Predicting International Happiness Scores

Alex Bender

April 17, 2019

Introduction

The objective of this project is to find the most important factors in predicting (and increasing) the happiness score of a country.I am hoping to provide a data-driven playbook that will help decision makers know what's important in regards to hppiness score, and develop and stretch my skills in all facets of the Data Science pipeline in an area of immense interest for me.

The dataset descriptions are:

Countries of the World - This data comes from the US CIA. It is general characteristsics on different nations of the world. Some columns included in the data are Region, Population, Area (sq. mi.), Pop. Density (per sq. mi.), Coastline (coast/area ratio), Net migration, Arable (%), Crops (%), Other (%), Climate.

UN Human Development Data - This comes from the UN's 2015 Human Development report, which was used to calculate the Human Development Index. The datasets measures status of different nations in different metrics of human development. Some columns included in the data are Life Expectancy at Birth, Expected Years of Education, Mean Years of Education, Gross National Income (GNI) per Capita, GNI per Capita Rank Minus HDI Rank.

World Happiness - The World Happiness Report was released at the United Nations at an event celebrating International Day of Happiness on March 20th. The report continues to gain global recognition. Happiness Score is based on the World Happiness Report which includes GDP per Capita, Family, Life Expectancy, Freedom, Generosity, Trust Government Corruption, etc.

I am going to explore world happiness(WH), human development(HDI), and country characteristics(CC) data in order to derive interesting insights. I will be operating on the assumption of using the HDI data from 2014, WH data from 2016, and CC data from as recent as 2017. Though the data is not all from the same time period, I will be assuming that the variation between the few years won't be significant enough to skew results significantly.

My overall plan of action:

- I am going to use the 2016 World Happiness data as my basis for dependent variables.

- I plan to do numeric prediction/regression by predicting the happiness score of a nation using the happiness score column from the wh2016 data. I will be exploring multiple linear regression, regression tree, neural network, and kNN models.
- As my independent variables I will be using the Human Development Index and Country Characteristics data in order to attempt to predict the target/response variables from the world happiness data.
- I won't be using the additional features in the World Happiness data as predictors for two reasons. 1) The world happiness score is a direct calculation from these features, so I don't want to risk overfitting by the model just memorizing the calculation essentially. 2) I want to derive novel insights from a various set of predictors from the other two datasets.

Data Understanding and Preparation

The loaded data has different numbers of rows, meaning that different countries are included in the different datasets. This will be reconcilied by only including the countries located in the dataset with the least amount of countries (world happiness 2016). Since my analysis relies on merging the unqiue nation metrics from the various datasets, it would be wise to only included the countries with full data. Still, this leaves us with over 150 countries, which is enough to run both numeric prediction (regression) and classification data mining tasks.The data set is a similar size to the built in Iris dataset, with more dimensions, so it should still be fine. I will combat this small amount of data with the use of k-fold Cross-validation.

Now we have to merge the datasets properly. First we have to match up the rows based on Country name. If different datasets name countries differently, this will pose a challenge.

To reduce chances for errors down the line, I trimmed any additional whitespaces from the data using the str_trim function.

```
#Get rid of unnecessary whitespace
cc_data$Country <- str_trim(cc_data$Country) %>% as.factor()
cc_data$Region <- str_trim(cc_data$Region) %>% as.factor()
hdi2014$Country <- str_trim(hdi2014$Country) %>% as.factor()
wh2016$Country <- str_trim(wh2016$Country) %>% as.factor()
wh2016$Region <- str_trim(wh2016$Region) %>% as.factor()
```

The world happiness dataset has 157 countries.

I am going to standardize all of the country names using a convenient function in the standardize text package. The function recognizes common variations of country names and essentially "Autocorrects" them into a standard format. This will help a lot when joining datasets together.

```
#install.packages("StandardizeText")
library(StandardizeText)
```

```r
#Standardize column using default country names
hdi2014$Country <- standardize.countrynames(hdi2014$Country,suggest="auto", v
erbose = T)
```

```
##
## The following names were not recoginized and left unchanged:
## [1] "Arab States"                    "CÃ´te d'Ivoire"
## [3] "Cabo Verde"                     "East Asia and the Pacific"
## [5] "Europe and Central Asia"        "Latin America and the Caribbean"
## [7] "South Asia"                     "Sub-Saharan Africa"
##
## The following names were changed:
##                                    Original
## 1           Bolivia (Plurinational State of)
## 2                                      Congo
## 3         Congo (Democratic Republic of the)
## 4                 Iran (Islamic Republic of)
## 5                   Korea (Republic of)
## 6           Lao People's Democratic Republic
## 7           Micronesia (Federated States of)
## 8                     Moldova (Republic of)
## 9                       Palestine, State of
## 10                    Saint Kitts and Nevis
## 11                               Saint Lucia
## 12          Saint Vincent and the Grenadines
## 13                                  Slovakia
## 14              Tanzania (United Republic of)
## 15 The former Yugoslav Republic of Macedonia
## 16        Venezuela (Bolivarian Republic of)
## 17                                  Viet Nam
##                          Modified
## 1                         Bolivia
## 2                 Congo Republic
## 3       Congo Democratic Republic
## 4                            Iran
## 5                 Korea Republic
## 6                            Laos
## 7                      Micronesia
## 8                         Moldova
## 9           Palestinian Territory
## 10            St. Kitts and Nevis
## 11                       St. Lucia
## 12 St. Vincent and the Grenadines
## 13                Slovak Republic
## 14                       Tanzania
## 15                      Macedonia
## 16                      Venezuela
## 17                        Vietnam
##
```

```
## The following suggested changes were applied:
##                 Original Suggested
## 1 Hong Kong, China (SAR) Hong Kong
```

I can see here that some useful changes were made! Also it seems there was an encoding error for "CÃ´te d'Ivoire", so I will manually change this to "Cote d'Ivoire" in the hdi2014 dataset.

```
#Replace mistake manually
hdi2014$Country[which(hdi2014$Country=="CÃ´te d'Ivoire")] <- "Cote d'Ivoire"

#Make sure it worked
hdi2014$Country[which(hdi2014$Country=="CÃ´te d'Ivoire")]

## character(0)

hdi2014$Country[which(hdi2014$Country=="Cote d'Ivoire")]

## [1] "Cote d'Ivoire"

#Standardize column using default country names
wh2016$Country <- standardize.countrynames(wh2016$Country,suggest="auto", ver
bose = T)

##
## The following names were not recoginized and left unchanged:
## [1] "North Cyprus"      "Somaliland Region"
##
## The following names were changed:
##                  Original                  Modified
## 1     Congo (Brazzaville)            Congo Republic
## 2        Congo (Kinshasa) Congo Democratic Republic
## 3             Ivory Coast             Cote d'Ivoire
## 4 Palestinian Territories     Palestinian Territory
## 5                  Russia        Russian Federation
## 6               Slovakia           Slovak Republic
## 7             South Korea            Korea Republic
## 8                   Syria      Syrian Arab Republic
```

This data had 8 names that required standardization! Also, some names weren't recognized, but we will see if we need those.

```
#Standardize column using default country names
cc_data$Country <- standardize.countrynames(cc_data$Country,suggest="auto")

##
## Note: 3 names were not recoginized and left unchanged.
##
## The following names were changed:
##                            Original                       Modified
## 1                 Antigua & Barbuda            Antigua and Barbuda
## 2                      Bahamas, The                        Bahamas
```

```
## 3               Bosnia & Herzegovina         Bosnia and Herzegovina
## 4                 British Virgin Is.              Virgin Islands BR
## 5                             Brunei              Brunei Darussalam
## 6               Central African Rep.       Central African Republic
## 7                   Congo, Dem. Rep.      Congo Democratic Republic
## 8                         East Timor                    Timor-Leste
## 9                       Gambia, The                         Gambia
## 10                     Korea, North      Korea Democratic Republic
## 11                     Korea, South                Korea Republic
## 12                            Macau                         Macao
## 13                           Russia             Russian Federation
## 14                     Saint Helena                      St. Helena
## 15              Saint Kitts & Nevis            St. Kitts and Nevis
## 16                      Saint Lucia                       St. Lucia
## 17 Saint Vincent and the Grenadines St. Vincent and the Grenadines
## 18              Sao Tome & Principe          Sao Tome and Principe
## 19                         Slovakia                Slovak Republic
## 20              St Pierre & Miquelon        St. Pierre and Miquelon
## 21                             Syria           Syrian Arab Republic
## 22                Trinidad & Tobago            Trinidad and Tobago
## 23                Turks & Caicos Is       Turks and Caicos Islands
##
## The following suggested changes were applied:
##                Original      Suggested
## 1 Congo, Repub. of the Congo Republic
## 2 Micronesia, Fed. St.     Micronesia
```

Perfect! Now all country names should be standardized, so we can do a join. Let's see which countries from the world happiness data do not have a match in the human development data using an anti-join.

```
#make sure the country name standardization worked
non_matches <- anti_join(wh2016[1], hdi2014[2], by = "Country")
non_matches2 <- anti_join(wh2016[1], cc_data[1] , by = "Country")

#print out distinct non-matches
aa<-distinct(rbind(non_matches,non_matches2))
```

There are 10 countries that have a happiness score but do not have any data in either the HDI data or the CC data are Puerto Rico, Taiwan, North Cyprus, Somalia, Kosovo, Somaliland Region, Montenegro, Palestinian Territory, Myanmar, and South Sudan. Because these countries don't have any data from the other datasets for the dependent/response/target variable, they would essentially be complete empty rows. From the eventual merged data, I will be removing these countries for the analysis.

In order to merge the datasets, I am going to do an inner join, which will essentially pull all the data that both datasets have based on matching the "Country" column. For example, a sample row will contain the data columns located in the all three datasets for a single country name, such as France.

```r
#merge datasets to make final dataset
world_df <- inner_join(wh2016, hdi2014, by = "Country")

world_df <- inner_join(world_df, cc_data, by = "Country")

nrow(world_df)
```

```
## [1] 147
```

147 countries remaining, perfect!

Here we see some features that will require attention. 1) there are two region columns from 2 different datasets. We will only use one of these. I am going to use the regions from the World Happiness data and remove the other column. 2) HDI rank acts as a row number in the hdi data and doesn't add information beyond the HDI score, so we will remove the rank column. 3) Since the rank column won't be used I will also remove the "GNI.per.Capita.Rank.Minus.HDI.Rank" columns since it relies on rank and I could not find an explanation of what this column means. 4) The punctuation located within the feature names got coerced into "." periods, so I will be renaming some columns to make them more readable. 5) The gross national income columns is currently a character instead of a numeric.

```r
anyNA(world_df)
```

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

There are NA values, so let's try to handle these.

The aren't very many NA values since the datasets were pretty full, but there are 3 in Literacy.percent, 1 in Phones.per.1000, 16 in Climate, 1 in Birthrate, and 1 in Deathrate. Since there aren't a lot, I am going to impute these values. This can be done in a variety of ways. Since they are all numerical features, I could impute them with the mean or median. Also, I could impute using similar countries based on a model such as kNN. Since the dimensionality is quite high for kNN which works best on low dimensions 5-15 and there are 25, I won't use this method. Additionaly, due to the small amount of NAs, a central tendency imputation will likely be quite accurate and/or not skew the model much.

I will replace all with median, since it is less sensitive to outliers.

```r
#make sure it worked
anyNA(world_df)
```

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

Awesome! All NA values have been imputed with the median.

Now are there any outliers?!?

There are multiple ways to detect outliers, such as using the IQR and the z-score, then we will make a determination of what to do with these values if any. Outliers can also be detected by plotting a linear regression model and with principal component analysis.

I am going to detect outliers using the Z/score.

Outlier = +/- 3 standard deviations from the mean. (A.k.a +/- z-score of 3). 3 is a rule of thumb, it does not work in every instance. We are going to use 3 in this instance since the variance is relatively standard and there are no observable clusters in the data. We could explore other values for the cutoff based on variance in the features because sometimes if there is pretty low variance and values tend to stick around a certain point, then a lower threshold may be better than the 3 heuristic. However, we are making the decision to explore IQR as another outlier detection method rather than other z-score cutoffs.

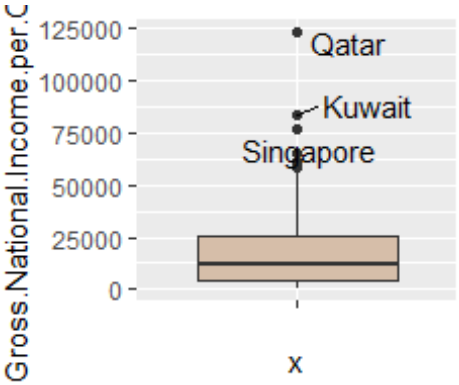
```
#create a function that calculates outliers based on zscore and formula above
#function takes in a continuous variable and spits out the z scores
outlier.z <- function(cvar) {
a <- sd(cvar)
b <- mean(cvar)
c <- ((b-cvar)/(a))
c
}
```

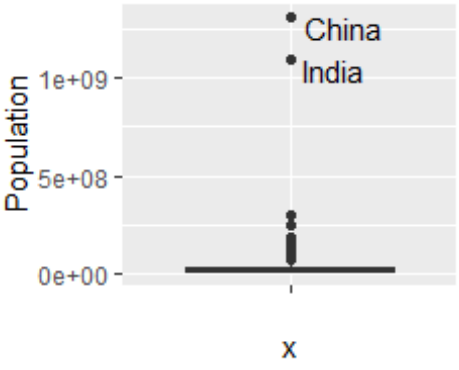
First let's output all observation numbers of the rows that have a abs(z-score) >3

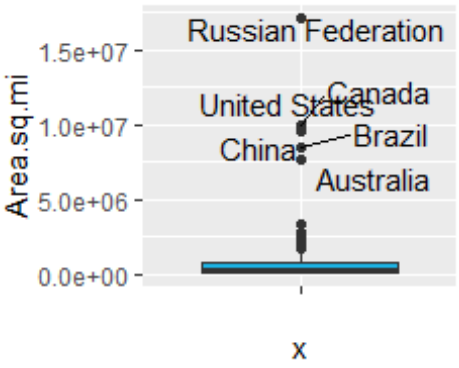
Based on the z-score method we can see that many countries have values that are larger than 3 z-scores from the mean in certain features. The countries fall on both high and low ends of the spectrum. These listed countries would be considered outliers by this method.

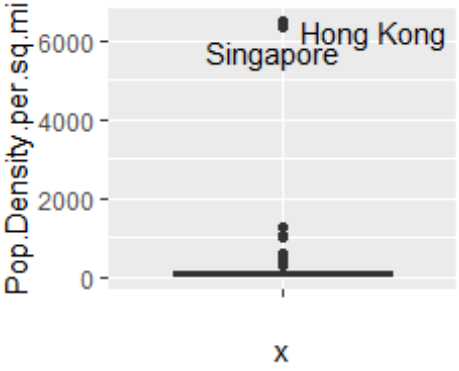
For example, you can see that China and India are outliers in terms of population, and the 6 largest countries are outliers in terms of land area. Singapore and Hong Kong are outliers in terms of Population Density. Luxembourg is an outlier in terms of GDP per capita. All of these findings match with what we would expect logically, which is a good sign.

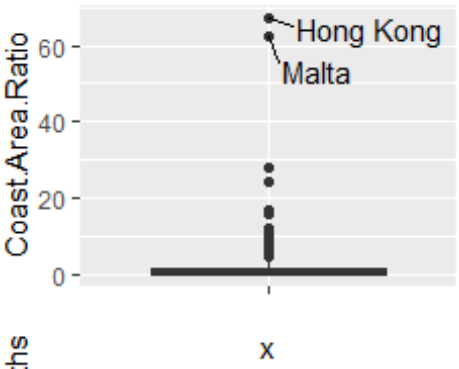
Now let's explore some of these outliers visually.

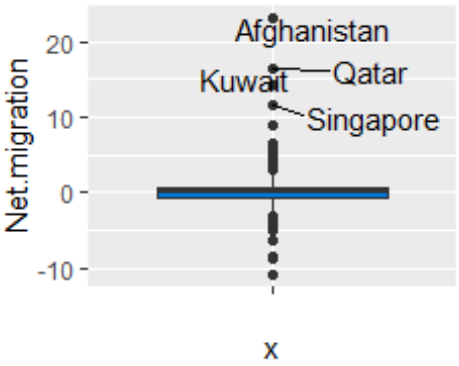

Gross.National.Income.per.C
125000
100000
75000
50000
25000
0
Qatar
Kuwait
Singapore
x

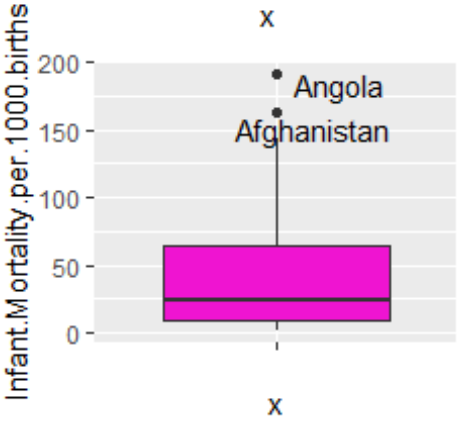

Population
1e+09
5e+08
0e+00
China
India
x

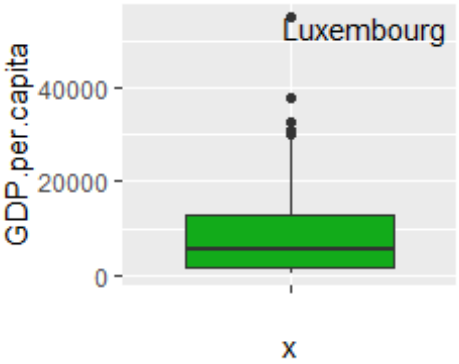

Area.sq.mi
1.5e+07
1.0e+07
5.0e+06
0.0e+00
Russian Federation
Canada
United States
China
Brazil
Australia
x

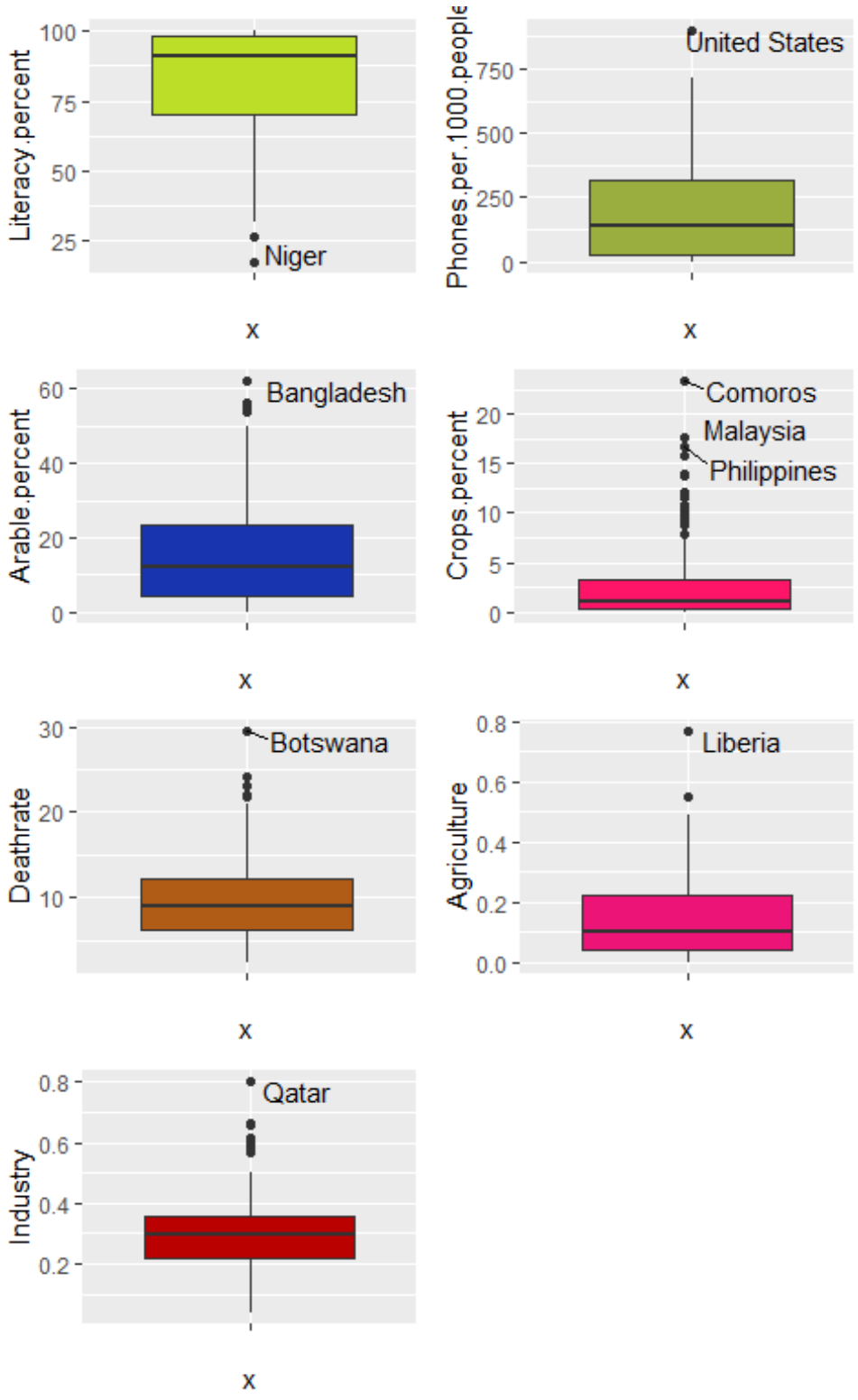

Pop.Density.per.sq.mi
6000
4000
2000
0
Hong Kong
Singapore
x


Coast.Area.Ratio
60
40
20
0
Hong Kong
Malta
x


Net.migration
20
10
0
-10
Afghanistan
Kuwait
Qatar
Singapore
x


Infant.Mortality.per.1000.births
200
150
100
50
0
Angola
Afghanistan
x


GDP.per.capita
40000
20000
0
Luxembourg
x



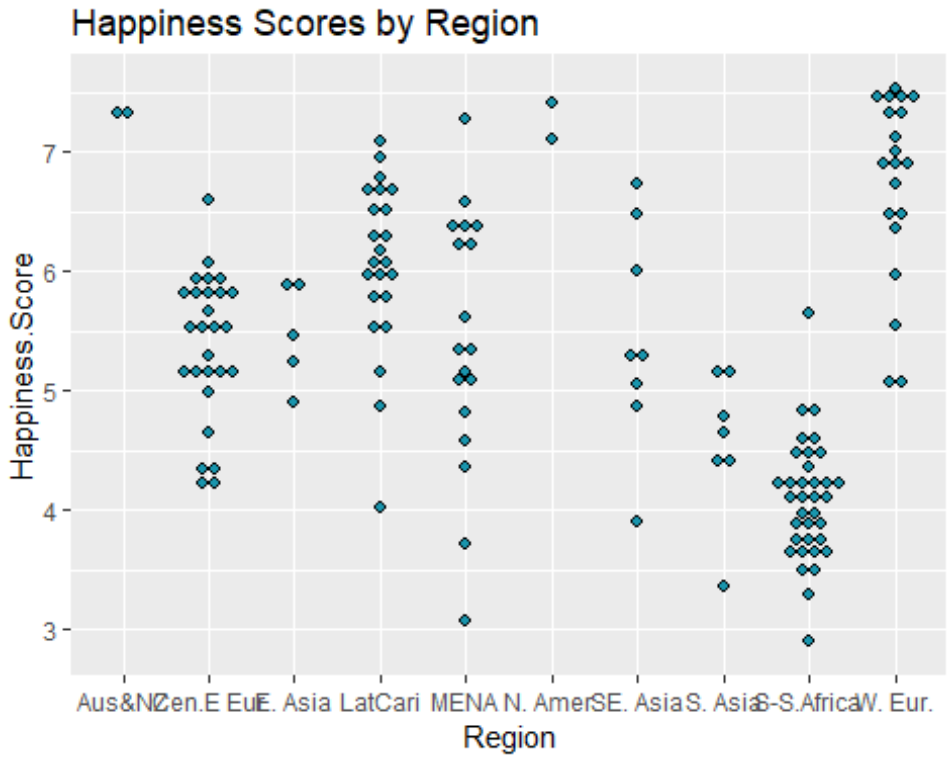
These boxplots (representing all of the features that contain outliers based on z-score) clearly show a good story into the data. All of the findings make sense with what we would expect and accessible layout the distributions of the features as well as label the outlier countries. The removal of outliers is something to take seriously. It could have big negative

ramifications if you remove outliers without justification. Since all of the outliers here represent actual conditions out in the world and are not based on data input error, I am going to make the careful decision to leave them all in the dataset. This is an assumption that will be marked. I will pay particular attention to output result with these in mind. However, I fully expect that the presence of these data points won't affect our analysis to a grand extent.

Also, since we don't have very many observations, once we scale our data, outliers now may not remain outliers. Lastly, once more data is collected (such as with the other countries in the world), it's possible outliers won't be outliers anymore.

Let's do some more exploratory visualizations to get a sense of relationships between variables.

```
##  [1] "Australia and New Zealand"       "Central and Eastern Europe"
##  [3] "Eastern Asia"                    "Latin America and Caribbean"
##  [5] "Middle East and Northern Africa" "North America"
##  [7] "Southeastern Asia"               "Southern Asia"
##  [9] "Sub-Saharan Africa"              "Western Europe"
```



This dot plot nicely shows some distribution of happiness scores by region. You can see some regions such as the Middle East and North Africa have a large varaince. For example, there is a country with a happiness score of just above 3, but there is also a country with a happiness score of over 7. Also, you can see that the regions North America and Australia & NZ only have two members each.

Because the variance within regions is high and some regions only have a few members, I anticipate my eventual classification of region may be difficult. I can try to combat this with a few methods such as stratified sampling, sampling with replacement, and k-fold cross validation. We will see how it turns out!

My first model I am going to explore is Multiple Linear Regression in hopes of predicting the Happiness score of a nation.

Multiple Linear Regression

Let's do some correlation/collinearity analysis, since multicollinearity can doom a regression model.

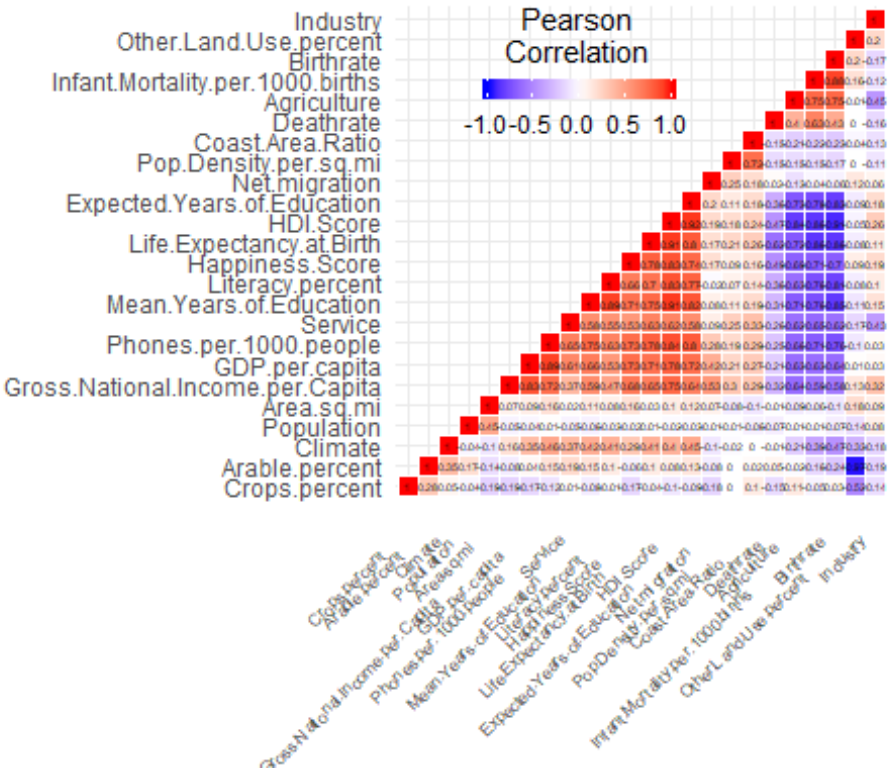
Let's explore correlations to the response variable Happiness Score since the full correlation table would be hard to digest.

High values indicate high correlations, and when there are multiple features correlated with one another (which is not visualized here…yet), that indicates collinearity, which is not ideal for a regression analysis. Essentially, the same information is conveyed by multiple variables. Right off the bat I can see some high correlations such as between HDI Score and Happiness Score.

This is a little difficult to visualize, though. Let's see if we can visualize it better.

Using starter code from STHDA, we are going to create a correlation matrix that is shaded by intensity of correlation.

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

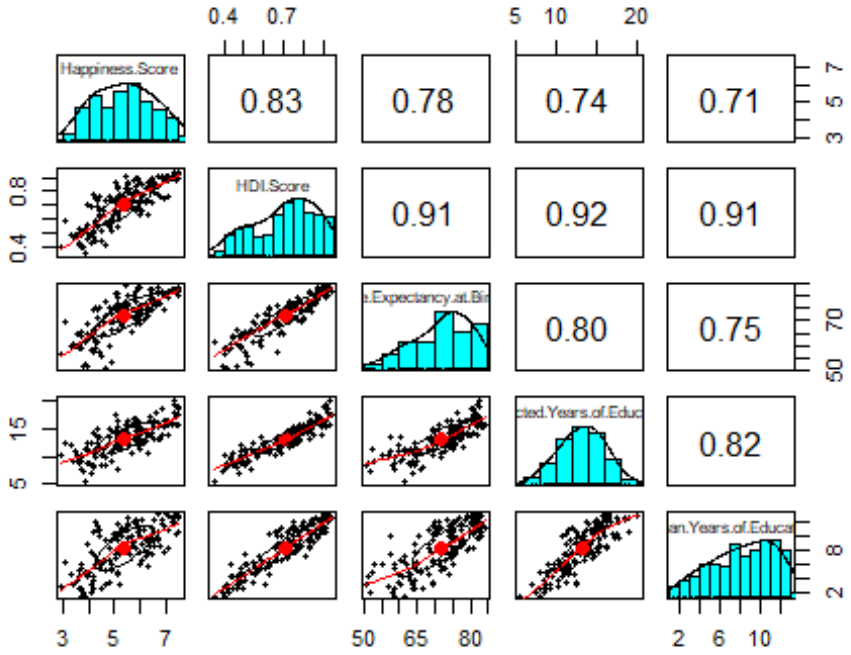


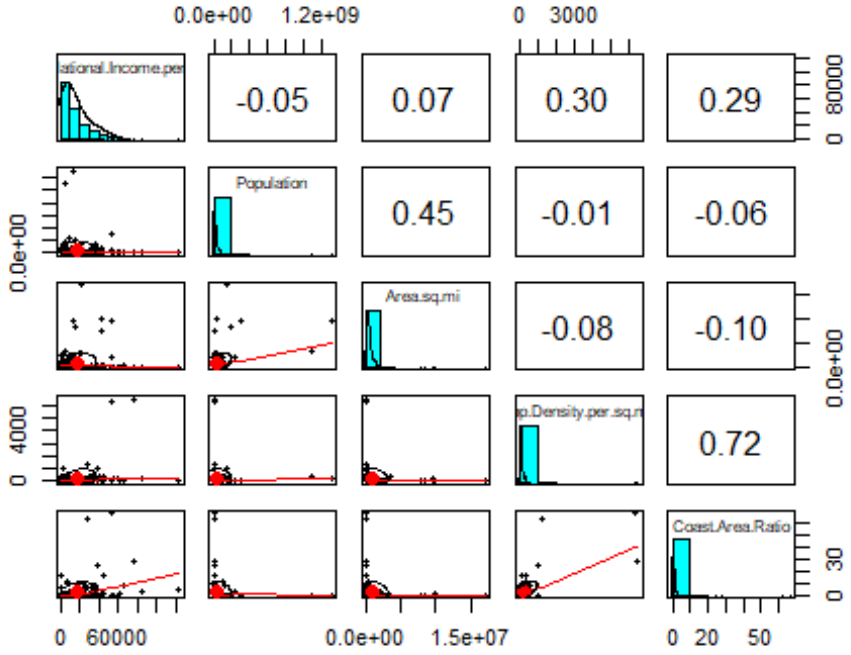
Ahh much better. Now we can visualize our correlations. As we can see, some overall correlations between the variables are pretty high which means there is likely collinearity. The strongest correlations (which we are going to count as ones with an absolute value >=.75) are between HDI.Score and other variables. Since HDI Score is essentially another dependent variable that was calculated based on a variety of factors, I am going to drop it and keep the other features and see how this improves collinearity. Collinearity exists when too many features explain eachother, and it seems that the correlations between the variables are high enough to explain eachother.

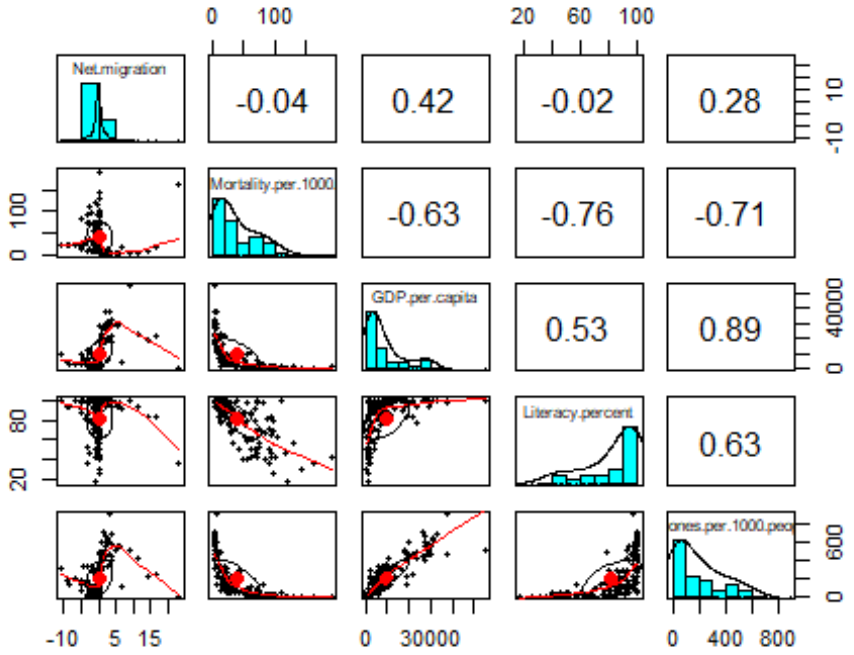
Since feature removal is a big deal and can have large adverse impacts to a model if done incorrectly, I won't make any additional removals before creating a model and finding the Variance Inflation Factor (VIF) for each predictor. This VIF value will help me understand when it's safe to remove features.

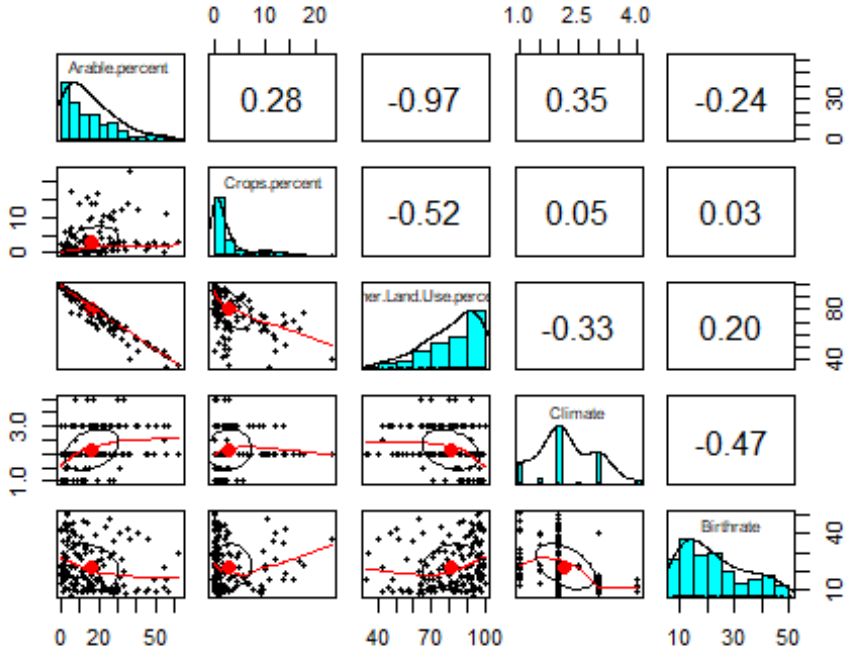
I will be building this regression model shortly.

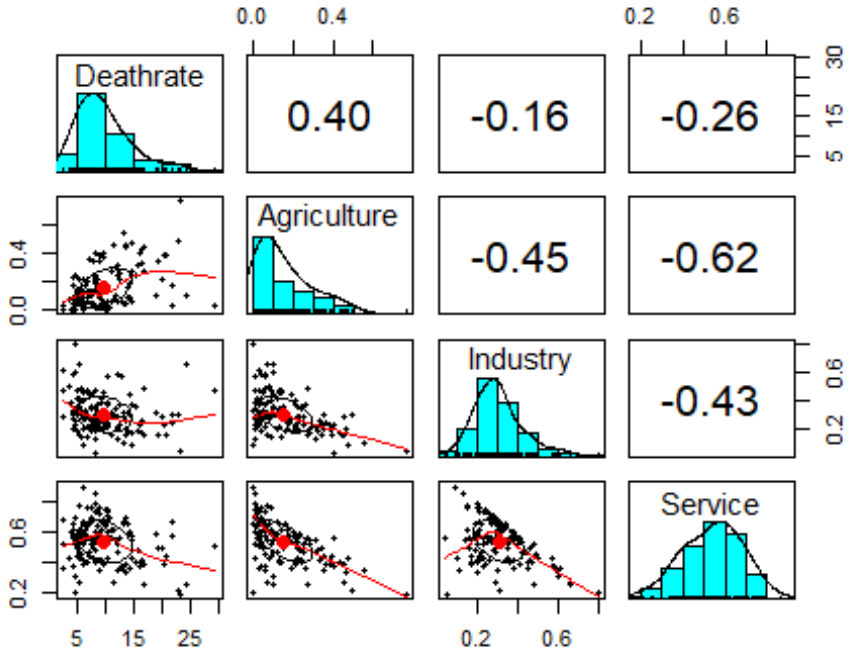
First, I want to do some more exploratory data visualization with pairs.panels() and inspect distribution of features to see if I need to apply transforms in order to make a normally distributed dataset. Linear regression is parametric and assuming features are normally distributed.


Happiness.Score
HDI.Score
0.83
0.78
0.74
0.71
0.91
0.92
0.91
0.80
0.75
0.82


Population
Area.sq.mi
Coast.Area.Ratio
-0.05
0.07
0.30
0.29
0.45
-0.01
-0.06
-0.08
-0.10
0.72


Net.migration
Mortality.per.1000
GDP.per.capita
Literacy.percent
-0.04
0.42
-0.02
0.28
-0.63
-0.76
-0.71
0.53
0.89
0.63


Arable.percent
Crops.percent
Climate
Birthrate
0.28
-0.97
0.35
-0.24
-0.52
0.05
0.03
-0.33
0.20
-0.47



When the oval (correlation elipse) is stretched, it means a strong correlation. We can see that Expected years of education and mean years of education have strong correlations and likely explain eachother, for example. There could be collinearity between these.

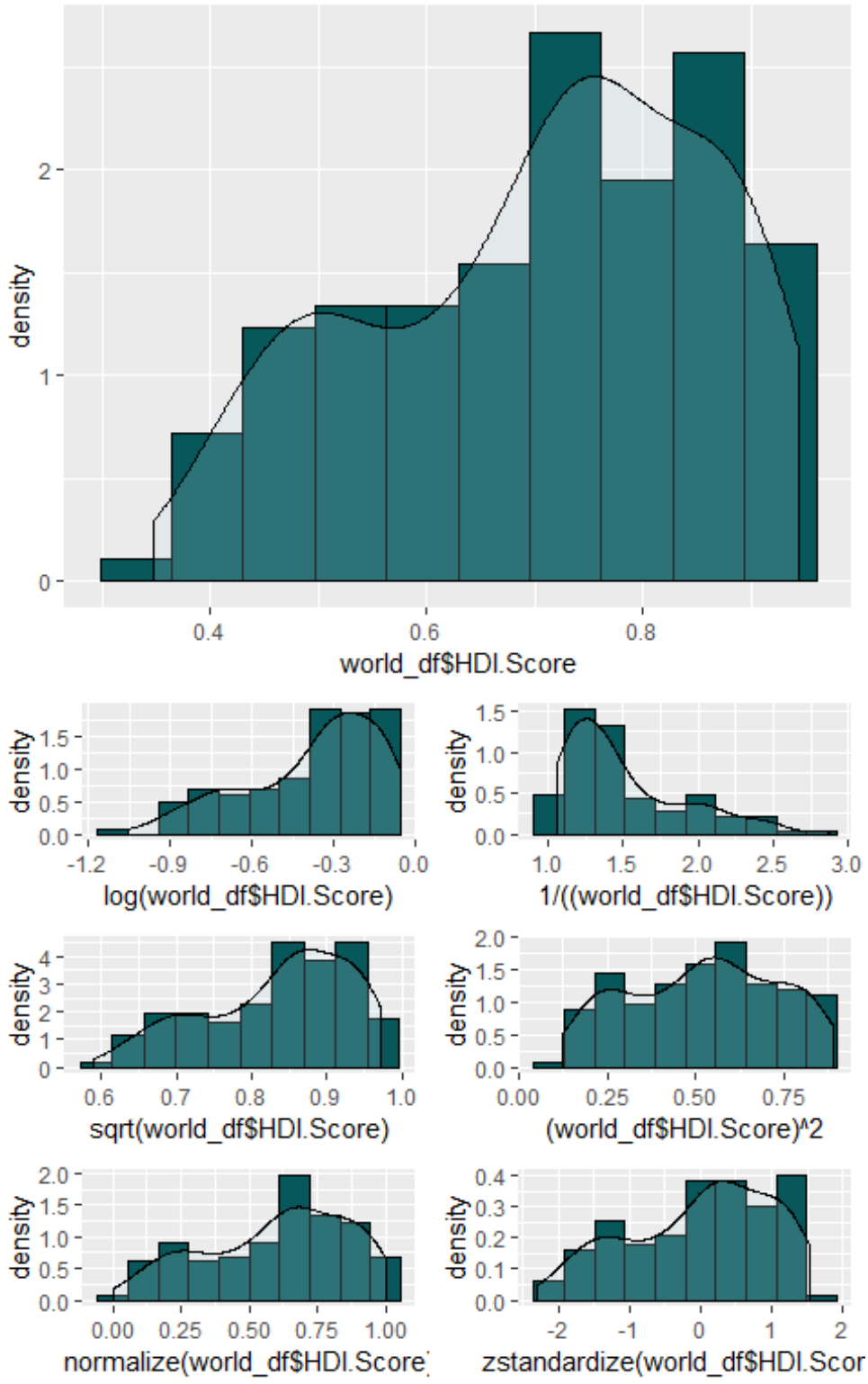
Regression assumes normality, so let's see if any transforms help the data look more normally distributed. Age and absences in particular look off.

Let's explore a few of these closer. In particular we are concerned about transforming the variables such as HDI.Score, Life Expectancy, Mean years of education, Gross National Income, Population, Area sq mi, population density, coast area ratio, net migration, infant mortality, GDP per capita, literacy, phones per 1000, arable percent, crops percent, other land use percent, climate, birthrate, deathrate, agriculture to make them resemble normal distributions more closely. I am deeming the other features to be fairly normally distributed.

Disclaimer: it is possible that some of these features may not even make it into the final model due to feature selection and backfitting, but I am going to normalize them for good measure.

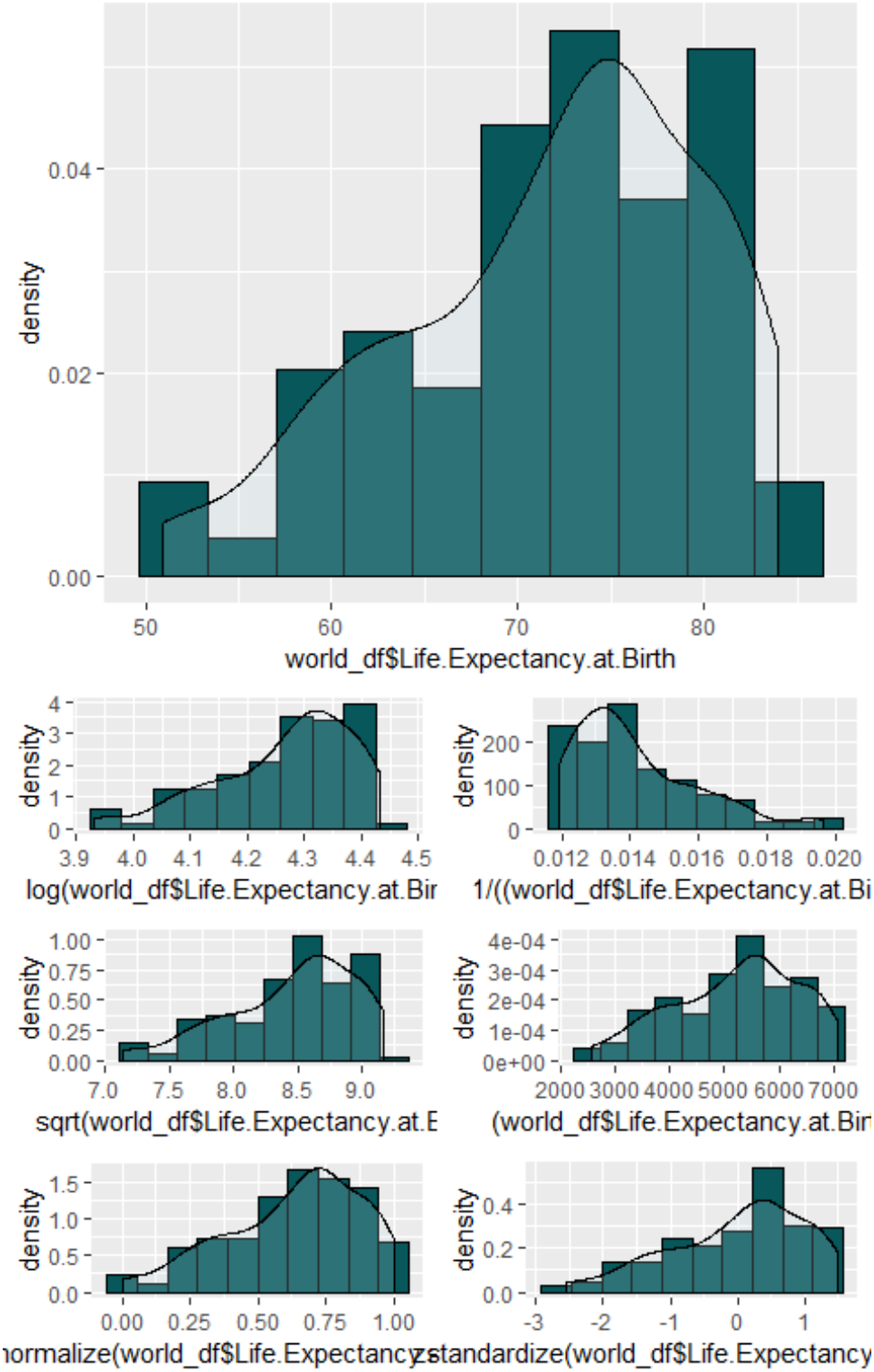
To make this faster I will make function to min/max and z-score transform features

First let's start with HDI.Score.



It looks like the Square ^2 transform makes it most resemble a normal distribution, so let's replace it with its square.

Now Life Expectancy.



It looks like the min/max transform makes it most resemble a normal distribution since it slightly reduces the left skew, so let's replace it with it.

The above iterations of testing and transforming features shows the process visually. Now, in orer to speed things up for the remaining 18 features I want to transform, I am going to use the bestNormalize package. This package checks all of the transforms (plus more complicated ones) similar to how I have been doing, then transforms the data based on the best transform. The best transform is determined by the Estimated Normality Statistics (Pearson P / df). The lower the value ==> the more normal it is. The function is doing repeated CV in order to find the best transform.

The orderNorm method guarantees normality, so I will set this to false since it is not as natural of a transform.

Use bestNormalize for remaining features.

The best transform is chosen. This takes quite a while to run because it is doing 5 fold CV with 3 repeats for every feature to ensure the best transform is chosen.

Now that we have found the best transform for all of the remaining features, I am going to show an example output and then replace the values in our dataframe with the transformed values. The transformed values from the bestNormalize function can be accessed in the $x.t call.

An example output for the Infant Mortality feature is:

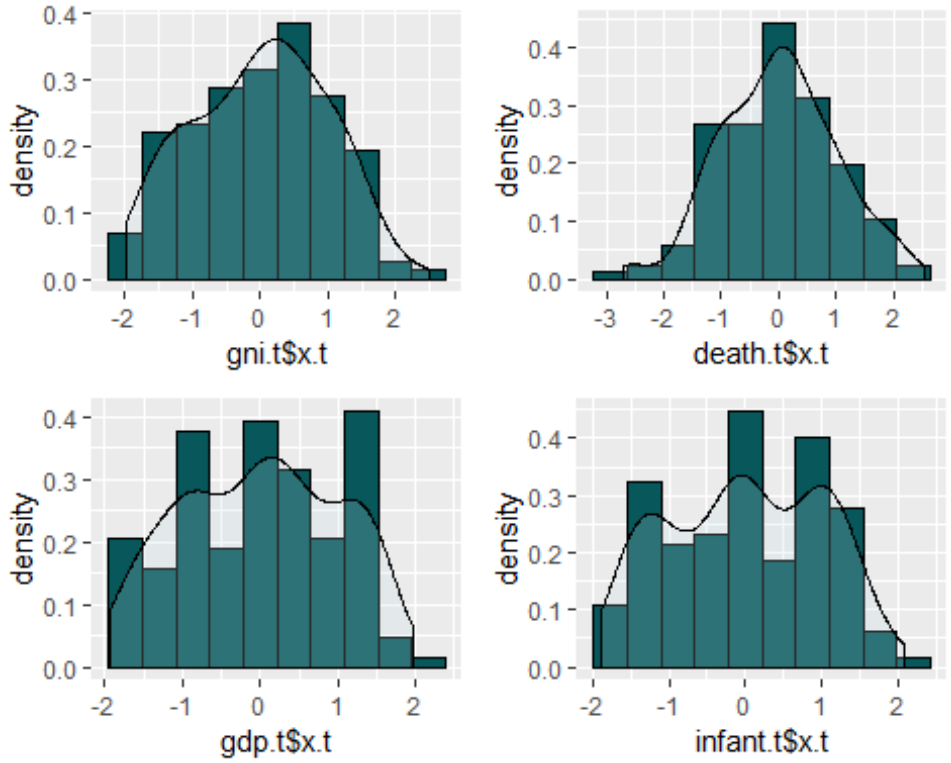
```
#Show chosen transform and statistics
infant.t
```

```
## Best Normalizing transformation with 147 Observations
##  Estimated Normality Statistics (Pearson P / df, lower => more normal):
##  - No transform: 4.068
##  - Box-Cox: 1.5484
##  - Log_b(x+a): 1.7473
##  - sqrt(x+a): 2.2009
##  - exp(x): 18.7236
##  - arcsinh(x): 1.7036
##  - Yeo-Johnson: 1.5773
## Estimation method: Out-of-sample via CV with 5 folds and 3 repeats
##
## Based off these, bestNormalize chose:
## Standardized Box Cox Transformation with 147 nonmissing obs.:
##  Estimated statistics:
##  - lambda = 0.1229329
##  - mean (before standardization) = 3.953822
##  - sd (before standardization) = 1.638873
```

```
#Show transformed values
head(infant.t$x.t)
```

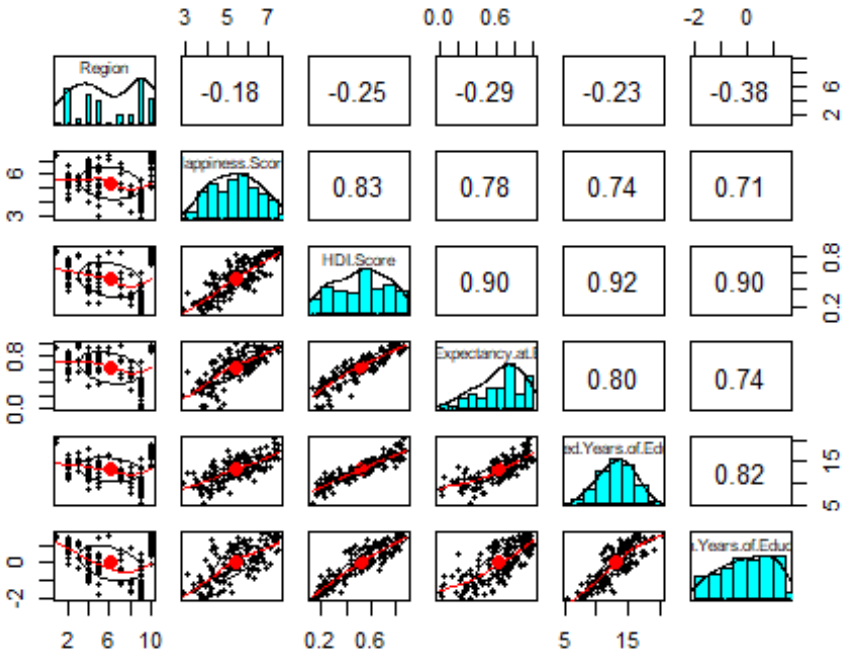
```
## [1] -1.394716 -1.422587 -1.625708 -1.546428 -1.572004 -1.364624
```

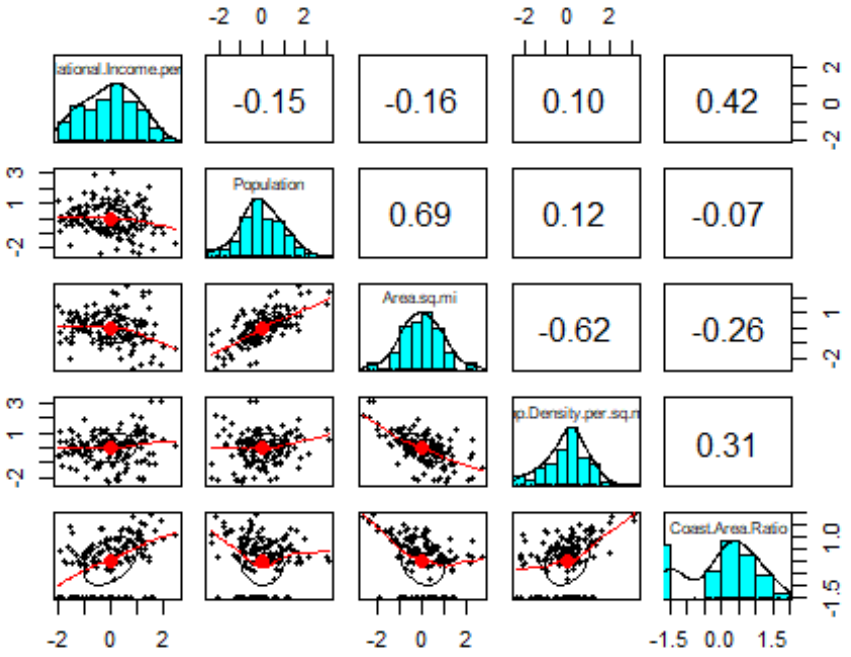
Let's see if this actually works visually.

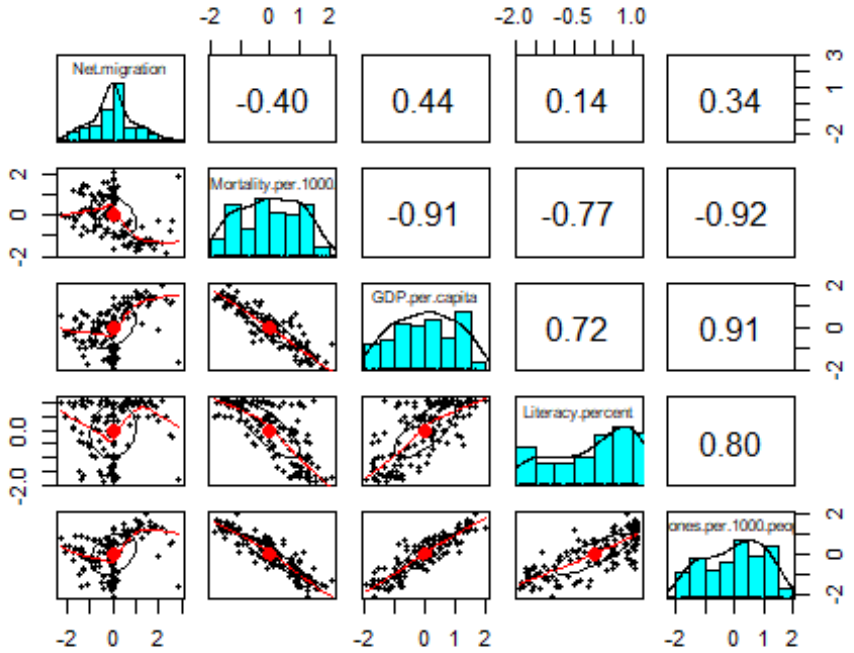


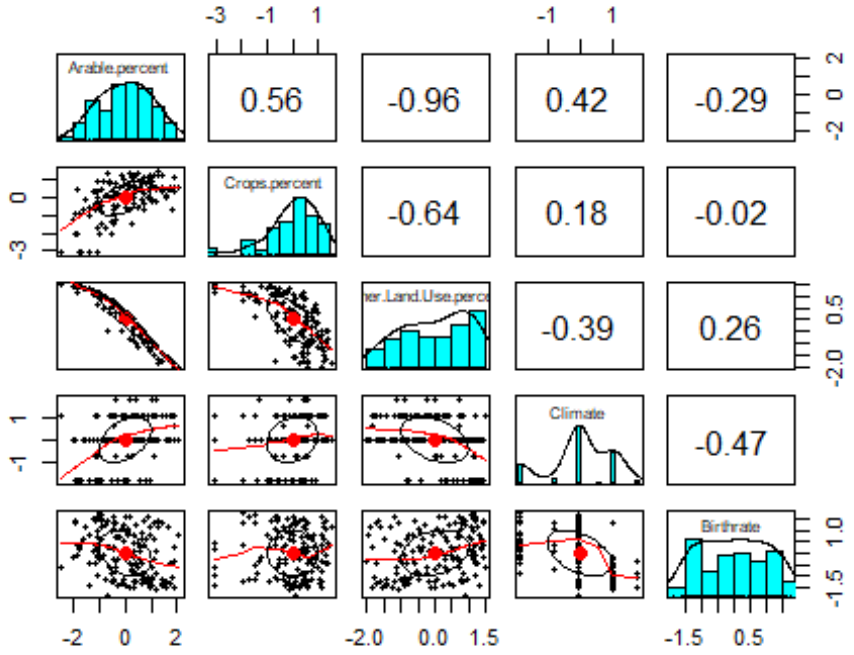
Looks a lot better than before! Now let's apply all of these to the normally distributed data frame. If at the end of our regression analysis we have to reverse any transform, we can easily access which transform was applied using the $chosen_transform call.

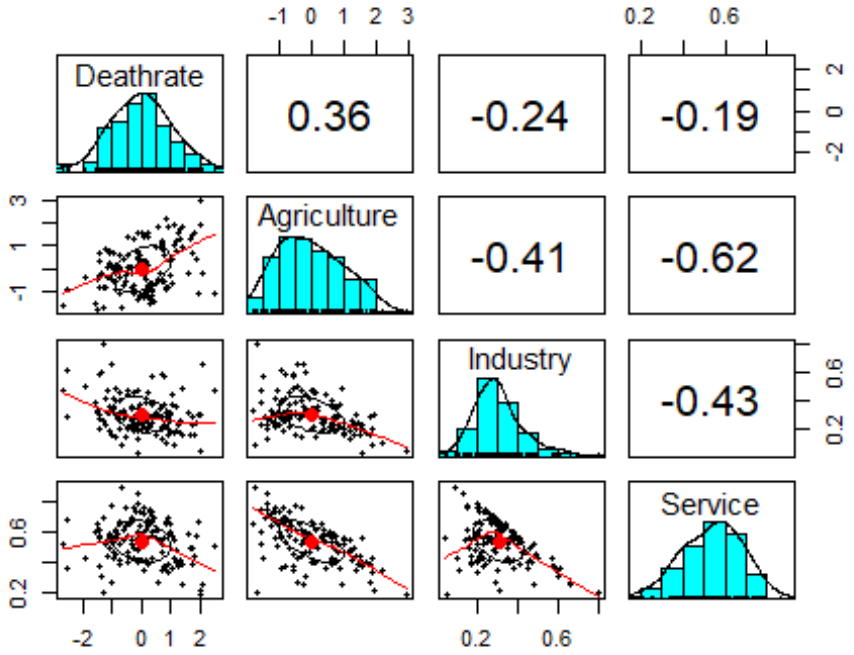
Look at the pairs.panels again.






NetMigration
Mortality.per.1000
GDP.per.capita
Literacy.percent
-0.40
0.44
0.14
0.34
-0.91
-0.77
-0.92
0.72
0.91
0.80


Arable.percent
Crops.percent
Climate
Birthrate
0.56
-0.96
0.42
-0.29
-0.64
0.18
-0.02
-0.39
0.26
-0.47



It looks much better!

Now dummy code the "Region" category for the normalized dataset since in this instance it is a predictor and not a response variable.

```
region_vars <- model.matrix( ~ Region - 1, data=world_norm_dist )
head(region_vars[,-10])
```

```
##   RegionAustralia and New Zealand RegionCentral and Eastern Europe
## 1                               0                                 0
## 2                               0                                 0
## 3                               0                                 0
## 4                               0                                 0
## 5                               0                                 0
## 6                               0                                 0
##   RegionEastern Asia RegionLatin America and Caribbean
## 1                  0                                 0
## 2                  0                                 0
## 3                  0                                 0
## 4                  0                                 0
## 5                  0                                 0
## 6                  0                                 0
##   RegionMiddle East and Northern Africa RegionNorth America
## 1                                     0                    0
## 2                                     0                    0
## 3                                     0                    0
## 4                                     0                    0
```

```
## 5                                     0                   0
## 6                                     0                   1
##   RegionSoutheastern Asia RegionSouthern Asia RegionSub-Saharan Africa
## 1                       0                   0                        0
## 2                       0                   0                        0
## 3                       0                   0                        0
## 4                       0                   0                        0
## 5                       0                   0                        0
## 6                       0                   0                        0
```

```
#add dummy columns -1 to the data. There is always one less columns than ther
e are levels
world_norm_dist <- cbind(world_norm_dist, region_vars[,-10])

#do a quick spot check
head(world_df$Region)
```

```
## [1] Western Europe Western Europe Western Europe Western Europe
## [5] Western Europe North America
## 10 Levels: Australia and New Zealand ... Western Europe
```

The binary dummy variables match with the actual values! Sweet. If all values are 0, this means that the region is Western Europe. This will be represented by the intercept in the regression model.

Now I am going to remove the original region column.

Remove spaces and special characters from new variable names.

Create an easy list of predictors to pull from for the regression model.

```
#prepare the list of predictor names for multiple regression
var_names <- names(world_norm_dist[3:34])
formula <- as.formula(paste('Happiness.Score ~ ' ,paste(var_names,collapse='+
')))

#make sure it worked
formula
```

```
## Happiness.Score ~ HDI.Score + Life.Expectancy.at.Birth + Expected.Years.of
.Education +
##     Mean.Years.of.Education + Gross.National.Income.per.Capita +
##     Population + Area.sq.mi + Pop.Density.per.sq.mi + Coast.Area.Ratio +
##     Net.migration + Infant.Mortality.per.1000.births + GDP.per.capita +
##     Literacy.percent + Phones.per.1000.people + Arable.percent +
##     Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
##     Deathrate + Agriculture + Industry + Service + Region.AusNZ +
##     Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
##     Region.N.Amer + Region.SE.Asia + Region.S.Asia + Region.SS.Africa
```

I am going to build a multiple regression model with the aim of using the VIF to help with feature selection. If there are variables that explain eachother too much, I will know to remove them. Any VIF above 20 or so is considered high

```
#make model for all features
m1 <- lm(Happiness.Score ~ HDI.Score + Life.Expectancy.at.Birth + Expected.Ye
ars.of.Education +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Area.sq.mi + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Infant.Mortality.per.1000.births + GDP.per.capita +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service + Region.AusNZ +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
    Region.N.Amer + Region.SE.Asia + Region.S.Asia + Region.SS.Africa, data =
world_norm_dist)
```

Now lets looks at the Varaince Inflation Factor numbers to get a sense of multicollinearity.

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:psych':
##
##     logit

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some

##                          HDI.Score         Life.Expectancy.at.Birth
##                             467.72                            32.33
##      Expected.Years.of.Education          Mean.Years.of.Education
##                             23.64                            43.46
## Gross.National.Income.per.Capita                       Population
##                             96.31                            72.51
##                        Area.sq.mi            Pop.Density.per.sq.mi
##                            117.14                            61.54
##                  Coast.Area.Ratio                   Net.migration
##                              2.08                             2.51
## Infant.Mortality.per.1000.births                  GDP.per.capita
##                             20.08                            17.41
##                  Literacy.percent           Phones.per.1000.people
##                             11.75                            17.02
##                    Arable.percent                    Crops.percent
```

```
##                            23.39                            3.50
##           Other.Land.Use.percent                         Climate
##                            21.80                            2.56
##                        Birthrate                       Deathrate
##                            15.70                            6.35
##                      Agriculture                        Industry
##                            60.83                           28.34
##                          Service                     Region.AusNZ
##                            36.70                            1.52
##                 Region.Cen.E.Eur                    Region.E.Asia
##                             5.20                            1.75
##                    Region.LatCari                     Region.MENA
##                             4.72                            5.33
##                     Region.N.Amer                    Region.SE.Asia
##                             1.68                            2.80
##                     Region.S.Asia                  Region.SS.Africa
##                             3.21                            9.82
```

Here we can see that several features have very high VIFs. This signals that features explain eachother and there is multicollinearity. However, it is important to note that multicollinearity can sometimes be ignored, if the collinearity does not affect statistical significance. For example, "If your model has x, z, and xz, both x and z are likely to be highly correlated with their product. This is not something to be concerned about, however, because the p-value for xz is not affected by the multicollinearity."[___]. It is not always reason for alarm when features are derived from eachother. It makes sense that they would explain eachother, yet they don't affect p-values.

Yet, the HDI.Score VIF is extremely high. We saw high correlations earlier in the correlation matrix too. Because HDI.Score is a direct calculation from every other feature in the HDI datset, it is explain by all the other features. I am going to remove HDI.Score.

```
##          Life.Expectancy.at.Birth      Expected.Years.of.Education
##                             11.91                             7.45
##           Mean.Years.of.Education Gross.National.Income.per.Capita
##                             14.30                            20.98
##                        Population                        Area.sq.mi
##                             72.31                           117.09
##             Pop.Density.per.sq.mi                  Coast.Area.Ratio
##                             61.47                             2.08
##                      Net.migration Infant.Mortality.per.1000.births
##                              2.51                            17.37
##                     GDP.per.capita                  Literacy.percent
##                             17.34                            11.56
##             Phones.per.1000.people                    Arable.percent
##                             17.02                            23.36
##                      Crops.percent           Other.Land.Use.percent
##                              3.49                            21.75
##                           Climate                        Birthrate
##                              2.47                            15.44
##                         Deathrate                       Agriculture
```

```
##                          5.71                          59.94
##                      Industry                        Service
##                         28.27                          36.40
##                    Region.AusNZ                 Region.Cen.E.Eur
##                          1.52                           4.90
##                    Region.E.Asia                   Region.LatCari
##                          1.75                           4.59
##                     Region.MENA                    Region.N.Amer
##                          5.24                           1.67
##                   Region.SE.Asia                   Region.S.Asia
##                          2.68                           3.02
##                Region.SS.Africa
##                          9.45
```

Now I am going to remove Area, as it's information is explained by other features such as the land usage % stats.

```
##         Life.Expectancy.at.Birth      Expected.Years.of.Education
##                            11.73                             7.44
##          Mean.Years.of.Education Gross.National.Income.per.Capita
##                            14.26                            20.96
##                       Population           Pop.Density.per.sq.mi
##                            1.81                             5.39
##                  Coast.Area.Ratio                    Net.migration
##                            2.07                             2.50
## Infant.Mortality.per.1000.births                   GDP.per.capita
##                            16.93                            17.29
##                  Literacy.percent          Phones.per.1000.people
##                            11.56                            16.83
##                    Arable.percent                   Crops.percent
##                            23.36                             3.40
##           Other.Land.Use.percent                          Climate
##                            21.74                             2.38
##                         Birthrate                       Deathrate
##                            14.49                             5.46
##                       Agriculture                        Industry
##                            59.55                            28.00
##                           Service                     Region.AusNZ
##                            36.18                             1.48
##                  Region.Cen.E.Eur                    Region.E.Asia
##                            4.90                             1.75
##                    Region.LatCari                     Region.MENA
##                            4.58                             5.22
##                     Region.N.Amer                    Region.SE.Asia
##                            1.49                             2.68
##                     Region.S.Asia                 Region.SS.Africa
##                            3.02                             9.35
```

Agriculture, service, and indsutry are all dependent on one another, so there is no cause for alarm that their VIFs are now the highest. We are going to leave the remaining feature selection to PCA.

Principal component analysis - works best when there is high correlation between variables.. perfect!

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     3.363189 1.9512695 1.58377157 1.39397981 1.33750782
## Proportion of Variance 0.353470 0.1189829 0.07838539 0.06072437 0.05590397
## Cumulative Proportion  0.353470 0.4724529 0.55083831 0.61156267 0.66746665
##                            Comp.6     Comp.7     Comp.8     Comp.9
## Standard deviation     1.27432844 1.08905268 1.06494212 1.02606244
## Proportion of Variance 0.05074728 0.03706362 0.03544068 0.03290013
## Cumulative Proportion  0.71821393 0.75527755 0.79071822 0.82361835
##                           Comp.10    Comp.11    Comp.12    Comp.13
## Standard deviation     1.00227067 0.92221033 0.80675731 0.75109153
## Proportion of Variance 0.03139208 0.02657725 0.02033929 0.01762933
## Cumulative Proportion  0.85501043 0.88158768 0.90192697 0.91955630
##                           Comp.14    Comp.15     Comp.16     Comp.17
## Standard deviation     0.72290785 0.58508640 0.552073197 0.520098268
## Proportion of Variance 0.01633112 0.01069769 0.009524525 0.008453194
## Cumulative Proportion  0.93588742 0.94658511 0.956109632 0.964562826
##                            Comp.18     Comp.19     Comp.20     Comp.21
## Standard deviation     0.463662392 0.434279806 0.419441117 0.329772123
## Proportion of Variance 0.006718213 0.005893717 0.005497839 0.003398427
## Cumulative Proportion  0.971281038 0.977174756 0.982672595 0.986071021
##                            Comp.22     Comp.23     Comp.24     Comp.25
## Standard deviation     0.313791327 0.274180777 0.249352117 0.236167021
## Proportion of Variance 0.003077031 0.002349222 0.001943015 0.001742964
## Cumulative Proportion  0.989148053 0.991497274 0.993440289 0.995183254
##                            Comp.26    Comp.27     Comp.28      Comp.29
## Standard deviation     0.209379820 0.19678861 0.191040325 0.1456040729
## Proportion of Variance 0.001369997 0.00121018 0.001140513 0.0006625171
## Cumulative Proportion  0.996553251 0.99776343 0.998903943 0.9995664605
##                             Comp.30      Comp.31      Comp.32
## Standard deviation     0.0910302638 0.0630503432 4.014237e-02
## Proportion of Variance 0.0002589534 0.0001242296 5.035655e-05
## Cumulative Proportion  0.9998254139 0.9999496435 1.000000e+00
```

1st component explains 35% of variance in data, 2nd component explains 11% (cumulative 46%).

Eigen values = standard deviation of PCs squared. We could use the first 5, but I'm only going to do 3

All 32 components explain the full variation in the data.

Now let's calculate loadings. These tell us the correlations between each feature and the components. Theoretically, the features most highly correlated to the first few components are the best to use. And the features most correlated with the last components are the ones that are explained by other features.

```
##
## Loadings:
##                                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## HDI.Score                       0.292
## Life.Expectancy.at.Birth        0.273
## Expected.Years.of.Education     0.268
## Mean.Years.of.Education         0.264         0.154         0.139
## Gross.National.Income.per.Capita 0.272  0.132                       0.112
## Population                                          0.634        -0.194
## Area.sq.mi                             0.229  0.187  0.480  0.160 -0.234
## Pop.Density.per.sq.mi                 -0.358 -0.300        -0.180  0.107
## Coast.Area.Ratio                0.136        -0.282        -0.223 -0.168
## Net.migration                   0.107  0.117         0.147 -0.470  0.273
## Infant.Mortality.per.1000.births -0.284
## GDP.per.capita                  0.278
## Literacy.percent                0.250         0.180         0.186
## Phones.per.1000.people          0.286
## Arable.percent                        -0.453         0.185
## Crops.percent                         -0.355 -0.272         0.118
## Other.Land.Use.percent                 0.456        -0.135
## Climate                         0.119 -0.246  0.225                -0.106
## Birthrate                      -0.274
## Deathrate                      -0.123 -0.110  0.429        -0.132  0.189
## Agriculture                    -0.258 -0.115
## Industry                               0.226 -0.217  0.136  0.340  0.302
## Service                         0.207         0.119 -0.102 -0.247 -0.256
## Region.AusNZ                                                -0.111
## Region.Cen.E.Eur                      -0.164  0.263         0.467  0.243
## Region.E.Asia                                        0.126 -0.123 -0.101
## Region.LatCari                               -0.144 -0.242  0.194 -0.594
## Region.MENA                            0.176 -0.390                0.279
## Region.N.Amer                                0.136  0.265 -0.119 -0.157
## Region.SE.Asia                              -0.168  0.162
## Region.S.Asia                                        0.238 -0.102
## Region.SS.Africa               -0.213         0.179        -0.267  0.109
##                                Comp.7 Comp.8 Comp.9 Comp.10 Comp.11
## HDI.Score
## Life.Expectancy.at.Birth
## Expected.Years.of.Education
## Mean.Years.of.Education
## Gross.National.Income.per.Capita                            0.127
## Population                                          -0.140
## Area.sq.mi                                          -0.125
## Pop.Density.per.sq.mi                  0.172
## Coast.Area.Ratio                -0.213 -0.155        -0.133   0.185
```

```
## Net.migration                                    0.115
## Infant.Mortality.per.1000.births
## GDP.per.capita                                                0.112
## Literacy.percent
## Phones.per.1000.people
## Arable.percent
## Crops.percent                          -0.198        -0.244
## Other.Land.Use.percent                  0.142
## Climate                        -0.193                         0.147
## Birthrate
## Deathrate                      -0.103                         0.339
## Agriculture                                  0.118          -0.246
## Industry                       -0.153                         0.431
## Service                         0.155                        -0.120
## Region.AusNZ                    0.186 -0.407  0.686 -0.283
## Region.Cen.E.Eur                                             -0.216
## Region.E.Asia                  -0.180  0.717  0.127 -0.394
## Region.LatCari                                                0.238
## Region.MENA                     0.294        -0.181 -0.227  -0.234
## Region.N.Amer                         -0.214 -0.523  0.149  -0.354
## Region.SE.Asia                 -0.665 -0.140  0.257  0.379  -0.287
## Region.S.Asia                   0.430  0.219  0.233  0.548   0.290
## Region.SS.Africa               -0.152 -0.114 -0.142 -0.271   0.163
##                               Comp.12 Comp.13 Comp.14 Comp.15 Comp.16
## HDI.Score
## Life.Expectancy.at.Birth       -0.101   0.137
## Expected.Years.of.Education             0.156                  -0.105
## Mean.Years.of.Education         0.156                  0.201
## Gross.National.Income.per.Capita
## Population                     -0.209  -0.140           0.255
## Area.sq.mi                     -0.169          -0.143
## Pop.Density.per.sq.mi                  -0.119   0.137   0.218   0.151
## Coast.Area.Ratio                0.310   0.140  -0.585   0.157  -0.414
## Net.migration                  -0.210           0.415   0.294  -0.446
## Infant.Mortality.per.1000.births  0.103
## GDP.per.capita
## Literacy.percent                       -0.130           0.215
## Phones.per.1000.people
## Arable.percent                         -0.121          -0.370  -0.180
## Crops.percent                   0.114           0.199   0.475   0.170
## Other.Land.Use.percent                                  0.396   0.156
## Climate                        -0.317   0.686                   0.257
## Birthrate                                       0.101           0.102
## Deathrate                       0.119
## Agriculture                             0.261                  -0.331
## Industry                        0.200           0.198
## Service                        -0.101  -0.341  -0.208           0.288
## Region.AusNZ                    0.250           0.106  -0.102   0.142
## Region.Cen.E.Eur                0.155          -0.153   0.217  -0.180
## Region.E.Asia                   0.224   0.210   0.166  -0.187
```

```
## Region.LatCari                          -0.129   0.376          -0.245
## Region.MENA                    -0.255    0.122  -0.186
## Region.N.Amer                   0.538    0.182   0.182
## Region.SE.Asia                          -0.148          -0.144
## Region.S.Asia                   0.217    0.123  -0.147
## Region.SS.Africa                                                0.232
##                                Comp.17 Comp.18 Comp.19 Comp.20 Comp.21
## HDI.Score                                        0.108
## Life.Expectancy.at.Birth       -0.118            0.277  -0.164  -0.125
## Expected.Years.of.Education     0.363    0.203            0.179  -0.684
## Mean.Years.of.Education         0.249    0.239            0.226   0.282
## Gross.National.Income.per.Capita                         -0.117   0.250
## Population                     -0.191    0.132   0.101    0.102
## Area.sq.mi                                               -0.131
## Pop.Density.per.sq.mi          -0.324    0.243   0.231    0.292  -0.123
## Coast.Area.Ratio                                -0.115
## Net.migration                                   -0.316
## Infant.Mortality.per.1000.births  0.120                    0.255
## GDP.per.capita                          -0.110   0.185  -0.308   0.200
## Literacy.percent                0.347    0.187            0.206   0.207
## Phones.per.1000.people                  -0.132   0.252
## Arable.percent
## Crops.percent                   0.328   -0.358           -0.315
## Other.Land.Use.percent         -0.162             0.157          -0.121
## Climate                        -0.140            -0.196   0.180   0.187
## Birthrate                       0.128    0.244  -0.153    0.132   0.260
## Deathrate                               -0.489   0.329    0.334
## Agriculture                     0.126    0.130   0.337   -0.119
## Industry                                        -0.188
## Service                                 -0.213  -0.382
## Region.AusNZ                   -0.250                    0.118
## Region.Cen.E.Eur               -0.389            -0.293  -0.102  -0.160
## Region.E.Asia                           -0.135  -0.152
## Region.LatCari                 -0.120                    0.182
## Region.MENA                             -0.217            0.374
## Region.N.Amer
## Region.SE.Asia                  0.119   -0.181                    -0.103
## Region.S.Asia                   0.191
## Region.SS.Africa                         0.358           -0.242  -0.237
##                                Comp.22 Comp.23 Comp.24 Comp.25 Comp.26
## HDI.Score                       0.105    0.223
## Life.Expectancy.at.Birth       -0.503    0.427            0.356   0.304
## Expected.Years.of.Education     0.255    0.183                    -0.171
## Mean.Years.of.Education                          0.449            0.249
## Gross.National.Income.per.Capita  0.396   0.166  -0.207    0.194
## Population
## Area.sq.mi
## Pop.Density.per.sq.mi           0.103
## Coast.Area.Ratio
## Net.migration
```

```
## Infant.Mortality.per.1000.births  0.157   0.226  -0.499   0.480        
## GDP.per.capita                    0.405                  0.176  -0.158
## Literacy.percent                 -0.306  -0.298  -0.331   0.165  -0.143
## Phones.per.1000.people                   -0.145  -0.549  -0.516   0.340
## Arable.percent                                           0.160        
## Crops.percent                             0.101                       
## Other.Land.Use.percent                                             -0.136
## Climate                                                                 
## Birthrate                         0.112   0.514          -0.336   0.257
## Deathrate                                 0.152   0.115           0.116
## Agriculture                       0.120                                 
## Industry                         -0.268                                 
## Service                                                                 
## Region.AusNZ                                                            
## Region.Cen.E.Eur                  0.223                           0.222
## Region.E.Asia                                                     0.115
## Region.LatCari                    0.169  -0.164   0.106   0.141   0.291
## Region.MENA                              -0.234   0.107   0.148   0.257
## Region.N.Amer                                                           
## Region.SE.Asia                    0.101                           0.211
## Region.S.Asia                            -0.205                   0.202
## Region.SS.Africa                         -0.262           0.253   0.467
##                                  Comp.27 Comp.28 Comp.29 Comp.30 Comp.31
## HDI.Score                                 0.185           0.122        
## Life.Expectancy.at.Birth                 -0.108                       
## Expected.Years.of.Education                                             
## Mean.Years.of.Education           0.446  -0.151                       
## Gross.National.Income.per.Capita          0.588   0.150                 
## Population                                                       -0.526
## Area.sq.mi                                               -0.144   0.673
## Pop.Density.per.sq.mi                            -0.105           0.482
## Coast.Area.Ratio                                                        
## Net.migration                                                           
## Infant.Mortality.per.1000.births  0.396  -0.231  -0.156                 
## GDP.per.capita                   -0.255  -0.603  -0.163                 
## Literacy.percent                 -0.427                                 
## Phones.per.1000.people            0.250  -0.166   0.116                 
## Arable.percent                           -0.156   0.665                 
## Crops.percent                                     0.113                 
## Other.Land.Use.percent                   -0.193   0.615                 
## Climate                                                                 
## Birthrate                        -0.414  -0.197                         
## Deathrate                        -0.215                                 
## Agriculture                                              0.668   0.109
## Industry                                 -0.126           0.446        
## Service                                                  0.515        
## Region.AusNZ                                                            
## Region.Cen.E.Eur                 -0.198                                 
## Region.E.Asia                                                           
## Region.LatCari                                                          
```

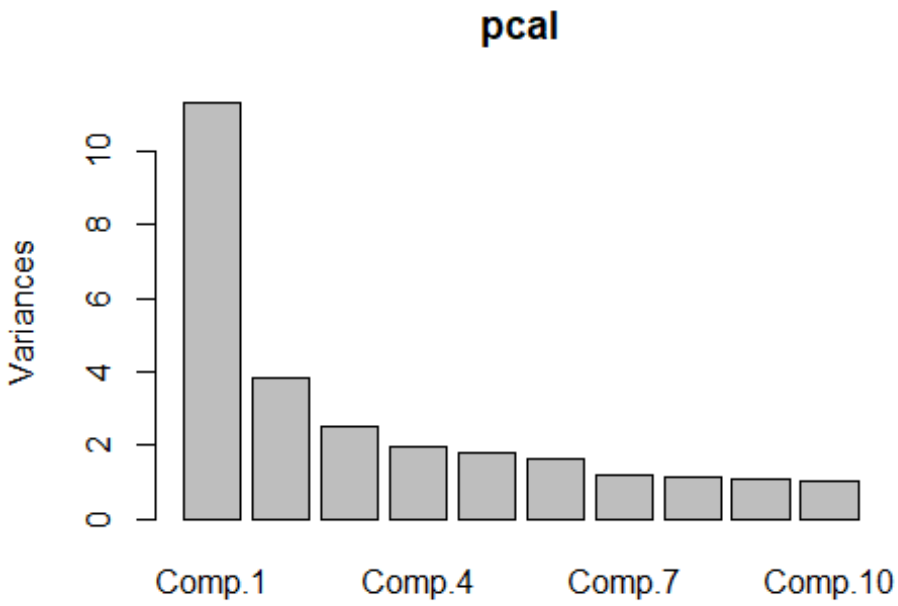
```
## Region.MENA                       -0.154
## Region.N.Amer
## Region.SE.Asia
## Region.S.Asia
## Region.SS.Africa
##                                  Comp.32
## HDI.Score                         0.864
## Life.Expectancy.at.Birth         -0.183
## Expected.Years.of.Education      -0.161
## Mean.Years.of.Education          -0.221
## Gross.National.Income.per.Capita -0.360
## Population
## Area.sq.mi
## Pop.Density.per.sq.mi
## Coast.Area.Ratio
## Net.migration
## Infant.Mortality.per.1000.births
## GDP.per.capita
## Literacy.percent
## Phones.per.1000.people
## Arable.percent
## Crops.percent
## Other.Land.Use.percent
## Climate
## Birthrate
## Deathrate
## Agriculture
## Industry
## Service
## Region.AusNZ
## Region.Cen.E.Eur
## Region.E.Asia
## Region.LatCari
## Region.MENA
## Region.N.Amer
## Region.SE.Asia
## Region.S.Asia
## Region.SS.Africa
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.031  0.031  0.031  0.031  0.031  0.031  0.031  0.031
## Cumulative Var  0.031  0.062  0.094  0.125  0.156  0.187  0.219  0.250
##                Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14 Comp.15
## SS loadings     1.000   1.000   1.000   1.000   1.000   1.000   1.000
## Proportion Var  0.031   0.031   0.031   0.031   0.031   0.031   0.031
## Cumulative Var  0.281   0.312   0.344   0.375   0.406   0.437   0.469
##                Comp.16 Comp.17 Comp.18 Comp.19 Comp.20 Comp.21 Comp.22
## SS loadings      1.000   1.000   1.000   1.000   1.000   1.000   1.000
## Proportion Var   0.031   0.031   0.031   0.031   0.031   0.031   0.031
```

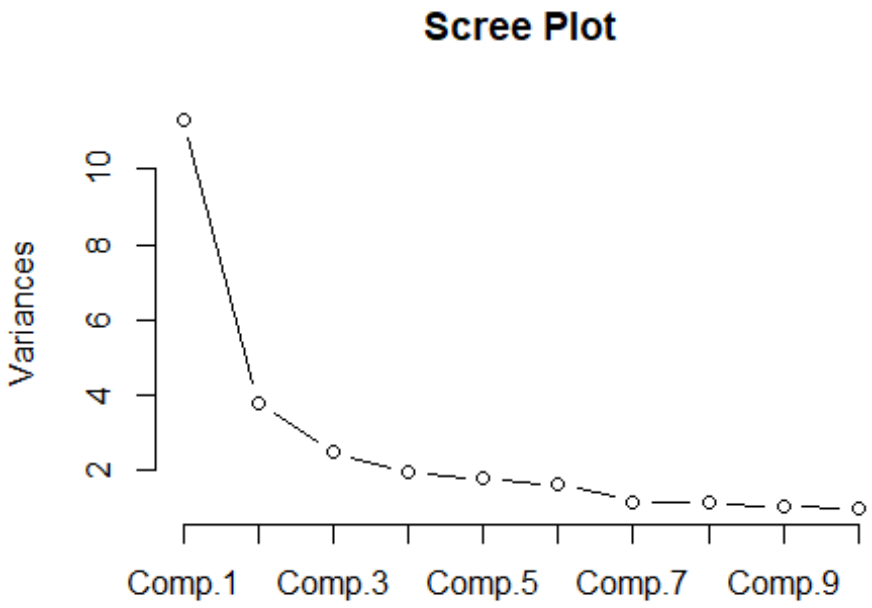
```
## Cumulative Var   0.500   0.531   0.562   0.594   0.625   0.656   0.687
##                Comp.23 Comp.24 Comp.25 Comp.26 Comp.27 Comp.28 Comp.29
## SS loadings      1.000   1.000   1.000   1.000   1.000   1.000   1.000
## Proportion Var   0.031   0.031   0.031   0.031   0.031   0.031   0.031
## Cumulative Var   0.719   0.750   0.781   0.812   0.844   0.875   0.906
##                Comp.30 Comp.31 Comp.32
## SS loadings      1.000   1.000   1.000
## Proportion Var   0.031   0.031   0.031
## Cumulative Var   0.938   0.969   1.000
```

This tells a similar story as before because HDI score is HIGHLY correlated with Component 32 (.864). Area sq mi is highly correlated with comp 31 (.673). Then agriculture would be next since it has a high correlation with Component 30. Other land use and arable percent are also multicollinear. This all makes sense logically because these features are mutually exclusive and are dependent on each others' calculated value.

DISCLAIMER: It is also important to note that Region.Western.Europe is excluded from the model since it is explained by the intercept (0 values in all other region dummies).

Let's now look at the scree plot of the eigenvalues





This shows us the importance of the first few components.

Scores of the components:

```
##     Comp.1     Comp.2    Comp.3      Comp.4     Comp.5      Comp.6
## 1 5.175588 -1.7033081 0.6554601 -0.02074137 -1.6199347  0.46277862
## 2 4.604376  0.2358656 0.8396040 -0.15183043 -0.7211958  1.07756950
## 3 4.299446  3.5512105 2.1296445 -2.52313167 -1.8961323 -0.44785387
## 4 4.577120  3.1169056 1.5314561 -0.34496318 -0.7133152  0.53488547
## 5 3.883556  1.5548817 1.5785306 -0.32933855 -0.7237401 -0.05012623
##         Comp.7      Comp.8      Comp.9      Comp.10    Comp.11    Comp.12
## 1 -0.047744728 -0.57281011 -0.33778276 -0.007054186  0.8779250 -0.4306642
## 2  0.079534176  0.35338573 -0.07631097  0.730619414  0.2826630 -1.1179809
## 3 -0.000828971  0.32857429  0.32623112  1.581041944 -0.5592776 -1.0686336
## 4 -0.772130933  0.18514636 -0.05790453  1.024657218  1.3047392 -0.5150073
## 5 -0.218092300  0.07668175 -0.05243349  0.638292556  0.6107551 -0.9818318
##      Comp.13    Comp.14    Comp.15    Comp.16      Comp.17     Comp.18
## 1 -0.3089150 -0.4740435 -1.0677930 -0.9450420  0.631270380  0.55532814
## 2 -0.1020641  1.6584938  0.4833660  1.0431751 -0.004716887  0.05374748
## 3  1.1396300 -1.5798728 -0.5794377 -0.6987329  0.635245393  0.37896393
## 4  0.7191781 -0.9934749 -0.5096451 -0.5740363 -0.233521139  0.95610525
## 5  0.3747044 -0.6379355 -0.3903499  0.0555760  0.191661105 -0.03838195
##       Comp.19    Comp.20     Comp.21     Comp.22      Comp.23      Comp.24
## 1 -0.16705639  0.4728736  0.28404524  0.06083372  0.355299310  0.008218471
## 2  0.79858255 -0.1922257  0.22269628 -0.05500484 -0.007848431 -0.010451781
## 3 -0.09412429 -0.2756977 -0.16548488  0.06679950  0.062321739 -0.421059191
## 4  0.41156465  0.3899458  0.48388236 -0.08472126  0.148797565  0.173668314
## 5  0.27562548 -0.1482611 -0.01099987 -0.25292171 -0.010647661  0.049841697
##       Comp.25     Comp.26     Comp.27     Comp.28     Comp.29     Comp.30
## 1 -0.08985138 -0.02409604  0.11476269 -0.06856710  0.05962942 -0.05324456
## 2 -0.15617129 -0.01171687  0.45463180 -0.01594777  0.04496703  0.08564744
## 3 -0.53697215 -0.01674172 -0.09190221 -0.09469452 -0.55018409  0.07806191
## 4 -0.07986937 -0.10266081 -0.01187825 -0.04270207 -0.11773428  0.08465605
## 5 -0.09301560 -0.40645393 -0.21577262  0.06626700  0.21934331 -0.01414491
##       Comp.31      Comp.32
## 1 -0.03809410  0.002925537
## 2 -0.02789215  0.035079381
## 3 -0.00399339 -0.031871110
## 4 -0.04024532  0.033203837
## 5 -0.03195102 -0.012052959
```

This also shows the relative importance of the different components.

I am going to start back-fitting my model using statistical significance (p-value) in order to find our final regression model to predict happiness score. The VIF analysis and PCA has led me to exclude HDI.Score and Area.sq.mi from my regression model equation. Feature removal is no small decision, but the justification behind this one is multicollinearity. The rest of the feature selection will be done by backfitting by p-value. Any feature that has a p-value of greater than 0.05 will be considered not statistically significant.

Now let's make training and test datasets in order to create and evaluate our model. The division strategy I am going to use is taking a random sample without replacement. I believe this will give an unordered subset of the data that should be representative of the range of happiness scores.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.695   4.380   5.420   5.456   6.177   7.526

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.905   4.383   5.314   5.370   6.296   7.509
```

They look pretty similar! Let's move forward with these training and test sets.

```
##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth + Expected.Years.o
f.Education +
##     Mean.Years.of.Education + Gross.National.Income.per.Capita +
##     Population + Pop.Density.per.sq.mi + Coast.Area.Ratio + Net.migration
+
##     Infant.Mortality.per.1000.births + GDP.per.capita + Literacy.percent +
##     Phones.per.1000.people + Arable.percent + Crops.percent +
##     Other.Land.Use.percent + Climate + Birthrate + Deathrate +
##     Agriculture + Industry + Service + Region.AusNZ + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.N.Amer +
##     Region.SE.Asia + Region.S.Asia + Region.SS.Africa, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35211 -0.26357  0.00542  0.27019  1.19922
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.409216   1.776413   2.482  0.01496 *
## Life.Expectancy.at.Birth         0.627332   0.652336   0.962  0.33885
## Expected.Years.of.Education     -0.004006   0.049196  -0.081  0.93528
## Mean.Years.of.Education          0.328199   0.184811   1.776  0.07921 .
## Gross.National.Income.per.Capita 0.673835   0.211022   3.193  0.00195 **
## Population                       0.062004   0.061384   1.010  0.31521
## Pop.Density.per.sq.mi            0.057551   0.108463   0.531  0.59703
## Coast.Area.Ratio                -0.070389   0.069066  -1.019  0.31092
## Net.migration                    0.058743   0.075110   0.782  0.43626
## Infant.Mortality.per.1000.births -0.068146   0.188571  -0.361  0.71868
## GDP.per.capita                   0.013662   0.192647   0.071  0.94363
## Literacy.percent                 0.118961   0.161535   0.736  0.46342
## Phones.per.1000.people           0.167908   0.204435   0.821  0.41368
## Arable.percent                   0.193228   0.233498   0.828  0.41017
## Crops.percent                   -0.060780   0.087584  -0.694  0.48953
## Other.Land.Use.percent           0.308639   0.216846   1.423  0.15818
## Climate                          0.062539   0.072672   0.861  0.39181
## Birthrate                        0.225117   0.179230   1.256  0.21243
## Deathrate                       -0.189387   0.108586  -1.744  0.08463 .
## Agriculture                      0.473199   0.388562   1.218  0.22655
## Industry                         1.094533   2.045669   0.535  0.59397
## Service                          1.034471   2.005123   0.516  0.60721
```

```
## Region.AusNZ                      -0.024041   0.618215  -0.039  0.96907
## Region.Cen.E.Eur                  -0.676182   0.277672  -2.435  0.01690 *
## Region.E.Asia                     -1.105588   0.327815  -3.373  0.00111 **
## Region.LatCari                     0.427772   0.282372   1.515  0.13338
## Region.MENA                       -0.627934   0.323644  -1.940  0.05556 .
## Region.N.Amer                     -0.144571   0.458845  -0.315  0.75345
## Region.SE.Asia                    -0.454491   0.345540  -1.315  0.19182
## Region.S.Asia                     -0.133049   0.411876  -0.323  0.74744
## Region.SS.Africa                  -0.203061   0.342615  -0.593  0.55492
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5091 on 88 degrees of freedom
## Multiple R-squared:  0.8589, Adjusted R-squared:  0.8109
## F-statistic: 17.86 on 30 and 88 DF,  p-value: < 2.2e-16
```

Our initial Adjusted R^2 value is .8109, which is very good!!

Let's remove Region Aus&NZ, as it is the highest p-value.

Now remove GDP per capita. This is a bit surprising - I would have expectd GDP to have a bigger impact.

Next remove expected year of education.

Next remove region North America.

Next remove region South Asia.

Next remove infant mortality.

Next remove region Sub-saharan Africa.

Next remove service.

Next remove industry.

Next remove population density.

Next remove crops percent.

Next remove literacy percent.

Next remove net migration.

Next remove population.

Next remove birthrate.

Next remove climate.

Next remove region SE Asia.

Next remove phones per 100 people.

Next remove arable percent.

Next remove deathrate.

Next remove region Middles East and North Africa.

Final model:

```
m.mlreg <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+ Coast.Area.Ra
tio+ Agriculture +Region.Cen.E.Eur + Region.E.Asia + Region.LatCari , data =
world_train)
summary(m.mlreg)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Ed
ucation +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio + Agriculture +
##     Region.Cen.E.Eur + Region.E.Asia + Region.LatCari, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42631 -0.30689  0.04681  0.35102  1.19265
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.61021    0.22398  20.583  < 2e-16 ***
## Life.Expectancy.at.Birth         1.28842    0.35346   3.645  0.00041 ***
## Mean.Years.of.Education          0.47835    0.10233   4.675 8.41e-06 ***
## Gross.National.Income.per.Capita 0.75876    0.13930   5.447 3.16e-07 ***
## Coast.Area.Ratio                -0.14867    0.05739  -2.591  0.01087 *
## Agriculture                      0.36780    0.12143   3.029  0.00306 **
## Region.Cen.E.Eur                -0.76376    0.16404  -4.656 9.07e-06 ***
## Region.E.Asia                   -0.77697    0.24257  -3.203  0.00178 **
## Region.LatCari                   0.81334    0.14239   5.712 9.63e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5071 on 110 degrees of freedom
## Multiple R-squared:  0.8251, Adjusted R-squared:  0.8124
## F-statistic: 64.87 on 8 and 110 DF,  p-value: < 2.2e-16
```

Now all of our features are statistically significant! For good measure, I will also be demonstrating automatically building a final model using the step function. This model does feature selection based on the Akaike information criterion (AIC). This is a way of measuring information gained to the prediction from each feature. Not every feature is always statistically significant when using AIC, however. Let's compare models.

```
#make model based on AIC minus HDI.score and area.sq.mi
m.step <- step(lm(Happiness.Score ~ Life.Expectancy.at.Birth + Expected.Years
```

```
.of.Education +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Infant.Mortality.per.1000.births + GDP.per.capita +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service + Region.AusNZ +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
    Region.N.Amer + Region.SE.Asia + Region.S.Asia + Region.SS.Africa, data =
world_train), trace = 0)

m.step

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Ed
ucation +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio + Phones.per.1000.
people +
##     Arable.percent + Other.Land.Use.percent + Deathrate + Agriculture +
##     Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA,
##     data = world_train)
##
## Coefficients:
##                       (Intercept)          Life.Expectancy.at.Birth
##                            5.0898                            0.7280
##          Mean.Years.of.Education  Gross.National.Income.per.Capita
##                            0.3637                            0.6452
##                 Coast.Area.Ratio            Phones.per.1000.people
##                           -0.1211                            0.2162
##                   Arable.percent            Other.Land.Use.percent
##                            0.3037                            0.4052
##                        Deathrate                       Agriculture
##                           -0.2037                            0.3085
##                 Region.Cen.E.Eur                      Region.E.Asia
##                           -0.7390                           -1.0288
##                   Region.LatCari                       Region.MENA
##                            0.5051                           -0.5761
```

It looks like the features are pretty similar. The AIC method kept more features, though.

I am going to treat the p-value model as our final one.

I am going to make a multiple regression model based on the factors:

Life.Expectancy.at.Birth, Mean.Years.of.Education, Gross.National.Income.per.Capita, Coast.Area.Ratio, Agriculture, Region.Cen.E.Eur, Region.E.Asia, Region.LatCari.

I chose these because I think they will be useful in determining a nation's happiness score, they are relevant to our objective, and all are statistically signifcant.

Some interesting things to note already. It seems the region in the world has a big imapct on happineess, as does some human development indicators such as life expectancy, gross national income, etc.

It is important to note that the subset of countries chose by our dataPartition has a big impact on the significance found in features. Different subsets will likely yield different significant features.

The residuals tell us how much our fitted values are off of actual values for each case. The majority of the cases are pretty good. Our median residual is only 0.05.

The multiple R squared values, which is also known as the coefficient of determination tells us how well the model overall explains the values of the dependent/response variable. Our model explains about 83% of the variation in happiness score.

The Adjusted R squared of the model is .81, which is relatively high to start with. This means that the selected features of the multiple regression equation explain the variations in the happiness score data fairly well. R squared is a measure of fit, and the closer to 1 the stonger the fit. .81 is fairly high, but this number could be improved by in a variety of ways such as adding non-linear relationships or converting some numeric variables to a binary indicator.

The standard error of 0.507 could be used later to calculate confidence intervals. This tells us the standard error of the model between fitted and actual values.

The p-value indicates statistical significance. The higher the p-value the more likely something can be attributed to chance. We are looking for very low p-values to make for a stronger model. All of our principal components (features) that make up our model are statistically significant. Gross national income has a p-value of 3.2e-7, which indicates very high statistical significance. This variable's impact on happiness score is not due to random chance. The same applies to the others.

The overall p-value for the model is 2.2e-16. This means that the overall model statistical significance is very high, which is a good sign.

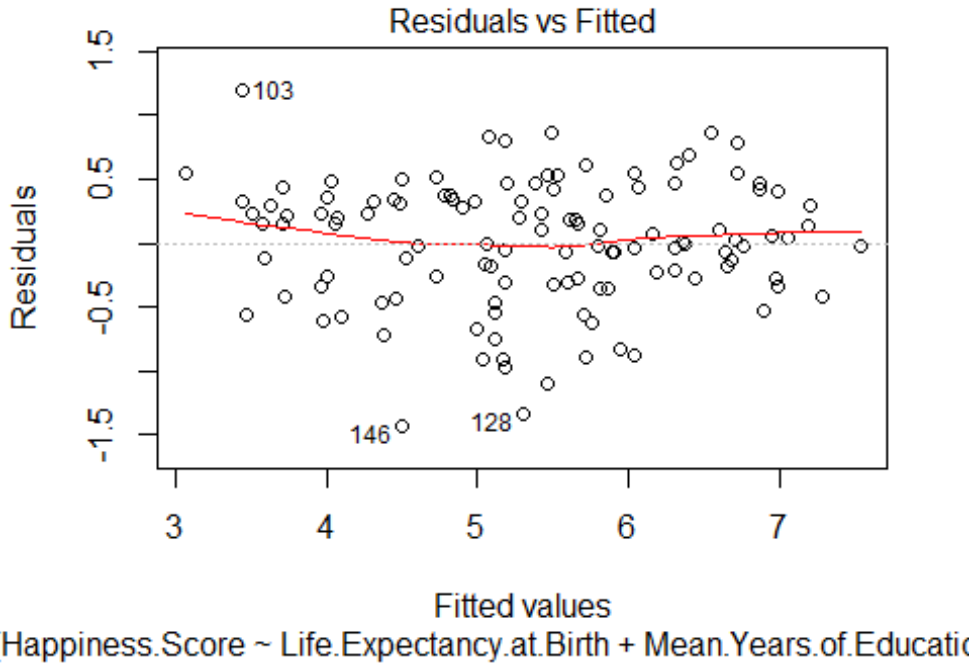
```
##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Ed
ucation +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio + Agriculture +
##     Region.Cen.E.Eur + Region.E.Asia + Region.LatCari, data = world_train)
##
## Coefficients:
##                       (Intercept)          Life.Expectancy.at.Birth
##                            4.6102                            1.2884
##          Mean.Years.of.Education  Gross.National.Income.per.Capita
##                            0.4784                            0.7588
##                  Coast.Area.Ratio                       Agriculture
##                           -0.1487                            0.3678
##                  Region.Cen.E.Eur                     Region.E.Asia
```

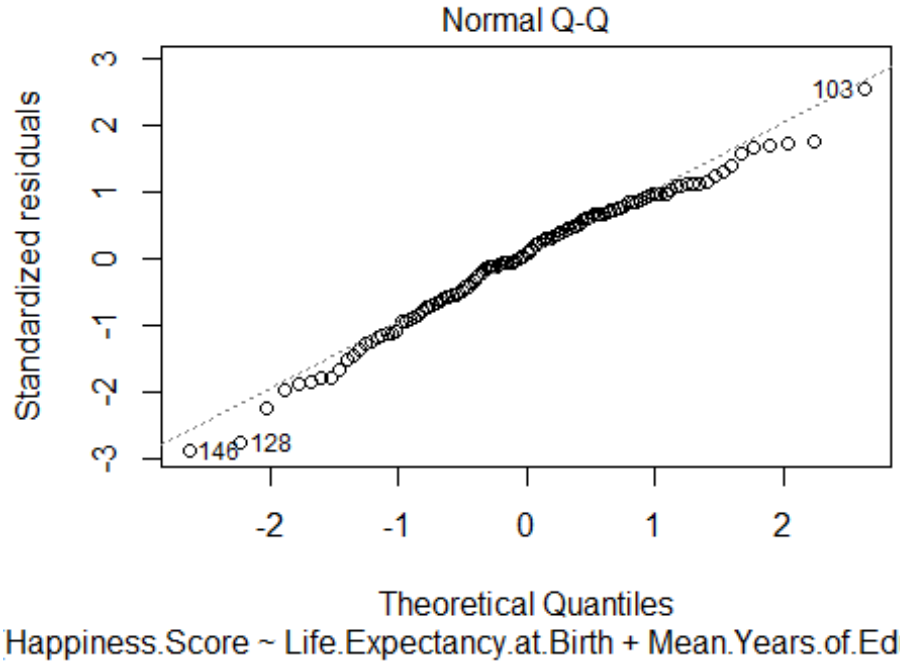
```
##                             -0.7638                                 -0.7770
##                   Region.LatCari
##                           0.8133
```

The final linear regression equation is: y = 5.7989053 + (Life.Expectancy.at.Birth)1.288 + (Mean.Years.of.Education)0.4783527 + (Gross.National.Income.per.Capita)0.7587606 - (Coast.Area.Ratio)0.1486733 + (Agriculture)0.3677959 - (Region.Cen.E.Eur)0.7637579 - (Region.E.Asia)0.7769728 + (Region.LatCari)0.8133442.

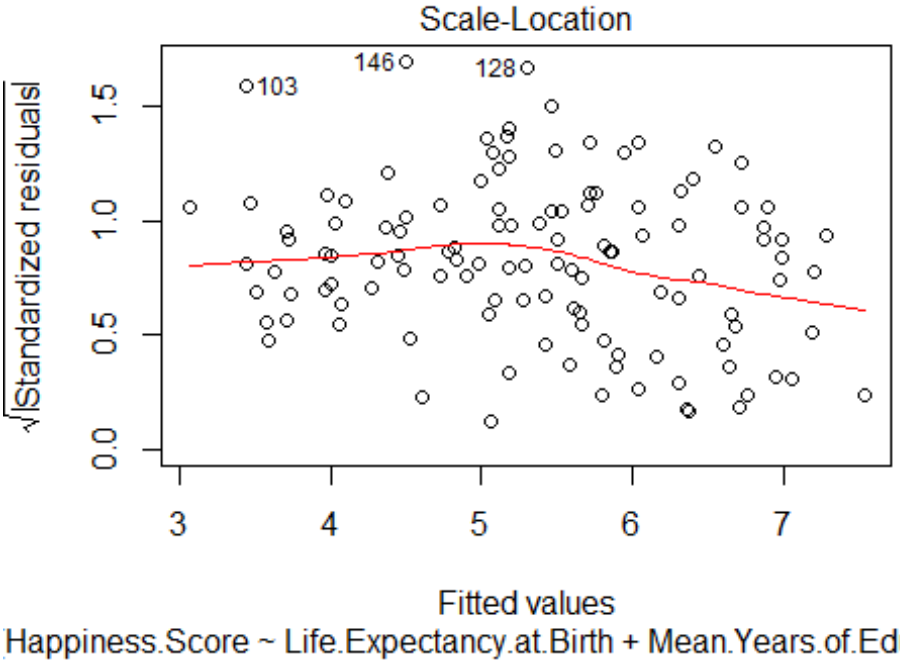
This tells us, on average, how much each unit change in one of the features will impact the end result happiness score. For example, If a nation is in the Central/East Europe region, the happiness score will decrease by, on average, 0.763. Remember that some of these values were transformed, and must be reverted back in order to get the actual happiness score.

Now let's try to evaluate our model using plot to derive some insights.

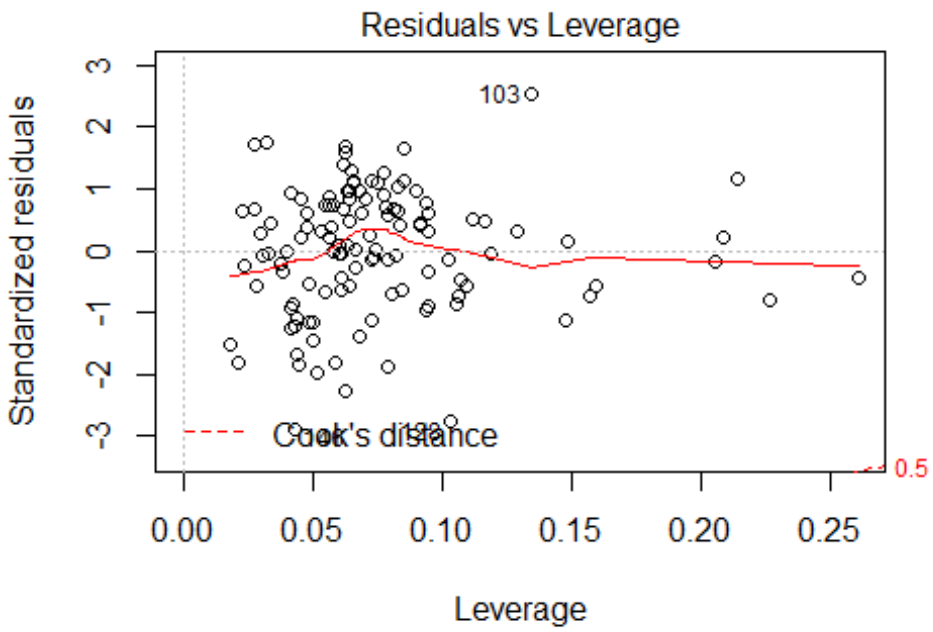




Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Educatic



Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Educatic



Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Educati Residuals vs. fitted: This helps us to detect non-linearity if the line is parabolic. Our line is a bit curved, but not entirely. This suggests there may be some linear regression features that could be described in a non-linear fashion. Also, all the labeled points are considered outliers, so this plot is helpful in identifying additional ones. Lastly, if the plot was in the shape of a funnel (which it is not), it would indicate heteroskedasticity. We could fix this problems with some form of dist transform such as log or sqrt.

Normal Q-Q: The more straight line in the Q-Q plot, the better. The majority of the points follow a straight line, but toward the beginning and end there's a very slight deviation. The more straight the line the more normally distributed the data. This linearity could be fixed with a non-linear transform if needed, but it looks very very good! Our transforms helped to make the data more normally distributed.

Scale-Location:

If there is no discernable patter in this plot it means it is good. I can't detect a pattern, meaning we don't have to fix it's non-linearity or heteroskedasticity.

Residuals vs Leverage:

The labeled points indicate additional outliers with a lot of leverage (they skew the model). These could theoretically be removed to help improve our model.

Now let's calculate the RMSE for this model on the training data.

```
## [1] 0.4875172
```

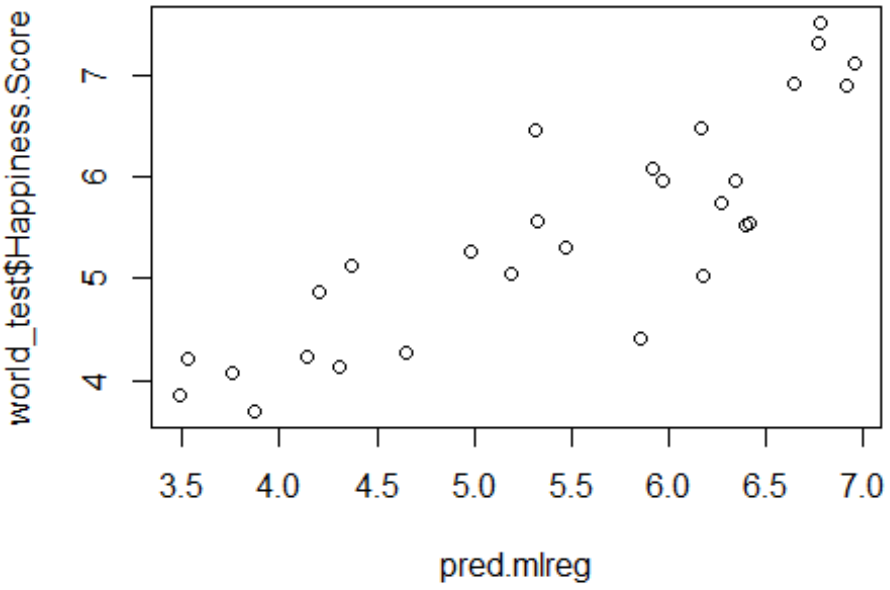
The RMSE of this model is 0.4875172. Which isn't the most meaningful on it's own but would be more meaningful when compared to other models. However, this is saying that on average, the square error between actual and predicted squared is 0.4875172 for happiness score.

Let's also calculate the Mean Absolute Error. This is also a measure of fit and accuracy of a regression model.

This is saying that on average the predicted value from the regression model is 0.389752 off the actual value. This seems very good!

These are both against the same data it was tested on, so let's also evaluate this regression model against the test data with the holdout method.

```
## [1] 0.8543146
```



85% correlation is pretty good! I will determine the best model later on. Let's look at our confidence interval for each prediction.

```
##         fit      lwr      upr
## 1  6.780507 6.523563 7.037450
## 8  6.771551 6.509736 7.033367
## 12 6.954193 6.641648 7.266738
## 17 6.646931 6.439761 6.854101
## 18 6.914875 6.668941 7.160810
## 30 6.164042 5.915214 6.412870
```

Here we can see the lower and upper limits of the 95% confidence interval for each prediction of this statistical linear model.

Regression Tree

Next I will build a decision tree for regression to try to predict happiness score of a nation.

I am using tree structure for this ML task because it offers transparent insight into the process. It will be very helpful for the to know what contributes to happy nation.

Install the rpart package for regression trees

Train the rpart model. We do not need to use the normally distributed data for this, so I will be making new train/test sets.

```
## n= 119
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 119 161.718500 5.369832
##    2) Life.Expectancy.at.Birth< 69.8 43  23.592060 4.287744
##      4) Literacy.percent< 86.8 35   9.803374 4.045057
##        8) Gross.National.Income.per.Capita< 3539.5 24   5.436969 3.855667
*
##        9) Gross.National.Income.per.Capita>=3539.5 11   1.627336 4.458273
*
##      5) Literacy.percent>=86.8 8   2.708686 5.349500 *
##    3) Life.Expectancy.at.Birth>=69.8 76  59.290000 5.982066
##      6) Agriculture>=0.1065 24   8.070207 5.157708
##       12) Region=Central and Eastern Europe,Middle East and Northern Afric
a,Southeastern Asia,Southern Asia 17   3.243062 4.939588 *
##       13) Region=Eastern Asia,Latin America and Caribbean 7   2.054122 5.6
87429 *
##      7) Agriculture< 0.1065 52  27.382740 6.362538
##       14) Region=Central and Eastern Europe,Eastern Asia,Sub-Saharan Afric
a 15   3.540084 5.628467 *
##       15) Region=Australia and New Zealand,Latin America and Caribbean,Mid
dle East and Northern Africa,North America,Southeastern Asia,Western Europe 3
7  12.482870 6.660135
##         30) GDP.per.capita< 25250 24   5.597872 6.374375
##           60) Other.Land.Use.percent< 78.175 7   1.399606 5.960429 *
##           61) Other.Land.Use.percent>=78.175 17   2.504908 6.544824 *
##         31) GDP.per.capita>=25250 13   1.307075 7.187692 *
```

This output shows the logic behind the tree. For example, any line with a * indicates a terminal node. A country with life expectancy < 69.8, literacy percent < 86.8, and GNI <3539 will be predicted to have a happiness score of 3.86.

Let's see what the most important variables are.

```
m.regtree$variable.importance
```

```
## Infant.Mortality.per.1000.births         Life.Expectancy.at.Birth
##                        86.4427581                       78.8363998
##                            Region                   GDP.per.capita
##                        77.4095714                       76.4714385
##           Phones.per.1000.people                        Birthrate
##                        69.3964713                       64.1283172
##                       Agriculture          Mean.Years.of.Education
##                        25.3310941                       20.4064294
## Gross.National.Income.per.Capita                 Literacy.percent
##                        19.6236503                       18.0717396
##      Expected.Years.of.Education                    Arable.percent
##                        17.6934842                        7.3030644
##           Other.Land.Use.percent                         Deathrate
##                         6.2372698                        6.0585498
##                     Net.migration            Pop.Density.per.sq.mi
##                         2.7699996                        2.0373234
##                          Industry                   Crops.percent
##                         1.7430435                        1.4514492
##                        Population
##                         0.7922923
```

Infant mortality, life expectancy, and region are the three most important. Not too far off from the multiple linear regression model!

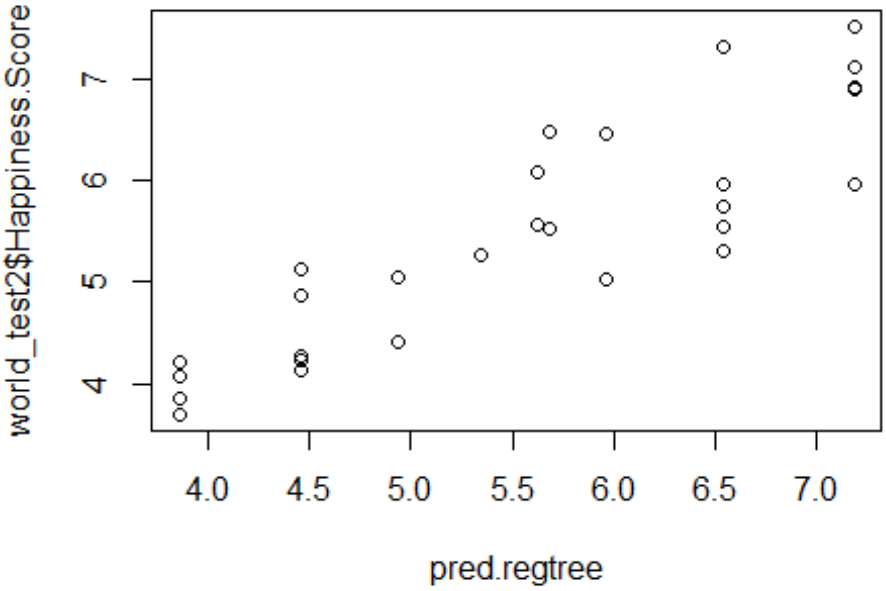
Now let's evaluate the tree performance using the predict() function.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.856   4.458   5.658   5.577   6.545   7.188

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.695   4.380   5.420   5.456   6.177   7.526
```

These quantiles show that our model is predicting pretty closely.

Let's check our the correlation between our prediction and the actuals.

```
## [1] 0.8771465
```



Correlation of .877 is pretty strong! It gives us some idea that our model predicitons aren't too far off actual values.

But now let's measure regression tree model performance using an error metric - Mean Absolute Error.

Let's try it out!

```
## [1] 0.4541424
```

The MAE is 0.4541424. This isn't super meaningful until we have a reference to compare it to (another model). However, this does tell us that on average our model is 0.4541424 happiness score away from the actual value.

Let's find the MAE if we compared each case to the mean quality value instead of our model. Is our model better than just guessing the mean?

```
## [1] 0.9147143
```

The MAE in this instance is 0.9147143, so my model is definitely better than only guessing the mean.

Now it is time to improve our model's performance by using a model tree. This replaces leaf nodes with regression models.

Examine the tree

```
m.regtree2
```

```
## M5 pruned regression tree:
## (using smoothed linear models)
##
## Life.Expectancy.at.Birth <= 65.3 :
## |   Mean.Years.of.Education <= 5.7 : LM1 (20/33.77%)
## |   Mean.Years.of.Education >  5.7 : LM2 (9/20.419%)
## Life.Expectancy.at.Birth >  65.3 :
## |   GDP.per.capita <= 8950 :
## |   |   Region=Latin America and Caribbean,Western Europe,North America,Au
stralia and New Zealand <= 0.5 : LM3 (37/55.92%)
## |   |   Region=Latin America and Caribbean,Western Europe,North America,Au
stralia and New Zealand >  0.5 : LM4 (12/48.537%)
## |   GDP.per.capita >  8950 :
## |   |   Region=Latin America and Caribbean,Western Europe,North America,Au
stralia and New Zealand <= 0.5 :
## |   |   |   Industry <= 0.315 : LM5 (7/20.561%)
## |   |   |   Industry >  0.315 : LM6 (12/36.159%)
## |   |   Region=Latin America and Caribbean,Western Europe,North America,Au
stralia and New Zealand >  0.5 : LM7 (22/47.705%)
##
## LM num: 1
## Happiness.Score =
##  + 4.3757
##
## LM num: 2
## Happiness.Score =
##  + 4.5415
##
## LM num: 3
## Happiness.Score =
##  + 5.283
##
## LM num: 4
## Happiness.Score =
##  + 5.5986
##
## LM num: 5
## Happiness.Score =
##  + 5.9682
##
## LM num: 6
## Happiness.Score =
##  + 6.0633
##
## LM num: 7
## Happiness.Score =
##  + 6.3225
##
## Number of Rules : 7
```

The LM leaf nodes are the linear models for given conditions/route through a tree. This helps to increase model performance rather than just predicting one value for a whole class.

Let's see if it actual performs better on the test data.

Let's see if the new model is predicting a wider range of values

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.376   5.098   5.441   5.471   6.323   6.323
```

Look's like the predictions are actually missing the very happy and very unhappy countries. The range is not very wide.

```
## [1] 0.8205657
```

The correlation is lower now as a result.

```
## [1] 0.5336717
```

The MAE is worse too. So overall, the new model tree model doesn't perform as well as the plain regression tree model, but not by much.

Neural Network

Neural Networks tend to work best with values scaled to around 0. Let's normalize the data. I will be using the standard range of 0-1, so I will be dummy coding the region category, as well.

Now apply the normalize function to every row of the world dataset.

```
world_standard <- as.data.frame(lapply(world_standard, normalize))
```

Check to make sure it works.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3198  0.5213  0.5369  0.7339  1.0000

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.905   4.383   5.314   5.386   6.296   7.526
```

It works!

Now lets make train and test sets. 80% to training and 20% to testing.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1710  0.3192  0.5444  0.5520  0.7081  1.0000

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3198  0.5213  0.5334  0.7339  0.9963
```

The distribution of happiness scores between the two sets is pretty similar!

Now our two sets are created. We have to be careful about overfitting!

I am going to use the neuralnet package to implement this NN.

Now let's train our model.

Create an easy list of predictors to pull from for the neural net (nn) model.

```
## Happiness.Score ~ Life.Expectancy.at.Birth + Expected.Years.of.Education +
##     Mean.Years.of.Education + Gross.National.Income.per.Capita +
##     Population + Pop.Density.per.sq.mi + Coast.Area.Ratio + Net.migration
+
##     Infant.Mortality.per.1000.births + GDP.per.capita + Literacy.percent +
##     Phones.per.1000.people + Arable.percent + Crops.percent +
##     Other.Land.Use.percent + Climate + Birthrate + Deathrate +
##     Agriculture + Industry + Service + RegionAustralia.and.New.Zealand +
##     RegionCentral.and.Eastern.Europe + RegionEastern.Asia + RegionLatin.Am
erica.and.Caribbean +
##     RegionMiddle.East.and.Northern.Africa + RegionNorth.America +
##     RegionSoutheastern.Asia + RegionSouthern.Asia + RegionSub.Saharan.Afri
ca

m.nn <- neuralnet(formula2, data = world_train3)
```

Let's visualize the model to help our understanding.

This NN with only 1 node in the hidden layer is similar to a regression model.

Now let's evaluate the model performance.

We can't use a confusion matrix since this is not a classification problem. So let's instead measure the correlation between the actual strength and predicted strength.

```
##           [,1]
## [1,] 0.8603149
```

0.8603149 is a pretty strong correlation!

Now let's try to improve model performance. Let's tune the model by increasing the number of hidden nodes to 5.

Now let's see if it actually did improve.

```
##           [,1]
## [1,] 0.8009931
```

Adding 5 hidden nodes actually made the prediction performance go down to 0.8009931. The added complexity did not help the model. Different random seeds will have varying results.

```
#create a data frame that compares predicted values to actual values
results <- data.frame(actual = world_test3$Happiness.Score, prediction = m.nn
_results$net.result)
```

```
#show a sample of actual vs. predicted
round(results[1:10,],2)
```

```
##    actual prediction
## 1    1.00       0.89
## 8    0.96       0.86
## 12   0.91       0.86
## 17   0.87       0.86
## 18   0.87       0.92
## 30   0.77       0.73
## 32   0.77       0.54
## 43   0.69       0.63
## 48   0.66       0.79
## 49   0.66       0.73
```

here we can see the differences between actual and predicted values. Some of them are much closer than others.

k-Nearest Neighbors

Though I believe my data has a dimensionality that is too high for kNN to be super effective, let's test it out and see. You never know! Since kNN relies on distance measure to find nearest neighbors, the data must be standardized. I am going to use the same world_standard data that we used for the neural network.

Because our target variable (Happiness.Score) is continuous, we can't use typical kNN classification. Let's use kNN regression instead. We have to average out the k nearest neighbors in order to arrive at a predicted value on a continuous scale.

I am choosing to use the caret package "knnreg" method. This is used to "return the average value for the neighbours," as opposed to giving you the mode of a categorical variable among the neighbors.

In order to properly train and validate the kNN, I must remove the target colum from the data we use for training/testing.

I am choosing to start with k = 11 because that is close to the square root of 119, which is the size of the cases in the training data set. In practice, it would be necessary to try several different values of k in order to arrive at the best one. In this instance, I made the decision to just predict using the k=11.

kNNregTrain is a modification of the "knn" function in the class package. train dictates what the training data set is, while test dictates the test dataset. y refers to what is the target vector in which the knn algorithm can learn patterns and average out neighbors using the training dataset. k is the number of nearest neighbors analyzed (find distance for).

```
##      test3.scores pred.knnreg
## [1,]    1.0000000   0.8387992
```

```
## [2,]    0.9584506   0.8874899
## [3,]    0.9119238   0.8710629
## [4,]    0.8708072   0.8710629
## [5,]    0.8660463   0.8710629
## [6,]    0.7738585   0.7299089
```

This shows that the predictions are actually quite good. When one is on the high end, so is the prediction, even though the values don't match up exactly.

Comparison of all models

Though I did some preliminary evaluation earlier with the holdout method, I am now going to compare all models using 10-fold Cross Validation. This will be especially helpful given the size of my dataset. I will compare the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of the 6 models I made. These models are Multiple Linear Regression (m.mlreg), Regression Tree(m.regtree), Regression Tree with linear models at each terminal node (m.regtree2), Neural Networks with 1 hidden node (m.nn), Neural Networks with 5 hidden nodes (m.nn2), and k-nearest neighbors regression (m.knnreg).

```
##               MAE      RMSE
## Fold01 0.4450784 0.4927008
## Fold02 0.4801853 0.6637326
## Fold03 0.6719529 0.7648012
## Fold04 0.4293377 0.5234877
## Fold05 0.2879256 0.3953018
## Fold06 0.5166429 0.6402256
## Fold07 0.2891041 0.3568858
## Fold08 0.4641370 0.5781349
## Fold09 0.3586869 0.4890157
## Fold10 0.3487544 0.4199365

## [1] 0.4291805

## [1] 0.5324223
```

The average MAE between the 10 folds for the Multiple Linear Regression Model is 0.4291805 and the average RMSE is 0.5324223.

Now let's look at the first regression tree model.

```
##               MAE      RMSE
## Fold01 0.6793305 0.7795853
## Fold02 0.5501552 0.6832267
## Fold03 0.5874614 0.6513424
## Fold04 0.6014492 0.7226679
## Fold05 0.4164481 0.5502912
## Fold06 0.5474025 0.7102669
## Fold07 0.5774327 0.6785706
## Fold08 0.6000950 0.7570093
```

```
## Fold09 0.5013498 0.6176150
## Fold10 0.4904004 0.5771354

## [1] 0.5551525

## [1] 0.6727711
```

The average MAE between the 10 folds for the Regression Tree Model is 0.5551525 and the average RMSE is 0.6727711.

Now let's look at the second regression tree model.

```
##               MAE      RMSE
## Fold01 0.7610169 0.8549888
## Fold02 0.5181112 0.6655477
## Fold03 0.5582186 0.7115356
## Fold04 0.6418967 0.7691979
## Fold05 0.6085230 0.7163760
## Fold06 0.6661608 0.9437810
## Fold07 0.5281894 0.6321273
## Fold08 0.5099813 0.6433344
## Fold09 0.6102045 0.6799756
## Fold10 0.4583585 0.6064697

## [1] 0.5860661

## [1] 0.7223334
```

The average MAE between the 10 folds for the Regression Tree Model is 0.5860661 and the average RMSE is 0.7223334.

Now let's look at the first neural network.

The predictions are normalized, so we have to reverse this.

```
##               MAE      RMSE
## Fold01 0.5827220 0.6958636
## Fold02 0.6534532 0.7985916
## Fold03 0.8215584 0.9023990
## Fold04 1.0078615 1.2224748
## Fold05 0.4873797 0.5870419
## Fold06 0.6487698 0.7986104
## Fold07 0.4638579 0.6570713
## Fold08 0.2959129 0.3384436
## Fold09 0.4429736 0.5076976
## Fold10 0.4308708 0.6073707

## [1] 0.583536

## [1] 0.7115564
```

This neural network model gives an average MAE of 0.583536 and an average RMSE of 0.7115564.

Now let's look at the second neural network. This one has 5 nodes in the hidden layer.

```
##               MAE      RMSE
## Fold01 0.7239713 0.9418257
## Fold02 0.9480370 1.5025846
## Fold03 1.3493231 1.5926444
## Fold04 0.8798172 1.3002129
## Fold05 0.9440457 1.0980664
## Fold06 0.8807669 1.1884811
## Fold07 0.8841174 1.1209755
## Fold08 0.5972530 0.7407766
## Fold09 0.7116596 0.8420012
## Fold10 0.7117115 1.1948547

## [1] 0.8630703

## [1] 1.152242
```

This neural network model gives an average MAE of 0.8630703 and an average RMSE of 1.1522423.

Last but not least, let's evaluate the kNNregression model.

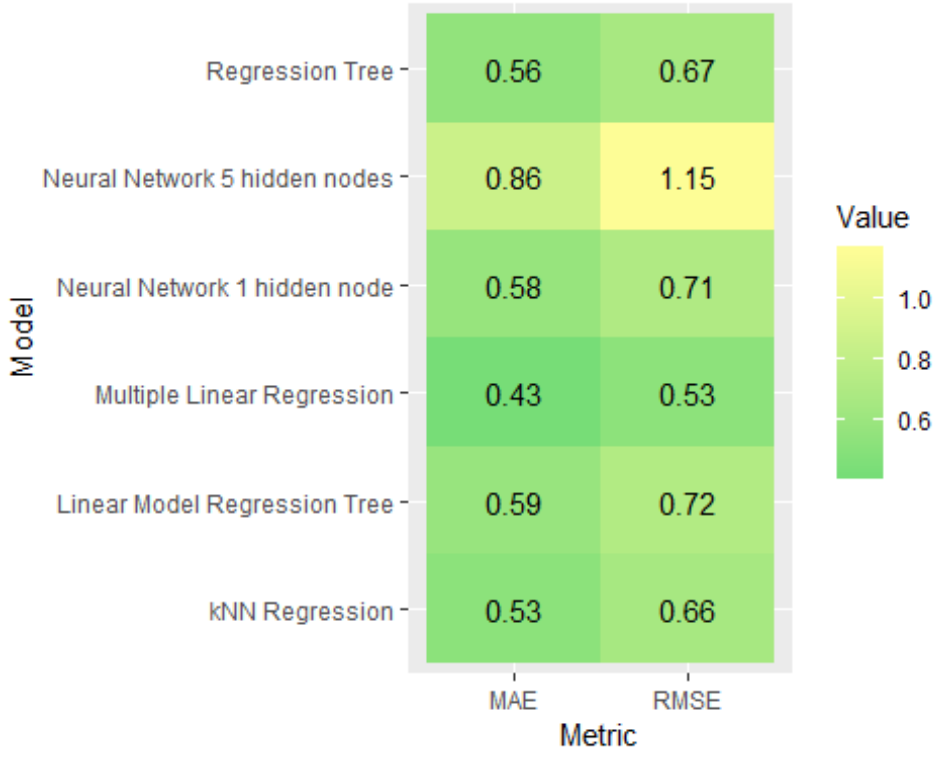
```
##               MAE      RMSE
## Fold01 0.6110966 0.6996027
## Fold02 0.6756104 0.8508154
## Fold03 0.7074924 0.7908895
## Fold04 0.4731212 0.6238676
## Fold05 0.4900182 0.6859397
## Fold06 0.6326713 0.8481861
## Fold07 0.5118606 0.6105268
## Fold08 0.3772670 0.4822302
## Fold09 0.4736864 0.6084162
## Fold10 0.3401023 0.4436521

## [1] 0.5292926

## [1] 0.6644126
```

Much to my surprise, kNN actually performs very well despite the high dimensionality. I can make sense of this logically because there are natural neareast neighbors when comparing nations (aka peer nations). Also the number of data points is quite low. The average MAE for knnregression is 0.5292926 and the average RMSE is 0.6644126.

In order to visualize this easiy, I will make a dataframe of all of the MAE and RMSE values.

```
##                                    MAE      RMSE
## Multiple Linear Regression    0.4291805 0.5324223
## Regression Tree               0.5551525 0.6727711
```

```
## Linear Model Regression Tree  0.5860661 0.7223334
## Neural Network 1 hidden node  0.5835360 0.7115564
## Neural Network 5 hidden nodes 0.8630703 1.1522423
## kNN Regression                0.5292926 0.6644126
```



As we can see, the first Multiple Linear Regression and the kNN regression yield the lowest error.

Let's create a stacked ensemble model now that averages results from the different models. Ensemble learning takes weaker learners and strengthens them, by getting more "perspectives".

```
head(pred.mlreg)#no transform applied
```

```
##        1        8       12       17       18       30
## 6.780507 6.771551 6.954193 6.646931 6.914875 6.164042
```

```
head(pred.regtree)#no transform applied
```

```
##        1        8       12       17       18       30
## 7.187692 6.544824 7.187692 7.187692 7.187692 5.687429
```

```
head(pred.regtree2)#no transform applied
```

```
## [1] 6.322541 6.322541 6.322541 6.322541 6.322541 5.598570
```

```
head(pred.nn)#min/max normalization transform applied
```

```
##         [,1]
## 1  0.8917306
## 8  0.8626436
## 12 0.8631498
## 17 0.8551880
## 18 0.9230205
## 30 0.7307645
```

```
head(pred.nn2)#min/max normalization transform applied
```

```
##         [,1]
## 1  0.8704636
## 8  0.7560773
## 12 0.9273900
## 17 0.9652176
## 18 0.9056535
## 30 0.7940645
```

```
head(pred.knnreg)#min/max normalization transform applied
```

```
## [1] 0.8387992 0.8874899 0.8710629 0.8710629 0.8710629 0.7299089
```

Remember to unnormalize the predictions!

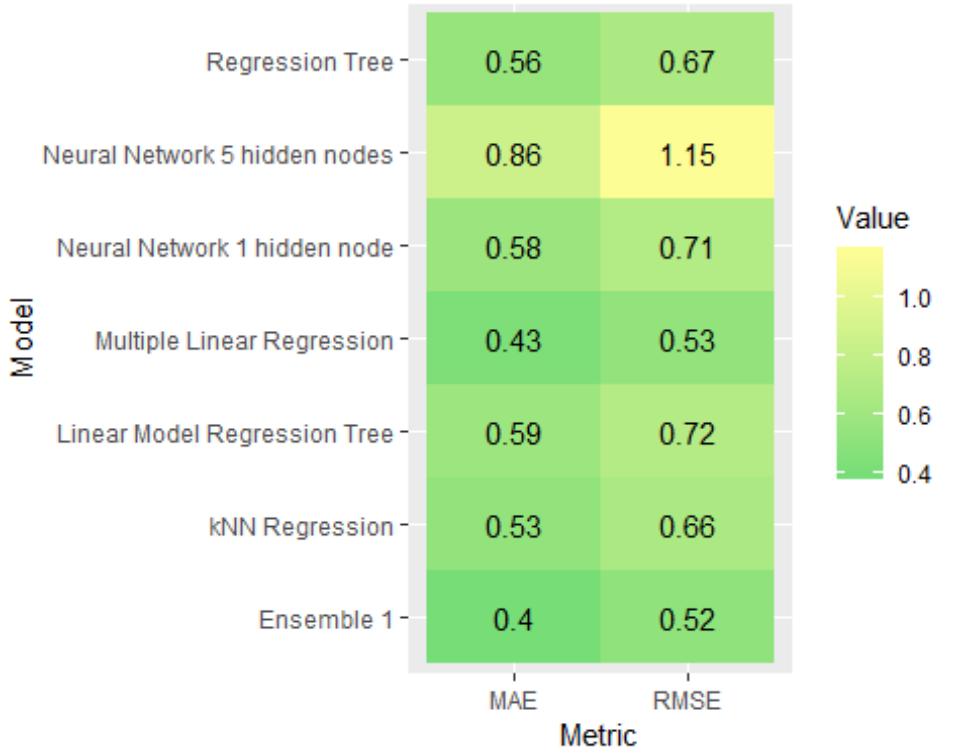
```
##    pred.mlreg pred.regtree pred.regtree2  pred.nn pred.nn2 pred.knnreg
## 1    6.780507     7.187692      6.322541 7.025687 6.927412    6.781091
## 8    6.771551     6.544824      6.322541 6.891276 6.398833    7.006091
## 12   6.954193     7.187692      6.322541 6.893615 7.190469    6.930182
## 17   6.646931     7.187692      6.322541 6.856824 7.365270    6.930182
## 18   6.914875     7.187692      6.322541 7.170278 7.090025    6.930182
## 30   6.164042     5.687429      5.598570 6.281863 6.574372    6.277909
##    average.pred
## 1      6.837488
## 8      6.655853
## 12     6.913116
## 17     6.884907
## 18     6.935932
## 30     6.097364
```

Now let's see the MAE and RMSE of this simple averaging ensemble model.

```
## [1] 0.3966315
```

```
## [1] 0.5220735
```

This is the lowest yet! Now let's add this to the heatmap.



As you can see, the Ensemble now has the best performance with a MAE of 0.3966315 and a RMSE of 0.5220735.

Now, finally let's stack another model on top of other models and use their averaged predictions as inputs to train the model!

I am choosing to stack a regression tree on top of the simple ensemble, mostly due to its short training phase. In order to do this, I am going to add the predictions as a new predictor to the regression tree.

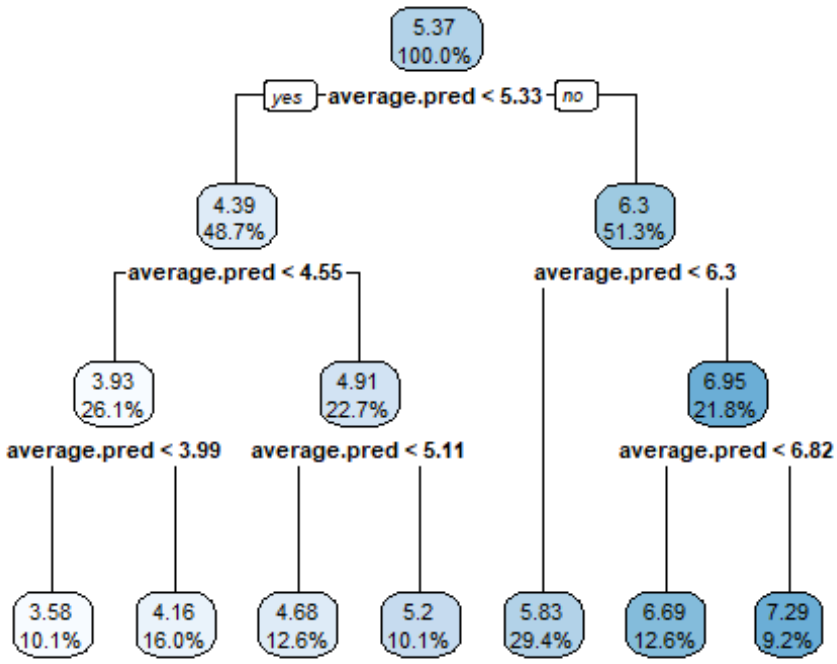
```
## n= 119
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 119 161.7185000 5.369832
##    2) average.pred< 5.332955 58  24.6331200 4.386586
##      4) average.pred< 4.5469 31   6.4524390 3.933097
##        8) average.pred< 3.990249 12   0.9347943 3.575750 *
##        9) average.pred>=3.990249 19   3.0174770 4.158789 *
##      5) average.pred>=4.5469 27   4.4857390 4.907259
##       10) average.pred< 5.11109 15   1.8444750 4.677067 *
##       11) average.pred>=5.11109 12   0.8528980 5.195000 *
##    3) average.pred>=5.332955 61  27.6974500 6.304721
##      6) average.pred< 6.295554 35   5.2728730 5.826486 *
##      7) average.pred>=6.295554 26   3.6440230 6.948500
```

```
##       14) average.pred< 6.815708 15   0.9000609 6.694933 *
##       15) average.pred>=6.815708 11   0.4643742 7.294273 *
```

This output shows the logic behind the tree. As you can see, the previous predictions are weighing heavily on the tree's new predictions.

Now let's visualize the tree.

```
#Create visual
rpart.plot(ensemble2, digits = 3, tweak =1)
```

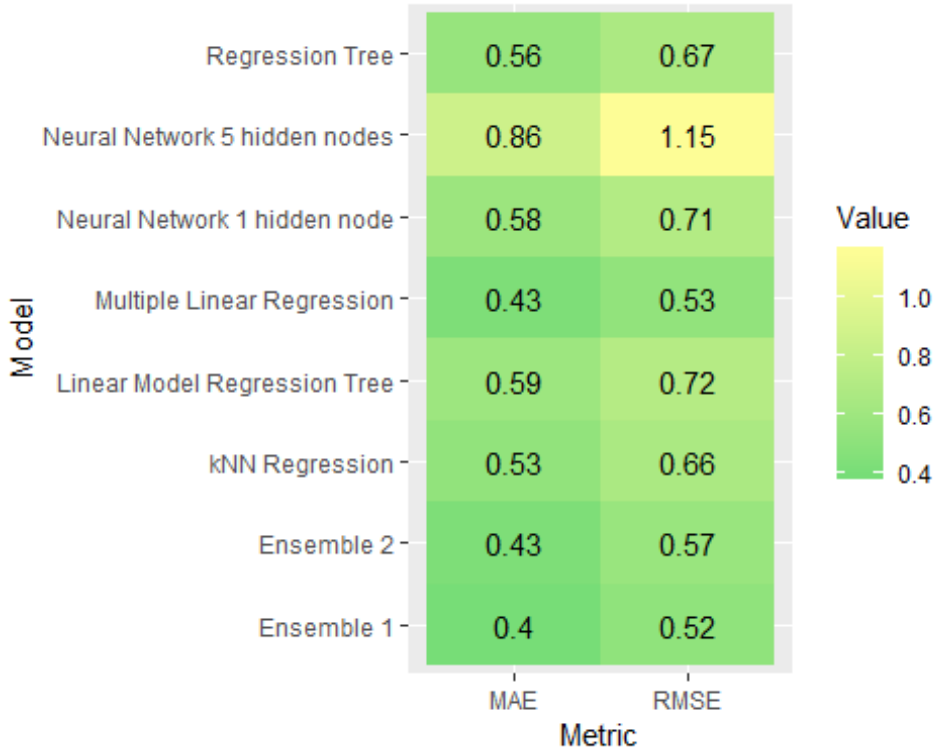


Now let's evaluate this final ensemble by finding the MAE and RMSE.

```
## [1] 0.4278825
```

```
## [1] 0.5680639
```

The simple averaging ensemble had a lower MAE and RMSE. The stacked model had a MAE of 0.4278825 and a RMSE of 0.5680639. Now let's add this to the heatmap.



The ensemble models did improve performance over any singular model, which is a good sign! The overall errors are not that high. The predicted happiness scores tend to be pretty darn close to the actuals. Stacking focuses on removing bias by taking into account under and over predicters, hence why the averaging makes for better predictions.

Next Steps

If time allowed, for further analysis, I would like to:

-look at classification of a nation into a region of the world

-explore binary classification of a “happy” or “unhappy” threshold

- do a clustering analysis with kmeans to see if there are clusters in the data
- do forecasting on historical human development statistics

References

https://cran.r-project.org/web/packages/bestNormalize/vignettes/bestNormalize.html

http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/

https://www.r-bloggers.com/how-to-make-a-simple-heatmap-in-ggplot2/

https://cran.r-project.org/web/packages/bestNormalize/vignettes/bestNormalize.html