World Happiness Report - Capstone Project

Pamela Stameris Casey

2022-05-28

Contents

1 Introduction and Goals

1.1 How Can We Measure Happiness?

How can we measure and analyze happiness? What external and personal situations are factors of people's happiness? Is it different across the globe? Is anyone keeping track?

Well, someone is keeping track and even sharing that information.

The World Happiness Report (WHR), a publication of the Sustainable Development Solutions Network, is a compilation of data, statistics and analysis based on results from a series of surveys by the Gallup World Poll (GWP) over the last decade, investigating the level of life satisfaction of our global community. The central purpose, as stated in the introduction to each annual report, is "to review the science of measuring and understanding subjective well-being, and to use survey measures of life satisfaction to track the quality of lives as they are being lived in more than 150 countries."

Participants in the Gallup World Poll are asked to imagine their current life situation as a ladder with their best possible life as a 10 and their worst as a 0. Each respondent provides their numerical evaluation, which becomes averaged with other respondents from that country (along with some weighting to ensure population-representative averages), and becomes the "Happiness Score" for the country. Typically about 1000 people are polled from each country surveyed.

Along with the Happiness Score, the World Happiness Report includes six variables (or factors) to help explain the country happiness scores. These six factors are GDP per capita, Social Support, Healthy Life Expectancy, Freedom to Make Life Choices, Generosity, and Perceptions of Corruption. GDP per capita is based on information from the World Bank, while Life Expectancy is based on information from the World Health Organization. The other four factors are based on individual poll answers by respondents. These factors will be described in more detail once the data is downloaded and available for analysis.

1.2 Goals for this Project

This project is the final piece of work to be submitted for the edX Data Science Certificate Program offered through Harvard University. It is a self-directed project with student-selected data and student-defined objectives. It seemed very intimidating at first, but after completing the MovieLens Recommendation System project, I felt a sense of confidence in moving forward to this project.

One of the skills at which I wished to become more proficient was Data Visualization. On the MovieLens project, I used basic R plots for graphing my data. For this current project I decided to challenge myself to gain more knowledge and facility with using ggplot and some of its more flexible and visually appealing features. To this end, I have prioritized using ggplot to help me visualize and gain an understanding of the data and to look for patterns within the data with which to create a model.

2 Install Programming Packages and Load Data

2.1 Install Required Packages

The following packages are used in the analysis and code for creating the model.

```
if(!require(tidyverse)) install.packages(
  "tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages(
  "caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages(
```

```
  "data.table", repos = "http://cran.us.r-project.org")
if(!require(ggcorrplot)) install.packages(
  "ggcorrplot", repos = "http://cran.us.r-project.org")
if(!require(ggpubr)) install.packages(
  "ggpubr", repos = "http://cran.us.r-project.org")
library(tidyverse)
library(caret)
library(data.table)
library(ggcorrplot)
library(ggpubr)
```

2.2 Download Data Files

I have obtained data from several sources. I originally discovered the data on Kaggle, one of the course-recommended websites containing a collection of publicly available curated datasets. Kaggle contains datasets of World Happiness survey data for the years 2015 through 2019. These datasets are found here: https://www.kaggle.com/datasets/unsdsn/world-happiness.

In researching more about the WHR, I found that data is available directly from their website, which is the source of my data for the years 2020 through 2022: https://worldhappiness.report/.

The following code using relative file paths reads the files into my R Script.

```
# read WHR files
whr2015 <- read.csv("data\\2015.csv", header=TRUE, stringsAsFactors = FALSE)
whr2016 <- read.csv("data\\2016.csv", header=TRUE, stringsAsFactors = FALSE)
whr2017 <- read.csv("data\\2017.csv", header=TRUE, stringsAsFactors = FALSE)
whr2018 <- read.csv("data\\2018.csv", header=TRUE, stringsAsFactors = FALSE)
whr2019 <- read.csv("data\\2019.csv", header=TRUE, stringsAsFactors = FALSE)
whr2020 <- read.csv("data\\2020.csv", header=TRUE, stringsAsFactors = FALSE)
whr2021 <- read.csv("data\\2021.csv", header=TRUE, stringsAsFactors = FALSE)
whr2022 <- read.csv("data\\2022.csv", header=TRUE, stringsAsFactors = FALSE)
```

2.3 Pre-processing of Data

In the sourced data, the column headings differed across the years, so I have standardized the headings and the order of the columns using the following code.

```
# whr2015 Drop columns then reorder when needed
whr15 <- whr2015[-c(3,5)]
whr15 <- whr15[,c(1,3:7,9,8,10,2)]
whr16 <- whr2016[-c(3,5,6)]
whr16 <- whr16[,c(1,3:7,9,8,10,2)]
whr17 <- whr2017[-c(2,4,5)]
whr18 <- whr2018[-1]
whr19 <- whr2019[-1]
whr20 <- whr2020[-c(4:13)]
whr20 <- whr20[,c(1,3:10,2)]
whr21 <- whr2021[-c(4:13)]
whr21 <- whr21[,c(1,3:10,2)]
whr22 <- whr2022[-c(1,4,5)]
```

```
whr22 <- whr22[-c(147),c(1,2,4:9,3)]
#Standardize column names across years
colnames <-
  c("Country", "Happiness", "*GDPperCap", "*SocSupport","*LifeExp","*Freedom",
  ↪  "*Generosity", "*GovTrust","DystRes", "Region")
colnames(whr15) <- colnames
colnames(whr16) <- colnames
colnames(whr17) <- colnames[-10]
colnames(whr18) <- colnames[-c(9,10)]
colnames(whr19) <- colnames[-c(9,10)]
colnames(whr20) <- colnames
colnames(whr21) <- colnames
colnames(whr22) <- colnames[-10]
```

I then add a column containing the year of the report to each dataset.

3 Understanding the Data

3.1 Structure of the Data

An example of the structure of the data files for each individual year can be seen below with the first few lines of the dataset from the year 2015. This data is arranged by year and in descending order of Happiness Score for each year.

```
head(whr15)
```

```
##   year     Country Happiness *GDPperCap *SocSupport *LifeExp *Freedom
## 1 2015 Switzerland     7.587    1.39651     1.34951  0.94143  0.66557
## 2 2015     Iceland     7.561    1.30232     1.40223  0.94784  0.62877
## 3 2015     Denmark     7.527    1.32548     1.36058  0.87464  0.64938
## 4 2015      Norway     7.522    1.45900     1.33095  0.88521  0.66973
## 5 2015      Canada     7.427    1.32629     1.32261  0.90563  0.63297
## 6 2015     Finland     7.406    1.29025     1.31826  0.88911  0.64169
##   *Generosity *GovTrust DystRes         Region
## 1     0.29678   0.41978 2.51738 Western Europe
## 2     0.43630   0.14145 2.70201 Western Europe
## 3     0.34139   0.48357 2.49204 Western Europe
## 4     0.34699   0.36503 2.46531 Western Europe
## 5     0.45811   0.32957 2.45176  North America
## 6     0.23351   0.41372 2.61955 Western Europe
```

The number of countries surveyed each year varies from a low of 146 countries in 2022 to a high of 158 in 2015. The columns I mainly used are Year, Country and Region, along with Happiness and values for the six factors which the WHR have identified as relating to general feelings of happiness. There is one other column which gives the residual from "Dystopia", which represents a made-up country with the combined lowest values of the Happiness Scores and all the factors.

It should be noted that the data in each of these columns is not raw data. For this reason, I have included an asterisk as part of the column headings. These reported values have been processed by the authors/analysts of World Happiness Report and represent the contributions of each factor to the Happiness Score. I was not able to find an account of specifically how these factors were calculated, but together with the base Dystopia

value for that year and error, they sum to the Happiness Score. A description of Happiness Score and each of the six factors follows.

The Happiness Score is an average of the responses from the people surveyed from each country of their numerical evaluation of the satisfaction of their lives on a scale from a low of 0 to a high of 10. Two of the factors - **GDP per Capita**, representing the wealth of the country (presented as Logged GDP per Capita), and **Life Expectancy**, representing the health of the people of the country - are data that the WHR got from the World Bank and the World Health Organization respectively. It is important to restate that the values in the datasets are not raw data, but weighted values, calculated by the WHR authors and analysts, corresponding to how much they contribute to the Happiness Score.

The other four factors, also weighted components of the Happiness Score, are based on respondents' answers to the following questions on the survey: **Social Support** - "If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?" **Freedom to make your own choices** - "Are you satisfied or dissatisfied with your freedom to choose what you do with your life?" **Generosity** - the residual of regressing the national average of GWP responses to the question "Have you donated money to a charity in the past month?" on log GDP per capita. **Perceptions of Corruption (Government Trust)** - "Is corruption widespread throughout the government (or within businesses) in this country, or not?" (This information on the factors came from the World Happiness Report of 2022.)

Although this data is already analyzed and processed by the WHR organization, I felt it would be informative to conduct some investigations and visualizations to see what I could learn from it.

3.2 Create Data Set Merging All Years

For flexibility and efficiency in investigating the data, it will be useful to combine all the years into one dataset, which requires reducing the columns to only the shared columns, then binding them all together.

```
bind15 <- whr15[-c(10, 11)]
bind16 <- whr16[-c(10, 11)]
bind17 <- whr17[-10]
bind18 <- whr18
bind19 <- whr19
bind20 <- whr20[-c(10,11)]
bind21 <- whr21[-c(10,11)]
bind22 <- whr22[-10]
#Bind annual data sets
whr_all <- rbind(bind15, bind16, bind17, bind18,
                 bind19, bind20, bind21, bind22)
```

4 Model Development Plan

My initial plan for this project was to follow a similar strategy as the MovieLens project, by building a model which would be able to predict the Happiness Score of a country based on certain inputs corresponding to the factors that are given in the World Happiness Report. I would first create a Naive Model, which would be the average of all the Happiness Scores, then analyze the other factors and see which ones seemed to correspond most directly with Happiness, and use those to add biases to the mean Happiness Score. I soon realized that, because the factors have been already modified to fit the Happiness Score, this would not be a logical or valid method.

Since coming to that conclusion, it was necessary to familiarize myself with the data through analysis and visualization to see what track this project should take. The following graphs and analysis are the output of this investigative process. As I learned more from the data, a new plan emerged for the mission of my project. More on that later....

4.1 Summarize Annual Happiness Score Mean and Range.

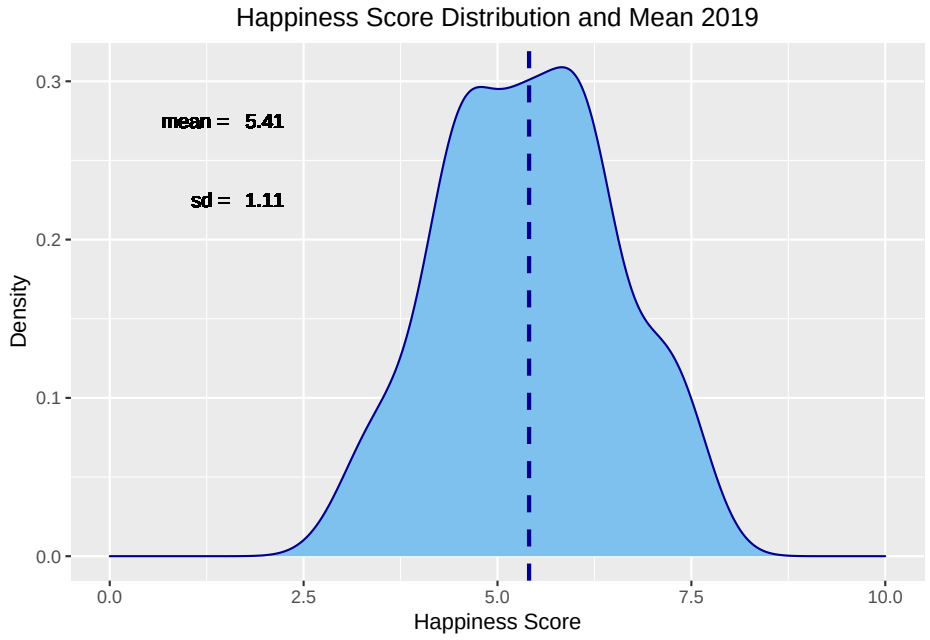
Because the World Happiness Report is focused on the global happiness level of people, I wanted to get a good idea of what the distribution of Happiness Scores looks like over the possible scores from 0 to 10. My first pass at this is to make a table of minimum, mean and maximum scores over the years. This table is shown below. We can see that the scores are between about 2.4 and 7.8, and are quite similar over the years.

```
annual_Happ_range <- whr_all %>%
  group_by(year) %>% summarize(Happiness.low=min(Happiness),
  ↪  Happiness.mean=mean(Happiness), Happiness.high=max(Happiness))
knitr::kable(annual_Happ_range)
```

year	Happiness.low	Happiness.mean	Happiness.high
2015	2.8390	5.375734	7.5870
2016	2.9050	5.382185	7.5260
2017	2.6930	5.354019	7.5370
2018	2.9050	5.375917	7.6320
2019	2.8530	5.407096	7.7690
2020	2.5669	5.473240	7.8087
2021	2.5230	5.532839	7.8420
2022	2.4040	5.553575	7.8210

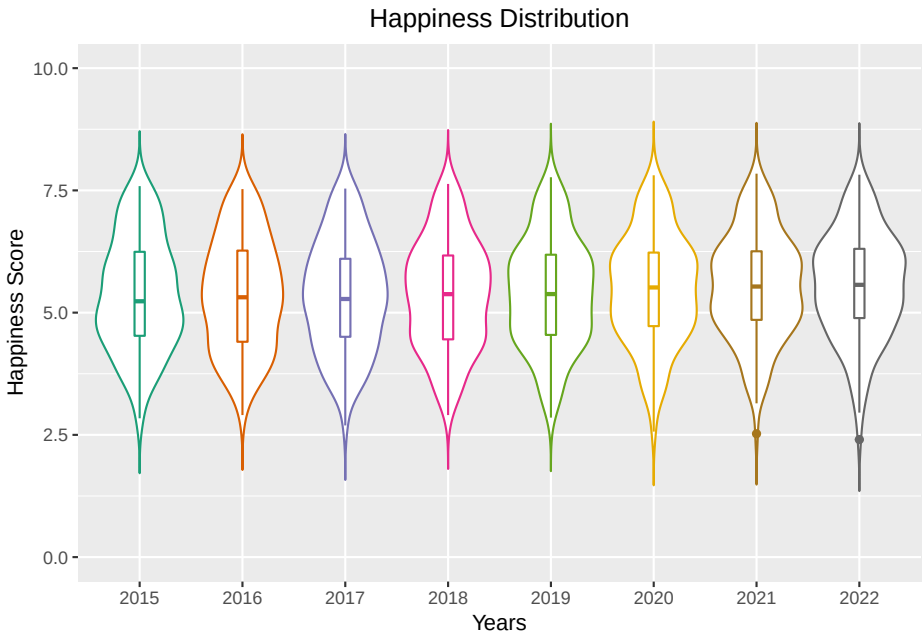
4.2 Happiness Score Distribution Curve

It is also helpful to see a visualization of Happiness Scores in in the form of a Density Plot. I chose the year 2019 for this data (for no particular reason). It is a fairly symmetrical distribution around the mean.



4.3 Visualization of Annual Happiness Scores

Finally, I wanted to see if and how the happiness score distribution changed from year to year. The following violin plots, combined with box plots, shows the distribution of happiness scores from 2015 to 2022. It is evident that there hasn't been much change in range and quartiles. I was expecting that there would have been a decrease in happiness scores in report years 2021 and 2022 because of the life changes caused by the pandemic, but was surprised to see that they remain at basically the same levels as in all the previous years.

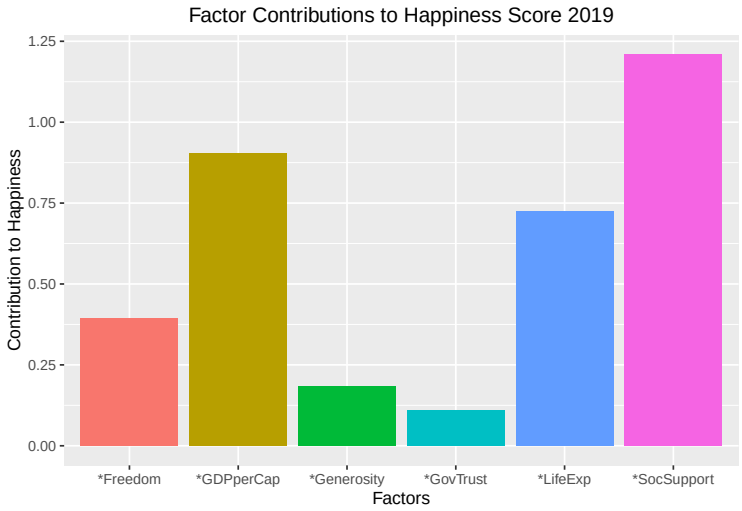


4.4 Factor Contribution to Happiness

My next investigation is to visualize the contribution level of each of the factors to the Happiness Score. For this analysis, I chose the year 2019, and averaged the values from each factor column, resulting in the average contribution over all the countries of the contribution of each factor. The actual averages are shown below, followed by a bar graph representing the level of contribution for each factor.

```
factors <- whr19 %>% select(4:9)
factor_means <- colMeans(factors)
factor_means
```

```
##  *GDPperCap *SocSupport    *LifeExp    *Freedom *Generosity   *GovTrust
##   0.9051474   1.2088141   0.7252436   0.3925705   0.1848462   0.1106026
```



5 The 2020 Dataset "Bonus" Columns

5.1 Details of the Bonus Columns

While investigating the WHR data for the various years, I noticed that the 2020 World Happiness Report has some data columns not included in the datasets of the other years. In addition to the columns with the "as explained by" columns for the six factors as in the datasets of the other years, it has original "bonus" data which has not been processed like the factor columns in the other years. It is more like "raw" data based on actual averages of survey results, or factual data regarding GDP per capita (in dollars) and Life Expectancy (in years). It is these columns that led me to my ultimate project plan.

These extra columns in the 2020 dataset are the values that led to the processed data that appears in the columns with the asterisks in the data of the other years. In the WHR dataset, they have kind of awkward names, so I will rename them later.

```
## [1] "Logged.GDP.per.capita"        "Social.support"
## [3] "Healthy.life.expectancy"      "Freedom.to.make.life.choices"
## [5] "Generosity"                   "Perceptions.of.corruption"
```

These columns represent the raw data from which the "as explained by" columns got their values. The Logged.GDP is the actual log (base 10) of the GDP for each country. The Life Expectancy column is the expected life span (in years) of people in each country. The Social Support, Freedom to Make Life Choices, and Perceptions of Corruption (aka Government Trust) columns are based on averages of the survey responses. The Generosity column has some negative numbers in it, so I don't know how it was calculated.

5.2 Reformulation and Refocus of Model Development

Once I discovered the existence and utility of these columns, I realized that they could possibly be used as predictors for creating a model for predicting the Happiness Score. This finally became the objective of my project: **To create a model which predicts the Happiness Scores of countries based on the numerical values of measured and surveyed factors.**

To this end, I decided to use the year 2020, and these factors along with the country Happiness Scores to create the model. I will use the root mean square error (RMSE) as a measure to assess the accuracy of the

model. In determining a target RMSE, I thought back to the MovieLens project. In that project, our goal was below 0.86490, given a range of 5 stars. The Happiness Score in this data also has a range of about 5 points on the ladder score of 1 to 10 (low of 2.4 to high of 7.8), so I figured that this might be an acceptable RMSE target for this project as well. Perhaps it will be unattainable, or perhaps it is not an aggressive enough target, but I will use it as a preliminary goal, at least.

5.3 Streamlining Data for Analysis

At this point, I will streamline the 2020 data by selecting the necessary columns and simplifying the factor column names to the same as the column names from the other years, but with the asterisk eliminated. Shown below are the first 5 and last 5 rows of the streamlined 2020 dataset.

```
##       Country         Region Happiness Log_GDP_percap SocSupport  LifeExp
## 1     Finland Western Europe    7.8087       10.63927  0.9543297 71.90083
## 2     Denmark Western Europe    7.6456       10.77400  0.9559908 72.40250
## 3 Switzerland Western Europe    7.5599       10.97993  0.9428466 74.10245
## 4     Iceland Western Europe    7.5045       10.77256  0.9746696 73.00000
## 5      Norway Western Europe    7.4880       11.08780  0.9524866 73.20078
##     Freedom  Generosity  GovTrust
## 1 0.9491722 -0.05948202 0.1954446
## 2 0.9514443  0.06620178 0.1684895
## 3 0.9213367  0.10591104 0.3037284
## 4 0.9488919  0.24694422 0.7117097
## 5 0.9557503  0.13453263 0.2632182
```

```
##                       Country             Region Happiness Log_GDP_percap
## 149 Central African Republic Sub-Saharan Africa    3.4759       6.625160
## 150                   Rwanda Sub-Saharan Africa    3.3123       7.600104
## 151                 Zimbabwe Sub-Saharan Africa    3.2992       7.865712
## 152              South Sudan Sub-Saharan Africa    2.8166       7.425360
## 153              Afghanistan         South Asia    2.5669       7.462861
##     SocSupport  LifeExp   Freedom  Generosity  GovTrust
## 149  0.3194599 45.20000 0.6408806  0.08241036 0.8918067
## 150  0.5408354 61.09885 0.9005894  0.05548395 0.1835412
## 151  0.7630928 55.61726 0.7114579 -0.07206395 0.8102370
## 152  0.5537071 51.00000 0.4513136  0.01651855 0.7634173
## 153  0.4703670 52.59000 0.3965730 -0.09642940 0.9336866
```

6 Investigating and Analyzing the 2020 Data

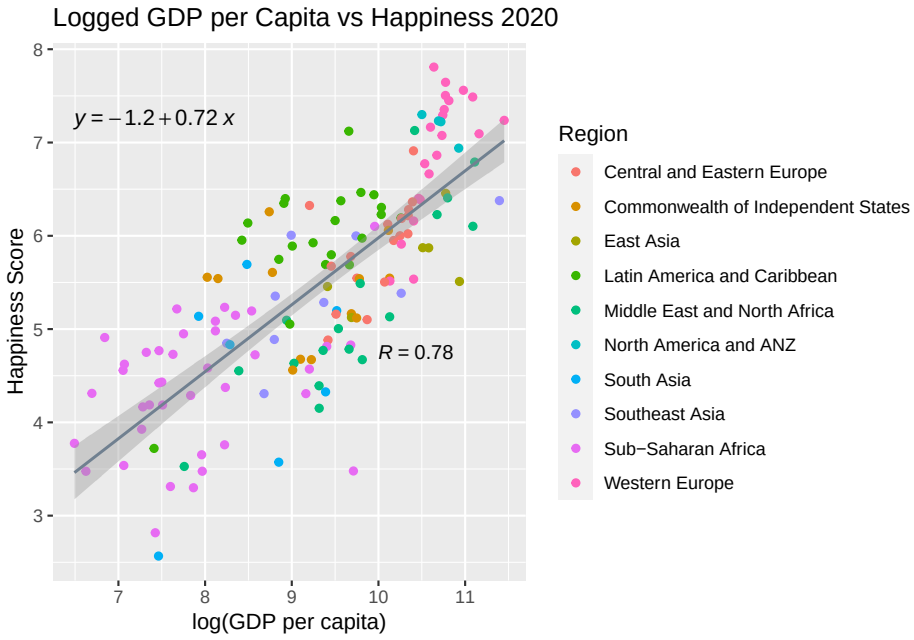
The following section includes analysis tools and visualization I used to help me understand the extra data from the 2020 dataset better and to see if any patterns emerged that would help me in my attempt to better model the actual values of Happiness Score based on Wealth (GDP), Social Support, Health (Life Expectancy), Freedom to Make Choices, Generosity and Government Trust.

6.1 2020 Wealth and Happiness

6.1.1 Graph of Logged GDP vs Happiness Score

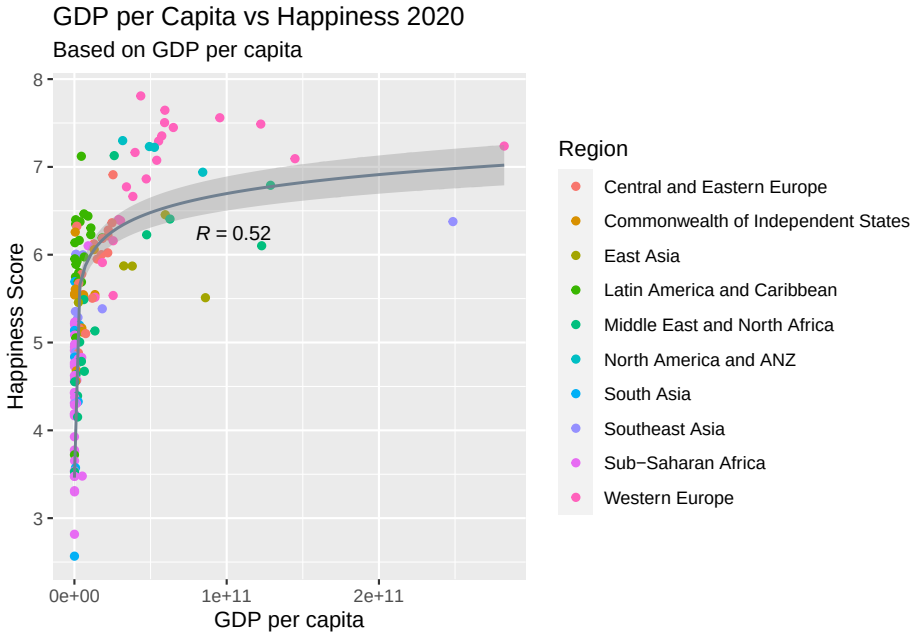
Wealth data, as reported in the World Happiness Report, is based on GDP per capita for each country. In the 2020 Report, it is reported as "Logged GDP per capita". I was interested in the relationship between

the wealth of the country and the Happiness Score, so I graphed the given data. I included a color key for region, in case a regional pattern that became evident. The graph shows a linear relationship, as shown by the regression line in the graph below, with an R value of 0.78.



6.1.2 Graph of GDP vs Happiness Score

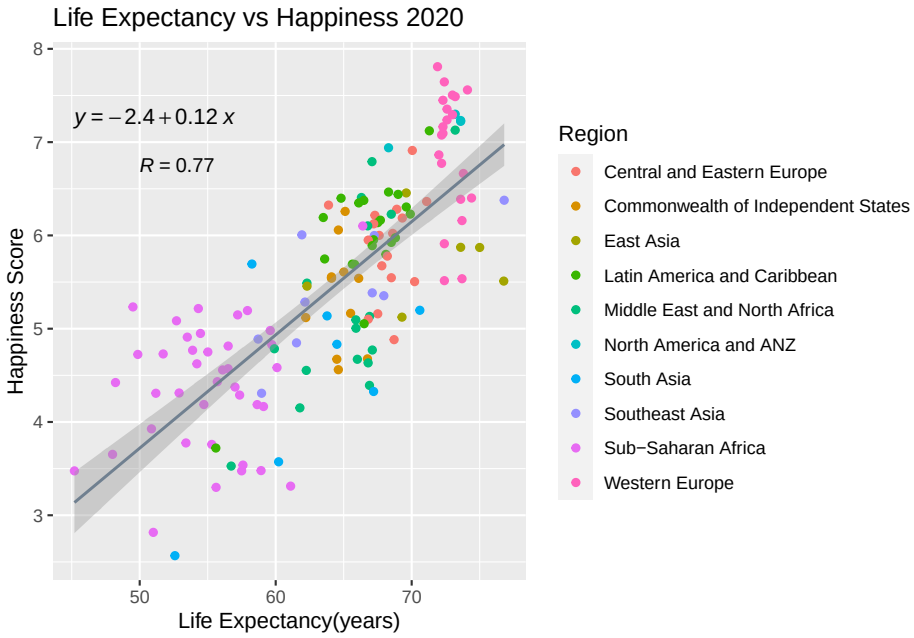
In sharing my investigation observations with some friends, one of them said that she was of the impression that "you can only buy happiness up to a certain level" which corresponds with at least having basic needs met, but then levels off. That statement got me thinking, so I then wrote a formula to calculate the actual GDP (10^(log(GDP))), and used this as the independent variable. This adjusted graph (below the logged GDP graph) clearly shows the leveling off of the Happiness Score after a certain level of GDP.



Based on the behavior shown on these graphs, I feel is is safe to say that GDP per capita, which can be extended to be an indicator of individual income level, can be considered a factor in Happiness level.

6.2 2020 Health and Happiness

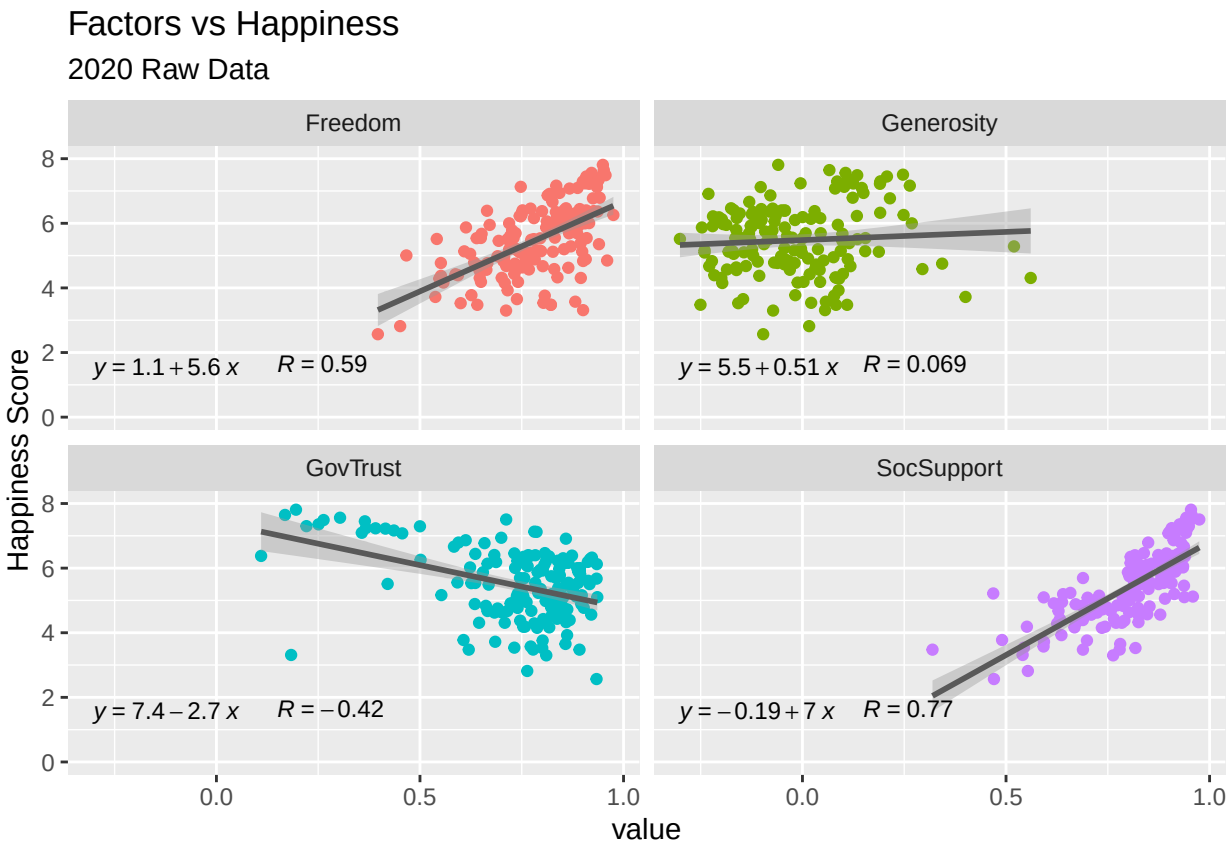
The relationship of health and Happiness, shown here as a graph of Life Expectancy vs Happiness Score also shows a linear relationship with an R value of 0.77.



Like wealth, it seems that health is also a factor in happiness level.

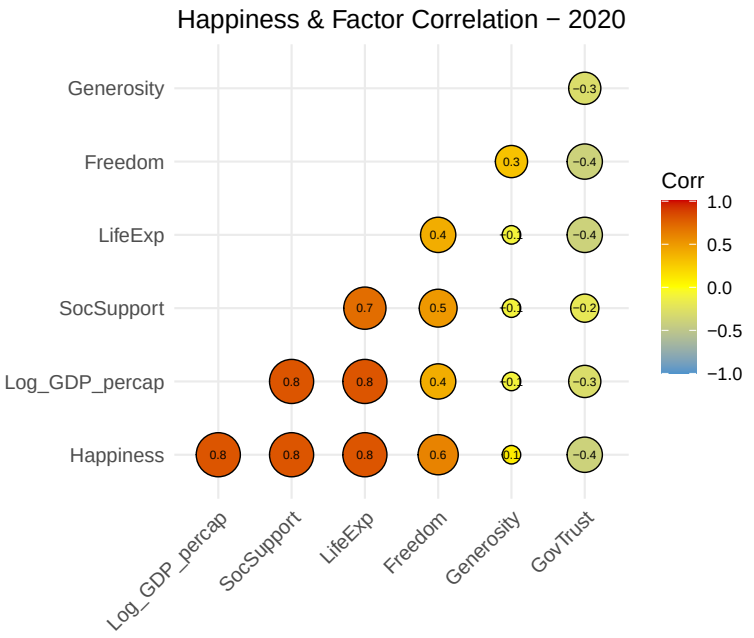
6.3 Social Support, Freedom, Generosity and Government Trust

The following facet plot shows scatterplots and regression equations for the remaining factors. I was able to combine these into a facet plot, as the values were all of the same magnitude. Due to the smaller size, I did not specify the regions for these graphs.



6.4 Happiness and Factor Correlation 2020

Combining all the factors from 2020 into one visualization, we can see by looking at the bottom row, the correlation of the "raw" 2020 factors have with the Happiness Score. The factors that have the highest correlation with happiness are GDP per capita, social support, life expectancy and freedom to make life choices. Government trust has a lower correlation, while generosity seems to have no correlation to the Happiness Score.



Based on this correlation information, I will use all the factors except generosity in creating the model to predict happiness.

7 Model Development using K-Nearest Neighbors

Now that I have a pretty good understanding of the data, it is time to decide on a path forward to create a model for predicting the Happiness Score. My plan is to try K-Nearest Neighbors (KNN) machine learning algorithm to see if it results in a model with an acceptable RMSE. The K-Nearest Neighbors algorithm is a supervised machine learning algorithm which uses similarity between features to predict values of new data points. In the context of this project, it uses distances between the country factors to group countries with similar characteristics and (as this is a regression case vs classification) averages the values of the desired result (Happiness Score) to assign a predicted Happiness Score. I thought it would be an effective predictive algorithm in this situation, as it will look at other countries which have similar factor values (GDP, Life Expectancy, etc) to the query country, and use the Happiness Scores of those countries to predict the Happiness Scores of the country in question. The algorithm also samples the data to choose the optimal value of k (the number of neighbors to use). By the way, "neighbors" in this case does not refer to neighboring countries, but closest countries in terms of factor characteristics.

In using the KNN machine learning algorithm, it is necessary to partition the data, leaving a validation set available for testing to calculate the final RMSE, without using it for training. The secondary training set created below will be used to determine the best number of neighbors to use in the model. That value of k will become part of the model, which will then be used on the validation set to determine the final RMSE against the actual Happiness Scores. To use KNN, the data must also be scaled and centered.

7.1 Partitioning of the training set

To properly train and test a model it is necessary to create three data sets: a training set, a test set to use to evaluate the model as it is being developed, and a validation set which will be used only after the model has been finalized. The following code first partitions the 2020 dataset into two sets - **haptrain** and **hapval**. It then partitions **haptrain** into a secondary training and test set for the knn model development - **knntrain**

and **knntest**. The final validation set **hapval** will not be used until the very end to test the final model, using **haptrain** as its training set. I chose my partition proportions based on wanting no fewer than about 25 observations in my test and validation sets, while still keeping enough observations in my training set to effectively train the model.

```
set.seed(1, sample.kind="Rounding")
model_test_index <-
  createDataPartition(y = happy$Happiness, times = 1, p = 0.17, list = FALSE)
haptrain <- happy[-model_test_index,]
hapval <- happy[model_test_index,]

set.seed(2, sample.kind="Rounding")
model_test_index2 <-
  createDataPartition(y = haptrain$Happiness, times = 1, p = 0.2, list = FALSE)
knntrain <- haptrain[-model_test_index2,]
knntest <- haptrain[model_test_index2,]
```

7.2 Prepare the data

We will use all predictors except for Generosity, as it has a very low correlation to the Happiness Score in the training dataset. The following code selects the proper columns, then scales and centers the data in preparation for the KNN algorithm.

```
knntrain_x <- knntrain[, c(4:7,9)]
knntrain_x <- scale(knntrain_x)[,]
knntrain_y <- knntrain[,3]

knntest_x <- knntest[, c(4:7,9)]
knntest_x <- scale(knntest_x)[,]
knntest_y <- knntest[,3]
```

7.3 Use K-Nearest Neighbor Regression to train using test set

The code below creates a KNN model using the **knntrain** partition, which tests various k values (number of neighbors) from 3 to 25 and reports the results of the samples taken, and finally reports the optimal k and the final RMSE.

```
knnmodel <- train(knntrain_x, knntrain_y, method = "knn", tuneGrid = data.frame(k =
↪  seq(3, 21, 1)))
knnmodel
```

```
## k-Nearest Neighbors
##
## 97 samples
##  5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 97, 97, 97, 97, 97, 97, ...
## Resampling results across tuning parameters:
```

```
##
##    k   RMSE       Rsquared   MAE
##     3  0.6616338  0.6583171  0.5121402
##     4  0.6546188  0.6606661  0.5031362
##     5  0.6417495  0.6715553  0.4910288
##     6  0.6393839  0.6727273  0.4895661
##     7  0.6349431  0.6784956  0.4839943
##     8  0.6360899  0.6801024  0.4869279
##     9  0.6280582  0.6930706  0.4793457
##    10  0.6323864  0.6902310  0.4822822
##    11  0.6325231  0.6917800  0.4812451
##    12  0.6331541  0.6944886  0.4812146
##    13  0.6337719  0.6967757  0.4821260
##    14  0.6307626  0.7012908  0.4806363
##    15  0.6292068  0.7072961  0.4827506
##    16  0.6302384  0.7090644  0.4843839
##    17  0.6303470  0.7101771  0.4845312
##    18  0.6336969  0.7084606  0.4882536
##    19  0.6359658  0.7082692  0.4898311
##    20  0.6397140  0.7064797  0.4937926
##    21  0.6426367  0.7058409  0.4972717
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```

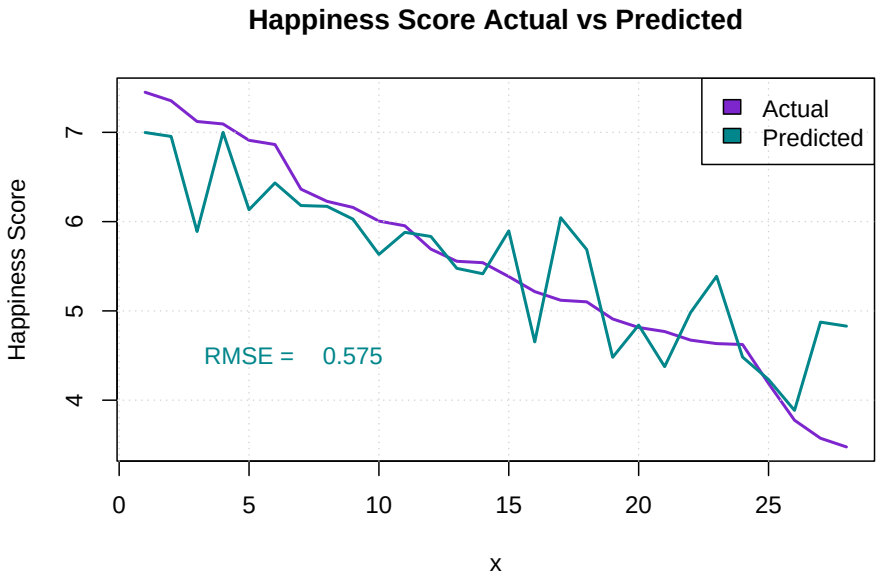
```
predknn_y <- predict(knnmodel, data.frame(knntest_x))
RMSE <- function(true_score, predicted_score){
  sqrt(mean((true_score - predicted_score)^2))
}
rmse_knn <- RMSE(knntest_y, predknn_y)

cat(" RMSE: ", round(rmse_knn,4))
```

```
##  RMSE:  0.575
```

A note regarding the KNN outcome model: Each time model is run, it is possible to have a different result (k-value and RMSE). In doing research on this, I learned that it is due to the stochastic nature of the KNN learning algorithm. I believe this is due to the randomness of the bootstrap resampling. The RMSE values that I kept track of after realizing what was happening ranged from 0.575 to 0.6117. I think this stochasticity just needs to be an accepted variability in the model.

7.4 Plot of Test and KNN Predicted Data

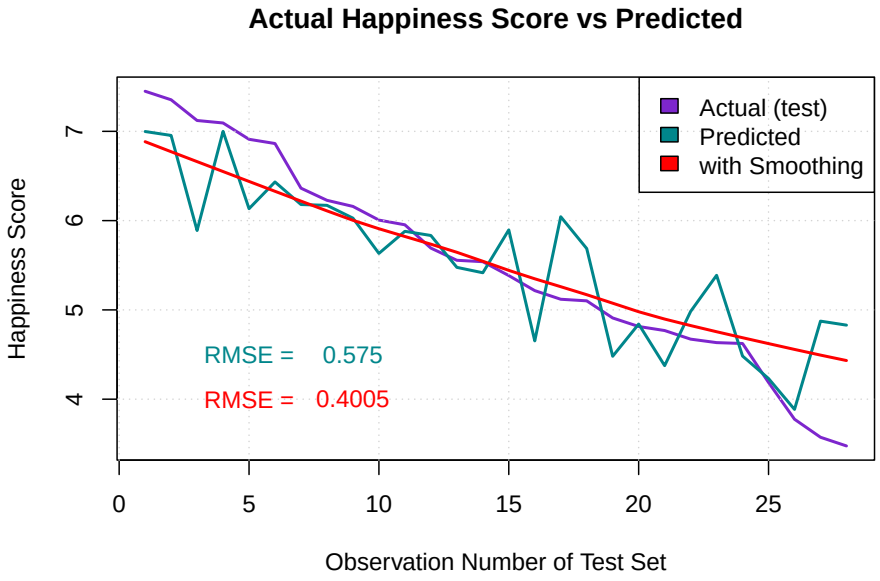
The plot created below shows the actual Happiness Scores for the **knntest** partition compared to the the predicted Scores.



That prediction line looks very choppy, so perhaps it would improve the model to smooth using the Lowess smoothing function. I specified the default f value of 2/3 as shown in the code. The new RMSE is shown below, which is an improvement over the original KNN model, along with the visual result of the smoothing is shown in the following plot.

```
lowess_test <- lowess(predknn_y ~ x, f=2/3)
rmse_lowess_test <- RMSE(knntest_y, lowess_test$y)
cat(" RMSE: ", round(rmse_lowess_test,4))
```

```
##  RMSE:  0.4005
```



Based on this low RMSE which is well below my target, I will consider this model using K-Nearest-Neighbors and Lowess Smoothing to be the final model. Below are the RMSE results and visualization of the model as run on the validation set.

8 Results of using the KNN with Lowess Smoothing Model to Predict Happiness Scores on Validation Set

8.1 Run Model on the Validation Set

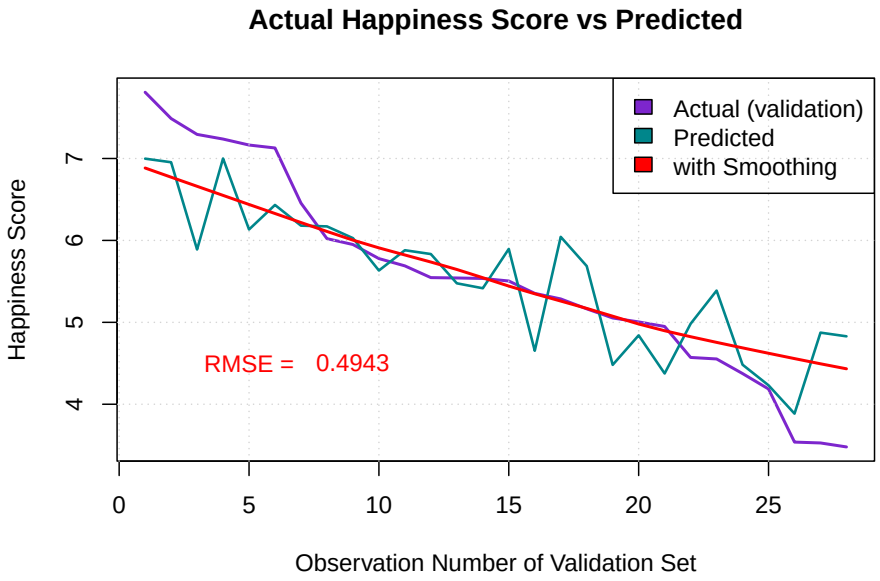
The following code will prepare the validation set for the KNN machine learning algorithm, then use the Lowess Smoothing function to produce the final model results and display the RMSE.

```
# Prepare the data (select factors, scale and center)
haptrain_x <- haptrain[, c(4:7,9)]
haptrain_x <- scale(haptrain_x)[,]
haptrain_y <- haptrain[,3]

hapval_x <- knntest[, c(4:7,9)]
hapval_x <- scale(knntest_x)[,]
hapval_y <- hapval[,3]

#Run the model and display the final RMSE
pred_y_knn <- predict(knnmodel, data.frame(hapval_x))
lowess_val <- lowess(pred_y_knn ~ x, f=2/3)
rmse_lowess_val <- RMSE(hapval_y, lowess_test$y)

cat(" Final Model RMSE: ", round(rmse_lowess_val,4))
```

```
##  Final Model RMSE:  0.4943
```

8.2 Plot of Validation and Predicted Data

The following plot shows the comparison of the actual Happiness Scores with the Scores generated using only KNN and finally, in red, the Scores generated by using Lowess smoothing after KNN. The RMSE is also noted on the plot.



8.3 Results Summary

The final model that I developed for predicting the Happiness Score of countries uses the supervised machine learning algorithm K Nearest Neighbors along with the Lowess smoothing algorithm. Using these algorithms has resulted in an RMSE of the predicted Happiness Scores vs the actual Happiness Scores of the validation set of approximately 0.5. This RMSE is well below the target I had set for my project, which was based on the MovieLens project target RMSE. I really had no idea how to set a target, so relied on the only experience I had.

The KNN algorithm has randomness built into it, as it takes random samples of the training set to determine the optimal k-value (number of nearest neighbors) to use in the model. Because of this stochasticity of the KNN algorithm, the k-value, and thus the predicted Happiness Scores can change each time the algorithm is called. After running the model multiple times, I could see the k-value and the RMSE vary each time. The selected k-value varied between 9, 11 and 13. The RMSE on the validation set that has been produced by the model varied, in my iterations, between 0.4943 and 0.5703, so I am not able to report an exact RMSE for my model.

9 Conclusions and Further Study

This model, although predictive of Happiness Score, does not provide the contribution of each factor to happiness. As such, I feel that the standalone model is not particularly useful for understanding happiness, or being able to use it to improve the happiness of the global community. The data from the World Happiness Report includes a breakdown of each factor's contribution to the Happiness Score, which is a more useful product than being able to predict happiness level based on the values of the factors.

That said, I feel the real value of this project is the personal benefit to me from all that I have learned by doing this project. I gained understanding and facility with using ggplot to visualize data. I spent time researching KNN and Lowess to be able to use them in my model. I learned what "stochasticity" is, and how randomness affects data analysis and model development. I had to figure out how to create a report that accesses and reads in data from online sources, and how to use relative file paths. I will follow the World Happiness Report in the future, and look for a future year where the unprocessed survey results are

provided (what I called “bonus columns”), so I can try my model out on a full set of countries to see how it performs!