DS6050.FinalProject.Bender

Alex Bender

March 24, 2019

Data Description: Countries of the World - This data comes from the US CIA. It is general characterstsics 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.

```r
#load country characteristics data
cc_data <- read.csv("countries of the world.csv", dec=",")

#load 2016 world happiness data
wh2016 <- read.csv("happiness2016.csv")

#load 2014 HDI data
hdi2014 <- read.csv("human_development.csv", stringsAsFactors = FALSE)

#gather data understanding
str(cc_data)

## 'data.frame':    227 obs. of  20 variables:
##  $ Country                          : Factor w/ 227 levels "Afghanistan
",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Region                           : Factor w/ 11 levels "ASIA (EX. NEAR
```

```
EAST)          ",..: 1 4 7 9 11 10 5 5 5 3 ...
## $ Population                      : int  31056997 3581655 32930091
57794 71201 12127071 13477 69108 39921833 2976372 ...
## $ Area..sq..mi..                  : int  647500 28748 2381740 199 468
1246700 102 443 2766890 29800 ...
## $ Pop..Density..per.sq..mi..      : num  48 124.6 13.8 290.4 152.1 ...
## $ Coastline..coast.area.ratio.    : num  0 1.26 0.04 58.29 0 ...
## $ Net.migration                   : num  23.06 -4.93 -0.39 -20.71 6.6
...
## $ Infant.mortality..per.1000.births.: num  163.07 21.52 31 9.27 4.05 ...
## $ GDP....per.capita.              : int  700 4500 6000 8000 19000 1900
8600 11000 11200 3500 ...
## $ Literacy....                    : num  36 86.5 70 97 100 42 95 89
97.1 98.6 ...
## $ Phones..per.1000.               : num  3.2 71.2 78.1 259.5 497.2 ...
## $ Arable....                      : num  12.13 21.09 3.22 10 2.22 ...
## $ Crops....                       : num  0.22 4.42 0.25 15 0 0.24 0
4.55 0.48 2.3 ...
## $ Other....                       : num  87.7 74.5 96.5 75 97.8 ...
## $ Climate                         : num  1 3 1 2 3 NA 2 2 3 4 ...
## $ Birthrate                       : num  46.6 15.11 17.14 22.46 8.71
...
## $ Deathrate                       : num  20.34 5.22 4.61 3.27 6.25 ...
## $ Agriculture                     : num  0.38 0.232 0.101 NA NA 0.096
0.04 0.038 0.095 0.239 ...
## $ Industry                        : num  0.24 0.188 0.6 NA NA 0.658
0.18 0.22 0.358 0.343 ...
## $ Service                         : num  0.38 0.579 0.298 NA NA 0.246
0.78 0.743 0.547 0.418 ...
```

```
summary(cc_data)
```

```
##         Country                                Region
## Afghanistan   : 1   SUB-SAHARAN AFRICA             :51
## Albania       : 1   LATIN AMER. & CARIB            :45
## Algeria       : 1   ASIA (EX. NEAR EAST)           :28
## American Samoa: 1   WESTERN EUROPE                 :28
## Andorra       : 1   OCEANIA                        :21
## Angola        : 1   NEAR EAST                      :16
## (Other)       :221  (Other)                        :38
##   Population        Area..sq..mi..    Pop..Density..per.sq..mi..
## Min.   :7.026e+03   Min.   :      2   Min.   :    0.00
## 1st Qu.:4.376e+05   1st Qu.:   4648   1st Qu.:   29.15
## Median :4.787e+06   Median :  86600   Median :   78.80
## Mean   :2.874e+07   Mean   : 598227   Mean   :  379.05
## 3rd Qu.:1.750e+07   3rd Qu.: 441811   3rd Qu.:  190.15
## Max.   :1.314e+09   Max.   :17075200   Max.   :16271.50
##
## Coastline..coast.area.ratio. Net.migration
## Min.   : 0.00                Min.   :-20.99000
```

```
##   1st Qu.:  0.10                    1st Qu.: -0.92750
##   Median :  0.73                    Median :  0.00000
##   Mean   : 21.17                    Mean   :  0.03812
##   3rd Qu.: 10.35                    3rd Qu.:  0.99750
##   Max.   :870.66                    Max.   : 23.06000
##                                     NA's   :3
##   Infant.mortality..per.1000.births. GDP....per.capita.  Literacy....
##   Min.   :  2.29                       Min.   :  500     Min.   : 17.60
##   1st Qu.:  8.15                       1st Qu.: 1900     1st Qu.: 70.60
##   Median : 21.00                       Median : 5550     Median : 92.50
##   Mean   : 35.51                       Mean   : 9690     Mean   : 82.84
##   3rd Qu.: 55.70                       3rd Qu.:15700     3rd Qu.: 98.00
##   Max.   :191.19                       Max.   :55100     Max.   :100.00
##   NA's   :3                            NA's   :1         NA's   :18
##   Phones..per.1000.   Arable....      Crops....         Other....
##   Min.   :   0.2    Min.   : 0.00   Min.   : 0.000   Min.   : 33.33
##   1st Qu.:  37.8    1st Qu.: 3.22   1st Qu.: 0.190   1st Qu.: 71.65
##   Median : 176.2    Median :10.42   Median : 1.030   Median : 85.70
##   Mean   : 236.1    Mean   :13.80   Mean   : 4.564   Mean   : 81.64
##   3rd Qu.: 389.6    3rd Qu.:20.00   3rd Qu.: 4.440   3rd Qu.: 95.44
##   Max.   :1035.6    Max.   :62.11   Max.   :50.680   Max.   :100.00
##   NA's   :4         NA's   :2       NA's   :2        NA's   :2
##      Climate          Birthrate        Deathrate        Agriculture
##   Min.   :1.000    Min.   : 7.29    Min.   : 2.290   Min.   :0.00000
##   1st Qu.:2.000    1st Qu.:12.67    1st Qu.: 5.910   1st Qu.:0.03775
##   Median :2.000    Median :18.79    Median : 7.840   Median :0.09900
##   Mean   :2.139    Mean   :22.11    Mean   : 9.241   Mean   :0.15084
##   3rd Qu.:3.000    3rd Qu.:29.82    3rd Qu.:10.605   3rd Qu.:0.22100
##   Max.   :4.000    Max.   :50.73    Max.   :29.740   Max.   :0.76900
##   NA's   :22       NA's   :3        NA's   :4        NA's   :15
##      Industry         Service
##   Min.   :0.0200   Min.   :0.0620
##   1st Qu.:0.1930   1st Qu.:0.4293
##   Median :0.2720   Median :0.5710
##   Mean   :0.2827   Mean   :0.5653
##   3rd Qu.:0.3410   3rd Qu.:0.6785
##   Max.   :0.9060   Max.   :0.9540
##   NA's   :16       NA's   :15
```

```
head(cc_data)
```

```
##          Country                      Region Population
## 1    Afghanistan      ASIA (EX. NEAR EAST)     31056997
## 2        Albania   EASTERN EUROPE               3581655
## 3        Algeria   NORTHERN AFRICA             32930091
## 4 American Samoa   OCEANIA                        57794
## 5        Andorra   WESTERN EUROPE                 71201
## 6         Angola   SUB-SAHARAN AFRICA          12127071
##   Area..sq..mi.. Pop..Density..per.sq..mi.. Coastline..coast.area.ratio.
## 1         647500                       48.0                         0.00
```

```
## 2          28748                          124.6                          1.26
## 3        2381740                           13.8                          0.04
## 4            199                          290.4                         58.29
## 5            468                          152.1                          0.00
## 6        1246700                            9.7                          0.13
##   Net.migration Infant.mortality..per.1000.births. GDP....per.capita.
## 1         23.06                             163.07                700
## 2         -4.93                              21.52               4500
## 3         -0.39                              31.00               6000
## 4        -20.71                               9.27               8000
## 5          6.60                               4.05              19000
## 6          0.00                             191.19               1900
##   Literacy.... Phones..per.1000. Arable.... Crops.... Other.... Climate
## 1         36.0               3.2      12.13      0.22     87.65       1
## 2         86.5              71.2      21.09      4.42     74.49       3
## 3         70.0              78.1       3.22      0.25     96.53       1
## 4         97.0             259.5      10.00     15.00     75.00       2
## 5        100.0             497.2       2.22      0.00     97.78       3
## 6         42.0               7.8       2.41      0.24     97.35      NA
##   Birthrate Deathrate Agriculture Industry Service
## 1     46.60     20.34       0.380    0.240   0.380
## 2     15.11      5.22       0.232    0.188   0.579
## 3     17.14      4.61       0.101    0.600   0.298
## 4     22.46      3.27          NA       NA      NA
## 5      8.71      6.25          NA       NA      NA
## 6     45.11     24.20       0.096    0.658   0.246
```

#gather data understanding
str(hdi2014)

```
## 'data.frame':    195 obs. of  8 variables:
##  $ HDI.Rank                        : int  1 2 3 4 5 6 6 8 9 9 ...
##  $ Country                         : chr  "Norway" "Australia"
"Switzerland" "Denmark" ...
##  $ Human.Development.Index..HDI.   : num  0.944 0.935 0.93 0.923
0.922 0.916 0.916 0.915 0.913 0.913 ...
##  $ Life.Expectancy.at.Birth        : num  81.6 82.4 83 80.2 81.6
80.9 80.9 79.1 82 81.8 ...
##  $ Expected.Years.of.Education     : num  17.5 20.2 15.8 18.7 17.9
16.5 18.6 16.5 15.9 19.2 ...
##  $ Mean.Years.of.Education         : num  12.6 13 12.8 12.7 11.9
13.1 12.2 12.9 13 12.5 ...
##  $ Gross.National.Income..GNI..per.Capita: chr  "64,992" "42,261" "56,431"
"44,025" ...
##  $ GNI.per.Capita.Rank.Minus.HDI.Rank    : int  5 17 6 11 9 11 16 3 11 23
...
```

summary(hdi2014)

```
##     HDI.Rank        Country         Human.Development.Index..HDI.
##  Min.   : 1.00   Length:195        Min.   :0.3480
```

```
##   1st Qu.: 47.75   Class :character   1st Qu.:0.5770
##   Median : 94.00   Mode  :character   Median :0.7210
##   Mean   : 94.31                      Mean   :0.6918
##   3rd Qu.:141.25                      3rd Qu.:0.8000
##   Max.   :188.00                      Max.   :0.9440
##   NA's   :7
##   Life.Expectancy.at.Birth Expected.Years.of.Education
##   Min.   :49.00            Min.   : 4.10
##   1st Qu.:65.75            1st Qu.:11.10
##   Median :73.10            Median :13.10
##   Mean   :71.07            Mean   :12.86
##   3rd Qu.:76.80            3rd Qu.:14.90
##   Max.   :84.00            Max.   :20.20
##
##   Mean.Years.of.Education Gross.National.Income..GNI..per.Capita
##   Min.   : 1.400          Length:195
##   1st Qu.: 5.550          Class :character
##   Median : 8.400          Mode  :character
##   Mean   : 8.079
##   3rd Qu.:10.600
##   Max.   :13.100
##
##   GNI.per.Capita.Rank.Minus.HDI.Rank
##   Min.   :-84.0000
##   1st Qu.: -9.0000
##   Median :  1.5000
##   Mean   :  0.1862
##   3rd Qu.: 11.0000
##   Max.   : 47.0000
##   NA's   :7
```

head(hdi2014)

```
##   HDI.Rank     Country Human.Development.Index..HDI.
## 1        1      Norway                         0.944
## 2        2   Australia                         0.935
## 3        3 Switzerland                         0.930
## 4        4     Denmark                         0.923
## 5        5 Netherlands                         0.922
## 6        6     Germany                         0.916
##   Life.Expectancy.at.Birth Expected.Years.of.Education
## 1                     81.6                        17.5
## 2                     82.4                        20.2
## 3                     83.0                        15.8
## 4                     80.2                        18.7
## 5                     81.6                        17.9
## 6                     80.9                        16.5
##   Mean.Years.of.Education Gross.National.Income..GNI..per.Capita
## 1                    12.6                                 64,992
## 2                    13.0                                 42,261
```

```
## 3                        12.8                          56,431
## 4                        12.7                          44,025
## 5                        11.9                          45,435
## 6                        13.1                          43,919
##   GNI.per.Capita.Rank.Minus.HDI.Rank
## 1                                  5
## 2                                 17
## 3                                  6
## 4                                 11
## 5                                  9
## 6                                 11
```

#gather data understanding
str(wh2016)

```
## 'data.frame':    157 obs. of  13 variables:
##  $ Country                   : Factor w/ 157 levels "Afghanistan",..:
## 38 135 58 104 45 26 98 99 7 134 ...
##  $ Region                    : Factor w/ 10 levels "Australia and New
## Zealand",..: 10 10 10 10 10 6 10 1 1 10 ...
##  $ Happiness.Rank            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Happiness.Score           : num  7.53 7.51 7.5 7.5 7.41 ...
##  $ Lower.Confidence.Interval : num  7.46 7.43 7.33 7.42 7.35 ...
##  $ Upper.Confidence.Interval : num  7.59 7.59 7.67 7.58 7.47 ...
##  $ Economy..GDP.per.Capita.  : num  1.44 1.53 1.43 1.58 1.41 ...
##  $ Family                    : num  1.16 1.15 1.18 1.13 1.13 ...
##  $ Health..Life.Expectancy.  : num  0.795 0.863 0.867 0.796 0.811 ...
##  $ Freedom                   : num  0.579 0.586 0.566 0.596 0.571 ...
##  $ Trust..Government.Corruption.: num  0.445 0.412 0.15 0.358 0.41 ...
##  $ Generosity                : num  0.362 0.281 0.477 0.379 0.255 ...
##  $ Dystopia.Residual         : num  2.74 2.69 2.83 2.66 2.83 ...
```

summary(wh2016)

```
##          Country                           Region    Happiness.Rank
##   Afghanistan:  1   Sub-Saharan Africa          :38   Min.   :  1.00
##   Albania    :  1   Central and Eastern Europe  :29   1st Qu.: 40.00
##   Algeria    :  1   Latin America and Caribbean :24   Median : 79.00
##   Angola     :  1   Western Europe              :21   Mean   : 78.98
##   Argentina  :  1   Middle East and Northern Africa:19   3rd Qu.:118.00
##   Armenia    :  1   Southeastern Asia           : 9   Max.   :157.00
##   (Other)    :151   (Other)                     :17
##   Happiness.Score Lower.Confidence.Interval Upper.Confidence.Interval
##   Min.   :2.905   Min.   :2.732             Min.   :3.078
##   1st Qu.:4.404   1st Qu.:4.327             1st Qu.:4.465
##   Median :5.314   Median :5.237             Median :5.419
##   Mean   :5.382   Mean   :5.282             Mean   :5.482
##   3rd Qu.:6.269   3rd Qu.:6.154             3rd Qu.:6.434
##   Max.   :7.526   Max.   :7.460             Max.   :7.669
##
##   Economy..GDP.per.Capita.      Family        Health..Life.Expectancy.
```

```
##   Min.   :0.0000              Min.   :0.0000    Min.   :0.0000
##   1st Qu.:0.6702              1st Qu.:0.6418    1st Qu.:0.3829
##   Median :1.0278              Median :0.8414    Median :0.5966
##   Mean   :0.9539              Mean   :0.7936    Mean   :0.5576
##   3rd Qu.:1.2796              3rd Qu.:1.0215    3rd Qu.:0.7299
##   Max.   :1.8243              Max.   :1.1833    Max.   :0.9528
##
##      Freedom       Trust..Government.Corruption.   Generosity
##   Min.   :0.0000   Min.   :0.00000                Min.   :0.0000
##   1st Qu.:0.2575   1st Qu.:0.06126                1st Qu.:0.1546
##   Median :0.3975   Median :0.10547                Median :0.2225
##   Mean   :0.3710   Mean   :0.13762                Mean   :0.2426
##   3rd Qu.:0.4845   3rd Qu.:0.17554                3rd Qu.:0.3119
##   Max.   :0.6085   Max.   :0.50521                Max.   :0.8197
##
##   Dystopia.Residual
##   Min.   :0.8179
##   1st Qu.:2.0317
##   Median :2.2907
##   Mean   :2.3258
##   3rd Qu.:2.6646
##   Max.   :3.8377
##
```

head(wh2016)

```
##         Country          Region Happiness.Rank Happiness.Score
## 1       Denmark Western Europe             1          7.526
## 2   Switzerland Western Europe             2          7.509
## 3       Iceland Western Europe             3          7.501
## 4        Norway Western Europe             4          7.498
## 5       Finland Western Europe             5          7.413
## 6        Canada  North America             6          7.404
##    Lower.Confidence.Interval Upper.Confidence.Interval
## 1                      7.460                     7.592
## 2                      7.428                     7.590
## 3                      7.333                     7.669
## 4                      7.421                     7.575
## 5                      7.351                     7.475
## 6                      7.335                     7.473
##    Economy..GDP.per.Capita.  Family Health..Life.Expectancy. Freedom
## 1                   1.44178 1.16374                   0.79504 0.57941
## 2                   1.52733 1.14524                   0.86303 0.58557
## 3                   1.42666 1.18326                   0.86733 0.56624
## 4                   1.57744 1.12690                   0.79579 0.59609
## 5                   1.40598 1.13464                   0.81091 0.57104
## 6                   1.44015 1.09610                   0.82760 0.57370
##    Trust..Government.Corruption. Generosity Dystopia.Residual
## 1                        0.44453    0.36171           2.73939
## 2                        0.41203    0.28083           2.69463
```

```
## 3                             0.14975    0.47678            2.83137
## 4                             0.35776    0.37895            2.66465
## 5                             0.41004    0.25492            2.82596
## 6                             0.31329    0.44834            2.70485
```

As you can see all of 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.

My plan of attack: - 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.

If I have time: - Classification for region of the world based on the features - Derive an attribute that is a binary indicator of happy or not in order to do binary classification using SVM and/or Logistic regression.

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.

```
library(tidyverse)

## -- Attaching packages -------------------------------------------------
----------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0        v purrr   0.3.2
## v tibble  2.1.1        v dplyr   0.8.0.1
## v tidyr   0.8.3        v stringr 1.4.0
## v readr   1.3.1        v forcats 0.4.0

## -- Conflicts ----------------------------------------------------------
----------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

To reduce chances for errors down the line, let's trim any additional whitespaces from the data.

```r
#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()
```

Let's only select the two target/response variables and the country from the world happiness data.

```r
#select only the needed columns
wh2016 <- select(wh2016, Country, Region, Happiness.Score)
```

We know from this dataset that we will be looking at 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.

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

#Standardize column using default country names
hdi2014$Country <- standardize.countrynames(hdi2014$Country,suggest="auto",
verbose = 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",
verbose = 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
distinct(rbind(non_matches,non_matches2))

##                    Country
## 1            Puerto Rico
## 2                 Taiwan
## 3            North Cyprus
## 4                Somalia
## 5                 Kosovo
## 6       Somaliland Region
## 7             Montenegro
## 8   Palestinian Territory
## 9                Myanmar
## 10            South Sudan
```

The 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. This will leave us with 147 countries to analyze, which is still enough to draw meaningful insights.

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.

```
#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!

Let's make sure the join worked correctly by spot checking a country

```r
cc_data[which(cc_data$Country=="France"),"Population"]
```

```
## [1] 60876136
```

```r
hdi2014[which(hdi2014$Country=="France"),"Mean.Years.of.Education"]
```

```
## [1] 11.1
```

```r
wh2016[which(wh2016$Country=="France"),"Happiness.Score"]
```

```
## [1] 6.478
```

```r
world_df[which(world_df$Country=="France"),c("Population",
"Mean.Years.of.Education", "Happiness.Score")]
```

```
##    Population Mean.Years.of.Education Happiness.Score
## 31   60876136                    11.1           6.478
```

Perfect, they match!

Now let's explore the data!

```r
str(world_df)
```

```
## 'data.frame':    147 obs. of  29 variables:
##  $ Country                        : chr  "Denmark" "Switzerland"
"Iceland" "Norway" ...
##  $ Region.x                       : Factor w/ 10 levels "Australia
and New Zealand",..: 10 10 10 10 10 6 10 1 1 10 ...
##  $ Happiness.Score                : num  7.53 7.51 7.5 7.5 7.41 ...
##  $ HDI.Rank                       : int  4 3 16 1 24 9 5 9 2 14 ...
##  $ Human.Development.Index..HDI.  : num  0.923 0.93 0.899 0.944
0.883 0.913 0.922 0.913 0.935 0.907 ...
##  $ Life.Expectancy.at.Birth       : num  80.2 83 82.6 81.6 80.8 82
81.6 81.8 82.4 82.2 ...
##  $ Expected.Years.of.Education    : num  18.7 15.8 19 17.5 17.1
15.9 17.9 19.2 20.2 15.8 ...
##  $ Mean.Years.of.Education        : num  12.7 12.8 10.6 12.6 10.3
13 11.9 12.5 13 12.1 ...
##  $ Gross.National.Income..GNI..per.Capita: chr  "44,025" "56,431" "35,182"
"64,992" ...
##  $ GNI.per.Capita.Rank.Minus.HDI.Rank    : int  11 6 12 5 0 11 9 23 17 -1
...
##  $ Region.y                       : Factor w/ 11 levels "ASIA (EX.
NEAR EAST)",..: 11 11 11 11 11 8 11 9 9 11 ...
##  $ Population                     : int  5450661 7523934 299388
4610820 5231372 33098932 16491461 4076140 20264082 9016596 ...
##  $ Area..sq..mi..                 : int  43094 41290 103000 323802
338145 9984670 41526 268680 7686850 449964 ...
##  $ Pop..Density..per.sq..mi..     : num  126.5 182.2 2.9 14.2 15.5
```

```
...
##  $ Coastline..coast.area.ratio.       : num   16.97 0 4.83 7.77 0.37 ...
##  $ Net.migration                      : num   2.48 4.05 2.38 1.74 0.95
5.96 2.91 4.05 3.98 1.67 ...
##  $ Infant.mortality..per.1000.births. : num   4.56 4.39 3.31 3.7 3.57
4.75 5.04 5.85 4.69 2.77 ...
##  $ GDP....per.capita.                 : int   31100 32700 30900 37800
27400 29800 28600 21600 29000 26800 ...
##  $ Literacy....                       : num   100 99 99.9 100 100 97 99
99 100 99 ...
##  $ Phones..per.1000.                  : num   615 681 648 462 405 ...
##  $ Arable....                         : num   54.02 10.42 0.07 2.87 7.19
...
##  $ Crops....                          : num   0.19 0.61 0 0 0.03 0.02
0.97 6.99 0.04 0.01 ...
##  $ Other....                          : num   45.8 89 99.9 97.1 92.8 ...
##  $ Climate                            : num   3 3 3 3 3 NA 3 3 1 3 ...
##  $ Birthrate                          : num   11.13 9.71 13.64 11.46
10.45 ...
##  $ Deathrate                          : num   10.36 8.49 6.72 9.4 9.86
...
##  $ Agriculture                        : num   0.018 0.015 0.086 0.021
0.028 0.022 0.021 0.043 0.038 0.011 ...
##  $ Industry                           : num   0.246 0.34 0.15 0.415
0.295 0.294 0.244 0.273 0.262 0.282 ...
##  $ Service                            : num   0.735 0.645 0.765 0.564
0.676 0.684 0.736 0.684 0.7 0.707 ...
```

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.

```
#get rid of duplicate region column, HDI rank column,
GNI.per.Capita.Rank.Minus.HDI.Rank column
world_df <-select(world_df, -Region.y)
world_df <- select(world_df, -HDI.Rank)
world_df <- select(world_df, -GNI.per.Capita.Rank.Minus.HDI.Rank)
```

Now rename columns for readability.

```
world_df <- rename(world_df, Region = "Region.x", HDI.Score =
"Human.Development.Index..HDI.", Gross.National.Income.per.Capita
="Gross.National.Income..GNI..per.Capita", Area.sq.mi =
```

```
"Area..sq..mi..",Pop.Density.per.sq.mi =
"Pop..Density..per.sq..mi..",Coast.Area.Ratio=
"Coastline..coast.area.ratio.", Infant.Mortality.per.1000.births=
"Infant.mortality..per.1000.births.", GDP.per.capita = "GDP....per.capita.",
Literacy.percent ="Literacy....", Phones.per.1000.people
="Phones..per.1000.", Arable.percent="Arable....", Crops.percent =
"Crops....", Other.Land.Use.percent= "Other....")
```

Now change Gross National Income (GNI) into a numeric using regex to recognize the

```
world_df$Gross.National.Income.per.Capita <- as.numeric(gsub(",", "",
world_df$Gross.National.Income.per.Capita))
```

Now let's explore the data some more.

```
anyNA(world_df)
```

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

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

```
summary(world_df)
```

```
##     Country                                          Region   Happiness.Score
##  Length:147         Sub-Saharan Africa                   :35   Min.   :2.905
##  Class :character   Central and Eastern Europe           :27   1st Qu.:4.383
##  Mode  :character   Latin America and Caribbean          :23   Median :5.314
##                     Western Europe                       :20   Mean   :5.386
##                     Middle East and Northern Africa:18         3rd Qu.:6.296
##                     Southeastern Asia                    : 8   Max.   :7.526
##                     (Other)                              :16
##     HDI.Score      Life.Expectancy.at.Birth Expected.Years.of.Education
##  Min.   :0.3480   Min.   :50.90             Min.   : 5.40
##  1st Qu.:0.5885   1st Qu.:65.90             1st Qu.:11.25
##  Median :0.7330   Median :74.00             Median :13.50
##  Mean   :0.7056   Mean   :71.75             Mean   :13.19
##  3rd Qu.:0.8355   3rd Qu.:77.50             3rd Qu.:15.25
##  Max.   :0.9440   Max.   :84.00             Max.   :20.20
##
##  Mean.Years.of.Education Gross.National.Income.per.Capita
##  Min.   : 1.400          Min.   :    680
##  1st Qu.: 6.000          1st Qu.:   4198
##  Median : 8.500          Median :  12122
##  Mean   : 8.291          Mean   :  18226
##  3rd Qu.:10.900          3rd Qu.:  25486
##  Max.   :13.100          Max.   :123124
##
##    Population           Area.sq.mi        Pop.Density.per.sq.mi
##  Min.   :2.877e+05   Min.   :    316    Min.   :   1.8
##  1st Qu.:4.493e+06   1st Qu.:  64894    1st Qu.:  27.1
##  Median :1.018e+07   Median : 236800    Median :  66.9
##  Mean   :4.317e+07   Mean   : 877074    Mean   : 206.3
```

```
##    3rd Qu.:2.968e+07    3rd Qu.:  700057    3rd Qu.: 127.2
##    Max.   :1.314e+09    Max.   :17075200    Max.   :6482.2
##
##    Coast.Area.Ratio Net.migration      Infant.Mortality.per.1000.births
##    Min.   : 0.000   Min.   :-10.8300   Min.   :  2.290
##    1st Qu.: 0.005   1st Qu.: -0.7750   1st Qu.:  8.685
##    Median : 0.240   Median :  0.0000   Median : 24.600
##    Mean   : 2.665   Mean   :  0.2145   Mean   : 39.395
##    3rd Qu.: 1.365   3rd Qu.:  0.6700   3rd Qu.: 64.605
##    Max.   :67.120   Max.   : 23.0600   Max.   :191.190
##
##    GDP.per.capita  Literacy.percent Phones.per.1000.people Arable.percent
##    Min.   :  500   Min.   : 17.60   Min.   :  0.20         Min.   : 0.070
##    1st Qu.: 1850   1st Qu.: 69.78   1st Qu.: 26.95         1st Qu.: 4.465
##    Median : 5400   Median : 90.80   Median :139.00         Median :12.310
##    Mean   : 9635   Mean   : 81.51   Mean   :201.71         Mean   :16.120
##    3rd Qu.:13050   3rd Qu.: 98.00   3rd Qu.:317.90         3rd Qu.:23.330
##    Max.   :55100   Max.   :100.00   Max.   :898.00         Max.   :62.110
##                    NA's   :3        NA's   :1
##    Crops.percent   Other.Land.Use.percent    Climate       Birthrate
##    Min.   : 0.000  Min.   :33.91         Min.   :1.000  Min.   : 7.29
##    1st Qu.: 0.260  1st Qu.:70.39         1st Qu.:2.000  1st Qu.:11.95
##    Median : 1.080  Median :85.38         Median :2.000  Median :20.41
##    Mean   : 3.045  Mean   :80.84         Mean   :2.172  Mean   :22.64
##    3rd Qu.: 3.165  3rd Qu.:93.85         3rd Qu.:3.000  3rd Qu.:30.86
##    Max.   :23.320  Max.   :99.93         Max.   :4.000  Max.   :50.73
##                                          NA's   :16     NA's   :1
##     Deathrate       Agriculture       Industry        Service
##    Min.   : 2.410  Min.   :0.0000  Min.   :0.0400  Min.   :0.1770
##    1st Qu.: 6.213  1st Qu.:0.0400  1st Qu.:0.2210  1st Qu.:0.4255
##    Median : 8.870  Median :0.1010  Median :0.2940  Median :0.5500
##    Mean   : 9.827  Mean   :0.1534  Mean   :0.3070  Mean   :0.5390
##    3rd Qu.:12.207  3rd Qu.:0.2255  3rd Qu.:0.3575  3rd Qu.:0.6475
##    Max.   :29.500  Max.   :0.7690  Max.   :0.8010  Max.   :0.9060
##    NA's   :1
```

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.

```
#replace Literacy.percent with the median
world_df$Literacy.percent[is.na(world_df$Literacy.percent)] <-
median(world_df$Literacy.percent, na.rm = T)
```

```
#replace Phones.per.1000 with the median
world_df$Phones.per.1000.people[is.na(world_df$Phones.per.1000.people)] <-
median(world_df$Phones.per.1000.people, na.rm = T)

#replace Climate with the median
world_df$Climate[is.na(world_df$Climate)] <- median(world_df$Climate, na.rm =
T)

#replace Birthrate with the median
world_df$Birthrate[is.na(world_df$Birthrate)] <- median(world_df$Birthrate,
na.rm = T)

#replace Deathrate with the median
world_df$Deathrate[is.na(world_df$Deathrate)] <- median(world_df$Deathrate,
na.rm = T)

#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

```r
#detect which countries have large z-scores
n <- ncol(world_df)
for (i in 3:n){
p <- world_df[abs(outlier.z(world_df[,i]))>3,"Country"]
print(p)
}

## character(0)
## character(0)
## character(0)
## character(0)
## character(0)
## [1] "Singapore" "Qatar"      "Kuwait"
## [1] "China" "India"
## [1] "Canada"            "Australia"          "United States"
## [4] "Brazil"            "Russian Federation" "China"
## [1] "Singapore" "Hong Kong"
## [1] "Malta"      "Hong Kong"
## [1] "Singapore"    "Qatar"         "Kuwait"        "Afghanistan"
## [1] "Angola"       "Afghanistan"
## [1] "Luxembourg"
## [1] "Niger"
## [1] "United States"
## [1] "Bangladesh"
## [1] "Malaysia"     "Philippines" "Comoros"
## character(0)
## character(0)
## character(0)
## [1] "Botswana"
## [1] "Liberia"
## [1] "Qatar"
## character(0)
```

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.

```r
library(ggplot2)
#install.packages("GridExtra")
library(gridExtra)

##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

#install.packages("ggrepel")
library(ggrepel)
attach(world_df)

#make boxplot for GNI per capita that labels outliers
g1 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Gross.National.Income.per.Capita))>3,
Country, "")) %>%
  ggplot(., aes(x = "", y = Gross.National.Income.per.Capita)) +
    geom_boxplot(fill = "#d6bea9") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for population that labels outliers
g2 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Population))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Population)) +
    geom_boxplot(fill = "#0066cc") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)


#make boxplot for land area that labels outliers
g3 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Area.sq.mi))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Area.sq.mi)) +
    geom_boxplot(fill = "#1cb2e3") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)
#make boxplot for population density that labels outliers
g4 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Pop.Density.per.sq.mi))>3, Country,
"")) %>%
  ggplot(., aes(x = "", y = Pop.Density.per.sq.mi)) +
    geom_boxplot(fill = "#54acbe") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for coast area ratio that labels outliers
g5 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Coast.Area.Ratio))>3, Country, ""))
%>%
  ggplot(., aes(x = "", y = Coast.Area.Ratio)) +
    geom_boxplot(fill = "#cbe123") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for net migration that labels outliers
g6 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Net.migration))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Net.migration)) +
```

```r
    geom_boxplot(fill = "#0077dd") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for infant mortality that labels outliers
g7 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Infant.Mortality.per.1000.births))>3,
Country, "")) %>%
  ggplot(., aes(x = "", y = Infant.Mortality.per.1000.births)) +
    geom_boxplot(fill = "#ef14d1") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for GDP Per capita that labels outliers
g8 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(GDP.per.capita))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = GDP.per.capita)) +
    geom_boxplot(fill = "#12ab1a") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for literacy that labels outliers
g9 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Literacy.percent))>3, Country, ""))
%>%
  ggplot(., aes(x = "", y = Literacy.percent)) +
    geom_boxplot(fill = "#bcde28") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for phones that labels outliers
g10 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Phones.per.1000.people))>3, Country,
"")) %>%
  ggplot(., aes(x = "", y = Phones.per.1000.people)) +
    geom_boxplot(fill = "#9aad3f") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for arable that labels outliers
g11 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Arable.percent))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Arable.percent)) +
    geom_boxplot(fill = "#1834af") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for crops that labels outliers
g12 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Crops.percent))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Crops.percent)) +
    geom_boxplot(fill = "#fe1568") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for deathrate that labels outliers
```

```
g13 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Deathrate))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Deathrate)) +
    geom_boxplot(fill = "#b15c16") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)


#make boxplot for agriculture that labels outliers
g14 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Agriculture))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Agriculture)) +
    geom_boxplot(fill = "#ed1478") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)

#make boxplot for industry that labels outliers
g15 <- world_df %>%
  mutate(outlier = ifelse(abs(outlier.z(Industry))>3, Country, "")) %>%
  ggplot(., aes(x = "", y = Industry)) +
    geom_boxplot(fill = "#bb0000") +
    geom_text_repel(aes(label = outlier), hjust = -0.2)
detach(world_df)

grid.arrange(g1, g2, g3, g4, nrow = 2)
```
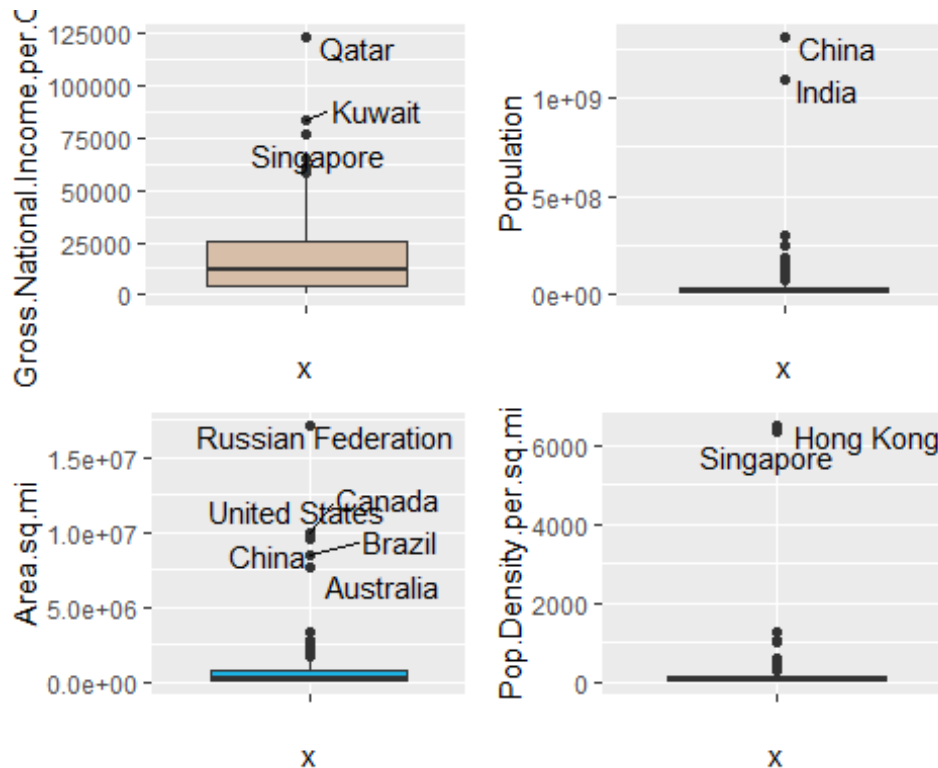


```
grid.arrange(g5, g6, g7, g8, nrow = 2)
```
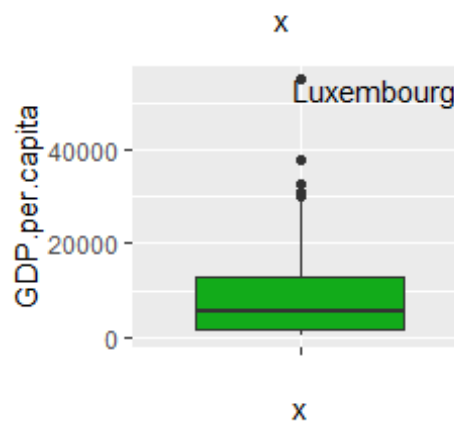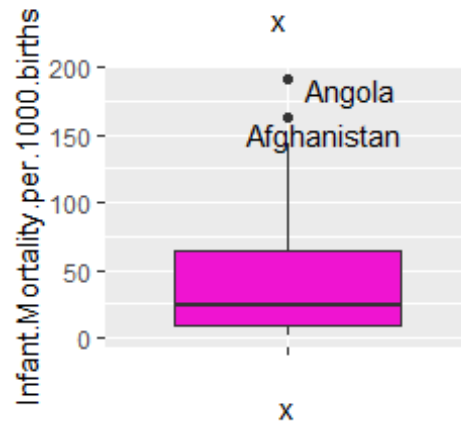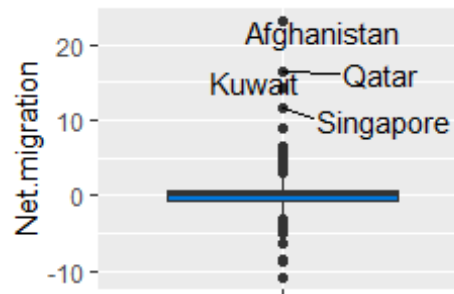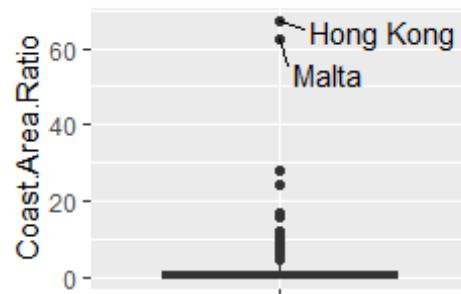
```
grid.arrange(g9, g10, g11, g12, nrow = 2)
```
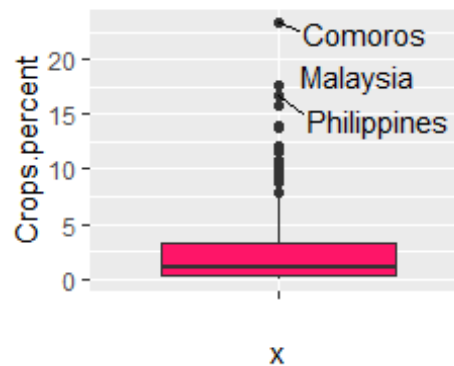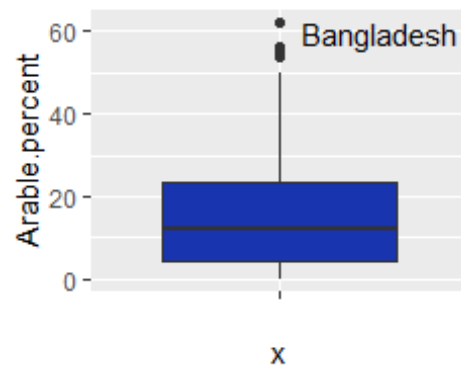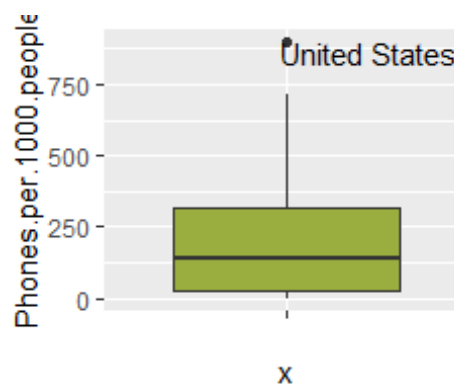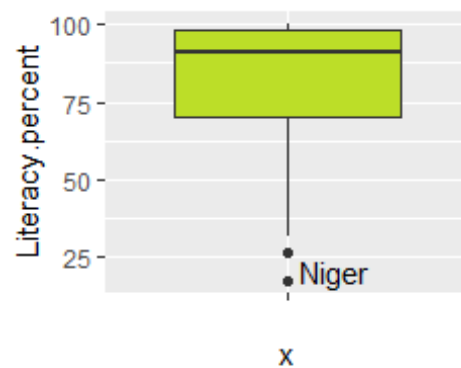


```
grid.arrange(g13, g14, g15, nrow = 2)
```
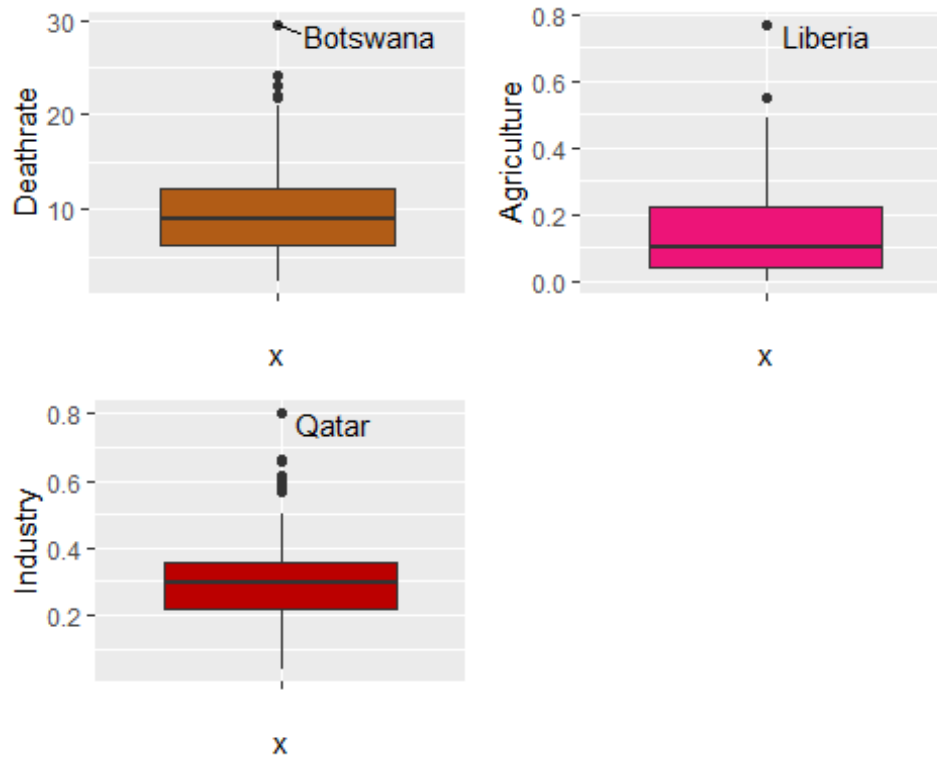
These boxplots (representing all of the features that contain outliers based on z-score) clearly show a good story into the data. All of the findings make sense with what we would expect and accessible layout the distributions of the features as well as label the outlier countries. The removal of outliers 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.

```
r1 <- ggplot(world_df, aes(Region, Happiness.Score))

#show region names
levels(world_df$Region)

##  [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"

#abbreviate region names to make graph more legible
reg_abbr <- c("Australia and New Zealand" = "Aus&NZ", "Central and Eastern
Europe" = "Cen.E Eur","Eastern Asia" = "E. Asia", "Latin America and
Caribbean" = "LatCari", "Middle East and Northern Africa" = "MENA", "North
America" = "N. Amer.", "Southeastern Asia" = "SE. Asia", "Southern Asia" =
"S. Asia", "Sub-Saharan Africa" = "S-S.Africa", "Western Europe" = "W. Eur.")

#plot dotplot to show distribution
r1 + geom_dotplot(binaxis = "y", stackdir = "center", binwidth = 0.1, fill =
"#1995AD") + scale_x_discrete(labels = reg_abbr) + ggtitle("Happiness Scores
by Region")
```


Happiness Scores by Region
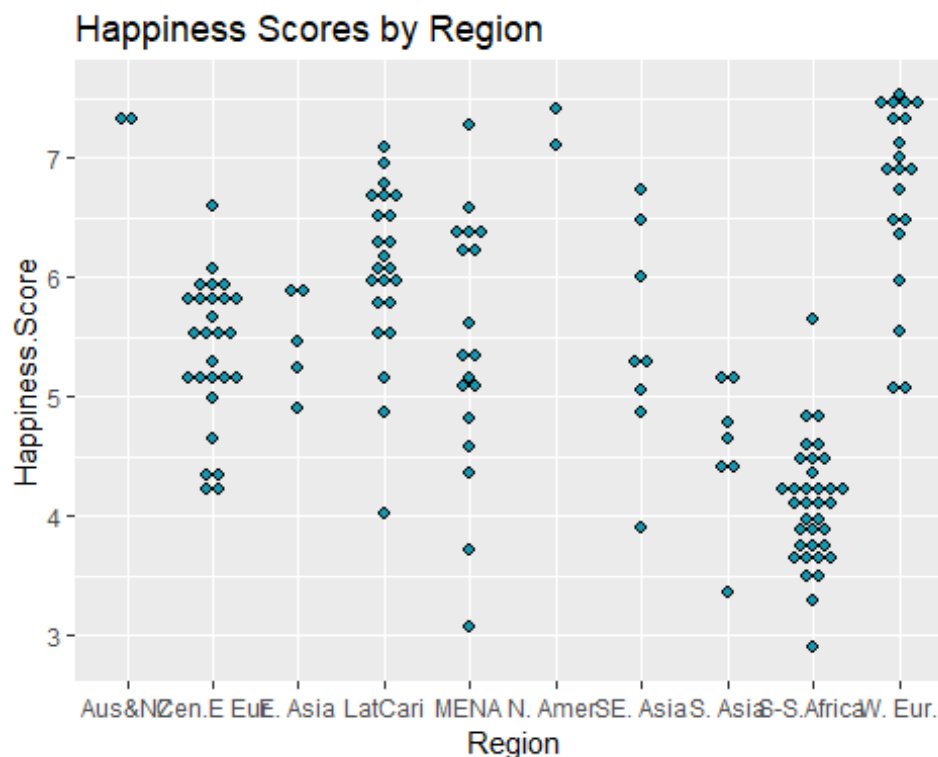
This dot plot nicely shows some distribution of happiness scores by region. You can see some regions such as the Middle East and North Africa have a large varaince. For example, there is a country with a happiness score of just above 3, but there is also a country with a happiness score of over 7. Also, 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.

```
#correlations just to the response variable Sale Price
cormatx.response <- round(cor(world_df[c(4:26)], world_df[3]),2)
cormatx.response

##                                       Happiness.Score
## HDI.Score                                        0.83
## Life.Expectancy.at.Birth                         0.78
## Expected.Years.of.Education                      0.74
## Mean.Years.of.Education                          0.71
## Gross.National.Income.per.Capita                 0.68
## Population                                      -0.02
## Area.sq.mi                                       0.16
## Pop.Density.per.sq.mi                            0.09
## Coast.Area.Ratio                                 0.16
## Net.migration                                    0.17
## Infant.Mortality.per.1000.births                -0.71
## GDP.per.capita                                   0.73
## Literacy.percent                                 0.66
## Phones.per.1000.people                           0.73
## Arable.percent                                  -0.06
## Crops.percent                                   -0.17
## Other.Land.Use.percent                           0.09
## Climate                                          0.29
## Birthrate                                       -0.70
## Deathrate                                       -0.49
## Agriculture                                     -0.69
## Industry                                         0.19
## Service                                          0.53
```

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.

```r
#create full correlation matrix
cormatx <- round(cor(world_df[3:26]), 2)

reorder_cormatx <- function(cormat){
# Use correlation between variables as distance
dd <- as.dist((1-cormatx)/2)
hc <- hclust(dd)
cormat <-cormat[hc$order, hc$order]
}

# Get upper triangle of the correlation matrix
  get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)
  }

#install.packages("reshape2")
library(reshape2)

##
## Attaching package: 'reshape2'

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

# Reorder the correlation matrix
cormatx <- reorder_cormatx(cormatx)
upper_tri <- get_upper_tri(cormatx)
# Melt the correlation matrix
melted_cormatx <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormatx, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 6, hjust = 1.5))+
 coord_fixed()

#format heatmap
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 1.2) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
```
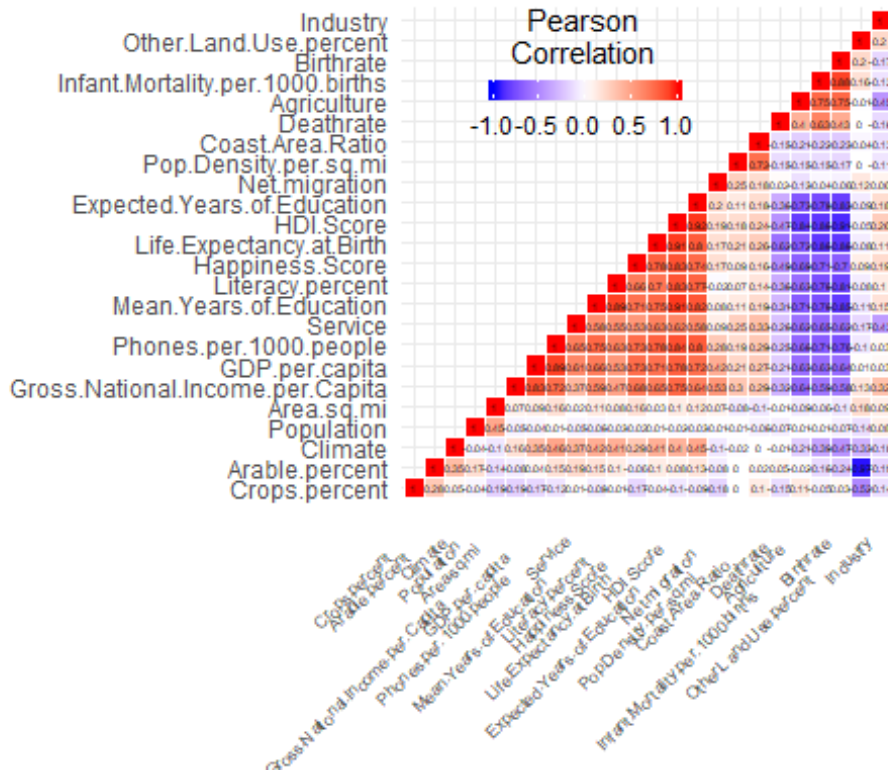
```
legend.position = c(0.6, 0.7),
legend.direction = "horizontal")+
guides(fill = guide_colorbar(barwidth = 5, barheight = .5,
                  title.position = "top", title.hjust = 0.5))
```


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.
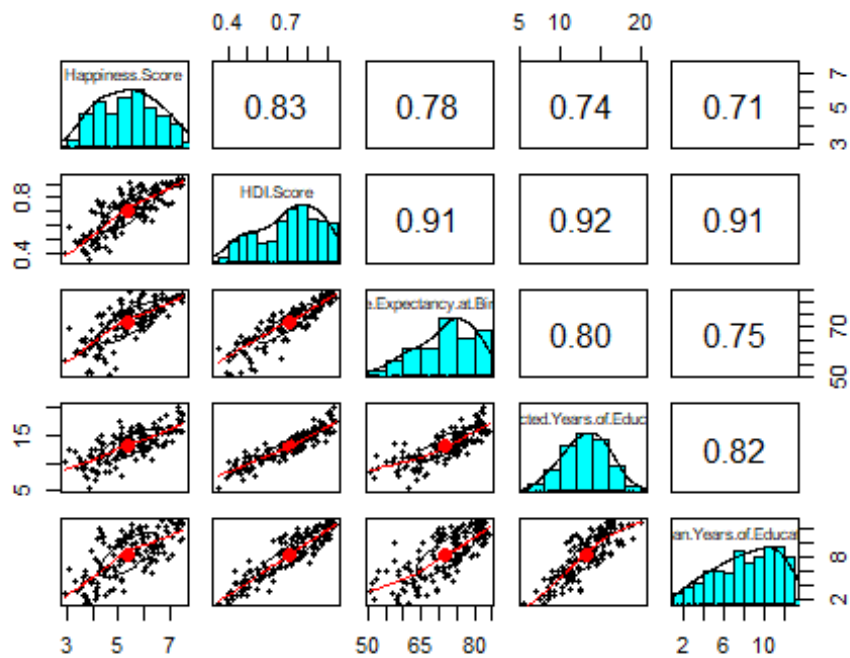
```r
#install.packages("psych")
library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##      %+%, alpha

pairs.panels(world_df[3:7])
```
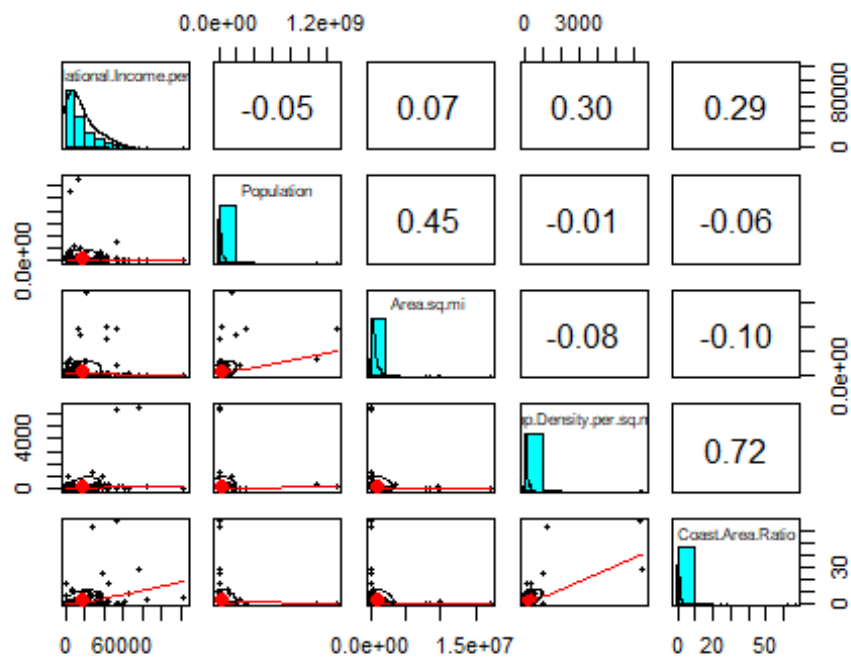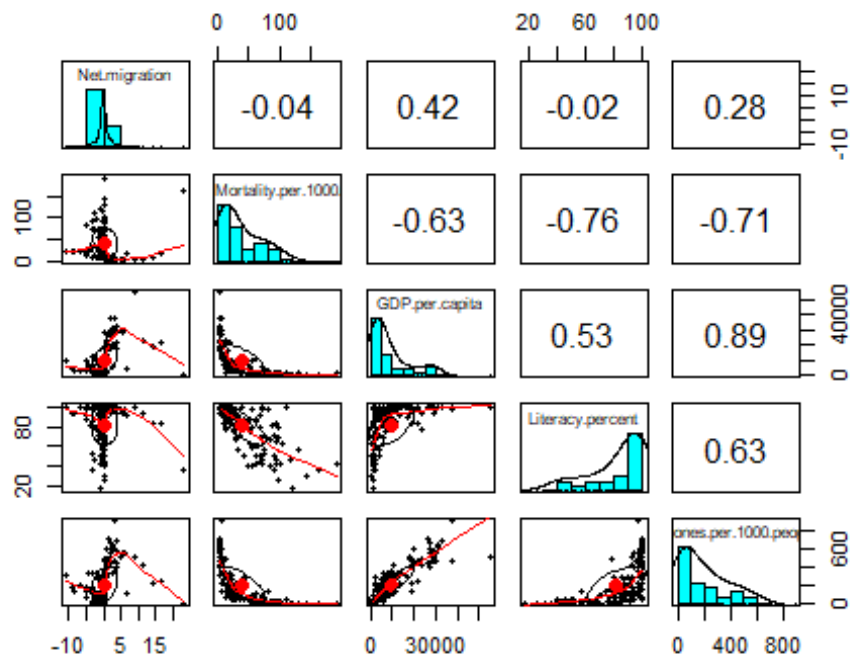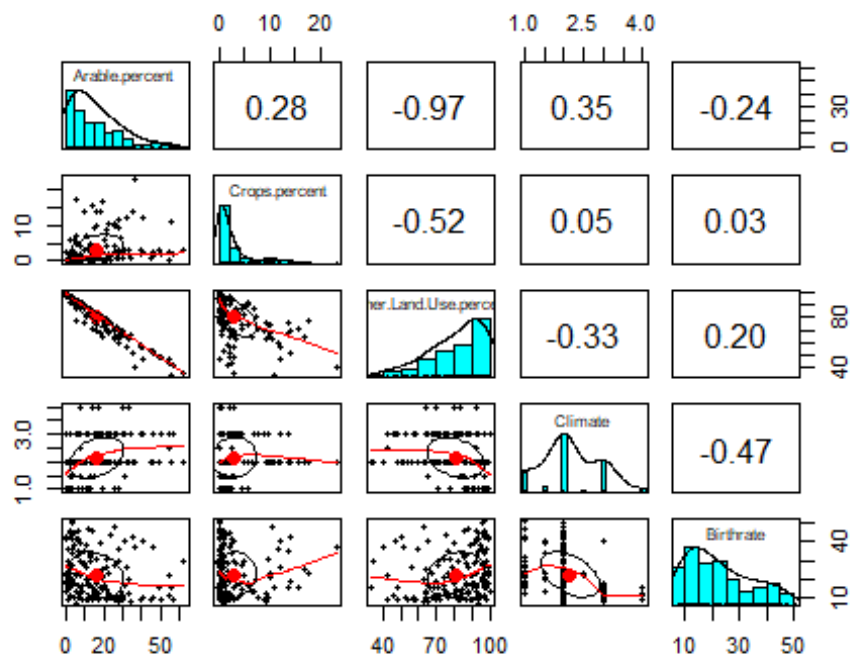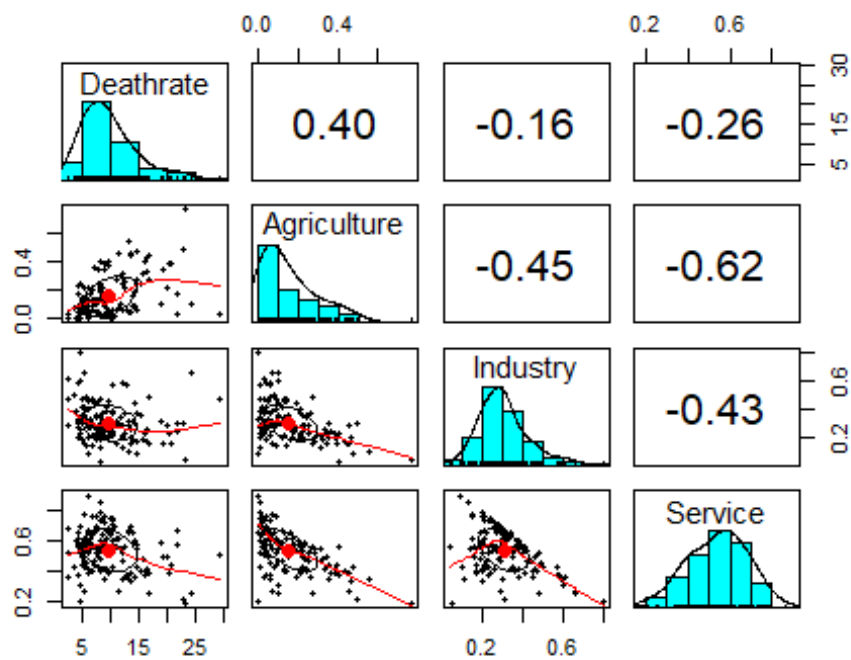


```r
pairs.panels(world_df[8:12])
```

pairs.panels(world_df[13:17])

pairs.panels(world_df[18:22])

```
pairs.panels(world_df[23:26])
```

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

```r
#normalize columns with min-max normalization by creating a function that
takes in an argument "x" and normalizes between 0-1 using the min and max
method
normalize <- function(x) {
  return( (x-min(x))/ diff(range(x)))
}


#standardize columns with z-score standardization by creating a function that
takes in an argument "y" and standardizes between +/- z-scores using z-score
standardization method
zstandardize <- function(y) {
  return( (y-mean(y))/ (sd(y)))
}
```

First let's start with HDI.Score.

```r
# Histogram with density instead of count on y-axis
# Overlay with transparent density plot

#ORIGINAL
a <- ggplot(world_df, aes(x=world_df$HDI.Score)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#LOG TRANSFORM
a1 <- ggplot(world_df, aes(x=log(world_df$HDI.Score))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")
```

```r
#INVERSE TRANSFORM
a2 <- ggplot(world_df, aes(x=1/((world_df$HDI.Score)))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#SQRT TRANSFORM
a3 <- ggplot(world_df, aes(x=sqrt(world_df$HDI.Score))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#SQUARE TRANSFORM
a4 <- ggplot(world_df, aes(x=(world_df$HDI.Score)^2)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#MIN/MAX TRANSFORM
a5 <- ggplot(world_df, aes(x=normalize(world_df$HDI.Score))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#Z-SCORE TRANSFORM
a6 <- ggplot(world_df, aes(x= zstandardize(world_df$HDI.Score))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")
#print original
a
```
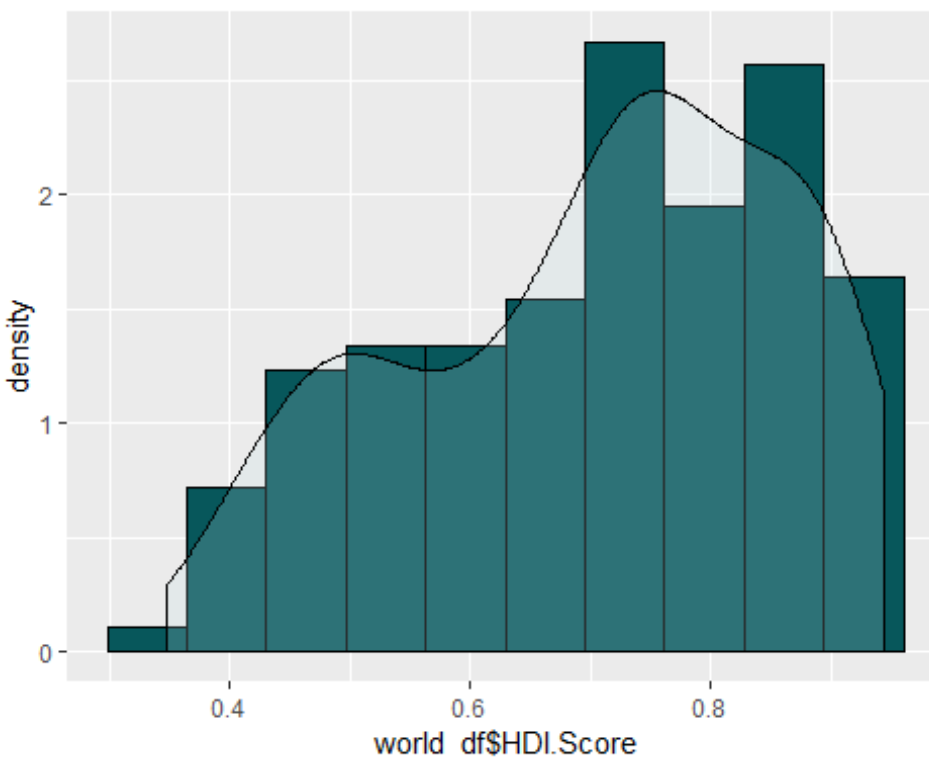
```
#print options
grid.arrange(a1,a2,a3,a4,a5,a6, nrow=3)
```


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

```
#make new data frame that is more normally distributed
world_norm_dist <- world_df

#replace feature
world_norm_dist$HDI.Score <- (world_df$HDI.Score)^2
```

Now Life Expectancy.

```
# Histogram with density instead of count on y-axis
# Overlay with transparent density plot

#ORIGINAL
a <- ggplot(world_df, aes(x=world_df$Life.Expectancy.at.Birth)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#LOG TRANSFORM
a1 <- ggplot(world_df, aes(x=log(world_df$Life.Expectancy.at.Birth))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")
```

```r
#INVERSE TRANSFORM
a2 <- ggplot(world_df, aes(x=1/((world_df$Life.Expectancy.at.Birth)))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#SQRT TRANSFORM
a3 <- ggplot(world_df, aes(x=sqrt(world_df$Life.Expectancy.at.Birth))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#SQUARE TRANSFORM
a4 <- ggplot(world_df, aes(x=(world_df$Life.Expectancy.at.Birth)^2)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#MIN/MAX TRANSFORM
a5 <- ggplot(world_df, aes(x=normalize(world_df$Life.Expectancy.at.Birth))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#Z-SCORE TRANSFORM
a6 <- ggplot(world_df, aes(x=
zstandardize(world_df$Life.Expectancy.at.Birth))) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")
#print original
a
```
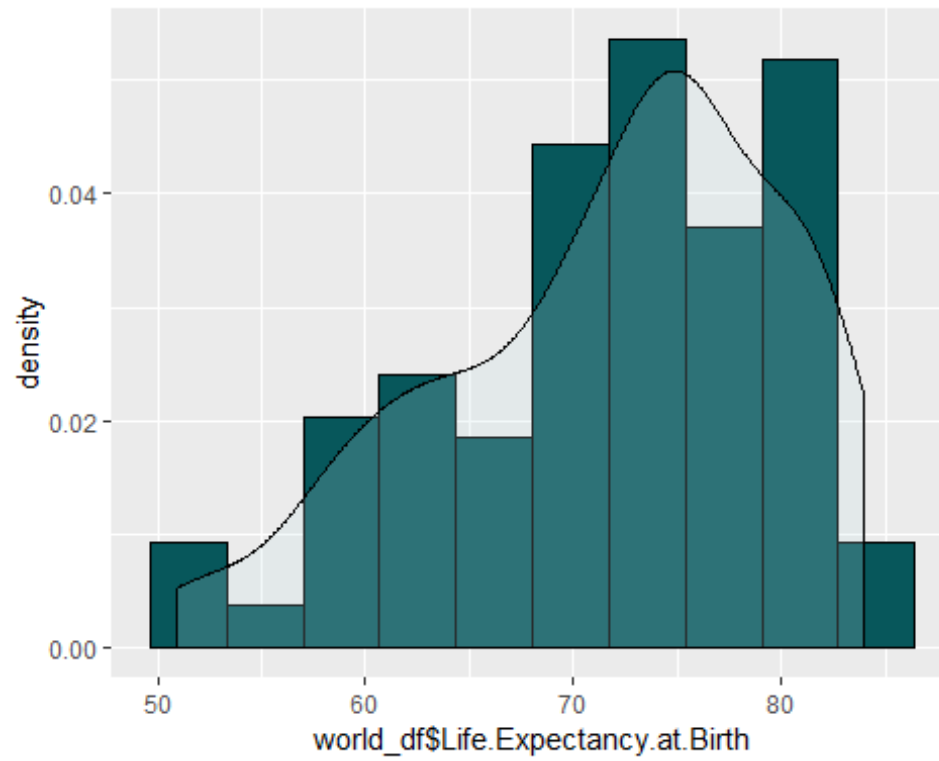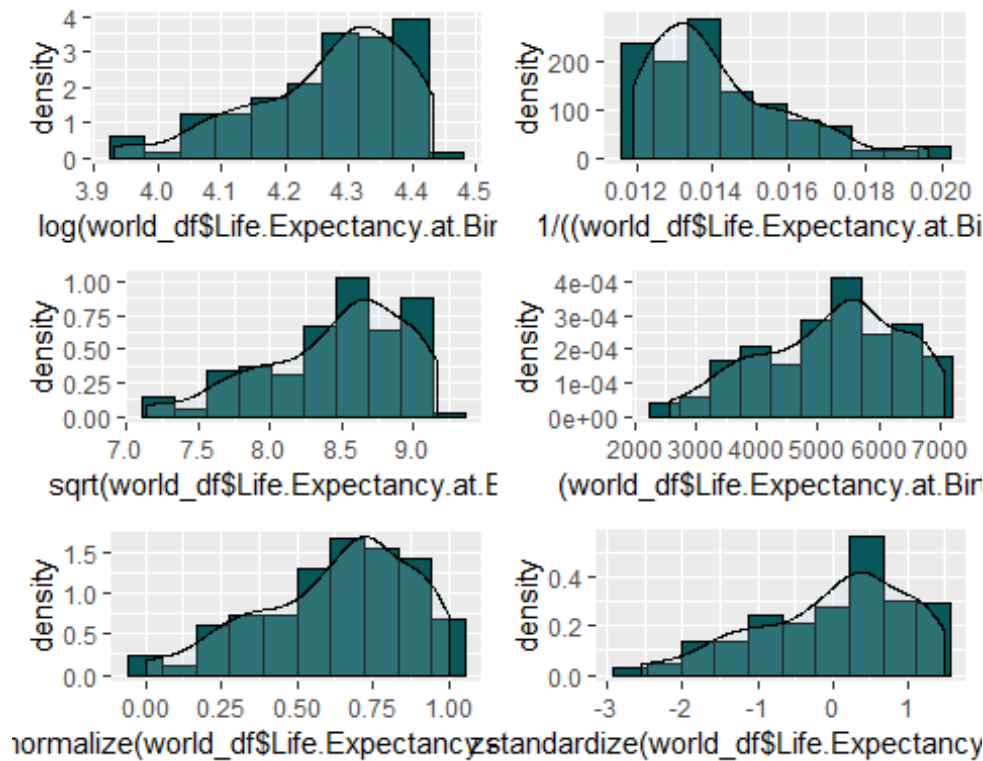
```r
#print options
grid.arrange(a1,a2,a3,a4,a5,a6, nrow=3)
```

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.

```
#replace feature
world_norm_dist$Life.Expectancy.at.Birth <-
normalize(world_df$Life.Expectancy.at.Birth)
```

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.

```
#install.packages("bestNormalize")
library(bestNormalize)
set.seed(300)


# Pick the best one automatically for the remaining features
#k = number of folds and r = number of repeats for the CV. Helps with
run=time performance
```

```
mean.edu.t <- bestNormalize(world_df$Mean.Years.of.Education, allow_orderNorm
= F, k = 5, r = 3)

gni.t <- bestNormalize(world_df$Gross.National.Income.per.Capita,
allow_orderNorm = F, k = 5, r = 3)

## Warning in bestNormalize(world_df$Gross.National.Income.per.Capita,
allow_orderNorm = F, : exp_x  did not work;  Error in exp_x(standardize =
TRUE, warn = TRUE, x = c(44025, 56431, 35182,  :
##    Transformation finite for less than 3 x values

pop.t <- bestNormalize(world_df$Population, allow_orderNorm = F, k = 5, r =
3)

## Warning in bestNormalize(world_df$Population, allow_orderNorm = F, k = 5,
: exp_x  did not work;  Error in exp_x(standardize = TRUE, warn = TRUE, x =
c(5450661L, 7523934L,  :
##    Transformation finite for less than 3 x values

area.t <- bestNormalize(world_df$Area.sq.mi, allow_orderNorm = F, k = 5, r =
3)

## Warning in bestNormalize(world_df$Area.sq.mi, allow_orderNorm = F, k = 5,
: exp_x  did not work;  Error in exp_x(standardize = TRUE, warn = TRUE, x =
c(43094L, 41290L,  :
##    Transformation finite for less than 3 x values

pop.den.t <- bestNormalize(world_df$Pop.Density.per.sq.mi, allow_orderNorm =
F, k = 5, r = 3)

## Warning in bestNormalize(world_df$Pop.Density.per.sq.mi, allow_orderNorm =
F, : exp_x  did not work;  Error in exp_x(standardize = TRUE, warn = TRUE, x
= c(126.5, 182.2, 2.9,  :
##    Transformation finite for less than 3 x values

coast.t <- bestNormalize(world_df$Coast.Area.Ratio, allow_orderNorm = F, k =
5, r = 3)

## Warning in bestNormalize(world_df$Coast.Area.Ratio, allow_orderNorm = F, :
boxcox  did not work;  Error in estimate_boxcox_lambda(x, ...) : x must be
positive

migrate.t <- bestNormalize(world_df$Net.migration, allow_orderNorm = F, k =
5, r = 3)

## Warning in bestNormalize(world_df$Net.migration, allow_orderNorm = F, k =
5, : boxcox  did not work;  Error in estimate_boxcox_lambda(x, ...) : x must
be positive

infant.t <- bestNormalize(world_df$Infant.Mortality.per.1000.births,
allow_orderNorm = F, k = 5, r = 3)
```

```r
gdp.t <- bestNormalize(world_df$GDP.per.capita, allow_orderNorm = F, k = 5, r
= 3)

## Warning in bestNormalize(world_df$GDP.per.capita, allow_orderNorm = F, k =
5, : exp_x  did not work;  Error in exp_x(standardize = TRUE, warn = TRUE, x
= c(31100L, 32700L,  :
##    Transformation finite for less than 3 x values

literacy.t <- bestNormalize(world_df$Literacy.percent, allow_orderNorm = F, k
= 5, r = 3)

phone.t <- bestNormalize(world_df$Phones.per.1000.people, allow_orderNorm =
F, k = 5, r = 3)

## Warning in bestNormalize(world_df$Phones.per.1000.people, allow_orderNorm
= F, : exp_x  did not work;  Error in exp_x(standardize = TRUE, warn = TRUE,
x = c(614.6, 680.9, 647.7,  :
##    Transformation finite for less than 3 x values

arable.t <- bestNormalize(world_df$Arable.percent, allow_orderNorm = F, k =
5, r = 3)

crop.t <- bestNormalize(world_df$Crops.percent, allow_orderNorm = F, k = 5, r
= 3)

## Warning in bestNormalize(world_df$Crops.percent, allow_orderNorm = F, k =
5, : boxcox  did not work;  Error in estimate_boxcox_lambda(x, ...) : x must
be positive

other.t <- bestNormalize(world_df$Other.Land.Use.percent, allow_orderNorm =
F, k = 5, r = 3)

climate.t <- bestNormalize(world_df$Climate, allow_orderNorm = F, k = 5, r =
3)

birth.t <- bestNormalize(world_df$Birthrate, allow_orderNorm = F, k = 5, r =
3)

death.t <- bestNormalize(world_df$Deathrate, allow_orderNorm = F, k = 5, r =
3)

agricul.t <- bestNormalize(world_df$Agriculture, allow_orderNorm = F, k = 5,
r = 3)

## Warning in bestNormalize(world_df$Agriculture, allow_orderNorm = F, k = 5,
: boxcox  did not work;  Error in estimate_boxcox_lambda(x, ...) : x must be
positive
```

As you can see, not every transform works for every feature. Yet, the best one is still 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.

```
#spot check for gni
a1 <- ggplot(world_df, aes(x= gni.t$x.t)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#spot check for deathrate
a2 <- ggplot(world_df, aes(x= death.t$x.t)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#spot check for gdp per capita
a3 <- ggplot(world_df, aes(x= gdp.t$x.t)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

#Box Cox transform for infant mortality
```

```
a4 <- ggplot(world_df, aes(x= infant.t$x.t)) +
geom_histogram(aes(y=..density..),bins=10, colour="black",
fill="#07575b")+geom_density(alpha=.2, fill="#c4dfe6")

grid.arrange(a1,a2,a3,a4,nrow=2)
```
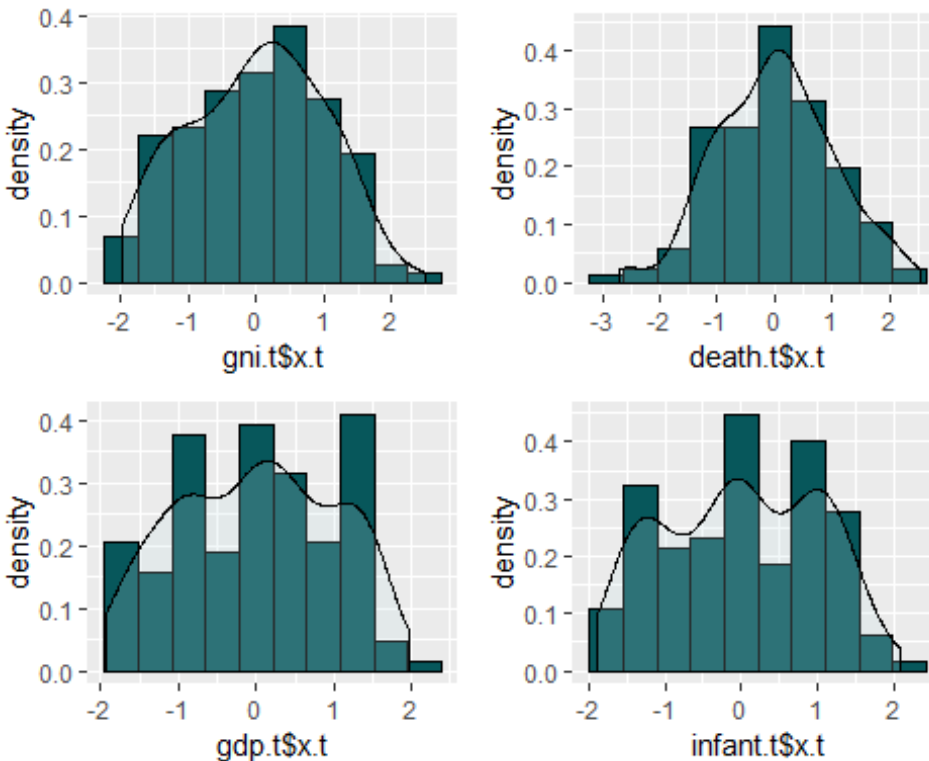

Looks a lot better than before! Now let's apply all of these to the normally distributed data frame. If at the end of our regression analysis we have to reverse any transform, we can easily access which transform was applied using the $chosen_transform call.

```
#replace features with transforms
world_norm_dist$Mean.Years.of.Education <- mean.edu.t$x.t
world_norm_dist$Gross.National.Income.per.Capita <- gni.t$x.t
world_norm_dist$Population <- pop.t$x.t
world_norm_dist$Area.sq.mi <- area.t$x.t
world_norm_dist$Pop.Density.per.sq.mi <- pop.den.t$x.t
world_norm_dist$Coast.Area.Ratio <- coast.t$x.t
world_norm_dist$Net.migration <- migrate.t$x.t
world_norm_dist$Infant.Mortality.per.1000.births <- infant.t$x.t
world_norm_dist$GDP.per.capita <- gdp.t$x.t
world_norm_dist$Literacy.percent <- literacy.t$x.t
world_norm_dist$Phones.per.1000.people <- phone.t$x.t
world_norm_dist$Arable.percent <- arable.t$x.t
world_norm_dist$Crops.percent <- crop.t$x.t
world_norm_dist$Other.Land.Use.percent <- other.t$x.t
world_norm_dist$Climate <- climate.t$x.t
world_norm_dist$Birthrate <- birth.t$x.t
```

```
world_norm_dist$Deathrate <- death.t$x.t
world_norm_dist$Agriculture <- agricul.t$x.t
```

Look at the pairs.panels again.

```
pairs.panels(world_norm_dist[2:7])
```
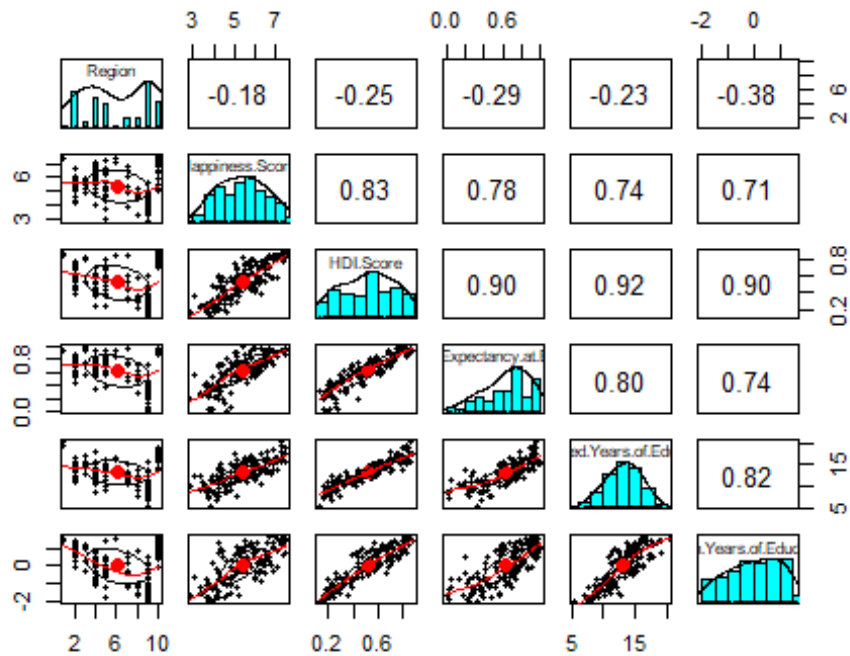


```
pairs.panels(world_norm_dist[8:12])
```
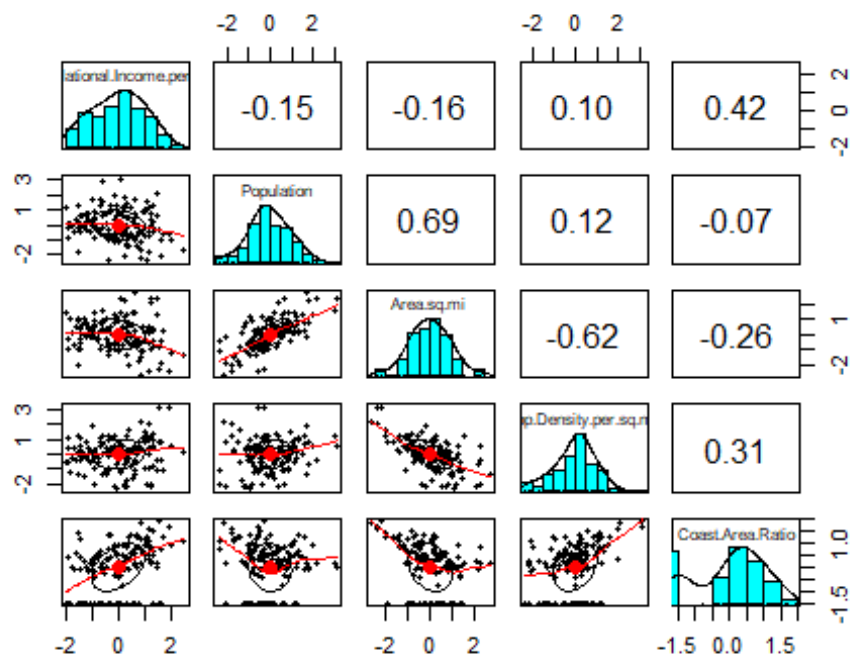
```
pairs.panels(world_norm_dist[13:17])
```
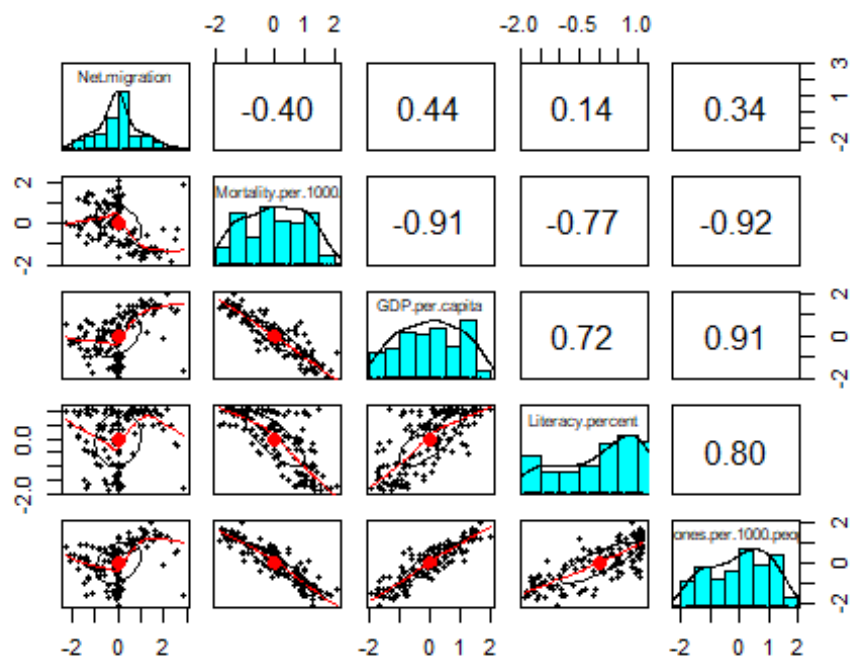


```
pairs.panels(world_norm_dist[18:22])
```
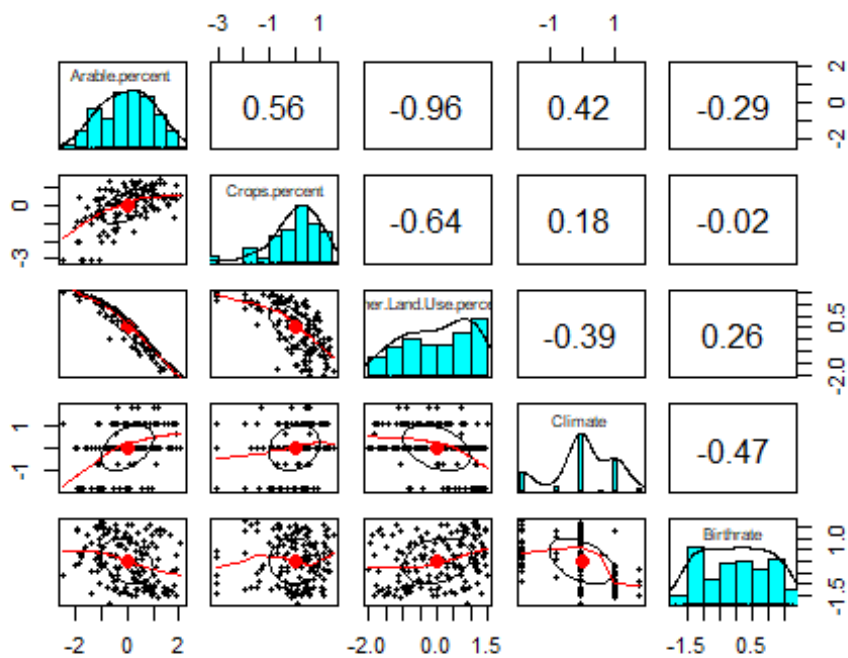
```
pairs.panels(world_norm_dist[23:26])
```
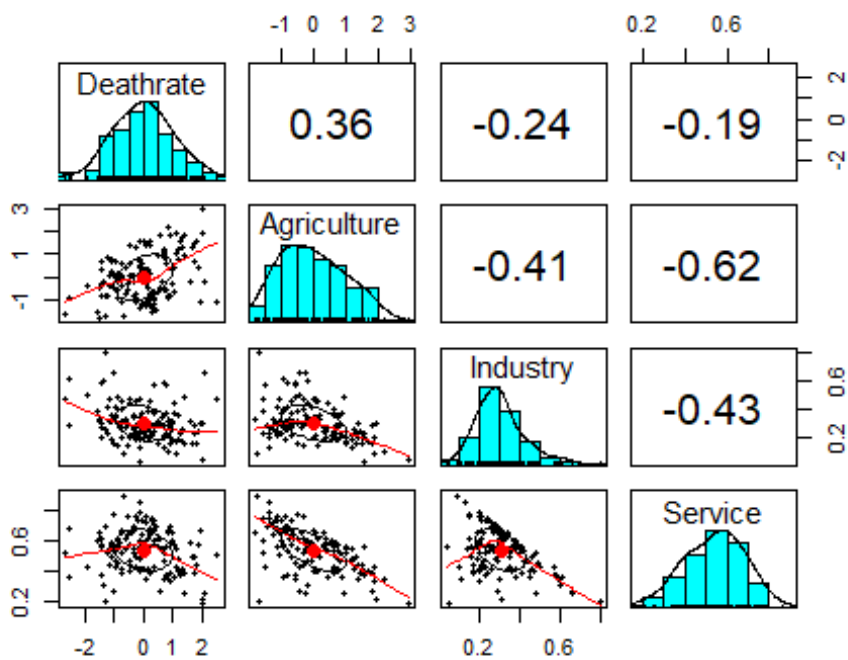

It looks much better!

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
there 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.

```
world_norm_dist <- world_norm_dist[-2]
```

Remove spaces and special characters from new variable names.

```
world_norm_dist <- rename(world_norm_dist, Region.AusNZ = "RegionAustralia
and New Zealand", Region.Cen.E.Eur = "RegionCentral and Eastern Europe",
Region.E.Asia = "RegionEastern Asia", Region.LatCari = "RegionLatin America
and Caribbean", Region.MENA = "RegionMiddle East and Northern Africa",
Region.N.Amer = "RegionNorth America", Region.SE.Asia = "RegionSoutheastern
Asia", Region.S.Asia = "RegionSouthern Asia", Region.SS.Africa = "RegionSub-
Saharan Africa")
```

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.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, data =
world_norm_dist)
```

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

```r
#install.packages("car")
library(car)

## 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

round(vif(m1),2)

##                          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.

```
#make model for features
m2 <- lm(Happiness.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, data =
world_norm_dist)

round(vif(m2),2)
```

```
##          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.

```
#make model for features
m3 <- 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_norm_dist)

round(vif(m3),2)
```

```
##          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!

```r
wdata <-world_norm_dist[,-c(1,2)]
pcal <- princomp(wdata, scores = TRUE, cor = TRUE)
summary(pcal)
```

```
## 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.

```
#same thing
pcal$loadings

##
## 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
```

```
##                                                      -0.135 -0.152
## Region.E.Asia
## 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

```
plot(pcal)
```
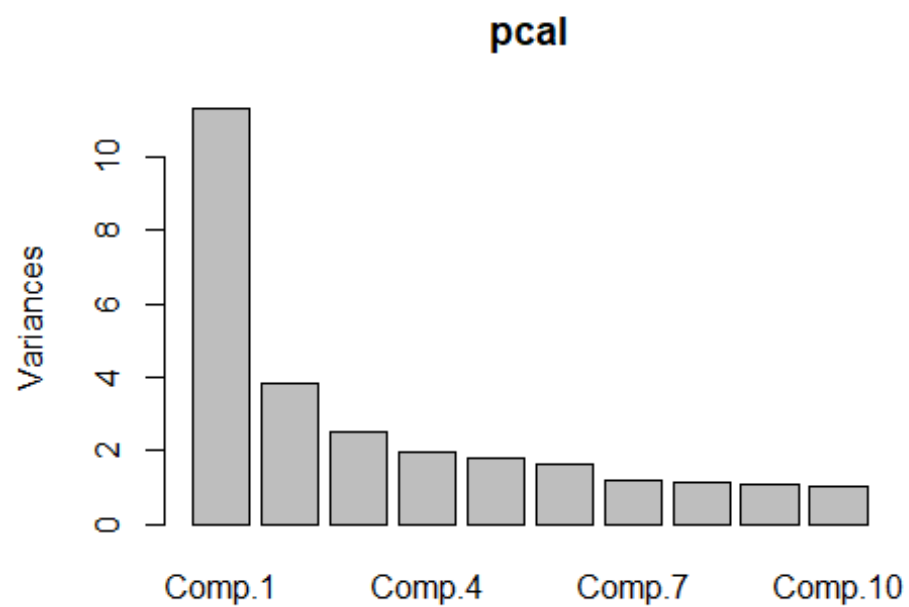
pcal

```
screeplot(pcal, type = "line", main = "Scree Plot")
```


Scree Plot
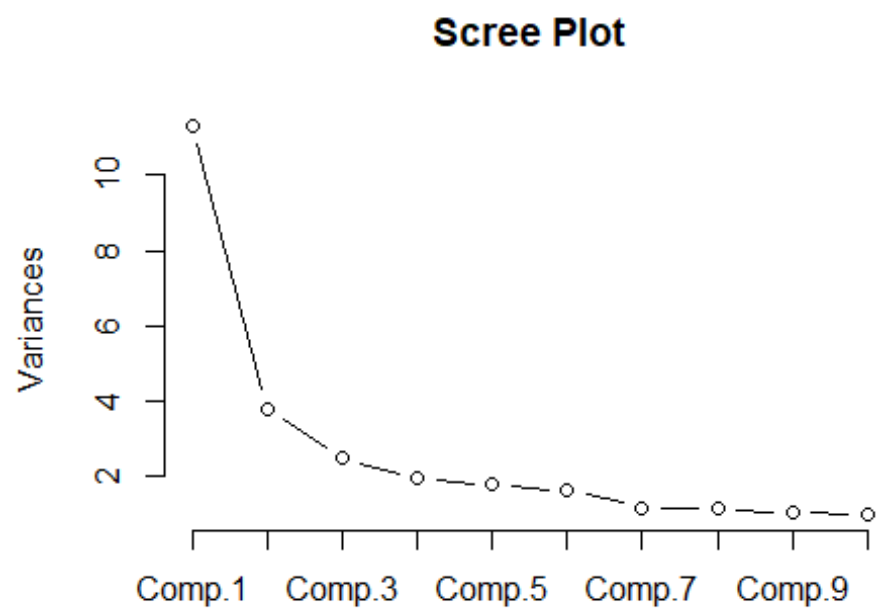
This shows us the importance of the first few components.

Now a biplot of score variables

```
biplot(pcal)
```
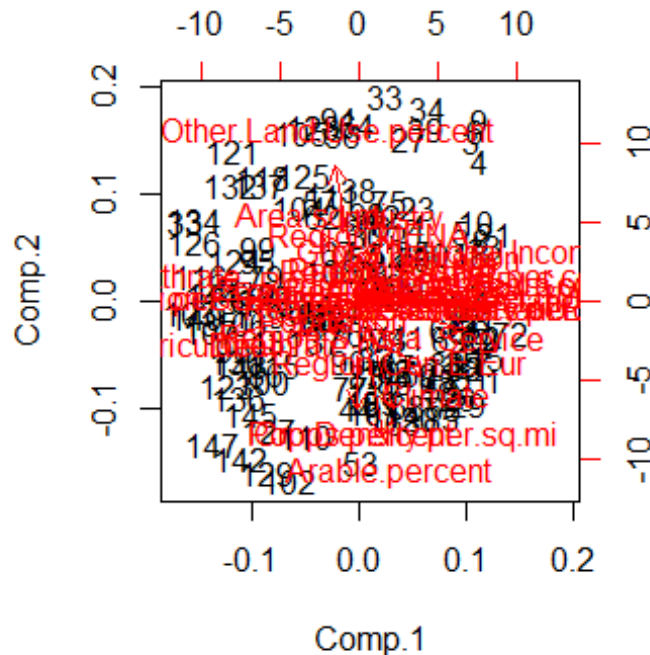

This is too messy to read.

Scores of the components.

```
pcal$scores[1:5,]
```

```
##      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.

```
#remove HDI score and Area Sq mi
world_norm_dist2 <- world_norm_dist[-c(3,9)]
```

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.

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

## Loading required package: lattice

##
## Attaching package: 'caret'

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

set.seed(12)
# y = happiness score bc it is the vector of outcomes
#80% for training, 20% for testing
```

```
indxTrain <- createDataPartition(y = world_norm_dist2$Happiness.Score, p =
0.80, list = FALSE)

#80% of the full dataset goes to training and the rest to testing. the [-]
syntax places all not-yet indexed values to the remaining set.
world_train <- world_norm_dist2[indxTrain,]
world_test <- world_norm_dist2[-indxTrain,]

#make sure distributions are similar in traing and test
summary(world_test$Happiness.Score)

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

summary(world_train$Happiness.Score)

##    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.

```
#make model for all features except country
m4 <- 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)
summary(m4)

##
## Call:
## lm(formula = 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)
##
## 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.

```
m5 <- 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.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)
summary(m5)

##
## Call:
## lm(formula = 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.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
## -1.35092 -0.26414  0.00483  0.27068  1.20077
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.413763   1.762589   2.504 0.014098 *
## Life.Expectancy.at.Birth          0.625499   0.646970   0.967 0.336258
## Expected.Years.of.Education      -0.004412   0.047804  -0.092 0.926666
## Mean.Years.of.Education           0.327477   0.182840   1.791 0.076684 .
## Gross.National.Income.per.Capita  0.674032   0.209773   3.213 0.001829 **
## Population                        0.061956   0.061026   1.015 0.312741
## Pop.Density.per.sq.mi             0.058400   0.105641   0.553 0.581774
## Coast.Area.Ratio                 -0.070540   0.068569  -1.029 0.306386
## Net.migration                     0.058595   0.074592   0.786 0.434222
## Infant.Mortality.per.1000.births -0.068023   0.187484  -0.363 0.717599
## GDP.per.capita                    0.014024   0.191340   0.073 0.941738
## Literacy.percent                  0.119937   0.158676   0.756 0.451727
## Phones.per.1000.people            0.167696   0.203213   0.825 0.411452
## Arable.percent                    0.191991   0.230019   0.835 0.406137
## Crops.percent                    -0.061247   0.086271  -0.710 0.479598
## Other.Land.Use.percent            0.307638   0.214102   1.437 0.154260
## Climate                           0.063498   0.067977   0.934 0.352772
## Birthrate                         0.224946   0.178168   1.263 0.210049
## Deathrate                        -0.189241   0.107910  -1.754 0.082926 .
## Agriculture                       0.473080   0.386364   1.224 0.224018
## Industry                          1.096501   2.033539   0.539 0.591090
```

```
## Service                              1.034646    1.993839    0.519 0.605103
## Region.Cen.E.Eur                      -0.675150   0.274847   -2.456 0.015972 *
## Region.E.Asia                         -1.104298   0.324296   -3.405 0.000994 ***
## Region.LatCari                         0.429134   0.278616    1.540 0.127051
## Region.MENA                           -0.625363   0.315036   -1.985 0.050218 .
## Region.N.Amer                         -0.141163   0.447864   -0.315 0.753354
## Region.SE.Asia                        -0.454258   0.343544   -1.322 0.189467
## Region.S.Asia                         -0.131763   0.408238   -0.323 0.747633
## Region.SS.Africa                      -0.202465   0.340346   -0.595 0.553434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5063 on 89 degrees of freedom
## Multiple R-squared:  0.8589, Adjusted R-squared:  0.813
## F-statistic: 18.69 on 29 and 89 DF,  p-value: < 2.2e-16
```

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

```
m6 <- 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 +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service +
    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)
summary(m6)

##
## Call:
## lm(formula = 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 + Literacy.percent +
Phones.per.1000.people +
##      Arable.percent + Crops.percent + Other.Land.Use.percent +
##      Climate + Birthrate + Deathrate + Agriculture + Industry +
##      Service + 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
## -1.34608 -0.26303  0.00632  0.27097  1.20306
```

```
## 
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          4.433555   1.732129   2.560 0.012147 *
## Life.Expectancy.at.Birth             0.626771   0.643154   0.975 0.332407
## Expected.Years.of.Education         -0.004526   0.047514  -0.095 0.924319
## Mean.Years.of.Education              0.327188   0.181785   1.800 0.075233 .
## Gross.National.Income.per.Capita     0.680403   0.189863   3.584 0.000550 ***
## Population                           0.061941   0.060688   1.021 0.310154
## Pop.Density.per.sq.mi                0.056319   0.101187   0.557 0.579198
## Coast.Area.Ratio                    -0.070401   0.068163  -1.033 0.304448
## Net.migration                        0.058491   0.074165   0.789 0.432384
## Infant.Mortality.per.1000.births    -0.070267   0.183942  -0.382 0.703357
## Literacy.percent                     0.119601   0.157731   0.758 0.450275
## Phones.per.1000.people               0.169931   0.199798   0.851 0.397296
## Arable.percent                       0.192789   0.228488   0.844 0.401042
## Crops.percent                       -0.060312   0.084849  -0.711 0.479039
## Other.Land.Use.percent               0.307680   0.212915   1.445 0.151907
## Climate                              0.063372   0.067578   0.938 0.350880
## Birthrate                            0.222365   0.173687   1.280 0.203742
## Deathrate                           -0.189585   0.107211  -1.768 0.080392 .
## Agriculture                          0.467709   0.377250   1.240 0.218278
## Industry                             1.070697   1.991733   0.538 0.592200
## Service                              1.015311   1.965358   0.517 0.606700
## Region.Cen.E.Eur                    -0.678783   0.268844  -2.525 0.013327 *
## Region.E.Asia                       -1.106888   0.320577  -3.453 0.000847 ***
## Region.LatCari                       0.427828   0.276505   1.547 0.125308
## Region.MENA                         -0.626860   0.312630  -2.005 0.047955 *
## Region.N.Amer                       -0.142415   0.445059  -0.320 0.749716
## Region.SE.Asia                      -0.453965   0.341617  -1.329 0.187251
## Region.S.Asia                       -0.130884   0.405801  -0.323 0.747798
## Region.SS.Africa                    -0.200505   0.337415  -0.594 0.553843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5035 on 90 degrees of freedom
## Multiple R-squared:  0.8589, Adjusted R-squared:  0.815
## F-statistic: 19.57 on 28 and 90 DF,  p-value: < 2.2e-16
```

Now expected year of education.

```
m7 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    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 +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service +
    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)
summary(m7)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
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 +
##      Literacy.percent + Phones.per.1000.people + Arable.percent +
##      Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
##      Deathrate + Agriculture + Industry + Service + 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
## -1.34515 -0.26512  0.00537  0.27081  1.20083
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        4.39145    1.66563   2.637 0.009849 **
## Life.Expectancy.at.Birth           0.62160    0.63736   0.975 0.332011
## Mean.Years.of.Education            0.32359    0.17685   1.830 0.070559 .
## Gross.National.Income.per.Capita   0.67835    0.18761   3.616 0.000491 ***
## Population                         0.06197    0.06036   1.027 0.307223
## Pop.Density.per.sq.mi              0.05956    0.09478   0.628 0.531304
## Coast.Area.Ratio                  -0.07111    0.06738  -1.055 0.294036
## Net.migration                      0.05714    0.07240   0.789 0.432021
## Infant.Mortality.per.1000.births  -0.06787    0.18121  -0.375 0.708895
## Literacy.percent                   0.11763    0.15552   0.756 0.451366
## Phones.per.1000.people             0.16803    0.19772   0.850 0.397628
## Arable.percent                     0.18879    0.22338   0.845 0.400232
## Crops.percent                     -0.05998    0.08431  -0.711 0.478676
## Other.Land.Use.percent             0.30593    0.21096   1.450 0.150452
## Climate                            0.06230    0.06627   0.940 0.349672
## Birthrate                          0.22363    0.17224   1.298 0.197440
## Deathrate                         -0.18813    0.10554  -1.783 0.077997 .
## Agriculture                        0.46441    0.37361   1.243 0.217045
## Industry                           1.05502    1.97409   0.534 0.594343
## Service                            1.00201    1.94969   0.514 0.608544
## Region.Cen.E.Eur                  -0.67617    0.26598  -2.542 0.012706 *
## Region.E.Asia                     -1.10981    0.31736  -3.497 0.000730 ***
## Region.LatCari                     0.42605    0.27437   1.553 0.123932
## Region.MENA                       -0.63168    0.30682  -2.059 0.042372 *
## Region.N.Amer                     -0.13410    0.43403  -0.309 0.758055
## Region.SE.Asia                    -0.45598    0.33910  -1.345 0.182075
## Region.S.Asia                     -0.14092    0.38974  -0.362 0.718498
## Region.SS.Africa                  -0.20819    0.32585  -0.639 0.524490
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5007 on 91 degrees of freedom
## Multiple R-squared:  0.8589, Adjusted R-squared:  0.8171
## F-statistic: 20.52 on 27 and 91 DF,  p-value: < 2.2e-16
```

Now region North America.

```
m8 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    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 +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia + Region.S.Asia + Region.SS.Africa , data = world_train)
summary(m8)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
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 +
##     Literacy.percent + Phones.per.1000.people + Arable.percent +
##     Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
##     Deathrate + Agriculture + Industry + Service + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia +
##     Region.S.Asia + Region.SS.Africa, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.34191 -0.26442  0.00698  0.27416  1.19809
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.30730    1.63512   2.634 0.009893 **
## Life.Expectancy.at.Birth          0.61589    0.63395   0.972 0.333840
## Mean.Years.of.Education           0.31536    0.17397   1.813 0.073133 .
## Gross.National.Income.per.Capita  0.68433    0.18568   3.685 0.000386 ***
## Population                        0.05883    0.05920   0.994 0.322935
## Pop.Density.per.sq.mi             0.06705    0.09117   0.736 0.463903
## Coast.Area.Ratio                 -0.07297    0.06678  -1.093 0.277417
## Net.migration                     0.05417    0.07140   0.759 0.449990
## Infant.Mortality.per.1000.births -0.07743    0.17767  -0.436 0.663990
## Literacy.percent                  0.12828    0.15091   0.850 0.397498
## Phones.per.1000.people            0.15414    0.19159   0.805 0.423153
```

```
## Arable.percent                    0.18405    0.22175    0.830 0.408696
## Crops.percent                     -0.06005    0.08390   -0.716 0.475974
## Other.Land.Use.percent            0.30382    0.20981    1.448 0.151009
## Climate                           0.06037    0.06565    0.920 0.360218
## Birthrate                         0.21788    0.17038    1.279 0.204202
## Deathrate                        -0.18287    0.10364   -1.764 0.080987 .
## Agriculture                       0.48333    0.36675    1.318 0.190816
## Industry                          1.16291    1.93338    0.601 0.548992
## Service                           1.08931    1.91961    0.567 0.571781
## Region.Cen.E.Eur                 -0.67612    0.26466   -2.555 0.012273 *
## Region.E.Asia                    -1.10427    0.31530   -3.502 0.000714 ***
## Region.LatCari                    0.43575    0.27122    1.607 0.111556
## Region.MENA                      -0.61521    0.30066   -2.046 0.043593 *
## Region.SE.Asia                   -0.45439    0.33739   -1.347 0.181364
## Region.S.Asia                    -0.13178    0.38670   -0.341 0.734041
## Region.SS.Africa                 -0.20370    0.32392   -0.629 0.531002
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4983 on 92 degrees of freedom
## Multiple R-squared:  0.8588, Adjusted R-squared:  0.8189
## F-statistic: 21.52 on 26 and 92 DF,  p-value: < 2.2e-16
```

Now region South Asia.

```
m9 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    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 +
    Literacy.percent + Phones.per.1000.people + Arable.percent +
    Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
    Deathrate + Agriculture + Industry + Service +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia + Region.SS.Africa , data = world_train)
summary(m9)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
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 +
##     Literacy.percent + Phones.per.1000.people + Arable.percent +
##     Crops.percent + Other.Land.Use.percent + Climate + Birthrate +
##     Deathrate + Agriculture + Industry + Service + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia +
##     Region.SS.Africa, data = world_train)
##
## Residuals:
```

```
##       Min       1Q   Median       3Q      Max
## -1.35152 -0.26412  0.00566  0.26274  1.20777
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.31483    1.62718   2.652 0.009415 **
## Life.Expectancy.at.Birth          0.59839    0.62886   0.952 0.343793
## Mean.Years.of.Education           0.31631    0.17312   1.827 0.070888 .
## Gross.National.Income.per.Capita  0.68260    0.18473   3.695 0.000371 ***
## Population                        0.05685    0.05863   0.970 0.334747
## Pop.Density.per.sq.mi             0.06039    0.08862   0.681 0.497301
## Coast.Area.Ratio                 -0.07223    0.06643  -1.087 0.279716
## Net.migration                     0.05226    0.07084   0.738 0.462598
## Infant.Mortality.per.1000.births -0.09142    0.17204  -0.531 0.596421
## Literacy.percent                  0.13792    0.14752   0.935 0.352259
## Phones.per.1000.people            0.16774    0.18650   0.899 0.370743
## Arable.percent                    0.19434    0.21864   0.889 0.376374
## Crops.percent                    -0.05705    0.08304  -0.687 0.493770
## Other.Land.Use.percent            0.31312    0.20704   1.512 0.133838
## Climate                           0.06091    0.06532   0.933 0.353479
## Birthrate                         0.22303    0.16890   1.320 0.189916
## Deathrate                        -0.18387    0.10311  -1.783 0.077804 .
## Agriculture                       0.48080    0.36492   1.318 0.190895
## Industry                          1.11484    1.91904   0.581 0.562690
## Service                           1.04147    1.90535   0.547 0.585959
## Region.Cen.E.Eur                 -0.64793    0.25021  -2.590 0.011155 *
## Region.E.Asia                    -1.06914    0.29654  -3.605 0.000504 ***
## Region.LatCari                    0.47314    0.24685   1.917 0.058349 .
## Region.MENA                      -0.56481    0.26053  -2.168 0.032716 *
## Region.SE.Asia                   -0.39781    0.29232  -1.361 0.176843
## Region.SS.Africa                 -0.13574    0.25405  -0.534 0.594396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4959 on 93 degrees of freedom
## Multiple R-squared:  0.8586, Adjusted R-squared:  0.8206
## F-statistic: 22.59 on 25 and 93 DF,  p-value: < 2.2e-16
```

Now infant mortality.

```
m10 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Literacy.percent + Phones.per.1000.people +
Arable.percent + Crops.percent + Other.Land.Use.percent + Climate + Birthrate
+ Deathrate + Agriculture + Industry + Service +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia + Region.SS.Africa , data = world_train)
summary(m10)
```

```
## 
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Population + Pop.Density.per.sq.mi
+
##     Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people +
##     Arable.percent + Crops.percent + Other.Land.Use.percent +
##     Climate + Birthrate + Deathrate + Agriculture + Industry +
##     Service + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari +
##     Region.MENA + Region.SE.Asia + Region.SS.Africa, data = world_train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.32318 -0.26228  0.02153  0.25243  1.21123
## 
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        4.41974    1.60898   2.747 0.007211 **
## Life.Expectancy.at.Birth           0.70173    0.59574   1.178 0.241803
## Mean.Years.of.Education            0.31801    0.17243   1.844 0.068291 .
## Gross.National.Income.per.Capita   0.66956    0.18240   3.671 0.000401 ***
## Population                         0.05427    0.05821   0.932 0.353509
## Pop.Density.per.sq.mi              0.05421    0.08752   0.619 0.537158
## Coast.Area.Ratio                  -0.06947    0.06597  -1.053 0.295030
## Net.migration                      0.06039    0.06890   0.876 0.383016
## Literacy.percent                   0.13455    0.14682   0.916 0.361811
## Phones.per.1000.people             0.19251    0.17989   1.070 0.287293
## Arable.percent                     0.18783    0.21746   0.864 0.389932
## Crops.percent                     -0.06166    0.08227  -0.750 0.455424
## Other.Land.Use.percent             0.30193    0.20518   1.472 0.144484
## Climate                            0.07080    0.06237   1.135 0.259200
## Birthrate                          0.19942    0.16233   1.228 0.222330
## Deathrate                         -0.20194    0.09697  -2.083 0.040006 *
## Agriculture                        0.42777    0.34967   1.223 0.224255
## Industry                           0.87730    1.85912   0.472 0.638099
## Service                            0.85435    1.86536   0.458 0.648003
## Region.Cen.E.Eur                  -0.64512    0.24920  -2.589 0.011161 *
## Region.E.Asia                     -1.07613    0.29512  -3.646 0.000436 ***
## Region.LatCari                     0.46620    0.24557   1.898 0.060703 .
## Region.MENA                       -0.58042    0.25788  -2.251 0.026732 *
## Region.SE.Asia                    -0.39467    0.29114  -1.356 0.178475
## Region.SS.Africa                  -0.11253    0.24931  -0.451 0.652772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.494 on 94 degrees of freedom
## Multiple R-squared:  0.8582, Adjusted R-squared:  0.8219
## F-statistic:  23.7 on 24 and 94 DF,  p-value: < 2.2e-16
```

Now region Sub-saharan Africa.

```
m11 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Literacy.percent + Phones.per.1000.people +
Arable.percent + Crops.percent + Other.Land.Use.percent + Climate + Birthrate
+ Deathrate + Agriculture + Industry + Service +
    Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia , data = world_train)
summary(m11)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##      Gross.National.Income.per.Capita + Population + Pop.Density.per.sq.mi
+
##      Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people +
##      Arable.percent + Crops.percent + Other.Land.Use.percent +
##      Climate + Birthrate + Deathrate + Agriculture + Industry +
##      Service + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari +
##      Region.MENA + Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31211 -0.26597  0.01693  0.25652  1.20039
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.18655    1.51738   2.759 0.006955 **
## Life.Expectancy.at.Birth          0.80639    0.54646   1.476 0.143344
## Mean.Years.of.Education           0.32740    0.17045   1.921 0.057750 .
## Gross.National.Income.per.Capita  0.67950    0.18030   3.769 0.000285 ***
## Population                        0.05805    0.05736   1.012 0.314070
## Pop.Density.per.sq.mi             0.05584    0.08708   0.641 0.522916
## Coast.Area.Ratio                 -0.07249    0.06536  -1.109 0.270173
## Net.migration                     0.06510    0.06782   0.960 0.339525
## Literacy.percent                  0.12494    0.14466   0.864 0.389949
## Phones.per.1000.people            0.19663    0.17890   1.099 0.274496
## Arable.percent                    0.19587    0.21582   0.908 0.366420
## Crops.percent                    -0.06595    0.08137  -0.810 0.419706
## Other.Land.Use.percent            0.30652    0.20407   1.502 0.136410
## Climate                           0.06881    0.06196   1.111 0.269515
## Birthrate                         0.19932    0.16165   1.233 0.220603
## Deathrate                        -0.20452    0.09639  -2.122 0.036466 *
## Agriculture                       0.46053    0.34062   1.352 0.179575
## Industry                          1.01820    1.82503   0.558 0.578219
## Service                           0.98976    1.83335   0.540 0.590552
```

```
## Region.Cen.E.Eur                    -0.60382    0.23082  -2.616 0.010350 *
## Region.E.Asia                       -1.05377    0.28971  -3.637 0.000448 ***
## Region.LatCari                       0.52163    0.21176   2.463 0.015565 *
## Region.MENA                         -0.54484    0.24450  -2.228 0.028216 *
## Region.SE.Asia                      -0.35131    0.27368  -1.284 0.202387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4919 on 95 degrees of freedom
## Multiple R-squared:  0.8579, Adjusted R-squared:  0.8234
## F-statistic: 24.93 on 23 and 95 DF,  p-value: < 2.2e-16
```

Now service.

```
m12 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Literacy.percent + Phones.per.1000.people +
Arable.percent + Crops.percent + Other.Land.Use.percent + Climate + Birthrate
+ Deathrate + Agriculture + Industry + Region.Cen.E.Eur + Region.E.Asia +
Region.LatCari + Region.MENA + Region.SE.Asia , data = world_train)
summary(m12)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Population + Pop.Density.per.sq.mi
+
##     Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people +
##     Arable.percent + Crops.percent + Other.Land.Use.percent +
##     Climate + Birthrate + Deathrate + Agriculture + Industry +
##     Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
##     Region.SE.Asia, data = world_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.32575 -0.29040  0.02473  0.24162  1.20297
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.97596    0.40385  12.321  < 2e-16 ***
## Life.Expectancy.at.Birth         0.84539    0.53966   1.567 0.120515
## Mean.Years.of.Education          0.29393    0.15819   1.858 0.066212 .
## Gross.National.Income.per.Capita 0.64644    0.16896   3.826 0.000232 ***
## Population                       0.06208    0.05666   1.096 0.275948
## Pop.Density.per.sq.mi            0.03731    0.07974   0.468 0.640863
## Coast.Area.Ratio                -0.07484    0.06497  -1.152 0.252193
## Net.migration                    0.05750    0.06610   0.870 0.386511
```

```
## Literacy.percent                    0.13562    0.14277    0.950 0.344554
## Phones.per.1000.people              0.21369    0.17544    1.218 0.226211
## Arable.percent                      0.19042    0.21479    0.887 0.377535
## Crops.percent                      -0.06802    0.08098   -0.840 0.403012
## Other.Land.Use.percent              0.28672    0.20001    1.434 0.154944
## Climate                             0.06755    0.06168    1.095 0.276183
## Birthrate                           0.20427    0.16079    1.270 0.207001
## Deathrate                          -0.21389    0.09446   -2.264 0.025807 *
## Agriculture                         0.29235    0.13724    2.130 0.035713 *
## Industry                            0.08499    0.58324    0.146 0.884448
## Region.Cen.E.Eur                   -0.56978    0.22122   -2.576 0.011531 *
## Region.E.Asia                      -1.04111    0.28769   -3.619 0.000474 ***
## Region.LatCari                      0.51361    0.21045    2.440 0.016502 *
## Region.MENA                        -0.55271    0.24317   -2.273 0.025258 *
## Region.SE.Asia                     -0.34451    0.27238   -1.265 0.208995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4901 on 96 degrees of freedom
## Multiple R-squared:  0.8574, Adjusted R-squared:  0.8247
## F-statistic: 26.24 on 22 and 96 DF,  p-value: < 2.2e-16
```

Now industry.

```
m13 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Pop.Density.per.sq.mi + Coast.Area.Ratio +
    Net.migration + Literacy.percent + Phones.per.1000.people +
Arable.percent + Crops.percent + Other.Land.Use.percent + Climate + Birthrate
+ Deathrate + Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari
+ Region.MENA + Region.SE.Asia , data = world_train)
summary(m13)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Population + Pop.Density.per.sq.mi
+
##     Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people +
##     Arable.percent + Crops.percent + Other.Land.Use.percent +
##     Climate + Birthrate + Deathrate + Agriculture + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia,
##     data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.32680 -0.29875  0.03032  0.24470  1.21021
##
```

```
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.00172    0.36127  13.845  < 2e-16 ***
## Life.Expectancy.at.Birth         0.83923    0.53528   1.568 0.120176
## Mean.Years.of.Education          0.29238    0.15703   1.862 0.065636 .
## Gross.National.Income.per.Capita 0.65344    0.16117   4.054 0.000102 ***
## Population                       0.06328    0.05578   1.134 0.259402
## Pop.Density.per.sq.mi            0.03436    0.07673   0.448 0.655266
## Coast.Area.Ratio                -0.07626    0.06392  -1.193 0.235771
## Net.migration                    0.05596    0.06492   0.862 0.390802
## Literacy.percent                 0.13222    0.14014   0.943 0.347786
## Phones.per.1000.people           0.20881    0.17134   1.219 0.225937
## Arable.percent                   0.18911    0.21352   0.886 0.377966
## Crops.percent                   -0.06541    0.07857  -0.832 0.407183
## Other.Land.Use.percent           0.28653    0.19899   1.440 0.153118
## Climate                          0.06710    0.06129   1.095 0.276359
## Birthrate                        0.20219    0.15934   1.269 0.207517
## Deathrate                       -0.21271    0.09364  -2.272 0.025320 *
## Agriculture                      0.28546    0.12819   2.227 0.028276 *
## Region.Cen.E.Eur                -0.56139    0.21252  -2.642 0.009621 **
## Region.E.Asia                   -1.03765    0.28526  -3.638 0.000443 ***
## Region.LatCari                   0.51758    0.20763   2.493 0.014366 *
## Region.MENA                     -0.54141    0.22931  -2.361 0.020224 *
## Region.SE.Asia                  -0.33087    0.25449  -1.300 0.196637
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4876 on 97 degrees of freedom
## Multiple R-squared:  0.8574, Adjusted R-squared:  0.8265
## F-statistic: 27.77 on 21 and 97 DF,  p-value: < 2.2e-16
```

Now population density.

```
m14 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people + Arable.percent + Crops.percent +
Other.Land.Use.percent + Climate + Birthrate + Deathrate + Agriculture +
Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia , data = world_train)
summary(m14)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##      Gross.National.Income.per.Capita + Population + Coast.Area.Ratio +
##      Net.migration + Literacy.percent + Phones.per.1000.people +
##      Arable.percent + Crops.percent + Other.Land.Use.percent +
##      Climate + Birthrate + Deathrate + Agriculture + Region.Cen.E.Eur +
```

```
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia,
##     data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33617 -0.29617  0.02634  0.25829  1.19173
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.98571    0.35803  13.926  < 2e-16 ***
## Life.Expectancy.at.Birth         0.86737    0.52941   1.638 0.104549
## Mean.Years.of.Education          0.28754    0.15602   1.843 0.068349 .
## Gross.National.Income.per.Capita 0.64769    0.16000   4.048 0.000103 ***
## Population                       0.05650    0.05347   1.057 0.293230
## Coast.Area.Ratio                -0.07143    0.06275  -1.138 0.257732
## Net.migration                    0.06174    0.06336   0.974 0.332254
## Literacy.percent                 0.12764    0.13920   0.917 0.361420
## Phones.per.1000.people           0.20346    0.17023   1.195 0.234891
## Arable.percent                   0.21844    0.20239   1.079 0.283098
## Crops.percent                   -0.05074    0.07113  -0.713 0.477309
## Other.Land.Use.percent           0.29984    0.19595   1.530 0.129193
## Climate                          0.06707    0.06104   1.099 0.274577
## Birthrate                        0.20264    0.15869   1.277 0.204627
## Deathrate                       -0.21844    0.09238  -2.364 0.020029 *
## Agriculture                      0.27259    0.12442   2.191 0.030832 *
## Region.Cen.E.Eur                -0.56156    0.21166  -2.653 0.009303 **
## Region.E.Asia                   -1.00070    0.27195  -3.680 0.000382 ***
## Region.LatCari                   0.50172    0.20375   2.462 0.015544 *
## Region.MENA                     -0.54858    0.22782  -2.408 0.017912 *
## Region.SE.Asia                  -0.30387    0.24624  -1.234 0.220127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4856 on 98 degrees of freedom
## Multiple R-squared:  0.8571, Adjusted R-squared:  0.8279
## F-statistic: 29.39 on 20 and 98 DF,  p-value: < 2.2e-16
```

Now crops percent.

```
m15 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Coast.Area.Ratio + Net.migration + Literacy.percent +
Phones.per.1000.people + Arable.percent + Other.Land.Use.percent + Climate +
Birthrate + Deathrate + Agriculture + Region.Cen.E.Eur + Region.E.Asia +
Region.LatCari + Region.MENA + Region.SE.Asia , data = world_train)
summary(m15)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
```

```
Mean.Years.of.Education +
##      Gross.National.Income.per.Capita + Population + Coast.Area.Ratio +
##      Net.migration + Literacy.percent + Phones.per.1000.people +
##      Arable.percent + Other.Land.Use.percent + Climate + Birthrate +
##      Deathrate + Agriculture + Region.Cen.E.Eur + Region.E.Asia +
##      Region.LatCari + Region.MENA + Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -1.31083 -0.27889  0.02596  0.24973  1.19385
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.00343    0.35628  14.044  < 2e-16 ***
## Life.Expectancy.at.Birth         0.86757    0.52809   1.643  0.10359
## Mean.Years.of.Education          0.27976    0.15525   1.802  0.07459 .
## Gross.National.Income.per.Capita 0.65979    0.15870   4.157 6.85e-05 ***
## Population                       0.05548    0.05332   1.041  0.30063
## Coast.Area.Ratio                -0.07985    0.06147  -1.299  0.19698
## Net.migration                    0.06277    0.06319   0.993  0.32293
## Literacy.percent                 0.13385    0.13858   0.966  0.33644
## Phones.per.1000.people           0.19965    0.16972   1.176  0.24227
## Arable.percent                   0.21891    0.20189   1.084  0.28085
## Other.Land.Use.percent           0.33353    0.18971   1.758  0.08181 .
## Climate                          0.06322    0.06065   1.042  0.29977
## Birthrate                        0.19048    0.15738   1.210  0.22903
## Deathrate                       -0.21715    0.09214  -2.357  0.02040 *
## Agriculture                      0.27168    0.12411   2.189  0.03094 *
## Region.Cen.E.Eur                -0.58308    0.20897  -2.790  0.00632 **
## Region.E.Asia                   -1.02367    0.26936  -3.800  0.00025 ***
## Region.LatCari                   0.46596    0.19699   2.365  0.01996 *
## Region.MENA                     -0.58765    0.22059  -2.664  0.00902 **
## Region.SE.Asia                  -0.31977    0.24462  -1.307  0.19417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4844 on 99 degrees of freedom
## Multiple R-squared:  0.8563, Adjusted R-squared:  0.8288
## F-statistic: 31.06 on 19 and 99 DF,  p-value: < 2.2e-16
```

Now literacy percent.

```
m16 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Coast.Area.Ratio + Net.migration + Phones.per.1000.people +
Arable.percent + Other.Land.Use.percent + Climate + Birthrate + Deathrate +
Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA
+ Region.SE.Asia , data = world_train)
summary(m16)
```

```
##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##       Gross.National.Income.per.Capita + Population + Coast.Area.Ratio +
##       Net.migration + Phones.per.1000.people + Arable.percent +
##       Other.Land.Use.percent + Climate + Birthrate + Deathrate +
##       Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari +
##       Region.MENA + Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29058 -0.27354  0.04733  0.25906  1.14086
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       4.99400    0.35602  14.027  < 2e-16 ***
## Life.Expectancy.at.Birth          0.86106    0.52787   1.631 0.105995
## Mean.Years.of.Education           0.37850    0.11680   3.240 0.001621 **
## Gross.National.Income.per.Capita  0.64507    0.15791   4.085  8.9e-05 ***
## Population                        0.05272    0.05322   0.991 0.324300
## Coast.Area.Ratio                 -0.09038    0.06048  -1.494 0.138203
## Net.migration                     0.06118    0.06315   0.969 0.334952
## Phones.per.1000.people            0.23525    0.16562   1.420 0.158591
## Arable.percent                    0.20416    0.20124   1.015 0.312782
## Other.Land.Use.percent            0.32816    0.18956   1.731 0.086505 .
## Climate                           0.06669    0.06053   1.102 0.273153
## Birthrate                         0.17197    0.15615   1.101 0.273409
## Deathrate                        -0.21708    0.09211  -2.357 0.020382 *
## Agriculture                       0.28337    0.12347   2.295 0.023823 *
## Region.Cen.E.Eur                 -0.56553    0.20811  -2.717 0.007755 **
## Region.E.Asia                    -0.99194    0.26726  -3.711 0.000339 ***
## Region.LatCari                    0.50804    0.19205   2.645 0.009479 **
## Region.MENA                      -0.61083    0.21920  -2.787 0.006375 **
## Region.SE.Asia                   -0.28985    0.24257  -1.195 0.234935
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4843 on 100 degrees of freedom
## Multiple R-squared:  0.855,  Adjusted R-squared:  0.8289
## F-statistic: 32.76 on 18 and 100 DF,  p-value: < 2.2e-16
```

Now net migration.

```
m17 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita +
    Population + Coast.Area.Ratio + Phones.per.1000.people + Arable.percent +
Other.Land.Use.percent + Climate + Birthrate + Deathrate + Agriculture +
Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
```

```
Region.SE.Asia , data = world_train)
summary(m17)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Population + Coast.Area.Ratio +
##     Phones.per.1000.people + Arable.percent + Other.Land.Use.percent +
##     Climate + Birthrate + Deathrate + Agriculture + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia,
##     data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26841 -0.23479  0.04136  0.27042  1.16507
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        5.01889    0.35499  14.138  < 2e-16 ***
## Life.Expectancy.at.Birth           0.87063    0.52762   1.650 0.102026
## Mean.Years.of.Education            0.36411    0.11582   3.144 0.002191 **
## Gross.National.Income.per.Capita   0.67966    0.15378   4.420 2.49e-05 ***
## Population                         0.04881    0.05305   0.920 0.359786
## Coast.Area.Ratio                  -0.09653    0.06013  -1.605 0.111526
## Phones.per.1000.people             0.24829    0.16502   1.505 0.135543
## Arable.percent                     0.22761    0.19972   1.140 0.257129
## Other.Land.Use.percent             0.35109    0.18802   1.867 0.064757 .
## Climate                            0.06103    0.06022   1.013 0.313308
## Birthrate                          0.16215    0.15578   1.041 0.300408
## Deathrate                         -0.21292    0.09198  -2.315 0.022645 *
## Agriculture                        0.29333    0.12301   2.385 0.018962 *
## Region.Cen.E.Eur                  -0.64072    0.19305  -3.319 0.001257 **
## Region.E.Asia                     -1.03115    0.26410  -3.904 0.000171 ***
## Region.LatCari                     0.43426    0.17626   2.464 0.015436 *
## Region.MENA                       -0.64273    0.21665  -2.967 0.003758 **
## Region.SE.Asia                    -0.29158    0.24249  -1.202 0.231994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4841 on 101 degrees of freedom
## Multiple R-squared:  0.8536, Adjusted R-squared:  0.829
## F-statistic: 34.65 on 17 and 101 DF,  p-value: < 2.2e-16
```

Now population.

```
m18 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Phones.per.1000.people + Arable.percent +
Other.Land.Use.percent + Climate + Birthrate + Deathrate + Agriculture +
```

```
Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA +
Region.SE.Asia , data = world_train)
summary(m18)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio +
Phones.per.1000.people +
##     Arable.percent + Other.Land.Use.percent + Climate + Birthrate +
##     Deathrate + Agriculture + Region.Cen.E.Eur + Region.E.Asia +
##     Region.LatCari + Region.MENA + Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29592 -0.23213  0.01639  0.28555  1.13808
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       5.06220    0.35159  14.398  < 2e-16 ***
## Life.Expectancy.at.Birth          0.80358    0.52217   1.539 0.126916
## Mean.Years.of.Education           0.36554    0.11572   3.159 0.002085 **
## Gross.National.Income.per.Capita  0.67019    0.15332   4.371 2.98e-05 ***
## Coast.Area.Ratio                 -0.10199    0.05979  -1.706 0.091066 .
## Phones.per.1000.people            0.25508    0.16473   1.549 0.124598
## Arable.percent                    0.31017    0.17829   1.740 0.084923 .
## Other.Land.Use.percent            0.41792    0.17329   2.412 0.017667 *
## Climate                           0.06230    0.06016   1.036 0.302839
## Birthrate                         0.15459    0.15544   0.994 0.322338
## Deathrate                        -0.22277    0.09128  -2.440 0.016392 *
## Agriculture                       0.28944    0.12284   2.356 0.020376 *
## Region.Cen.E.Eur                 -0.67731    0.18876  -3.588 0.000514 ***
## Region.E.Asia                    -0.98780    0.25967  -3.804 0.000243 ***
## Region.LatCari                    0.42435    0.17579   2.414 0.017564 *
## Region.MENA                      -0.62850    0.21593  -2.911 0.004430 **
## Region.SE.Asia                   -0.24908    0.23787  -1.047 0.297504
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4837 on 102 degrees of freedom
## Multiple R-squared:  0.8524, Adjusted R-squared:  0.8293
## F-statistic: 36.82 on 16 and 102 DF,  p-value: < 2.2e-16
```

Now birthrate.

```
m19 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Phones.per.1000.people + Arable.percent +
Other.Land.Use.percent + Climate + Deathrate + Agriculture + Region.Cen.E.Eur
```

```
+ Region.E.Asia + Region.LatCari + Region.MENA + Region.SE.Asia , data =
world_train)
summary(m19)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio +
Phones.per.1000.people +
##     Arable.percent + Other.Land.Use.percent + Climate + Deathrate +
##     Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari +
##     Region.MENA + Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29865 -0.25516  0.01912  0.29470  1.15998
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        5.18975    0.32734  15.854  < 2e-16 ***
## Life.Expectancy.at.Birth           0.64258    0.49641   1.294  0.19840
## Mean.Years.of.Education            0.37130    0.11557   3.213  0.00176 **
## Gross.National.Income.per.Capita   0.65372    0.15241   4.289 4.05e-05 ***
## Coast.Area.Ratio                  -0.11697    0.05786  -2.022  0.04579 *
## Phones.per.1000.people             0.18562    0.14918   1.244  0.21623
## Arable.percent                     0.27444    0.17462   1.572  0.11910
## Other.Land.Use.percent             0.39120    0.17119   2.285  0.02435 *
## Climate                            0.05166    0.05920   0.873  0.38491
## Deathrate                         -0.24961    0.08720  -2.862  0.00509 **
## Agriculture                        0.30515    0.12182   2.505  0.01381 *
## Region.Cen.E.Eur                  -0.77375    0.16193  -4.778 5.89e-06 ***
## Region.E.Asia                     -1.06841    0.24668  -4.331 3.45e-05 ***
## Region.LatCari                     0.42546    0.17578   2.420  0.01725 *
## Region.MENA                       -0.62439    0.21588  -2.892  0.00467 **
## Region.SE.Asia                    -0.29787    0.23274  -1.280  0.20348
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4837 on 103 degrees of freedom
## Multiple R-squared:  0.851,  Adjusted R-squared:  0.8293
## F-statistic: 39.21 on 15 and 103 DF,  p-value: < 2.2e-16
```

Now climate.

```
m20 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + 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 + Region.SE.Asia , data =
```

```
world_train)
summary(m20)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     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 +
##     Region.SE.Asia, data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.34960 -0.25885  0.01643  0.29905  1.11448
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        5.12187    0.31760  16.127  < 2e-16 ***
## Life.Expectancy.at.Birth           0.75611    0.47850   1.580  0.11711
## Mean.Years.of.Education            0.36734    0.11535   3.185  0.00191 **
## Gross.National.Income.per.Capita   0.65370    0.15224   4.294 3.95e-05 ***
## Coast.Area.Ratio                  -0.11348    0.05765  -1.968  0.05169 .
## Phones.per.1000.people             0.19374    0.14872   1.303  0.19554
## Arable.percent                     0.28427    0.17406   1.633  0.10545
## Other.Land.Use.percent             0.38913    0.17097   2.276  0.02490 *
## Deathrate                         -0.23584    0.08566  -2.753  0.00697 **
## Agriculture                        0.31613    0.12102   2.612  0.01033 *
## Region.Cen.E.Eur                  -0.76700    0.16156  -4.747 6.61e-06 ***
## Region.E.Asia                     -1.08976    0.24518  -4.445 2.21e-05 ***
## Region.LatCari                     0.43019    0.17550   2.451  0.01590 *
## Region.MENA                       -0.67065    0.20904  -3.208  0.00178 **
## Region.SE.Asia                    -0.29873    0.23247  -1.285  0.20164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4832 on 104 degrees of freedom
## Multiple R-squared:  0.8499, Adjusted R-squared:  0.8297
## F-statistic: 42.05 on 14 and 104 DF,  p-value: < 2.2e-16
```

Now region SE Asia.

```
m21 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + 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)
summary(m21)
```

```
## 
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##      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)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36828 -0.26870  0.02221  0.29742  1.08878
## 
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        5.08985    0.31760  16.026  < 2e-16 ***
## Life.Expectancy.at.Birth           0.72801    0.47948   1.518  0.13194
## Mean.Years.of.Education            0.36368    0.11567   3.144  0.00217 **
## Gross.National.Income.per.Capita   0.64517    0.15256   4.229 5.03e-05 ***
## Coast.Area.Ratio                  -0.12110    0.05752  -2.105  0.03765 *
## Phones.per.1000.people             0.21620    0.14814   1.459  0.14743
## Arable.percent                     0.30372    0.17393   1.746  0.08370 .
## Other.Land.Use.percent             0.40525    0.17104   2.369  0.01965 *
## Deathrate                         -0.20370    0.08218  -2.479  0.01478 *
## Agriculture                        0.30848    0.12125   2.544  0.01241 *
## Region.Cen.E.Eur                  -0.73897    0.16058  -4.602 1.18e-05 ***
## Region.E.Asia                     -1.02878    0.24129  -4.264 4.41e-05 ***
## Region.LatCari                     0.50511    0.16604   3.042  0.00297 **
## Region.MENA                       -0.57613    0.19627  -2.935  0.00409 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4847 on 105 degrees of freedom
## Multiple R-squared:  0.8475, Adjusted R-squared:  0.8286
## F-statistic: 44.88 on 13 and 105 DF,  p-value: < 2.2e-16
```

Now phones per 100 people.

```
m22 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Arable.percent + Other.Land.Use.percent + Deathrate +
Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA
, data = world_train)
summary(m22)
```

```
## 
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
```

```
##     Gross.National.Income.per.Capita + Coast.Area.Ratio + Arable.percent +
##     Other.Land.Use.percent + Deathrate + Agriculture + Region.Cen.E.Eur +
##     Region.E.Asia + Region.LatCari + Region.MENA, data = world_train)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -1.20743 -0.27625  0.02846  0.27603  1.15114
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.86303    0.27845  17.465  < 2e-16 ***
## Life.Expectancy.at.Birth         1.07916    0.41695   2.588  0.01100 *
## Mean.Years.of.Education          0.43316    0.10598   4.087 8.51e-05 ***
## Gross.National.Income.per.Capita 0.70160    0.14837   4.729 6.99e-06 ***
## Coast.Area.Ratio                -0.11445    0.05765  -1.985  0.04969 *
## Arable.percent                   0.28262    0.17425   1.622  0.10780
## Other.Land.Use.percent           0.36690    0.16991   2.159  0.03307 *
## Deathrate                       -0.17801    0.08070  -2.206  0.02956 *
## Agriculture                      0.29100    0.12130   2.399  0.01819 *
## Region.Cen.E.Eur                -0.74166    0.16142  -4.594 1.20e-05 ***
## Region.E.Asia                   -1.00757    0.24213  -4.161 6.46e-05 ***
## Region.LatCari                   0.53560    0.16560   3.234  0.00163 **
## Region.MENA                     -0.52540    0.19420  -2.706  0.00795 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4872 on 106 degrees of freedom
## Multiple R-squared:  0.8444, Adjusted R-squared:  0.8268
## F-statistic: 47.94 on 12 and 106 DF,  p-value: < 2.2e-16
```

Now arable percent.

```
m23 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Other.Land.Use.percent + Deathrate + Agriculture +
Region.Cen.E.Eur + Region.E.Asia + Region.LatCari + Region.MENA , data =
world_train)
summary(m23)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio +
Other.Land.Use.percent +
##     Deathrate + Agriculture + Region.Cen.E.Eur + Region.E.Asia +
##     Region.LatCari + Region.MENA, data = world_train)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.25456 -0.27487  0.02588  0.28051  1.26819
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.75065    0.27174  17.483  < 2e-16 ***
## Life.Expectancy.at.Birth         1.24981    0.40652   3.074 0.002677 **
## Mean.Years.of.Education          0.44189    0.10665   4.144 6.86e-05 ***
## Gross.National.Income.per.Capita 0.65838    0.14706   4.477 1.90e-05 ***
## Coast.Area.Ratio                -0.13434    0.05675  -2.367 0.019726 *
## Other.Land.Use.percent           0.10440    0.05211   2.004 0.047638 *
## Deathrate                       -0.15191    0.07968  -1.906 0.059281 .
## Agriculture                      0.27259    0.12168   2.240 0.027149 *
## Region.Cen.E.Eur                -0.72399    0.16228  -4.461 2.02e-05 ***
## Region.E.Asia                   -0.97858    0.24331  -4.022 0.000108 ***
## Region.LatCari                   0.53642    0.16685   3.215 0.001725 **
## Region.MENA                     -0.51077    0.19546  -2.613 0.010262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4909 on 107 degrees of freedom
## Multiple R-squared:  0.8405, Adjusted R-squared:  0.8241
## F-statistic: 51.27 on 11 and 107 DF,  p-value: < 2.2e-16
```

Now deathrate.

```
m24 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Other.Land.Use.percent + Agriculture +Region.Cen.E.Eur +
Region.E.Asia + Region.LatCari + Region.MENA , data = world_train)
summary(m24)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##     Gross.National.Income.per.Capita + Coast.Area.Ratio +
Other.Land.Use.percent +
##     Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari +
##     Region.MENA, data = world_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.45197 -0.28418  0.04341  0.31772  1.16125
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.46770    0.23038  19.393  < 2e-16 ***
## Life.Expectancy.at.Birth         1.59587    0.36815   4.335 3.29e-05 ***
## Mean.Years.of.Education          0.41732    0.10715   3.895 0.000171 ***
## Gross.National.Income.per.Capita 0.69501    0.14757   4.710 7.41e-06 ***
```

```
## Coast.Area.Ratio                      -0.12445     0.05720   -2.176 0.031760 *
## Other.Land.Use.percent                 0.10788     0.05271    2.047 0.043107 *
## Agriculture                            0.31704     0.12088    2.623 0.009982 **
## Region.Cen.E.Eur                       -0.69955     0.16373   -4.273 4.18e-05 ***
## Region.E.Asia                          -0.90420     0.24307   -3.720 0.000318 ***
## Region.LatCari                         0.69002     0.14788    4.666 8.86e-06 ***
## Region.MENA                            -0.28838     0.15873   -1.817 0.072030 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4969 on 108 degrees of freedom
## Multiple R-squared:  0.8351, Adjusted R-squared:  0.8199
## F-statistic:  54.7 on 10 and 108 DF,  p-value: < 2.2e-16
```

Now region Middles East and North Africa.

```
m25 <- lm(Happiness.Score ~ Life.Expectancy.at.Birth +
    Mean.Years.of.Education + Gross.National.Income.per.Capita+
Coast.Area.Ratio + Other.Land.Use.percent + Agriculture +Region.Cen.E.Eur +
Region.E.Asia + Region.LatCari , data = world_train)
summary(m25)

##
## Call:
## lm(formula = Happiness.Score ~ Life.Expectancy.at.Birth +
Mean.Years.of.Education +
##      Gross.National.Income.per.Capita + Coast.Area.Ratio +
Other.Land.Use.percent +
##      Agriculture + Region.Cen.E.Eur + Region.E.Asia + Region.LatCari,
##      data = world_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40229 -0.29328  0.04516  0.34251  1.18961
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          4.48501    0.23260  19.282  < 2e-16 ***
## Life.Expectancy.at.Birth             1.48464    0.36683   4.047 9.72e-05 ***
## Mean.Years.of.Education              0.48552    0.10141   4.788 5.34e-06 ***
## Gross.National.Income.per.Capita     0.66278    0.14803   4.477 1.87e-05 ***
## Coast.Area.Ratio                    -0.13087    0.05769  -2.268 0.025269 *
## Other.Land.Use.percent               0.09413    0.05271   1.786 0.076896 .
## Agriculture                          0.33270    0.12184   2.731 0.007372 **
## Region.Cen.E.Eur                    -0.70825    0.16538  -4.283 3.99e-05 ***
## Region.E.Asia                       -0.82640    0.24178  -3.418 0.000888 ***
## Region.LatCari                       0.76367    0.14371   5.314 5.74e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.5021 on 109 degrees of freedom
## Multiple R-squared:  0.8301, Adjusted R-squared:  0.8161
## F-statistic: 59.16 on 9 and 109 DF,  p-value: < 2.2e-16

m.mlreg <- lm(Happiness.Score ~ 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 , data = world_train)
summary(m.mlreg)

##
## Call:
## lm(formula = Happiness.Score ~ 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, 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.Education +
##      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.

```
#model summary again
summary(m.mlreg)

##
## Call:
## lm(formula = Happiness.Score ~ 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, 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
```

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.

```
m.mlreg

##
## Call:
## lm(formula = Happiness.Score ~ 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, 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.

```
plot(m.mlreg)
```
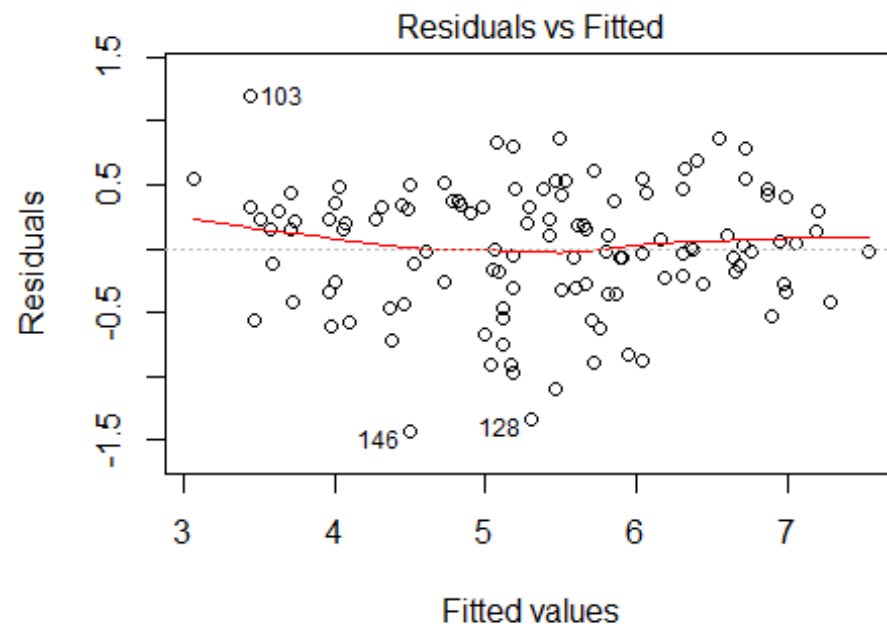
Residuals vs Fitted

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

Normal Q-Q

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

Scale-Location

√|Standardized residuals|

146○ 128○
○103
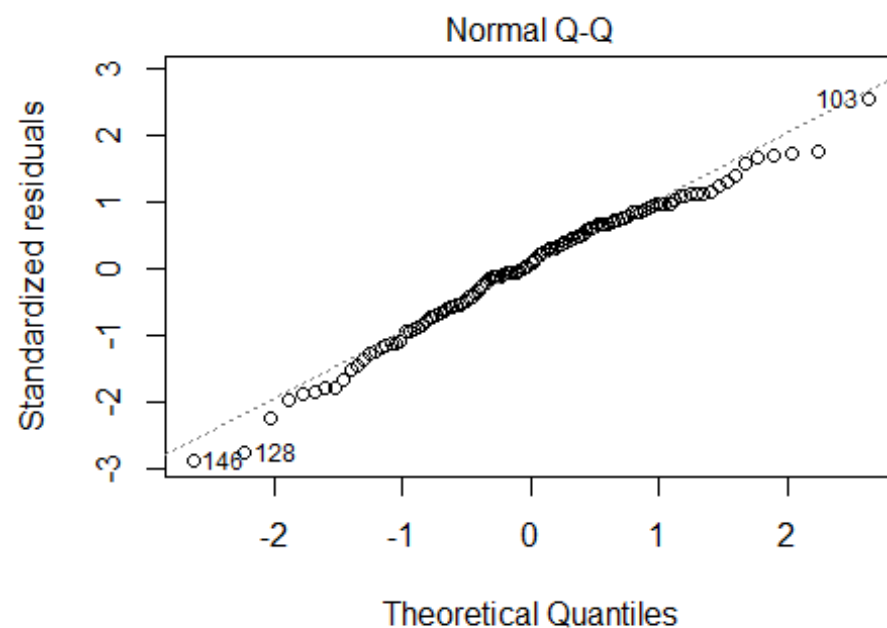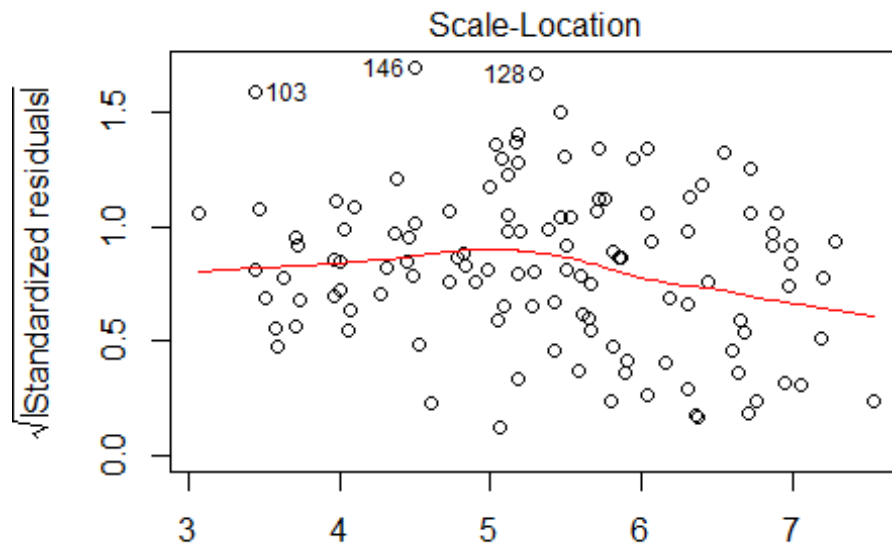
Fitted values
Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Educatic

Residuals vs Leverage

Standardized residuals

103○

Cook's distance

0.5

Leverage
Happiness.Score ~ Life.Expectancy.at.Birth + Mean.Years.of.Educatic
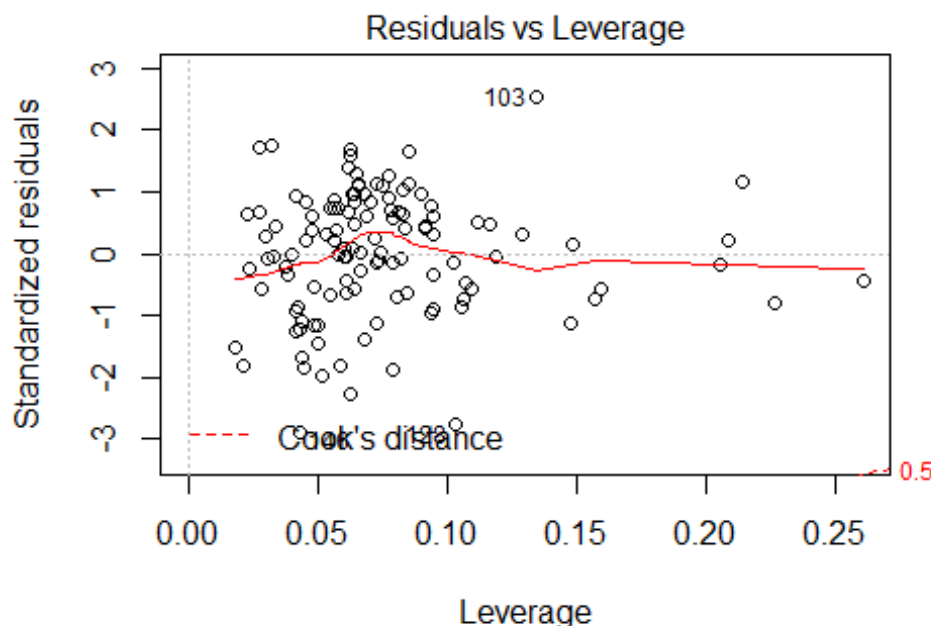
Residuals vs. fitted: This helps us to detect non-linearity if the line is parabolic. Our line is a bit curved, but not entirely. This suggests there may be some linear regression features that could be described in a non-linear fashion. Also, all the labeled points are considered outliers, so this plot 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.

```
#calculate RMSE by squaring the residuals to make them positive then taking
the square root
RMSE <- sqrt(mean((m.mlreg$residuals)^2))
RMSE

## [1] 0.4875172
```

The RMSE of this model is 0.49. 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 0.487 for happiness score.

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

```
#calculate the MAE
mean(abs(m.mlreg$residuals))

## [1] 0.389752
```

This is saying that on average the predicted value from the regression model is 0.39 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.

```
pred.mlreg <- predict(m.mlreg, world_test)

#explore relationship
cor(pred.mlreg, world_test$Happiness.Score)

## [1] 0.8543146
```

```
plot(pred.mlreg, world_test$Happiness.Score)
```
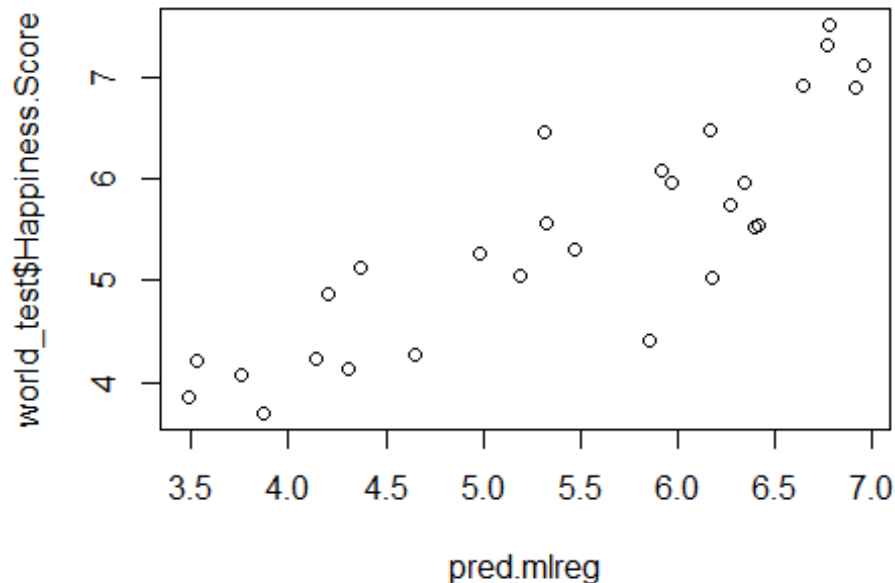

85% correlation is pretty good! I will determine the best model later on. Let's look at our confidence interval for each prediction.

```
#Check prediction intervals
pred_intervals <- predict(m.mlreg, world_test, interval = "confidence")
head(pred_intervals)
```

```
##           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

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

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.

```
set.seed(12)
# y = happiness score bc it is the vector of outcomes
#80% for training, 20% for testing
indxTrain2 <- createDataPartition(y = world_df$Happiness.Score, p = 0.80,
list = FALSE)

#80% of the full dataset goes to training and the rest to testing. the [-]
syntax places all not-yet indexed values to the remaining set.
world_train2 <- world_df[indxTrain2,]
world_test2 <- world_df[-indxTrain2,]

#make sure distributions are similar in traing and test
summary(world_test2$Happiness.Score)

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

summary(world_train2$Happiness.Score)

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

#use rpart to train reg tree model
m.regtree <- rpart(Happiness.Score ~ Region + 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, data = world_train2)
#print it out
m.regtree

## 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
Africa,Southeastern Asia,Southern Asia 17    3.243062 4.939588 *
##         13) Region=Eastern Asia,Latin America and Caribbean 7    2.054122
5.687429 *
##       7) Agriculture< 0.1065 52   27.382740 6.362538
##         14) Region=Central and Eastern Europe,Eastern Asia,Sub-Saharan
Africa 15    3.540084 5.628467 *
##         15) Region=Australia and New Zealand,Latin America and
Caribbean,Middle East and Northern Africa,North America,Southeastern
Asia,Western Europe 37   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.

Now let's visualize the tree.

```
#install.packages("rpart.plot")
library(rpart.plot)

#Create visual
rpart.plot(m.regtree, digits = 3, tweak = 1.8)
```
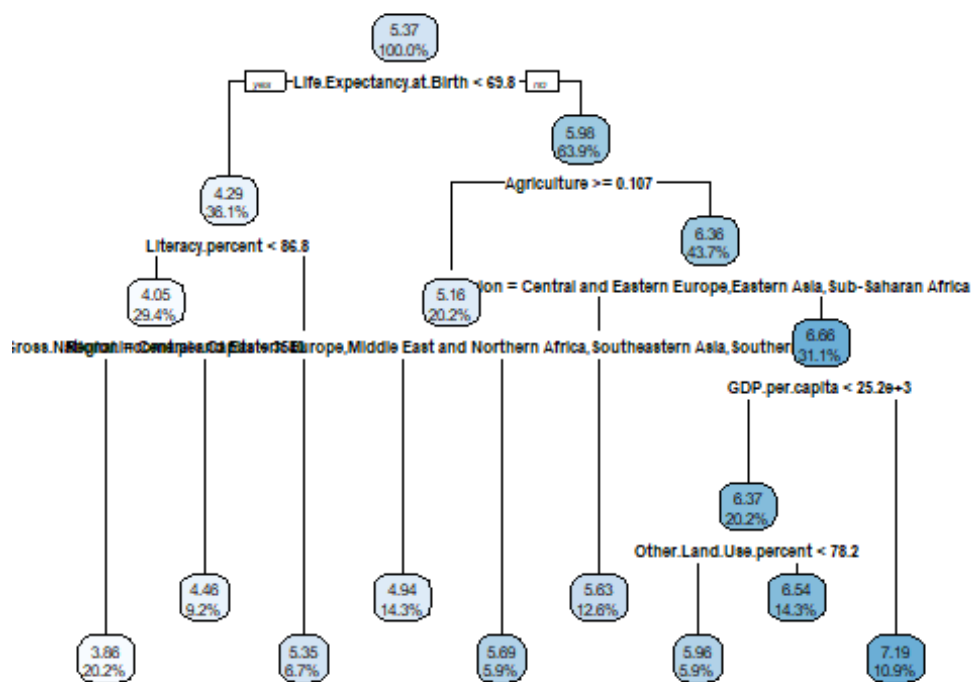
It's a little hard to read this, but it is essentially the same tree structure, just visualized.

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.

```
#prediction on validation data
pred.regtree <- predict(m.regtree, world_test2)

#check it out - quick comparison
summary(pred.regtree)

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

summary(world_test2$Happiness.Score)

##    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.

```
#explore relationship
cor(pred.regtree, world_test2$Happiness.Score)

## [1] 0.8771465

plot(pred.regtree, world_test2$Happiness.Score)
```
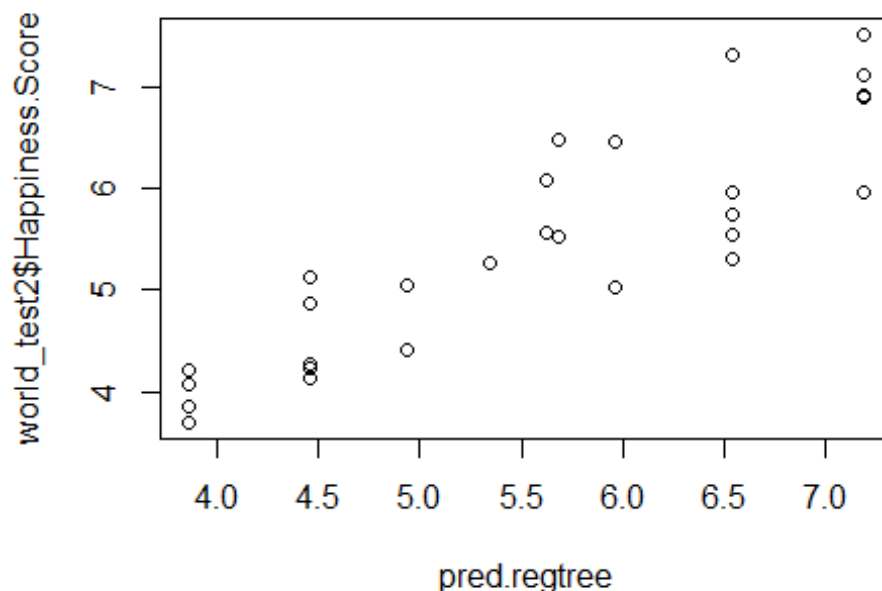
Correlation of .877 is pretty strong! It gives us some idea that our model predicitons aren't too far off actual values.

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

```
#create a function that takes in actual and predicted values and calculates
the MAE
MAE <- function(actual, predicted) {
mean(abs(actual - predicted))
}
```

Let's try it out!

```
#Calculate MAE
MAE(pred.regtree, world_test2$Happiness.Score)
```

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

The MAE is .454. 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 .454 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?

```
#avg of happiness
avg <- mean(world_test2$Happiness.Score)
```

```
#check it out
MAE(avg, world_test2$Happiness.Score)

## [1] 0.9147143
```

The MAE in this instance is .915, 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.

```
#install.packages("RWeka")
library(RWeka)
m.regtree2 <- M5P(Happiness.Score ~ Region + 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, data = world_train2,
control=Weka_control(R=TRUE))
```

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,Australia and New Zealand <= 0.5 : LM3 (37/55.92%)
## |   |   Region=Latin America and Caribbean,Western Europe,North
America,Australia and New Zealand >  0.5 : LM4 (12/48.537%)
## |   GDP.per.capita >  8950 :
## |   |   Region=Latin America and Caribbean,Western Europe,North
America,Australia 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,Australia 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.

```
#predict on validation data
pred.regtree2 <- predict(m.regtree2, world_test2)
```

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

```
summary(pred.regtree2)
```

```
##    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.

```
#check correlation
cor(pred.regtree2, world_test2$Happiness.Score)
```

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

The correlation is lower now as a result.

```
#check MAE
MAE(world_test2$Happiness.Score, pred.regtree2)

## [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.

```
world_standard <- world_df

#add dummy coded region columns
world_standard <- cbind(world_standard,region_vars[,-10])
#remove unused columns
world_standard <- world_standard[-c(1,2,4,10)]
```

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.

```
#normalized
summary(world_standard$Happiness.Score)

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

#original
summary(world_df$Happiness.Score)

##    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.

```
set.seed(12)
# y = happiness score bc it is the vector of outcomes
#80% for training, 20% for testing
indxTrain3 <- createDataPartition(y = world_standard$Happiness.Score, p =
0.80, list = FALSE)

#80% of the full dataset goes to training and the rest to testing. the [-]
syntax places all not-yet indexed values to the remaining set.
world_train3 <- world_standard[indxTrain3,]
world_test3 <- world_standard[-indxTrain3,]
```

```
#make sure distributions are similar in traing and test
summary(world_test3$Happiness.Score)

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

summary(world_train3$Happiness.Score)

##    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.

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

##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':
##
##      compute
```

Now let's train our model.

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

```
#prepare the list of predictor names for multiple regression
var_names2 <- names(world_standard[2:31])
formula2 <- as.formula(paste('Happiness.Score ~ '
,paste(var_names2,collapse='+')))

#make sure it worked
formula2

## 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.America.and.Caribbean +
##      RegionMiddle.East.and.Northern.Africa + RegionNorth.America +
```

```
##       RegionSoutheastern.Asia + RegionSouthern.Asia +
RegionSub.Saharan.Africa

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

Let's visualize the model to help our understanding.

```
plot(m.nn)
```

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

Now let's evaluate the model performance.

```
m.nn_results <- compute(m.nn, world_test3[2:31])

#see results
pred.nn <- m.nn_results$net.result
```

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.

```
#find correlation
cor(pred.nn, world_test3$Happiness.Score)

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

.86 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.

```
set.seed(12)
#new model with hidden layer having 5 nodes
m.nn2 <- neuralnet(formula2, data = world_train3, hidden = 5)

#visualize model
plot(m.nn2)
```

Now let's see if it actually did improve.

```
m.nn2_results <- compute(m.nn2, world_test3[2:31])

#see results
pred.nn2 <- m.nn2_results$net.result

#see correlation
cor(pred.nn2, world_test3$Happiness.Score)

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

Adding 5 hidden nodes actually made the prediction performance go down. 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.

```
#remove happiness score column
world_train4 <- world_train3[-1]
world_test4 <- world_test3[-1]
```

I am choosing to start with k = 11 because that is 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).

```
#Make vector of all test prices
train3.scores <- world_train3[,1]
test3.scores <- world_test3[,1]
#this trains the knnreg function on the training data set
pred.knnreg <- knnregTrain(train = world_train4, test = world_test4, y =
train3.scores, k= 11)

#initial comparison between our model and the test data fram prices
head(cbind(test3.scores, pred.knnreg))

##      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).

```
#create a function that takes in actual and predicted values and calculates
the MAE
MAE <- function(actual, predicted) {
mean(abs(actual - predicted))
}

#create a function that takes in actual and predicted values and calculates
the RMSE
RMSE <- function(actual, predicted) {
sqrt(mean(((actual - predicted)^2)))
}
```

```r
## Automating 10-fold CV for each model using lapply() ----
#multiple linear regression
set.seed(12)

#create folds for cv
folds <- createFolds(world_norm_dist2$Happiness.Score, k = 10)

#first apply the model to predict for each fold, then apply the error
calculations
cv_mlreg <- lapply(folds, function(x) {
  nations_train <- world_norm_dist2[-x, ]

  nations_test <- world_norm_dist2[x, ]

  nations_model <- lm(Happiness.Score ~ 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 , data = nations_train)

  nations_pred <- predict(nations_model, nations_test)
  nations_actual <- nations_test$Happiness.Score

  MAE <- MAE(nations_actual, nations_pred)
  RMSE <- RMSE(nations_actual, nations_pred)


  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
cv_mlreg <- as.data.frame(t(as.data.frame(cv_mlreg)))
cv_mlreg

##               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

#average MAE
mae_mlreg <- mean(cv_mlreg$MAE)
mae_mlreg
```

```
## [1] 0.4291805

#average RMSE
rmse_mlreg <- mean(cv_mlreg$RMSE)
rmse_mlreg

## [1] 0.5324223
```

The average MAE between the 10 folds for the Multiple Linear Regression Model is 0.429 and the average RMSE is 0.532.

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

```
## Automating 10-fold CV for each model using lapply() ----
#regression tree model 1
set.seed(12)

#create folds for cv
folds2 <- createFolds(world_df$Happiness.Score, k = 10)

#first apply the model to predict for each fold, then apply the error
calculations
cv_regtree <- lapply(folds2, function(x) {
  nations_train <- world_df[-x, ]

  nations_test <- world_df[x, ]

  nations_model <- rpart(Happiness.Score ~ Region + 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, data = nations_train)

  nations_pred <- predict(nations_model, nations_test)
  nations_actual <- nations_test$Happiness.Score

  MAE <- MAE(nations_actual, nations_pred)
  RMSE <- RMSE(nations_actual, nations_pred)


  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
cv_regtree <- as.data.frame(t(as.data.frame(cv_regtree)))
cv_regtree
```

```
##               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

#average MAE
mae_regtree <- mean(cv_regtree$MAE)
mae_regtree

## [1] 0.5551525

#average RMSE
rmse_regtree <-mean(cv_regtree$RMSE)
rmse_regtree

## [1] 0.6727711
```

The average MAE between the 10 folds for the Regression Tree Model is 0.555 and the average RMSE is 0.673.

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

```
## Automating 10-fold CV for each model using lapply() ----
#regression tree model 2
set.seed(12)

#first apply the model to predict for each fold, then apply the error calculations
cv_regtree2 <- lapply(folds2, function(x) {
  nations_train <- world_df[-x, ]

  nations_test <- world_df[x, ]

  nations_model <- M5P(Happiness.Score ~ Region + 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, data = nations_train,
control=Weka_control(R=TRUE))

  nations_pred <- predict(nations_model, nations_test)
```

```r
  nations_actual <- nations_test$Happiness.Score

  MAE <- MAE(nations_actual, nations_pred)
  RMSE <- RMSE(nations_actual, nations_pred)


  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
cv_regtree2 <- as.data.frame(t(as.data.frame(cv_regtree2)))
cv_regtree2

##               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

#average MAE
mae_regtree2 <- mean(cv_regtree2$MAE)
mae_regtree2

## [1] 0.5860661

#average RMSE
rmse_regtree2 <- mean(cv_regtree2$RMSE)
rmse_regtree2

## [1] 0.7223334
```

The average MAE between the 10 folds for the Regression Tree Model is 0.586 and the average RMSE is 0.722.

Now let's look at the first neural network.

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

```r
#unnormalize columns with reverse min-max normalization by creating a
function that takes in two arguments "y"  and "x". y is the normalized value
and x is the original vector.
#Then, it unnormalizes previously between 0-1 values
unnormalize <- function(y, x) {
```

```r
  return((y*(diff(range(x))))+min(x))
}

## Automating 10-fold CV for each model using lapply() ----
#neural network model 1
set.seed(12)

#create folds for cv
folds3 <- createFolds(world_standard$Happiness.Score, k = 10)

#first apply the model to predict for each fold, then apply the error
calculations
cv_nn <- lapply(folds3, function(x) {
  nations_train <- world_standard[-x, ]

  nations_test <- world_standard[x, ]

  nations_model <- neuralnet(formula2, data = nations_train)
  #use compute for neuralnet package
  nations_results <- compute(nations_model, nations_test[2:31])

  nations_pred <- nations_results$net.result
  nations_actual <- nations_test$Happiness.Score

  #remember that predictions were normalized
  MAE <-
MAE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations_
pred,world_df$Happiness.Score))
  RMSE <-
RMSE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations
_pred,world_df$Happiness.Score))

  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
cv_nn <- as.data.frame(t(as.data.frame(cv_nn)))
cv_nn

##                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
```

```
#average MAE
mae_nn <- mean(cv_nn$MAE)
mae_nn
```

```
## [1] 0.583536
```

```
#average RMSE
rmse_nn <-mean(cv_nn$RMSE)
rmse_nn
```

```
## [1] 0.7115564
```

This neural network model gives an average MAE of 0.584 and an average RMSE of 0.711.

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

```
## Automating 10-fold CV for each model using lapply() ----
#neural network model 2
set.seed(12)

#first apply the model to predict for each fold, then apply the error
calculations
cv_nn2 <- lapply(folds3, function(x) {
  nations_train <- world_standard[-x, ]

  nations_test <- world_standard[x, ]

  nations_model <- neuralnet(formula2, data = nations_train, hidden = 5)

  nations_results <- compute(nations_model, nations_test[2:31])

  nations_pred <- nations_results$net.result
  nations_actual <- nations_test$Happiness.Score

  MAE <-
MAE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations_
pred,world_df$Happiness.Score))
  RMSE <-
RMSE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations
_pred,world_df$Happiness.Score))


  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
```

```
cv_nn2 <- as.data.frame(t(as.data.frame(cv_nn2)))
cv_nn2
```

```
##               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
```

```
#average MAE
mae_nn2 <- mean(cv_nn2$MAE)
mae_nn2
```

```
## [1] 0.8630703
```

```
#average RMSE
rmse_nn2 <-mean(cv_nn2$RMSE)
rmse_nn2
```

```
## [1] 1.152242
```

This neural network model gives an average MAE of 0.863 and an average RMSE of 1.15.

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

```
## Automating 10-fold CV for each model using lapply() ----
#knn regression model
set.seed(12)

#first apply the model to predict for each fold, then apply the error
calculations
cv_knnreg <- lapply(folds3, function(x) {
  nations_train <- world_standard[-x, ]

  nations_test <- world_standard[x, ]

  y <- nations_train[,1]

  nations_pred <- knnregTrain(train = nations_train[-1], test =
nations_test[-1], y = y, k= 11)

  nations_actual <- nations_test$Happiness.Score

  MAE <-
MAE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations_
```

```
pred,world_df$Happiness.Score))
  RMSE <-
RMSE(unnormalize(nations_actual,world_df$Happiness.Score),unnormalize(nations
_pred,world_df$Happiness.Score))


  error <- rbind("MAE" = MAE, "RMSE" = RMSE)
  return(error)
})

#print out the MAE and RMSE for every fold
cv_knnreg <- as.data.frame(t(as.data.frame(cv_knnreg)))
cv_knnreg

##                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

#average MAE
mae_knnreg <- mean(cv_knnreg$MAE)
mae_knnreg

## [1] 0.5292926

#average RMSE
rmse_knnreg <- mean(cv_knnreg$RMSE)
rmse_knnreg

## [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 0.529 and the average RMSE is 0.664.

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

```
#store all MAE values in a vector and bind them together
MAE.Values <- rbind(mae_mlreg, mae_regtree, mae_regtree2, mae_nn, mae_nn2,
mae_knnreg)
#store all RMSE values in a vector and bind them together
RMSE.Values <- rbind(rmse_mlreg, rmse_regtree, rmse_regtree2, rmse_nn,
rmse_nn2, rmse_knnreg)
```

```r
#format final dataframe
model_comparison <- as.data.frame(cbind(MAE.Values, RMSE.Values)) %>%
rename(MAE = "V1", RMSE = "V2")

#change row names
row.names(model_comparison) <- c("Multiple Linear Regression", "Regression
Tree", "Linear Model Regression Tree", "Neural Network 1 hidden node",
"Neural Network 5 hidden nodes", "kNN Regression")

#print it out
model_comparison
```

```
##                                      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
```

```r
#reformat dataframe in order to visualize
viz <- as.data.frame(rbind(MAE.Values, RMSE.Values)) %>% rename(Value = "V1")
viz$Model <- c("Multiple Linear Regression", "Regression Tree", "Linear Model
Regression Tree", "Neural Network 1 hidden node", "Neural Network 5 hidden
nodes", "kNN Regression", "Multiple Linear Regression", "Regression Tree",
"Linear Model Regression Tree", "Neural Network 1 hidden node", "Neural
Network 5 hidden nodes", "kNN Regression")
viz$Metric[1:6] <- "MAE"
viz$Metric[7:12] <- "RMSE"

#make heatmap of error metrics for each model
ggplot(data = viz, aes(x = Metric, y = Model)) + geom_tile(aes(fill = Value))
+ scale_fill_gradient(low = "#77dd77", high = "#fdfd96") +
geom_text(aes(label = round(Value,2)))
```
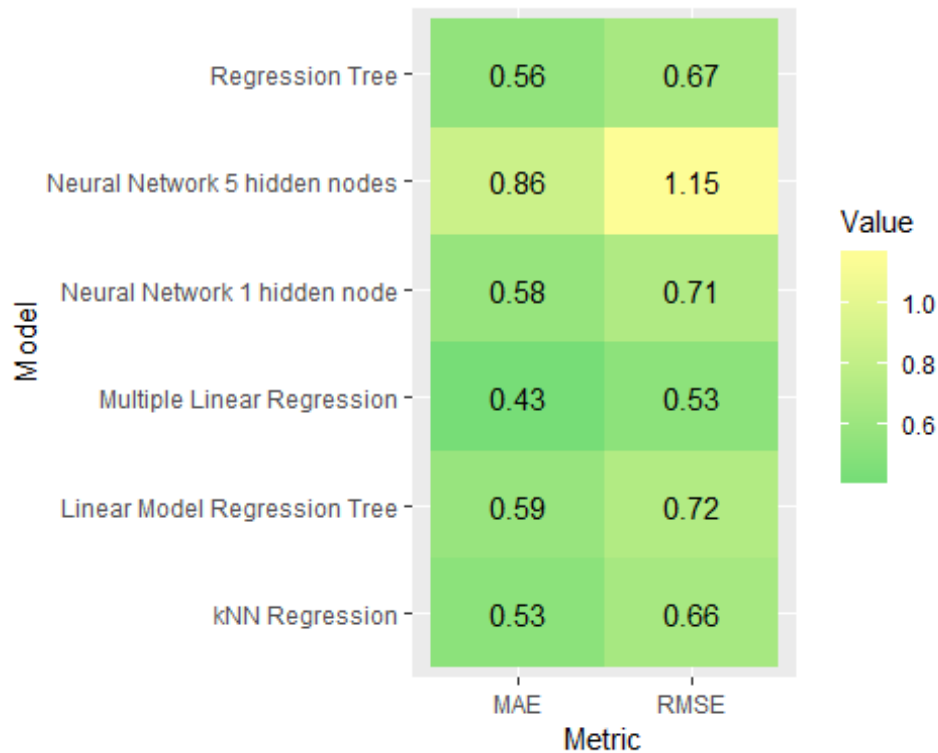
As we can see, the first Multiple Linear Regression and the kNN regression yield the lowest error.

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.nn <- as.vector(unnormalize(pred.nn, world_df$Happiness.Score))
pred.nn2 <- as.vector(unnormalize(pred.nn2, world_df$Happiness.Score))
pred.knnreg <- unnormalize(pred.knnreg, world_df$Happiness.Score)

#make a dataframe with all predictions
pred_df <- as.data.frame(cbind(pred.mlreg, pred.regtree, pred.regtree2,
pred.nn, pred.nn2, pred.knnreg))

#make a new column of the averages
pred_df$average.pred <- rowMeans(pred_df)

#make sure it worked
head(pred_df)

##     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.

```
MAE_e1 <- MAE(world_test$Happiness.Score, pred_df$average.pred)
MAE_e1
```

```
## [1] 0.3966315

RMSE_e1 <- RMSE(world_test$Happiness.Score, pred_df$average.pred)
RMSE_e1

## [1] 0.5220735
```
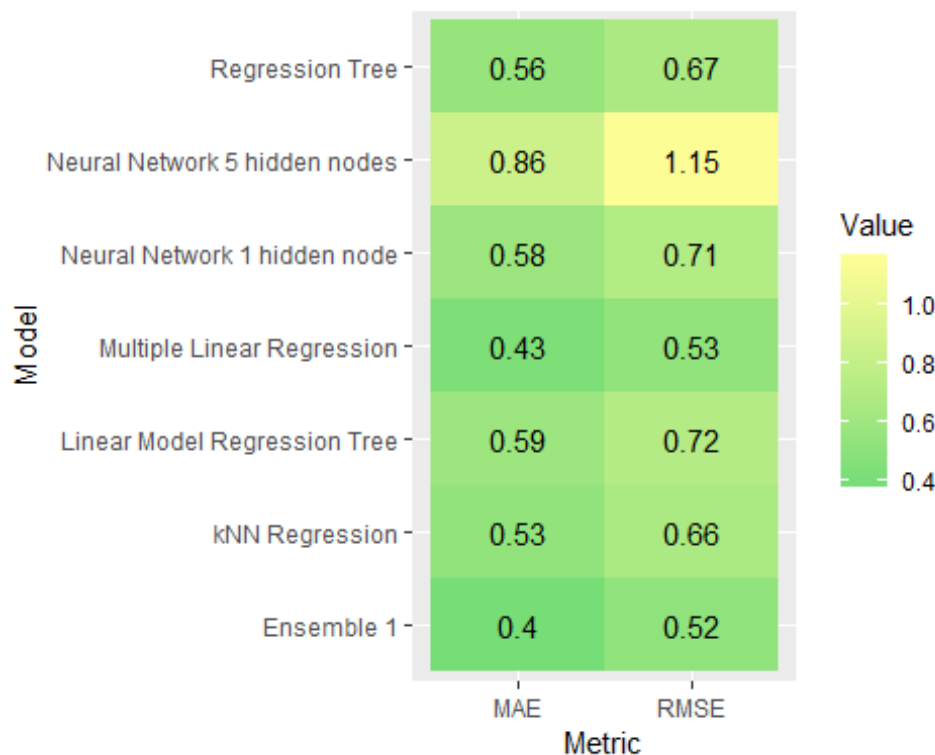
This is the lowest yet! Now let's add this to the heatmap.

```
#make heatmap of error metrics for each model
viz <- rbind(viz, c(MAE_e1,"Ensemble 1", "MAE"))
viz <- rbind(viz, c(RMSE_e1,"Ensemble 1", "RMSE"))
viz$Value <- viz$Value %>% as.numeric
```

```
ggplot(data = viz, aes(x = Metric, y = Model)) + geom_tile(aes(fill = Value))
+ scale_fill_gradient(low = "#77dd77", high = "#fdfd96") +
geom_text(aes(label = round(Value,2)))
```


As you can see, the Ensemble now has the best performance.

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.

```r
#find predicted values for the training data for every model
mlreg_fitted_values <- m.mlreg$fitted.values %>% as.vector()
nn_fitted_values <- unnormalize(m.nn$net.result[[1]]
,world_df$Happiness.Score)  %>% as.vector()
nn2_fitted_values <-
unnormalize(m.nn2$net.result[[1]],world_df$Happiness.Score) %>% as.vector
knn_fitted_values <- unnormalize(knnregTrain(train = world_train4, test =
world_train4, y = train3.scores, k= 11), world_df$Happiness.Score) %>%
as.vector
regtree_fitted_values <- predict(m.regtree, world_train2) %>% as.vector
regtree2_fitted_values <- predict(m.regtree2, world_train2) %>% as.vector

#make dataframe of fitted values
pred_df2 <- as.data.frame(cbind(mlreg_fitted_values, regtree_fitted_values,
regtree2_fitted_values, nn_fitted_values, nn2_fitted_values,
knn_fitted_values))

#make a new column of the averages
pred_df2$average.pred <- rowMeans(pred_df2)

#make sure it worked
head(pred_df2)

##   mlreg_fitted_values regtree_fitted_values regtree2_fitted_values
## 1            7.535516              7.187692               6.322541
## 2            6.725299              7.187692               6.322541
## 3            7.200435              7.187692               6.322541
## 4            6.543357              7.187692               6.322541
## 5            6.992755              7.187692               6.322541
## 6            6.873162              7.187692               6.322541
##   nn_fitted_values nn2_fitted_values knn_fitted_values average.pred
## 1         7.240736          7.687369          7.089091     7.177158
## 2         7.231709          7.517703          6.930182     6.985854
## 3         7.286160          7.515528          6.965909     7.079711
## 4         6.870703          7.181328          6.930182     6.839301
## 5         7.228506          7.403524          7.240818     7.062639
## 6         7.020329          7.290507          6.930182     6.937402

#add averaged predictions to dataset
ensemble2_train <- world_train2
ensemble2_test <- world_test2
ensemble2_train$average.pred <- pred_df2$average.pred
ensemble2_test$average.pred <- pred_df$average.pred

#use rpart to train ensemble model
ensemble2 <- rpart(Happiness.Score ~ Region + 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 +average.pred, data =
ensemble2_train)
#print it out
ensemble2

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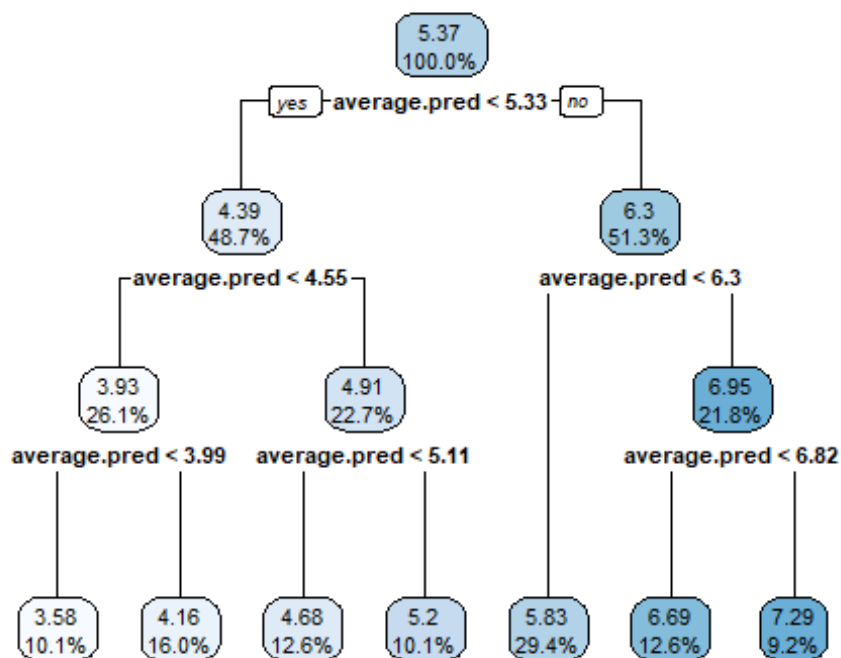
This output shows the logic behind the tree. As you can see, the previous predictions are weighing heavily on the tree's new predictions.

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.

```
#prediction on validation data
pred.ensemble2 <- predict(ensemble2, ensemble2_test)

MAE_e2 <- MAE(ensemble2_test$Happiness.Score, pred.ensemble2)
MAE_e2
```

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

```
RMSE_e2 <- RMSE(ensemble2_test$Happiness.Score, pred.ensemble2)
RMSE_e2
```

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

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

```
#make heatmap of error metrics for each model
viz <- rbind(viz, c(MAE_e2,"Ensemble 2", "MAE"))
viz <- rbind(viz, c(RMSE_e2,"Ensemble 2", "RMSE"))
viz$Value <- viz$Value %>% as.numeric


ggplot(data = viz, aes(x = Metric, y = Model)) + geom_tile(aes(fill = Value))
+ scale_fill_gradient(low = "#77dd77", high = "#fdfd96") +
geom_text(aes(label = round(Value,2)))
```
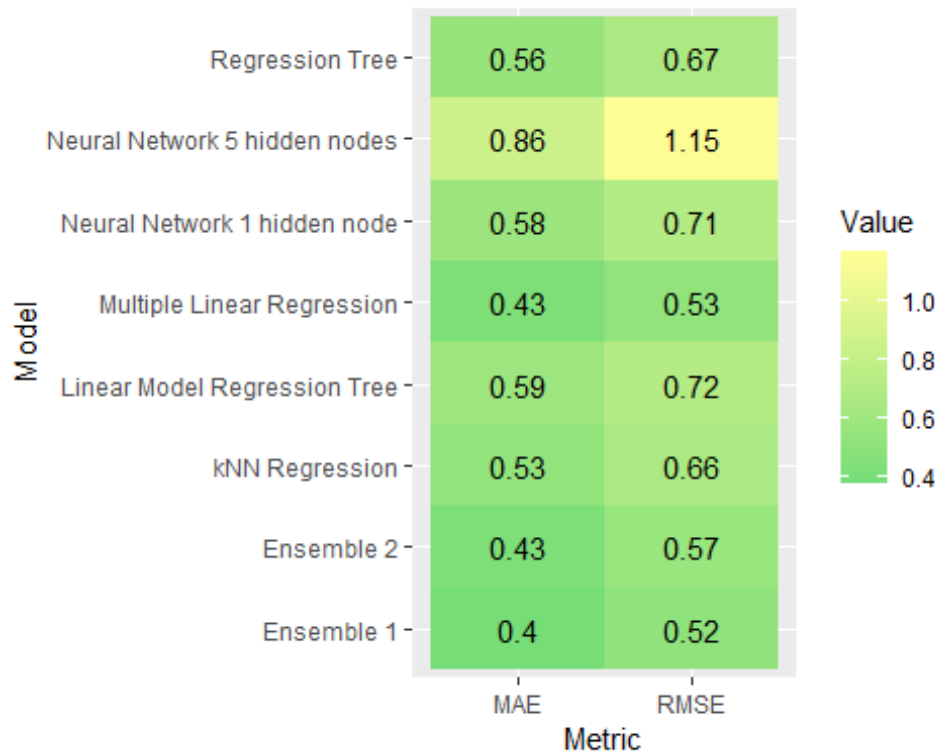
The ensemble models did improve performance over any singular model, which is a good sign! The overall errors are not that high. The predicted happiness scores tend to be pretty darn close to the actuals. Stacking focuses on removing bias by taking into account under and over predicters, hence why the 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