LifeExpectancy

Douglas Melis

10/18/2020

## Data Preprocessing

Not all of the data sets I used have the standardized country code, so I had to use country names to match up the records. Unfortuantely, a number of countries have multiple names or slight variations in spellings or format. Therefore, the first step was to create a function to clean the country names column in each dataset. The function I created runs through a set of name variations and replaces each one with the variation I selected for the final dataset.

clean\_country\_names <- function(data){  
 data[data == "Bolivia (Plurinational State of)"] <- "Bolivia"  
 data[data == "Democratic People's Republic of Korea"] <- "North Korea"  
 data[data == "Cote d'Ivoire"] <- "Côte d’Ivoire"  
 data[data == "CÃ´te d'Ivoire"] <- "Côte d’Ivoire"  
 data[data == "Czech Republic"] <- "Czechia"  
 data[data == "Egypt, Arab Rep."] <- "Egypt"  
 data[data == "Swaziland"] <- "Eswatini"  
 data[data == "Gambia, The"] <- "Gambia"  
 data[data == "Guinea Bissau"] <- "Guinea-Bissau"  
 data[data == "Iran (Islamic Republic of)"] <- "Iran"  
 data[data == "Iran, Islamic Rep."] <- "Iran"  
 data[data == "Kyrgyz Republic"] <- "Kyrgyzstan"  
 data[data == "Lao People's Democratic Republic"] <- "Laos"  
 data[data == "Lao PDR"] <- "Laos"  
 data[data == "Micronesia, Fed. Sts."] <- "Micronesia"  
 data[data == "Republic of Korea"] <- "South Korea"  
 data[data == "Korea, Rep."] <- "South Korea"  
 data[data == "Republic of Moldova"] <- "Moldova"  
 data[data == "Macedonia (TFYR)"] <- "North Macedonia"  
 data[data == "The former Yugoslav Republic of Macedonia"] <- "North Macedonia"  
 data[data == "Micronesia (Federated States of)"] <- "Micronesia"  
 data[data == "Micronesia (country)"] <- "Micronesia"  
 data[data == "Russian Federation"] <- "Russia"  
 data[data == "St. Lucia"] <- "Saint Lucia"  
 data[data == "St. Vincent and the Grenadines"] <- "Saint Vincent and the Grenadines"  
 data[data == "Slovak Republic"] <- "Slovakia"  
 data[data == "Sudan (until 2011)"] <- "Sudan"  
 data[data == "Syrian Arab Republic"] <- "Syria"  
 data[data == "Timor Leste"] <- "Timor-Leste"  
 data[data == "United Kingdom of Great Britain and Northern Ireland"] <- "United Kingdom"  
 data[data == "United States of America"] <- "United States"  
 data[data == "Tanzania"] <- "United Republic of Tanzania"  
 data[data == "Venezuela (Bolivarian Republic of)"] <- "Venezuela"  
 data[data == "Venezuela, RB"] <- "Venezuela"  
 data[data == "Viet Nam"] <- "Vietnam"  
 data[data == "Yemen, Rep."] <- "Yemen"  
 data[data == "Brunei Darussalam"] <- "Brunei"  
 data[data == "Cape Verde"] <- "Cabo Verde"  
 data[data == "Congo, Rep."] <- "Congo"  
 data[data == "Congo, Dem. Rep."] <- "Democratic Republic of Congo"  
 data[data == "Democratic Republic of the Congo"] <- "Democratic Republic of Congo"  
 data[data == "DR Congo"] <- "Democratic Republic of Congo"  
 data[data == "Macedonia"] <- "North Macedonia"  
 data[data == "Timor"] <- "Timor-Leste"  
 data[data == "Bahamas, The"] <- "Bahamas"  
 data[data == "Korea, Dem. People’s Rep."] <- "North Korea"  
  
 return(data)  
}

Next I read in the data for the life expectancy variable, and removed the unnecessary columns. This data frame will act as the base that I joined to the data frames for the other variables.

life <- read.csv("data/life expectancy.csv", header = T, stringsAsFactors = F)  
  
data <- life[,c(1,2,5)]  
data <- rename(data, Country = Location,  
 Year = Period,  
 Life.Expectancy = First.Tooltip)  
data <- clean\_country\_names(data)  
  
print(head(data))

## Country Year Life.Expectancy  
## 1 Afghanistan 2016 62.69  
## 2 Afghanistan 2015 63.18  
## 3 Afghanistan 2014 62.97  
## 4 Afghanistan 2013 62.74  
## 5 Afghanistan 2012 62.21  
## 6 Afghanistan 2011 61.72

Both BMI and obesity are included in the same dataset; however, they were split up by sex, which scatters each record across two rows. In order to fix that, I had to spread the data. I filtered all of the male records into one table and all of the female ones into another. I renamed the columns and joined them into a single table where each record represented one year for one country, with columns for mean male BMI, mean female BMI, the prevalence of obesity in men, and prevalence of obesity in women. Then I was able to take the average of both BMI columns to create the mean BMI for the total population, and then take the average of both obesity prevalence columns to create the mean obesity prevalence for the total population. This method assumes that the countries have equal male and female populations for each year.

bmi <- read.csv("data/bmi.csv", header = T, stringsAsFactors = F)  
bmi <- bmi[,c(1,3,4,5,8)]  
names(bmi)[5] = "Obesity"  
names(bmi)[1] = "Country"  
  
bmi.men <- filter(bmi, Sex == 'Men')  
bmi.men <- rename(bmi.men, BMI.M = Mean.BMI,  
 Obesity.M = Obesity)  
bmi.men <- bmi.men[-2]  
  
bmi.women <- filter(bmi, Sex == 'Women')  
bmi.women <- rename(bmi.women, BMI.W = Mean.BMI,  
 Obesity.W = Obesity)  
bmi.women <- bmi.women[-2]  
  
bmi.both <- left\_join(bmi.men, bmi.women, by = c("Country", "Year"))  
  
bmi.both[,"BMI"] <- rowMeans(bmi.both[,c('BMI.M', 'BMI.W')], na.rm=TRUE)  
bmi.both[,"Obesity"] <-rowMeans(bmi.both[,c('Obesity.M', 'Obesity.W')], na.rm=TRUE)  
  
bmi.both <- bmi.both[,c(1,2,7,8)]  
  
bmi.both <- clean\_country\_names(bmi.both)  
data <- left\_join(data, bmi.both, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424

The alcohol dataset was split into two files to be joined, the first having columns for the years 2000-2009, and the second having columns for the years 2010-2018. After being combined, they were still in the wrong format, as the columns were not different variables, just different values of the same variable: year. In order to clean that, I had to gather the columns into a new pair of variables, putting all of the years into one variable and the alcohol consumption levels into a second variable.

alcohol.00 <- read.csv("data/alcohol 00-09.csv", header = T, skip = 1, check.names = F, stringsAsFactors = F)  
alcohol.00 <- alcohol.00[,-c(2,3)]  
  
alcohol.00 <- gather(alcohol.00, `2009`, `2008`, `2007`, `2006`, `2005`, `2004`, `2003`, `2002`, `2001`, `2000`, key = "Year", value = "Alcohol")  
  
alcohol.10 <- read.csv("data/alcohol 10-18.csv", header = T, skip = 1, check.names = F, stringsAsFactors = F)  
alcohol.10 <- alcohol.10[,-c(2,3)]  
  
alcohol.10 <- gather(alcohol.10, `2018`, `2017`, `2016`, `2015`, `2014`, `2013`, `2012`, `2011`, `2010`, key = "Year", value = "Alcohol")  
  
alcohol <- rbind(alcohol.00, alcohol.10)  
  
alcohol[,"Year"] <- as.numeric(alcohol[,"Year"])  
  
alcohol <- clean\_country\_names(alcohol)  
data <- left\_join(data, alcohol, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity Alcohol  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314 0.02  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362 0.00  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183 0.01  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080 0.01  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575 0.01  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424 0.01

The GDP dataset had the same structure as the alcohol consumption, so I used the same gather process, creating a year variable and a GDP variable.

gdp <- read.csv("data/GDP per capita.csv", header = T, check.names = F, stringsAsFactors = F)  
gdp <- gdp[,c('Country Name', '2018', '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010',  
 '2009', '2008', '2007', '2006', '2005', '2004', '2003', '2002', '2001', '2000')]  
  
gdp <- gather(gdp, `2018`, `2017`, `2016`, `2015`, `2014`, `2013`, `2012`, `2011`, `2010`, `2009`,  
 `2008`, `2007`, `2006`, `2005`, `2004`, `2003`, `2002`, `2001`, `2000`, key = "Year", value = "GDP")  
  
gdp[,"Year"] <- as.numeric(gdp[,"Year"])  
names(gdp)[1] <- "Country"  
  
gdp <- clean\_country\_names(gdp)  
data <- left\_join(data, gdp, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity Alcohol GDP  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314 0.02 547.2281  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362 0.00 578.4664  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183 0.01 613.8567  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080 0.01 637.1655  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575 0.01 641.8715  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424 0.01 591.1628

The HIV/AIDS dataset was already in the required format, so I only had to remove the unnecessary columns.

hiv <- read.csv("data/deaths-and-new-cases-of-hiv.csv", header = T, stringsAsFactors = F)  
hiv <- hiv[,c(1,3,5,6)]  
names(hiv) <- c("Country", "Year", "HIV.Prevalence", "HIV.Incidence")  
  
hiv <- clean\_country\_names(hiv)  
data <- left\_join(data, hiv, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity Alcohol GDP  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314 0.02 547.2281  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362 0.00 578.4664  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183 0.01 613.8567  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080 0.01 637.1655  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575 0.01 641.8715  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424 0.01 591.1628  
## HIV.Prevalence HIV.Incidence  
## 1 4133.235 790.0594  
## 2 3630.161 679.0092  
## 3 3200.788 584.7016  
## 4 2788.420 495.3635  
## 5 2429.128 418.4731  
## 6 2164.778 362.4356

The diabetes dataset had the same structure as the BMI and obesity dataset, so I used the same process to restructure the age-standardized diabetes prevalence data.

diabetes <- read.csv("data/diabetes.csv", header = T, stringsAsFactors = F)  
diabetes <- diabetes[,c(1,3,4,5)]  
names(diabetes)[4] = "Diabetes"  
names(diabetes)[1] = "Country"  
  
diabetes.men <- filter(diabetes, Sex == 'Men')  
diabetes.men <- rename(diabetes.men, Diabetes.M = Diabetes)  
diabetes.men <- diabetes.men[-2]  
  
diabetes.women <- filter(diabetes, Sex == 'Women')  
diabetes.women <- rename(diabetes.women, Diabetes.W = Diabetes)  
diabetes.women <- diabetes.women[-2]  
  
diabetes.both <- left\_join(diabetes.men, diabetes.women, by = c("Country", "Year"))  
  
diabetes.both[,"Diabetes"] <- rowMeans(diabetes.both[,c('Diabetes.M', 'Diabetes.W')], na.rm=TRUE)  
  
diabetes.both <- diabetes.both[,c(1,2,5)]  
  
diabetes.both <- clean\_country\_names(diabetes.both)  
data <- left\_join(data, diabetes.both, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity Alcohol GDP  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314 0.02 547.2281  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362 0.00 578.4664  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183 0.01 613.8567  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080 0.01 637.1655  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575 0.01 641.8715  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424 0.01 591.1628  
## HIV.Prevalence HIV.Incidence Diabetes  
## 1 4133.235 790.0594 NA  
## 2 3630.161 679.0092 NA  
## 3 3200.788 584.7016 0.1190786  
## 4 2788.420 495.3635 0.1162422  
## 5 2429.128 418.4731 0.1133741  
## 6 2164.778 362.4356 0.1105226

For the immunizations, I anticipated that using all ten of them would result in serious multicollinearity problems, so I decided to compute an average immunization coverage value for all ten immunizations. In order to do so, I joined the datasets for each vaccine into one immunization table. Then I created a new column that was the mean of the values in all the other columns.

bcg <- read.csv("data/bcg.csv", header = T, stringsAsFactors = F)  
bcg <- bcg[,c(1,3,4)]  
names(bcg) <- c("Country", "Year", "BCG")  
  
bcg <- subset(bcg, Year >= 2000)  
  
immunization <- bcg  
  
  
add\_vaccine <- function(immunization, file.name, vaccine.name){  
 vaccine <- read.csv(file.name, header = T, stringsAsFactors = F)  
 vaccine <- vaccine[,c(1,3,4)]  
 names(vaccine) <- c("Country", "Year", vaccine.name)  
   
 vaccine <- subset(vaccine, Year >= 2000)  
  
 immunization <- full\_join(immunization, vaccine, by = c("Country", "Year"))  
 return(immunization)  
}  
  
immunization <- add\_vaccine(immunization, "data/dtp3.csv", "DTP3")  
immunization <- add\_vaccine(immunization, "data/hepb3.csv", "HepB3")  
immunization <- add\_vaccine(immunization, "data/hib3.csv", "Hib3")  
immunization <- add\_vaccine(immunization, "data/mcv1.csv", "MCV1")  
immunization <- add\_vaccine(immunization, "data/mcv2.csv", "MCV2")  
immunization <- add\_vaccine(immunization, "data/pab.csv", "PAB")  
immunization <- add\_vaccine(immunization, "data/pcv3.csv", "PCV3")  
immunization <- add\_vaccine(immunization, "data/pol3.csv", "Pol3")  
immunization <- add\_vaccine(immunization, "data/rotac.csv", "RotaC")  
  
print(head(immunization))

## Country Year BCG DTP3 HepB3 Hib3 MCV1 MCV2 PAB PCV3 Pol3 RotaC  
## 1 Afghanistan 2019 78 66 66 66 64 39 68 65 73 55  
## 2 Afghanistan 2018 78 66 66 66 64 39 68 65 73 45  
## 3 Afghanistan 2017 78 66 66 66 64 39 70 65 73 NA  
## 4 Afghanistan 2016 78 66 66 66 64 39 70 65 73 NA  
## 5 Afghanistan 2015 76 65 65 65 63 39 70 65 69 NA  
## 6 Afghanistan 2014 74 62 62 62 60 39 70 40 62 NA

immunization.avg <- immunization  
immunization.avg[,"Immunization"] <- rowMeans(immunization.avg[,c('BCG', 'DTP3', 'HepB3', 'Hib3','Pol3', 'MCV1',  
 'MCV2', 'PAB', 'PCV3', 'RotaC')], na.rm=TRUE)  
  
immunization.avg <- immunization.avg[,c("Country", "Year", "Immunization")]  
  
data <- left\_join(data, immunization.avg, by = c("Country", "Year"))  
print(head(data))

## Country Year Life.Expectancy BMI Obesity Alcohol GDP  
## 1 Afghanistan 2016 62.69 23.45830 0.05681314 0.02 547.2281  
## 2 Afghanistan 2015 63.18 23.35209 0.05396362 0.00 578.4664  
## 3 Afghanistan 2014 62.97 23.24576 0.05124183 0.01 613.8567  
## 4 Afghanistan 2013 62.74 23.13956 0.04864080 0.01 637.1655  
## 5 Afghanistan 2012 62.21 23.03342 0.04615575 0.01 641.8715  
## 6 Afghanistan 2011 61.72 22.92832 0.04378424 0.01 591.1628  
## HIV.Prevalence HIV.Incidence Diabetes Immunization  
## 1 4133.235 790.0594 NA 65.22222  
## 2 3630.161 679.0092 NA 64.11111  
## 3 3200.788 584.7016 0.1190786 59.00000  
## 4 2788.420 495.3635 0.1162422 60.62500  
## 5 2429.128 418.4731 0.1133741 62.25000  
## 6 2164.778 362.4356 0.1105226 62.25000

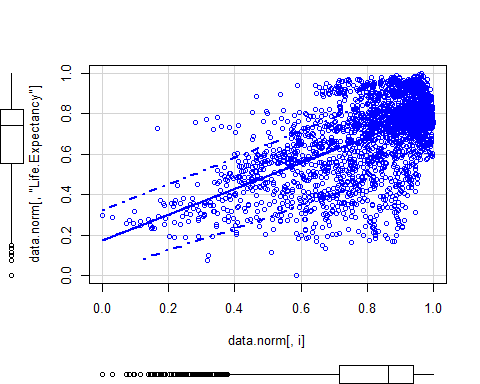
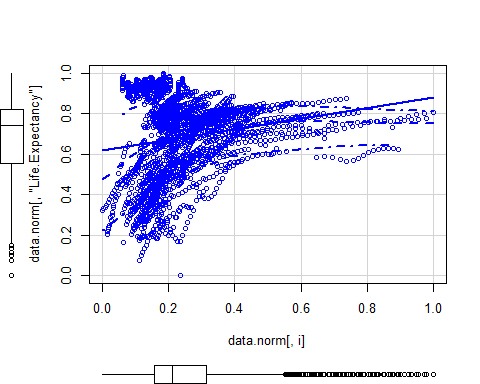
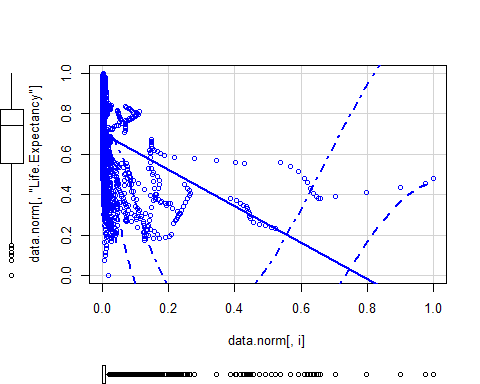
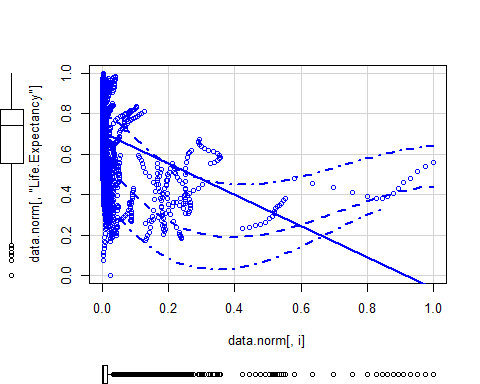
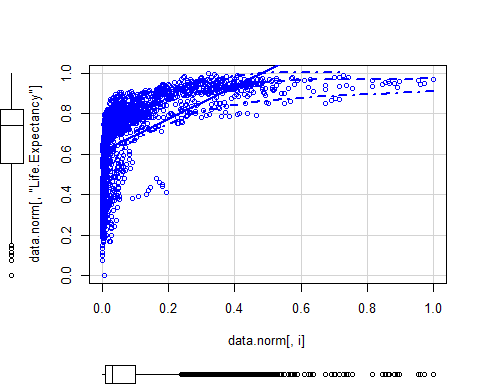
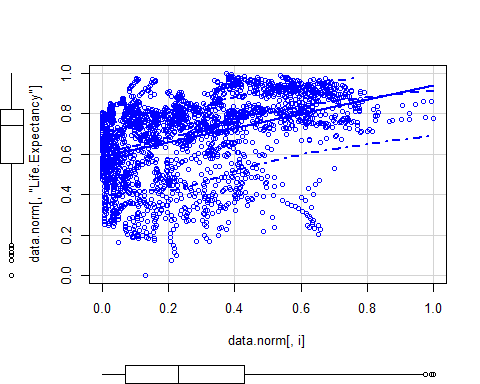
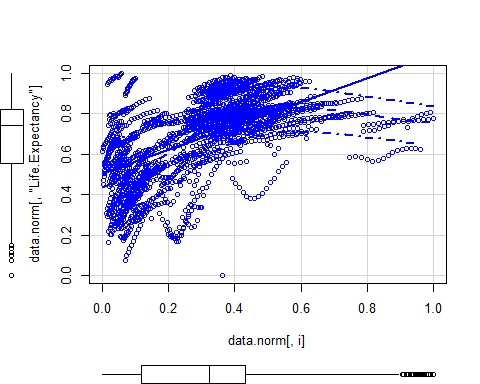
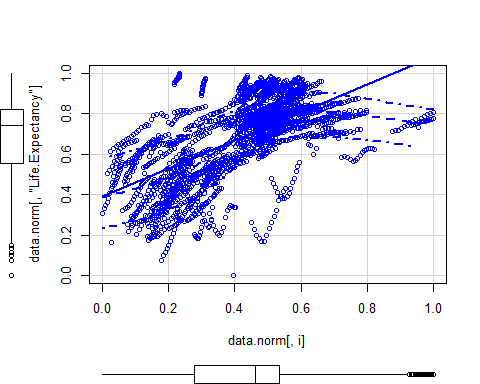
Since linear regression cannot handle missing values, I removed all records with at least one missing value. Next I used min/max normalization on the data since the scales are different for most of the variables.

data <- data[,c("Life.Expectancy", "BMI", "Obesity", "Alcohol", "GDP", "HIV.Prevalence", "HIV.Incidence",   
 "Diabetes", "Immunization")]  
  
  
data <- data[complete.cases(data),]  
  
normalize <- function(x){  
 return((x- min(x)) /(max(x)-min(x)))  
}  
  
data.norm <- lapply(data, normalize)  
  
data.norm <- as.data.frame(data.norm)

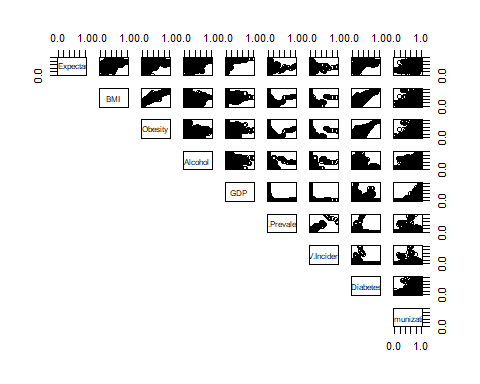
## Exploring the Data

I created scatterplots with life expectancy on the left axis for each of my regressors. I also created a scatterplot matrix of all variables to examine how each variable interacted with each of the others.

for(i in 2:ncol(data.norm)){  
 scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,i])  
}

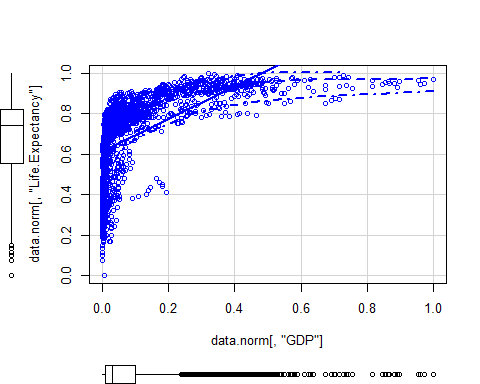


pairs(data.norm, lower.panel = NULL)

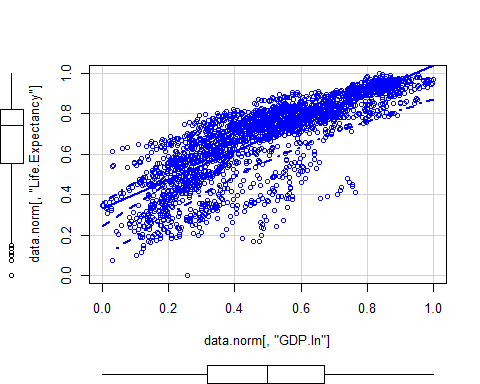


I noticed that the fourth graph (life expectancy versus GDP) had a sharp bend, indicating that the relationship could be exponential. I created a new variable to be the natural log of the original, non-normalized GDP variable. I then normalized this new natural log of GDP variable. I then made a scatterplot with life expectancy on the y-axis and the natural log of GDP on the x-axis and compared it to the original. The graph of life expectancy versus natural log of GDP did not have the bend in it and closely resembled a strong linear relationship. Therefore, I used the natual log of GDP for the regression.

data.norm[,"GDP.ln"] <- as.data.frame(normalize(log(data[,"GDP"])))  
  
scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"GDP"])

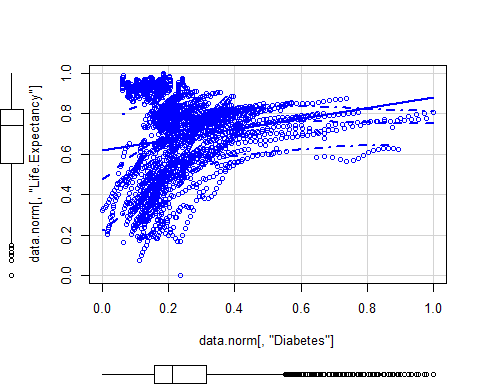


scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"GDP.ln"])

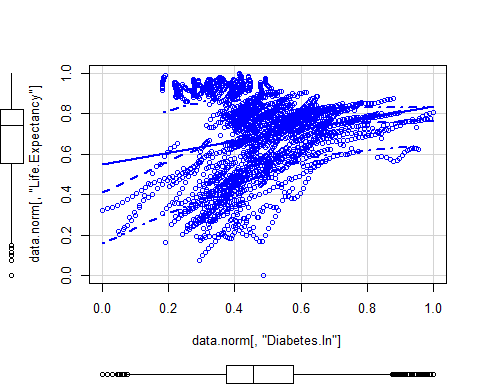


I noticed that the seventh graph (life expectancy versus diabetes) had a bend, similar to the bend in GDP. I carried out the same process on the diabetes variable to create the natural log of Diabetes. The graph of life expectancy versus natural log of diabetes appears to be more linear than that of the original. Therefore, I used the natural log of diabetes for the regression.

data.norm[,"Diabetes.ln"] <- as.data.frame(normalize(log(data[,"Diabetes"])))  
  
scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"Diabetes"])

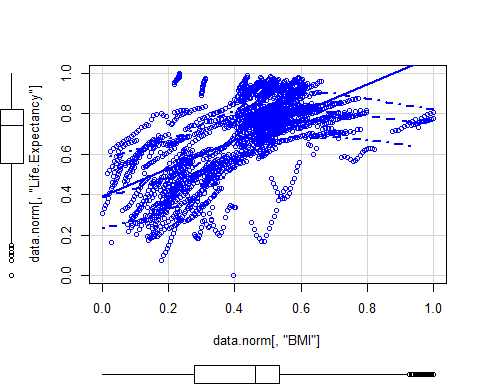


scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"Diabetes.ln"])

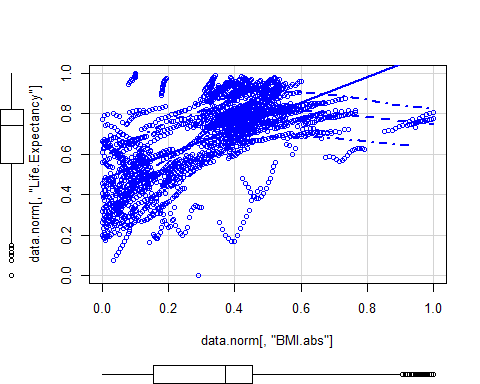


I noticed that the first graph (life expectancy versus BMI) was curved and that the highest values were in the middle, indicating that the relationship may not be linear. I took the median value of the healthy BMI range, 21.7, and subtracted it from the original, non-normalized BMI column, took the absolute value, and normalized it. I then created a scatterplot with life expectancy on the y-axis and the new absolute value BMI on the x-axis and compared it to the original. The graph of life expectancy versus absolute value BMI appeared only slightly more linear than the original BMI graph, if anything. Therefore, I did not use the absolute value of BMI for the regression.

data.norm[,"BMI.abs"] <- as.data.frame(normalize(abs(data[,"BMI"] - 21.7)))  
  
scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"BMI"])



scatterplot(y=data.norm[,"Life.Expectancy"], x=data.norm[,"BMI.abs"])



data.norm <- subset(data.norm, select = -BMI.abs)

I randomly split the data into a test set and a train set.

set.seed(43)  
data.split = sort(sample(nrow(data.norm), nrow(data.norm)\*.8))  
  
data.train = data.norm[data.split,]  
data.test = data.norm[-data.split,]

## Finding the Best Predictors

For each of the models, I calculated the PRESS statistic, and variance inflation factors for the model. Then I tested for normality. I created a function to perform these tests.

model\_info <- function(model){  
 print(summary(model))  
  
 print(paste("PRESS:", PRESS(model), sep=' '))  
}

I created a model named baseline with all of the original regressors. This baseline does not include the modified regressors, such as natural log of GDP. This baseline will be used to compare to the models with the best predictors.

baseline <- lm(Life.Expectancy ~ GDP + Immunization + HIV.Prevalence + BMI + HIV.Incidence + Diabetes + Alcohol + Obesity, data.train)  
model\_info(baseline)

##   
## Call:  
## lm(formula = Life.Expectancy ~ GDP + Immunization + HIV.Prevalence +   
## BMI + HIV.Incidence + Diabetes + Alcohol + Obesity, data = data.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.56436 -0.06781 0.01548 0.07801 0.30287   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.24088 0.01351 17.832 < 0.0000000000000002 \*\*\*  
## GDP 0.46902 0.02266 20.700 < 0.0000000000000002 \*\*\*  
## Immunization 0.31333 0.01689 18.548 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.21190 0.08465 -2.503 0.01239 \*   
## BMI 0.39456 0.04591 8.593 < 0.0000000000000002 \*\*\*  
## HIV.Incidence -0.20021 0.10143 -1.974 0.04855 \*   
## Diabetes -0.08989 0.03375 -2.663 0.00780 \*\*   
## Alcohol 0.04915 0.01748 2.813 0.00496 \*\*   
## Obesity 0.01293 0.03718 0.348 0.72798   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1151 on 1936 degrees of freedom  
## Multiple R-squared: 0.6595, Adjusted R-squared: 0.6581   
## F-statistic: 468.7 on 8 and 1936 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 25.9595633206265"

For Model 1, I used all regressors, but used natural log of GDP instead of the original GDP variable.

model.1 <- lm(Life.Expectancy ~ GDP.ln + Immunization + HIV.Prevalence + BMI + HIV.Incidence + Diabetes + Alcohol + Obesity, data.train)  
model\_info(model.1)

##   
## Call:  
## lm(formula = Life.Expectancy ~ GDP.ln + Immunization + HIV.Prevalence +   
## BMI + HIV.Incidence + Diabetes + Alcohol + Obesity, data = data.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.53682 -0.04934 0.01206 0.06723 0.28185   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.20170 0.01199 16.827 < 0.0000000000000002 \*\*\*  
## GDP.ln 0.52770 0.01595 33.090 < 0.0000000000000002 \*\*\*  
## Immunization 0.23950 0.01518 15.773 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.19701 0.07477 -2.635 0.00848 \*\*   
## BMI 0.20612 0.04101 5.027 0.000000545 \*\*\*  
## HIV.Incidence -0.17948 0.08959 -2.003 0.04528 \*   
## Diabetes -0.14269 0.02907 -4.909 0.000000994 \*\*\*  
## Alcohol -0.06955 0.01610 -4.321 0.000016312 \*\*\*  
## Obesity 0.01658 0.03245 0.511 0.60936   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1017 on 1936 degrees of freedom  
## Multiple R-squared: 0.7344, Adjusted R-squared: 0.7333   
## F-statistic: 669 on 8 and 1936 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 20.2313750603823"

For Model 2, I used all regressors, but used natural log of diabetes instead of the original diabetes variable.

model.2 <- lm(Life.Expectancy ~ GDP + Immunization + HIV.Prevalence + BMI + HIV.Incidence + Diabetes.ln + Alcohol + Obesity, data.train)  
model\_info(model.2)

##   
## Call:  
## lm(formula = Life.Expectancy ~ GDP + Immunization + HIV.Prevalence +   
## BMI + HIV.Incidence + Diabetes.ln + Alcohol + Obesity, data = data.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.55399 -0.06697 0.01681 0.07885 0.27877   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.208568 0.015485 13.469 < 0.0000000000000002 \*\*\*  
## GDP 0.507261 0.023200 21.865 < 0.0000000000000002 \*\*\*  
## Immunization 0.311829 0.016901 18.450 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.218370 0.084604 -2.581 0.00992 \*\*   
## BMI 0.279214 0.047800 5.841 0.000000006059 \*\*\*  
## HIV.Incidence -0.200459 0.101388 -1.977 0.04817 \*   
## Diabetes.ln 0.092318 0.031030 2.975 0.00296 \*\*   
## Alcohol 0.109953 0.017195 6.394 0.000000000201 \*\*\*  
## Obesity -0.002033 0.036772 -0.055 0.95592   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1151 on 1936 degrees of freedom  
## Multiple R-squared: 0.6598, Adjusted R-squared: 0.6584   
## F-statistic: 469.3 on 8 and 1936 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 25.9349106458725"

For Model 2, I used all regressors, but used both natural log of diabetes and natural log of GDP instead of the original diabetes and GDP variables.

model.3 <- lm(Life.Expectancy ~ GDP.ln + Immunization + HIV.Prevalence + BMI + HIV.Incidence + Diabetes.ln + Alcohol + Obesity, data.train)  
model\_info(model.3)

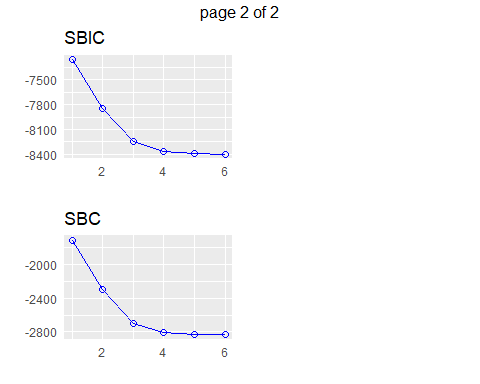
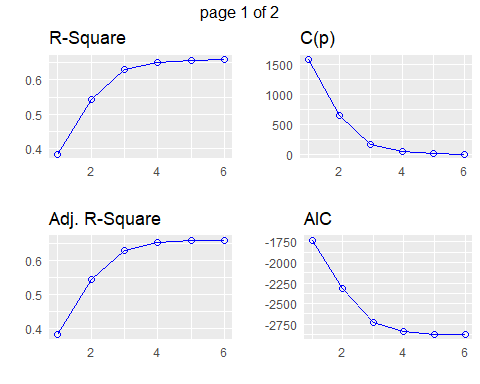
##   
## Call:  
## lm(formula = Life.Expectancy ~ GDP.ln + Immunization + HIV.Prevalence +   
## BMI + HIV.Incidence + Diabetes.ln + Alcohol + Obesity, data = data.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.53339 -0.05005 0.01218 0.06905 0.27740   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.205428 0.013706 14.988 < 0.0000000000000002 \*\*\*  
## GDP.ln 0.534900 0.015929 33.579 < 0.0000000000000002 \*\*\*  
## Immunization 0.241801 0.015260 15.845 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.201823 0.075104 -2.687 0.00727 \*\*   
## BMI 0.173461 0.042665 4.066 0.0000498 \*\*\*  
## HIV.Incidence -0.179243 0.090005 -1.991 0.04657 \*   
## Diabetes.ln -0.065274 0.026071 -2.504 0.01237 \*   
## Alcohol -0.046108 0.016193 -2.847 0.00445 \*\*   
## Obesity -0.005165 0.032422 -0.159 0.87345   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1022 on 1936 degrees of freedom  
## Multiple R-squared: 0.7319, Adjusted R-squared: 0.7308   
## F-statistic: 660.7 on 8 and 1936 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 20.4178041741081"

I made Models 4-7 to replicate the above, but used stepwise selection to eliminate unnecessary regressors. Model 4 is similar to the baseline, but obesity and HIV incidence were not used.

stepwise.4 <- ols\_step\_forward\_p(baseline, pent=.04)  
stepwise.4

##   
## Selection Summary   
## -----------------------------------------------------------------------------------  
## Variable Adj.   
## Step Entered R-Square R-Square C(p) AIC RMSE   
## -----------------------------------------------------------------------------------  
## 1 BMI 0.3829 0.3826 1567.7268 -1735.2953 0.1547   
## 2 GDP 0.5437 0.5433 655.1416 -2320.6806 0.1331   
## 3 Immunization 0.6304 0.6298 164.3846 -2728.4128 0.1198   
## 4 HIV.Prevalence 0.6516 0.6509 45.7295 -2841.4230 0.1163   
## 5 Alcohol 0.6576 0.6567 14.0030 -2872.8264 0.1154   
## 6 Diabetes 0.6588 0.6577 9.1434 -2877.6911 0.1152   
## -----------------------------------------------------------------------------------

plot(stepwise.4, which=1)



model.4 <- stepwise.4$model  
model\_info(model.4)

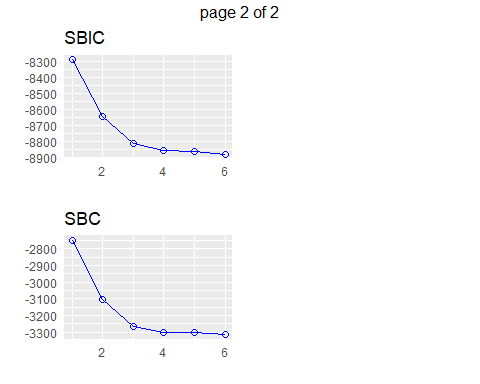
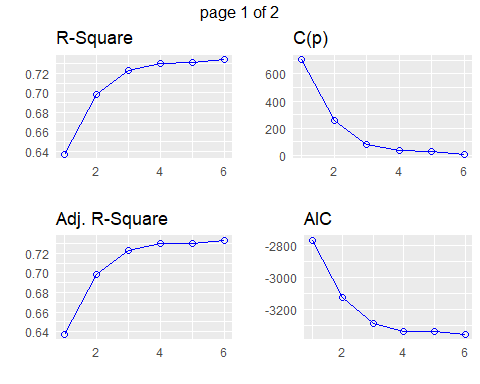
##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.56249 -0.06670 0.01578 0.07808 0.31413   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.23829 0.01299 18.344 < 0.0000000000000002 \*\*\*  
## BMI 0.40387 0.02917 13.846 < 0.0000000000000002 \*\*\*  
## GDP 0.47214 0.02214 21.326 < 0.0000000000000002 \*\*\*  
## Immunization 0.31605 0.01673 18.896 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.36577 0.03178 -11.509 < 0.0000000000000002 \*\*\*  
## Alcohol 0.04763 0.01745 2.730 0.00639 \*\*   
## Diabetes -0.08746 0.03341 -2.618 0.00892 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1152 on 1938 degrees of freedom  
## Multiple R-squared: 0.6588, Adjusted R-squared: 0.6577   
## F-statistic: 623.5 on 6 and 1938 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 25.9298390062755"

Model 5 is similar to Model 1, but obesity and HIV incidence were not used.

stepwise.5 <- ols\_step\_forward\_p(model.1, pent=.04)  
stepwise.5

##   
## Selection Summary   
## ----------------------------------------------------------------------------------  
## Variable Adj.   
## Step Entered R-Square R-Square C(p) AIC RMSE   
## ----------------------------------------------------------------------------------  
## 1 GDP.ln 0.6372 0.6370 703.3393 -2768.3662 0.1186   
## 2 Immunization 0.6985 0.6982 258.5825 -3126.3145 0.1082   
## 3 HIV.Prevalence 0.7233 0.7229 79.7280 -3291.3540 0.1037   
## 4 BMI 0.7301 0.7295 32.4010 -3337.5180 0.1024   
## 5 Diabetes 0.7310 0.7303 27.2846 -3342.5661 0.1023   
## 6 Alcohol 0.7337 0.7329 9.4577 -3360.3386 0.1018   
## ----------------------------------------------------------------------------------

plot(stepwise.5, which=1)



model.5 <- stepwise.5$model  
model\_info(model.5)

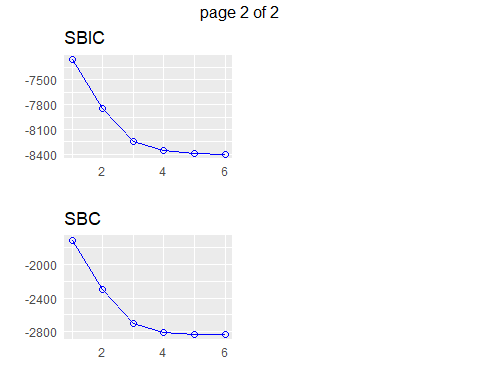
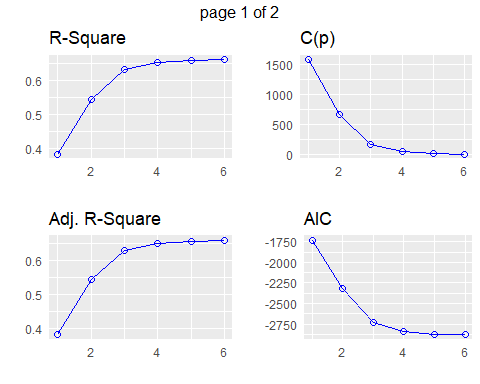
##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.54039 -0.04900 0.01188 0.06745 0.28459   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.19866 0.01147 17.326 < 0.0000000000000002 \*\*\*  
## GDP.ln 0.52989 0.01577 33.596 < 0.0000000000000002 \*\*\*  
## Immunization 0.24148 0.01502 16.076 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.33473 0.02810 -11.913 < 0.0000000000000002 \*\*\*  
## BMI 0.21902 0.02695 8.128 0.000000000000000771 \*\*\*  
## Diabetes -0.14060 0.02890 -4.865 0.000001236918947270 \*\*\*  
## Alcohol -0.07141 0.01605 -4.450 0.000009078041152990 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1018 on 1938 degrees of freedom  
## Multiple R-squared: 0.7337, Adjusted R-squared: 0.7329   
## F-statistic: 890.1 on 6 and 1938 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 20.2118563507014"

Model 6 is similar to Model 2, but obesity and HIV incidence are not used.

stepwise.6 <- ols\_step\_forward\_p(model.2, pent=.04)  
stepwise.6

##   
## Selection Summary   
## -----------------------------------------------------------------------------------  
## Variable Adj.   
## Step Entered R-Square R-Square C(p) AIC RMSE   
## -----------------------------------------------------------------------------------  
## 1 BMI 0.3829 0.3826 1570.9015 -1735.2953 0.1547   
## 2 GDP 0.5437 0.5433 657.4888 -2320.6806 0.1331   
## 3 Immunization 0.6304 0.6298 166.2859 -2728.4128 0.1198   
## 4 HIV.Prevalence 0.6516 0.6509 47.5217 -2841.4230 0.1163   
## 5 Alcohol 0.6576 0.6567 15.7646 -2872.8264 0.1154   
## 6 Diabetes.ln 0.6591 0.6581 8.9182 -2879.6759 0.1151   
## -----------------------------------------------------------------------------------

plot(stepwise.6, which=1)



model.6 <- stepwise.6$model  
model\_info(model.6)

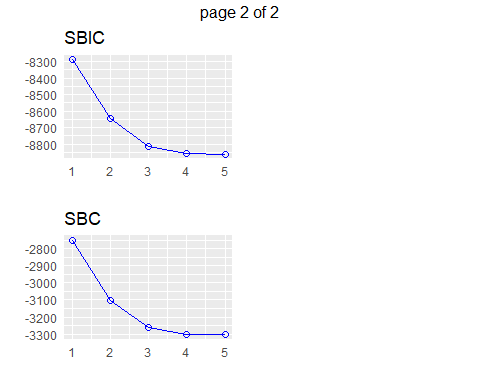
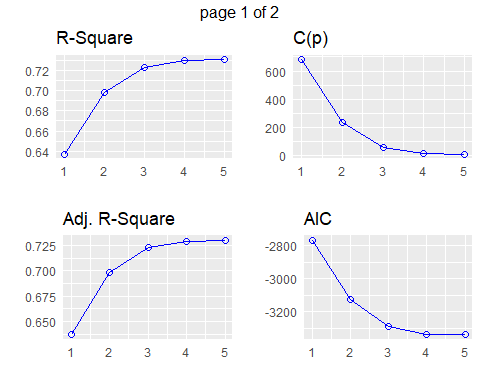
##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.55339 -0.06700 0.01637 0.07866 0.29015   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.20752 0.01512 13.725 < 0.0000000000000002 \*\*\*  
## BMI 0.27464 0.02855 9.619 < 0.0000000000000002 \*\*\*  
## GDP 0.50835 0.02284 22.254 < 0.0000000000000002 \*\*\*  
## Immunization 0.31521 0.01673 18.846 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.37319 0.03172 -11.764 < 0.0000000000000002 \*\*\*  
## Alcohol 0.10868 0.01700 6.394 0.000000000202 \*\*\*  
## Diabetes.ln 0.09229 0.03104 2.973 0.00299 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1151 on 1938 degrees of freedom  
## Multiple R-squared: 0.6591, Adjusted R-squared: 0.6581   
## F-statistic: 624.5 on 6 and 1938 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 25.9050040794056"

Model 7 is similar to Model 3, but alcohol, obesity, and natural log of diabetes are unused.

stepwise.7 <- ols\_step\_forward\_p(model.3, pent=.04)  
stepwise.7

##   
## Selection Summary   
## ----------------------------------------------------------------------------------  
## Variable Adj.   
## Step Entered R-Square R-Square C(p) AIC RMSE   
## ----------------------------------------------------------------------------------  
## 1 GDP.ln 0.6372 0.6370 679.2910 -2768.3662 0.1186   
## 2 Immunization 0.6985 0.6982 238.5971 -3126.3145 0.1082   
## 3 HIV.Prevalence 0.7233 0.7229 61.3873 -3291.3540 0.1037   
## 4 BMI 0.7301 0.7295 14.5090 -3337.5180 0.1024   
## 5 HIV.Incidence 0.7307 0.7300 12.1010 -3339.9207 0.1023   
## ----------------------------------------------------------------------------------

plot(stepwise.7, which=1)



model.7 <- stepwise.7$model  
model\_info(model.7)

##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.53044 -0.05134 0.01052 0.07058 0.27345   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.18852 0.01125 16.754 < 0.0000000000000002 \*\*\*  
## GDP.ln 0.52510 0.01382 38.007 < 0.0000000000000002 \*\*\*  
## Immunization 0.23927 0.01514 15.801 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.20568 0.07487 -2.747 0.00607 \*\*   
## BMI 0.12311 0.01732 7.109 0.00000000000163 \*\*\*  
## HIV.Incidence -0.18813 0.08975 -2.096 0.03619 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1023 on 1939 degrees of freedom  
## Multiple R-squared: 0.7307, Adjusted R-squared: 0.73   
## F-statistic: 1052 on 5 and 1939 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 20.4573188887185"

## Selecting the Best Model

I performed cross validation on my models, and compared the results.

folds <- cvFolds(nrow(data.train), K = 10, R =100)  
cvfit0 <- cvLm(baseline, cost = rtmspe, folds = folds)  
cvfit1 <- cvLm(model.1, cost = rtmspe, folds = folds)  
cvfit2 <- cvLm(model.2, cost = rtmspe, folds = folds)  
cvfit3 <- cvLm(model.3, cost = rtmspe, folds = folds)  
cvfit4 <- cvLm(lm(model.4$terms, data.train), cost = rtmspe, folds = folds)  
cvfit5 <- cvLm(lm(model.5$terms, data.train), cost = rtmspe, folds = folds)  
cvfit6 <- cvLm(lm(model.6$terms, data.train), cost = rtmspe, folds = folds)  
cvfit7 <- cvLm(lm(model.7$terms, data.train), cost = rtmspe, folds = folds)  
cvFits <- cvSelect(Baseline = cvfit0, Model1 = cvfit1, Model2 = cvfit2, Model3 = cvfit3, Model4 = cvfit4,  
 Model5 = cvfit5, Model6 = cvfit6, Model7 = cvfit7)  
cvFits

##   
## 10-fold CV results:  
## Fit CV  
## 1 Baseline 0.06568576  
## 2 Model1 0.05452070  
## 3 Model2 0.06752637  
## 4 Model3 0.05555647  
## 5 Model4 0.06558756  
## 6 Model5 0.05439383  
## 7 Model6 0.06738395  
## 8 Model7 0.05581415  
##   
## Best model:  
## CV   
## "Model5"

Model 5 performed best in the cross validation, therefore it is the final model I selected. I then ran the test data through Model 5, and recorded several performance metrics.

test.labels <- data.test[,1]  
test <- data.test[,-1]  
pred <- predict(model.5, test)  
  
mae <- mae(test.labels, pred)  
print(paste("MAE", mae, sep=' '))

## [1] "MAE 0.0736231122483488"

mse <- mse(test.labels, pred)  
print(paste("MSE", mse, sep=' '))

## [1] "MSE 0.0100236718723161"

rmse <- rmse(test.labels, pred)  
print(paste("RMSE", rmse, sep=' '))

## [1] "RMSE 0.10011828939967"

mape <- mape(test.labels, pred)  
print(paste("MAPE", mape, sep=' '))

## [1] "MAPE 0.153815733230571"

smape <- smape(test.labels, pred)\*100  
print(paste("SMAPE", smape, sep=' '))

## [1] "SMAPE 13.1063273275383"

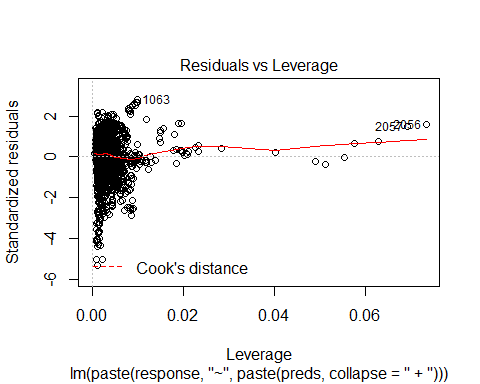
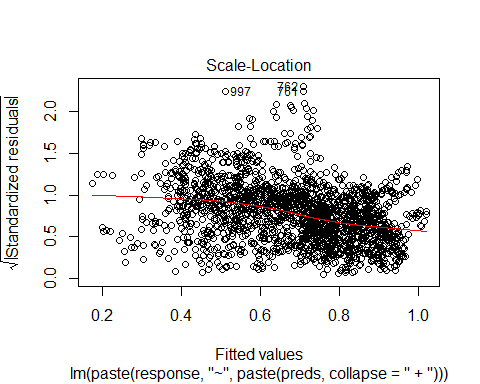
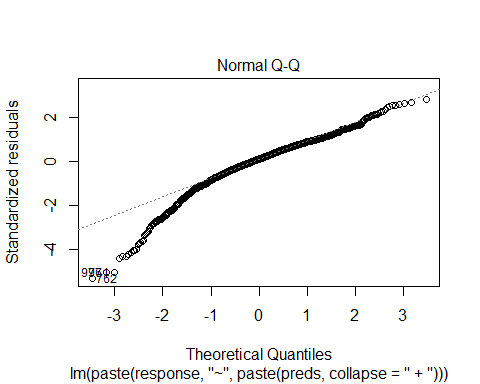
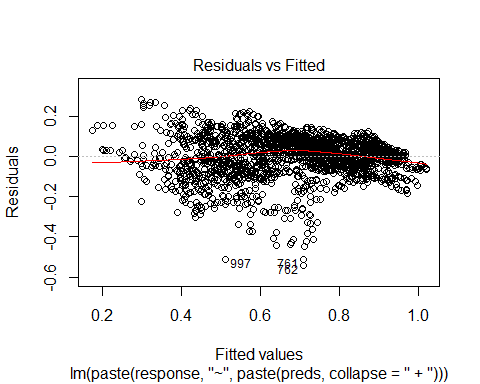
rsq <- function(x, y){  
 return(cor(x, y) ^ 2)  
}  
  
rsq <- rsq(test.labels, pred)  
print(paste("RSQ", rsq, sep=' '))

## [1] "RSQ 0.745315803860855"

model\_info(model.5)

##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.54039 -0.04900 0.01188 0.06745 0.28459   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.19866 0.01147 17.326 < 0.0000000000000002 \*\*\*  
## GDP.ln 0.52989 0.01577 33.596 < 0.0000000000000002 \*\*\*  
## Immunization 0.24148 0.01502 16.076 < 0.0000000000000002 \*\*\*  
## HIV.Prevalence -0.33473 0.02810 -11.913 < 0.0000000000000002 \*\*\*  
## BMI 0.21902 0.02695 8.128 0.000000000000000771 \*\*\*  
## Diabetes -0.14060 0.02890 -4.865 0.000001236918947270 \*\*\*  
## Alcohol -0.07141 0.01605 -4.450 0.000009078041152990 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1018 on 1938 degrees of freedom  
## Multiple R-squared: 0.7337, Adjusted R-squared: 0.7329   
## F-statistic: 890.1 on 6 and 1938 DF, p-value: < 0.00000000000000022  
##   
## [1] "PRESS: 20.2118563507014"

plot(model.5)



print("VIF:")

## [1] "VIF:"

print(vif(model.5))

## GDP.ln Immunization HIV.Prevalence BMI Diabetes   
## 2.320338 1.391657 1.127257 4.320755 3.415292   
## Alcohol   
## 2.406942

ols\_test\_normality(model.5)

## -----------------------------------------------  
## Test Statistic pvalue   
## -----------------------------------------------  
## Shapiro-Wilk 0.9486 0.0000   
## Kolmogorov-Smirnov 0.0687 0.0000   
## Cramer-von Mises 533.5365 0.0000   
## Anderson-Darling 20.4093 0.0000   
## -----------------------------------------------

## Results

In the final model, every regressor estimate has a p-value < .05, indicating that they are all significant at the 95% confidence level. The model had a symmetric mean absolute percentage error on the test data of 13.11, indicating that on average, the model is off by 13.11% on data it has never seen. The model achieved an R2 of .7453, indicating the model accounts for about 74.53% of the data variation.

The variance inflation factors for Model 5, are all below 5, indicating that there is no serious multicollinearity in the model.

The Normal Q-Q plot for the model is very close to normal on the right side but deviates significantly on the left end. The residuals versus fitted plot indicates that the residuals do not have constant variance. The left side of the plot is more skewed towards positive residuals, the middle is more skewed towards negative residuals, and the right side has less variation. These plots indicate that while the model performs pretty well, it is not exactly adequate. Further, I performed four different tests for normality, Shapiro-Wilk, Kolmogorov-Smirnov, Cramer-von Mises, and Anderson-Darling. The p-values for all four tests were 0; therefore, we reject the null hypothesis that the data is normal. This rejection further indicates that the model is not exactly adequate for the data, even though it performs pretty well.