes(x = dataset_2$Life_Expectancy, y = dataset_2$Population))
Relationship_LE_Population + geom_point(color = "#00FF0A", size = 2) + geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Population')

# Life Expectancy ~ Schooling
Relationship_LE_Population <- ggplot(dataset_2, aes(x = dataset_2$Life_Expectancy, y = dataset_2$Schooling))
Relationship_LE_Population + geom_point(color = "#FF4D88", size = 2) + geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Schooling')
```

Figure 51: Code for plotting some factors and life expectancy relationship

2.4.3.1 Adult mortality

Adult mortality is inverse proportional to the life expectancy. The higher adult mortality, the less life expectancy that country has.

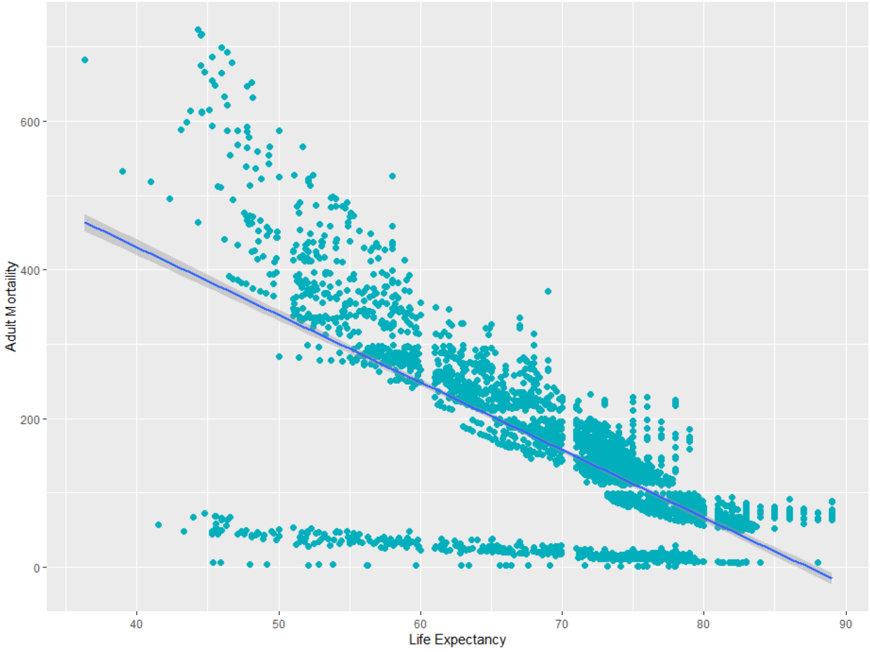


Figure 52: Visualization of relationship between adult mortality and life expectancy

2.4.3.2 Infant deaths

Infant deaths rate is inverse proportional to the life expectancy. The higher Infant deaths rate, the less life expectancy that country has.

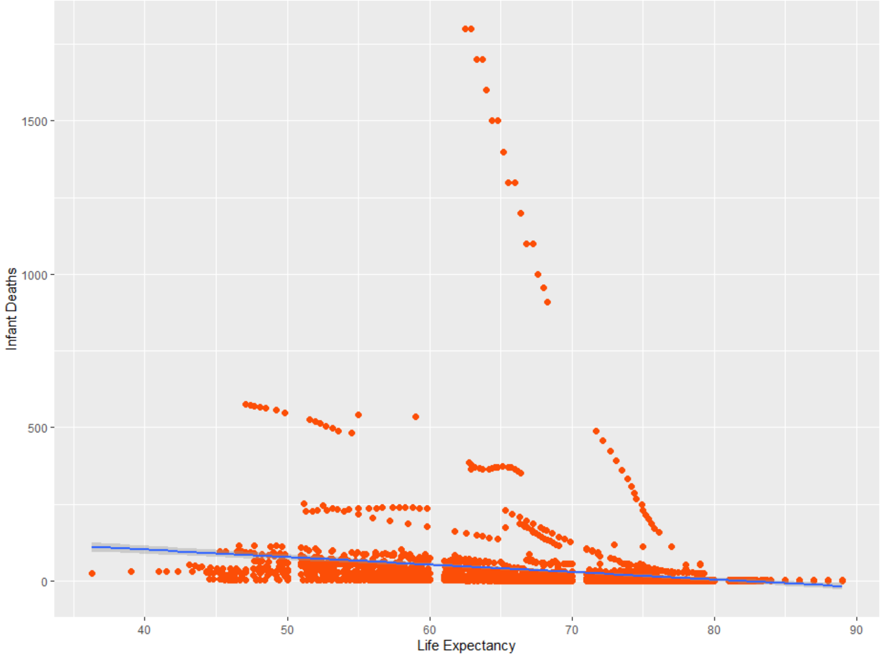


Figure 53: Visualization of relationship between Infant deaths and life expectancy

### 2.4.3.3 Alcohol

Alcohol consumption per capita (liter) is direct proportional to the life expectancy. The higher alcohol consumption per capita (liter), the higher life expectancy that country has.

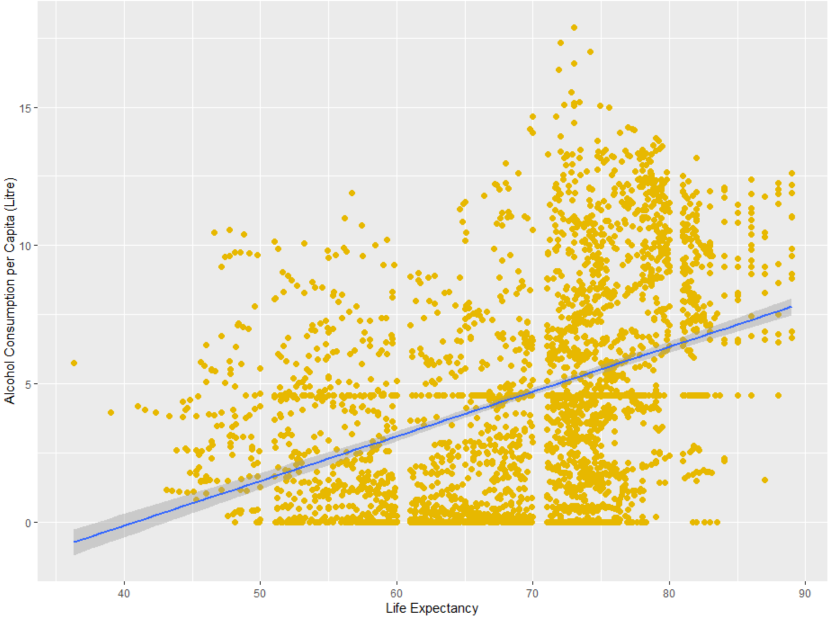


Figure 54: Visualization of relationship between alcohol consumption per capita (liter) and life expectancy

### 2.4.3.4 Population

Amount of population does not have effect to the life expectancy.

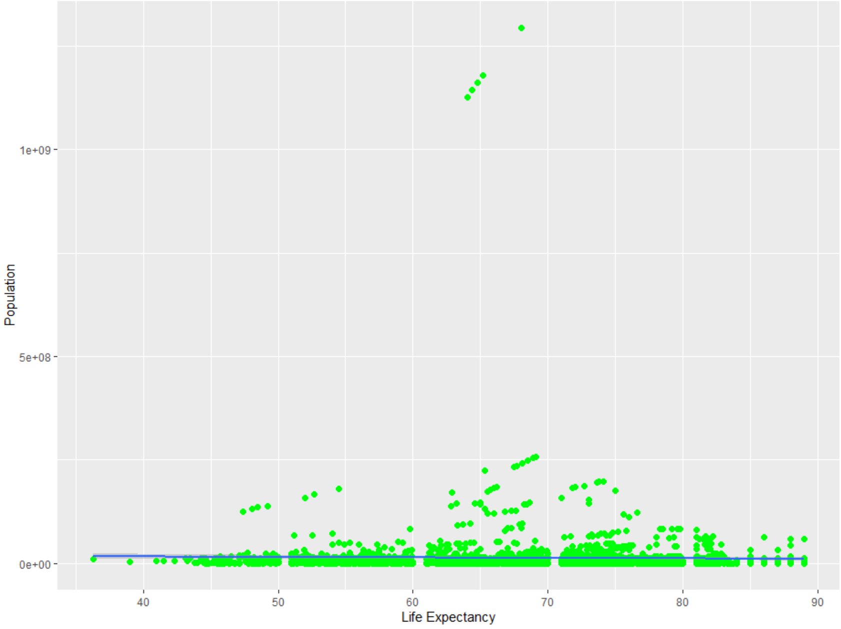


Figure 55: Visualization of relationship between amount of population and life expectancy

### 2.4.3.5 Schooling

Schooling is direct proportional to the life expectancy. The higher years in schooling, the higher life expectancy that country has.

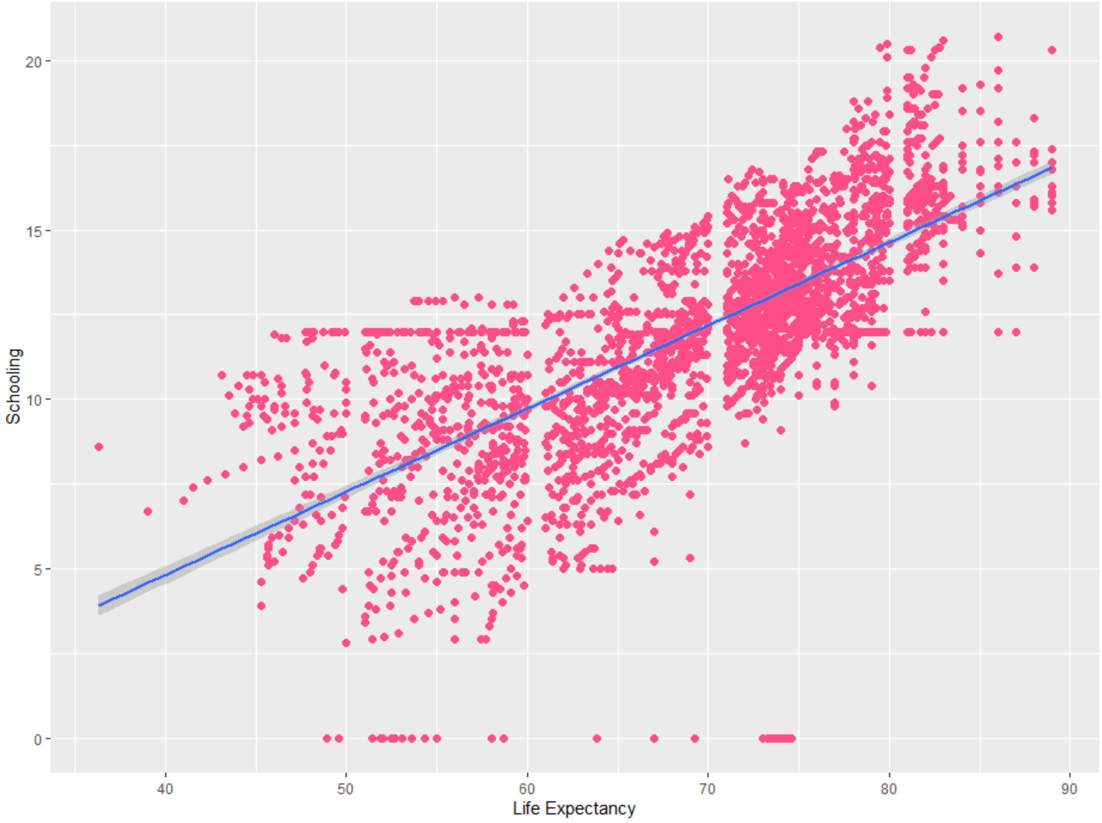
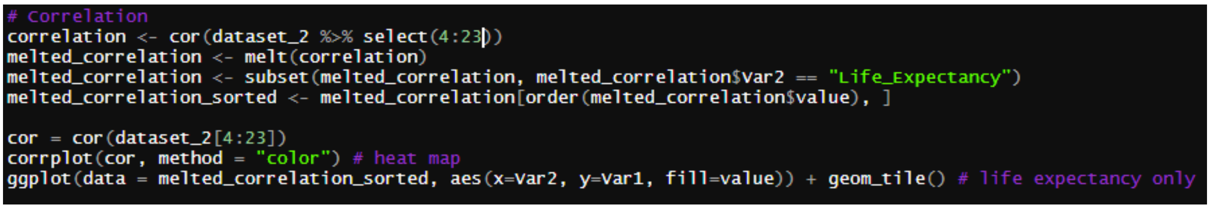


Figure 56: Visualization of relationship between amount of population and life expectancy

### 2.4.3 Correlation

Heat map is created to represent how strong each factor affects together. However, I create new data frame to represent only each factor affects to life expectancy only for easily interpretation.



```
# Correlation
correlation <- cor(dataset_2 %>% select(4:23))
melted_correlation <- melt(correlation)
melted_correlation <- subset(melted_correlation, melted_correlation$Var2 == "Life_Expectancy")
melted_correlation_sorted <- melted_correlation[order(melted_correlation$value), ]

cor = cor(dataset_2[4:23])
corrplot(cor, method = "color") # heat map
ggplot(data = melted_correlation_sorted, aes(x=Var2, y=Var1, fill=value)) + geom_tile() # life expectancy only
```

Figure 57: Code for correlation creating, including values and heat map

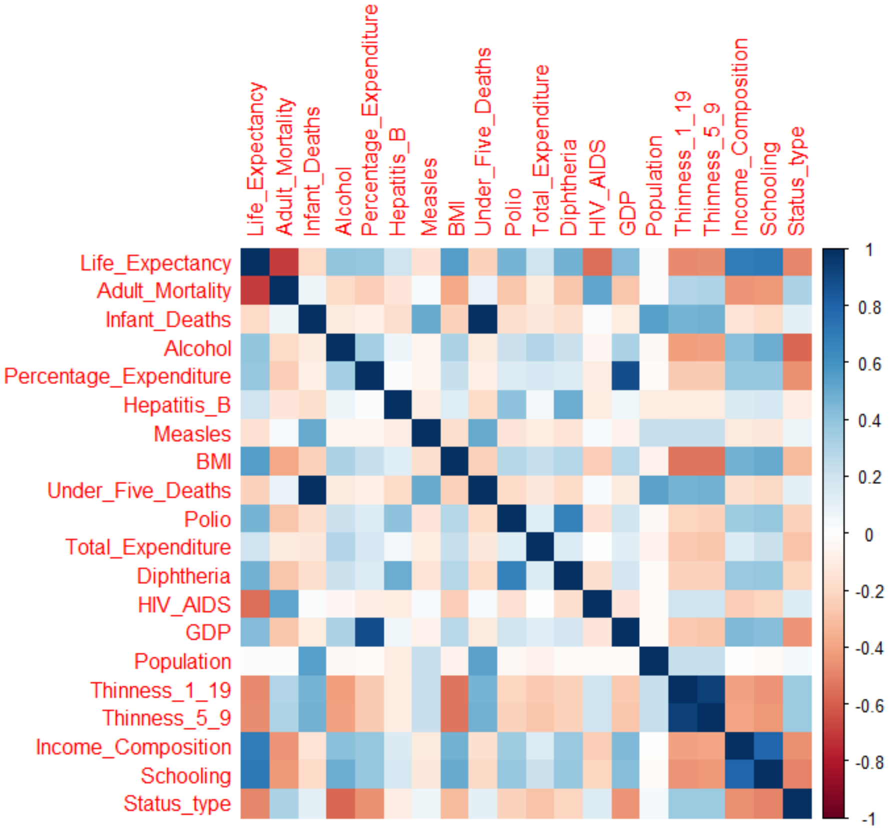
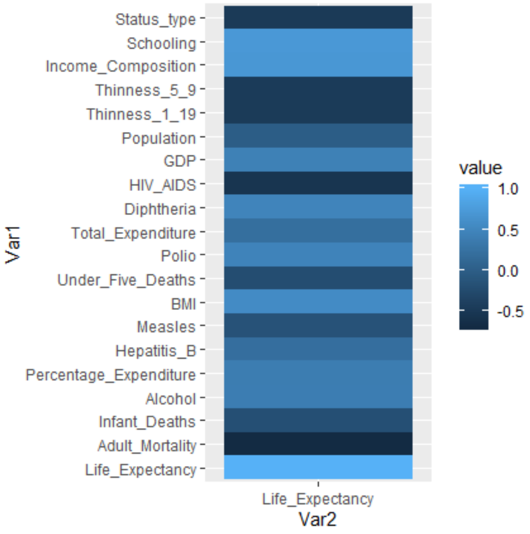
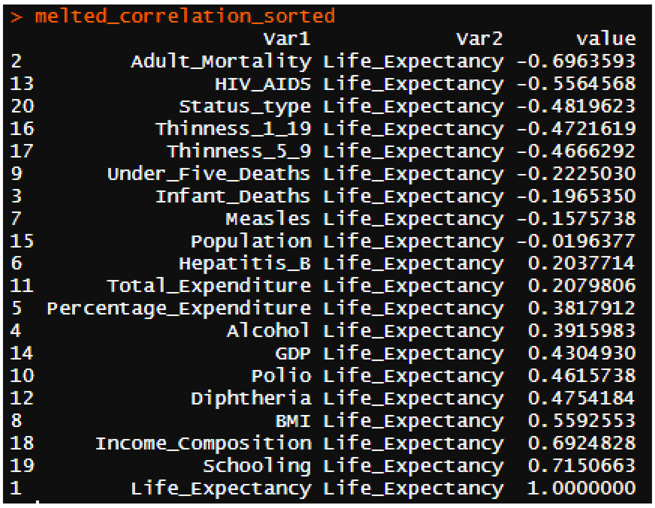


Figure 58: Correlation heat map of each factor

The factor that has the highest directly impact to life expectancy is schooling and the highest inversely impact to life expectancy is adult mortality. The rest has been ordered from lowest to highest as shown below.





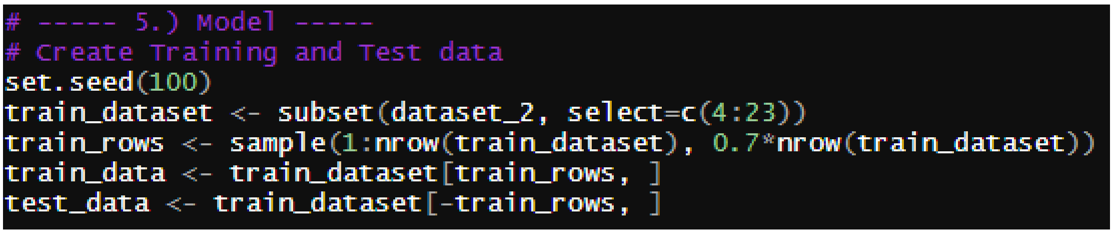
```
> melted_correlation_sorted
                     Var1            Var2       value
2         Adult_Mortality Life_Expectancy -0.6963593
13               HIV_AIDS Life_Expectancy -0.5564568
20            Status_type Life_Expectancy -0.4819623
16          Thinness_1_19 Life_Expectancy -0.4721619
17           Thinness_5_9 Life_Expectancy -0.4666292
9       Under_Five_Deaths Life_Expectancy -0.2225030
3           Infant_Deaths Life_Expectancy -0.1965350
7                 Measles Life_Expectancy -0.1575738
15             Population Life_Expectancy -0.0196377
6             Hepatitis_B Life_Expectancy  0.2037714
11      Total_Expenditure Life_Expectancy  0.2079806
5  Percentage_Expenditure Life_Expectancy  0.3817912
4                 Alcohol Life_Expectancy  0.3915983
14                    GDP Life_Expectancy  0.4304930
10                  Polio Life_Expectancy  0.4615738
12             Diphtheria Life_Expectancy  0.4754184
8                     BMI Life_Expectancy  0.5592553
18     Income_Composition Life_Expectancy  0.6924828
19              Schooling Life_Expectancy  0.7150663
1         Life_Expectancy Life_Expectancy  1.0000000
```

Figure 59: Correlation of each factor to life expectancy as color (left) and value from lowest to highest (right)

2.5 Modelling

2.5.1 Create train and test dataset

The dataset has been selected from fourth column to the last column (4 to 23) and separated to train 70% and test 30% by the code below

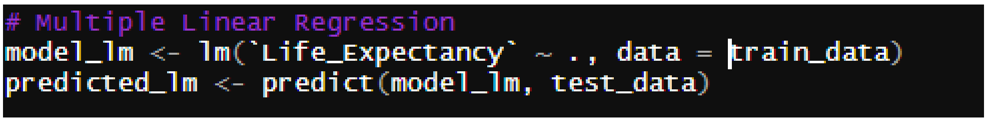


```
# ----- 5.) Model -----
# Create Training and Test data
set.seed(100)
train_dataset <- subset(dataset_2, select=c(4:23))
train_rows <- sample(1:nrow(train_dataset), 0.7*nrow(train_dataset))
train_data <- train_dataset[train_rows, ]
test_data <- train_dataset[-train_rows, ]
```

Figure 60: Split the tidied dataset as train 70% and test 30%

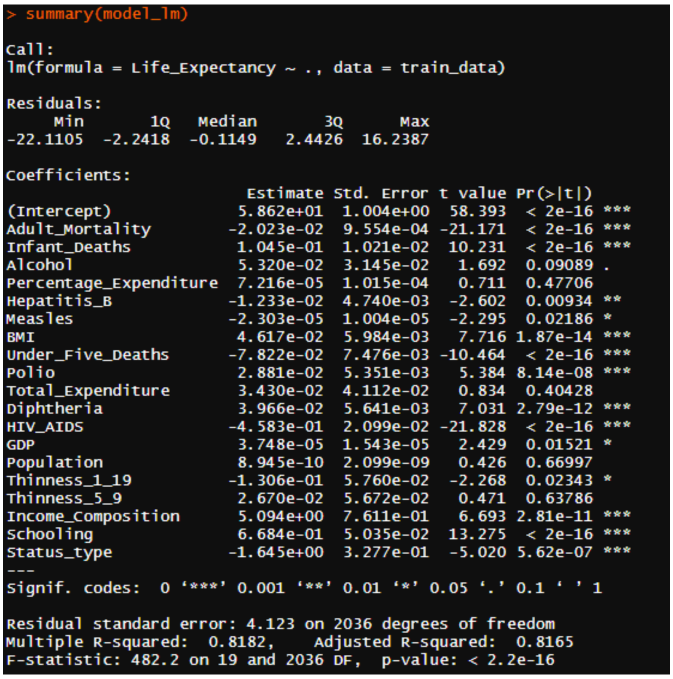
2.5.2 Multiple linear regression

For multiple linear regression, function ‘lm()’ has been used. First, the model selects all attributes as the features.



```
# Multiple Linear Regression
model_lm <- lm(`Life_Expectancy` ~ ., data = train_data)
predicted_lm <- predict(model_lm, test_data)
```

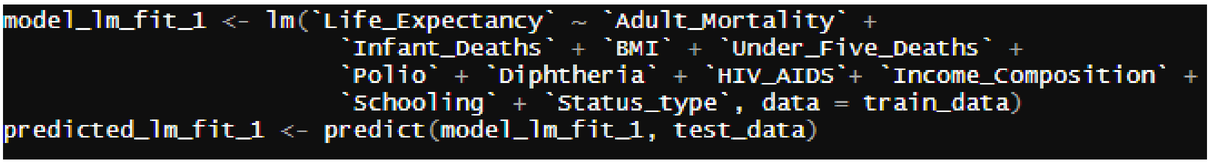
Figure 61: Multiple linear regression model



```
> summary(model_lm)

Call:
lm(formula = Life_Expectancy ~ ., data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-22.1105  -2.2418  -0.1149   2.4426  16.2387 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            5.862e+01  1.004e+00  58.393  < 2e-16 ***
Adult_Mortality       -2.023e-02  9.554e-04 -21.171  < 2e-16 ***
Infant_Deaths          1.045e-01  1.021e-02  10.231  < 2e-16 ***
Alcohol                5.320e-02  3.145e-02   1.692  0.09089 .  
Percentage_Expenditure 7.216e-05  1.015e-04   0.711  0.47706    
Hepatitis_B           -1.233e-02  4.740e-03  -2.602  0.00934 ** 
Measles               -2.303e-05  1.004e-05  -2.295  0.02186 *  
BMI                    4.617e-02  5.984e-03   7.716 1.87e-14 ***
Under_Five_Deaths     -7.822e-02  7.476e-03 -10.464  < 2e-16 ***
Polio                  2.881e-02  5.351e-03   5.384 8.14e-08 ***
Total_Expenditure      3.430e-02  4.112e-02   0.834  0.40428    
Diphtheria             3.966e-02  5.641e-03   7.031 2.79e-12 ***
HIV_AIDS              -4.583e-01  2.099e-02 -21.828  < 2e-16 ***
GDP                    3.748e-05  1.543e-05   2.429  0.01521 *  
Population             8.945e-10  2.099e-09   0.426  0.66997    
Thinness_1_19         -1.306e-01  5.760e-02  -2.268  0.02343 *  
Thinness_5_9           2.670e-02  5.672e-02   0.471  0.63786    
Income_Composition     5.094e+00  7.611e-01   6.693 2.81e-11 ***
Schooling              6.684e-01  5.035e-02  13.275  < 2e-16 ***
Status_type           -1.645e+00  3.277e-01  -5.020 5.62e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.123 on 2036 degrees of freedom
Multiple R-squared:  0.8182,	Adjusted R-squared:  0.8165 
F-statistic: 482.2 on 19 and 2036 DF,  p-value: < 2.2e-16
```

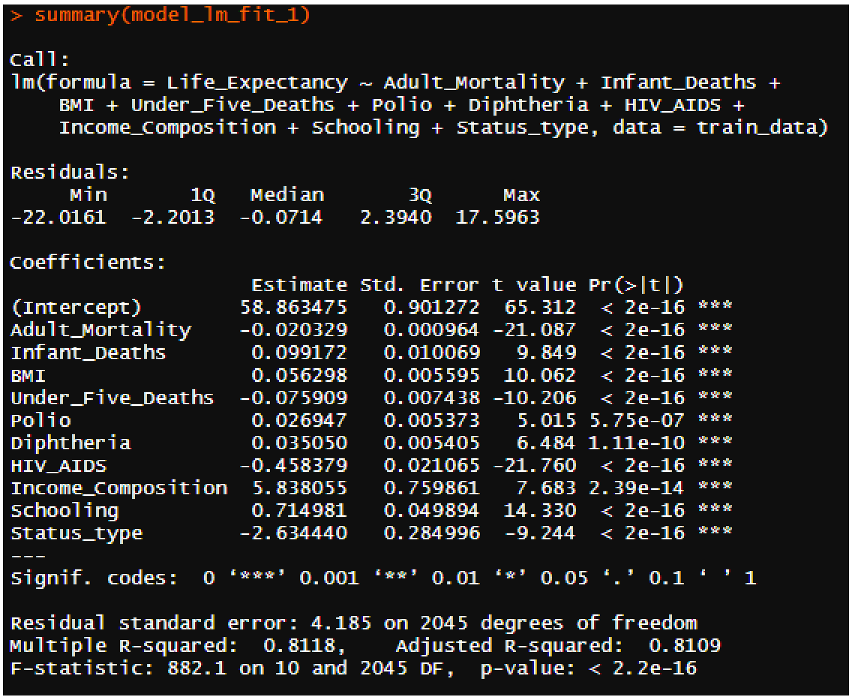
Figure 62: Summary of multiple linear regression model

In addition, I also trained another multiple linear regression model by selected only some features which have significance code as '***' from figure 6.2



```
model_lm_fit_1 <- lm(`Life_Expectancy` ~ `Adult_Mortality` +
                       `Infant_Deaths` + `BMI` + `Under_Five_Deaths` +
                       `Polio` + `Diphtheria` + `HIV_AIDS`+ `Income_Composition` +
                       `Schooling` + `Status_type`, data = train_data)
predicted_lm_fit_1 <- predict(model_lm_fit_1, test_data)
```

Figure 63: Multiple linear regression model with only high significance features

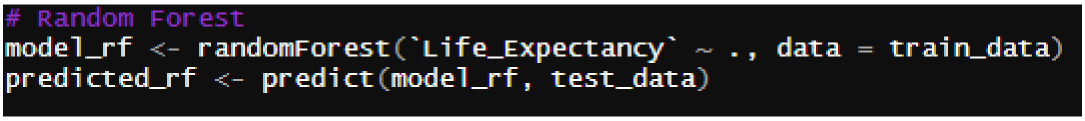


```
> summary(model_lm_fit_1)

Call:
lm(formula = Life_Expectancy ~ Adult_Mortality + Infant_Deaths +
    BMI + Under_Five_Deaths + Polio + Diphtheria + HIV_AIDS +
    Income_Composition + Schooling + Status_type, data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-22.0161  -2.2013  -0.0714   2.3940  17.5963

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        58.863475   0.901272  65.312  < 2e-16 ***
Adult_Mortality    -0.020329   0.000964 -21.087  < 2e-16 ***
Infant_Deaths       0.099172   0.010069   9.849  < 2e-16 ***
BMI                 0.056298   0.005595  10.062  < 2e-16 ***
Under_Five_Deaths  -0.075909   0.007438 -10.206  < 2e-16 ***
Polio               0.026947   0.005373   5.015 5.75e-07 ***
Diphtheria          0.035050   0.005405   6.484 1.11e-10 ***
HIV_AIDS           -0.458379   0.021065 -21.760  < 2e-16 ***
Income_Composition  5.838055   0.759861   7.683 2.39e-14 ***
Schooling           0.714981   0.049894  14.330  < 2e-16 ***
Status_type        -2.634440   0.284996  -9.244  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.185 on 2045 degrees of freedom
Multiple R-squared:  0.8118,	Adjusted R-squared:  0.8109
F-statistic: 882.1 on 10 and 2045 DF,  p-value: < 2.2e-16
```

Figure 64: Summary of multiple linear regression model with only high significance features

2.5.3 Random forest

Random forest is another algorithm that was applied to create the model. Function 'randomForest()' is used.

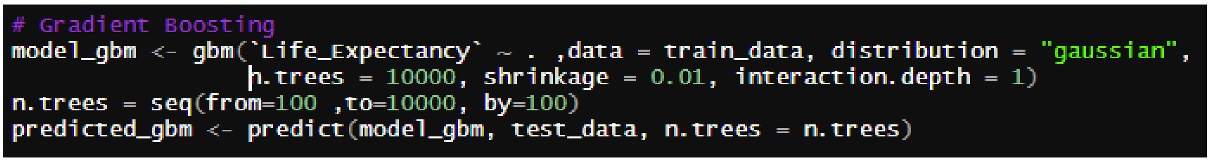


```
# Random Forest
model_rf <- randomForest(`Life_Expectancy` ~ ., data = train_data)
predicted_rf <- predict(model_rf, test_data)
```

Figure 65: Random forest model

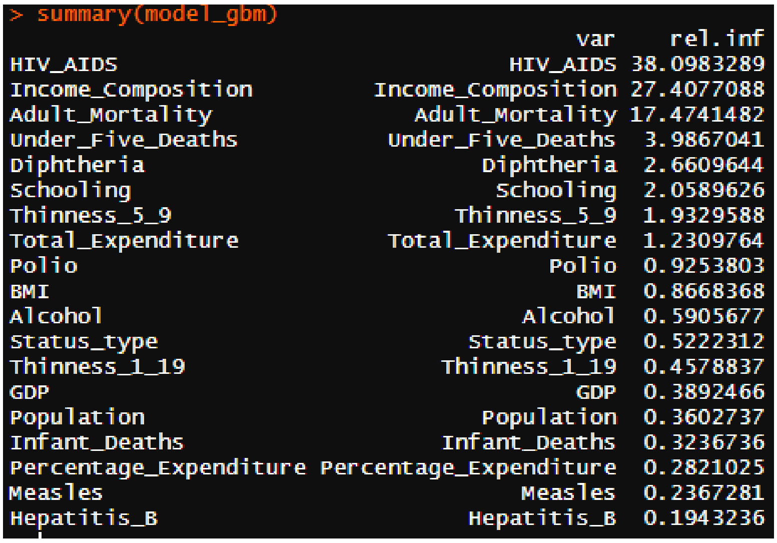
### 2.5.4 Gradient boosting

Gradient boosting is another technique that is used in this project. Some parameters need to be set – distribution type as Gaussian, number of trees as 10,000, learning rate (shrinkage) = 0.01.



```
# Gradient Boosting
model_gbm <- gbm(`Life_Expectancy` ~ . ,data = train_data, distribution = "gaussian",
                 n.trees = 10000, shrinkage = 0.01, interaction.depth = 1)
n.trees = seq(from=100 ,to=10000, by=100)
predicted_gbm <- predict(model_gbm, test_data, n.trees = n.trees)
```

Figure 66: Gradient boosting model



```
> summary(model_gbm)
                                           var    rel.inf
HIV_AIDS                              HIV_AIDS 38.0983289
Income_Composition          Income_Composition 27.4077088
Adult_Mortality                Adult_Mortality 17.4741482
Under_Five_Deaths            Under_Five_Deaths  3.9867041
Diphtheria                          Diphtheria  2.6609644
Schooling                            Schooling  2.0589626
Thinness_5_9                      Thinness_5_9  1.9329588
Total_Expenditure            Total_Expenditure  1.2309764
Polio                                    Polio  0.9253803
BMI                                        BMI  0.8668368
Alcohol                                Alcohol  0.5905677
Status_type                        Status_type  0.5222312
Thinness_1_19                    Thinness_1_19  0.4578837
GDP                                        GDP  0.3892466
Population                          Population  0.3602737
Infant_Deaths                    Infant_Deaths  0.3236736
Percentage_Expenditure Percentage_Expenditure  0.2821025
Measles                                Measles  0.2367281
Hepatitis_B                        Hepatitis_B  0.1943236
```

Figure 67: Summary of gradient booster model

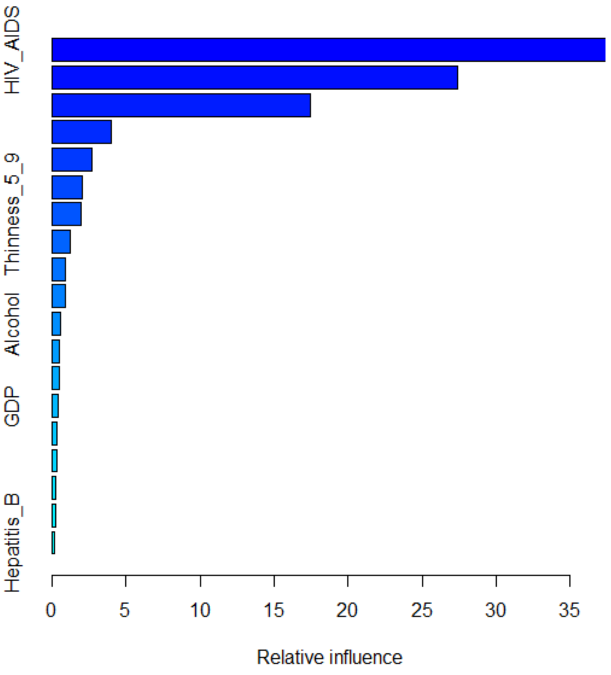
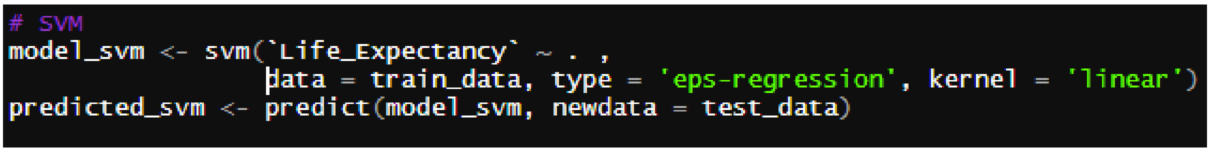


Figure 68: Visualization of gradient booster model summary

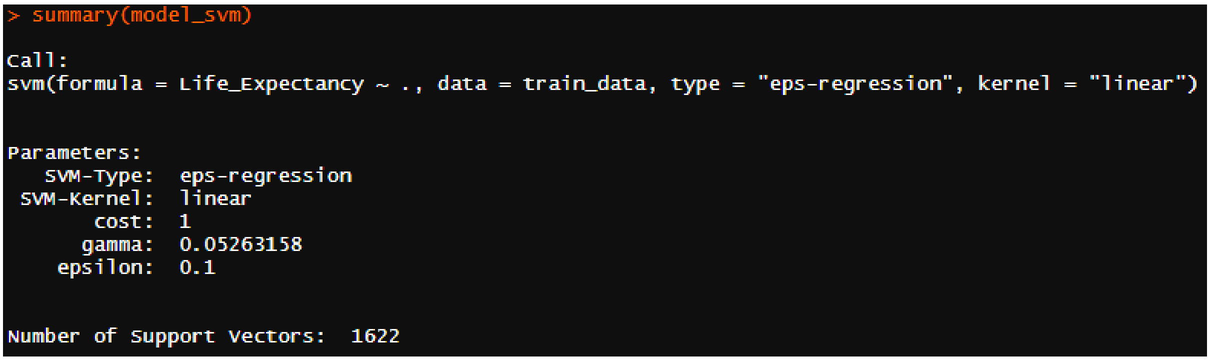
### 2.5.5 Supported vector machine (SVM)

Supported vector machine is mainly used in classification but it can also generate regression model by set the parameter ‘type’ as ‘eps-regression’ and ‘kernel’ as ‘linear’



```
# SVM
model_svm <- svm(`Life_Expectancy` ~ . ,
                 data = train_data, type = 'eps-regression', kernel = 'linear')
predicted_svm <- predict(model_svm, newdata = test_data)
```

Figure 69: Supported vector machine model



```
> summary(model_svm)

Call:
svm(formula = Life_Expectancy ~ ., data = train_data, type = "eps-regression", kernel = "linear")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.05263158
    epsilon:  0.1


Number of Support Vectors:  1622
```

Figure 70: Summary of supported vector machine model

### 2.6 Accuracy

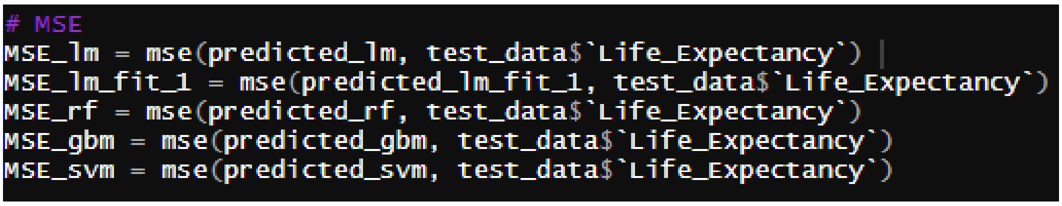
To evaluate each model efficient. Predictions from each model need to be compared with the test dataset. The criteria that is used consists of mean-squared error, root-mean-squared error, and min/max accuracy.

| Tools | Criteria |
|---|---|
| Mean-squared error (MSE) | Less is good |
| Root-mean-squared error (RMSE) | Less is good |
| Min/max accuracy | More is good |

Table 2: Accuracy tools and criteria

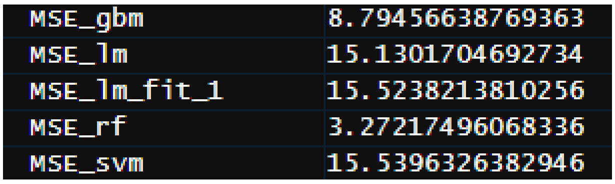
2.6.1 Mean-squared error (MSE)

Five models accuracy were calculated by ‘mse()’ function.



```
# MSE
MSE_lm = mse(predicted_lm, test_data$`Life_Expectancy`)
MSE_lm_fit_1 = mse(predicted_lm_fit_1, test_data$`Life_Expectancy`)
MSE_rf = mse(predicted_rf, test_data$`Life_Expectancy`)
MSE_gbm = mse(predicted_gbm, test_data$`Life_Expectancy`)
MSE_svm = mse(predicted_svm, test_data$`Life_Expectancy`)
```

Figure 71: Mean-squared error calculation for five model – multiple linear regression, multiple linear regression with high significance features, random forest, gradient boost, and supported vector machine.

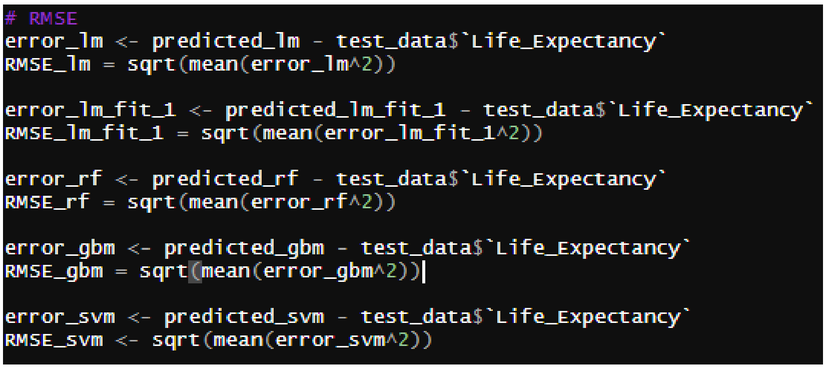


```
MSE_gbm         8.79456638769363
MSE_lm          15.1301704692734
MSE_lm_fit_1    15.5238213810256
MSE_rf          3.27217496068336
MSE_svm         15.5396326382946
```

Figure 72: Result of mean-squared error

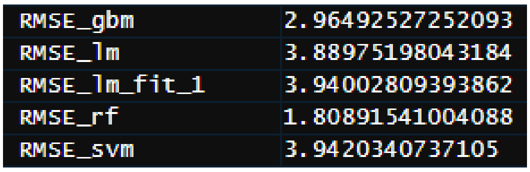
2.6.2 Root-mean-squared error (RMSE)

Five models accuracy were calculated by root-mean-squared error formula.



```
# RMSE
error_lm <- predicted_lm - test_data$`Life_Expectancy`
RMSE_lm = sqrt(mean(error_lm^2))

error_lm_fit_1 <- predicted_lm_fit_1 - test_data$`Life_Expectancy`
RMSE_lm_fit_1 = sqrt(mean(error_lm_fit_1^2))

error_rf <- predicted_rf - test_data$`Life_Expectancy`
RMSE_rf = sqrt(mean(error_rf^2))

error_gbm <- predicted_gbm - test_data$`Life_Expectancy`
RMSE_gbm = sqrt(mean(error_gbm^2))

error_svm <- predicted_svm - test_data$`Life_Expectancy`
RMSE_svm <- sqrt(mean(error_svm^2))
```

Figure 73: Root-mean-squared error calculation for five model – multiple linear regression, multiple linear regression with high significance features, random forest, gradient boost, and supported vector machine.

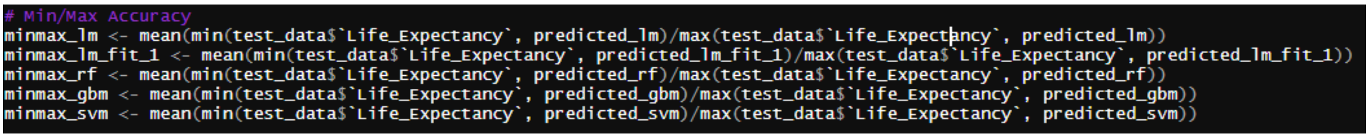


```
RMSE_gbm        2.96492527252093
RMSE_lm         3.88975198043184
RMSE_lm_fit_1   3.94002809393862
RMSE_rf         1.80891541004088
RMSE_svm        3.9420340737105
```

Figure 74: Result of root-mean-squared error

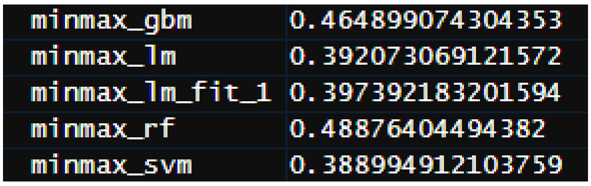
### 2.6.3 Min/max accuracy

Five models accuracy were calculated by min/max accuracy formula.



```
# Min/Max Accuracy
minmax_lm <- mean(min(test_data$`Life_Expectancy`, predicted_lm)/max(test_data$`Life_Expectancy`, predicted_lm))
minmax_lm_fit_1 <- mean(min(test_data$`Life_Expectancy`, predicted_lm_fit_1)/max(test_data$`Life_Expectancy`, predicted_lm_fit_1))
minmax_rf <- mean(min(test_data$`Life_Expectancy`, predicted_rf)/max(test_data$`Life_Expectancy`, predicted_rf))
minmax_gbm <- mean(min(test_data$`Life_Expectancy`, predicted_gbm)/max(test_data$`Life_Expectancy`, predicted_gbm))
minmax_svm <- mean(min(test_data$`Life_Expectancy`, predicted_svm)/max(test_data$`Life_Expectancy`, predicted_svm))
```

Figure 75: Min/max accuracy calculation for five model – multiple linear regression, multiple linear regression with high significance features, random forest, gradient boost, and supported vector machine.

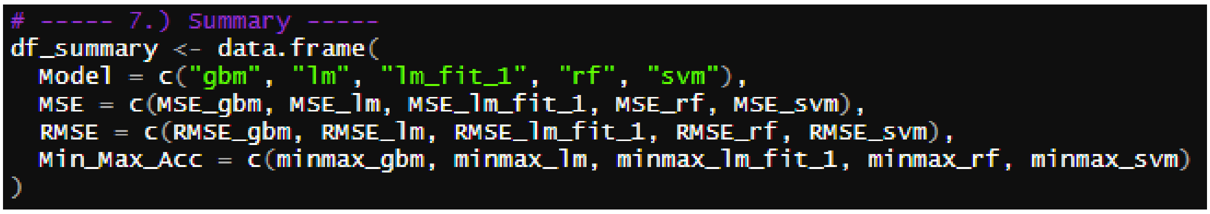


```
minmax_gbm       0.464899074304353
minmax_lm        0.392073069121572
minmax_lm_fit_1  0.397392183201594
minmax_rf        0.48876404494382
minmax_svm       0.388994912103759
```

Figure 76: Result of min/max accuracy

## 3. Summary

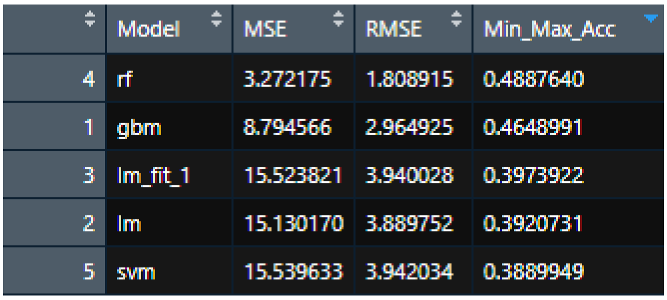
All models accuracy are merge as a new data frame and show the result below.



```
# ----- 7.) Summary -----
df_summary <- data.frame(
  Model = c("gbm", "lm", "lm_fit_1", "rf", "svm"),
  MSE = c(MSE_gbm, MSE_lm, MSE_lm_fit_1, MSE_rf, MSE_svm),
  RMSE = c(RMSE_gbm, RMSE_lm, RMSE_lm_fit_1, RMSE_rf, RMSE_svm),
  Min_Max_Acc = c(minmax_gbm, minmax_lm, minmax_lm_fit_1, minmax_rf, minmax_svm)
)
```

Figure 77: Merge all accuracy as new data frame

From observation, random forest technique gives the best accuracy and least error, followed by gradient boost, multiple linear regression with only high significant features, multiple linear regression, and supported vector machine (SVM) respectively.



|  | Model | MSE | RMSE | Min_Max_Acc |
|---|---|---|---|---|
| 4 | rf | 3.272175 | 1.808915 | 0.4887640 |
| 1 | gbm | 8.794566 | 2.964925 | 0.4648991 |
| 3 | lm_fit_1 | 15.523821 | 3.940028 | 0.3973922 |
| 2 | lm | 15.130170 | 3.889752 | 0.3920731 |
| 5 | svm | 15.539633 | 3.942034 | 0.3889949 |

Figure 78: All models accuracy, sorted by min/max accuracy

In the aspect of life expectancy, the factor that has the highest directly impact to life expectancy is schooling and the highest inversely impact to life expectancy is adult mortality. The rest has been ordered from lowest to highest as shown in figure 59.

## 4. Appendix

```
#####################################################
#
# INC 491 - Data Science and Intelligent Techniques
# Mini Project  - Credit Approval Data Set
#
# Life Expectancy (WHO)
# Statistical Analysis on factors influencing Life
# Expectancy
#
# Developed by: Twitty Manymoon
#
#####################################################

# ----- 1.) Import -----
library(gbm)
library(readr)
library(e1071)
library(Metrics)
library(ggplot2)
library(reshape2)
library(corrplot)
library(tidyverse)
library(randomForest)
dataset <- read_csv("Life Expectancy Data.csv")

# ----- 2.) Tidy up -----
# pattern of data, and find modelling strategies
str(dataset)
summary(dataset)
View(dataset)

# NA removeing
#dataset_2 <- na.omit(dataset)
dataset_2 <- dataset
for(i in 4:ncol(dataset_2)){
  dataset_2[is.na(dataset_2[,i]), i] <- colMeans(dataset_2[,i], na.rm =
TRUE)
}

# ----- 3.) Transform -----
names(dataset_2)[names(dataset_2) == "Life expectancy"] <-
"Life_Expectancy"
names(dataset_2)[names(dataset_2) == "Adult Mortality"] <-
"Adult_Mortality"
names(dataset_2)[names(dataset_2) == "infant deaths"] <- "Infant_Deaths"
names(dataset_2)[names(dataset_2) == "percentage expenditure"] <-
"Percentage_Expenditure"
names(dataset_2)[names(dataset_2) == "Hepatitis B"] <- "Hepatitis_B"
names(dataset_2)[names(dataset_2) == "under-five deaths"] <-
"Under_Five_Deaths"
```

```
names(dataset_2)[names(dataset_2) == "Total expenditure"] <-
"Total_Expenditure"
names(dataset_2)[names(dataset_2) == "HIV/AIDS"] <- "HIV_AIDS"
names(dataset_2)[names(dataset_2) == "thinness  1-19 years"] <-
"Thinness_1_19"
names(dataset_2)[names(dataset_2) == "thinness 5-9 years"] <-
"Thinness_5_9"
names(dataset_2)[names(dataset_2) == "Income composition of resources"] <-
"Income_Composition"

dataset_2$Status <- as.factor(dataset_2$Status)
dataset_2$Status_type <- as.numeric(dataset_2$Status)
df_life_developing <- subset(dataset_2, dataset_2$Status == "Developing")
df_life_developed <- subset(dataset_2, dataset_2$Status == "Developed")

# ----- 4.) Visualize -----
# Status
df_status <- data.frame(
  Status = c("Developing", "Developed"),
  Sum = c(sum(dataset_2$Status == "Developing")*100/nrow(dataset_2),
          sum(dataset_2$Status == "Developed")*100/nrow(dataset_2))
)
ggplot(df_status, aes(x="", y=Sum, fill=Status)) +
geom_bar(stat="identity", width=1) + coord_polar("y", start=0)

# : Life Expectancy
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Life_Expectancy), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Life_Expectancy), fill = "blue", color = "blue", alpha =
0.5) +
  xlab('Life Expectancy') +
  ylab('Density')

# : Adult Mortality
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Adult_Mortality), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Adult_Mortality), fill = "blue", color = "blue", alpha =
0.5) +
  xlab('Adult Mortality') +
  ylab('Density')

# : Infant Deaths
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Infant_Deaths), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Infant_Deaths), fill = "blue", color = "blue", alpha =
0.5) +
  xlab('Infant Deaths') +
  ylab('Density')

# : Alcohol
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Alcohol), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Alcohol), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Alcohol') +
  ylab('Density')

# : Percentage Expenditure
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Percentage_Expenditure), fill = "red", color = "red",
alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Percentage_Expenditure), fill = "blue", color = "blue",
alpha = 0.5) +
  xlab('Percentage Expenditure') +
  ylab('Density')

# : Hepatitis B
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Hepatitis_B), fill = "red", color = "red", alpha = 0.5)
+
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Hepatitis_B), fill = "blue", color = "blue", alpha = 0.5)
+
  xlab('Hepatitis B') +
  ylab('Density')

# : Measles
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Measles), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Measles), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Measles') +
  ylab('Density')

# : BMI
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$BMI),
fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$BMI),
fill = "blue", color = "blue", alpha = 0.5) +
  xlab('BMI') +
  ylab('Density')

# : Under-5 Deaths
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Under_Five_Deaths), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Under_Five_Deaths), fill = "blue", color = "blue", alpha
= 0.5) +
  xlab('Under-5 Deaths') +
  ylab('Density')

# : Polio
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Polio), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$Polio),
fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Polio') +
  ylab('Density')

# : Total Expenditure
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Total_Expenditure), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Total_Expenditure), fill = "blue", color = "blue", alpha
= 0.5) +
  xlab('Total Expenditure') +
  ylab('Density')

# : Diphtheria
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Diphtheria), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Diphtheria), fill = "blue", color = "blue", alpha = 0.5)
+
  xlab('Diphtheria') +
  ylab('Density')

# : HIV/AIDS
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$HIV_AIDS), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$HIV_AIDS), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('HIV/AIDS') +
  ylab('Density')

# : GDP
ggplot() +
  geom_density(data = df_life_developing, aes(x = df_life_developing$GDP),
fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x = df_life_developed$GDP),
fill = "blue", color = "blue", alpha = 0.5) +
  xlab('GDP') +
  ylab('Density')

# : Population
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Population), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Population), fill = "blue", color = "blue", alpha = 0.5)
+
  xlab('Population') +
  ylab('Density')

# thinness  10-19 years
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Thinness_1_19), fill = "red", color = "red", alpha =
0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Thinness_1_19), fill = "blue", color = "blue", alpha =
0.5) +
  xlab('Thinness 1-19 yrs.') +
  ylab('Density')

# thinness 5-9 years
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Thinness_5_9), fill = "red", color = "red", alpha = 0.5)
+
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Thinness_5_9), fill = "blue", color = "blue", alpha =
0.5) +
  xlab('Thinness 5-9 yrs.') +
  ylab('Density')

# Income composition of resources
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Income_Composition), fill = "red", color = "red", alpha
= 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Income_Composition), fill = "blue", color = "blue", alpha
= 0.5) +
  xlab('Income Composition') +
  ylab('Density')

# Schooling
ggplot() +
  geom_density(data = df_life_developing, aes(x =
df_life_developing$Schooling), fill = "red", color = "red", alpha = 0.5) +
  geom_density(data = df_life_developed, aes(x =
df_life_developed$Schooling), fill = "blue", color = "blue", alpha = 0.5) +
  xlab('Schooling') +
  ylab('Density')

# Life Expectancy ~ Adult Mortality
Relationship_LE_AdultMortality <- ggplot(dataset_2, aes(x =
dataset_2$Life_Expectancy, y = dataset_2$Adult_Mortality))
Relationship_LE_AdultMortality + geom_point(color = "#00AFBB", size = 2) +
geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Adult
Mortaility')

# Life Expectancy ~ Infant Deaths
Relationship_LE_InfantDeaths <- ggplot(dataset_2, aes(x =
dataset_2$Life_Expectancy, y = dataset_2$Infant_Deaths))
Relationship_LE_InfantDeaths + geom_point(color = "#FC4E07", size = 2) +
geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Infant Deaths')

# Life Expectancy ~ Alcohol
Relationship_LE_Alcohol <- ggplot(dataset_2, aes(x =
dataset_2$Life_Expectancy, y = dataset_2$Alcohol))
Relationship_LE_Alcohol + geom_point(color = "#E7B800", size = 2) +
geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Alcohol
Consumption per Capita (Litre)')

# Life Expectancy ~ Population
Relationship_LE_Population <- ggplot(dataset_2, aes(x =
dataset_2$Life_Expectancy, y = dataset_2$Population))
```

```
Relationship_LE_Population + geom_point(color = "#00FF0A", size = 2) +
geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Population')

# Life Expectancy ~ Schooling
Relationship_LE_Population <- ggplot(dataset_2, aes(x =
dataset_2$Life_Expectancy, y = dataset_2$Schooling))
Relationship_LE_Population + geom_point(color = "#FF4D88", size = 2) +
geom_smooth(method = lm) + xlab('Life Expectancy') + ylab('Schooling')

# Correlation
correlation <- cor(dataset_2 %>% select(4:23))
melted_correlation <- melt(correlation)
melted_correlation <- subset(melted_correlation, melted_correlation$Var2 ==
"Life_Expectancy")
melted_correlation_sorted <-
melted_correlation[order(melted_correlation$value), ]

cor = cor(dataset_2[4:23])
corrplot(cor, method = "color") # heat map
ggplot(data = melted_correlation_sorted, aes(x=Var2, y=Var1, fill=value)) +
geom_tile() # life expectancy only

# ----- 5.) Model -----
# Create Training and Test data
set.seed(100)
train_dataset <- subset(dataset_2, select=c(4:23))
train_rows <- sample(1:nrow(train_dataset), 0.7*nrow(train_dataset))
train_data <- train_dataset[train_rows, ]
test_data <- train_dataset[-train_rows, ]

# Multiple Linear Regression
model_lm <- lm(`Life_Expectancy` ~ ., data = train_data)
predicted_lm <- predict(model_lm, test_data)

model_lm_fit_1 <- lm(`Life_Expectancy` ~ `Adult_Mortality` +
                       `Infant_Deaths` + `BMI` + `Under_Five_Deaths` +
                       `Polio` + `Diphtheria` + `HIV_AIDS`+
`Income_Composition` +
                       `Schooling` + `Status_type`, data = train_data)
predicted_lm_fit_1 <- predict(model_lm_fit_1, test_data)

# Random Forest
model_rf <- randomForest(`Life_Expectancy` ~ ., data = train_data)
predicted_rf <- predict(model_rf, test_data)

# Gradient Boosting
model_gbm <- gbm(`Life_Expectancy` ~ . ,data = train_data, distribution =
"gaussian",
                 n.trees = 10000, shrinkage = 0.01, interaction.depth = 1)
n.trees = seq(from=100 ,to=10000, by=100)
predicted_gbm <- predict(model_gbm, test_data, n.trees = n.trees)

# SVM
model_svm <- svm(`Life_Expectancy` ~ . ,
                 data = train_data, type = 'eps-regression', kernel =
'linear')
predicted_svm <- predict(model_svm, newdata = test_data)

# ----- 6.) Accuracy -----

# MSE
```

```r
MSE_lm = mse(predicted_lm, test_data$`Life_Expectancy`)
MSE_lm_fit_1 = mse(predicted_lm_fit_1, test_data$`Life_Expectancy`)
MSE_rf = mse(predicted_rf, test_data$`Life_Expectancy`)
MSE_gbm = mse(predicted_gbm, test_data$`Life_Expectancy`)
MSE_svm = mse(predicted_svm, test_data$`Life_Expectancy`)

# RMSE
error_lm <- predicted_lm - test_data$`Life_Expectancy`
RMSE_lm = sqrt(mean(error_lm^2))

error_lm_fit_1 <- predicted_lm_fit_1 - test_data$`Life_Expectancy`
RMSE_lm_fit_1 = sqrt(mean(error_lm_fit_1^2))

error_rf <- predicted_rf - test_data$`Life_Expectancy`
RMSE_rf = sqrt(mean(error_rf^2))

error_gbm <- predicted_gbm - test_data$`Life_Expectancy`
RMSE_gbm = sqrt(mean(error_gbm^2))

error_svm <- predicted_svm - test_data$`Life_Expectancy`
RMSE_svm <- sqrt(mean(error_svm^2))

# Min/Max Accuracy
minmax_lm <- mean(min(test_data$`Life_Expectancy`,
predicted_lm)/max(test_data$`Life_Expectancy`, predicted_lm))
minmax_lm_fit_1 <- mean(min(test_data$`Life_Expectancy`,
predicted_lm_fit_1)/max(test_data$`Life_Expectancy`, predicted_lm_fit_1))
minmax_rf <- mean(min(test_data$`Life_Expectancy`,
predicted_rf)/max(test_data$`Life_Expectancy`, predicted_rf))
minmax_gbm <- mean(min(test_data$`Life_Expectancy`,
predicted_gbm)/max(test_data$`Life_Expectancy`, predicted_gbm))
minmax_svm <- mean(min(test_data$`Life_Expectancy`,
predicted_svm)/max(test_data$`Life_Expectancy`, predicted_svm))

# ----- 7.) Summary -----
df_summary <- data.frame(
  Model = c("gbm", "lm", "lm_fit_1", "rf", "svm"),
  MSE = c(MSE_gbm, MSE_lm, MSE_lm_fit_1, MSE_rf, MSE_svm),
  RMSE = c(RMSE_gbm, RMSE_lm, RMSE_lm_fit_1, RMSE_rf, RMSE_svm),
  Min_Max_Acc = c(minmax_gbm, minmax_lm, minmax_lm_fit_1, minmax_rf,
minmax_svm)
)

summary(model_lm)
summary(model_lm_fit_1)
summary(model_rf)
summary(model_gbm)
summary(model_svm)
```