Project

11/30/2019

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```r
library(GGally)
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.5.2

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.5.2
```

```r
library(ROCR)
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.5.2
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(class)
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.2
```

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

Variable Explanations *Life expectancy* - Life Expectancy measured in ages

Adult Mortality - Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population) *infant deaths* - the number of Infant Deaths per 1000 population *Alcohol* - per capita (15+) alcohol consumption measured in liters of pure alcohol *percentage expenditure* - Expenditure on health as a percentage of Gross Domestic Product per capita(%) *Hepatitis B* - the percentage of immunization coverage among 1-year-olds *Measles* - number of reported cases per 1000 population *BMI* - Average Body Mass Index of entire population *under-five deaths* - the number of under-five deaths per 1000 population *Polio* - the percentage of immunization coverage among 1-year-olds *Total expenditure* - General government expenditure on health as a percentage of total government expenditure *Diphtheria* - the percentage of tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds *HIV/AIDS* - the number of deaths per 1 000 live births HIV/AIDS among 0-4-year olds *GDP* - Gross Domestic Product per capita measured in US dollars Population - the number of population of the country *thinness 1-19 years* - the percentage of prevalence of thinness among children and adolescents for Age 10 to 19 *thinness 5-9 years* - the percentage of 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 School

Before proceeding, clean data to transform it from longtidutinal to cross-sectional data.

First of all we load the data *Life Expectancy data.csv* and check the summary and structure to see

the datatypes of our variables and the possible absence of some values. We have 2938 observations of 22 variables.

Life expectancy is the dependent variable which we will predict throughout the project.

Life expectancy is statistical measure of a average

Later we take Status as a dependent variable and measure how Life Expectancy measures the Status of the country.

Status is a categorical variable with two possible levels: Developing and Developed.

```r
#Data Cleaning
life_expectancy <- read.csv("Life Expectancy Data.csv")
head(life_expectancy, n = 3)
```

```
##       Country Year     Status Life.expectancy Adult.Mortality infant.deaths
## 1 Afghanistan 2015 Developing            65.0             263            62
## 2 Afghanistan 2014 Developing            59.9             271            64
## 3 Afghanistan 2013 Developing            59.9             268            66
##   Alcohol percentage.expenditure Hepatitis.B Measles  BMI under.five.deaths
## 1    0.01               71.27962          65    1154 19.1                83
## 2    0.01               73.52358          62     492 18.6                86
## 3    0.01               73.21924          64     430 18.1                89
##   Polio Total.expenditure Diphtheria HIV.AIDS      GDP Population
## 1     6              8.16         65      0.1 584.2592   33736494
## 2    58              8.18         62      0.1 612.6965     327582
```

```
## 3    62               8.13         64      0.1 631.7450   31731688
##   thinness..1.19.years thinness.5.9.years Income.composition.of.resources
## 1                 17.2               17.3                           0.479
## 2                 17.5               17.5                           0.476
## 3                 17.7               17.7                           0.470
##   Schooling
## 1      10.1
## 2      10.0
## 3       9.9
```

```
str(life_expectancy)
```

```
## 'data.frame':    2938 obs. of  22 variables:
##  $ Country                        : Factor w/ 193 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Year                           : int  2015 2014 2013 2012 2011 2010 2009 2008 2007 2006 ...
##  $ Status                         : Factor w/ 2 levels "Developed","Developing": 2 2 2 2 2 2 2 2 2 2
##  $ Life.expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 57.3 ...
##  $ 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 0.03 ...
##  $ 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 1990 ...
##  $ 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              : 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 7.43 ...
##  $ 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 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 ...
```

As the dataset is comprised of five years, we take only year 2012, which has the least missing values. WE previously took Year 2015, which had only two complete cases.

By using dplyr we filter the data and took only Year 2012 with completed cases of 129.

```
life_expect <- life_expectancy %>%
  filter(complete.cases(.)) %>%
  filter(Year == "2012")
head(life_expect, n = 3)
```

```
##       Country Year     Status Life.expectancy Adult.Mortality infant.deaths
## 1 Afghanistan 2012 Developing            59.5             272            69
```

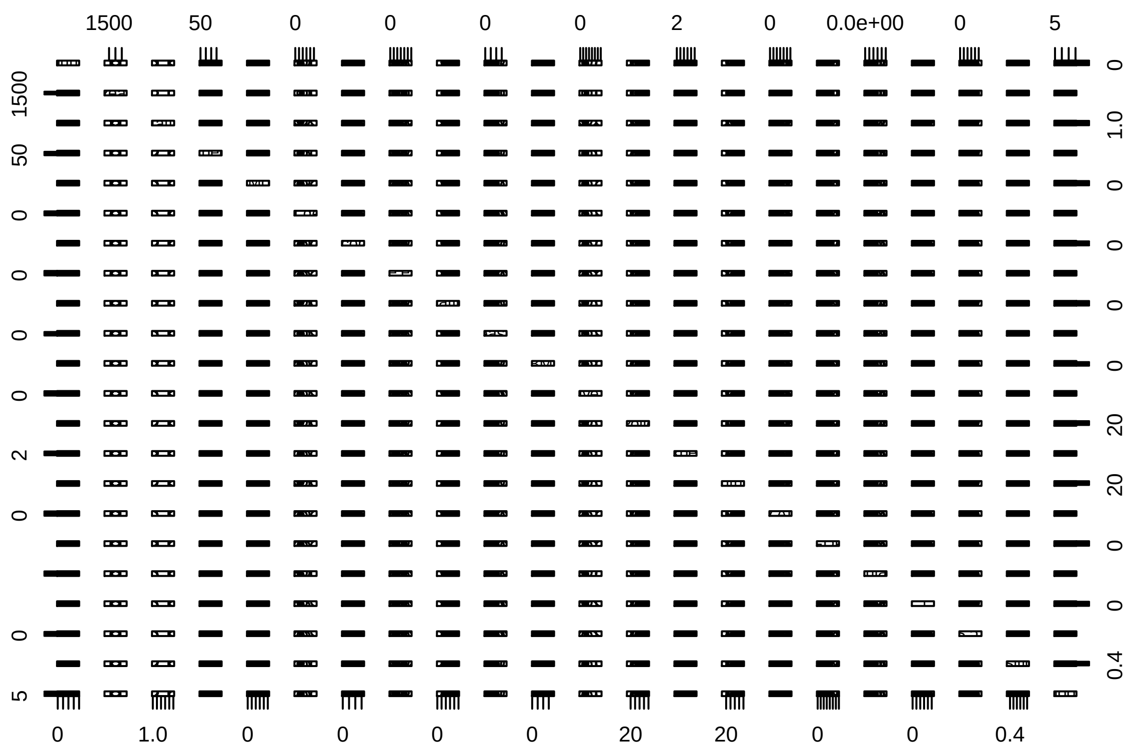
```
## 2     Albania 2012 Developing            76.9              86            0
## 3     Algeria 2012 Developing            75.1             113           21
##   Alcohol percentage.expenditure Hepatitis.B Measles  BMI under.five.deaths
## 1    0.01               78.18422          67    2787 17.6                93
## 2    5.14              412.44336          99       9 55.8                 1
## 3    0.66              555.92608          95      18 56.1                24
##   Polio Total.expenditure Diphtheria HIV.AIDS      GDP Population
## 1    67              8.52         67      0.1  669.959    3696958
## 2    99              5.59         99      0.1 4247.614       2941
## 3    95              6.14         95      0.1 5564.826   37565847
##   thinness..1.19.years thinness.5.9.years Income.composition.of.resources
## 1                 17.9               18.0                           0.463
## 2                  1.3                1.4                           0.752
## 3                  5.9                5.8                           0.732
##   Schooling
## 1       9.8
## 2      14.2
## 3      14.4
```

```r
str(life_expect)
```

```
## 'data.frame':    129 obs. of  22 variables:
##  $ Country                        : Factor w/ 193 levels "Afghanistan",..: 1 2 3 4 6 7 8 9 10 13 ...
##  $ Year                           : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ Status                         : Factor w/ 2 levels "Developed","Developing": 2 2 2 2 2 2 1 1 2 2
##  $ Life.expectancy                : num  59.5 76.9 75.1 56 75.9 74.4 82.3 88 71.9 77 ...
##  $ Adult.Mortality                : int  272 86 113 358 12 121 61 7 123 137 ...
##  $ infant.deaths                  : int  69 0 21 72 9 1 1 0 5 111 ...
##  $ Alcohol                        : num  0.01 5.14 0.66 8.24 8.35 ...
##  $ percentage.expenditure         : num  78.2 412.4 555.9 256.1 1133.6 ...
##  $ Hepatitis.B                    : int  67 99 95 75 91 95 91 92 88 94 ...
##  $ Measles                        : int  2787 9 18 4458 2 0 199 36 0 1986 ...
##  $ BMI                            : num  17.6 55.8 56.1 21.5 61 52.6 65 56.1 49.7 16.4 ...
##  $ under.five.deaths              : int  93 1 24 110 10 1 1 0 6 139 ...
##  $ Polio                          : int  67 99 95 75 99 96 92 92 92 94 ...
##  $ Total.expenditure              : num  8.52 5.59 6.14 3.3 5.2 ...
##  $ Diphtheria                     : int  67 99 95 75 91 95 92 92 89 94 ...
##  $ HIV.AIDS                       : num  0.1 0.1 0.1 2.6 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ GDP                            : num  670 4248 5565 4598 12970 ...
##  $ Population                     : num  3696958 2941 37565847 259615 4296739 ...
##  $ thinness..1.19.years           : num  17.9 1.3 5.9 8.8 1 2 0.6 1.8 2.8 18.5 ...
##  $ thinness.5.9.years             : num  18 1.4 5.8 8.6 0.9 2.1 0.6 2 2.8 19 ...
##  $ Income.composition.of.resources: num  0.463 0.752 0.732 0.508 0.822 0.732 0.93 0.884 0.742 0.557
##  $ Schooling                      : num  9.8 14.2 14.4 10.3 17.2 12.7 20.1 15.7 11.8 9.9 ...
```

Ploting variables together to see the highest correlations.

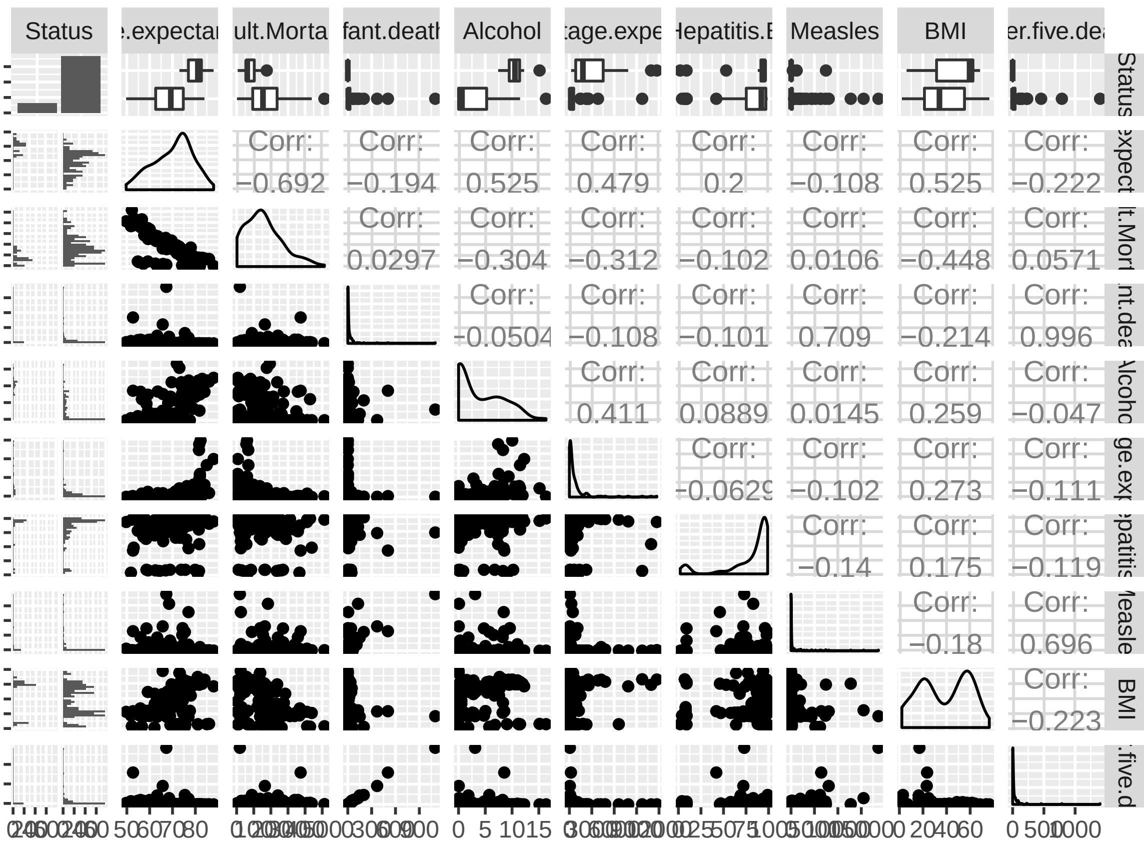
We tried to plot the dataset with plot function, but because of the high number of variables it failed to visualize the correlations.



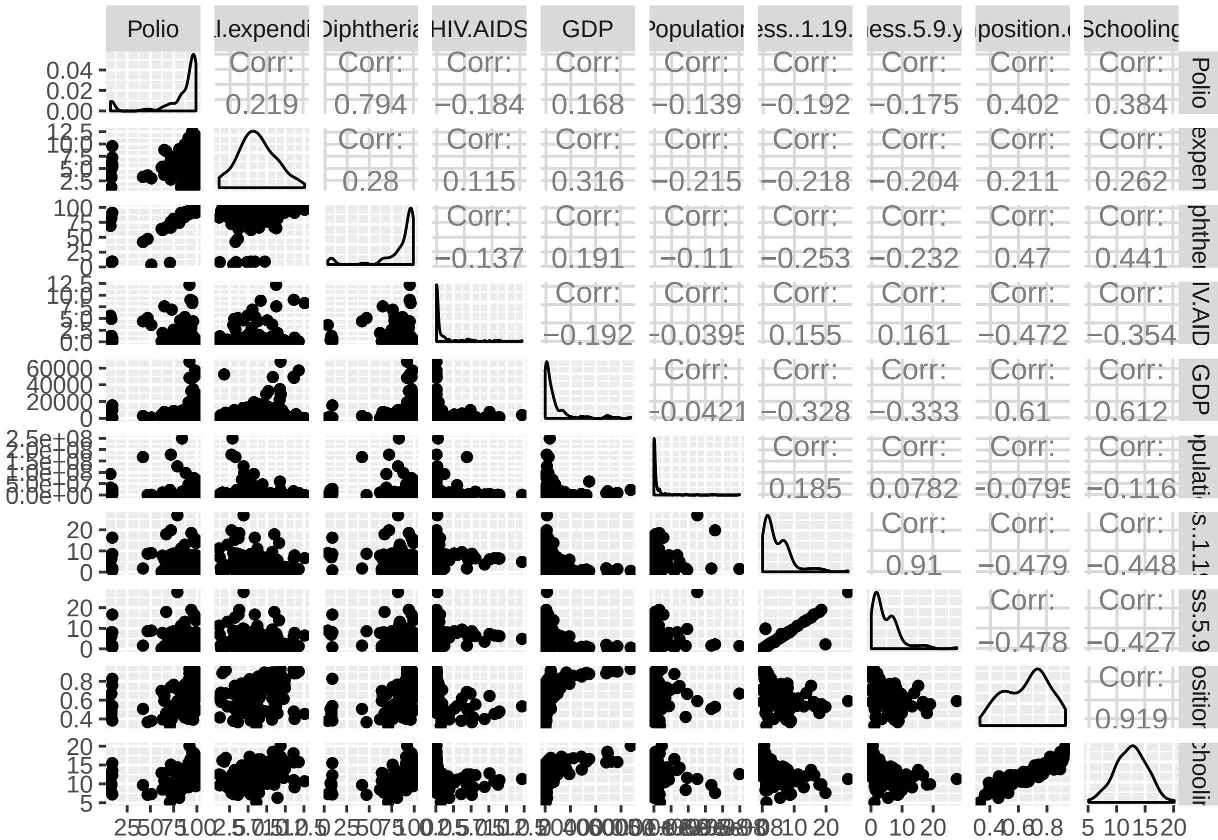
We use ggpairs to see the correlations between different variables, since the number of variables was too many, we did it twice by seperating into two subgroups: [3:12], [13:22]. The data had too many variables resulting in non-readability of the data. We applied ggpairs function in order to understand overall correlations between the variables. See next for correlations of selected variables with life expectancy.

```
ggpairs(life_expect,3:12)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggpairs(life_expect,13:22)
```

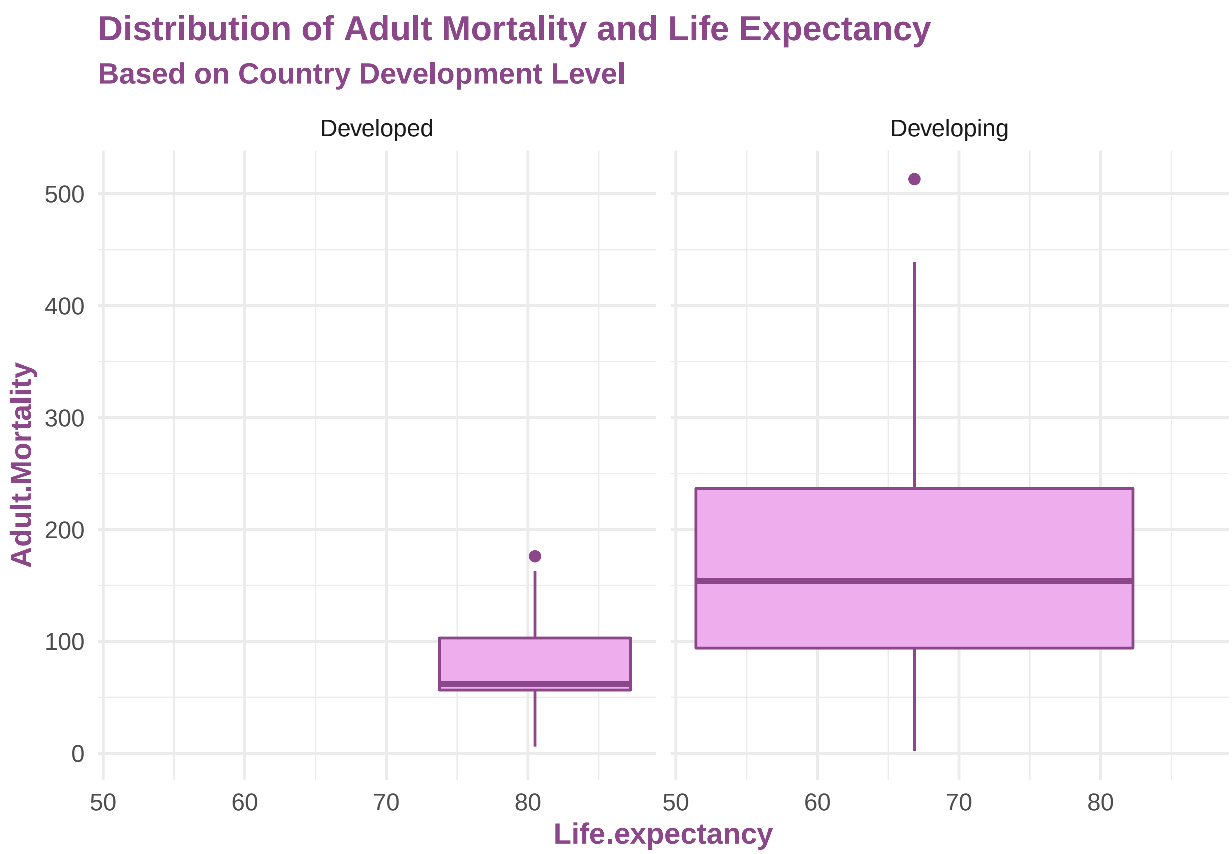


##Based on the ggpair output we formulated the general idea about the variable correlation in the dataset, thus trying to eliminate the variables that have high correlation with each other. In this way we avoide multicollinearity.

Afterwards we decided to plot some of the uncorrelated variables with ggplot in order to see how their changes affect the dependent variable Life Expectancy.

```
ggplot(life_expect, aes(Life.expectancy, Adult.Mortality)) +
  geom_boxplot(fill = "plum2", color = "orchid4")+
  facet_grid(~Status) +
  labs(title = "Distribution of Adult Mortality and Life Expectancy", subtitle = "Based on Country Devel
  theme_minimal()+
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank())
```

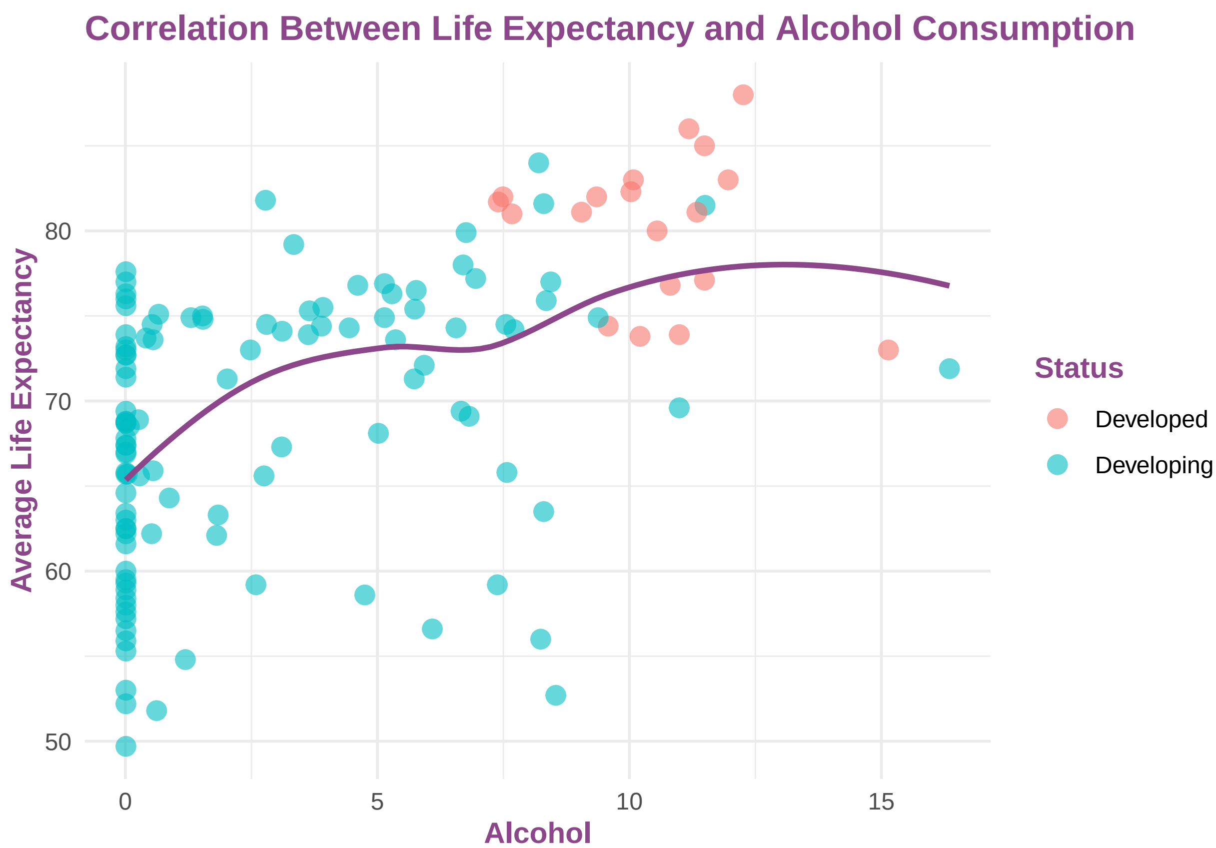
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



##The boxplot shows that Life Expectancy for Developing countries has wider range of 53-83, and Life Expectancy of Developed countries has a range of from 74-88. ##Adult Mortality of Developed countries has more than two time higher mean compared to Developing countries. ##Both levels of Status have outliers.

```
ggplot(life_expect, aes(Alcohol, Life.expectancy, col = Status)) +
  geom_point(size = 3, alpha = 0.6)+
  theme_minimal() +
  scale_colour_discrete(drop=TRUE,
        limits = levels(life_expect$Status)) +
  geom_smooth(se = FALSE, col = "orchid4") +
  labs(title = "Correlation Between Life Expectancy and Alcohol Consumption", y = "Average Life Expectan
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank())
```

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




```
corr.test(life_expect$Life.expectancy, life_expect$Alcohol)
```

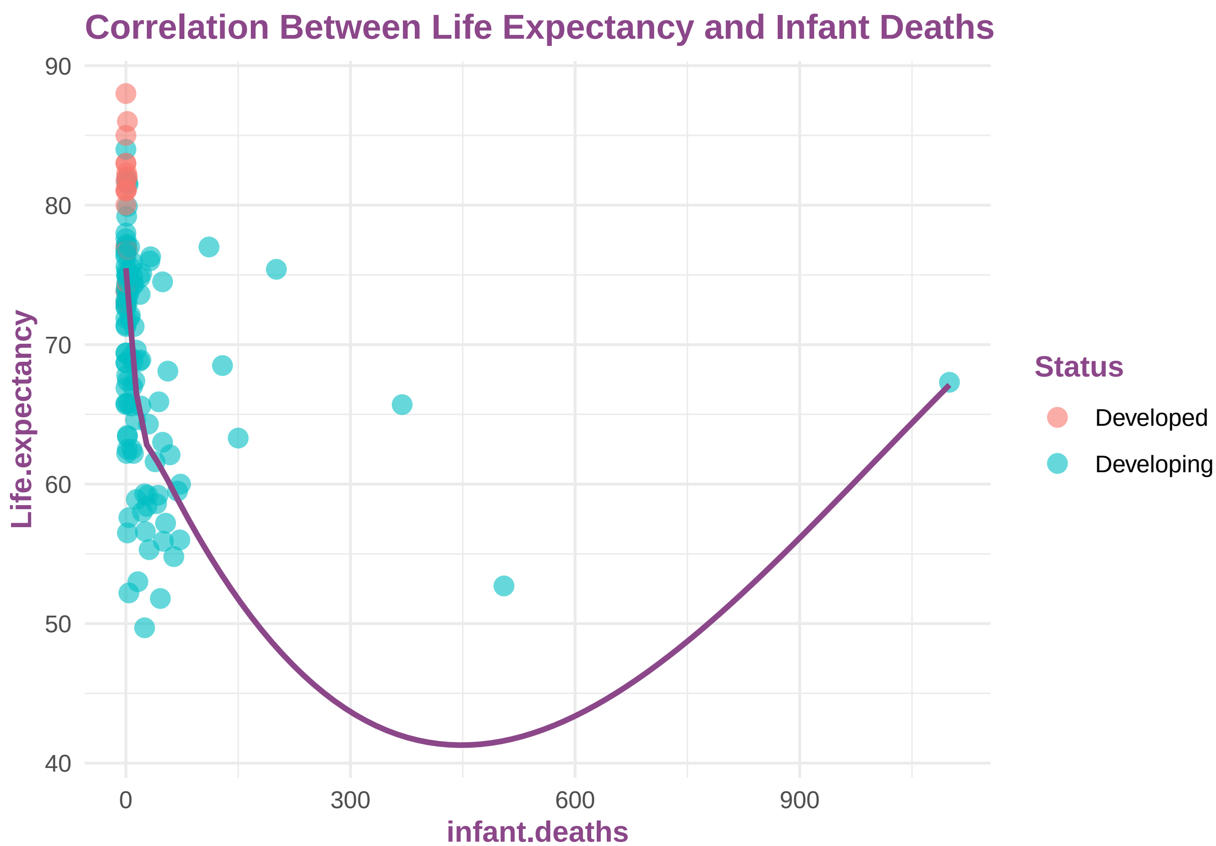
```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$Alcohol)
## Correlation matrix
## [1] 0.53
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

The correlation coefficient between the Alcohol and Life Expectancy is 53%, which implies that Alcohol Consumption explains the variation in Life Expectancy weakly.

```
ggplot(life_expect, aes(infant.deaths,Life.expectancy, col = Status)) +
  geom_point(size = 3, alpha = 0.6)+
  scale_colour_discrete(drop=TRUE,
        limits = levels(life_expect$Status)) +
  geom_smooth(se = FALSE, col = "orchid4") +
  theme_minimal() +
  labs(title = "Correlation Between Life Expectancy and Infant Deaths") +
```

```r
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank())
```

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



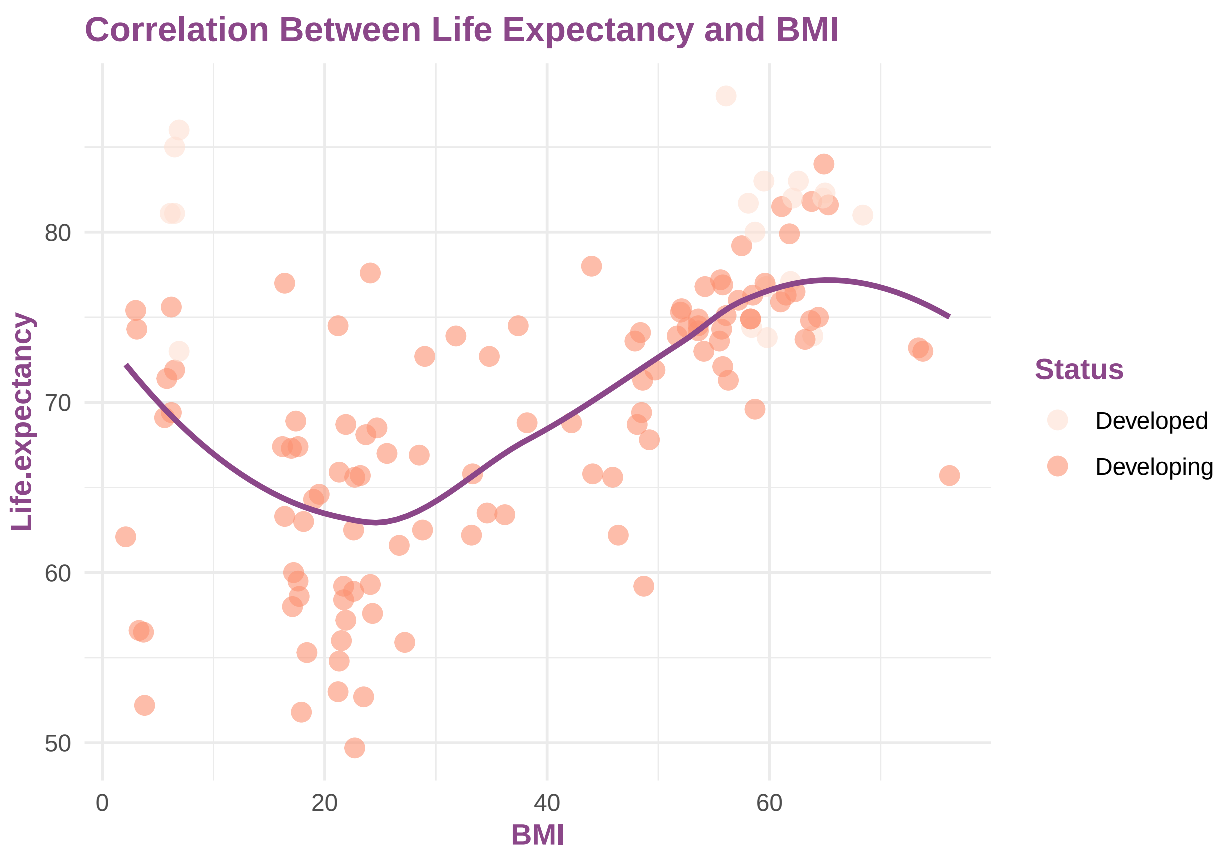

```r
corr.test(life_expect$Life.expectancy, life_expect$infant.deaths)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$infant.deaths)
## Correlation matrix
## [1] -0.19
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0.03
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

As the chart shows there is almsot no correlation between the two variables. Correlation coefficient shows the same with its -0.19 figure, implying that they are weakly negatively correlated.

```r
ggplot(life_expect, aes(BMI,Life.expectancy, col = Status)) +
  geom_point(size = 3, alpha = 0.6)+
    scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "orchid4") +
  theme_minimal() +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  labs(title = "Correlation Between Life Expectancy and BMI")
```

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



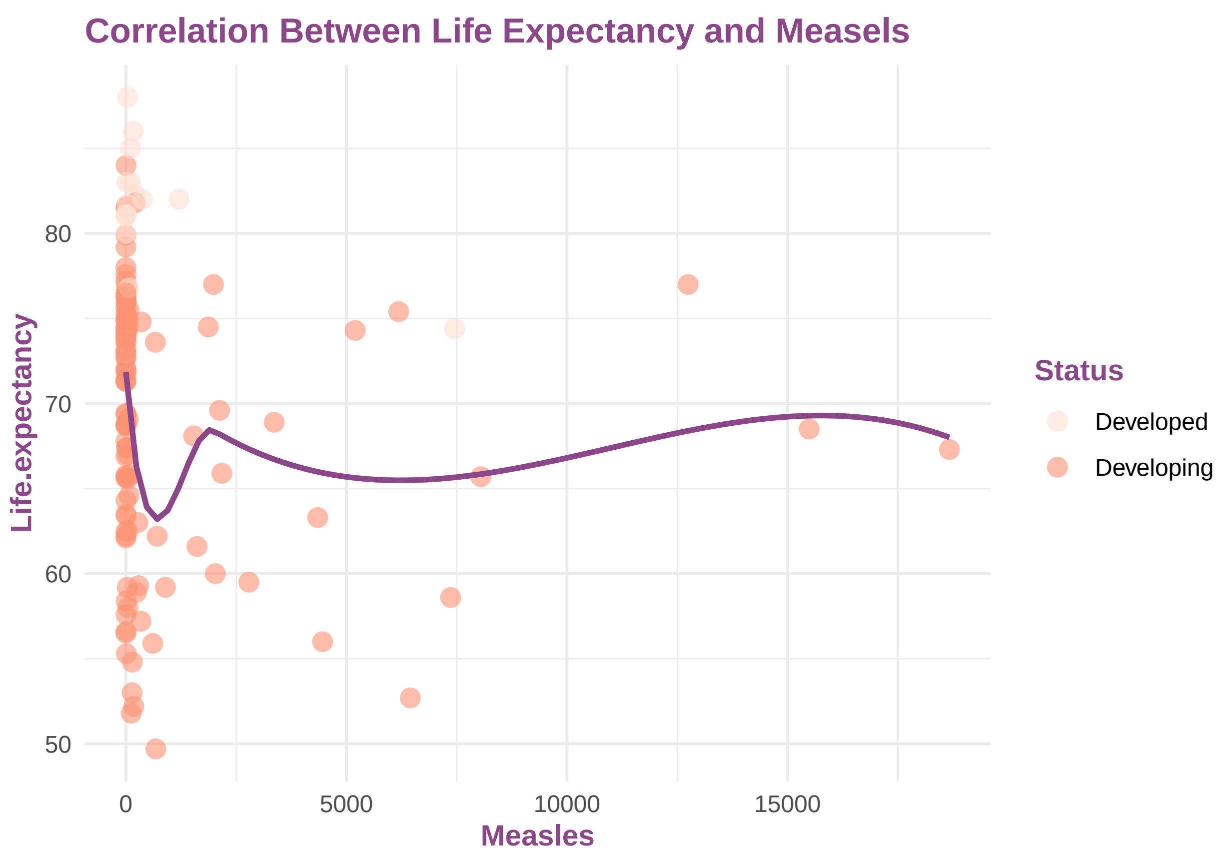

```r
corr.test(life_expect$Life.expectancy, life_expect$BMI)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$BMI)
## Correlation matrix
## [1] 0.52
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

BMI (Body Mass Index) is moderately correlated with dependent variables. The correlation coefficient is 52%.

```
ggplot(life_expect, aes(Measles, Life.expectancy, col = Status)) +
  geom_point(size = 3, alpha = 0.6)+
    scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "orchid4") +
  theme_minimal() +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  labs(title = "Correlation Between Life Expectancy and Measels")
```

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




```
corr.test(life_expect$Life.expectancy, life_expect$Measles)
```

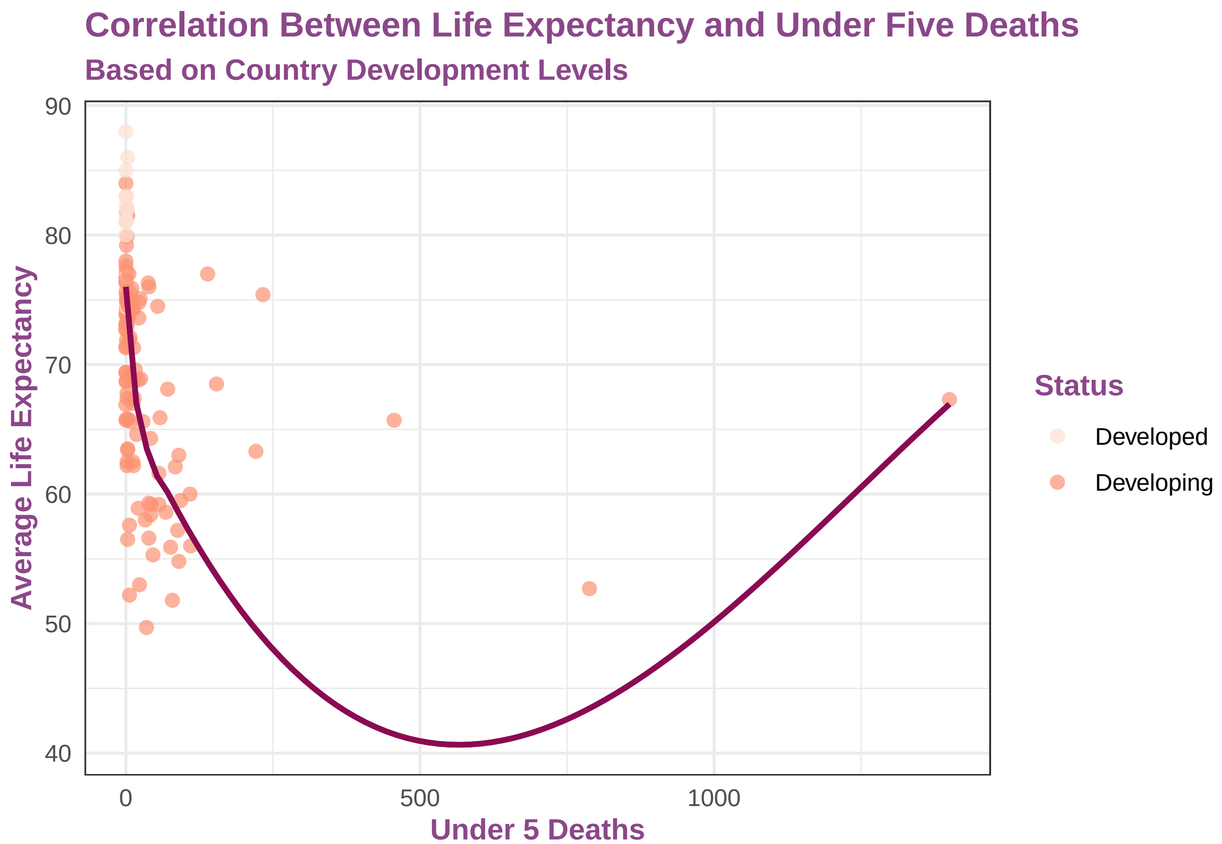
```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$Measles)
## Correlation matrix
## [1] -0.11
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
```

```
## [1] 0.22
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

The variable Measles is weakly negatively correlated with Life Expectancy.

```r
ggplot(life_expect, aes(under.five.deaths, Life.expectancy, col = Status)) +
 geom_point(size = 2, alpha = 0.7)+
  theme_bw() +
  scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "blueviolet") +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  geom_smooth(se = FALSE, col = "deeppink4") +
  labs(title = "Correlation Between Life Expectancy and Under Five Deaths", x = "Under 5 Deaths", y = "
```

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



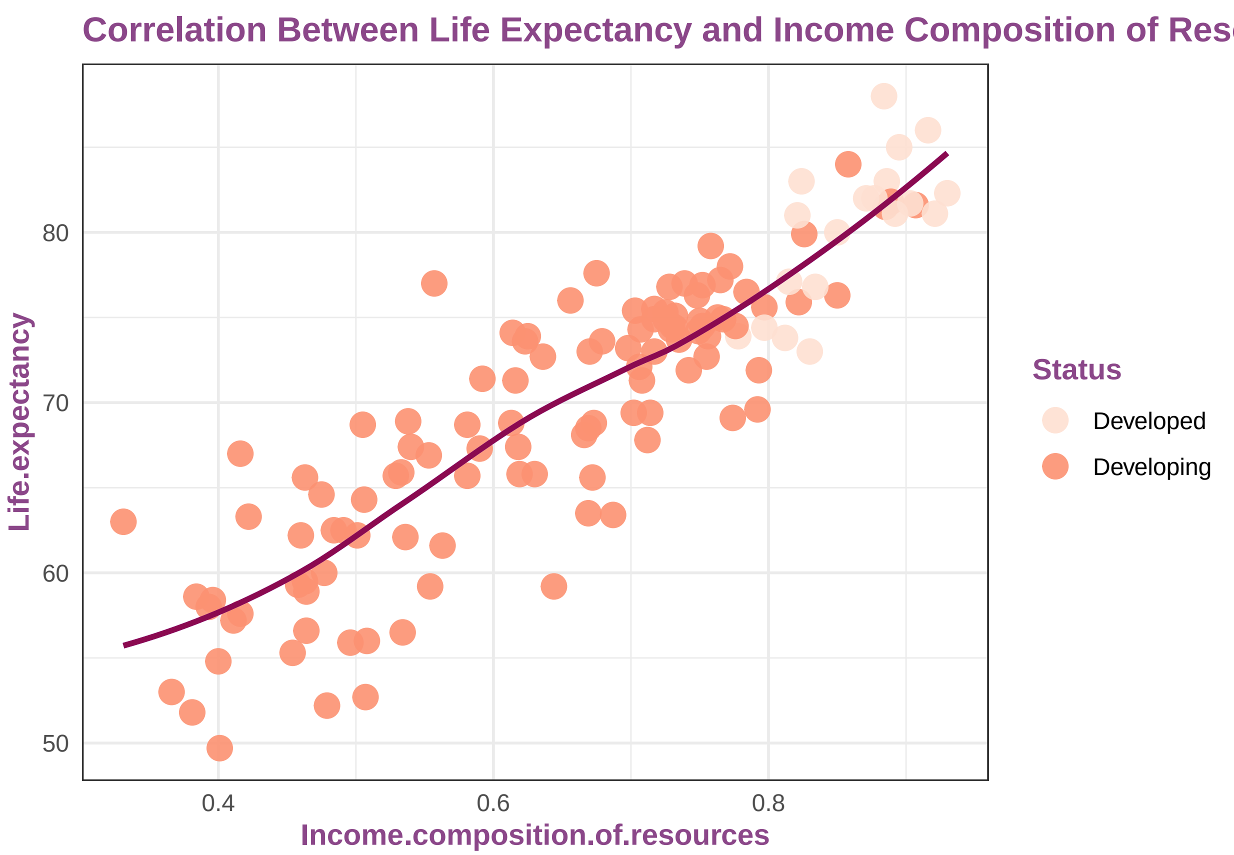

```r
corr.test(life_expect$Life.expectancy, life_expect$under.five.deaths)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$under.five.deaths)
```

```
## Correlation matrix
## [1] -0.22
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0.01
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
ggplot(life_expect, aes(Income.composition.of.resources,Life.expectancy, col = Status)) +
  geom_point(size = 4, alpha = 0.9)+
  theme_bw() +
  scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "blueviolet") +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  geom_smooth(se = FALSE, col = "deeppink4") +
  labs(title = "Correlation Between Life Expectancy and Income Composition of Resources")
```

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



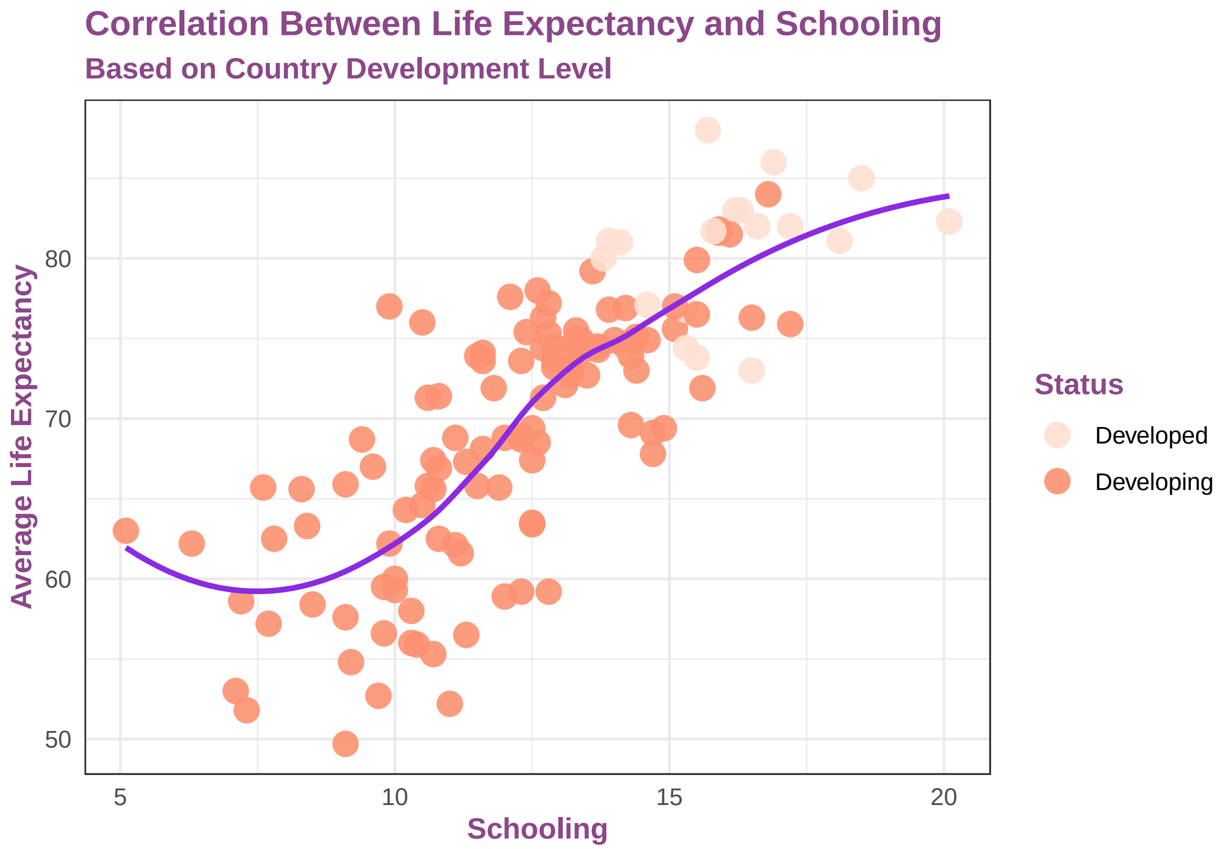
##There is a strong positive correlation between Income Composition of Resources and the Life Expectancy of the coutnries. ##Variation in Life Expectancy can be 89% explained by the variation in Income composition.

```
corr.test(life_expect$Life.expectancy, life_expect$Income.composition.of.resources)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$Income.composition.of.resources)
## Correlation matrix
## [1] 0.89
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
ggplot(life_expect, aes(Schooling,Life.expectancy, col = Status)) +
  geom_point(size = 4, alpha = 0.9)+
  theme_bw() +
  scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "blueviolet") +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  labs(title = "Correlation Between Life Expectancy and Schooling", y = "Average Life Expectancy", subti
```

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



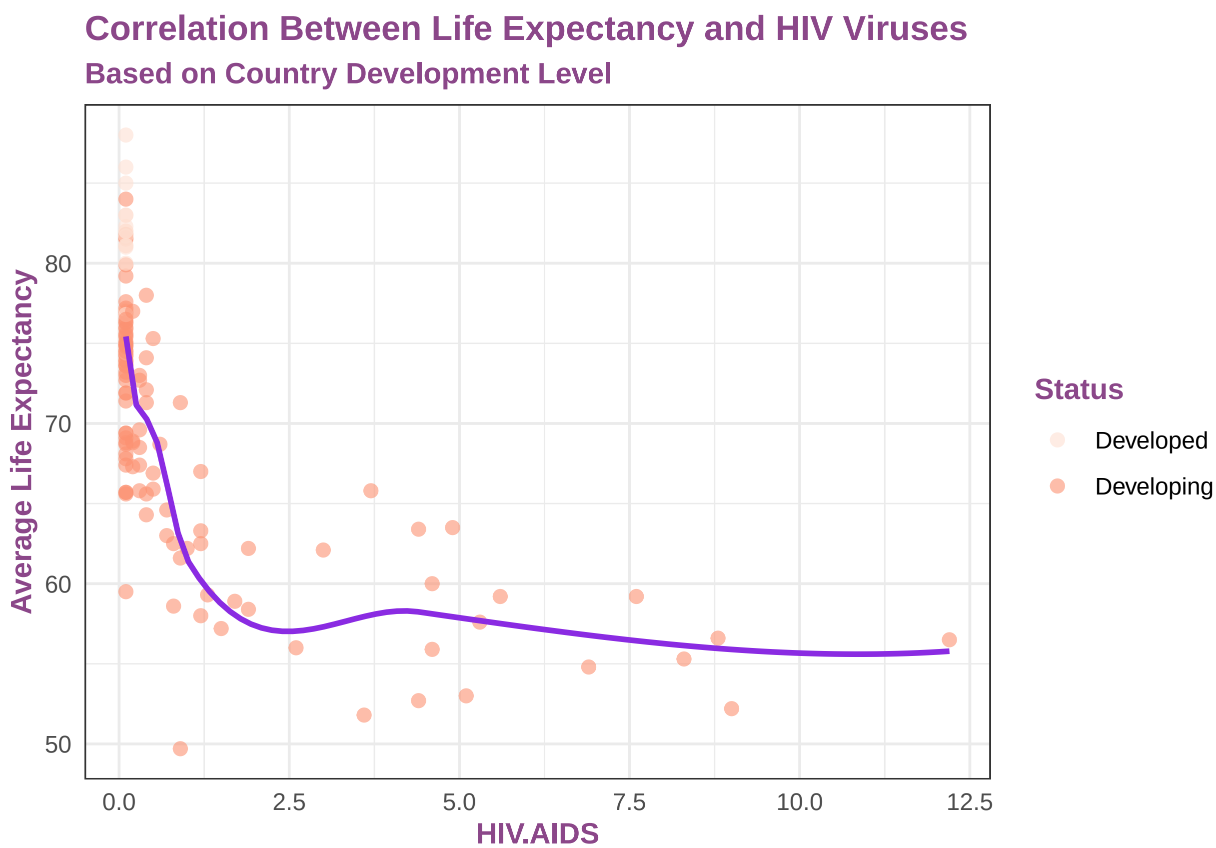
```r
corr.test(life_expect$Life.expectancy, life_expect$Schooling)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$Schooling)
## Correlation matrix
## [1] 0.78
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Here again we see high correlation, this time between schooling the the dependent variable. The correlation coefficient is 78%.

```r
ggplot(life_expect, aes(HIV.AIDS,Life.expectancy, col = Status)) +
  geom_point(size = 2, alpha = 0.6)+
  theme_bw() +
  scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "blueviolet") +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  labs(title = "Correlation Between Life Expectancy and HIV Viruses", y = "Average Life Expectancy", sul
```

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



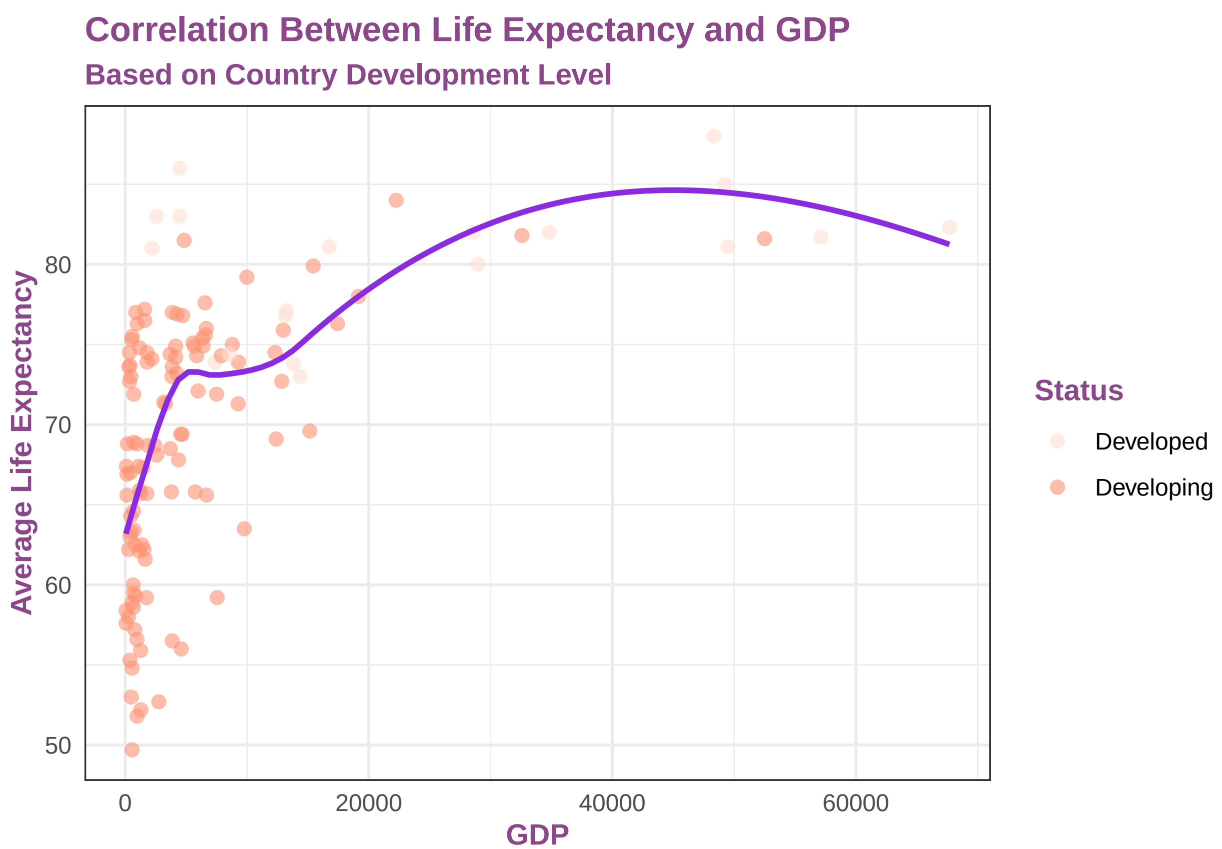

```r
corr.test(life_expect$Life.expectancy, life_expect$HIV.AIDS)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$HIV.AIDS)
## Correlation matrix
## [1] -0.65
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Here we have a high negative correlation(-65%) between HIV/AIDs and Life Expectancy, since the increase in HIV decreases life expectancy.

```r
ggplot(life_expect, aes(GDP,Life.expectancy, col = Status)) +
  geom_point(size = 2, alpha = 0.6)+
  theme_bw() +
  scale_color_brewer(palette = "Reds") +
  geom_smooth(se = FALSE, col = "blueviolet") +
  theme(title = element_text(color = "orchid4", face = "bold"),
        axis.title = element_text(color = "orchid4", face = "bold"),
        axis.ticks = element_blank()) +
  labs(title = "Correlation Between Life Expectancy and GDP", y = "Average Life Expectancy", subtitle =
```

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




```r
corr.test(life_expect$Life.expectancy, life_expect$GDP)
```

```
## Call:corr.test(x = life_expect$Life.expectancy, y = life_expect$GDP)
## Correlation matrix
## [1] 0.54
## Sample Size
## [1] 129
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

The correlation coefficient between Life Expectancy and GDP is moderate(54%).

After having an overall idea about our dataset, its individual variables and their possible relationships, we move on to building models and measuring the significance of the variables.

Linear Regression

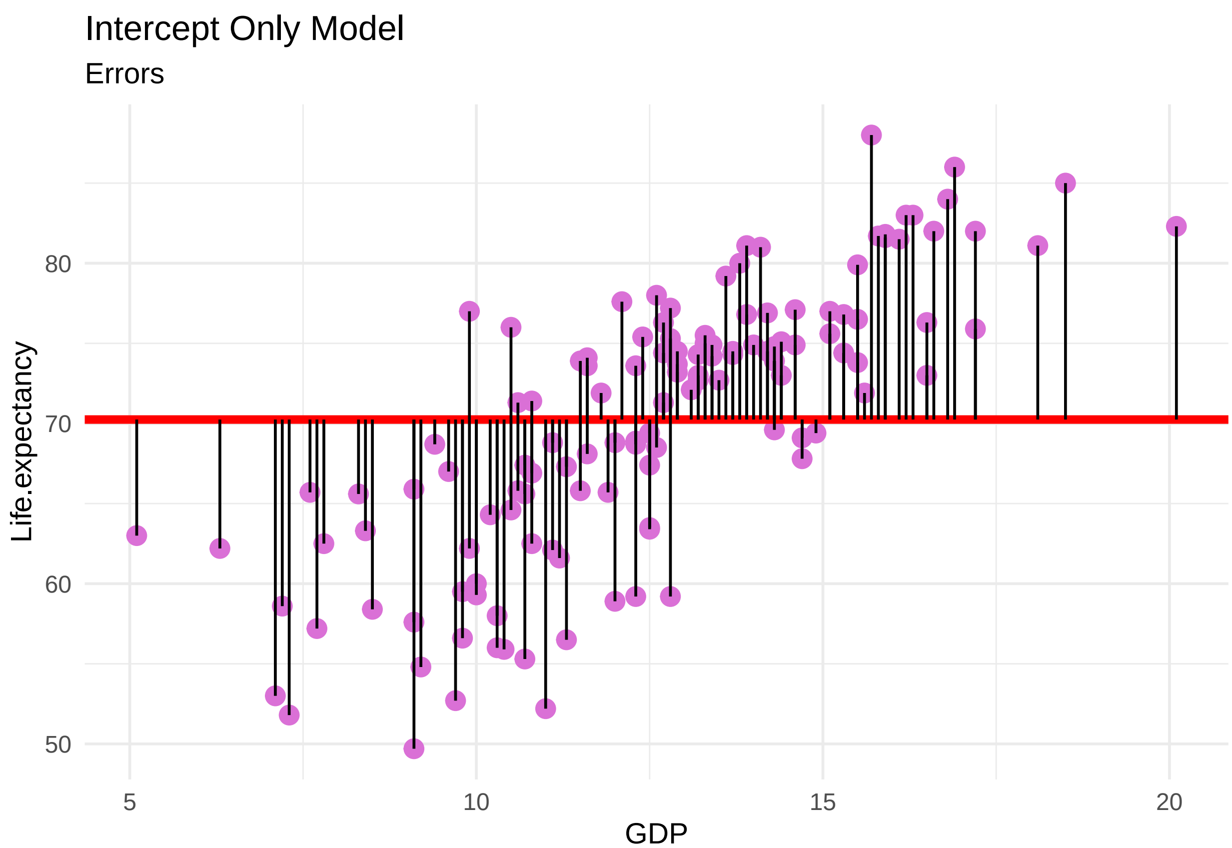
We start with linear regression to identify the best model explaining the life expectancy.

```
set.seed(2)
#Intercept only model
model0 <- lm(Life.expectancy~1, data = life_expect)
summary(model0)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ 1, data = life_expect)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.545  -5.945   2.455   5.655  17.755
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.2450     0.7496   93.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.514 on 128 degrees of freedom
```

With Intercept only model, we calculated the mean of the variable Life Expectancy.

```
ggplot(life_expect, aes(x = Schooling, y = Life.expectancy)) +
  geom_point(shape = 19, size = 3, col = "orchid") +
  geom_hline(yintercept = mean(life_expect$Life.expectancy), col = "red", size = 1.5) +
  theme_minimal() +
  geom_segment(aes(xend = Schooling, yend = mean(life_expect$Life.expectancy, alpha = 0.2, col = "purpl
  labs(x = "GDP", y = "Life.expectancy", title = "Intercept Only Model", subtitle = "Errors")
```



##This model shows the errors of Life Expectancy with different values of GDP compared to the mean value of Life Expectancy.

```
model1 <- lm(Life.expectancy~BMI+Income.composition.of.resources+Schooling+GDP, data = life_expect)
summary(model1)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ BMI + Income.composition.of.resources +
##     Schooling + GDP, data = life_expect)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.177  -2.215   0.043   2.130  11.805
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.925e+01  1.884e+00  20.835  < 2e-16 ***
## BMI                             3.167e-02  1.926e-02   1.644   0.1026
## Income.composition.of.resources 5.742e+01  6.106e+00   9.404 3.51e-16 ***
## Schooling                      -6.631e-01  3.228e-01  -2.054   0.0421 *
## GDP                             1.301e-05  3.547e-05   0.367   0.7145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.854 on 124 degrees of freedom
```

```
## Multiple R-squared:  0.8014, Adjusted R-squared:  0.795
## F-statistic: 125.1 on 4 and 124 DF,  p-value: < 2.2e-16
```

Although we found correlation between the independent variables and the dependent variable, we can see that the P-values for BMI, Schooling and GDP are higher than the alpha, which means that the variables are not significant. High correlations cause overfitting of data, thus, we eliminate those next.

```r
#Eliminating the overfitiing variables.
model2 <- lm(Life.expectancy~Income.composition.of.resources+infant.deaths+Adult.Mortality+HIV.AIDS, da
summary(model2)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Income.composition.of.resources +
##     infant.deaths + Adult.Mortality + HIV.AIDS, data = life_expect)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9488 -1.9472  0.0506  1.7992  9.6863
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     49.037482   1.933817  25.358  < 2e-16 ***
## Income.composition.of.resources 36.790527   2.359241  15.594  < 2e-16 ***
## infant.deaths                   -0.003610   0.002476  -1.458  0.14743
## Adult.Mortality                 -0.012515   0.003328  -3.761  0.00026 ***
## HIV.AIDS                        -1.008684   0.150850  -6.687 6.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.076 on 124 degrees of freedom
## Multiple R-squared:  0.8736, Adjusted R-squared:  0.8695
## F-statistic: 214.2 on 4 and 124 DF,  p-value: < 2.2e-16
```

Even though the model adjusted R-squared improved (0.8817 or 88.17%), we can see that Infant.deaths variable is still insignificant. Thus, we need to eliminate that one too before proceeding with the rest of variables.

```r
#Eliminating variable "Infant.deaths".
model3 <- lm(Life.expectancy~Income.composition.of.resources+Adult.Mortality+HIV.AIDS, data = life_expe
summary(model3)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Income.composition.of.resources +
##     Adult.Mortality + HIV.AIDS, data = life_expect)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -7.8579 -2.0067 -0.1016  1.8318  9.4897 
## 
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                   48.344289   1.882858  25.676  < 2e-16 ***
## Income.composition.of.resources 37.543898   2.312278  16.237  < 2e-16 ***
## Adult.Mortality               -0.012010   0.003324  -3.613 0.000437 ***
## HIV.AIDS                      -1.006020   0.151516  -6.640 8.62e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.09 on 125 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8683 
## F-statistic: 282.3 on 3 and 125 DF,  p-value: < 2.2e-16
```

After eliminating the infant.deaths variable, we see that the model adjusted R-squared remains high at 0.8818. Next, we are going to add variables of interest based on literature review: alcohol, Hepatitis B, and Polio

```
#Adding variables of interest
model4 <- lm(Life.expectancy~Income.composition.of.resources+Adult.Mortality+HIV.AIDS+Hepatitis.B+Polio+
               thinness..1.19.years+Alcohol, data = life_expect)
summary(model4)
```

```
## 
## Call:
## lm(formula = Life.expectancy ~ Income.composition.of.resources + 
##     Adult.Mortality + HIV.AIDS + Hepatitis.B + Polio + thinness..1.19.years + 
##     Alcohol, data = life_expect)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -7.6915 -1.9223  0.0163  1.7453  9.8686 
## 
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                   48.237526   2.388236  20.198  < 2e-16 ***
## Income.composition.of.resources 39.227687   3.260488  12.031  < 2e-16 ***
## Adult.Mortality               -0.012053   0.003355  -3.592 0.000475 ***
## HIV.AIDS                      -1.008167   0.153590  -6.564 1.37e-09 ***
## Hepatitis.B                    0.018268   0.013194   1.385 0.168742    
## Polio                         -0.024983   0.014603  -1.711 0.089689 .  
## thinness..1.19.years          -0.030923   0.069498  -0.445 0.657156    
## Alcohol                       -0.066948   0.089195  -0.751 0.454364    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.092 on 121 degrees of freedom
## Multiple R-squared:  0.8753, Adjusted R-squared:  0.8681 
## F-statistic: 121.3 on 7 and 121 DF,  p-value: < 2.2e-16
```

The adjusted R squared improved, but we still need to eliminate the two variables that are statistically insignificant: Polio, Alcohol, Hepatitis B and thinness of the population.

```
#Final model
model_final <- lm(Life.expectancy~Income.composition.of.resources+Adult.Mortality+HIV.AIDS, data = life_
summary(model_final)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ Income.composition.of.resources +
##     Adult.Mortality + HIV.AIDS, data = life_expect)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8579 -2.0067 -0.1016  1.8318  9.4897
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     48.344289   1.882858  25.676  < 2e-16 ***
## Income.composition.of.resources 37.543898   2.312278  16.237  < 2e-16 ***
## Adult.Mortality                 -0.012010   0.003324  -3.613 0.000437 ***
## HIV.AIDS                        -1.006020   0.151516  -6.640 8.62e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.09 on 125 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8683
## F-statistic: 282.3 on 3 and 125 DF,  p-value: < 2.2e-16
```

The final adjusted R-squared is 0.8683 or 86.83%, which is high taking into account the fact that the independent variables are not intercorrelated. The variables explaining Life expectancy are Income Composition of Resources, Adult Mortality, and HIV Aids available.

Hereby we devide our dependent variable into four categories with thresholds of below 61, between 61 and 70, between 70 and 75, above 75.

The division is implemented, since the variable is numeric and we need to have categories in order to continue further with our predictions.

```
life_expect$Life.expectancy.cat1 <- ifelse(life_expect$Life.expectancy >= 75, 4,
                                     ifelse(life_expect$Life.expectancy < 75 & life_expect$Life.expecta
                                            ifelse(life_expect$Life.expectancy <= 70 & life_expect$Life
                                                   ifelse(life_expect$Life.expectancy < 61, 1, NA))))

head(life_expect$Life.expectancy.cat1, n = 5)
```

```
## [1] 1 4 4 1 4
```

```r
class(life_expect$Life.expectancy.cat1)
```

```
## [1] "numeric"
```

```r
life_expect$Life.expectancy.cat1 <- as.factor(life_expect$Life.expectancy.cat1)
life_expect$Life.expectancy.cat1 <- factor(life_expect$Life.expectancy.cat1, levels = c(1,2,3,4),
                                         labels = c("Very low", "Low", "Medium", "High"))
class(life_expect$Life.expectancy.cat1)
```

```
## [1] "factor"
```

Since the categories of the variables very numeric we changed it to factor.

For the rest of our project, we will be doing predictions and checking the accuracy of models based on different methods.

For the predcition we need to have our dataset seperated into train and test sets, with 30 and 70 weigths respectively.

```r
#Constructing training and testing datasets, which willl be used for the rest of our codes.
set.seed(2)
index <- createDataPartition(life_expect$Life.expectancy.cat1, p = 0.7, list = FALSE)
Train <- life_expect[index,]
Test <- life_expect[-index,]
```

Naive Bayes Model

Naive bias model helps us to solve the classificaiton probelsm using probabilistic approach. It assumes the independent variables used are not dependent.

```r
#Constructing Naive Bayes Model to determine High, Low, Medium and Very Low Classes of Life Expectancy.
model_NB = naiveBayes(Life.expectancy.cat1~Income.composition.of.resources+Adult.Mortality+HIV.AIDS, da
names(model_NB)
```

```
## [1] "apriori"   "tables"    "levels"    "isnumeric" "call"
```

```r
model_NB$apriori
```

```
## Y
## Very low      Low   Medium     High
##       16       26       23       28
```

```r
model_NB$tables
```

```
## $Income.composition.of.resources
##           Income.composition.of.resources
## Y                [,1]       [,2]
##   Very low 0.4596875 0.07696555
##   Low      0.5961154 0.10972632
##   Medium   0.7231304 0.06156468
##   High     0.7998214 0.08742830
##
## $Adult.Mortality
##           Adult.Mortality
## Y                [,1]      [,2]
##   Very low 273.25000 147.13962
##   Low      183.30769  86.93251
##   Medium   136.91304  46.58122
##   High      81.03571  43.62337
##
## $HIV.AIDS
##           HIV.AIDS
## Y                [,1]       [,2]
##   Very low 4.2687500 3.25447768
##   Low      0.8692308 1.32386381
##   Medium   0.1913043 0.19048513
##   High     0.1285714 0.09371803
```

```r
pred_class_NB = predict(model_NB, newdata = Test)
confusion_NB = confusionMatrix(pred_class_NB, Test$Life.expectancy.cat1, positive = "High")
confusion_NB
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Very low Low Medium High
##   Very low        5   0      0    0
##   Low             1   7      0    0
##   Medium          0   2      6    1
##   High            0   1      3   10
##
## Overall Statistics
##
##                Accuracy : 0.7778
##                  95% CI : (0.6085, 0.8988)
##     No Information Rate : 0.3056
##     P-Value [Acc > NIR] : 7.104e-09
##
##                   Kappa : 0.6972
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Very low Class: Low Class: Medium Class: High
## Sensitivity                   0.8333     0.7000        0.6667      0.9091
## Specificity                   1.0000     0.9615        0.8889      0.8400
## Pos Pred Value                1.0000     0.8750        0.6667      0.7143
```

```
## Neg Pred Value                    0.9677      0.8929        0.8889       0.9545
## Prevalence                        0.1667      0.2778        0.2500       0.3056
## Detection Rate                    0.1389      0.1944        0.1667       0.2778
## Detection Prevalence              0.1389      0.2222        0.2500       0.3889
## Balanced Accuracy                 0.9167      0.8308        0.7778       0.8745
```

Model accuracy is 0.722 or 72.22%, which is lower than the model accuracy for the final linear regression.

It is much higher than the No Information Rate(30.56%) of the model.

```
pred_prob_NB = predict(model_NB, newdata = Test, type = "raw")
head(pred_prob_NB)
```

```
##          Very low         Low       Medium         High
## [1,] 2.782867e-01 0.721600368 0.0001083864 4.500211e-06
## [2,] 1.870996e-06 0.007776065 0.4714595496 5.207625e-01
## [3,] 3.715153e-10 0.000548657 0.0109790846 9.884723e-01
## [4,] 4.379644e-07 0.006956238 0.4535532425 5.394901e-01
## [5,] 1.411209e-05 0.017881483 0.6860229215 2.960815e-01
## [6,] 4.139709e-08 0.001549599 0.0164671154 9.819832e-01
```

Hereby we show the predicted probability of each class for each case.

KNN

Knn is lazy learning algorithm, it does not create model but helps to predict the classification of a new sample point.

```
#We are checking on which column our dependent variable is and which ones are not numeric.
match("Life.expectancy.cat1", names(Train))
```

```
## [1] 23
```

```
#We remove all the categorical variables from the dataset, as knn requires numeric ones and we leave on
Train_knn <- Train[, c(5,16,21)]
Test_knn <- Test[, c(5,16,21)]
knn1 <- knn(train = Train_knn, test = Test_knn, k = 10, cl = Train$Life.expectancy.cat1)
```

When we take K as a random number, like 10, the average accuracy is 67.56%, which is pretty low.

The accuracy is low, which can be the result of an arbitrary choice of k value.

```
mean(knn1 == Test$Life.expectancy.cat1)
```

```
## [1] 0.6666667
```

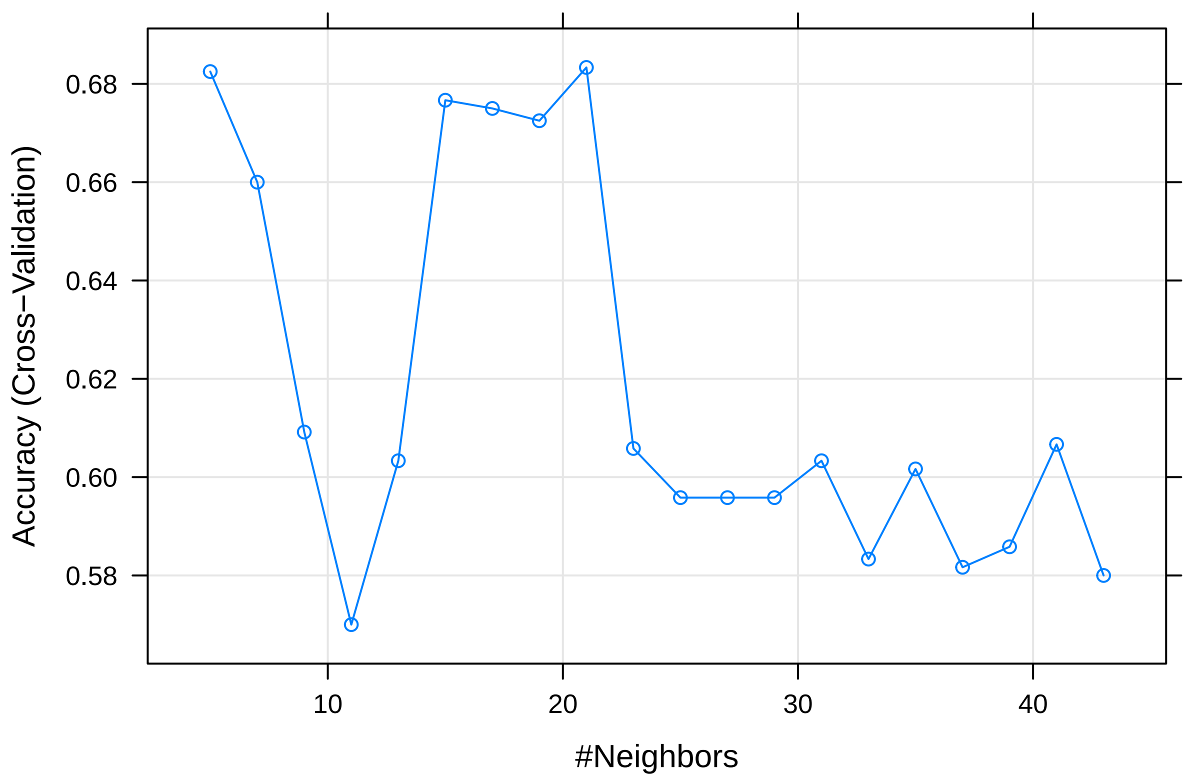
Next, we are going to find the optimal k value and construct a model based on that.

```
set.seed(2)
ctrl <- trainControl(method = "cv", number = 20)
set.seed(2)
knn2 <- train(Life.expectancy.cat1~Income.composition.of.resources+Adult.Mortality+HIV.AIDS, data = Tra
trControl = ctrl, tuneLength = 20)
set.seed(2)
knn2
```

```
## k-Nearest Neighbors
##
## 93 samples
##  3 predictor
##  4 classes: 'Very low', 'Low', 'Medium', 'High'
##
## No pre-processing
## Resampling: Cross-Validated (20 fold)
## Summary of sample sizes: 87, 90, 87, 89, 88, 89, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.6825000  0.5559489
##    7  0.6600000  0.5268145
##    9  0.6091667  0.4488633
##   11  0.5700000  0.3954281
##   13  0.6033333  0.4440968
##   15  0.6766667  0.5465836
##   17  0.6750000  0.5458665
##   19  0.6725000  0.5430887
##   21  0.6833333  0.5598108
##   23  0.6058333  0.4469338
##   25  0.5958333  0.4352379
##   27  0.5958333  0.4325826
##   29  0.5958333  0.4279387
##   31  0.6033333  0.4406838
```

```
##    33  0.5833333  0.4159160
##    35  0.6016667  0.4388161
##    37  0.5816667  0.4140483
##    39  0.5858333  0.4224842
##    41  0.6066667  0.4493047
##    43  0.5800000  0.4083728
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 21.
```

```
plot(knn2)
```



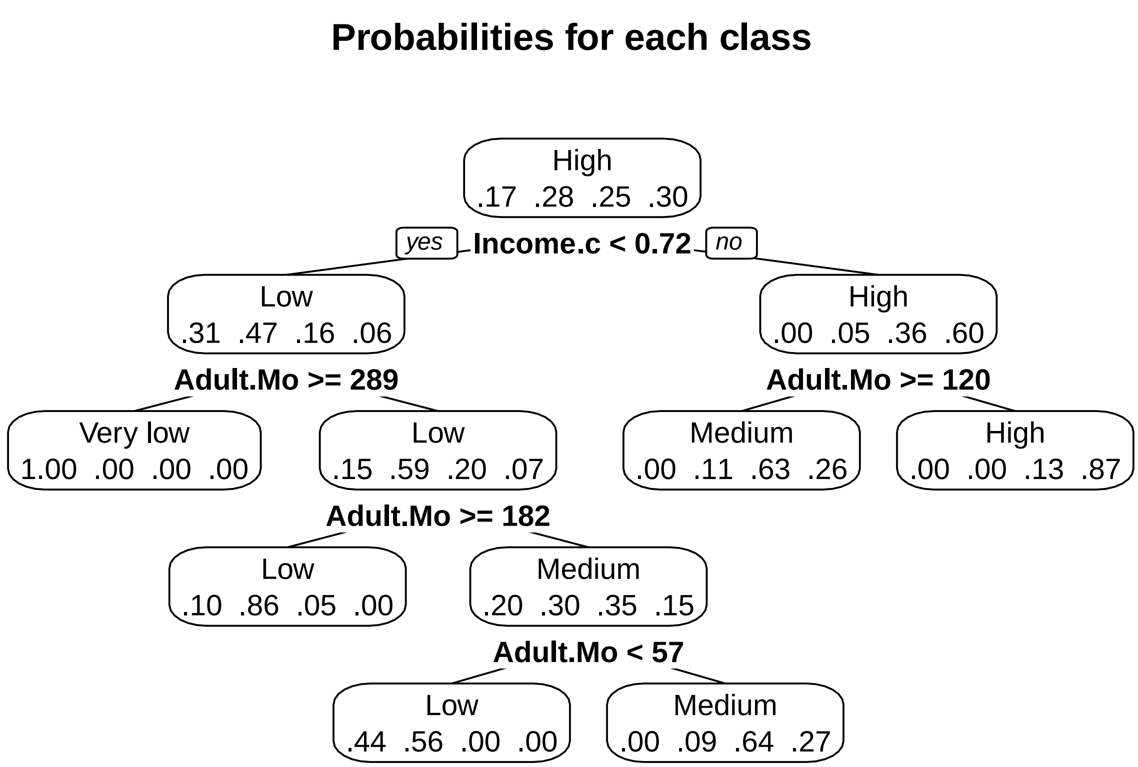
##k = 7 provides the highest accuracy taking into account the model with Income Composition of Resources, Adult Mortality and HIV Aids as independent variables. The accuracy is 69.33%, lower than the prediction with linear regression.

Decision Tree

Decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequesnces.

Below is the decision tree model with our three main independent variables and the decision tree showing probabilities for each case.

```
model_DT <- rpart(Life.expectancy.cat1~Income.composition.of.resources+Adult.Mortality+HIV.AIDS, data =
set.seed(2)
prp(model_DT, type = 2, extra = 4, main = "Probabilities for each class")
```



##Hereby we take the right hand decision path and explain it. ##If income composition of resources is more than 0.72, the adult mortality is lower than 137, income composition of resources is more than or equal to 0.75, then with 100% probability the life expectancy is high.

```
pred_class_DT <- predict(model_DT, Test, type = "class")
#printing out some predictions. Class levels should be very low, low, medium, high.
pred_class_DT[1:10]
```

```
##      1      9     12     19     30     33     35     40     49     50
##    Low Medium   High Medium Medium   High    Low    Low    Low    Low
## Levels: Very low Low Medium High
```

```r
#Confusion matrix for Decision tree
confusionMatrix(pred_class_DT, Test$Life.expectancy.cat1, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Very low Low Medium High
##   Very low        3   0      0    0
##   Low             3   7      0    0
##   Medium          0   3      8    2
##   High            0   0      1    9
##
## Overall Statistics
##
##                Accuracy : 0.75
##                  95% CI : (0.578, 0.8788)
##     No Information Rate : 0.3056
##     P-Value [Acc > NIR] : 5.136e-08
##
##                   Kappa : 0.6593
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Very low Class: Low Class: Medium Class: High
## Sensitivity                  0.50000     0.7000        0.8889      0.8182
## Specificity                  1.00000     0.8846        0.8148      0.9600
## Pos Pred Value               1.00000     0.7000        0.6154      0.9000
## Neg Pred Value               0.90909     0.8846        0.9565      0.9231
## Prevalence                   0.16667     0.2778        0.2500      0.3056
## Detection Rate               0.08333     0.1944        0.2222      0.2500
## Detection Prevalence         0.08333     0.2778        0.3611      0.2778
## Balanced Accuracy            0.75000     0.7923        0.8519      0.8891
```

The accuracy of the decision tree is 69.44%, lower than the accuracy of linear regression model, Naive Bayes and KNN. It is higher than NIF (30.56%).

Logistic Regression

We have successfully explained and measured life expectancy in 120 countries, and now we measure the status of these countries based on life expectancy. We use logistic regression to measure status which is binary variable which means the countries belong to two groups, either developed or developing.

We devided the dataset into again train and test sets, ensuring equal distribution in both of them.

Status is converted from character to factor variable.

```
set.seed(2)
index_logistic <- createDataPartition(life_expect$Status, p = 0.7, list = FALSE)
Train_logistic <- life_expect[index,]
Test_logistic <- life_expect[-index,]
```

The logistic regression model is created with all the variables and we can see that all of them is significant including Status variable, maybe because of overplotting.

```
life_expect$Status <- as.factor(life_expect$Status)
Train_logistic$Status <- as.factor(Train_logistic$Status)
Test_logistic$Status <- as.factor(Test_logistic$Status)
Train_logistic$Status<- factor(Train_logistic$Status,levels = c("Developing", "Developed"), labels = c(
Train_logistic$Status
```

```
##  [1] 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0
## [39] 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0
## [77] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```
model_LR <- glm(Status~.-Country, data = Train, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model_LR)
```

```
##
## Call:
## glm(formula = Status ~ . - Country, family = "binomial", data = Train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
```

```
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error   z value Pr(>|z|)
## (Intercept)                     1.286e+16  2.173e+08  59176008   <2e-16 ***
## Year                                   NA         NA        NA       NA
## Life.expectancy                -1.660e+14  3.774e+06 -43969131   <2e-16 ***
## Adult.Mortality                 2.729e+12  9.854e+04  27689747   <2e-16 ***
## infant.deaths                   1.996e+13  8.579e+05  23268251   <2e-16 ***
## Alcohol                        -8.834e+13  2.651e+06 -33326559   <2e-16 ***
## percentage.expenditure          5.708e+11  9.422e+03  60577320   <2e-16 ***
## Hepatitis.B                    -3.040e+12  4.574e+05  -6646819   <2e-16 ***
## Measles                        -1.619e+11  4.065e+03 -39829813   <2e-16 ***
## BMI                            -9.293e+12  4.953e+05 -18763543   <2e-16 ***
## under.five.deaths              -1.096e+13  6.426e+05 -17047500   <2e-16 ***
## Polio                          -2.334e+13  4.797e+05 -48651692   <2e-16 ***
## Total.expenditure               1.805e+14  3.459e+06  52188732   <2e-16 ***
## Diphtheria                      8.867e+12  7.030e+05  12613507   <2e-16 ***
## HIV.AIDS                        1.155e+14  5.334e+06  21650514   <2e-16 ***
## GDP                            -8.454e+10  1.647e+03 -51329907   <2e-16 ***
## Population                      3.635e+06  3.418e-01  10633313   <2e-16 ***
## thinness..1.19.years            1.420e+14  8.719e+06  16288035   <2e-16 ***
## thinness.5.9.years             -1.315e+14  8.353e+06 -15746775   <2e-16 ***
## Income.composition.of.resources -1.824e+15  2.108e+08  -8650444   <2e-16 ***
## Schooling                      -9.714e+13  7.864e+06 -12352572   <2e-16 ***
## Life.expectancy.cat1Low         3.871e+15  4.144e+07  93414118   <2e-16 ***
## Life.expectancy.cat1Medium      4.439e+15  6.087e+07  72926261   <2e-16 ***
## Life.expectancy.cat1High        5.541e+15  7.224e+07  76698706   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  63.484  on 92  degrees of freedom
## Residual deviance: 576.698  on 70  degrees of freedom
## AIC: 622.7
##
## Number of Fisher Scoring iterations: 13
```

We create a model taking only Life Expectancy as dependent variable.

```
model_LR_st <- glm(Status~Life.expectancy, data = Train_logistic, family = "binomial")
summary(model_LR_st)
```

```
##
## Call:
## glm(formula = Status ~ Life.expectancy, family = "binomial",
##     data = Train_logistic)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.39769  -0.45092  -0.18394  -0.04355   2.39389
##
```

```
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -24.77993    7.16271  -3.460 0.000541 ***
## Life.expectancy   0.30100    0.09195   3.273 0.001063 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 63.484  on 92  degrees of freedom
## Residual deviance: 42.822  on 91  degrees of freedom
## AIC: 46.822
##
## Number of Fisher Scoring iterations: 7
```

```
#predicting on Testing set
Test_logistic$Status
```

```
##  [1] Developing Developing Developed  Developed  Developing Developed
##  [7] Developing Developing Developing Developing Developing Developed
## [13] Developed  Developing Developing Developing Developed  Developing
## [19] Developing Developing Developing Developing Developed  Developing
## [25] Developing Developing Developing Developed  Developing Developed
## [31] Developing Developing Developing Developing Developing Developing
## Levels: Developed Developing
```

```
predict_LR_st <- predict(model_LR_st, newdata = Test_logistic, type = "response")
predict_LR_st[1:10]
```

```
##            1            9           12           19           30           33
## 0.0010371405 0.0415741502 0.5506553945 0.0733857404 0.0820053902 0.1718463010
##           35           40           49           50
## 0.0023346819 0.0032480764 0.0007450206 0.0005856773
```

Now we want to know the accuracy of prediction with only Life Expectancy variable. That is why we create a confusion matrix.

```
#confusion matrix for the model
pr_class_LR_st <- factor(ifelse(predict_LR_st > 0.5, "Developed", "Developing"))
table(Test_logistic$Status, pr_class_LR_st)
```

```
##             pr_class_LR_st
##              Developed Developing
##   Developed          3          6
##   Developing         0         27
```

```
confusionMatrix(pr_class_LR_st, Test_logistic$Status, positive = "Developed")
```

```
## Confusion Matrix and Statistics
```

```
##
##              Reference
## Prediction   Developed Developing
##   Developed          3          0
##   Developing         6         27
##
##                Accuracy : 0.8333
##                  95% CI : (0.6719, 0.9363)
##     No Information Rate : 0.75
##     P-Value [Acc > NIR] : 0.16839
##
##                   Kappa : 0.4286
##
##  Mcnemar's Test P-Value : 0.04123
##
##             Sensitivity : 0.33333
##             Specificity : 1.00000
##          Pos Pred Value : 1.00000
##          Neg Pred Value : 0.81818
##              Prevalence : 0.25000
##          Detection Rate : 0.08333
##    Detection Prevalence : 0.08333
##       Balanced Accuracy : 0.66667
##
##        'Positive' Class : Developed
##
```

```r
unique(pr_class_LR_st)
```

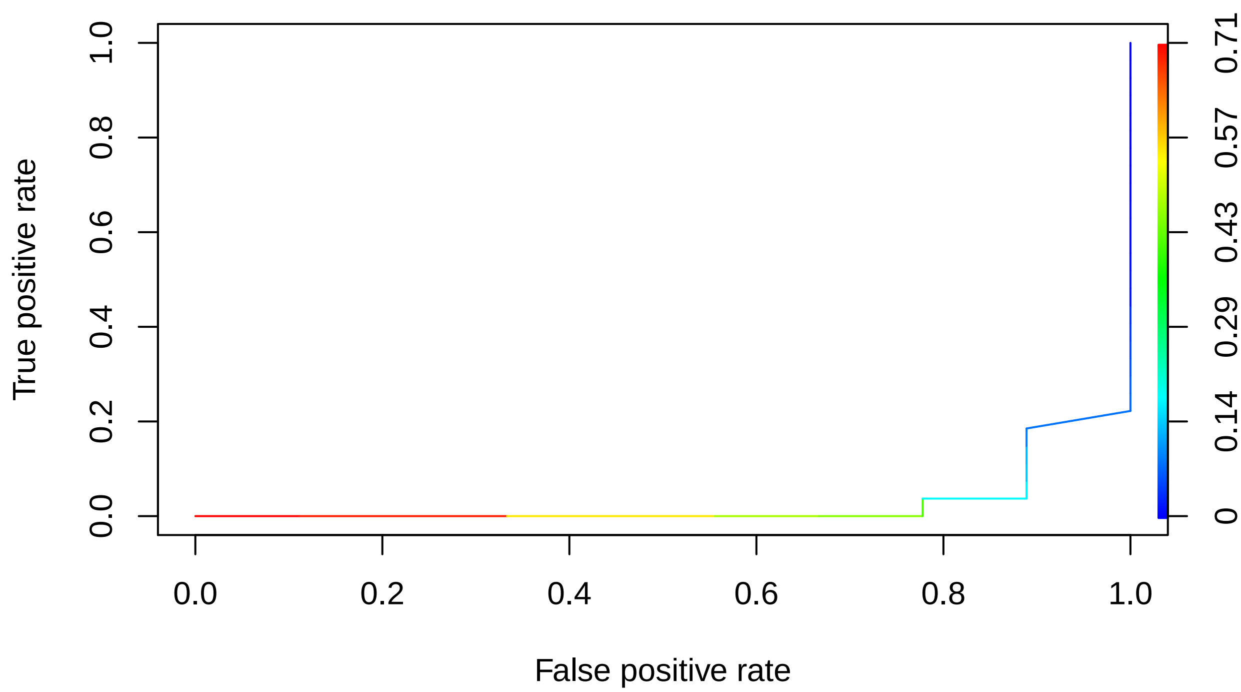
```
## [1] Developing Developed
## Levels: Developed Developing
```

```r
unique(Test_logistic$Status)
```

```
## [1] Developing Developed
## Levels: Developed Developing
```

The accuracy is 94.4%.

```r
P_Test_LR_st <- prediction(predict_LR_st, Test_logistic$Status)
perf_LR <- performance(P_Test_LR_st, "tpr", "fpr")
#coloring with treshhold values
plot(perf_LR, colorize = T)
```



##The area under the curve is below 1 almost zero, so with different threshold values. The accuracy is low.

```
#performance of the model
performance(P_Test_LR_st, "auc")@y.values
```

```
## [[1]]
## [1] 0.02674897
```

Random Forest

```
model_RF <- randomForest(Life.expectancy.cat1~., data = Train[,-1], importance = T)
model_RF
```

```
##
## Call:
##  randomForest(formula = Life.expectancy.cat1 ~ ., data = Train[,      -1], importance = T)
##               Type of random forest: classification
##                     Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 5.38%
## Confusion matrix:
##          Very low Low Medium High class.error
## Very low       15   1      0    0  0.06250000
```

```
## Low              0  26      0    0  0.00000000
## Medium           0   1     22    0  0.04347826
## High             0   1      2   25  0.10714286
```

```r
model2_RF <- randomForest(Life.expectancy.cat1~., data = Train[,-1], ntree = 500, mtry = 6, importance =
model2_RF
```

```
##
## Call:
##  randomForest(formula = Life.expectancy.cat1 ~ ., data = Train[,      -1], ntree = 500, mtry = 6, imp
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 3.23%
## Confusion matrix:
##          Very low Low Medium High class.error
## Very low       16   0      0    0  0.00000000
## Low             0  26      0    0  0.00000000
## Medium          0   1     22    0  0.04347826
## High            0   0      2   26  0.07142857
```

```r
pred_RF <- predict(model2_RF, Train, type = "class")

table(pred_RF, Train$Life.expectancy.cat1)
```

```
##
## pred_RF    Very low Low Medium High
##   Very low       16   0      0    0
##   Low             0  26      0    0
##   Medium          0   0     23    0
##   High            0   0      0   28
```

```r
pred_RF_test <- predict(model2_RF, Test, type = "class")

mean(pred_RF_test == Test$Life.expectancy.cat1)
```

```
## [1] 1
```

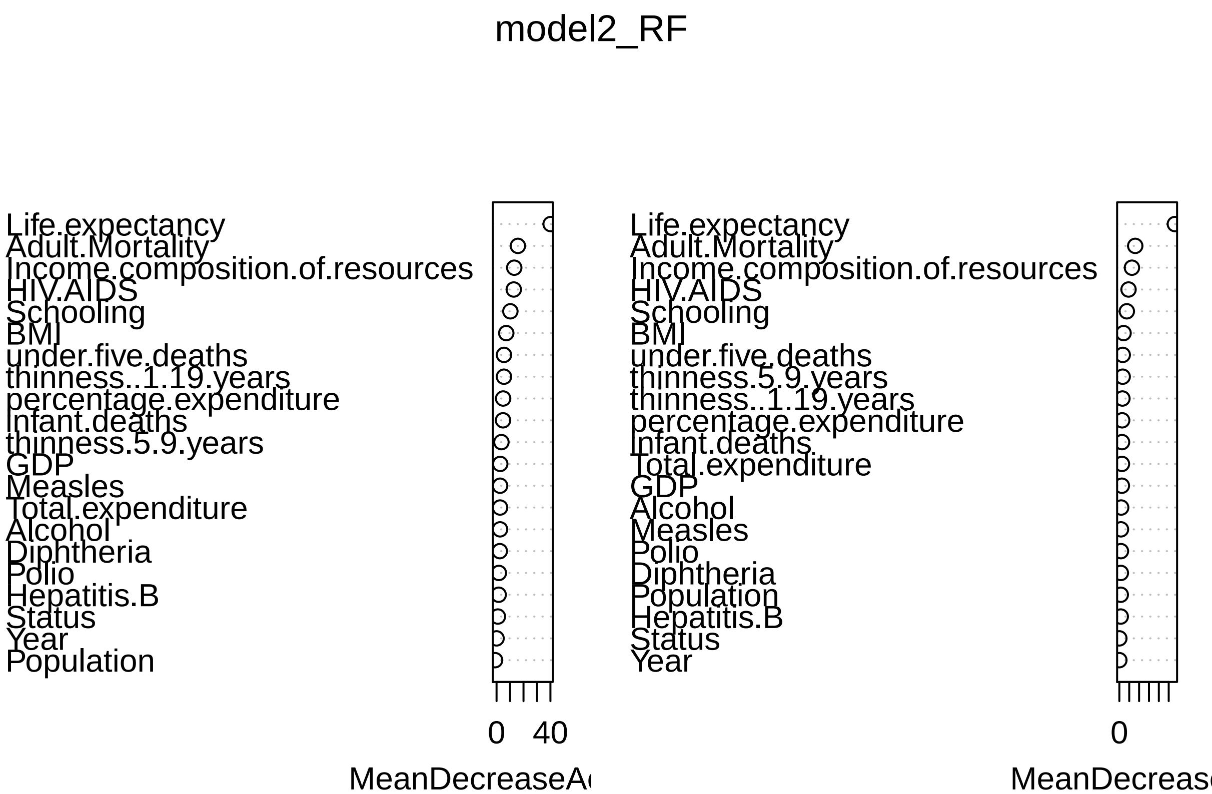
```r
table(pred_RF_test, Test$Life.expectancy.cat1)
```

```
##
## pred_RF_test Very low Low Medium High
##     Very low        6   0      0    0
##     Low             0  10      0    0
##     Medium          0   0      9    0
##     High            0   0      0   11
```

```r
importance(model2_RF)
```

```
##                                  Very low        Low      Medium        High
## Year                             0.0000000  0.0000000  0.00000000  0.00000000
## Status                           1.0010015  1.0010015 -1.00100150  0.00000000
## Life.expectancy                 25.8167921 25.9031076 26.30069886 33.01191116
## Adult.Mortality                  7.7499282  9.2641685  8.41646462 12.44362030
## infant.deaths                    6.4206997 -1.5234982 -1.20999129  2.37132503
## Alcohol                         -0.4793075  2.2616290  0.72090050  3.20152208
## percentage.expenditure           1.8358851  4.4207168  2.34921054  0.96435707
## Hepatitis.B                      0.2349945  1.1939800  0.78429056  0.51625497
## Measles                          3.4786232 -0.4090613  2.27642758 -2.50074132
## BMI                              6.9549439  3.2973230  1.55012370  1.87309157
## under.five.deaths                6.2103457 -1.6609913  0.78054170  3.64116262
## Polio                            2.5960110 -0.5879972  1.41718685  0.68217114
## Total.expenditure                0.1962284  3.2031715 -2.84591345  3.23177535
## Diphtheria                      -0.1315137  1.0362947  2.35313200  0.03072722
## HIV.AIDS                        13.7313457 -0.6633399  6.30937005  6.06075291
## GDP                              0.1046479  0.5335183  2.60418206  1.78728115
## Population                      -2.2051659  1.7697858 -0.98863592 -1.85193013
## thinness..1.19.years             3.0027579  0.9842016  0.88284805  4.81573020
## thinness.5.9.years               3.7181095 -0.5189761  0.01779144  4.39572115
## Income.composition.of.resources 12.5371674  6.2540322  3.71177420  9.13143306
## Schooling                        6.7929474  6.0234635  4.46559489  7.67578637
##                                 MeanDecreaseAccuracy MeanDecreaseGini
## Year                                        0.000000      0.000000000
## Status                                      1.001002      0.007620558
## Life.expectancy                            40.123967     28.111485706
## Adult.Mortality                            15.829442      8.038368568
## infant.deaths                               4.799190      1.354153477
## Alcohol                                     2.589555      0.922999465
## percentage.expenditure                      4.882397      1.429185504
## Hepatitis.B                                 1.567539      0.707356016
## Measles                                     2.625790      0.808790792
## BMI                                         7.208748      2.140042781
## under.five.deaths                           5.498750      1.714388728
## Polio                                       1.713904      0.798229934
## Total.expenditure                           2.596643      1.310408137
## Diphtheria                                  2.392485      0.767508464
## HIV.AIDS                                   12.770278      4.656517493
## GDP                                         2.739407      1.278826222
## Population                                 -1.095093      0.725421943
## thinness..1.19.years                        5.497186      1.556492480
## thinness.5.9.years                          3.607192      1.692406316
## Income.composition.of.resources            13.127769      6.370564333
## Schooling                                  10.248798      3.781405128
```

```r
varImpPlot(model2_RF)
```



##Conclusion:

The life expectancy is mostly dependent on HIV aids,Income contribution of resources and Adult Mortality

The best model was created with Random Forest with an accuracy of 97.3%.

Life Expectancy was analysed with countries Status. The low accuracy of 16% showed that the country???s development status is only 16% varied based onn variance of Life Expectancy. Other factors may have higher impact on the country???s status.