Applied Statistics
Project 1

Chad Reo
Paul Swenson
Indy Dhillon
Paul Huggins

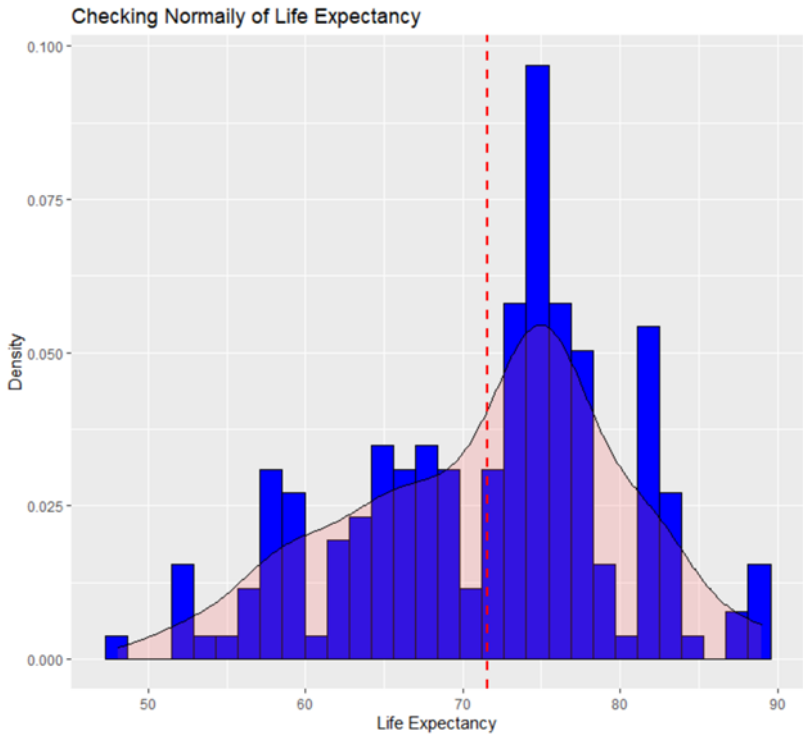
Introduction

This focus of this paper will be on creating models to predict life expectancy. The data set used for this analysis is made available by The Global Health Observatory data repository under World Health Organization (WHO). To make our predictions for life expectancy we will be using regression models to create a line of best fit. In addition, we will be using various techniques to assist in sifting through the data and finding the variables that work together to create the best fit. As we work through each section we will elaborate on the tools, techniques, and measures used to determine why we chose a given model along with the results and interpretation of the model.

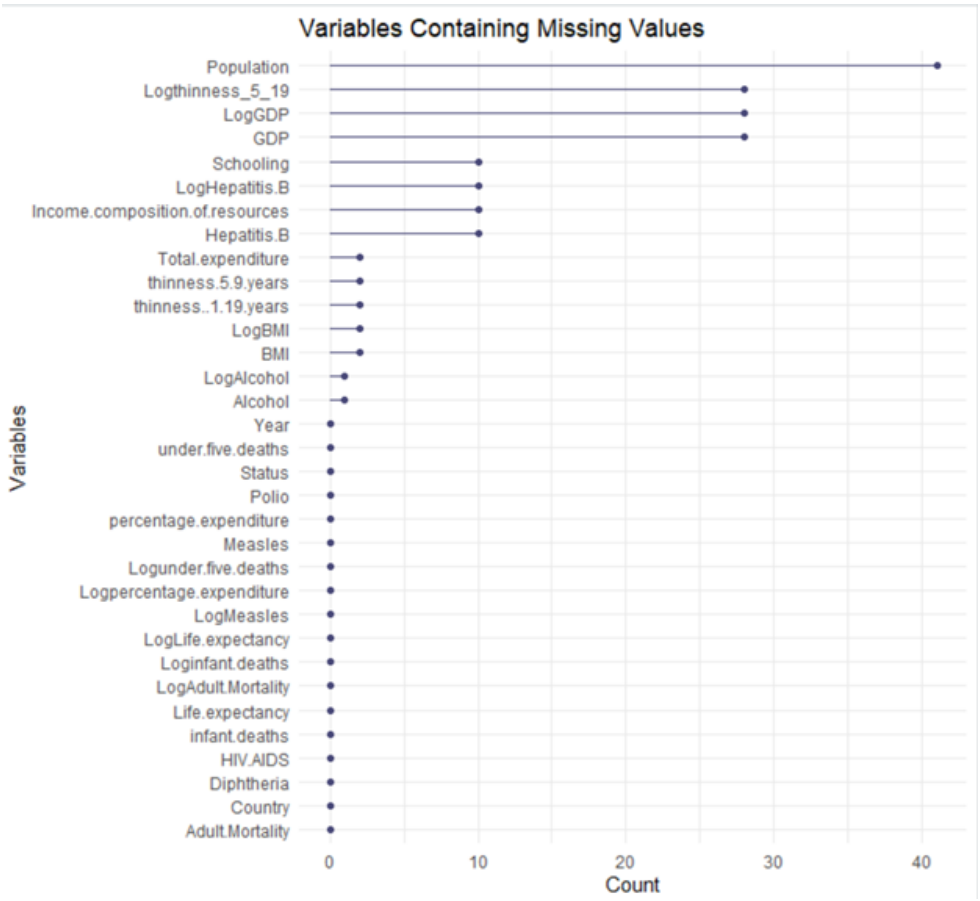
Objective 1: Regression Model

EDA

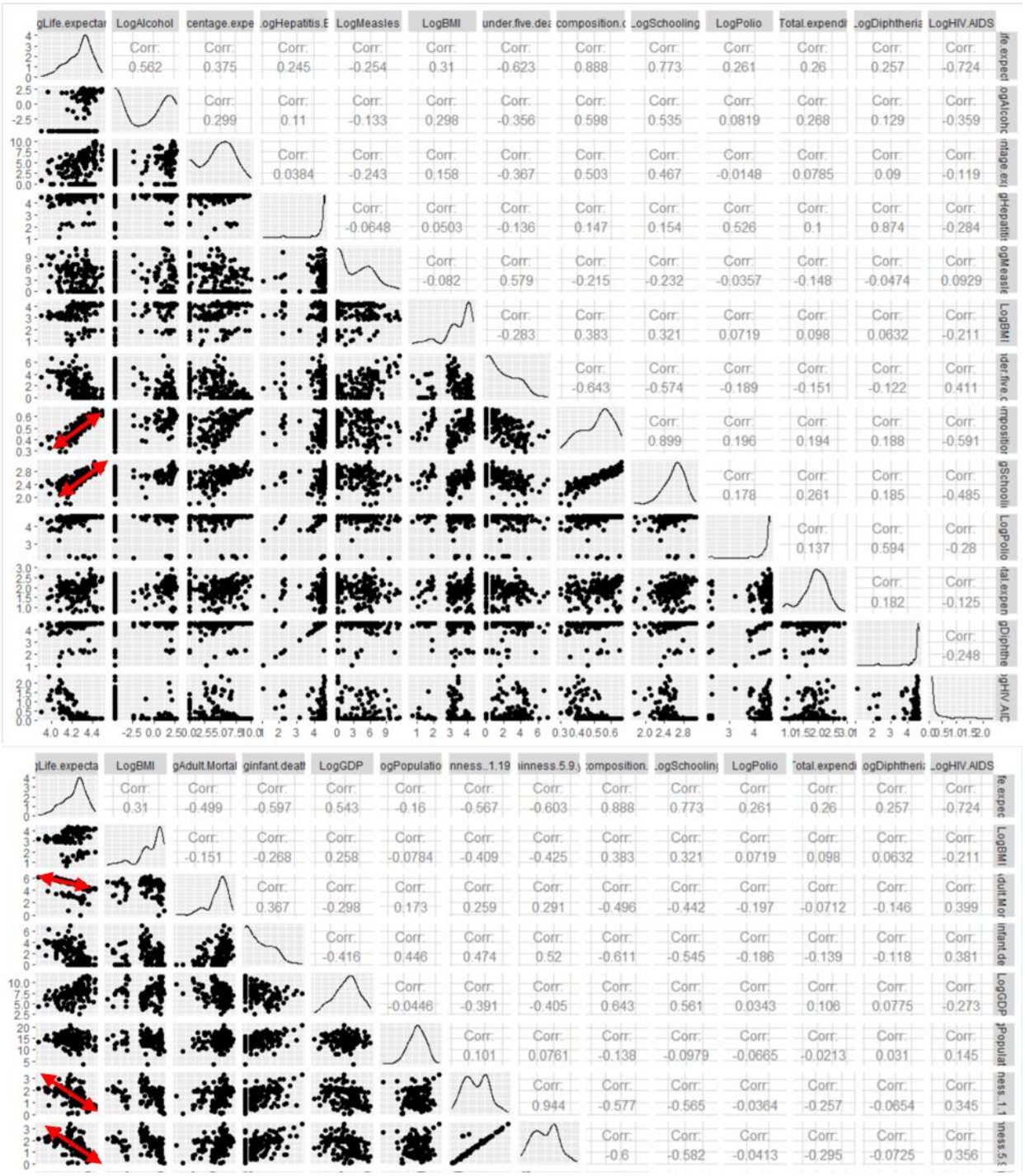
For this exercise we limited the WHO data down to the year 2014 only. This data contains thirty-two fields, including our predicted value, Life Expectancy, for 183 distinct countries. The values for Life Expectancy are fairly normal in distribution as judged by the histogram and density plots below. While the graph does show to be slightly left skewed, the data has more than enough data points to assume normality.



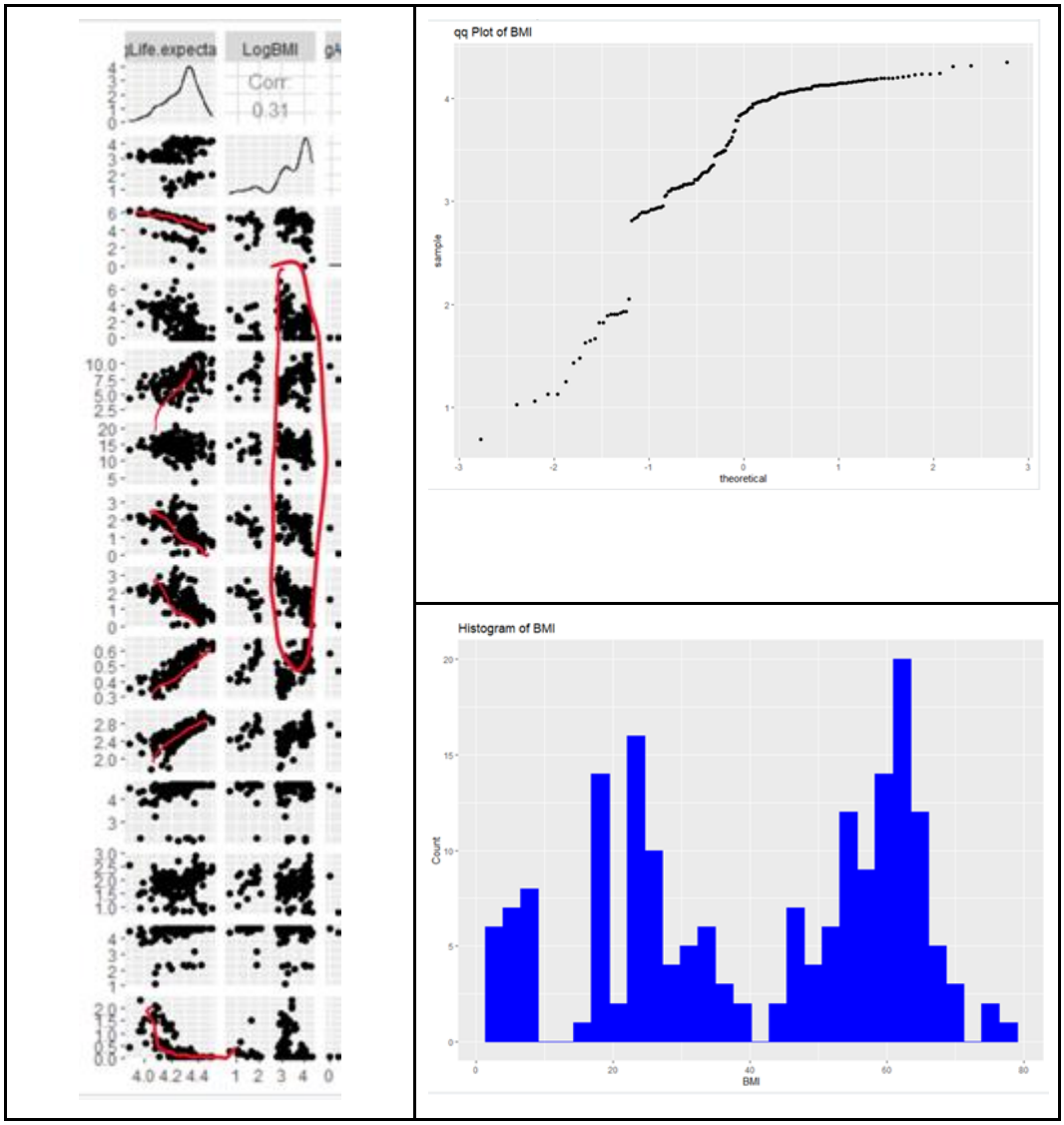
The next piece to look at is to check the quality of the data. For our thirty-two fields, we check to see how many, if any, are missing values. From the chart below we can see that our predicted value, Life Expectancy, is not missing any values; however there are few other variables that are. Depending on whether or not these variables are used in our model will dictate if we need to further modify these data points for missing values.



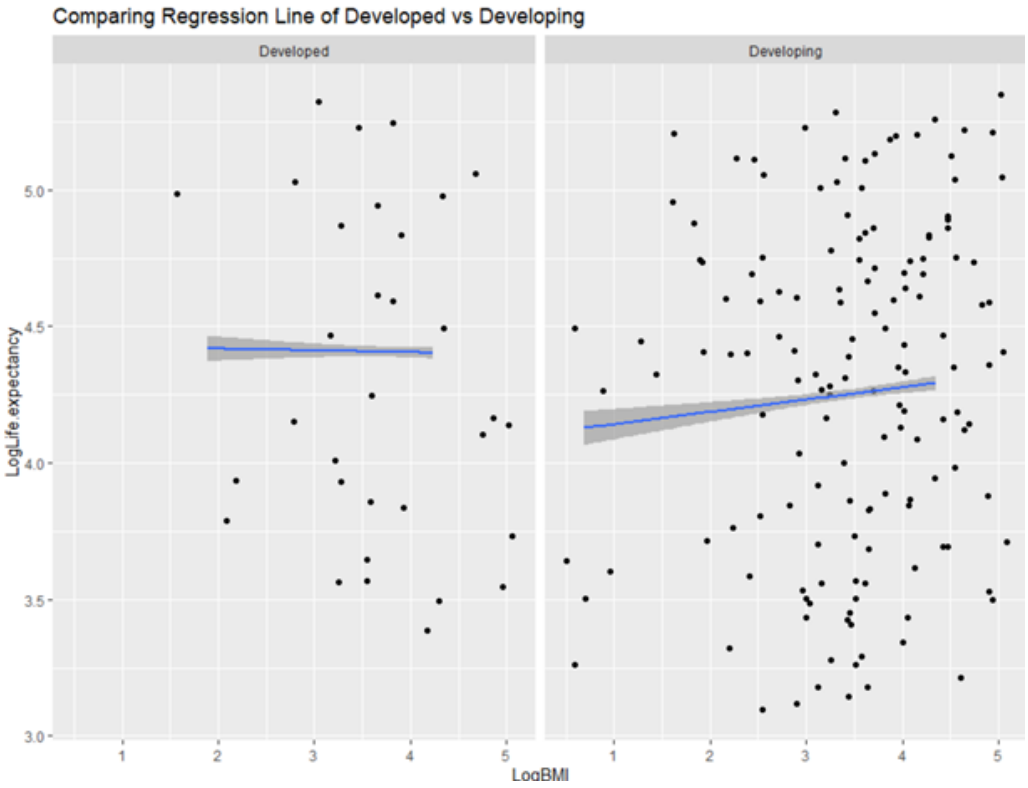
After acquiring, loading, and cleaning the data, much of the work begins with Exploratory Data Analysis (EDA). The first thing to do is look at the data to see what fields are available, what their values are, and what kind of relationships are present. The better the relationship the more it can be used for prediction. The red lines below highlight variables that appear to be positively and negatively correlated, i.e. as one variable goes up in value, life expectancy has a similar or opposite effect in a linear manner.



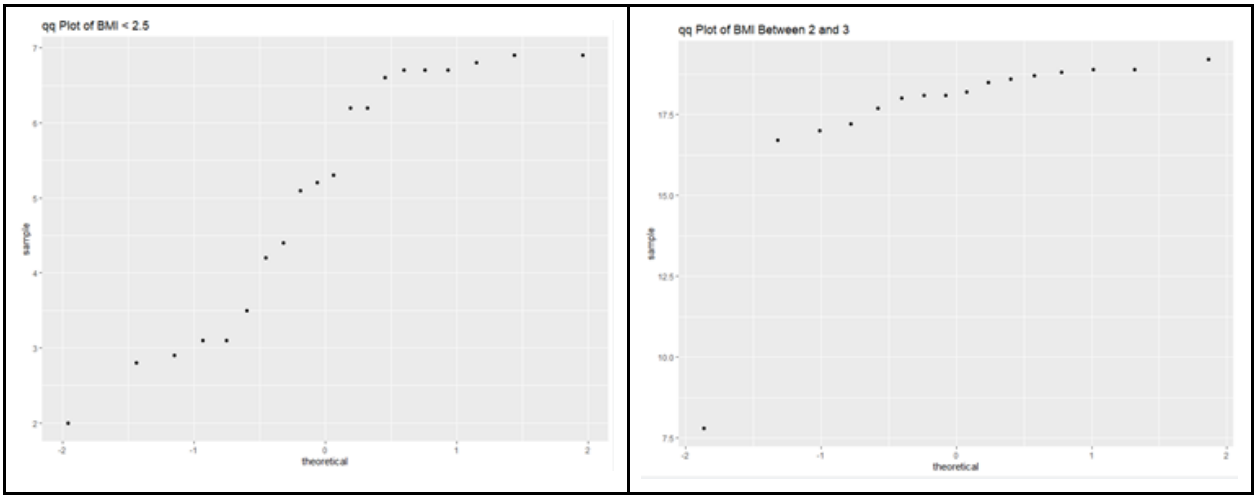
To further investigate this concept, let us review BMI. BMI data does not appear to be very normally distributed as can be seen from the non-normal histogram and the gaps and curves in the qq plot. In addition to the non-normal data, there appears to be blocks, or separations within the BMI data. The scatter, histogram, and the qq plot show there are clear gaps.

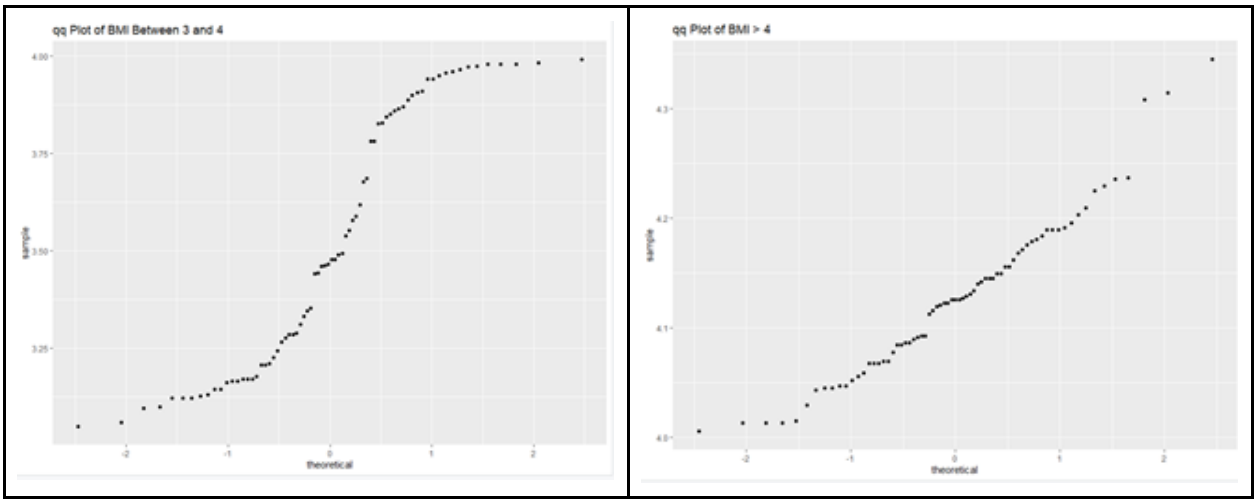


We can further segment this data into groups of Developed and Developing. From here a distinct difference can also be seen. In Developing countries, the higher the BMI, the higher the life expectancy. While in Developed countries, the opposite seems true, where the higher the BMI the lower the Life Expectancy.



If BMI is blocked into four distinct groups based on log transformation of BMI, only the > 4 is normally distributed as evidenced by the straight line qq plot.



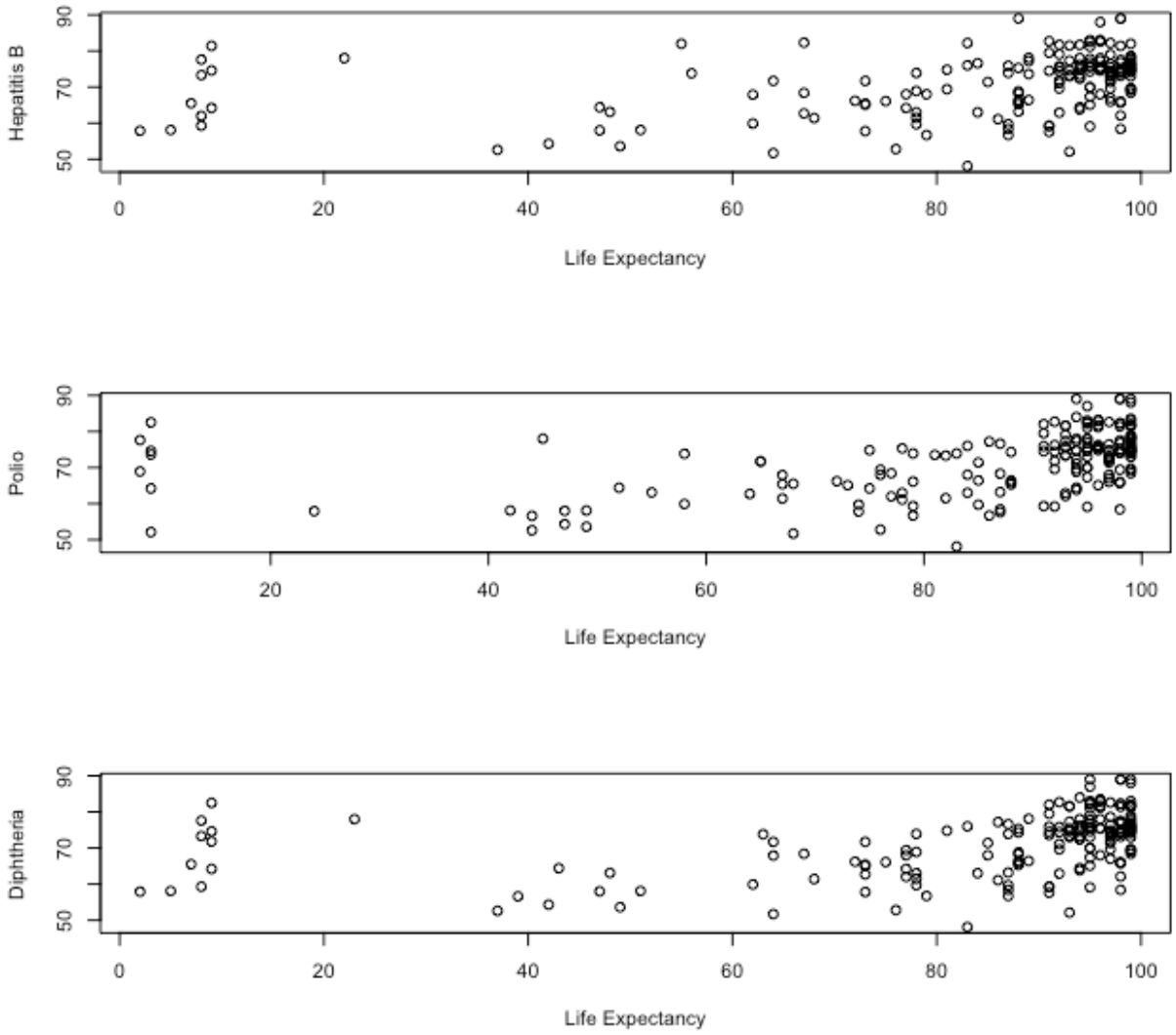


When creating a regression line for each block/bin combination they all appear to have different slopes.

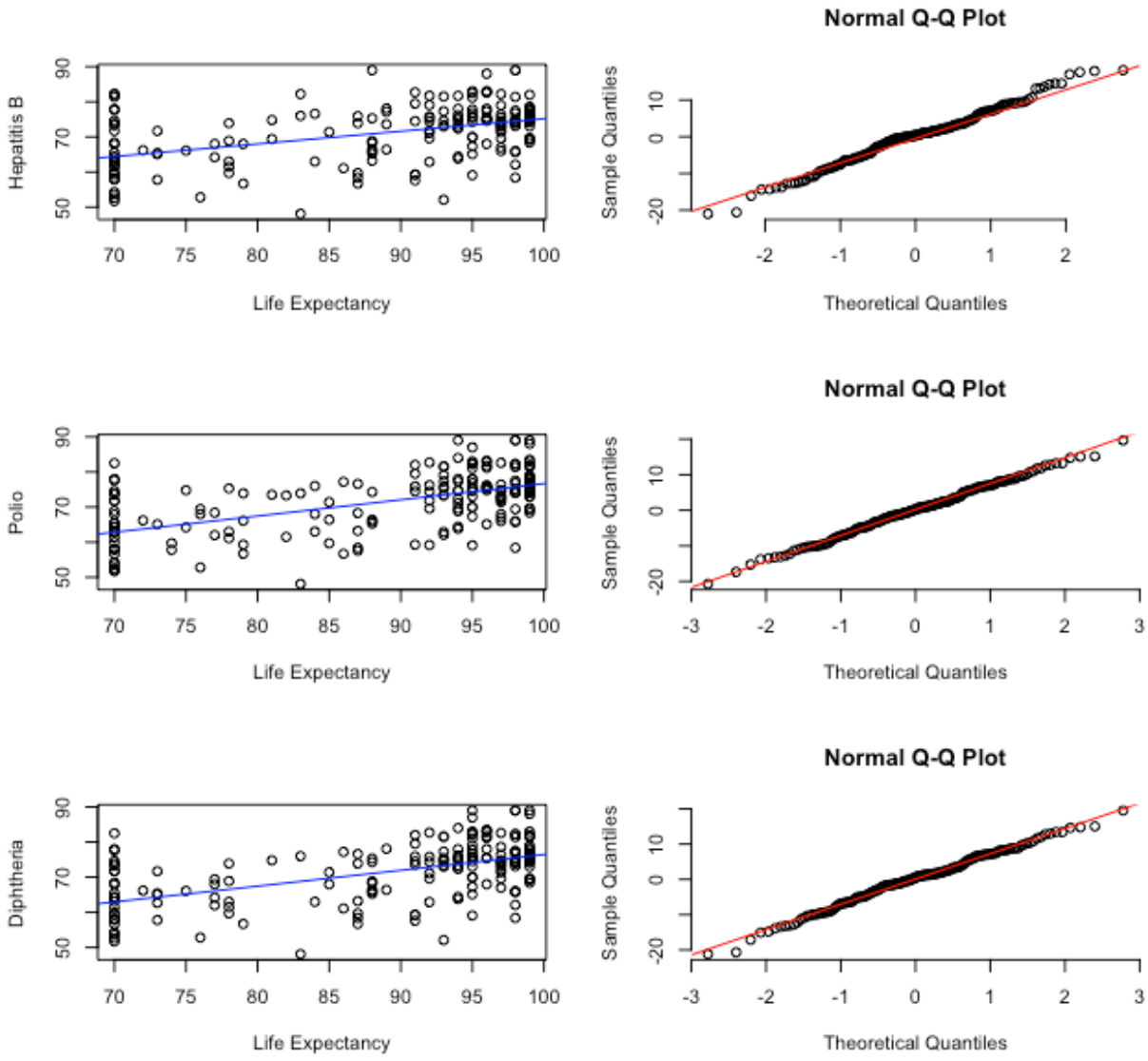
In fact, when testing the means of the different bins by combining the two lowest against the two highest, there is no statistical difference in the means, with a p-value of .1071, indicating that we cannot reject the null hypothesis that the means are equal. But when the lowest three

bins are tested against the highest bin we can see that there is a statistical difference in the means with a p-value of .02 X $10^{16.}$.

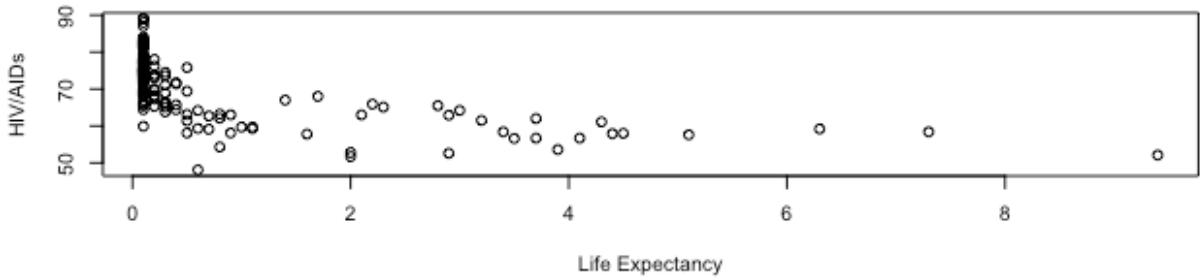
Now turning to EDA around Polio, Hepatitis.B, and Diphtheria features more closely, we notice that there are a group of countries far outside the normal range for each of these diseases.

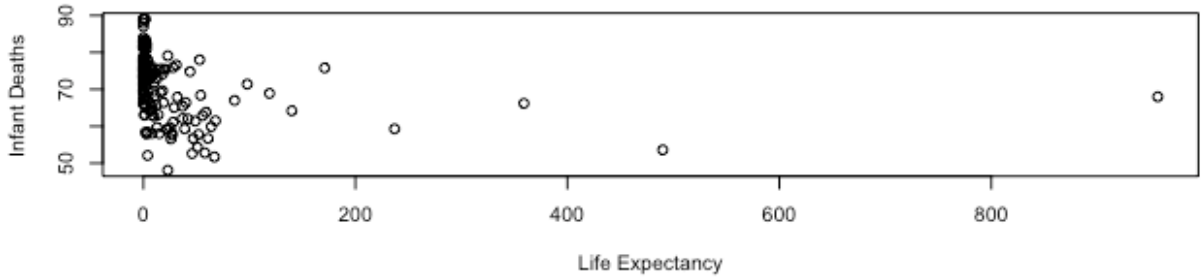


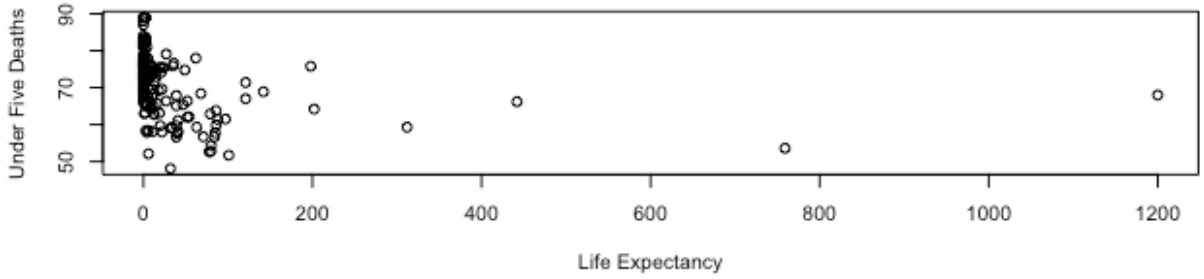
Excluding those points would give us a much more linear relationship, but since we have no valid reason to exclude them, we decided to truncate the value for all of these features at 70 instead (a value we found to optimize correlation with the Life Expectancy). The transformed disease features are more linear:



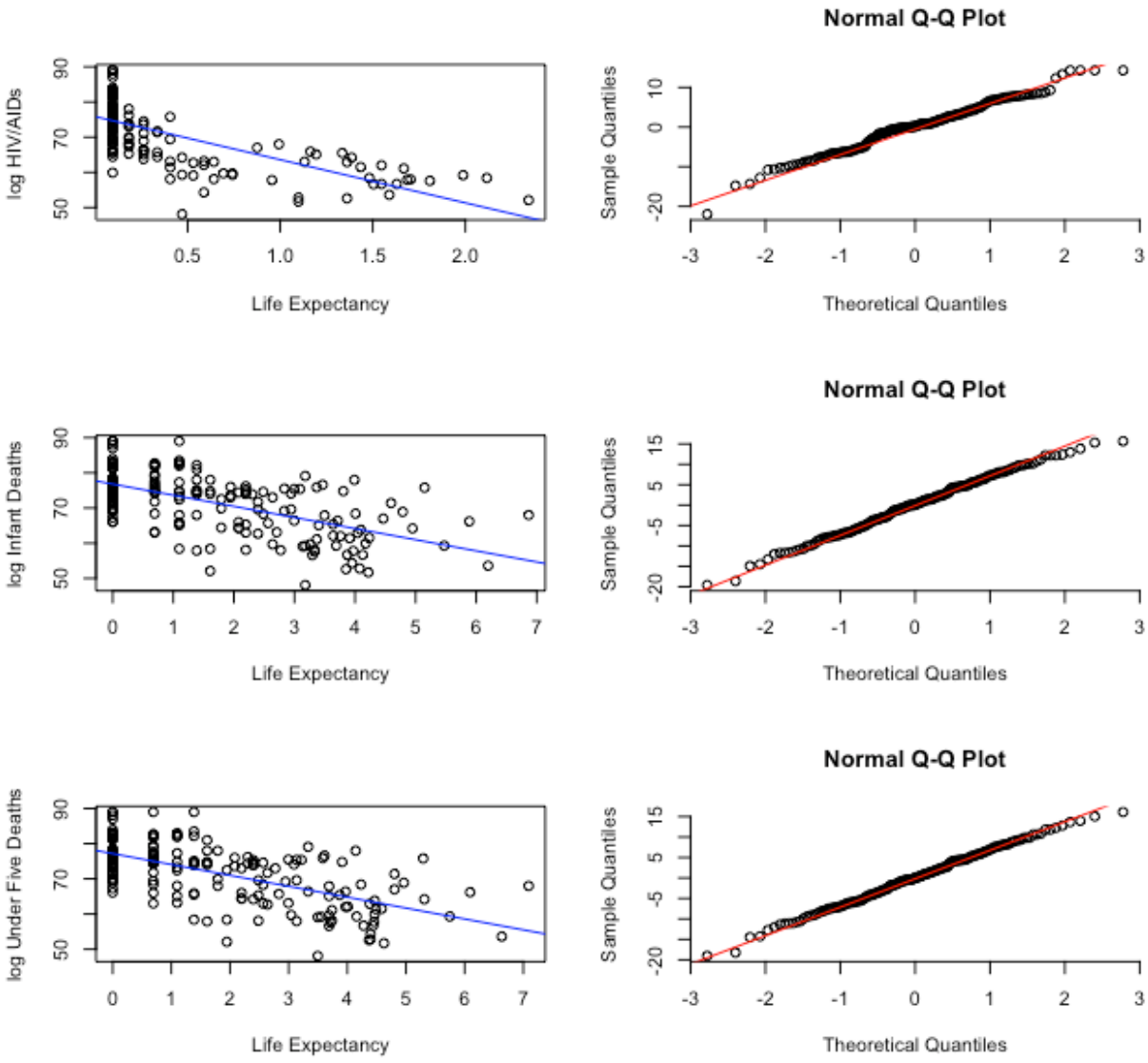
Likewise, we found that HIV/AIDs, Under 5 Deaths, and Infant mortality rate all showed a somewhat curved trend and have extreme values that would have an undue impact on linear regression.







Taking the log of these features reveals a more linear relationship and reduces the impact of the extreme points. There is still some divergence near the ends of the distributions shown by the normal QQ plots, but the residuals looks reasonably normal:



High Interpretability Model

To predict life expectancy, two models were built. One was to be highly interpretable and the second was designed to be highly predictable. To ensure that the difficulty of interpretability was kept at a minimum, a linear regression model was used for simplicity. The model resulted in a test ASE of 10.948 and an adj r2 of 0.795. The following four predictors were chosen for the model given their high correlation with life expectancy; 'Income.composition.of.resources', 'Adult.Mortality', 'under.five.deaths', and 'HIV.AIDS'. The intercept of the model (base life expectancy) was 58.6 years of age. The 95% confidence interval for the intercept was [51.55, 65.70] indicating that the model is 95% confident that the base life expectancy will fall within that range.

‘Income.composition.of.resources’ had a coefficient estimate of 19.982 and a p-value of 1.65e-05. These values suggest that the higher the human development index defined by the data, the higher the life expectancy will be. Based on the model, an increase of 1 point in the Income.composition.of.resources of a country correspond with an average increase of 19.9 years in life expectancy. The 95% confidence interval for the coefficients of this variable was [11.27, 28.69].

‘Adult.Mortality’ had a coefficient estimate of -0.02014 and a p-value of 0.000557. Based on the model, we expect an increase of 1 point in the Adult.Mortality to correspond to a 98.0% reduction in life expectancy (remembering that we log transformed this feature). Given that the coefficient is negative this would suggest that the mortality rates of adults has a negative impact on life expectancy. This notion makes logical sense when thinking about how if there is a higher adult mortality rate, the lower the average life expectancy will be. The 95% confidence interval for adult mortality was [-0.031, -0.009].

‘Under.five.deaths’ had a coefficient estimate of -0.7591 and a p-value of 0.01365. Based on the model, we expect an increase of 1 point in the under.five.deaths to correspond to a 53.2% reduction in life expectancy (remembering that we log transformed this feature). Similarly to adult mortality, deaths under 5 years old are negatively correlated with life expectancy. The 95% confidence interval for the coefficient of under.five.deaths was [-1.359, -0.160].

‘HIV.AIDS’ had a coefficient estimate of -2.410 and a p-value of 1.78e-06. Based on the model, we expect an increase of 1 point in HIV.AIDS of a country to correspond with a 91.1% reduction in life expectancy (remembering that we log transformed this feature). The 95% confidence interval for the coefficients for HIV/AIDSs was [-3.346, -1.474]. HIV/AIDS being negatively correlated with life expectancy makes sense given the severity of the disease.

Overall this model provides a high degree of interpretability but lacks a high degree of predictability. The next model will focus more on gaining predictability in the model but sacrificing interpretability.

```
Call:
lm(formula = Life.expectancy ~ Income.composition.of.resources +
    Adult.Mortality + under.five.deaths + HIV.AIDS, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5817  -2.2990   0.1076   2.3813   9.6707

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    58.626220   3.559662  16.470  < 2e-16 ***
Income.composition.of.resources 19.981865   4.385003   4.557 1.65e-05 ***
Adult.Mortality                -0.020142   0.005624  -3.581 0.000557 ***
under.five.deaths              -0.759101   0.301675  -2.516 0.013652 *
HIV.AIDS                       -2.410024   0.471124  -5.115 1.78e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.972 on 89 degrees of freedom
Multiple R-squared:  0.8037,    Adjusted R-squared:  0.7949
F-statistic: 91.08 on 4 and 89 DF,  p-value: < 2.2e-16
```


Variable	Coefficient	P-value
Intercept	58.626220	< 2e-16 ***
Income.composition.of.resources	19.981865	1.65e-05 ***
under.five.deaths	-0.759101	0.013652 *
Adult.Mortality	-0.020142	0.000557 ***
HIV.AIDS	-2.410024	1.78e-06 ***

High Predictability Model

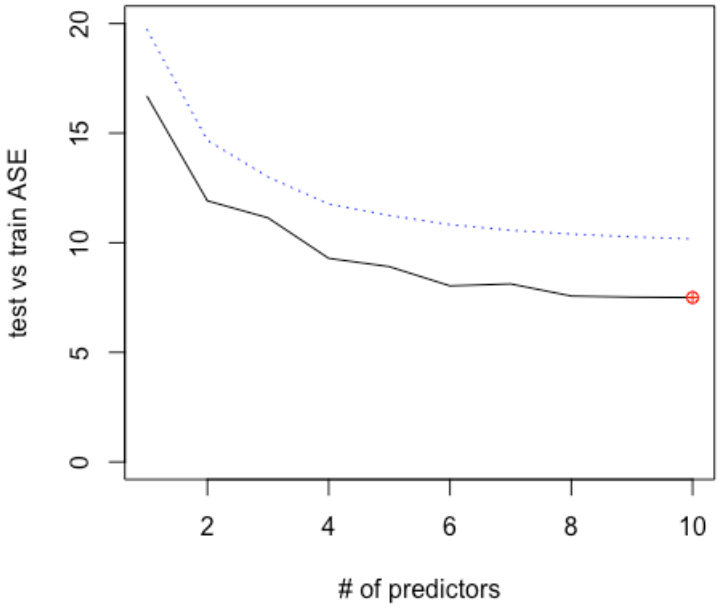
The second model was designed to be highly predictable in regards to life expectancy. The residual plots and assumptions are the same as the first model. A backwards model was used which resulted in a test ASE of 7.501 and AdjR2 of 0.712.

Compared to the highly interpretable model, there were 10 predictors used here and they provided a better prediction for life expectancy. The complexity brought on by adding predictors provided better results, but the drawback is the model became less interpretable.

The predictors used and their coefficients are below. Depending on the audience that the model is being presented to will determine if the highly interpretable or highly predictable model should be used. A third model was created that utilized a combination of log and quadratic terms which brought the test ASE down to below 0.004 and the AdjR2 to 0.83, but the model suffered from a high degree of over-fitting so it was not pursued any further.

Variable	Coefficient
Intercept	54.759
Income.Composition.of.resources	20.777
HIV.AIDS	-4.195
infant.deaths	-0.257
total.expenditure	0.264
adult.mortality	-0.017
alcohol	0.109
percent.expenditure	0.159
thinness.1.19.years	-0.093
Status	-1.616

DiphtheriaTr	0.085



Objective 2

Non-parametric Model

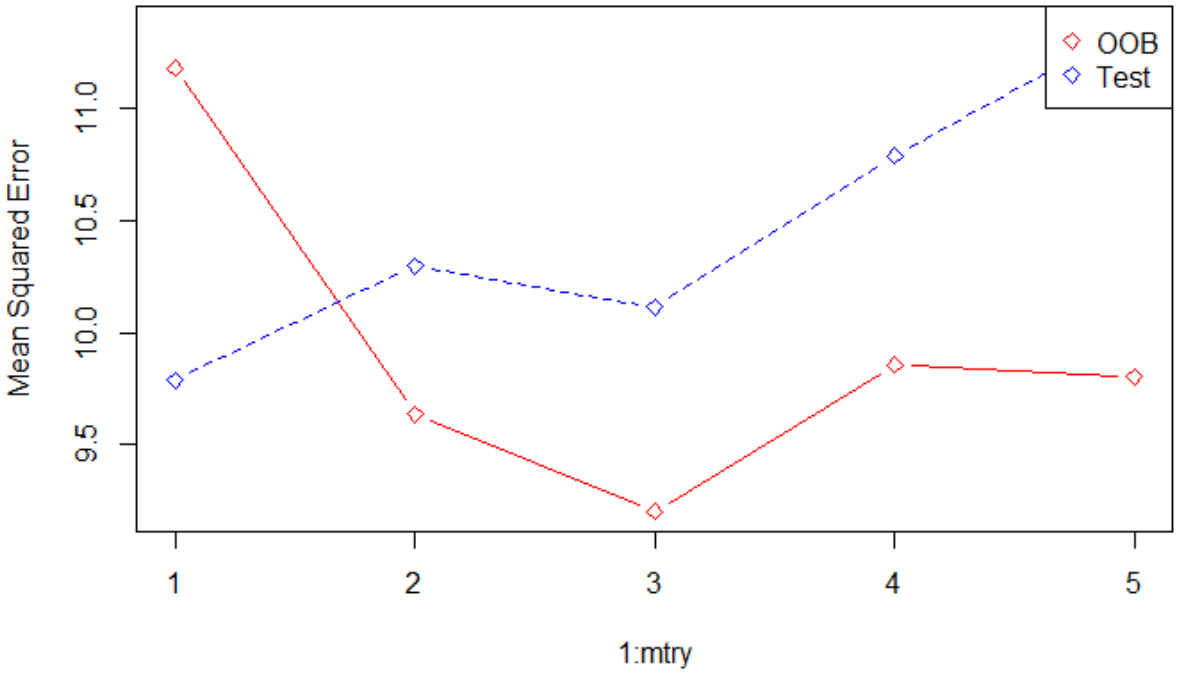
So far in our statistics classes we have primarily worked with parametric models such as regression. Parametric models require a specific set of assumptions about the data and are geared towards the creation of a predictive model which is highly interpretable. These models typically have a functional form i.e. $Y = b0 + b1*x1 + b2*x2$.

Non-parametric models are different from parametric models in that they do not require any assumptions about the data being used. This allows data scientists to be more flexible when working with data which may not fit a specific form. Non-parametric models are more capable of fitting complex patterns, but are not easily interpreted once created.

We are using random forests to predict life expectancy. Random forests are a form of decision trees with a similar methodology to bagging. Decision trees operate by taking the data and splitting each parameter along some boundary which gives the most information about the

value we are trying to predict and bagging is the method by which the parameters are selected and split. The parameter used to fit our random forest model is called mtry, this is the number of variables that are selected at each split of the trees.

Typically, parametric models are evaluated by their accuracy, predictor significance, and addressing assumptions about each predictor. Since no assumptions about the data are made for non-parametric models, we are only able to judge it based on its ability to accurately predict the response variable. Here, we use the test-set MSE and the out-of-bag MSE. We built 5 separate models using the predictors HIV.AIDS, Income.composition.of.resources, Adult.Mortality, under.five.deaths, and thinness.5.9.years and evaluated these errors in the chart below.



The model with an mtry of 3 performed the best with an MSE of 9 for the out-of-bag and an MSE of 10 for the test set. Compared to our earlier models with an MSE of 7.501 respectively, this model is slightly less capable of accurately predicting the life expectancy.

Unfortunately, this model provides limited ability for the data scientist to interpret each predictor's ability to give information about the response. These models also require large datasets, build more slowly, and are easier to overfit to the datasets. There exist some other non-parametric models, such as boosted trees, which are capable of more fine tuning allowing for lower MSE.

Appendix- R Code

```
---
title: "R Notebook"
output: html_notebook
editor_options:
  chunk_output_type: console
---
# Install packages & set seed
```{r}
library(GGally)
library(scales)
library(leaps)
library(lme4)
library(mlbench)
library(caret)
library(MASS)
library(randomForest)
library(arm)
library(glmnet)
library(imputeMissings)
library(ggvis)
library(mice)
library(ISLR)
library(dplyr)
library(naniar)

# set seed at 29
set.seed(29)
```
```

```

# Data for BMI EDA
```{r}
Life_Exp_Data = read.csv(file.choose(),header = TRUE)
summary(Life_Exp_Data$Life.expectancy)
summary(Life_Exp_Data_2014)

Life_Exp_Data_2014 = Life_Exp_Data %>% filter(Year == 2014)
summary(Life_Exp_Data_2014$Life.expectancy)
str(Life_Exp_Data)

Life_Exp_Data_2014$LogGDP = log(Life_Exp_Data_2014$GDP)
Life_Exp_Data_2014$Logthinness_5_19 = log(Life_Exp_Data_2014$GDP)
Life_Exp_Data_2014$LogBMI = log(Life_Exp_Data_2014$BMI)
Life_Exp_Data_2014$LogAdult.Mortality =
log(Life_Exp_Data_2014$Adult.Mortality)
Life_Exp_Data_2014$Loginfant.deaths =
log(Life_Exp_Data_2014$infant.deaths+1)
Life_Exp_Data_2014$LogAlcohol = log(Life_Exp_Data_2014$Alcohol)
Life_Exp_Data_2014$Logpercentage.expenditure =
log(Life_Exp_Data_2014$percentage.expenditure+1)
Life_Exp_Data_2014$LogHepatitis.B = log(Life_Exp_Data_2014$Hepatitis.B+1)
Life_Exp_Data_2014$LogMeasles = log(Life_Exp_Data_2014$Measles+1)
Life_Exp_Data_2014$Logunder.five.deaths =
log(Life_Exp_Data_2014$under.five.deaths+1)
Life_Exp_Data_2014$LogPolio = log(Life_Exp_Data_2014$Polio+1)
Life_Exp_Data_2014$LogTotal.expenditure =
log(Life_Exp_Data_2014$Total.expenditure+1)
Life_Exp_Data_2014$LogDiphtheria  = log(Life_Exp_Data_2014$Diphtheria+1)
Life_Exp_Data_2014$LogPopulation  = log(Life_Exp_Data_2014$Population+1)
Life_Exp_Data_2014$LogHIV.AIDS  = log(Life_Exp_Data_2014$HIV.AIDS+1)
Life_Exp_Data_2014$Logthinness..1.19.years  =
log(Life_Exp_Data_2014$thinness..1.19.years+1)
```
```

```
Life_Exp_Data_2014$Logthinness.5.9.years  =
log(Life_Exp_Data_2014$thinness.5.9.years+1)
Life_Exp_Data_2014$LogIncome.composition.of.resources  =
log(Life_Exp_Data_2014$Income.composition.of.resources+1)
Life_Exp_Data_2014$LogSchooling  = log(Life_Exp_Data_2014$Schooling+1)
Life_Exp_Data_2014$LogLife.expectancy  =
log(Life_Exp_Data_2014$Life.expectancy+1)
Life_Exp_Data_2014$LogHIV.AIDS_sqr  =
log(Life_Exp_Data_2014$HIV.AIDS+1)*log(Life_Exp_Data_2014$HIV.AIDS+1)
```

#filter & select
```{r}
Life_Exp_Data_2014 %>%
  select(BMI,
         Adult.Mortality,
         infant.deaths
         ) %>% ggpairs()  #aes(color = Life_Exp_Data$Status))

Life_Exp_Data_2014 %>%
  select(LogBMI,
         LogAdult.Mortality,
         Loginfant.deaths
  ) %>% ggpairs()  #aes(color = Life_Exp_Data$Status))


Life_Exp_Data_2014 %>%
  select(
         Alcohol,
         percentage.expenditure,
         Hepatitis.B,
         Measles,
         BMI,
         under.five.deaths) %>%
```

```
  ggpairs()

str(Life_Exp_Data_2014$Hepatitis.B)
  Life_Exp_Data_2014 %>% filter(Life_Exp_Data_2014$Hepatitis.B == 90)
```

#with Transformed data
```{r}
Life_Exp_Data_2014 %>%
  select(
    LogAlcohol,
    Logpercentage.expenditure,
   LogHepatitis.B,
    LogMeasles,
    LogBMI,
    Logunder.five.deaths) %>% ggpairs()

summary(Life_Exp_Data_2014$HIV.AIDS)
Life_Exp_Data_2014 %>%
    select(
        Polio,
        Total.expenditure,
        Diphtheria,
        HIV.AIDS) %>%
  ggpairs()

summary(Life_Exp_Data_2014$HIV.AIDS)
Life_Exp_Data_2014 %>%
  select(
    LogPolio,
    LogTotal.expenditure,
    LogDiphtheria,
    LogHIV.AIDS) %>%
  ggpairs()
```

```
Life_Exp_Data_2014 %>%
  select(
GDP,
Population,
thinness..1.19.years,
thinness.5.9.years,
Income.composition.of.resources,
Schooling) %>% ggpairs()

Life_Exp_Data_2014 %>%
  select (LogLife.expectancy,
          LogBMI,
          LogAdult.Mortality,
          Loginfant.deaths,
    LogGDP,
    LogPopulation,
    Logthinness..1.19.years,
    Logthinness.5.9.years,
    LogIncome.composition.of.resources,
    LogSchooling,
    LogPolio,
    LogTotal.expenditure,
    LogDiphtheria,
    LogHIV.AIDS) %>% ggpairs()

```

#life expectancy
```{r}
# Histogram with density plot
ggplot(Life_Exp_Data_2014, aes(x=Life.expectancy)) +
  geom_histogram(aes(y=..density..), colour="black", fill="blue")+
  geom_density(alpha=.2, fill="#FF6666") +
```

```
  geom_vline(aes(xintercept=mean(Life.expectancy)),
        color="red", linetype="dashed", size=1) + labs(y = "Density") + labs(x = "Life
Expectancy") +
  ggtitle("Checking Normaily of Life Expectancy")

#Missing Values
gg_miss_var(Life_Exp_Data_2014) + labs(y = "Count") +
  ggtitle("Variables Containing Missing Values")
```

#blocking
```{r}
Life_Exp_Data_2014 %>%
  select (LogLife.expectancy,
    LogAdult.Mortality,
    Loginfant.deaths,
LogAlcohol,
Logpercentage.expenditure,
LogHepatitis.B,
LogMeasles,
LogBMI,
Logunder.five.deaths,
LogIncome.composition.of.resources,
LogSchooling,
LogPolio,
LogTotal.expenditure,
LogDiphtheria,
LogHIV.AIDS) %>% ggpairs()

Life_Exp_Data_2014 %>%
  select (LogLife.expectancy,
        LogBMI,
        LogAdult.Mortality, #blocking in groups
        Loginfant.deaths,
```

```
        LogGDP,
        LogPopulation,
        Logthinness..1.19.years,
        Logthinness.5.9.years,
        LogIncome.composition.of.resources,
        LogSchooling,
        LogPolio,
        LogTotal.expenditure,
        LogDiphtheria,
        LogHIV.AIDS_sqr) %>% ggpairs()
```

#BMI EDA
```{r}
summary(Life_Exp_Data_2014$BMI)

Life_Exp_Data_2014 %>%
  select (BMI
        ) %>%
ggplot(aes(BMI)) + geom_histogram(color = "blue", fill = "blue") +
  ggtitle("Histogram of BMI") +
  xlab("BMI") + ylab("Count")

Life_Exp_Data_2014 %>%
  select (LogBMI
  ) %>%
  ggplot() + geom_qq(aes(sample = LogBMI)) +
  ggtitle("qq Plot of BMI")
```

#filtered looks at qq plot
```{r}
Life_Exp_Data_2014 %>% filter(LogBMI < 2) %>%
  select (BMI
```

```
  ) %>%
  ggplot() + geom_qq(aes(sample = BMI)) +
  ggtitle("qq Plot of BMI < 2.5")

Life_Exp_Data_2014 %>% filter(LogBMI < 3, LogBMI > 2) %>%
  select (BMI
  ) %>%
  ggplot() + geom_qq(aes(sample = BMI)) +
  ggtitle("qq Plot of BMI Between 2 and 3")

Life_Exp_Data_2014 %>% filter(LogBMI < 4, LogBMI > 3) %>%
  ggplot() + geom_qq(aes(sample = LogBMI)) +
  ggtitle("qq Plot of BMI Between 3 and 4")

Life_Exp_Data_2014 %>% filter(LogBMI >4) %>%
  select (LogBMI
  ) %>%
  ggplot() + geom_qq(aes(sample = LogBMI))+
  ggtitle("qq Plot of BMI > 4")
```

##Compared to Life Expectancy
```{r}
Life_Exp_Data_2014 %>% #filter(LogBMI < 2.5) %>%
  select (LogBMI, LogLife.expectancy
  ) %>%
  ggplot(aes(LogBMI, LogLife.expectancy)) + geom_point()+
  ggtitle("Scatter Plot of Log BMI")

Life_Exp_Data_2014 %>% #filter(LogBMI < 2.5) %>%
  select (LogBMI, LogLife.expectancy, Status
  ) %>%
  ggplot(aes(LogBMI, LogLife.expectancy)) +
  geom_jitter(height = 1, width = 1) +
```

```
  geom_smooth(method = lm, height = 2, width = 2) +
  facet_wrap(~Status) + geom_smooth(method = lm)+
  ggtitle("Comparing Regression Line of Developed vs Developing")

Life_Exp_Data_2014 %>% filter(Status == 'Developed') %>%
  select (LogBMI, LogLife.expectancy, Status) %>%
  ggplot() + geom_qq(aes(sample = LogBMI))

Life_Exp_Data_2014 %>% filter(Status == 'Developing') %>%
  select (LogBMI, LogLife.expectancy, Status) %>%
  ggplot() + geom_qq(aes(sample = LogBMI))
```

###bin BMI
```{r}
library(naniar)
vis_miss(Life_Exp_Data_2014)

breaks <- c(0,2,3,4,5)
tags <- c("<2", "2-3", "3-4", ">4")
Life_Exp_Data_2014$binLogBMI = cut(Life_Exp_Data_2014$LogBMI,
breaks=breaks, labels=tags)
summary(Life_Exp_Data_2014$binLogBMI)

Life_Exp_Data_2014 %>% filter(!is.na(binLogBMI)) %>%
  ggplot(aes(x = LogLife.expectancy, y = LogBMI, color = binLogBMI, fill =
binLogBMI)) +
  geom_jitter() + facet_wrap(~Status)

Life_Exp_Data_2014 %>% #filter(LogBMI < 2.5) %>%
  select (LogBMI, LogLife.expectancy, Status, binLogBMI) %>%
  ggplot(aes(LogBMI, LogLife.expectancy, col = binLogBMI)) +
  geom_jitter(height = 1, width = 1) +
  geom_smooth(method = lm) +
```

```
  facet_wrap(~Status)

Life_Exp_Data_2014 %>% #filter(!is.na(LogBMI)) %>%
  select (binLogBMI, LogLife.expectancy, LogBMI, Status) %>%
  ggplot() + geom_qq(aes(sample = LogBMI)) +
  facet_wrap(~binLogBMI)

BMI_L = Life_Exp_Data_2014 %>%
   filter(binLogBMI == "<2" | binLogBMI ==  "2-3"  |
          binLogBMI ==  "3-4" )

BMI_H = Life_Exp_Data_2014 %>%
  filter(binLogBMI == ">4" )
```

##t-test
```{r}
t.test(BMI_L$LogLife.expectancy,BMI_H$LogLife.expectancy)

cor(BMI_L$LogBMI, BMI_L$LogLife.expectancy, method = c("pearson", "kendall",
"spearman"))
cor.test(BMI_L$LogBMI, BMI_L$LogLife.expectancy, method=c("pearson",
"kendall", "spearman"))

cor(BMI_H$LogBMI, BMI_H$LogLife.expectancy, method = c("pearson", "kendall",
"spearman"))
cor.test(BMI_H$LogBMI, BMI_H$LogLife.expectancy, method=c("pearson",
"kendall", "spearman"))
```

Load in data and view. Subset to 2014 Only
```{r}
# read in data
```

```
loadin <-
read.csv("/Users/indy/Documents/MSDS/MSDS6372/WHO_project/datasets-12603-17232-Life Expectancy Data.csv", header = TRUE)

# Subset data
data <- subset(loadin, loadin$Year==2014)

# Drop the Year column
data <- data[-c(2)]

# view class of data
str(data)

# Count number of missing values
na_col <- sapply(data, function(x)sum(is.na(x)))
na_col

completerow <- sum(complete.cases(data))
completerow
```

Convert variables to numeric
```{r}
data$Status <- as.numeric(data$Status)
data$Adult.Mortality <- as.numeric(data$Adult.Mortality)
data$infant.deaths <- as.numeric(data$infant.deaths)
data$Hepatitis.B <- as.numeric(data$Hepatitis.B)
data$Measles <- as.numeric(data$Measles)
data$under.five.deaths <- as.numeric(data$under.five.deaths)
data$Polio <- as.numeric(data$Polio)
data$Diphtheria <- as.numeric(data$Diphtheria)

# verify all are numeric
str(data)
```

```

# GGPairs on data
```{r}
ggpairs(data[-1]) # dropped country column since it's not numeric
```

Diseases EDA
```{r}
par(mfrow=c(3,1))
plot(data$Hepatitis.B, data$Life.expectancy, xlab = 'Life Expectancy', ylab=
'Hepatitis B')
plot(data$Polio, data$Life.expectancy, xlab = 'Life Expectancy', ylab= 'Polio')
plot(data$Diphtheria, data$Life.expectancy, xlab = 'Life Expectancy', ylab=
'Diphtheria')
```

Log transforms needed
```{r}
par(mfrow=c(3,1))
plot(data$HIV.AIDS, data$Life.expectancy, xlab = 'Life Expectancy', ylab=
'HIV/AIDs')
plot(data$infant.deaths, data$Life.expectancy, xlab = 'Life Expectancy', ylab=
'Infant Deaths')
plot(data$under.five.deaths, data$Life.expectancy, xlab = 'Life Expectancy', ylab=
'Under Five Deaths')
```

Transform variables
```{r}
edadata = data
edadata$logHIV.AIDS <- log(edadata$HIV.AIDS+1)
edadata$loginfant.deaths <- log(edadata$infant.deaths+1)
edadata$logunder.five.deaths <- log(edadata$under.five.deaths+1)
```
```

```
edadata$PolioTr = edadata$Polio
edadata$PolioTr[edadata$Polio < 70] = 70.0
edadata$DiphtheriaTr = edadata$Diphtheria
edadata$DiphtheriaTr[edadata$Diphtheria < 70] = 70.0
edadata$Hepatitis.BTr = edadata$Hepatitis.B
edadata$Hepatitis.BTr[edadata$Hepatitis.B < 70] = 70.0
```

Plot transformed diseases

```{r}
par(mfrow=c(3,2))
plot(edadata$Hepatitis.BTr, edadata$Life.expectancy, xlab = 'Life Expectancy',
ylab= 'Hepatitis B')
reg = lm(Life.expectancy ~ Hepatitis.BTr, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')

plot(edadata$PolioTr, edadata$Life.expectancy, xlab = 'Life Expectancy', ylab=
'Polio')
reg = lm(Life.expectancy ~ PolioTr, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')

plot(edadata$DiphtheriaTr, edadata$Life.expectancy, xlab = 'Life Expectancy',
ylab= 'Diphtheria')
reg = lm(Life.expectancy ~ DiphtheriaTr, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')
```

plot log transformed features
```{r}
par(mfrow=c(3,2))
plot(edadata$logHIV.AIDS, edadata$Life.expectancy, xlab = 'Life Expectancy',
ylab= 'log HIV/AIDs')
reg = lm(Life.expectancy ~ logHIV.AIDS, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')

plot(edadata$loginfant.deaths, edadata$Life.expectancy, xlab = 'Life Expectancy',
ylab= 'log Infant Deaths')
reg = lm(Life.expectancy ~ loginfant.deaths, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')

plot(edadata$logunder.five.deaths, edadata$Life.expectancy, xlab = 'Life
Expectancy', ylab= 'log Under Five Deaths')
reg = lm(Life.expectancy ~ logunder.five.deaths, data = edadata)
abline(reg, col="blue")
qqnorm(reg$res, pch = 1, frame = FALSE)
qqline(reg$res, col='red')
```

Check correlations
```{r}
transformed_feats = edadata %>% select(Life.expectancy, Hepatitis.BTr, PolioTr,
DiphtheriaTr, Hepatitis.B, Polio, Diphtheria, HIV.AIDS, infant.deaths, logHIV.AIDS,
loginfant.deaths)
cor(na.omit(transformed_feats))
```

Impute missing values with mice package (not using life expectancy)
```{r}
labels = as.data.frame(select(data, 'Life.expectancy'))
feats = select(data, -one_of(c('Life.expectancy')))

tempdata <- mice(feats, m=1, maxit=1, method="cart", seed=29)
summary(tempdata)

feats_mice <- complete(tempdata,1)
vis_miss(feats_mice)

feats_mice$Life.expectancy = labels$Life.expectancy
datamice = feats_mice
```

Transform variables
```{r}
datamice$HIV.AIDS <- log(datamice$HIV.AIDS+1)
datamice$infant.deaths <- log(datamice$infant.deaths+1)
datamice$under.five.deaths <- log(datamice$under.five.deaths+1)
datamice$percentage.expenditure <- log(datamice$percentage.expenditure+1)

datamice$PolioTr = datamice$Polio
datamice$PolioTr[datamice$Polio < 70] = 70.0
datamice$DiphtheriaTr = datamice$Diphtheria
datamice$DiphtheriaTr[datamice$Diphtheria < 70] = 70.0
datamice$Hepatitis.BTr = datamice$Hepatitis.B
datamice$Hepatitis.BTr[datamice$Hepatitis.B < 70] = 70.0
```

Simple linear regression model
```{r}
```

```
index<-sample(1:dim(feats_mice)[1],94,replace=F)
train<-feats_mice[index,]
test<-feats_mice[-index,]
lm_min = lm(
  Life.expectancy ~ Income.composition.of.resources + Adult.Mortality + Schooling
,
  data = train
)
testMSE<-mean((test$Life.expectancy - life_exp_pred)^2)
testMSE
summary(lm_min)
```

Forward Model & Summary

```{r, echo=T, fig.height=3,fig.width=7}

# Forward Model
fwddata <- subset(datamice, select=-c(Country,Polio,Diphtheria,Hepatitis.B))
reg.fwd=regsubsets(Life.expectancy~.,data=fwddata,method="forward",nvmax=1
9)

# Summary Stats
summary(reg.fwd)$adjr2
summary(reg.fwd)$rss
summary(reg.fwd)$bic
coef(reg.fwd,19)
summary(reg.fwd)

par(mfrow=c(1,3))
bics<-summary(reg.fwd)$bic
plot(1:19,bics,type="l",ylab="BIC",xlab="# of predictors")
index<-which(bics==min(bics))
points(index,bics[index],col="red",pch=10)
```

```
adjr2<-summary(reg.fwd)$adjr2
plot(1:19,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of predictors")
index<-which(adjr2==max(adjr2))
points(index,adjr2[index],col="red",pch=10)

rss<-summary(reg.fwd)$rss
plot(1:19,rss,type="l",ylab="train RSS",xlab="# of predictors")
index<-which(rss==min(rss))
points(index,rss[index],col="red",pch=10)

# 10 predictors looks to provide the best adjusted r squared value
fwd.reg.best = regsubsets(Life.expectancy~., data = fwddata, method = "forward",
nvmax = 10)

# Coefficients & Summary
coef(fwd.reg.best, 10)
summary(fwd.reg.best)$adjr2
summary(fwd.reg.best)$rss
summary(fwd.reg.best)$bic
```

Forward Model ASE PLot
```{r}
index<-sample(1:dim(fwddata)[1],94,replace=F)
train<-fwddata[index,]
test<-fwddata[-index,]
reg.fwd = regsubsets(Life.expectancy~., data = fwddata, method = "forward",
nvmax = 10)

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()
#note my index is to 10 since that what I set it in regsubsets
for (i in 1:10){
  predictions<-predict.regsubsets(object=reg.fwd,newdata=test,id=i)
  testASE[i]<-mean((test$Life.expectancy-predictions)^2)
}
par(mfrow=c(1,1))
plot(1:10,testASE,type="l",xlab="# of predictors",ylab="test vs train
ASE",ylim=c(0,20))
index<-which(testASE==min(testASE))
points(index,testASE[index],col="red",pch=10)
rss<-summary(reg.fwd)$rss
lines(1:10,rss/183,lty=3,col="blue")  #Dividing by 183 since ASE=RSS/sample size
```

Backward Model & Summary
```{r, echo=T, fig.height=3,fig.width=7}

# Backward Model
bckdata <- subset(datamice, select=-c(Country,Polio,Diphtheria,Hepatitis.B))
reg.bck=regsubsets(Life.expectancy~.,data=bckdata,method="backward",nvmax=
19)

# Summary Stats
summary(reg.bck)$adjr2
summary(reg.bck)$rss
summary(reg.bck)$bic
coef(reg.bck,19)
summary(reg.bck)
```

```
par(mfrow=c(1,3))
bics<-summary(reg.bck)$bic
plot(1:19,bics,type="l",ylab="BIC",xlab="# of predictors")
index<-which(bics==min(bics))
points(index,bics[index],col="red",pch=10)

adjr2<-summary(reg.bck)$adjr2
plot(1:19,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of predictors")
index<-which(adjr2==max(adjr2))
points(index,adjr2[index],col="red",pch=10)

rss<-summary(reg.bck)$rss
plot(1:19,rss,type="l",ylab="train RSS",xlab="# of predictors")
index<-which(rss==min(rss))
points(index,rss[index],col="red",pch=10)

# 10 predictors looks to provide the best adjusted r squared value
bck.reg.best = regsubsets(Life.expectancy~., data = bckdata, method =
"backward", nvmax = 10)

# Coefficients & Summary
coef(bck.reg.best, 10)
summary(bck.reg.best)$adjr2
summary(bck.reg.best)$rss
summary(bck.reg.best)$bic
```

Backward Model ASE PLot
```{r}
index<-sample(1:dim(bckdata)[1],94,replace=F)
train<-bckdata[index,]
test<-bckdata[-index,]
reg.bck = regsubsets(Life.expectancy~., data = bckdata, method = "backward",
nvmax = 10)
```

```
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()
#note my index is to 10 since that what I set it in regsubsets
for (i in 1:10){
  predictions<-predict.regsubsets(object=reg.bck,newdata=test,id=i)
  testASE[i]<-mean((test$Life.expectancy-predictions)^2)
}
par(mfrow=c(1,1))
plot(1:10,testASE,type="l",xlab="# of predictors",ylab="test vs train
ASE",ylim=c(0,20))
index<-which(testASE==min(testASE))
points(index,testASE[index],col="red",pch=10)
rss<-summary(reg.bck)$rss
lines(1:10,rss/183,lty=3,col="blue")  #Dividing by 183 since ASE=RSS/sample size
```

Stepwise model & Summary
```{r, echo=T, fig.height=3,fig.width=7}
# Stepwise Model
stpdata <- subset(datamice, select=-c(Country,Polio,Diphtheria,Hepatitis.B))
reg.stp=regsubsets(Life.expectancy~.,data=stpdata,method="seqrep",nvmax=19)

# Summary Stats
summary(reg.stp)$adjr2
summary(reg.stp)$rss
summary(reg.stp)$bic
```

```
coef(reg.stp,19)
summary(reg.stp)

par(mfrow=c(1,3))
bics<-summary(reg.stp)$bic
plot(1:19,bics,type="l",ylab="BIC",xlab="# of predictors")
index<-which(bics==min(bics))
points(index,bics[index],col="red",pch=10)

adjr2<-summary(reg.stp)$adjr2
plot(1:19,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of predictors")
index<-which(adjr2==max(adjr2))
points(index,adjr2[index],col="red",pch=10)

rss<-summary(reg.stp)$rss
plot(1:19,rss,type="l",ylab="train RSS",xlab="# of predictors")
index<-which(rss==min(rss))
points(index,rss[index],col="red",pch=10)

# 11 predictors looks to provide the best adjusted r squared value
stp.reg.best = regsubsets(Life.expectancy~., data = bckdata, method = "seqrep",
nvmax = 11)

# Coefficients & Summary
coef(stp.reg.best, 11)
summary(stp.reg.best)$adjr2
summary(stp.reg.best)$rss
summary(stp.reg.best)$bic
```

Stepwise Model ASE PLot

```{r}
index<-sample(1:dim(stpdata)[1],94,replace=F)
```

```
train<-stpdata[index,]
test<-stpdata[-index,]
reg.stp = regsubsets(Life.expectancy~., data = stpdata, method = "seqrep",nvmax
= 11)

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()
#note my index is to 11 since that what I set it in regsubsets
for (i in 1:11){
  predictions<-predict.regsubsets(object=reg.stp,newdata=test,id=i)
  testASE[i]<-mean((test$Life.expectancy-predictions)^2)
}
par(mfrow=c(1,1))
plot(1:11,testASE,type="l",xlab="# of predictors",ylab="test vs train
ASE",ylim=c(0,25))
index<-which(testASE==min(testASE))
points(index,testASE[index],col="red",pch=10)
rss<-summary(reg.stp)$rss
lines(1:11,rss/183,lty=3,col="blue")  # Dividing by 183 since ASE=RSS/sample size
```

Lasso from class
```{r}

set.seed(29)

par(mfrow=c(1,1))
```

```
datalasso <- subset(datamice, select=-c(Country,Polio,Diphtheria,Hepatitis.B))

# setup lasso
x_vars <- model.matrix(Life.expectancy~. ,datalasso)
y_var <- datalasso$Life.expectancy
lambda_seq <- 10^seq(2, -2, by = -.1)

# Splitting the data into test and train
train = sample(1:nrow(x_vars), nrow(x_vars)/2)
x_test = (-train)
y_test = y_var[x_test]

# cross validation output
cv_output <- cv.glmnet(x_vars[train,], y_var[train],
                       alpha = 1, lambda = lambda_seq,
                       nfolds = 5)
plot(cv_output,xvar="lambda",label=TRUE)

# identifying best lamda
best_lam <- cv_output$lambda.min
best_lam

# Rebuilding the model with best lamda value identified
lasso_best <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = best_lam)
lasso_best

# R squared of model
lasso_best$dev.ratio

# Looking at Coefficients
coef(lasso_best)

# predict values
```

```
pred <- predict(lasso_best, s = best_lam, newx = x_vars[x_test,])

testMSE_LASSO2<-mean((y_test-pred)^2)
testMSE_LASSO2

# final model
#final <- cbind(y_var[test], pred)

# Checking the first six obs
#head(final)
```

Lasso Option 2 from internet
```{r}
set.seed(29)

par(mfrow=c(1,1))

datalasso <- subset(datamice, select=-c(Country,Polio,Diphtheria,Hepatitis.B))

Split <- floor(0.50 * nrow(datalasso))
train_ind <- sample(seq_len(nrow(datalasso)), size = Split)
train <- datalasso[train_ind,]
test <- datalasso[-train_ind,]

x=model.matrix(Life.expectancy~.,train)
y=train$Life.expectancy

xtest<-model.matrix(Life.expectancy~.,test)
ytest<-test$Life.expectancy


grid=10^seq(10,-2, length =100)
lasso.mod=glmnet(x,y,alpha=1, lambda =grid)
```

```
cv.out=cv.glmnet(x,y,alpha=1) #alpha=1 performs LASSO
plot(cv.out)
bestlambda<-cv.out$lambda.min  #Optimal penalty parameter.  You can make this call visually.
bestlambda
lasso.pred=predict (lasso.mod ,s=bestlambda ,newx=xtest)

# Mean squared error
testMSE_LASSO<-mean((ytest-lasso.pred)^2)
testMSE_LASSO

#  R Squared
lasso.mod$dev.ratio

# coeffiecients
coef(cv.out)
```

random forest

```{r}
# income.composition.of.resources, HIV.AIDS, Adult.Mortality, Under.Five.Deaths & Thinness.5.9.years
life_exp <- datamice %>% dplyr::select("Life.expectancy", "HIV.AIDS", "Income.composition.of.resources", "Adult.Mortality", "under.five.deaths", "thinness.5.9.years")

set.seed(101)
train = sample(1:nrow(life_exp), 100)

life_exp.rf = randomForest(Life.expectancy~., data = life_exp, subset = train)
life_exp.rf
```

```
#iterate and create many trees, then compare the MSE
var_count = 5
oob.err = double(var_count)
test.err = double(var_count)
for(mtry in 1:var_count){
  fit = randomForest(Life.expectancy~., data = life_exp, subset=train, mtry=mtry,
ntree = 350)
  oob.err[mtry] = fit$mse[350]
  pred = predict(fit, life_exp[-train,])
  test.err[mtry] = with(life_exp[-train,], mean( (Life.expectancy-pred)^2 ))
}
```

#plot random forest params
```{r}
matplot(1:mtry, cbind(test.err, oob.err), pch = 23, col = c("red", "blue"), type =
"b", ylab="Mean Squared Error")
legend("topright", legend = c("OOB", "Test"), pch = 23, col = c("red", "blue"))
```