# Introduction

The following data (data1.csv) was given to us to understand which drivers/predictors can be used to estimate the retail price of a vehicle.

The following data from the World Health Organization (WHO) Life Expectancy data that is posted on Kaggle.com for the purpose of health data analysis. Our team was tasked to identify any relationships between life expectancy for the countries included. Since this is an observational study our conclusions are limited to the data included in our analysis.

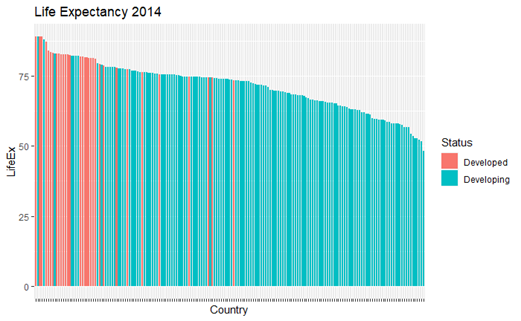
# Data Description

The original data set includes health and economic factors for 193 countries from 2000-2015. By our advisor’s instruction we limited our initial analysis to just the data from 2014. The health factors were collected from the WHO data repository and appended with economic factors from the United Nations website. A full list of the parameters included in the data are listed in Appendix 0.1.

# Exploratory Data Analysis (EDA)

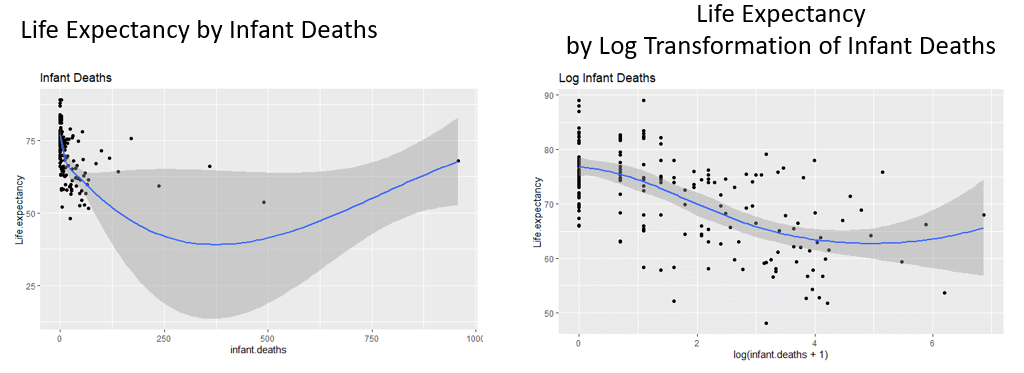
One of the first steps considered was: how are we going to treat the missing data? The missing data consisted of: 27 GDP, 41 population, and 10 schooling records (Appendix 0.2). Further, there were 27 percent expenditures that were set at zero which seemed strange as there were some prominent countries with zeros. Countries such as The United States, UK, were among the most noticeable.

Through research at World Bank, The UN Development report, and UNESCO we were able to find most of the missing values. Not all records could be captured in just based on the WHO’s data resources (shown in Appendix 0.3). This data we performed an amputation technique on 30 records by replacing the empty cells with the variable’s mean. In our validation, this technique did not change the distribution or overall variable mean (we only saw a 0-2% change in mean with amputation). Of those that were imputed, only 2 - Income Composition of Resources and Thinness 1-19, would eventually make it into the final models. Schooling’s factorized version would also make it into the models. A full list of the parameters included in the data are listed in Appendix 0.3.

After data cleaning we looked at how Life expectancy is distributed by country and status. The following column chart shows that life expectancy ranges from nearly 90 years of age (Belgium, Finland, Germany, Portugal) to just under 50 years of age (Sierra Leone). It is interesting to note that Finland is considered a developing country, as are a few other countries I would not expect such as: France, Israel, Canada, Greece, and South Korea. We can see a clear trend that developed countries tend to have higher life expectancy. This is more clearly shown in the Boxplot in Appendix 0.4.

To sort through many variables and their relationship with life expectancy, and each other quickly, we made a series of correlation charts (Appendix 0.5-0.8). In Appendix 0.5, the thinness categories are highly collinear, GDP and Percent expenditure are quite close, and infant deaths are perfectly correlated with under five deaths (which should be expected, one is a subset of the other).

To choose between the highly correlated variables, we looked at which one was most highly correlated with life expectancy and went with the best correlations. During inspection it was noticed that some of these variables have some odd relationships with life expectancy that did not make practical sense, specifically – GDP, infant deaths, under five deaths, HIV, and the Thinness categories. For example, the negative relationship between infant deaths and life expectancy seemed to be heavily influenced by scaling and a possible outlier. Once log transformed, the data resembled more of a random cloud than cluster, so we included these log transformed parameters as additions to the variable selection.



In Appendix 0.5 (on right), we can see that the logged version of the variable is more highly correlated with life expectancy than the unlogged version every time, with the most drastic difference coming from the child mortality variables. Since some of these final categories were the ones that had values researched and added in as well as imputed, we added categorical versions of HepB, Population, Schooling, Income Comp, and GDP. Boxplots of these factorized versions can be viewed in the appendix (boxplots below). Of the variables we factorized, only SchoolFactor would show up in the models, distribution of scales shown in Appendix 0.8.

The Cook’s D outlier analysis, for a linear regression with all parameters included, also pointed out that there may be a specific outlier influencing the data (see Appendix 0.9). However, what we found as we iterated on the different versions of linear models there were different countries that acted as influential points leaving our team to conclude that no data be removed from the training or test data sets.

# Objective 1: Linear Regression Methodology & Analysis

The World Health Organization requested independent analysis consulting groups to identify key relationships between immunizations, overall health metrics, and other economic factors that may influence a country’s life expectancy.

## Checking Assumptions

To verify the assumptions of the MLR model, we check if residuals are normally distributed, which we see is the case seen in the histogram (Appendix 1.0-1.2).

The next assumption for MLR that must be checked is the assumption of constant variance. That is to say that the variance of the residuals does not depend on the predicted value.

The model also assumes homoscedasticity. That is to say that the variance of error terms are similar for all values of the independent variables. The standardized residuals vs the predicted values shows equal distribution for all values of the independent variables.

Lastly, we assume for this data set the observations are independent of one another since the data description on Kaggle is very limited.

## Variable Selection

The best predictor variables for Model were selected using the correlation matrix (Appendix 0.7) correlation plot (Appendix 0.6) and multicollinearity plot (Appendix 0.05). The correlation matrix shows that School Factor, HIV/AID, Adult Mortality, Log Infant Death, Income Composition of resources, and Status are all highly correlated with life expectancy.

Variables that were highly correlated with each other, high multicollinearity, were not included in the model. For example, “infant deaths” and “under 5 deaths” were highly correlated and would be redundant to include both parameters in Model 1.

## Compare Competing Models

Model 1 – Simple

Model 2 – Complex

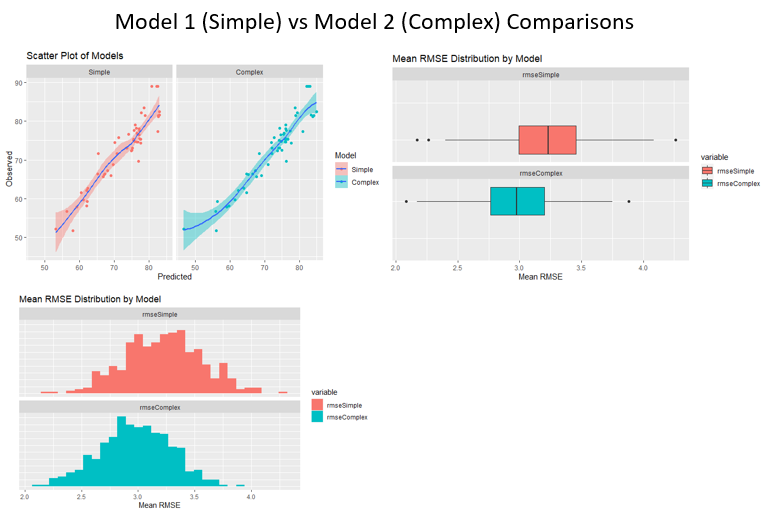
When reviewing the plots for Model 1, we were satisfied with the normality and distribution of the residuals. Reviewing this set of plots proved to be difficult because we were looking for the model that will do well over any train/test split, not just this one sampling instance. It turned out models with high bias or high variation did poorly when running them many times. It became a balance of determining the right amount of bias and variation that would do well over many runs. Due to this some of the fit metrics for this particular run may look less than optimal, but when hundreds of iterations are done the model performs better than other simple versions tried. The addition of income comp and status to the model helps smooth out the fit statistics between runs. Otherwise there is a risk of large variation due to the changing sets. Though we do see some high leverage points and a single high influence point, this is again due to this one run. The outlier (or few) generally change with every run. Due to this it did not seem prudent to start trimming outliers out of the set. Plots can be viewed in Appendix 1.1.

The fit in Model 2 also looks appealing (see Appendix 1.2). There is one high influence point in the leverage diagnostic, but as stated earlier we do not plan on removing any outliers based on the variety shown in each iteration. Cook’s D flagged a troubling point, but since this is a predictive comparison, we will let the model RMSE decide the best fit.

We ran the models against each other thousands of times (all 70/30 CV), though what is shown below in Appendix 1.3 is just one 500 shuffle iteration.

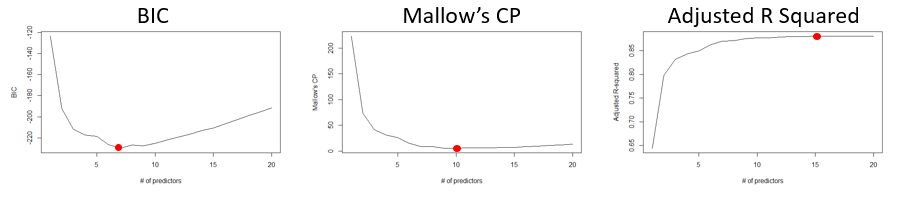
Model 1 performed worse on all metrics – Higher RMSE, Higher Variation, Larger range, and higher min/max values. To definitively conclude the model runs we performed a two-sample t-test on the difference in RMSE (Simple – Complex). We used a Welch’s t-test with the Satterthwaite approximation to degrees of freedom. This is likely a conservative test, but due to the shifting train/test sets we felt equal population variances may be violated1, though with equal sample sizes this risk is minimized. The results of the test provide extremely strong evidence that the complex model does in fact predict better than the simple model (two-sided p-value 2.2x10-16). A 95% CI gives us plausible values for the actual difference, (0.2089 to 0.2900). The difference in prediction (0.249) may be realistically unimportant.

1 Sample space is 128 C 55, so we are effectively testing 0% of the available space. This concern lead to the idea that there could be some wildly different variances between shuffles, and thus Welch’s t-test.



## Interpreting Coefficients

Model 1 was formed by going through EDA first, identifying multicollinearities and removing them, followed by some automated variable selection using 70/30 Cross Validation. The variable selection looked at 3 different metrics - BIC, Adj R2, and Mallow CP. The selection process was constructed to allow a reshuffle of the train/test sets every iteration so that we can get a better idea of the optimal number of predictors for any set. Through several runs the ideal minimal number of predictors given by the BIC was between 6 and 9, while the max was given by the Adj R2 at 16 to 20. Mallow Cp generally ran in the middle of the other two and suggested in between 10-15 predictors.



For our non-complex (simple) model we decided on the minimum number of predictors - 6. The final model we settled on was:

The reference level for SchoolFactor is low and for Status it is Developed. The SchoolFactor levels are Low: 0 – 12 years, Moderate: 12-16 years, and High 16+ years. This provides 6 predictive equations depending on the levels of the categorical variables SchoolFactor and Status. Without any interaction variables in the first model, the 6 equations’ only difference is the intercept.

Reference (Low School and Developed)

(Developing Change to Intercept):

(Moderate School and Developed)

(Developing Change to Intercept):

(High School and Developed)

(Developing Change to Intercept):

: Intercept, SchoolFactor, and Status – These categorical variables all affect the intercept and were thus grouped together.

The number of years of expected life expectancy due to SchoolFactor Low and Status Developed is 65.38 years, CI (59.95 to 70.81). All else equal we would expect to see a life expectancy change of -1.719 years CI (-3.70 to 0.223) if a country changed from Developed to Developing. Likewise, we would expect to see a positive change in intercept of 4.23 years CI (2.23 to 6.23) for school factor moving from low to moderate - which would indicate that significant portions of the country’s population are going to school for at least an average of 12 years, but not more than 16 years. And a positive change of 7.86 years CI (4.98 to 10.73) for those countries that move into high levels of schooling (an average of 16+ years).

: Log(HIV.AIDS) –

A multiplicative increase of (x) to the number of HIV-AIDS deaths per 1,000 people would be expected to produce a decrease in average life expectancy of 2.495Log(x), CI (-3.21 to -1.78). For instance, if a doubling of the deaths from HIV-AIDS occurred, all else held equal, we would expect average life expectancy to decrease by 2.495Log(2) or approximately 1.73 years CI (-2.25 years to -1.23 years).

: Adult Mortality –

A one-unit increase in Adult Mortality, all else held equal, would result in an expected decrease in average life expectancy of 0.012 years, CI (-0.02 to -0.003). This stat seems a bit underwhelming at first due to the way it is recorded. For example – Afghanistan from 2002 to 2003 saw an increase in Adult Mortality from 3 to 295, a nearly 100x increase. This would cause an expected decrease in average life expectancy of 1.2 years. *Example picked due to magnitude – this data point is not in our set.*

: Log(infant deaths) –

A multiplicative change to the number of infant deaths per 1,000 people, all else held equal, would be expected to see a decrease in average life expectancy of 0.51Log(x), CI (-1.00 to -0.02). For instance, a doubling of the infant death rate would be expected to result in a decrease of average life expectancy 0.35 years CI (-0.69 years to -0.014 years)

: Income Composition –

A one-unit increase in Income Composition, all else held equal, would be expected to increase average life expectancy by 4.98 years CI (-1.17 years to 11.13 years).

* Note these coefficients shown in Appendix 1.4 are from a single run of the model. The model was benchmarked using several runs of 500 iterations. All runs 70/30 CV.

# Objective 2: Non-Parametric Methodology & Analysis

We also wanted to explore non-parametric methods of fitting this data and see if leaving our assumptions of normal distributions produced a better fitting model. Our group tested KNN and Regression Tree against the regression model to see how it stacked up against one another.

Within the non-parametric tests, we couldn’t run the same parameters as was included in the regression because KNN functions better off of numeric variables. However, we did iterate 100 versions of three different models to ensure the highest accuracy of fit did not rely on a specific version of the data pulls. For the KNN variable selection, we used a trial and error replacement to come up with our final set of variables included in the model.

1. **Model 1:** Estimated life expectancy from Adult Mortality, Percentage Expenditure, Total Expenditure and Schooling
2. **Model 2:** Adult Mortality, Schooling, Log of HIV/AIDs, Log of infant deaths
3. **Model 3:** Adult Mortality, Percentage Expenditure, Diphtheria, and Schooling

Results are displayed in Appendix 2.1, Model 1 and Model 3 both performed well over the 100 iterations with Average RMSEs of 0.398 and 0.401 (according to the Welch’s t-test there was no significant difference between Model 1 & 3, p-value 0.70).

Since KNN can be difficult in interpretation and application of results, we also viewed a Regression tree analysis to see if it could provide a better fit than the KNN. Appendix 2.2 shows the pruned Regression tree and interpretation. The main concern we have surrounding the KNN model fit is that it may have high variance and low bias, resulting in an overfit. Since the KNN RMSE is leveraging scaled variables, the measurement can only be compared apples-to-apples with other models using scaled variables.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Linear Regression (Model 2) | KNN | Regression Tree |
| Average RMSE | 2.976 | 0.40 | 2.864 |

# Conclusion & Next Steps

Our Model 1 linear regression model highlights that life expectancy has a linear relationship with Schooling (factorized), HIV/AID (log transformed), Adult Mortality, Infant Deaths (log transformed), Income composition of resources and Status. These are the six levers that are available for the client to alter in hopes to effect the total life expectancy score for a country. Of course, practically, there a lot of macro-level confounding variables that are not included in this analysis, so there is not a perfect estimation of how to improve life expectancy over time.

Depending on the business objective of the client, we recommend using either the linear regression or the regression tree. If the goal is for prediction, the linear regression (Model 2) has the best fit of the life expectancy data. On the other hand, if the goal is for ease of interpretation, the regression tree can have a more practical understanding of how to interpret the model. The average RMSE says the difference between the linear regression model and the regression tree is 0.112, which in the case of a year would only be a difference of a month or so.

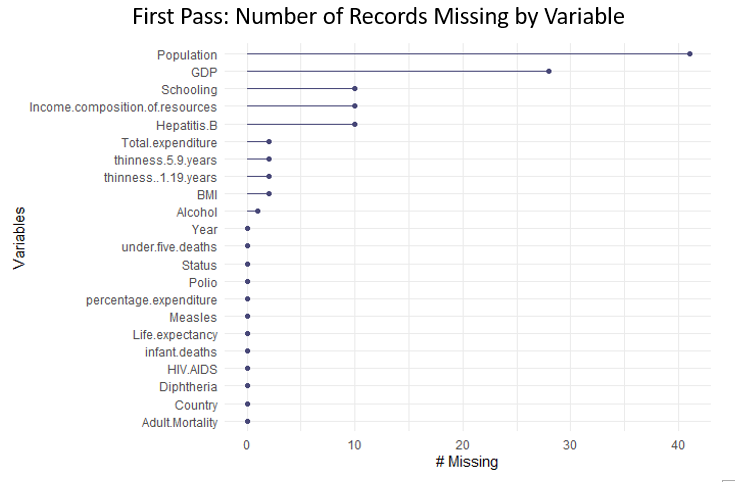
Given more time and resources we recommend testing the fit of the regression on alternative years to see if the factors hold true each year. It would also be interesting to note if the time series of the data has any effect on the parameters’ relationship with life expectancy.

# Appendix

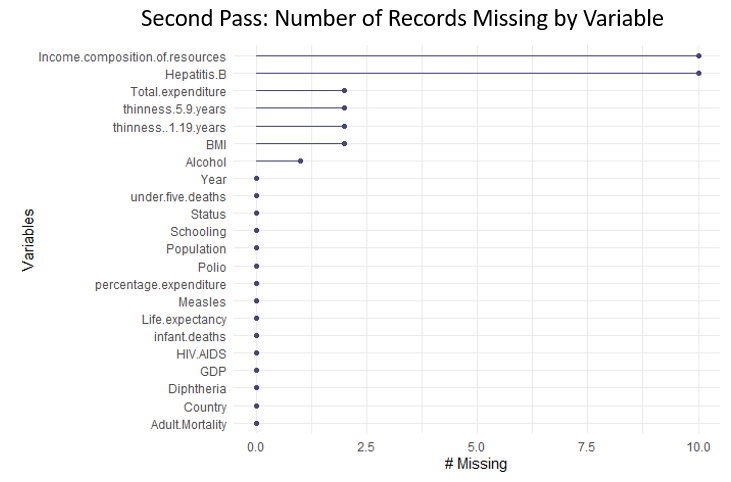
0.1  
Columns in the data

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| MSRP | Numeric | The response variable |
| Car Make | Factor | The company that made the car. Ex: Honda, Toyota, etc. |
| Car Model | Factor | The model of the car. Ex: 4Runner, Accord, etc. |
| Year | Numeric | Year the car was produced |
| Engine Fuel Type | Factor | Type of fuel the car accepts. Ex: Regular unleaded, Premium unleaded, Diesel |
| Engine HP | Numeric | Horsepower of the car’s engine. |
| Engine Cylinders | Numeric | Number of cylinders in the car’s engine. |
| Transmission Type | Factor | Type of transmission in the car. Usually manual or automatic, but there are a few specialty transmission types in the data. |
| Driven\_Wheels | Numeric | The wheels that are powered by the engine. Ex: Front Wheel, Rear Wheel, Four Wheel Drive |
| Number of Doors | Numeric | The number of doors that the car has. Usually 2 or 4 |
| Market Category | Factor | Various special factors for each car. Ex: Exotic, Luxury, High-Performance, Flex Fuel. Note: we created a new feature using Exotic/Not Exotic for our analysis |
| Vehicle Size | Factor | The size of the vehicle. Ex: Midsize, Large, Compact |
| Vehicle Style | Factor | Body type of the vehicle. Ex: Coupe, Convertible, etc. |
| Highway MPG | Numeric | Fuel efficiency on the highway in MPG |
| City MPG | Numeric | Fuel efficiency in the city in MPG |
| Popularity | Numeric | A popularity score for each car. The dataset does not detail how the popularity score is calculated. |

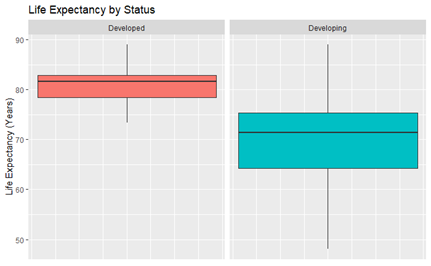
### 0.2



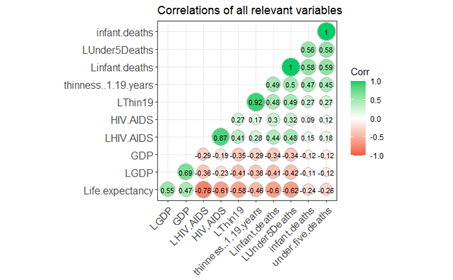
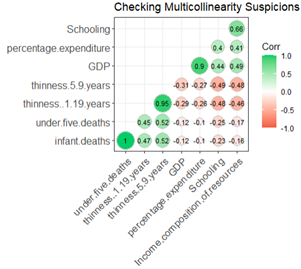
### 0.3



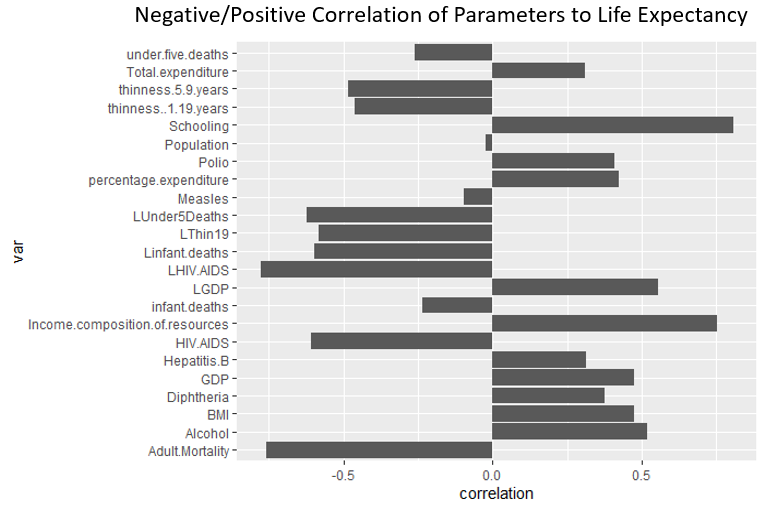
### 0.4



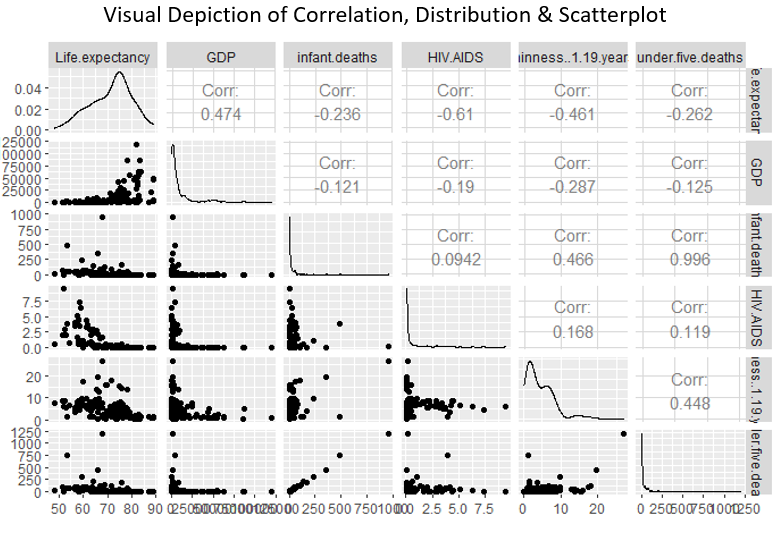
### 0.5



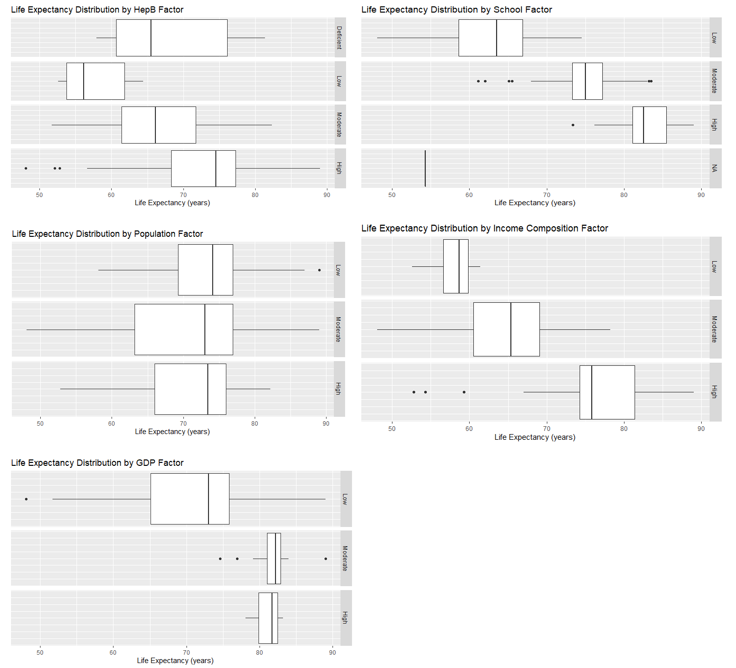
### 0.6



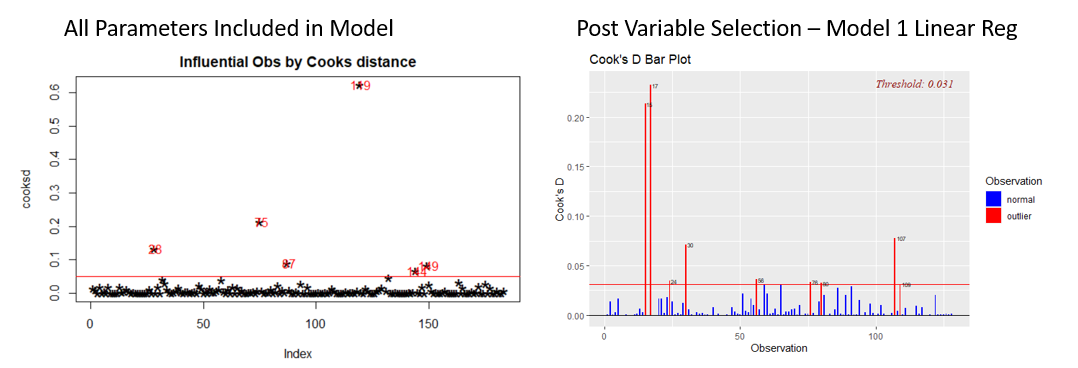
### 0.7



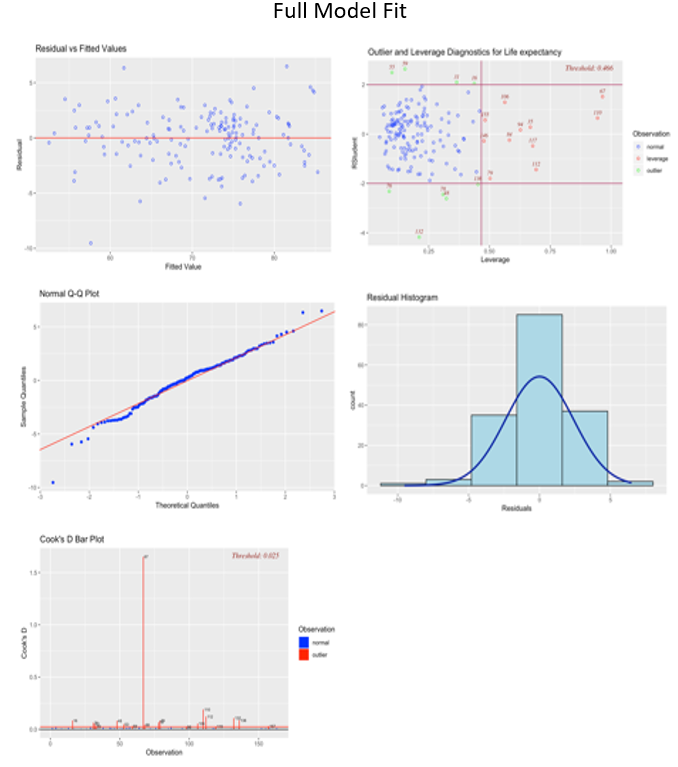
### 0.8



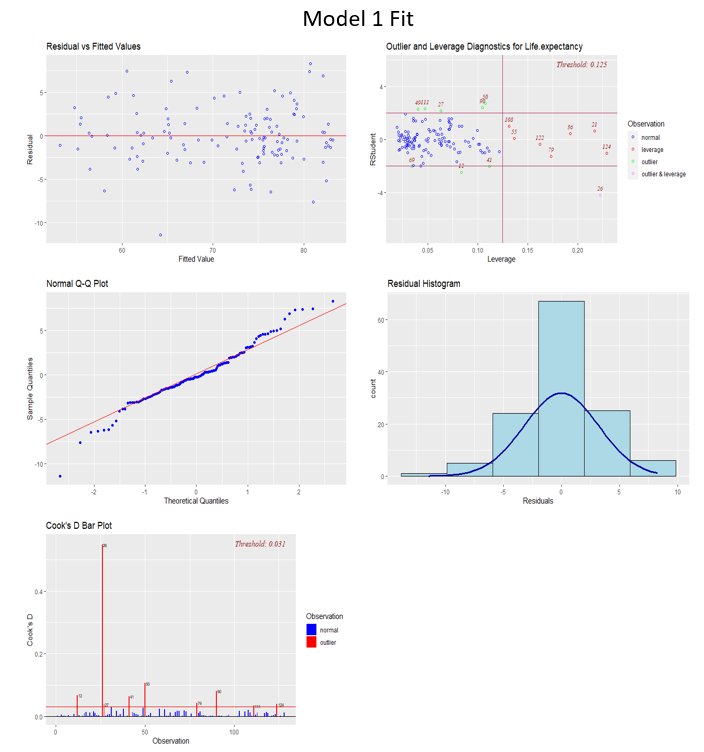
### 0.9



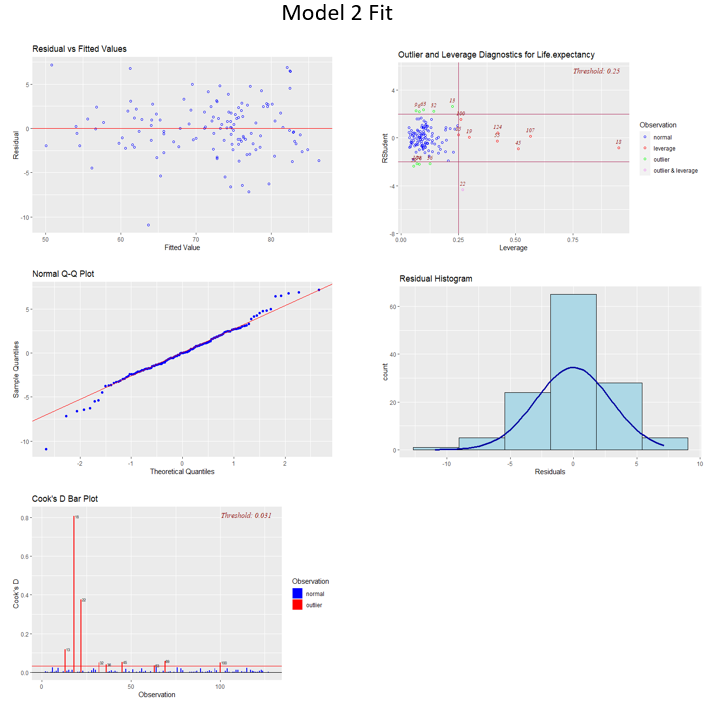
### 1.0



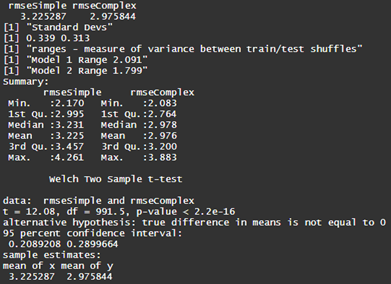
### 1.1



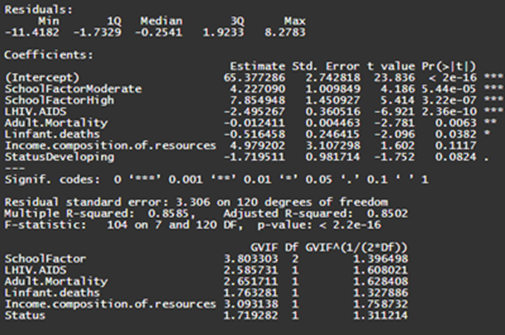
### 1.2



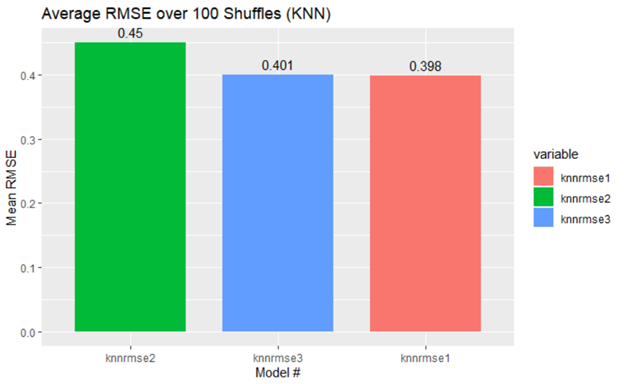
### 1.3



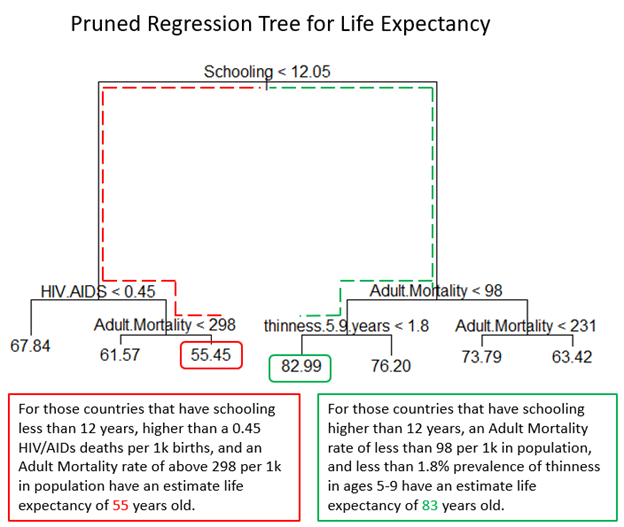
### 1.4



### 2.1



### 2.2



# Final R Code (we did not use any SAS)

---

title: "Combined\_Stats2\_Project1\_AC\_AP\_GL"

author: "Adam Canton, Anish Patel & Grace Lang"

date: "6/7/2020"

output: word\_document

---

Kaggle link for data reference: <https://www.kaggle.com/kumarajarshi/life-expectancy-who>

```{r setup, include=FALSE}

library(class)

library(caret)

library(e1071)

library(magrittr)

library(dplyr)

library(tidyr)

library(naniar)

library(ggplot2)

library(plotly)

library(forcats)

library(ggExtra)

library(glmnet)

library(GGally)

library(ggcorrplot)

library(car)

library(olsrr)

library(rgl)

library(tree)

library(ISLR)

library(leaps)

library(matrixStats)

library(FNN)

library(MASS)

library(reshape2)

#Adam's Connections

life <- read.csv(file = "F:/R For Real/Stats2 Project 1/Life Expectancy Data.csv")

#Anish's Connections

#life <- read.csv("Life Expectancy Data.csv")

#Grace's Connections

#life <-read.csv("C:/Users/david/OneDrive/Desktop/AppliedStats/6372\_AppliedStats\_GraceLang/Project1/LifeExpectancyData.csv")

#life <-read.csv("C:/Users/Dave/Desktop/AppliedStats/Stats2\_Project1/LifeExpectancyData.csv")

#Just pull out 2014 into the data

Life14 <- subset(life, Year==2014)

summary(Life14)

```

#Column definitions:

Adult.mortality: Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)

Infant.death: Number of Infant Deaths per 1000 population

Alcohol: Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)

percentage expenditure: Expenditure on health as a percentage of Gross Domestic Product per capita(%)

Hepatitis B: Hepatitis B (HepB) immunization coverage among 1-year-olds (%)

Measles - number of reported cases per 1000 population

BMI: Average Body Mass Index of entire population

Under-five-deaths: Number of under-five deaths per 1000 population

Polio: Polio (Pol3) immunization coverage among 1-year-olds (%)

Total expenditure: General government expenditure on health as a percentage of total government expenditure (%)

Diphtheria: Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)

HIV/AIDS: Deaths per 1 000 live births HIV/AIDS (0-4 years)

GDP: Gross Domestic Product per capita (in USD)

Population: Population of the country

thinness: Prevalence of thinness among children and adolescents for Age 10 to 19 (% )

Thinness 5-9: Prevalence of thinness among children for Age 5 to 9(%)

income composition: Human Development Index in terms of income composition of resources (index ranging from 0 to 1)

Schooling: Number of years of Schooling(years)

##Things to note from summary stats:

\* 83% of data comes from developing countries, life exp may be lower.

\* Mean life expectancy is 71.54 yr old

\* The max for Measles seems outrageously high 79k - which country? outlier?

\* Looks like there's a couple of positive correlations with life expectancy: income.compostion & schooling

\* Adult mortality seems to have a gender split, but cannot determine for certain

\* No outward evidence of any of the parameters having a non-linear relationship (adding a quadratic)

# Fixing NA's in GDP

```{r}

Life14$GDP[159] = 2520.2027

Life14$GDP[11] = 29436.13

Life14$GDP[20] = 3081.23

Life14$GDP[28] = 1559.38

Life14$GDP[39] = 6208.58

Life14$GDP[44] = 19771.65

Life14$GDP[45] = 1568.63

Life14$GDP[46] = 352.82

Life14$GDP[51] = 3378.68

Life14$GDP[54] = 1209.68

Life14$GDP[61] = 607.21

Life14$GDP[77] = 5608.62

Life14$GDP[89] = 1279.64

Life14$GDP[90] = 1998.49

Life14$GDP[95] = 6466.52

Life14$GDP[107] = 2971.73

Life14$GDP[131] = 27802.96

Life14$GDP[132] = 2674.35

Life14$GDP[136] = 8743.89

Life14$GDP[137] = 6684.67

Life14$GDP[146] = 18671.59

Life14$GDP[162] = 5487.92

Life14$GDP[173] = 47614.61

Life14$GDP[174] = 1000.00

Life14$GDP[175] = 55025.13

Life14$GDP[179] = 16053.24

Life14$GDP[180] = 2030.31

Life14$GDP[181] = 1674.28

```

# Fixing Population NA's

```{r}

Life14$Population[159] = 18715672

Life14$Population[11] = 370633

Life14$Population[20] = 10710000

Life14$Population[28] = 22650000

Life14$Population[39] = 5244359

Life14$Population[44] = 10510000

Life14$Population[45] = 25500000

Life14$Population[46] = 10780263

Life14$Population[51] = 90420000

Life14$Population[54] = 5580000

Life14$Population[61] = 2024000

Life14$Population[77] = 77470000

Life14$Population[89] = 5836000

Life14$Population[90] = 6640000

Life14$Population[95] = 6362000

Life14$Population[107] = 107446

Life14$Population[131] = 50750000

Life14$Population[132] = 3556000

Life14$Population[136] = 178296

Life14$Population[137] = 108861

Life14$Population[146] = 5420000

Life14$Population[162] = 2070000

Life14$Population[173] = 64350000

Life14$Population[174] = 49960000

Life14$Population[175] = 318400000

Life14$Population[179] = 30050000

Life14$Population[180] = 91710000

Life14$Population[181] = 25820000

Life14$Population[130] = 2459000

Life14$Population[145] = 5470000

Life14$Population[172] = 9214000

Life14$Population[88] = 3691000

Life14$Population[12] = 1336000

Life14$Population[140] = 30920000

Life14$Population[14] = 284825

Life14$Population[5] = 92562

Life14$Population[66] = 108902

Life14$Population[116] = 4510000

Life14$Population[24] = 409769

Life14$Population[121] = 4027000

Life14$Population[149] = 13420000

Life14$Population[42] = 11310000

```

# Fixing 27 - 0 Percent Expenditures

```{r}

Life14$percentage.expenditure[11] = 748.36

Life14$percentage.expenditure[20] = 115.35

Life14$percentage.expenditure[28] = 16.85

Life14$percentage.expenditure[39] = 34.78

Life14$percentage.expenditure[44] = 1249.23

Life14$percentage.expenditure[46] = 2.68

Life14$percentage.expenditure[51] = 48.66

Life14$percentage.expenditure[54] = 2.82

Life14$percentage.expenditure[61] = 4.93

Life14$percentage.expenditure[77] = 192.78

Life14$percentage.expenditure[89] = 35.5

Life14$percentage.expenditure[90] = 13.79

Life14$percentage.expenditure[95] = 197.84

Life14$percentage.expenditure[107] = 102.43

Life14$percentage.expenditure[131] = 1093.43

Life14$percentage.expenditure[132] = 113.87

Life14$percentage.expenditure[136] = 189.84

Life14$percentage.expenditure[137] = 178.51

Life14$percentage.expenditure[146] = 1024.94

Life14$percentage.expenditure[159] = 31.65

Life14$percentage.expenditure[162] = 219.74

Life14$percentage.expenditure[173] = 3682.48

Life14$percentage.expenditure[174] = 11.47

Life14$percentage.expenditure[175] = 4541.8

Life14$percentage.expenditure[179] = 235.35

Life14$percentage.expenditure[180] = 48.82

Life14$percentage.expenditure[181] = 12.9

```

# Fixing the 10 missing school variables

```{r}

Life14$Schooling[28] = 5.2

Life14$Schooling[44] = 12.7

Life14$Schooling[45] = 10.8

Life14$Schooling[46] = 6.8

Life14$Schooling[131] = 12.2

Life14$Schooling[132] = 11.6

Life14$Schooling[149] = 0

Life14$Schooling[173] = 17.4

Life14$Schooling[174] = 8

Life14$Schooling[175] = 16.3

```

#What variables are still missing

```{r}

gg\_miss\_var(Life14)

```

```{r}

#Can I add the avgs into the NAs without a lot of shift in data

#Life14 %>% ggplot(aes(x = Income.composition.of.resources)) + geom\_histogram() #normally distributed, low effect 2% change in mean

#Life14 %>% ggplot(aes(x = Hepatitis.B)) + geom\_histogram() #left-skewed, low effect 0% change in mean

#Life14 %>% ggplot(aes(x = Total.expenditure)) + geom\_histogram() #normally distributed, low effect 0% change in mean

#Life14 %>% ggplot(aes(x = thinness.5.9.years)) + geom\_histogram() #right-skewerd, low effect 0% change in mean

#Life14 %>% ggplot(aes(x = thinness..1.19.years)) + geom\_histogram() #right-skewerd, low effect 0% change in mean

#Life14 %>% ggplot(aes(x = BMI)) + geom\_histogram() #normally distributed, low effect 0% change in mean

#Life14 %>% ggplot(aes(x = Alcohol)) + geom\_histogram() #right-skewerd, low effect 0% change in mean

#Replacing NAs with the mean of the column

Life14$Income.composition.of.resources[is.na(Life14$Income.composition.of.resources)] <- round(mean(Life14$Income.composition.of.resources, na.rm = TRUE))

Life14$Hepatitis.B[is.na(Life14$Hepatitis.B)] <- round(mean(Life14$Hepatitis.B, na.rm = TRUE))

Life14$Total.expenditure[is.na(Life14$Total.expenditure)] <- round(mean(Life14$Total.expenditure, na.rm = TRUE))

Life14$thinness..1.19.years[is.na(Life14$thinness..1.19.years)] <- round(mean(Life14$thinness..1.19.years, na.rm = TRUE))

Life14$thinness.5.9.years[is.na(Life14$thinness.5.9.years)] <- round(mean(Life14$thinness.5.9.years, na.rm = TRUE))

Life14$BMI[is.na(Life14$BMI)] <- round(mean(Life14$BMI, na.rm = TRUE))

Life14$Alcohol[is.na(Life14$Alcohol)] <- round(mean(Life14$Alcohol, na.rm = TRUE))

```

# Adding in some feature engineering

```{r}

#Factorizing things

IncomeCompFactor = cut(Life14$Income.composition.of.resources, breaks = c(0,0.4,0.7,1), labels = c("Low", "Moderate", "High"))

HepBfactor = cut(Life14$Hepatitis.B, breaks = c(0,25,50,75,100), labels = c("Deficient", "Low", "Moderate", "High"))

PercentExpendFactor = cut(Life14$percentage.expenditure, breaks = c(0,5000,15000,20000), labels = c("Low", "Moderate", "High"))

PopulationFactor = cut(Life14$Population, breaks = c(0, 3.612e+05,2.196e+07,1.294e+09), labels = c("Low", "Moderate", "High"))

GdpFactor = cut(Life14$GDP, breaks = c(0,40000,80000,120000), labels = c("Low","Moderate","High"))

SchoolFactor = cut(Life14$Schooling, breaks = c(0,12,16,21), labels = c("Low","Moderate", "High"))

# Binding them back to Life14

Life14 = cbind(Life14, HepBfactor, PercentExpendFactor, PopulationFactor, GdpFactor, SchoolFactor, IncomeCompFactor)

#Fixing some errors where for some reason it wasn't picking up these 3

Life14$PercentExpendFactor[45] = as.factor("Low")

Life14$PercentExpendFactor[149] = as.factor("Low")

Life14$SchoolFactor[149] = as.factor("Low")

#adding log variables

Life14 = Life14 %>% mutate(LHIV.AIDS = log(HIV.AIDS))

Life14 = Life14 %>% mutate(Linfant.deaths = log(infant.deaths + 1))

Life14 = Life14 %>% mutate(LUnder5Deaths = log(under.five.deaths + 1))

Life14 = Life14 %>% mutate(LThin19 = log(thinness..1.19.years + 1))

Life14 = Life14 %>% mutate(LGDP = log(GDP))

```

#Visualizing some of the data

```{r}

# Life expectancy by Country and status - goes well with boxplot below

Life14 %>% group\_by(Country, Status) %>% summarise(LifeEx = Life.expectancy) %>%

ggplot(aes(x = reorder(Country, -LifeEx), y = LifeEx, fill = Status)) +

geom\_col(width = 0.75, na.rm = TRUE) +

theme(axis.text.x = element\_blank()) +

xlab("Country") +

ggtitle("Life Expectancy 2014")

```

Boxplot of life expectancy by country status:

```{r}

Life14 %>% ggplot(aes(y = Life.expectancy)) + geom\_boxplot(aes(fill = Status)) + facet\_wrap(~Status) +

ylab("Life Expectancy (Years)") + ggtitle("Life Expectancy by Status") +

theme(axis.title.x = element\_blank(), axis.text.x = element\_blank(), axis.ticks.x = element\_blank(), legend.position = "none")

```

Pairwise plots for selected variables:

```{r message = FALSE, warning = FALSE}

df2 <- dplyr::select(Life14, c(Life.expectancy, Adult.Mortality, Total.expenditure, HIV.AIDS, Income.composition.of.resources, Schooling))

library(GGally)

ggpairs(df2)

```

Correlation between life expectancy and all the numeric independent variables:

```{r}

# Good now

cor.xy <- cor(Life14 %>% dplyr::select(-c(Country, Year, Status,IncomeCompFactor,HepBfactor,PercentExpendFactor,PopulationFactor,GdpFactor,SchoolFactor)), use = "complete.obs")

LE.cor <- data.frame(var = rownames(cor.xy)[-1], correlation = cor.xy[-1, 1])

LE.cor %>% ggplot(aes(x = var, y = correlation)) + geom\_col() + coord\_flip()

```

# Checking Correlations between original variables and their logged counterparts as well as the response - we may only achieve some minor inprovements in GDP and HIV.AIDS but under 5 deaths and infant deaths have changed significantly. Also HIV.AIDS and infant deaths without the log are oddly related

```{r}

corr <- Life14 %>% dplyr::select(Life.expectancy, LHIV.AIDS, Linfant.deaths, LUnder5Deaths,

LThin19, LGDP, GDP, infant.deaths, HIV.AIDS, under.five.deaths, thinness..1.19.years)

corr <- round(cor(corr), 2)

ggcorrplot(corr, hc.order = TRUE, type = "lower",

lab = TRUE, lab\_size = 3, method = "circle",

colors = c("tomato2", "white", "springgreen3"),

title = "Correlations of all relevant variables",

ggtheme = theme\_bw())

```

# Looking at Correlations for the rest of the predictors

```{r}

exclude\_factors <- c( "Country", "Year", "Status", "thinness.5.9.years", "under.five.deaths", "IncomeCompFactor",

"HepBfactor", "PercentExpendFactor", "SchoolFactor", "GdpFactor", "PopulationFactor",

"LHIV.AIDS", "Linfant.deaths", "LUnder5Deaths", "LThin19", "LGDP")

corr <- Life14 %>% dplyr::select(-all\_of(exclude\_factors))

corr <- round(cor(corr), 2)

ggcorrplot(corr, hc.order = TRUE, type = "lower",

lab = TRUE, lab\_size = 3, method = "circle",

colors = c("tomato2", "white", "springgreen3"),

title = "Correlations of all relevant variables",

ggtheme = theme\_bw())

```

Get rid of Country Name and Year

```{r}

exclude\_factors <- c("Country", "Year")

Life14 <- Life14 %>% dplyr::select( -all\_of(exclude\_factors))

```

# Reasoning behind why we log-transformed some of our variables

```{r outliers}

#Running full regression model

full.model <- lm(Life.expectancy~.,data=Life14)

vif(full.model)

ols\_plot\_comp\_plus\_resid(full.model)

#Observe residuals and outliers

par(mfrow=c(2,2))

plot(full.model, which =1)

#Which obs are outliers according to Cooks D

cooksd <- cooks.distance(full.model)

plot(cooksd, pch="\*", cex=2, main="Influential Obs by Cooks distance") # plot cook's distance

abline(h = 4\*mean(cooksd, na.rm=T), col="red") # add cutoff line

text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4\*mean(cooksd, na.rm=T),names(cooksd),""), col="red")

plot(rstudent(full.model))

# Influence Plot

influencePlot(full.model, main="Influence Plot")

#influencePlot(full.model, id.method="identify", main="Influence Plot", sub="Circle size is proportial to Cook's Distance" )

```

```{r, warning=FALSE}

# Set training and Testing Sets

#set.seed(1234)

index<-sample(1:dim(Life14)[1],128,replace=F)

train<-Life14[index,]

test<-Life14[-index,]

# Forward Selection

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

bics<-summary(reg.fwd)$bic

plot(1:20,bics,type="l",ylab="BIC",xlab="# of predictors")

index<-which(bics==min(bics))

points(index,bics[index],col="red",pch=10)

print("Min Bics is:")

which(bics==min(bics))

# Adjr2

adjr2<-summary(reg.fwd)$adjr2

plot(1:20,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of predictors")

index<-which(adjr2==max(adjr2))

points(index,adjr2[index],col="red",pch=10)

print("Max Adj R2 is:")

which(adjr2==max(adjr2))

MallowCP <- summary(reg.fwd)$cp

plot(1:20,MallowCP,type="l",ylab="Mallow's CP",xlab="# of predictors")

index<-which(MallowCP==min(MallowCP))

points(index,MallowCP[index],col="red",pch=10)

print("Min Mallow CP is:")

which(MallowCP==min(MallowCP))

```

# Used to quickly cycle though models while shuffling train/test sets

Ols step both does forward and backward selection at each step in this case based on model AIC

```{r, warning=FALSE}

index<-sample(1:dim(Life14)[1],128,replace=F)

train<-Life14[index,]

test<-Life14[-index,]

modelgenerator <- lm(Life.expectancy ~ . , data = train)

ols\_step\_both\_aic(modelgenerator)

# Gets significant interactions based on aic - can take a minute to run

#Interaction.modelgenerator <- lm(Life.expectancy ~ (.)^2 , data = train)

#ols\_step\_both\_aic(Interaction.modelgenerator)

```

# Gut Model - EDA first - human variable selection - Some tech use to look at optimum number of predictors and comparison of models - income comp increases model variance

```{r}

index<-sample(1:dim(Life14)[1],128,replace=F)

train<-Life14[index,]

test<-Life14[-index,]

modeltest1 <- lm(Life.expectancy ~ SchoolFactor + LHIV.AIDS + Adult.Mortality + Linfant.deaths + Income.composition.of.resources + Status, data = train)

summary(modeltest1)

vif(modeltest1)

ols\_plot\_resid\_fit(modeltest1)

ols\_plot\_resid\_lev(modeltest1)

ols\_plot\_resid\_qq(modeltest1)

ols\_plot\_resid\_hist(modeltest1)

ols\_plot\_cooksd\_bar(modeltest1)

```

# complex model - Contains 4 quadratic terms and 1 interaction. Adding more interactions seems to overfit. Will occasionally show some crazy Cook's D based on which train/test gets selected. For a better view - run the model benchmark - easy to see over 100-500 interations

```{r}

modeltest2 <- lm(Life.expectancy ~ SchoolFactor + poly(LHIV.AIDS,2) + poly(Adult.Mortality,2) + poly(under.five.deaths,2) + poly(thinness..1.19.years,2) +

LGDP + Income.composition.of.resources + percentage.expenditure + Income.composition.of.resources:Schooling + Status, data = train)

summary(modeltest2)

ols\_plot\_resid\_fit(modeltest2)

ols\_plot\_resid\_lev(modeltest2)

ols\_plot\_resid\_qq(modeltest2)

ols\_plot\_resid\_hist(modeltest2)

ols\_plot\_cooksd\_bar(modeltest2)

```

# Linear Model Benchmarking - Welch's t-test run at the end (likely a conservative option). 500 observations each (Robust to normal violations) - however sample SD can vary a decent bit based on train/test shuffles and the whole population is 128c55....

```{r, warning=FALSE}

# Set number of times you would like to repeat the sampling/testing

iterations = 1:500

# the initial values for the columns (might not need these now that ive switched to building columns)

rmseSimple = c()

rmseComplex = c()

# Start of Loop

for(i in iterations){

# Resets sample every iteration

index<- sample(1:dim(Life14)[1],128,replace=F)

train<- Life14[index,]

test<- Life14[-index,]

# the model runs

modeltest1

modeltest2

# predictors and column building

predictions1 <- modeltest1 %>% predict(test)

d1 = data.frame(R2 = R2(predictions1,test$Life.expectancy),

RMSE = RMSE(predictions1,test$Life.expectancy), MAE = MAE(predictions1, test$Life.expectancy))

rmseSimple = c(rmseSimple,d1$RMSE)

predictions2 <- modeltest2 %>% predict(test)

d2 = data.frame(R2 = R2(predictions2,test$Life.expectancy),

RMSE = RMSE(predictions2,test$Life.expectancy), MAE = MAE(predictions2, test$Life.expectancy))

rmseComplex = c(rmseComplex, d2$RMSE)

# End for

}

# putting the dataframe together and outputting relevant statistics

Model.Average.RMSE = cbind(rmseSimple, rmseComplex)

rmsedf = as.data.frame(Model.Average.RMSE)

Means = colMeans(Model.Average.RMSE)

SDs = round(colSds(Model.Average.RMSE), 3)

range1 = max(rmsedf$rmseSimple) - min(rmsedf$rmseSimple)

range2 = max(rmsedf$rmseComplex) - min(rmsedf$rmseComplex)

rmsedf1 = melt(rmsedf,rmse = c("n", "rmse"))

# Looking at descriptive stats

Means

print("Standard Devs")

SDs

print("ranges - measure of variance between train/test shuffles")

print(paste("Model 1 Range" , round(range1, 3)))

print(paste("Model 2 Range" , round(range2, 3)))

cat("Summary:

")

summary(Model.Average.RMSE)

# Scatter

Pred1 <- data.frame(Value = predictions1, Model = "Simple")

Pred2 <- data.frame(Value = predictions2, Model = "Complex")

PredActual <- data.frame(ActualValue = test$Life.expectancy)

PredAll <- rbind(Pred1, Pred2)

PredActual <- rbind(PredActual,PredActual)

PredAll <- cbind(PredAll, PredActual)

PredAll %>% ggplot(aes(x = Value, y = ActualValue, fill = Model)) + geom\_point(aes(color = Model)) + geom\_smooth(formula = y~x)+

facet\_wrap(facets = PredAll$Model) + ggtitle("Scatter Plot of Models") + xlab("Predicted") + ylab("Observed")

# Column

rmsedf1 %>% group\_by(variable) %>% summarise(mean = (mean(value))) %>%

ggplot(aes(x = reorder(variable, -mean), y = mean, fill = variable)) + geom\_col(width = 0.75) + geom\_text(aes(label = round(mean,3), vjust = -0.5)) +

ggtitle("Average RMSE over 500 Shuffles (Linear Models)") + xlab("Model #") + ylab("Mean RMSE")

# Boxplot

rmsedf1 %>% ggplot(aes(x = variable, y = value)) + geom\_boxplot(aes(fill = variable)) + facet\_wrap(~variable,ncol = TRUE) +

ggtitle("Mean RMSE Distribution by Model") + ylab("Mean RMSE") + coord\_flip() +

theme(axis.title.y = element\_blank(), axis.text.y = element\_blank(), axis.ticks.y = element\_blank())

# Histogram

rmsedf1 %>% ggplot(aes(x = value)) + geom\_histogram(aes(fill = variable)) + facet\_wrap(~variable,ncol = TRUE) +

ggtitle("Mean RMSE Distribution by Model") + xlab("Mean RMSE") +

theme(axis.title.y = element\_blank(), axis.text.y = element\_blank(), axis.ticks.y = element\_blank())

# Here we can see there is no significant difference between the models in terms of RMSE

t.test(rmseSimple,rmseComplex, var.equal = FALSE)

```

```{r}

# Creating scaled variables for knn testing

Life14Scale = Life14

Life14Scale$Life.expectancy = scale(Life14Scale$Life.expectancy)

Life14Scale$Adult.Mortality = scale(Life14Scale$Adult.Mortality)

Life14Scale$percentage.expenditure = scale(Life14Scale$percentage.expenditure)

Life14Scale$Diphtheria = scale(Life14Scale$Diphtheria)

Life14Scale$Hepatitis.B = scale(Life14Scale$Hepatitis.B)

Life14Scale$Income.composition.of.resources = scale(Life14Scale$Income.composition.of.resources)

Life14Scale$Schooling = scale(Life14Scale$Schooling)

Life14Scale$LHIV.AIDS = scale(Life14Scale$LHIV.AIDS)

Life14Scale$infant.deaths = scale(Life14Scale$infant.deaths)

Life14Scale$Linfant.deaths = scale(Life14Scale$Linfant.deaths)

Life14Scale$LUnder5Deaths = scale(Life14Scale$LUnder5Deaths)

Life14Scale$LThin19 = scale(Life14Scale$LThin19)

Life14Scale$LGDP = scale(Life14Scale$LGDP)

Life14Scale$Total.expenditure = scale(Life14Scale$Total.expenditure)

```

Knn Model benchmarking

```{r, warning=FALSE}

# Set knn vectors - these are the column #s that will be analyzed

knnvector1 = c(3,4,14,20) # Adult.Mortality, schooling, HIV, Infant deaths

knnvector2 = c(3,20,27,28) # Adult.Mortality, Schooling, Log(HIV), Log(infant)

knnvector3 = c(3,6,13,20) # Adult.Mortality, percentage.expenditure, Diptheria, schooling

# Start of Loop Variables

iterations = 1:500

i = 0

knnrmse1 = c()

knnrmse2 = c()

knnrmse3 = c()

# Loop start

for(i in iterations){

# Create train and test sets

index<-sample(1:dim(Life14Scale)[1],128,replace=F)

train<-Life14Scale[index,]

test<-Life14Scale[-index,]

#KNN1

predknn1 <- knn.reg(train[,knnvector1], test[, knnvector1], train$Life.expectancy, k = 4)

Knn\_Predictions\_1 = as.data.frame(predknn1$pred)

names(Knn\_Predictions\_1)[1] <- "Predictions"

trial <- data.frame(Predictions = c(Knn\_Predictions\_1), Actual = c(test$Life.expectancy))

trial = trial %>% mutate(ResidualSq1 = (Predictions - Actual)^2)

RMSEknn1 = sqrt(sum((trial$ResidualSq1)/nrow(test)))

knnrmse1 = c(knnrmse1, RMSEknn1)

#KNN2

predknn2 <- knn.reg(train[,knnvector2], test[, knnvector2], train$Life.expectancy, k = 9)

Knn\_Predictions\_2 = as.data.frame(predknn2$pred)

names(Knn\_Predictions\_2)[1] <- "Predictions"

trial <- data.frame(Predictions = c(Knn\_Predictions\_2), Actual = c(test$Life.expectancy))

trial = trial %>% mutate(ResidualSq2 = (Predictions - Actual)^2)

RMSEknn2 = sqrt(sum((trial$ResidualSq2)/nrow(test)))

knnrmse2 = c(knnrmse2, RMSEknn2)

# KNN3

predknn3 <- knn.reg(train[,knnvector3], test[, knnvector3], train$Life.expectancy, k = 4)

Knn\_Predictions\_3 = as.data.frame(predknn3$pred)

names(Knn\_Predictions\_3)[1] <- "Predictions"

trial <- data.frame(Predictions = c(Knn\_Predictions\_3), Actual = c(test$Life.expectancy))

trial = trial %>% mutate(ResidualSq3 = (Predictions - Actual)^2)

RMSEknn3 = sqrt(sum((trial$ResidualSq3)/nrow(test)))

knnrmse3 = c(knnrmse3, RMSEknn3)

# End For

}

# putting the dataframe together and outputting relevant statistics

Knn.Model.Average.RMSE = cbind(knnrmse1, knnrmse2, knnrmse3)

knnrmsedf = as.data.frame(Knn.Model.Average.RMSE)

Means = colMeans(Knn.Model.Average.RMSE)

SDs = round(colSds(Knn.Model.Average.RMSE), 3)

knnrange1 = max(knnrmsedf$knnrmse1) - min(knnrmsedf$knnrmse1)

knnrange2 = max(knnrmsedf$knnrmse2) - min(knnrmsedf$knnrmse2)

knnrange3 = max(knnrmsedf$knnrmse3) - min(knnrmsedf$knnrmse3)

knnrmsedf1 = melt(knnrmsedf,rmse = c("n", "rmse"))

# Looking at descriptive stats

Means

print("Standard Devs")

SDs

print("ranges - measure of variance between train/test shuffles")

print(paste("Model 1 Range" , round(knnrange1, 4)))

print(paste("Model 2 Range" , round(knnrange2, 4)))

print(paste("Model 3 Range" , round(knnrange3, 4)))

cat("Summary:

")

summary(Knn.Model.Average.RMSE)

# Column

knnrmsedf1 %>% group\_by(variable) %>% summarise(mean = (mean(value))) %>%

ggplot(aes(x = reorder(variable, -mean), y = mean, fill = variable)) + geom\_col(width = 0.75) + geom\_text(aes(label = round(mean,3), vjust = -0.5)) +

ggtitle("Average RMSE over 100 Shuffles (KNN)") + xlab("Model #") + ylab("Mean RMSE")

# Boxplot

knnrmsedf1 %>% ggplot(aes(x = variable, y = value)) + geom\_boxplot(aes(fill = variable)) + facet\_wrap(~variable,ncol = TRUE) +

ggtitle("Mean RMSE Distribution by Model") + ylab("Mean RMSE") + coord\_flip() +

theme(axis.title.y = element\_blank(), axis.text.y = element\_blank(), axis.ticks.y = element\_blank())

#

# Dont forget multiple comparison adjustment

t.test(knnrmse1,knnrmse2, var.equal = FALSE)

t.test(knnrmse1,knnrmse3, var.equal = FALSE)

t.test(knnrmse2,knnrmse3, var.equal = FALSE)

```

# Looks for optimal K value between 1-50. Just change the knnvector to run different models.

```{r}

iterations = 50

numks = 50

masterRMSE = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)

{

rmseknn = data.frame(rmse = numeric(50), k = numeric(50))

index<-sample(1:dim(Life14Scale)[1],128,replace=F)

train<-Life14Scale[index,]

test<-Life14Scale[-index,]

for(i in 3:numks)

{

predknn <- knn.reg(train[,knnvector3], test[,knnvector3], train$Life.expectancy, k = i)

Knn\_Predictions = as.data.frame(predknn$pred)

names(Knn\_Predictions)[1] <- "Predictions"

trial <- data.frame(Predictions = c(Knn\_Predictions), Actual = c(test$Life.expectancy))

trial = trial %>% mutate(ResidualSq = (Predictions - Actual)^2)

RMSEknn = sqrt(sum((trial$ResidualSq)/nrow(test)))

masterRMSE[j,i] = RMSEknn

}

}

masterRMSE = masterRMSE[,-(1:2)]

MeanRMSE = colMeans(masterRMSE)

# How can I title this plot and change the x axis label? I Suck at Base R

plot(seq(3,numks,1),MeanRMSE, type = "l")

which.min(MeanRMSE)

min(MeanRMSE)

```

This model is here to run against the knn for comparison. It uses the same scaled variables as the knnvector3 (linear version of the KNN).

```{r}

#Create train and test sets for scaled linear models - currently reflects knnvector3

indexS<-sample(1:dim(Life14Scale)[1],128,replace=F)

trainS<-Life14Scale[indexS,]

testS<-Life14Scale[-indexS,]

modeltest3 <- lm(Life.expectancy ~ Adult.Mortality + percentage.expenditure + Diphtheria + Schooling, data = trainS)

summary(modeltest3)

vif(modeltest3)

ols\_plot\_resid\_fit(modeltest3)

ols\_plot\_resid\_lev(modeltest3)

ols\_plot\_resid\_qq(modeltest3)

ols\_plot\_resid\_hist(modeltest3)

ols\_plot\_cooksd\_bar(modeltest3)

```

# Regression Tree Model

```{r tree}

#Building a tree out, based on ALL variables from the regression model

par(mfrow=c(1,1))

tree.life<-tree(Life.expectancy~.,train,minsize=5)

summary(tree.life)

plot(tree.life)

text(tree.life,pretty=0)

#Perform CV to deterine if we need to prune the tree. -- 7 parameters was the lowest

set.seed(1234)

cv.life<-cv.tree(tree.life,FUN=prune.tree,method="deviance")

plot(cv.life)

plot(cv.life$size, cv.life$dev, type='b')

#Building a tree out, based on SELECT variables from the regression model

par(mfrow=c(1,1))

tree.life2<-tree(Life.expectancy ~ Schooling + Adult.Mortality + HIV.AIDS + GDP + thinness.5.9.years + Diphtheria + Alcohol,Life14,minsize=5)

summary(tree.life2)

plot(tree.life2)

text(tree.life2,pretty=0)

#Perform CV to deterine if we need to prune the tree. -- All 7 parameters was the lowest

set.seed(1234)

cv.life2<-cv.tree(tree.life2,FUN=prune.tree,method="deviance")

plot(cv.life2)

#Fitting a final model for predicting future values.

#both versions of the tree identified that 7 parameters was the best fit

prune.life=prune.tree(tree.life,best=7)

plot(prune.life)

text(prune.life,pretty=0)

#Looking at the predictions of the pruned tree

tree.ASE <- mean((test$Life.expectancy - predict(prune.life,newdata =test))^2)

tree.ASE

tree.RMSE <- sqrt(mean((test$Life.expectancy - predict(prune.life,newdata =test))^2))

tree.RMSE

```