Life Expectancy Statistical Analysis

John Girard, Alexander Lopez, Duy Nguyen
Southern Methodist University

1 Introduction

What are some of the factors that influence life expectancy? Our first project for the Applied Statistics course in the Southern Methodist University Master of Science of Data Science program was to predict the life expectancy in countries according to the Global Health Observatory (GHO) data repository under World Health Organization (WHO). Within this dataset, we are tasked with exploratory data analysis, building and validating the competing regression models using what we have learned in the first two units of the course as well as applying what we already knew towards a non-parametric model like k-NN regression or regression tree.

2 Life Expectancy Data Set

The data set used in this project describes the life expectancy, health factors for 193 countries from 2000 to 2015. It was retrieved from data set hosting website Kaggle, where it is listed as a machine learning competition called *Life Expectancy (WHO): Statistical Analysis on factors influencing Life Expectancy*[1] . The data "consists of 22 columns and 2938 rows which means 20 predictor variables. All predicting variables were then divided into several broad categories: Immunization related factors, Mortality factors, Economical factors and Social factors."

3 Exploratory Data Analysis

3.1 Missing Data

We first looked at the missing values of the dataset, which represent about 4% of the dataset. Seven predictor variables are found to have missing values that represent more than 5% of themselves. Through visual analysis, we imputed the missing values using the median of the said columns that those missing values reside in [*see Section 7.1 & 7.2*].

3.2 Zero Variance

Zero variance is a common issue for any dataset. This issue depicts a level of column representing the near majority or the entirety of that column, which is proved to not be useful for analyses. Fortunately we did not come across any variables with this issue to exclude.

3.3 Variable Format

Character variables are then converted into factor variables for the sake of easier model building. *Country* seemed to be not useful for our analysis since it has too many levels and was removed. However, *Status* will be useful since it only has 2 levels, Developed and Developing, which will statistically influence life expectancy. We also decided to remove *Year* based on the instructions of our professor, reasons being that what we're doing during this level of the Data Science program does not respond kindly with autocorrelation issues when working with time related variables.

3.4 Significant Predictors

Looking at the correlation plot [*see Section 7.3*], predictors *infant.deaths* and *under.five.deaths* had a correlation value of 1 and were removed one at a time from the dataset. The resulting datasets were used to build initial models using all features of the dataset, with the exception of the aforementioned modifications. The dataset with *infant.deaths* removed produced a higher R^2 value and so was used for the remainder of the analysis.

Predictors that contributed positively to life expectancy (cor > 0.4) were *BMI, Polio, Diphtheria, GDP, Income.composition.of.resources* and *Schooling.* The predictors that contributed negatively to life expectancy (cor < -0.4) were *Adult.Mortality, HIV.AIDs, thinness.5.9.years* and *thinness..1.19.years*. We also looked at the bivariate analysis on *Life.expectancy* versus other features of the dataset by *Status* and *GDP* [*see Section 7.4*]. We will later come back to these specific predictors as they will become very significant to our analysis.

4 Addressing Objective 1

4.1 Building The First Model

The first objective of the project includes displaying the ability to build a final regression model complete with hypotheses, coefficients and confidence intervals. We used feature selection linear regression, the Durbin-Watson test, the Goldfeld-Quandt test, and a custom predict function to test the validity of our first model. With the key relationships between the features of the dataset identified and interpreted in the previous sections, we used statistics values and visualization to further aid our model building process in both senses of practical and statistical significance. We have also adhered to recommendations from our professor and divided the dataset into both a train set and a test set as recommended by our professor, which represents 85% and 15% of the entire dataset, respectively.

4.2 Feature Selection

A custom predict function was provided to us by our professor, which we used to find the number of predictors that produces the lowest test ASE. We were able to plot the number of predictors over the ASE of the test dataset, finding out the number of predictors to be nine [*see Section 7.5*]. The appropriate predictors are found to be *Adult.Mortality, BMI, Polio, Diphtheria, HIV.AIDS, GDP,*

Income.composition.of.resources, Schooling and *Status*. As expected from the last parts of our EDA, these nine predictors have already come up in our list of significant predictors as shown in the full correlation plot [*see Section 7.3*].

To further confirm the validity of our model, the true response from the train dataset was plotted against the predicted values [*see Section 7.6*]. The points of this plot follow very closely to the red reference line, indicating that our first model did very well, however there are some points that stray away from the line that will need attention. The assumptions of the first model will be addressed in the upcoming sections as well as remedies to the dataset in hopes of handling the spread of points.

First Model

$$Life.expectancy = \hat{\beta}_0 + \hat{\beta}_1 Status + \hat{\beta}_2 Adult.Mortality + \hat{\beta}_3 BMI + \hat{\beta}_4 Polio + \hat{\beta}_5 Diphtheria + \hat{\beta}_6 HIV.AIDS + \hat{\beta}_7 GDP + \hat{\beta}_8 Income.composition.of.resources + \hat{\beta}_9 Schooling$$

4.3 Assumptions

Linearity

There is some evidence against linearity observed from the scatterplots [*see Section 7.7*]. Against *Life.expectancy*, *Adult.Mortality* is observed to have a quadratic relationship, and the other 8 predictors is observed to have cubic relatinships. These will be addressed by using log-transformation to resolve the linearity issue then applying them as polynomials in the next section.

Normality

There is enough evidence to suggest that the residuals are normally distributed when looking at the histogram [*see Section 7.8*], however the residuals appear less normal when looking at the Q-Q plot. This can be fixed with a log-transformation.

Multicollinearity

The simplest way to address multicollinearity is to remove predictors with high VIF values, or rather:

$$VIF > \frac{1}{(1 - Adj.R^2)}$$

Since high VIF is determined as higher for our model is 5.053, all of our predictors have VIFs that are less than that and all of them will still be included in our model.

Homoscedasticity

Judging by the distribution of residuals on the residual versus fitted plot [*see Section 7.9*], we can see very strong evidence against homoscedasticity as observations fan out from left to right. Additionally, there is enough evidence to suggest that heteroscedasticity is present in our model (p-value = 0.01147 from a Goldfeld-Quandt test). A log-transformation usually fixes this issue.

Influential Point

Based on the Cook's distance plot, none of the observations have Cook's D > 0.05 [*see Section 7.10*]. And looking at the residuals versus leverage plot, we cannot see any observations that are out of place, therefore no removal of outliers is needed for our model.

4.4 Interpretation

Complete First Model

$$Life.expectancy = 52.736 - 2.425\ Status - .00222\ Adult.Mortality\ + 4.662\ BMI$$
$$+0.0203\ Polio + 0.032\ Diphtheria - 0.05\ HIV.AIDS$$
$$+0.0000291 GDP + 5.435\ Income.composition.of.resources + 0.0604\ Schooling$$

Interpretations

Using the first linear regression model we developed, we are 95% confident that, holding all other predictors constant, a country's predicted life expectancy:

- Decreases by 18.97 years, between 13.69 and 24.25, when it is a Developing country
- Decreases by 0.20 years, between 0.19 and 0.22, when it's Adult Mortality increases by 10
- Decreases by 4.62 years, between 4.24 and 5.00, when it's HIV/AIDS increases by 10
- Increases by 0.42 years, between 0.32 and 0.51, when it's Diphtheria increases by 10
- Increases by 6.93 years, between 6.04 and 7.83, when it's Schooling increases by 10
- Increases by 0.57 years, between 0.47 and 0.64, when it's BMI increases by 10
- Increases by 0.30 years, between 0.20 and 0.40, when it's Polio increases by 10
- Increases by 0.04 years, between 0.03 and 0.06, when it's GDP increases by 1000
- Increases by 6.83 years, between 5.44 and 8.22, when it's Income Composition by Resources increases by 1

Our model is a good fit for the data (p-value < 2.2E-16 for F-test with df(9, 2487)) [*see Section 7.11*]. It is estimated that the predictor variables of our model explains about 80.28% of the variation in life expectancy years of all the countries from a period of 2000 to 2015.

Since this is an observational study conducted by the World Health Organization, we cannot make any causal inferences. However, we can imply for all 193 countries during 2000 and 2015, the appropriate changes to our 9 predictor variables do relate to changes to life expectancy.

Practical Significance

Diphtheria plotted against life expectancy *[see section 7.9]* proves to be an interesting example when observed further. On first glance it is easy to assume that this disease is not very severe as the life expectancy appears to be rather high. Although when looking at the lower portion of the graph we can observe that the disease is quite severe in young children and almost exclusively in developing countries. An article from the Mayoclinic[2] states that contracting the disease is not only treatable but it is preventable with a vaccine. Once diagnosed with this disease, the infected will receive the vaccine to combat the illness. This is why we see a large trend of people being largely unaffected by this disease, as once they contract the disease, they will have prevention from it in their later ages. Unfortunately

developing countries might not have access to these vaccines which is why we see a large grouping of lower life expectancy in developing countries under the diphtheria predictor.

Schooling plotted against life expectancy *[see section 7.9]* is another predictor our group wanted to look at further. A study conducted [3] showed that education closely predicted life expectancy. The study deduced that opportunities to further your education increases your opportunity to attain wealth and being able to procure a healthier, less stressful lifestyle. Our graph shows that the relationship between schooling and life expectancy is completely positively related and shows that developed countries have people with a much higher life expectancy than those in developing countries without access to schooling.

5 Addressing Objective 2

5.1 Building The Second Model

The second objective of the project involves comparing multiple models to develop the best predicting model for our World Health Organization data-set. Comparisons were based on the testing metrics ASE and R^2. Complexity is added to our model for it to better predict our test dataset, or better yet predict future data, at the expense of interpretations. Using visual analysis, we were able to build a complex linear regression model based on the first model with polynomials to the second and third degree as well as log-transformation.

5.2 Complex Regression Model

We deduced that since both *HIV.AIDS* and *GDP* appeared less linear on a plot versus *Life.expectancy* compared to others, all 3 of those needed a log-transformation. The rest of the variables did not need the same treatment since they appeared linear enough. The changes made are reflected in the log-transformed plot *[see section 7.12].* We also used visualization aid to determine our second model, where quadratic observations on variables are treated as second degree in our model, and cubic observations are treated as third degree. This second model performed much better than the first one, achieving a test ASE of 12.242 and an R-Squared of 0.868. We also plotted the predicted values against the actual values *[see section 7.12]* and the observations appeared much closer to the red reference line and are more spread out along the line, rather than clumped up in one area like the plot of the first model.

5.3 Non-parametric Model

The second part of this objective is to perform a k-nearest neighbors' regression model using both the train and test sets to find the most appropriate K in KNN. Unlike our first 2 models, both of which were parametric, "non-parametric models do not explicitly assume a parametric form, and thereby provide an alternative and more flexible approach for performing regression"[4].

The KNN regression model is very similar to the KNN classification model. KNN regression models first work to identify the K training observations that are closest to the prediction point. The model then estimates a function by using the average of all the training responses that are close to the prediction points, see below.

$$f(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i.$$

Like with the parametric models, the KNN model optimization will depend on the bias versus variance tradeoff. KNN models will have more overfitting to the training data as k decreases so although a model with K = 1 may provide impressive result statistics, running that same model on a new test data set will likely yield poor metrics. The overfitting occurs when the model uses a small number of neighbors to make predictions. In the example above, only using one nearest neighbor (K = 1) would result in a rough step-like model that is fit almost precisely to the training data, while a model with more neighbors (K > 1) would yield a smoother fit.

The third and final model was a KNN Regression model. The model was given predictors and used various K values for comparison purposes. This proved to be our strongest model developed. After determining the best performing K value using cross validation with the entire data set *[see section 7.14]*, K = 3, our group was able to achieve a test ASE of 0.024 and a R^2 of 0.999. These scores indicate this model to be the best performing model, with the highest test metrics.

5.4 Comparing Models

Model	Test ASE	R^2
1st Feature Selection	16.9334	0.803
2nd Complex Regression	12.242	0.868
3rd KNN Regression	0.024	0.999

Between the three models produced, our group achieved satisfactory results. Logging the HIV.AIDS, GDP and Life.expectancy predictors after the first model was ran proved to be a good idea as our test ASE ended up being lower. Our predicted values were tighter around the reference line, so this was the correct decision tree to follow. Our final model using nearest neighbors was the strongest of all 3 and has the strongest predicted vs actual correlation *[see section 7.15]*. We are able to say that 3 was the most accurate model by making sure to play within the rules of KNN to ensure that the test was robust.

6 Conclusion

6.1 Recap of Objectives 1 and 2

The work done on objective 1 and 2 left us with a regression model able to predict the practical and statistical significance to help us accurately predict the strongest predictors to life expectancy. Our first model was a feature selection, we were able to achieve a test ASE of 16.9334.

Next, our group attempted to use a complex regression model to better predict future data. Our group first log transformed some predictors *[see section 7.12]*. Running this model left our group with an ASE of 12.242.

Finally, our group was tasked with using a non-parametric method to predict data. The catch here is that non-parametric models do not explicitly assume a parametric form, and thereby provide an alternative and more flexible approach for performing regression. Our group was able to come up with an ASE of .024.

In the end our strongest model to accurately predict data ended up being the KNN regression model (model 3).

6.2 Details and Comments

Because the subjects were not selected randomly from any population, extending this inference to any other group is speculative. This is an observational study generalized to the "participants" of this study and we can not correlate these results to people outside of the study or outside of the time frame this study was conducted in.

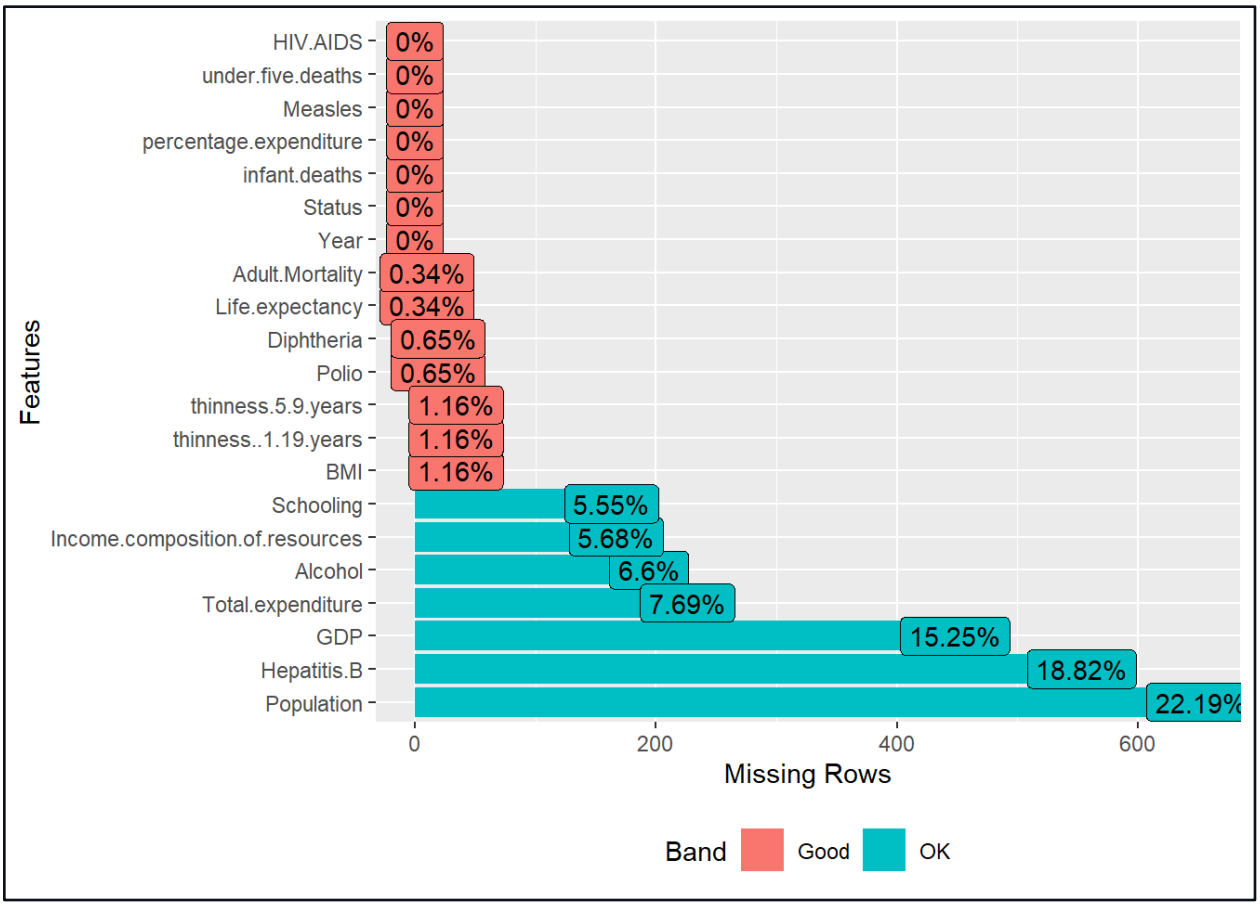
We could see this data being paired with a few other predictors to build a "Happiness Index". This could be calculated based on some of the ranges of the predictors. With this, we would like to log this data. The discrepancy between some of the nations' incomes greatly could influence the skew of said study.

We could also see this data being paired with covid collection methods to figure out which countries were hit hardest by covid and understanding what kind of predictors we can find to understand why they were hit so hard. Maybe a "Lockdown Strength" predictor could be included and address how seriously a country took their lockdown methods and see if it made an impact. This would be incredibly useful to understand in the case of another pandemic.
Lastly, another study that could be useful that this data has would be to understand the trajectory of a developing countries path to a "Developed" status. Taking countries that went from developing to developed would be focused on and we can see the predictors that they took to get there. For this, a time series analysis would be perfect.

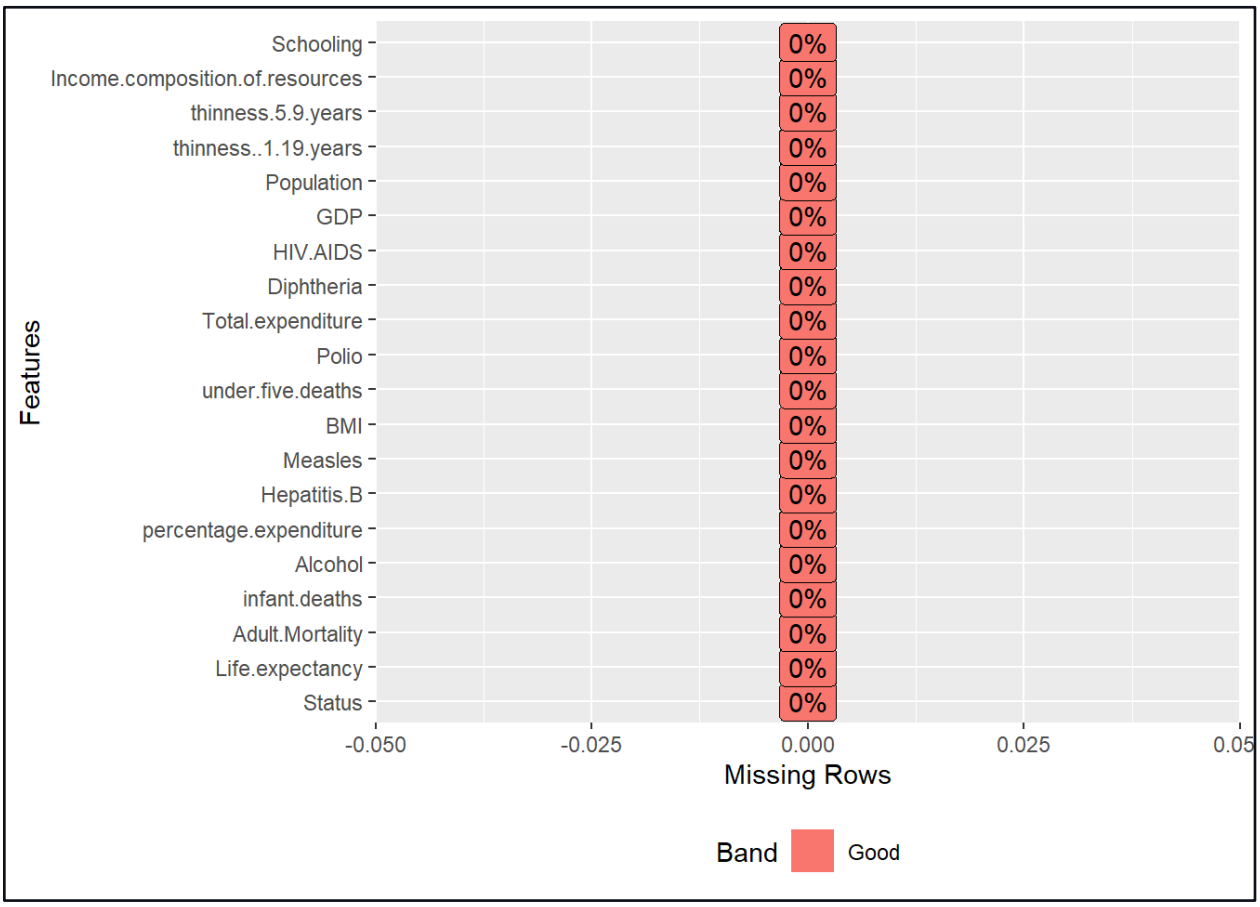
It is tough to conduct data analysis on this large of a scale. It is also very expensive to do something like this on this scale. The issue is not with the method of data collection itself, it would be perfect if everyone in the world was collected for this, but that is unlikely to happen.

7 Appendix

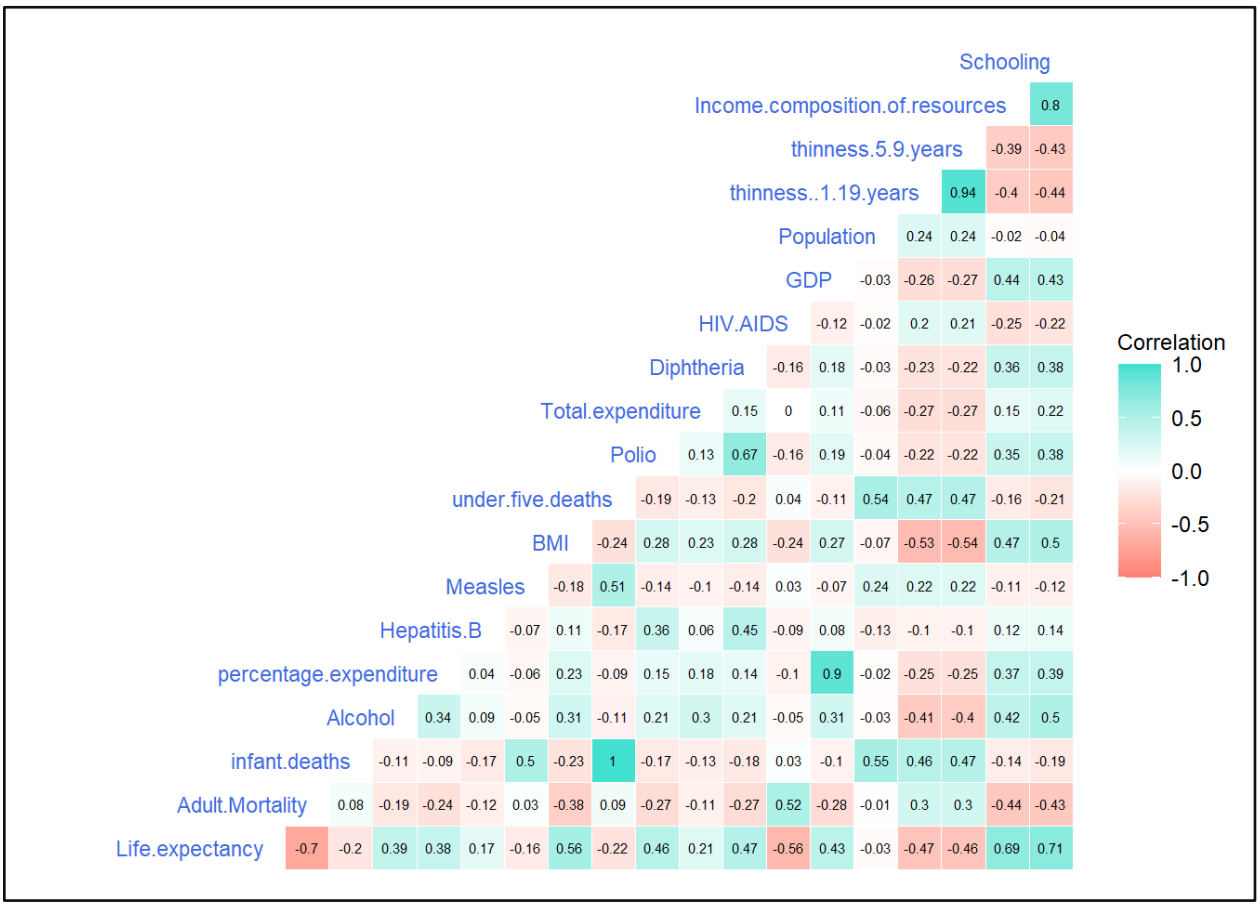
7.1 Missing Values Percentages



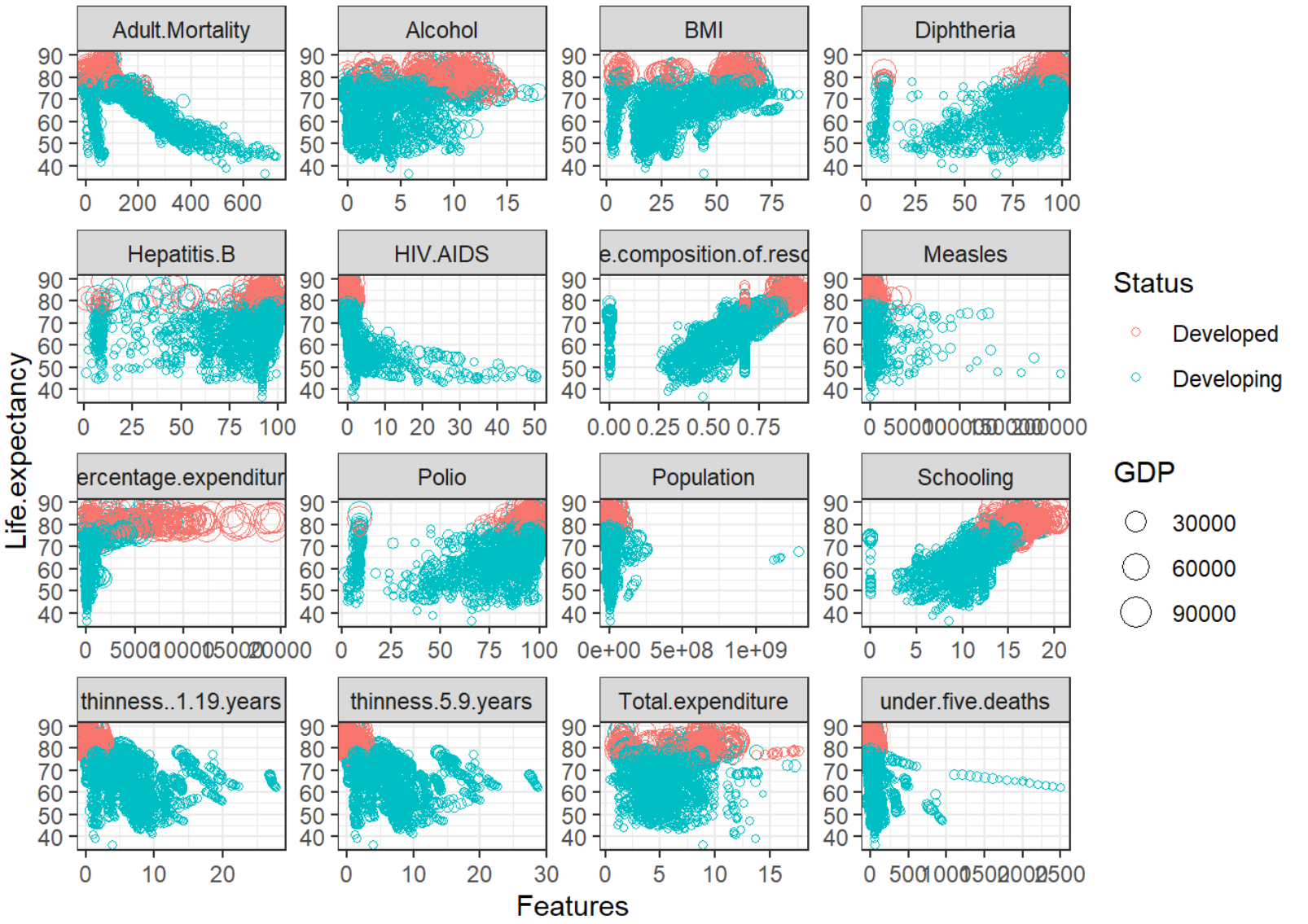
7.2 After Imputations



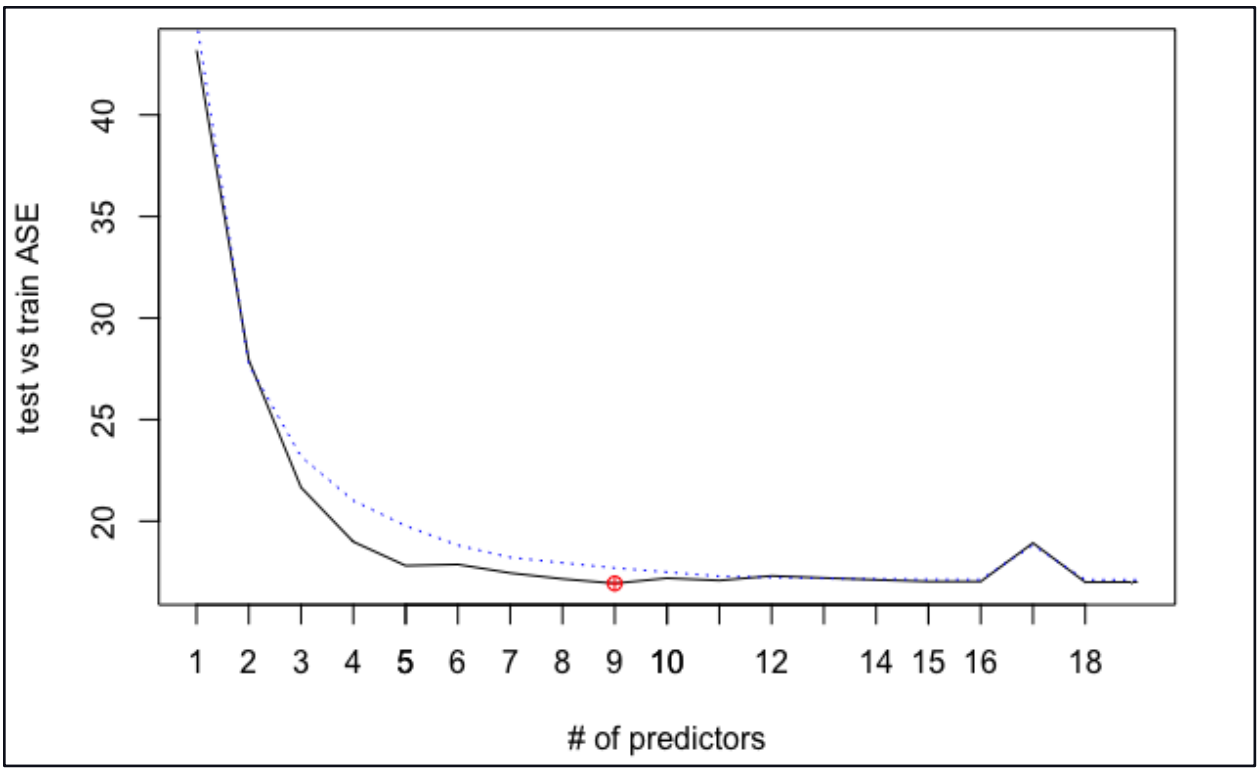
7.3 Correlations Plot



7.4 Bivariate Analysis

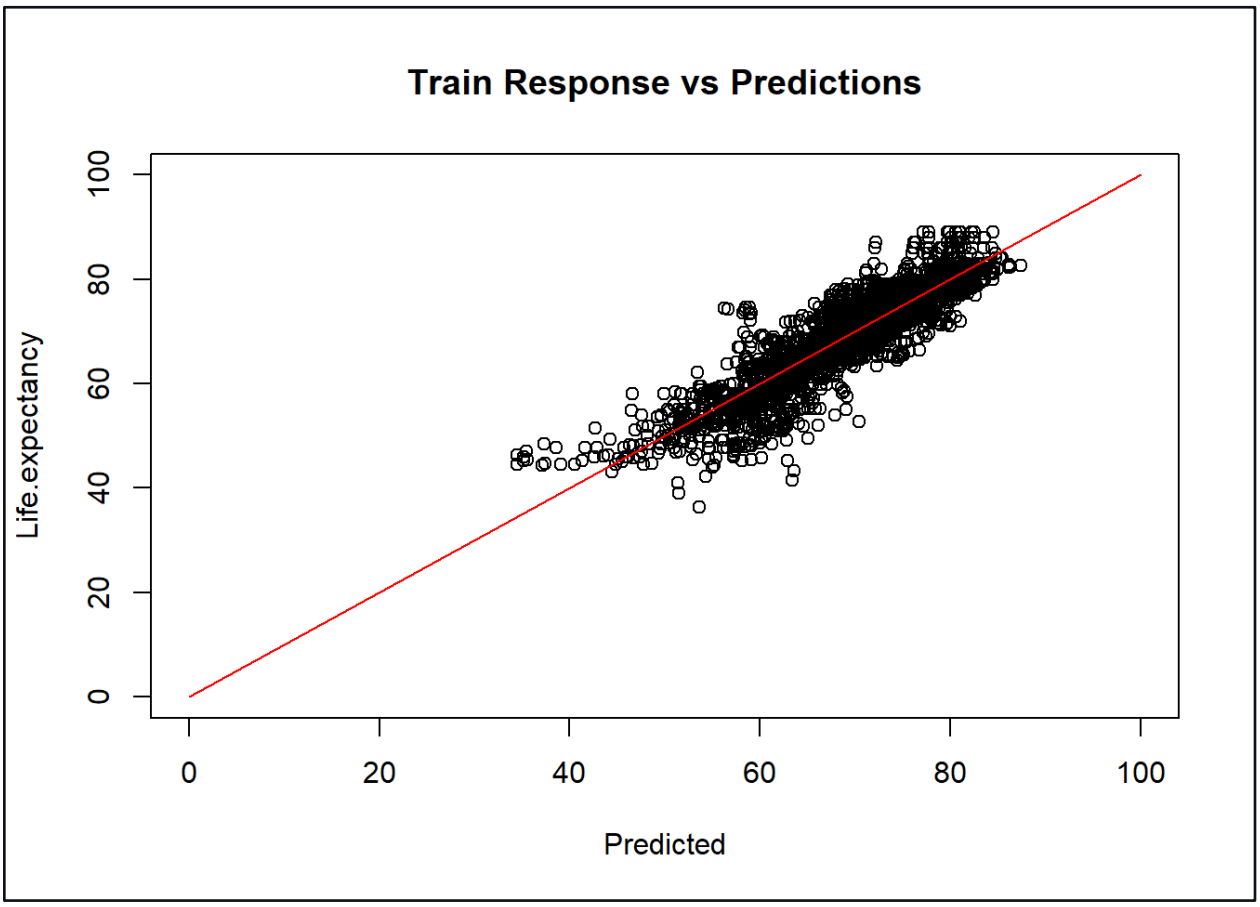


7.5 Most Efficient Number of Predictors

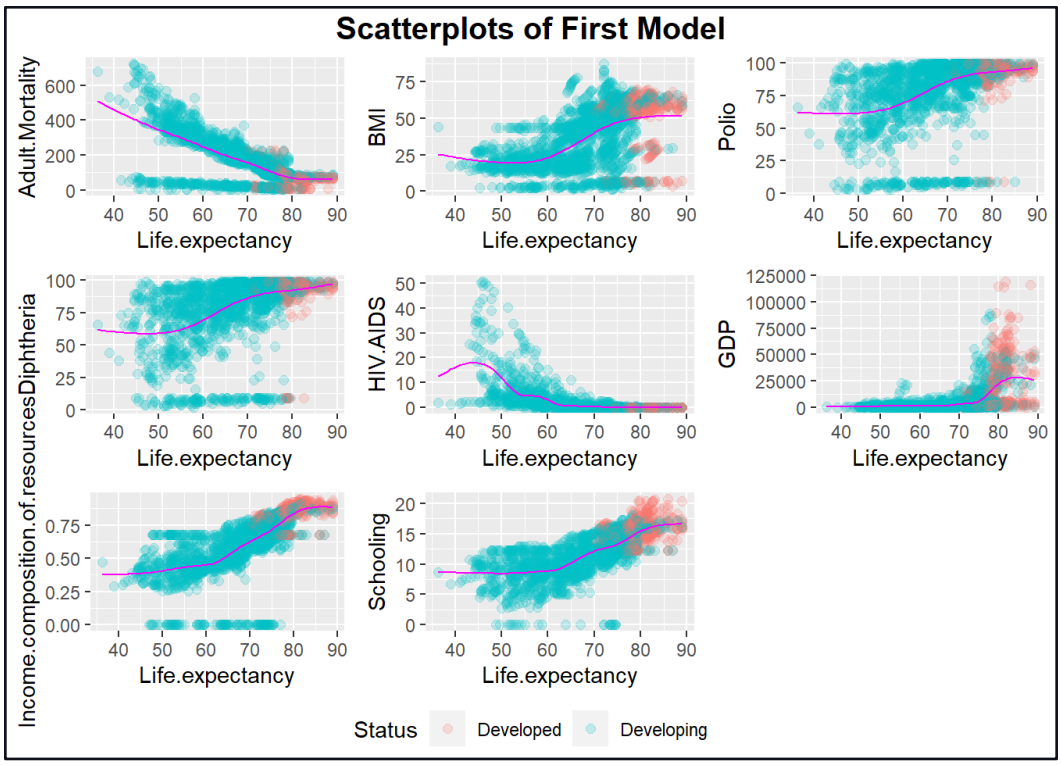


	Coefficients
(Intercept)	5.205059e+01
Adult.Mortality	-2.045099e-02
BMI	5.653341e-02
Polio	2.998498e-02
Diphtheria	4.182151e-02
HIV.AIDS	-4.624162e-01
GDP	4.353143e-05
Income.composition.of.resources	6.830094e+00
Schooling	6.934143e-01
Status_Num	1.897008e+00

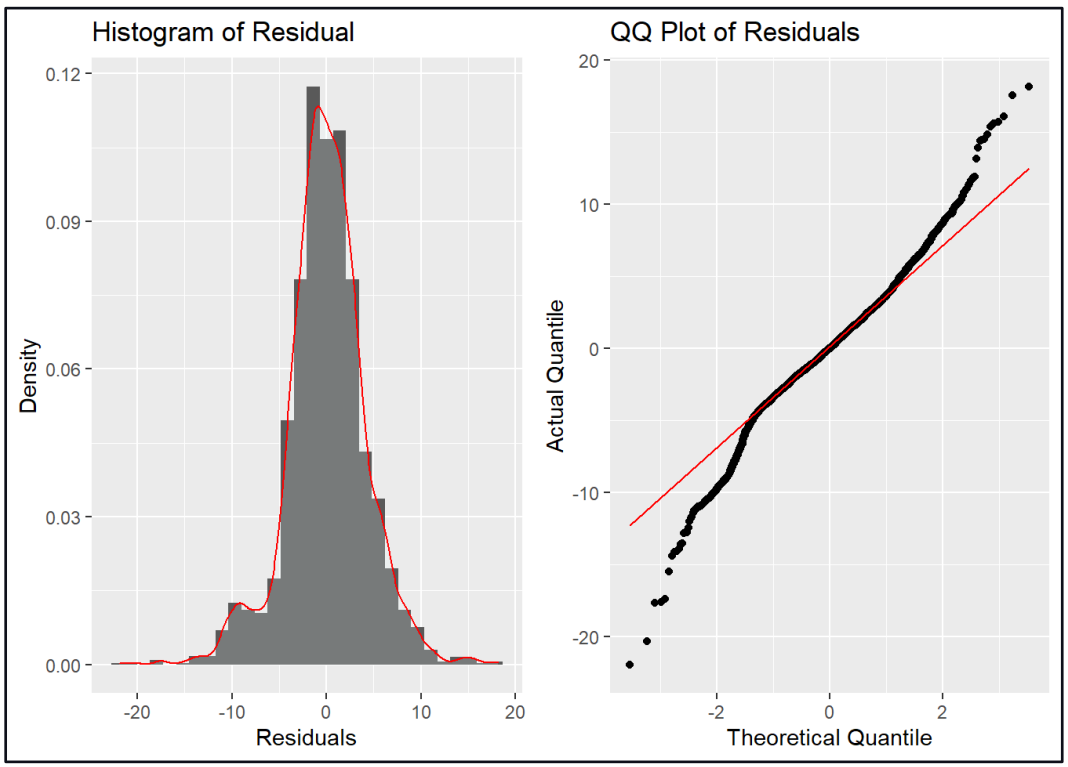
7.6 Train Response versus Predictions



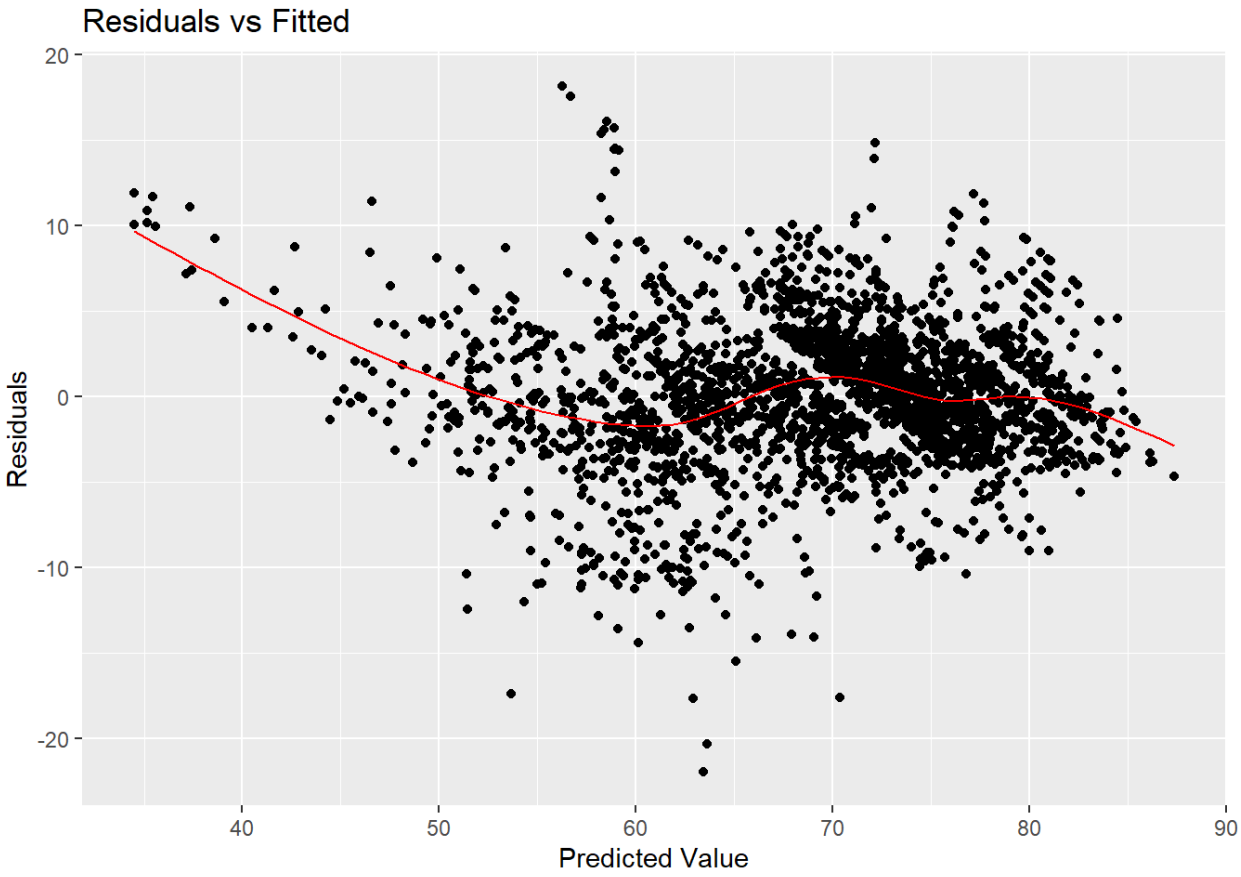
7.7 Linearity



7.8 Normality



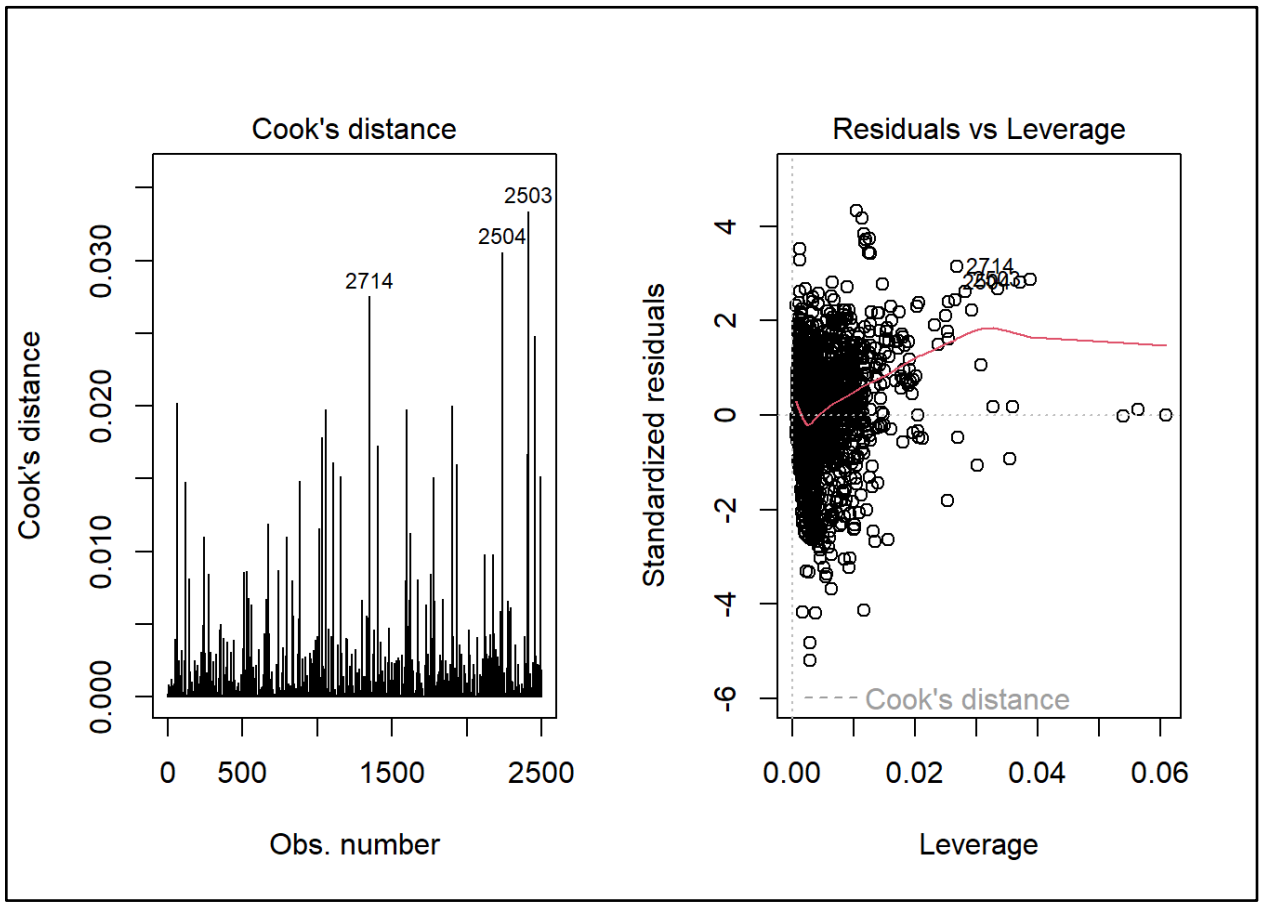
7.9 Homoscedasticity




```
lmtest::gqtest(first_model, fraction = (dim(train)[1]*.2))
```

```
##
##  Goldfeld-Quandt test
##
## data:  first_model
## GQ = 1.1558, df1 = 989, df2 = 988, p-value = 0.01147
## alternative hypothesis: variance increases from segment 1 to 2
```

7.10 Influential Point Analysis




```
as.numeric(names(cooks.distance(first_model))[(cooks.distance(first_model) > 0.05)])
```

```
## numeric(0)
```

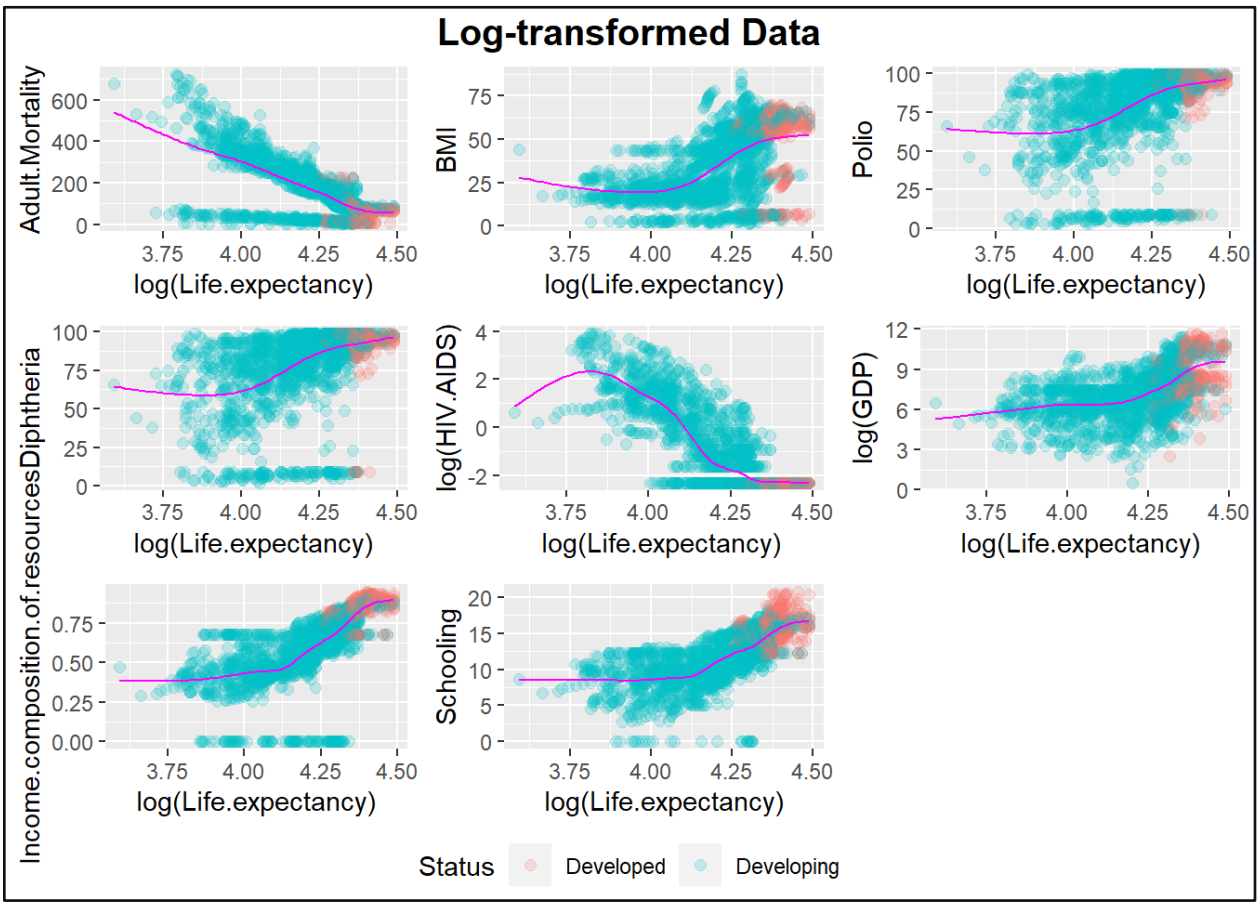
7.11 Complete First Linear Regression Model

```
##
## Call:
## lm(formula = Life.expectancy ~ Status + Adult.Mortality + BMI +
##     Polio + Diphtheria + HIV.AIDS + GDP + Income.composition.of.resources +
##     Schooling, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.9205  -2.2648  -0.0159   2.4528  18.1474
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.395e+01  6.178e-01  87.326  < 2e-16 ***
## StatusDeveloping               -1.897e+00  2.693e-01  -7.043 2.42e-12 ***
## Adult.Mortality                -2.045e-02  8.897e-04 -22.985  < 2e-16 ***
## BMI                             5.653e-02  5.054e-03  11.186  < 2e-16 ***
## Polio                           2.998e-02  4.941e-03   6.069 1.49e-09 ***
## Diphtheria                      4.182e-02  4.894e-03   8.545  < 2e-16 ***
## HIV.AIDS                       -4.624e-01  1.950e-02 -23.716  < 2e-16 ***
## GDP                             4.353e-05  7.337e-06   5.933 3.39e-09 ***
## Income.composition.of.resources  6.830e+00  7.113e-01   9.603  < 2e-16 ***
## Schooling                       6.934e-01  4.573e-02  15.164  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.215 on 2487 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.8021
## F-statistic:  1125 on 9 and 2487 DF,  p-value: < 2.2e-16
```

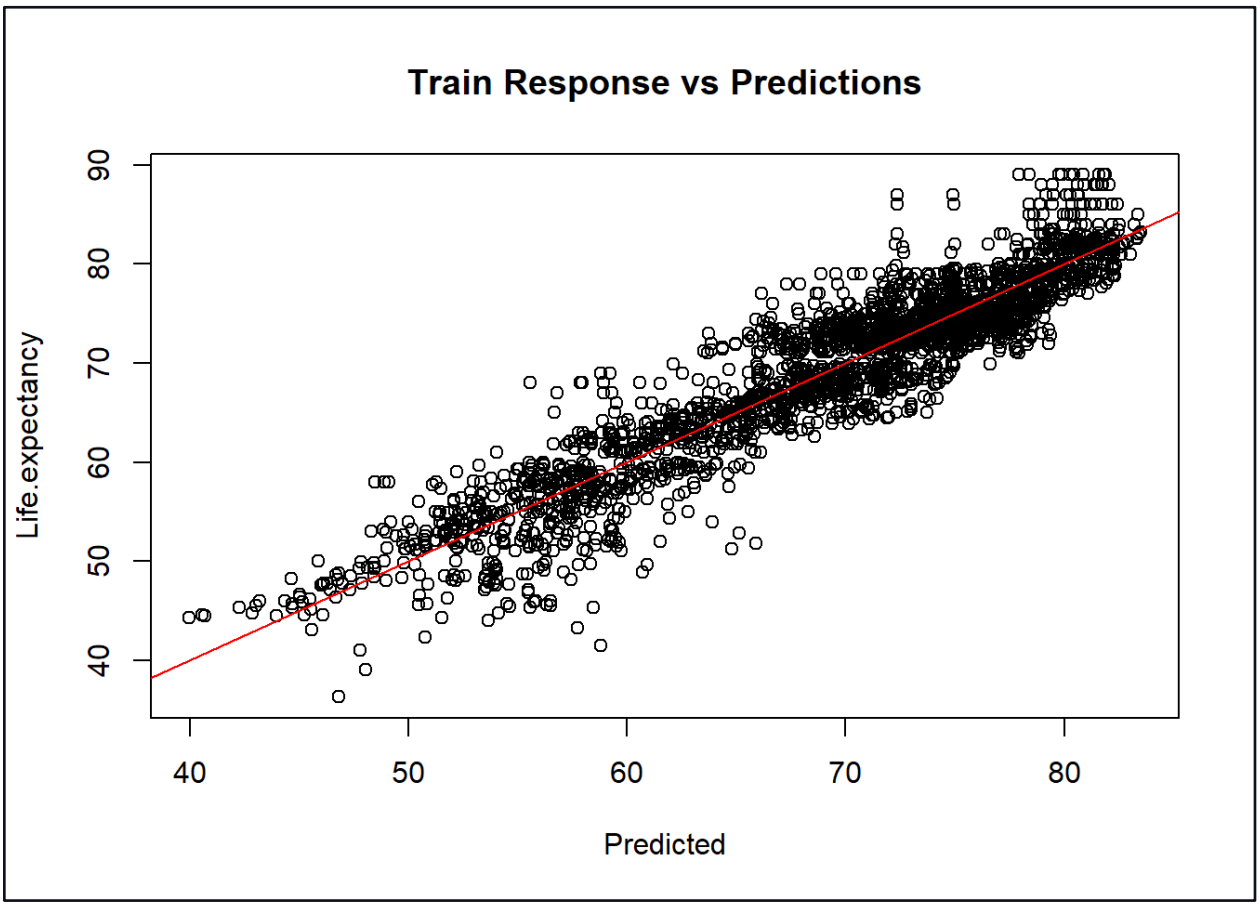
```
confint(first_model)
```

```
##                                       2.5 %        97.5 %
## (Intercept)                     5.273620e+01  5.515900e+01
## StatusDeveloping               -2.425167e+00 -1.368848e+00
## Adult.Mortality                -2.219571e-02 -1.870627e-02
## BMI                             4.662347e-02  6.644335e-02
## Polio                           2.029614e-02  3.967382e-02
## Diphtheria                      3.222454e-02  5.141848e-02
## HIV.AIDS                       -5.006498e-01 -4.241825e-01
## GDP                             2.914329e-05  5.791958e-05
## Income.composition.of.resources  5.435338e+00  8.224850e+00
## Schooling                       6.037481e-01  7.830804e-01
```

7.12 Log-transformed Data



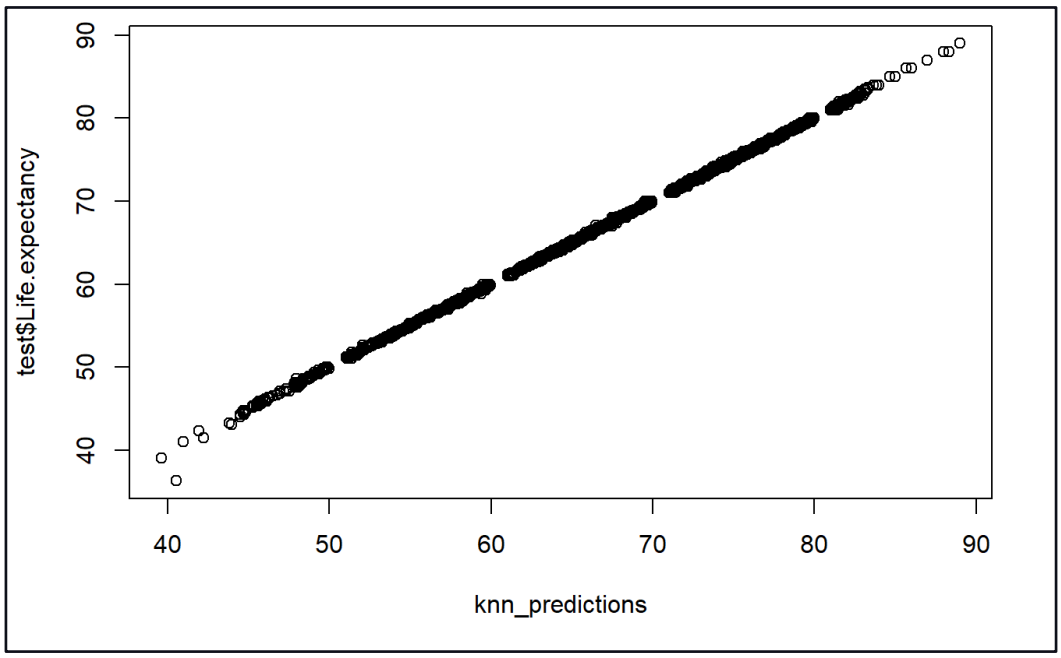
7.13 Second Model Predicted vs Actual Scatterplot



7.14 Third Model: Cross-Validation k-NN Regression

```
## k-Nearest Neighbors
##
## 2938 samples
##   17 predictor
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 2644, 2645, 2644, 2643, 2645, 2645, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    1  3.048414  0.8990076  1.810253
##    2  2.787359  0.9144539  1.768433
##    3  2.739158  0.9173738  1.792928
##    4  2.742154  0.9172711  1.826916
##    5  2.765006  0.9161476  1.870152
##    6  2.825174  0.9126242  1.929039
##    7  2.871578  0.9099259  1.980789
##    8  2.931813  0.9061907  2.034456
##    9  2.989935  0.9025860  2.082891
##   10  3.043195  0.8992411  2.122017
##   11  3.088082  0.8965254  2.160168
##   12  3.130053  0.8938859  2.197190
##   13  3.168937  0.8914584  2.233213
##   14  3.209752  0.8888868  2.266498
##   15  3.242239  0.8868224  2.294969
##   16  3.270925  0.8850250  2.321944
##   17  3.292818  0.8837108  2.344163
##   18  3.311946  0.8826450  2.363704
##   19  3.326783  0.8818634  2.380604
##   20  3.341254  0.8810687  2.394936
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 3.
```

7.15 k-NN Regression Predicted vs Actual



8 Code

Libraries

```
library(tidyverse)
library(psych)          # describe()
library(DataExplorer)   # plot_missing() | drop_columns()
library(caret)          # nearZeroVar() | knnreg()
library(inspectdf)      # inspect_cat() | show_plots()
library(ggstance)       # geom_boxploth()
library(corrplot)       # corrplot() | cor()
library(ggpubr)         # ggscatter()
library(MASS)           # stepAIC()
library(regclass)       # vif()
library(leaps)          # regsubsets()
library(ggplot2)        # ggplot()
library(purrr)          # map()
library(GGally)         # ggcorr()
library(lindia)         # gg_cooksd() | gg_scalelocation
library(gridExtra)      # grid.arrange
library(FNN)            # knn.reg()
library(Metrics)        # mse()
```

Import Data

```
getwd()
df = read.csv("Life Expectancy Data.csv")
```

Missing Values Alex

```
table(is.na(df))
```

```
##

## FALSE  TRUE

## 62073  2563

plot_missing(df)

### Impute missing columns with their medians

df$Life.expectancy[is.na(df$Life.expectancy)] = median(df$Life.expectancy,
na.rm = T)

df$Adult.Mortality[is.na(df$Adult.Mortality)] = median(df$Adult.Mortality,
na.rm = T)

df$Alcohol[is.na(df$Alcohol)] = median(df$Alcohol, na.rm = T)

df$Hepatitis.B[is.na(df$Hepatitis.B)] = median(df$Hepatitis.B, na.rm = T)

df$BMI[is.na(df$BMI)] = median(df$BMI, na.rm = T)

df$Polio[is.na(df$Polio)] = median(df$Polio, na.rm = T)

df$Total.expenditure[is.na(df$Total.expenditure)] =
median(df$Total.expenditure, na.rm = T)

df$Diphtheria[is.na(df$Diphtheria)] = median(df$Diphtheria, na.rm = T)

df$GDP[is.na(df$GDP)] = median(df$GDP, na.rm = T)

df$Population[is.na(df$Population)] = median(df$Population, na.rm = T)

df$thinness..1.19.years[is.na(df$thinness..1.19.years)] =
median(df$thinness..1.19.years, na.rm = T)

df$thinness.5.9.years[is.na(df$thinness.5.9.years)] =
median(df$thinness.5.9.years, na.rm = T)

df$Income.composition.of.resources[is.na(df$Income.composition.of.resources)]
= median(df$Income.composition.of.resources, na.rm = T)

df$Schooling[is.na(df$Schooling)] = median(df$Schooling, na.rm = T)

# Sanity check

table(is.na(df))

##

## FALSE
```

```
## 64636

plot_missing(df)
```

Missing Values

Zero Variance

```
# Identify the names of zero variance columns

zero_var_col_names = nearZeroVar(df, names = TRUE)

zero_var_col_names

## character(0)

# No zero variance columns found!
```

Preparing for EDA

```
str(df)

## 'data.frame':  2938 obs. of  22 variables:

##  $ Country                       : chr  "Afghanistan" "Afghanistan"
"Afghanistan" "Afghanistan" ...

##  $ Year                          : int  2015 2014 2013 2012 2011 2010 2009
2008 2007 2006 ...

##  $ 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                 : 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 ...
```

```
# Convert character variables into factors

df[sapply(df, is.character)] = lapply(df[sapply(df, is.character)],
as.factor)



# Remove Country because too many levels

# Remove Year because can cause autocorrelation

df = subset(df, select = -c(Country, Year))
```

EDA

```
str(df)

## 'data.frame':  2938 obs. of  20 variables:
```

```
##  $ 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                : num  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 ...

describe(df)

##                                  vars  n     mean        sd    median
## Status*                          1 2938      1.83        0.38        2.00
## Life.expectancy                  2 2938      69.23       9.51        72.10
## Adult.Mortality                  3 2938      164.73      124.09
      144.00
## infant.deaths                    4 2938      30.30       117.93      3.00
## Alcohol                          5 2938      4.55        3.92        3.76
## percentage.expenditure           6 2938      738.25      1987.91     64.91
## Hepatitis.B                      7 2938      83.02       23.00       92.00
## Measles                          8 2938      2419.59     11467.27
17.00
## BMI                              9 2938      38.38       19.94       43.50
## under.five.deaths                10 2938     42.04       160.45      4.00
## Polio                            11 2938     82.62       23.37       93.00
## Total.expenditure                12 2938     5.92        2.40        5.76
## Diphtheria                       13 2938     82.39       23.66       93.00
## HIV.AIDS                            14 2938        1.74        5.08
      0.10
## GDP                              15 2938     6611.52     13296.60
      1766.95
## Population                       16 2938 10230851.23 54022417.46
1386542.00
## thinness..1.19.years             17 2938      4.82        4.40        3.30
## thinness.5.9.years               18 2938     4.85        4.49        3.30
## Income.composition.of.resources   19 2938          0.63        0.21
      0.68
```

```
## Schooling                          20 2938        12.01          3.27
      12.30
##                                  trimmed     mad   min         max
## Status*                             1.91        0.00  1.00 2.000000e+00
## Life.expectancy                     69.93       8.45 36.30 8.900000e+01
## Adult.Mortality                  150.46      111.19  1.00 7.230000e+02
## infant.deaths                       10.20       4.45  0.00 1.800000e+03
## Alcohol                             4.17        4.62  0.01 1.787000e+01
## percentage.expenditure           230.74      96.24  0.00 1.947991e+04
## Hepatitis.B                         88.73       7.41  1.00 9.900000e+01
## Measles                          286.08      25.20  0.00 2.121830e+05
## BMI                                 39.12       23.87  1.00
8.730000e+01
## under.five.deaths                   14.15       5.93  0.00 2.500000e+03
## Polio                               88.11       8.90  3.00 9.900000e+01
## Total.expenditure                   5.83        2.16  0.37 1.760000e+01
## Diphtheria                          88.05       7.41  2.00 9.900000e+01
## HIV.AIDS                            0.54        0.00  0.10 5.060000e+01
## GDP                              3104.78     2115.86  1.68 1.191727e+05
## Population                    2859164.26 1804399.81 34.00 1.293859e+09
## thinness..1.19.years                4.12        3.41  0.10 2.770000e+01
## thinness.5.9.years                  4.14        3.41  0.10 2.860000e+01
## Income.composition.of.resources     0.65        0.17  0.00 9.500000e-01
## Schooling                           12.19       2.82  0.00 2.070000e+01
##                                        range  skew kurtosis          se
## Status*                          1.000000e+00 -1.72      0.95        0.01
## Life.expectancy                  5.270000e+01 -0.64      -0.23       0.18
## Adult.Mortality                  7.220000e+02  1.18      1.76        2.29
```

```
## infant.deaths                          1.800000e+03  9.78   115.76   2.18
## Alcohol                                1.786000e+01  0.65    -0.63   0.07
## percentage.expenditure                 1.947991e+04  4.65     26.51      36.68
## Hepatitis.B                            9.800000e+01 -2.28     4.39      0.42
## Measles                                2.121830e+05  9.43   114.58   211.56
## BMI                                   8.630000e+01 -0.23  -1.27      0.37
## under.five.deaths                      2.500000e+03  9.49   109.49   2.96
## Polio                                  9.600000e+01 -2.11     3.81      0.43
## Total.expenditure                      1.723000e+01  0.66    1.51   0.04
## Diphtheria                             9.700000e+01 -2.08     3.60      0.44
## HIV.AIDS                               5.050000e+01  5.39     34.80      0.09
## GDP                                    1.191711e+05  3.54     15.10 245.31
## Population                            1.293859e+09 17.95   380.22 996662.50
## thinness..1.19.years                   2.760000e+01  1.73     4.05      0.08
## thinness.5.9.years                     2.850000e+01  1.79     4.44      0.08
## Income.composition.of.resources 9.500000e-01 -1.21     1.69      0.00
## Schooling                              2.070000e+01 -0.63     1.12      0.06
table(is.na(df))
##
## FALSE
## 58760
dim(df)
## [1] 2938   20
# Correlations plot
ggcorr(df,
      label = T,
      label_size = 2,
```

```
      label_round = 2,
      hjust = 1,
      size = 3,
      color = "royalblue",
      layout.exp = 5,
      low = "salmon",
      mid = "white",
      high = "turquoise",
      name = "Correlation")
```

```
## Warning in ggcorr(df, label = T, label_size = 2, label_round = 2, hjust =
1, :
```

```
## data in column(s) 'Status' are not numeric and were ignored
```

```
# under.five.deaths and infant.deaths are problematic with corr = 1
```

```
## Dealing with under.five.deaths and infant.deaths
```

```
## under5_removed_df
```

```
under5_removed_df = subset(df, select = -c(under.five.deaths))
```

```
under5_removed_model = lm(Life.expectancy ~ ., data = under5_removed_df)
```

```
summary(under5_removed_model)$r.squared
```

```
## [1] 0.8107984
```

```
## infantdeaths_removed_df
```

```
infantdeaths_removed_df = subset(df, select = -c(infant.deaths))
```

```
infantdeaths_removed_model = lm(Life.expectancy ~ ., data =
infantdeaths_removed_df)
```

```
summary(infantdeaths_removed_model)$r.squared
```

```
## [1] 0.8111836
```

```
## Confirming infantdeaths_removed_df
```

```
df = infantdeaths_removed_df
```

```
# Further bivariate analysis

df %>% gather(-Life.expectancy, -Status, -GDP, key = "var", value =
"Features") %>%

  ggplot(aes(x = Features, y = Life.expectancy, color = Status, size = GDP))
+

  geom_point(shape = 1) +

  facet_wrap(~ var, scales = "free") + theme_bw()
```

Split the Data

```
## Seed to make partition reproducible

set.seed(1)


# train/test split

smp_size = floor(0.85 * nrow(df))


train_indices = sample(seq_len(nrow(df)), size = smp_size)


train = df[train_indices, ]

test = df[-train_indices, ]
```

Feature Selection

```
reg.mod = regsubsets(Life.expectancy~., data = train, nvmax = 20)

fwd.mod = regsubsets(Life.expectancy~., data = train, method = "forward",
nvmax = 20)

bwd.mod = regsubsets(Life.expectancy~., data = train, method = "backward",
nvmax = 20)


paste(max(summary(reg.mod)$rsq), " | ", max(summary(reg.mod)$adjr2), " | ",
min(summary(reg.mod)$rss), " | ", min(summary(reg.mod)$bic), " | ",
min(summary(reg.mod)$cp))
```

```
## [1] "0.809261916916481  |  0.808002022831507  |  42740.1888934973  |  -
4018.84838594768  |  14.3778941926116"
```

```
paste(max(summary(fwd.mod)$rsq), " | ", max(summary(fwd.mod)$adjr2), " | ",
min(summary(fwd.mod)$rss), " | ", min(summary(fwd.mod)$bic), " | ",
min(summary(fwd.mod)$cp))
```

```
## [1] "0.809261916916481  |  0.808002022831507  |  42740.1888934973  |  -
4018.84838594768  |  14.3778941926121"
```

```
paste(max(summary(bwd.mod)$rsq), " | ", max(summary(fwd.mod)$adjr2), " | ",
min(summary(bwd.mod)$rss), " | ", min(summary(fwd.mod)$bic), " | ",
min(summary(bwd.mod)$cp))
```

```
## [1] "0.809261916916481  |  0.808002022831507  |  42740.1888934973  |  -
4018.84838594768  |  14.3778941926121"
```

```
## Finding Number of Predictors For Feature Selection Model
```

```
### Predict Function
```

```
predict.regsubsets = function (object , newdata ,id ,...){

  form=as.formula (object$call [[2]])

  mat=model.matrix(form ,newdata )

  coefi=coef(object ,id=id)

  xvars=names(coefi)

  mat[,xvars]%*%coefi

}
```

```
testASE = c()
```

```
ncol(test)
```

```
## [1] 19
```

```
ncol(train)
```

```
## [1] 19
```

```
ncol(df)
```

```
## [1] 19
```

```
for (i in 1:(ncol(train)-2)){
  predictions = predict.regsubsets(object = reg.mod, newdata = test, id = i)
  testASE[i] = mean((test$Life.expectancy - predictions)^2)
  #print(testASE)
}


par(mfrow=c(1,1))
plot(1:(ncol(train)-2), testASE, type = "l", xlab = "Number of predictors",
ylab = "Test ASE")
axis(1, at = seq(round(min(1)), round(max((ncol(train)-2))), by = 1), labels
= 1:(ncol(train)-2))
index = which(testASE == min(testASE))
points(index, testASE[index], col = "red", pch = 10)
# We can see (via the red dot) that the minimum Average Squared Error happens
# with 2 predictors included in the model.


reg_final = regsubsets(Life.expectancy~., data = train, method = "backward",
nvmax = 10)
coef(reg_final, 9)
##                      (Intercept)            StatusDeveloping
##                     5.394760e+01                  -1.897008e+00
##                Adult.Mortality                             BMI
##                -2.045099e-02                  5.653341e-02
##                           Polio                   Diphtheria
##                     2.998498e-02                  4.182151e-02
##                      HIV.AIDS                             GDP
##                    -4.624162e-01                  4.353143e-05
## Income.composition.of.resources                    Schooling
```

```
##                          6.830094e+00                    6.934143e-01
```

First Regression Model

```
first_model = lm(Life.expectancy ~ Status + Adult.Mortality + BMI + Polio +
Diphtheria + HIV.AIDS +

                                    GDP + Income.composition.of.resources +
Schooling, data = train)

summary(first_model)

##

## Call:

## lm(formula = Life.expectancy ~ Status + Adult.Mortality + BMI +

##    Polio + Diphtheria + HIV.AIDS + GDP + Income.composition.of.resources +

##    Schooling, data = train)

##

## Residuals:

##    Min         1Q   Median       3Q    Max

## -21.9205  -2.2648  -0.0159   2.4528  18.1474

##

## Coefficients:

##                                  Estimate Std. Error t value Pr(>|t|)

## (Intercept)                      5.395e+01  6.178e-01  87.326  < 2e-16 ***

## StatusDeveloping                 -1.897e+00  2.693e-01  -7.043 2.42e-12
***

## Adult.Mortality                  -2.045e-02  8.897e-04 -22.985  < 2e-16
***

## BMI                              5.653e-02  5.054e-03  11.186  < 2e-16 ***

## Polio                            2.998e-02  4.941e-03   6.069 1.49e-09 ***

## Diphtheria                      4.182e-02  4.894e-03   8.545  < 2e-16 ***

## HIV.AIDS                         -4.624e-01  1.950e-02 -23.716  < 2e-16
***
```

```
## GDP                                4.353e-05  7.337e-06   5.933 3.39e-09 ***
## Income.composition.of.resources  6.830e+00  7.113e-01   9.603  < 2e-16 ***
## Schooling                          6.934e-01  4.573e-02  15.164  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.215 on 2487 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.8021
## F-statistic:  1125 on 9 and 2487 DF,  p-value: < 2.2e-16
```

```
## Assumptions
```

```
### Linearity
```

```
plot1 = ggplot(train, aes(x = Life.expectancy, y = Adult.Mortality, color =
Status, )) +
  geom_point(size = 2, alpha = 0.2) +
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")
plot2 = ggplot(train, aes(x = Life.expectancy, y = BMI, color = Status)) +
  geom_point(size = 2, alpha = 0.2) +
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")
plot3 = ggplot(train, aes(x = Life.expectancy, y = Polio, color = Status)) +
  geom_point(size = 2, alpha = 0.2) +
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")
plot4 = ggplot(train, aes(x = Life.expectancy, y = Diphtheria, color =
Status)) +
  geom_point(size = 2, alpha = 0.2) +
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")
plot5 = ggplot(train, aes(x = Life.expectancy, y = HIV.AIDS, color = Status))
+
  geom_point(size = 2, alpha = 0.2) +
```

```
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot6 = ggplot(train, aes(x = Life.expectancy, y = GDP, color = Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot7 = ggplot(train, aes(x = Life.expectancy, y =
Income.composition.of.resources, color = Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot8 = ggplot(train, aes(x = Life.expectancy, y = Schooling, color =
Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

figure = ggarrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8,
ncol = 3, nrow = 3,

                   common.legend = TRUE, legend = "bottom")

annotate_figure(figure, top = text_grob("Scatterplots of First Model", face =
"bold", size = 15))

# Adult.Mortality looks quadratic

# Schooling looks cubic


### Normality

## Residuals Histogram

p1 = ggplot(train, aes(resid(first_model))) +

  geom_histogram(aes(y = ..density..)) +

  geom_density(alpha = .2, color = "red", fill = "azure") +

  labs(title = "Histogram of Residual", x = "Residuals", y = "Density")

## Residuals QQ Plot

p4 = ggplot(train, aes(sample = resid(first_model))) +

  geom_qq() +
```

```
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y =
"Actual Quantile")
grid.arrange(p1, p4, ncol = 2)
# Data is observed to be normal.


### Multicollinearity
1/(1-summary(first_model)$adj.r.squared)
## [1] 5.052998
VIF(first_model)
##                             Status                    Adult.Mortality
##                           1.472095                           1.700100
##                                BMI                              Polio
##                           1.410836                           1.872359
##                           Diphtheria                            HIV.AIDS
##                           1.879537                           1.398239
##                                GDP Income.composition.of.resources
##                           1.376831                           2.951183
##                           Schooling
##                           3.110998
# All variables have VIF < 5


### Autocorrelation (not evaluated)
acf(first_model$residuals, type = "correlation")
lmtest::dwtest(first_model)
##
##  Durbin-Watson test
```

```
##
## data:  first_model
## DW = 2.0418, p-value = 0.8517
## alternative hypothesis: true autocorrelation is greater than 0
#### ACF plot is high in lag 1 but drops suddenly in lag 2
#### P-value > 0.5 from Durbin-Watson test so no autocorrelation detected


### Homoscedasticity
#### Residual vs Fitted
ggplot(first_model, aes(fitted(first_model), resid(first_model))) +
  geom_point() +
  geom_smooth(color = "red", se = FALSE, size = 0.5) +
  labs(title = "Residuals vs Fitted", x = "Predicted Value", y = "Residuals")
lmtest::gqtest(first_model, fraction = (dim(train)[1]*.2))
##
##  Goldfeld-Quandt test
##
## data:  first_model
## GQ = 1.1558, df1 = 989, df2 = 988, p-value = 0.01147
## alternative hypothesis: variance increases from segment 1 to 2
# No evidence against homoscedasticity seen in plot, no fan out from left to
right or other way
# P-value > 0.5 from Goldfeld-Quandt test so no heteroscedasticity detected


### Influential Points
par(mfrow=c(1,2))
plot(first_model, 4)
```

```
plot(first_model, 5)

as.numeric(names(cooks.distance(first_model))[(cooks.distance(first_model) >
0.05)])

## numeric(0)

# No observations detected with Cook's D > 0.05

# Leverage plot looks good as well


### Interpretation and CI

summary(first_model)

##

## Call:

## lm(formula = Life.expectancy ~ Status + Adult.Mortality + BMI +

##    Polio + Diphtheria + HIV.AIDS + GDP + Income.composition.of.resources +

##    Schooling, data = train)

##

## Residuals:

##    Min         1Q   Median       3Q    Max

## -21.9205  -2.2648  -0.0159   2.4528  18.1474

##

## Coefficients:

##                                  Estimate Std. Error t value Pr(>|t|)

## (Intercept)                      5.395e+01  6.178e-01  87.326  < 2e-16 ***

## StatusDeveloping                 -1.897e+00  2.693e-01  -7.043 2.42e-12
***

## Adult.Mortality                  -2.045e-02  8.897e-04 -22.985  < 2e-16
***

## BMI                              5.653e-02  5.054e-03  11.186  < 2e-16 ***

## Polio                            2.998e-02  4.941e-03   6.069 1.49e-09 ***
```

```
## Diphtheria                          4.182e-02  4.894e-03   8.545  < 2e-16 ***
## HIV.AIDS                            -4.624e-01  1.950e-02 -23.716  < 2e-16
***
## GDP                                 4.353e-05  7.337e-06   5.933 3.39e-09 ***
## Income.composition.of.resources  6.830e+00  7.113e-01   9.603  < 2e-16 ***
## Schooling                           6.934e-01  4.573e-02  15.164  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.215 on 2487 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.8021
## F-statistic:  1125 on 9 and 2487 DF,  p-value: < 2.2e-16
confint(first_model)
##                                        2.5 %       97.5 %
## (Intercept)                         5.273620e+01  5.515900e+01
## StatusDeveloping                    -2.425167e+00 -1.368848e+00
## Adult.Mortality                     -2.219571e-02 -1.870627e-02
## BMI                                 4.662347e-02  6.644335e-02
## Polio                               2.029614e-02  3.967382e-02
## Diphtheria                          3.222454e-02  5.141848e-02
## HIV.AIDS                            -5.006498e-01 -4.241825e-01
## GDP                                 2.914329e-05  5.791958e-05
## Income.composition.of.resources  5.435338e+00  8.224850e+00
## Schooling                           6.037481e-01  7.830804e-01
### Statistics
paste(mean((test$Life.expectancy - predict(first_model, test))^2), " | ", #
test ASE
      summary(first_model)$r.squared)                                  # R2
```

```
## [1] "16.9333662458191  |  0.802811287082203"

# ASE converted from log-transform is a little high

# R-squared leaves to be desired


### Predicted vs Actual Plot

par(mfrow=c(1,1))

plot(first_model$fitted.values, train$Life.expectancy,

      xlab = "Predicted", ylab = "Life.expectancy",

      main = "Train Response vs Predictions",

      xlim = c(0,100), ylim = c(0,100))

lines(c(0,100), c(0,100), col = "red")
```

Fitting the second model

Log-transformed Data Scatterplots

```
plot1 = ggplot(train, aes(x = log(Life.expectancy), y = Adult.Mortality,
color = Status, )) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot2 = ggplot(train, aes(x = log(Life.expectancy), y = BMI, color = Status))
+

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot3 = ggplot(train, aes(x = log(Life.expectancy), y = Polio, color =
Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot4 = ggplot(train, aes(x = log(Life.expectancy), y = Diphtheria, color =
Status)) +

  geom_point(size = 2, alpha = 0.2) +
```

```
  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot5 = ggplot(train, aes(x = log(Life.expectancy), y = log(HIV.AIDS), color
= Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot6 = ggplot(train, aes(x = log(Life.expectancy), y = log(GDP), color =
Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot7 = ggplot(train, aes(x = log(Life.expectancy), y =
Income.composition.of.resources, color = Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

plot8 = ggplot(train, aes(x = log(Life.expectancy), y = Schooling, color =
Status)) +

  geom_point(size = 2, alpha = 0.2) +

  geom_smooth(size = 0.5, se = FALSE, color = "magenta")

figure = ggarrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8,
ncol = 3, nrow = 3,

                   common.legend = TRUE, legend = "bottom")

annotate_figure(figure, top = text_grob("Log-transformed Data", face =
"bold", size = 15))

train$Status = ifelse(train$Status == 'Developed', 1, 0)

test$Status = ifelse(test$Status == 'Developed', 1, 0)



# Start fit
```

```
second_model = lm(log(Life.expectancy) ~ Status + Adult.Mortality +
I(Adult.Mortality^2) + BMI + I(BMI^2) + I(BMI^3) + Polio + I(Polio^2) +
I(Polio^3) + Diphtheria + I(Diphtheria^2) + I(Diphtheria^3) + log(HIV.AIDS) +
I(log(HIV.AIDS)^2) + I(log(HIV.AIDS)^3) + log(GDP) + I(log(GDP)^2) +
I(log(GDP)^3) + Income.composition.of.resources +
I(Income.composition.of.resources^2) + I(Income.composition.of.resources^3) +
Schooling + I(Schooling^2) + I(Schooling^3), data = train)

summary(second_model)

##

## Call:

## lm(formula = log(Life.expectancy) ~ Status + Adult.Mortality +

##     I(Adult.Mortality^2) + BMI + I(BMI^2) + I(BMI^3) + Polio +

##     I(Polio^2) + I(Polio^3) + Diphtheria + I(Diphtheria^2) +

##     I(Diphtheria^3) + log(HIV.AIDS) + I(log(HIV.AIDS)^2) +
I(log(HIV.AIDS)^3) +

##     log(GDP) + I(log(GDP)^2) + I(log(GDP)^3) +
Income.composition.of.resources +

##     I(Income.composition.of.resources^2) +
I(Income.composition.of.resources^3) +

##     Schooling + I(Schooling^2) + I(Schooling^3), data = train)

##

## Residuals:

##     Min         1Q   Median       3Q    Max

## -0.34829 -0.02833 -0.00014  0.02972  0.20199

##

## Coefficients:

##                                      Estimate Std. Error t value
Pr(>|t|)

## (Intercept)                           4.161e+00  3.571e-02 116.533  <
2e-16 ***

## Status                               1.744e-02  3.860e-03   4.519 6.51e-
06 ***
```

```
## Adult.Mortality                          4.174e-05  2.765e-05   1.510
0.131209

## I(Adult.Mortality^2)                    -6.098e-07  6.084e-08 -10.024  < 2e-
16 ***

## BMI                                      -2.084e-03  5.917e-04  -3.521
0.000437 ***

## I(BMI^2)                                 7.455e-05  1.790e-05   4.165 3.22e-
05 ***

## I(BMI^3)                                 -6.379e-07  1.559e-07  -4.092
4.42e-05 ***

## Polio                                    -1.734e-03  1.095e-03  -1.583
0.113453

## I(Polio^2)                               3.568e-05  2.437e-05   1.464
0.143297

## I(Polio^3)                               -1.932e-07  1.486e-07  -1.301
0.193520

## Diphtheria                               -7.097e-03  1.047e-03  -6.778
1.52e-11 ***

## I(Diphtheria^2)                          1.449e-04  2.324e-05   6.233 5.36e-
10 ***

## I(Diphtheria^3)                          -7.606e-07  1.422e-07  -5.347
9.75e-08 ***

## log(HIV.AIDS)                            -3.972e-02  2.168e-03 -18.323  <
2e-16 ***

## I(log(HIV.AIDS)^2)                       -3.654e-03  6.495e-04  -5.626
2.05e-08 ***

## I(log(HIV.AIDS)^3)                       1.278e-03  3.338e-04   3.828
0.000132 ***

## log(GDP)                                 3.638e-03  1.511e-02   0.241
0.809823

## I(log(GDP)^2)                            -4.946e-04  2.256e-03  -0.219
0.826526

## I(log(GDP)^3)                            3.793e-05  1.072e-04   0.354
0.723536
```

```
## Income.composition.of.resources        -1.912e-01  6.824e-02  -2.802
0.005112 **

## I(Income.composition.of.resources^2)  4.905e-01  1.873e-01   2.619
0.008879 **

## I(Income.composition.of.resources^3) -1.455e-01  1.318e-01  -1.104
0.269875

## Schooling                             -1.626e-03  4.005e-03  -0.406
0.684856

## I(Schooling^2)                        7.422e-04  4.571e-04   1.624
0.104528

## I(Schooling^3)                        -2.849e-05  1.474e-05  -1.932
0.053416 .

## ---

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 0.05335 on 2472 degrees of freedom

## Multiple R-squared:  0.8683, Adjusted R-squared:  0.867

## F-statistic: 678.9 on 24 and 2472 DF,  p-value: < 2.2e-16
```

```
### Statistics

paste(mean((test$Life.expectancy - exp(predict(second_model,test)))^2),  " |
", # test ASE

      summary(second_model)$r.squared)
      # R2
```

```
## [1] "12.2423912128029  |  0.868270622433304"
```

```
### Predicted vs Actual Plot

par(mfrow=c(1,1))

plot(exp(second_model$fitted.values), train$Life.expectancy,

      xlab = "Predicted", ylab = "Life.expectancy",

      main = "Train Response vs Predictions")

lines(c(0,100), c(0,100), col="red")
```

Fitting the third model

10-fold 10-repeats k-NN Regression

```
### Apply all columns as numerics

train = as.data.frame(apply(train, 2, as.numeric))

test = as.data.frame(apply(train, 2, as.numeric))


### Normalizing dataset, scale all columns except Life.expectancy

### Randomizing dataset

#### Train

MinMaxScaling = function(x)

{

  return ((x - min(x))/(max(x)-min(x)))

}

Life.expectancy = train$Life.expectancy

train = as.data.frame(apply(train, 2, MinMaxScaling))

train = train[sample(nrow(train)),]

train$Life.expectancy = Life.expectancy

#### Test

Life.expectancy = test$Life.expectancy

test = as.data.frame(apply(test, 2, MinMaxScaling))

test = test[sample(nrow(test)),]

test$Life.expectancy = Life.expectancy


### Cross-validation to find best value for k

### excluding Status since not numeric

cv_knn = train(Life.expectancy ~ ., data = df[-1], method = "knn",

                  trControl = trainControl(method = "repeatedcv",
```

```
                                          number = 10, repeats = 10),
                tuneGrid = expand.grid(k = c(1:20)),
                  preProcess = c("center", "scale"))
print(cv_knn)
## k-Nearest Neighbors
##
## 2938 samples
##   17 predictor
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 2644, 2645, 2644, 2643, 2645, 2645, ...
## Resampling results across tuning parameters:
##
##   k   RMSE     Rsquared   MAE
##    1  3.048414  0.8990076  1.810253
##    2  2.787359  0.9144539  1.768433
##    3  2.739158  0.9173738  1.792928
##    4  2.742154  0.9172711  1.826916
##    5  2.765006  0.9161476  1.870152
##    6  2.825174  0.9126242  1.929039
##    7  2.871578  0.9099259  1.980789
##    8  2.931813  0.9061907  2.034456
##    9  2.989935  0.9025860  2.082891
##   10  3.043195  0.8992411  2.122017
##   11  3.088082  0.8965254  2.160168
##   12  3.130053  0.8938859  2.197190
```

```
##   13  3.168937  0.8914584  2.233213
##   14  3.209752  0.8888868  2.266498
##   15  3.242239  0.8868224  2.294969
##   16  3.270925  0.8850250  2.321944
##   17  3.292818  0.8837108  2.344163
##   18  3.311946  0.8826450  2.363704
##   19  3.326783  0.8818634  2.380604
##   20  3.341254  0.8810687  2.394936
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 3.
### Start the algorithm
#str(train)
third_model = knn.reg(train = train, y = train$Life.expectancy, k = 3)



# Validating our knn regression model
knn_predictions = third_model$pred
## Peek into the ranges of our predicted and actual values, looks good!
summary(test$Life.expectancy)
##    Min. 1st Qu.  Median    Mean 3rd Qu.      Max.
##   36.30   63.30   72.10   69.26   75.60   89.00
summary(knn_predictions)
##    Min. 1st Qu.  Median    Mean 3rd Qu.      Max.
##   39.60   63.27   72.10   69.27   75.53   89.00
## Statistics
```

```r
mean((test$Life.expectancy - knn_predictions)^2)      # MSE or test ASE or
MPSE

## [1] 0.02376363

third_model$R2Pred                                    # R-Squared

## [1] 0.9997352

#caret::RMSE(test$Life.expectancy, knn_predictions)  # RMSE


### Predicted vs Actual Plot

par(mfrow=c(1,1))

plot(knn_predictions, test$Life.expectancy)
```

References

[1] Kaggle (2018). *Life Expectancy (WHO): Statistical Analysis on factors influencing Life Expectancy*. Data retrieved May 18, 2022, from https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

[2] Mayoclinic. Diphtheria: Data retrieved June 1, 2022 from https://www.mayoclinic.org/diseases-conditions/diphtheria/symptoms-causes/syc-20351897

[3] Newscientist(2018). *More education is what makes people live longer, not more money.* Data retrieved June 1, 2022, from https://www.newscientist.com/article/2166833-more-education-is-what-makes-people-live-longer-not-more-money/

[4] James, Gareth, et al. "An Introduction to Statistical Learning." from https://hastie.su.domains/ISLR2/ISLRv2_website.pdf. Accessed 4 June 2022.