World Happiness

Guillioli, Walter

1/23/2020

Introduction

What makes us happy? It's about community, family and doing stuff we love. It's also about learning to be at peace with yourself and accept life as it is. The best book I have read on the topic is Happiness: A Guide to Developing Life's Most Important Skill by Matthieu Ricard.
https://www.amazon.com/Happiness-Guide-Developing-Lifes-Important/dp/0316167258

But today let's ake a data science approach to explore it.

This report is written to walk through an example of the lifeycle of a data science project. We will load, explore and prepare data. Then we will use statistics and Machine Learning algorithms to understand why people in some countries are happier than in others. This report is written for a technical audience with a focus on the aspiring data scientist.

Data Overview

We will use the dataset provided by Kaggle. This is the World Happiness Report that was released by the United Nations as is now considered a landmark survey in the state of global happiness. The first release was in 2012 but for this report will use the data for 2019. The data ultimately comes from the Gallup World Poll. For more contest see https://www.kaggle.com/unsdsn/world-happiness.

Load Dataset

First, we load the data and explore the size and structure of the data frame of 156 observations and 9 variables.

```
#Set working directory
#setwd("C:/Users/wguil/OneDrive/Documents/GitHub/world_happiness/")

#Load happiness data for 2019
d2019 <- read.csv("../data/2019.csv", stringsAsFactors = FALSE)

#Make a working copy
d <- d2019

#Size of the data frame
dim(d)
```

```
## [1] 156   9
```

```
#Column names, type of variable and sample values
str(d)
```

```
## 'data.frame':    156 obs. of  9 variables:
##  $ Overall.rank               : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ Country.or.region           : chr  "Finland" "Denmark" "Norway" "Iceland" ...
##  $ Score                       : num  7.77 7.6 7.55 7.49 7.49 ...
##  $ GDP.per.capita              : num  1.34 1.38 1.49 1.38 1.4 ...
##  $ Social.support              : num  1.59 1.57 1.58 1.62 1.52 ...
##  $ Healthy.life.expectancy     : num  0.986 0.996 1.028 1.026 0.999 ...
##  $ Freedom.to.make.life.choices: num  0.596 0.592 0.603 0.591 0.557 0.572 0.574 0.585 0.584 0.532 ..
##  $ Generosity                  : num  0.153 0.252 0.271 0.354 0.322 0.263 0.267 0.33 0.285 0.244 ...
##  $ Perceptions.of.corruption   : num  0.393 0.41 0.341 0.118 0.298 0.343 0.373 0.38 0.308 0.226 ...
```

```
#Change column names to friendlier and shorter names
colnames(d) <- c("rank", "country", "score", "gpd_pc", "social_support", "life_expectancy", "freedom" ,

#Sample of 10 observations
kable(d[1:10,], row.names = FALSE)
```

rank	country	score	gpd_pc	social_support	life_expectancy	freedom	generosity	corruption
1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298
6	Switzerland	7.480	1.452	1.526	1.052	0.572	0.263	0.343
7	Sweden	7.343	1.387	1.487	1.009	0.574	0.267	0.373
8	New Zealand	7.307	1.303	1.557	1.026	0.585	0.330	0.380
9	Canada	7.278	1.365	1.505	1.039	0.584	0.285	0.308
10	Austria	7.246	1.376	1.475	1.016	0.532	0.244	0.226

One data point I would like to have is the continent of each country. I want to compare happiness in let's say Latin America and Europe. I noticed that region is present in the data from 2015 so I will add this to the data frame.

```
#Load happiness data for 2015
d2015 <- read.csv("../data/2015.csv", stringsAsFactors = FALSE)

#Add Region to my dataset by merging by country name
d <- merge(d, d2015[,c("Country", "Region")], by.x = "country", by.y = "Country", all.x = TRUE)

#Change col name so every column is lower case for consistency
names(d)[names(d) == "Region"] <- "region"

#Let's look at region names and their count of countries
d %>%
  count(region, sort = TRUE)
```

```
## # A tibble: 11 x 2
##    region                          n
##    <chr>                       <int>
##  1 Sub-Saharan Africa             36
##  2 Central and Eastern Europe     28
##  3 Latin America and Caribbean    20
##  4 Western Europe                 20
```

```
##  5 Middle East and Northern Africa    19
##  6 Southeastern Asia                   9
##  7 Southern Asia                       7
##  8 <NA>                                7
##  9 Eastern Asia                        6
## 10 Australia and New Zealand           2
## 11 North America                       2
```

```
#It seems 7 countries don't have a region so let's see which ones
d[is.na(d$region),]$country
```

```
## [1] "Gambia"            "Namibia"           "North Macedonia"
## [4] "Northern Cyprus"   "Somalia"           "South Sudan"
## [7] "Trinidad & Tobago"
```

```
#Let's add the region to these 7 countries
d[d$country=="Gambia", ]$region <- "Africa"
d[d$country=="Namibia", ]$region <- "Africa"
d[d$country=="North Macedonia", ]$region <- "Central and Eastern Europe"
d[d$country=="Northern Cyprus", ]$region <- "Central and Eastern Europe"
d[d$country=="Somalia", ]$region <- "Africa"
d[d$country=="South Sudan", ]$region <- "Africa"
d[d$country=="Trinidad & Tobago", ]$region <- "Latin America and Caribbean"

#This is not quite what I wanted as I wanted continents, so let's derive a continents column.
#I will use the 7 continents definition but minor adjustments based on areas I want to see
d <- d %>%
  mutate(continent = case_when(region == "Sub-Saharan Africa" ~ "Africa",
                               region == "Middle East and Northern Africa" ~ "Africa",
                               region == "Africa" ~ "Africa",
                               region == "Southeastern Asia" ~ "Asia",
                               region == "Southern Asia" ~ "Asia",
                               region == "Eastern Asia" ~ "Asia",
                               region == "Central and Eastern Europe" ~ "Europe_CEE",
                               region == "Western Europe" ~ "Europe_WE",
                               region == "Latin America and Caribbean" ~ "South America",
                               region == "Australia and New Zealand" ~ "Australasia",
                               region == "North America" ~ "North America"
                               ))

#Let's see what we have
d %>%
  count(continent, sort = TRUE)
```

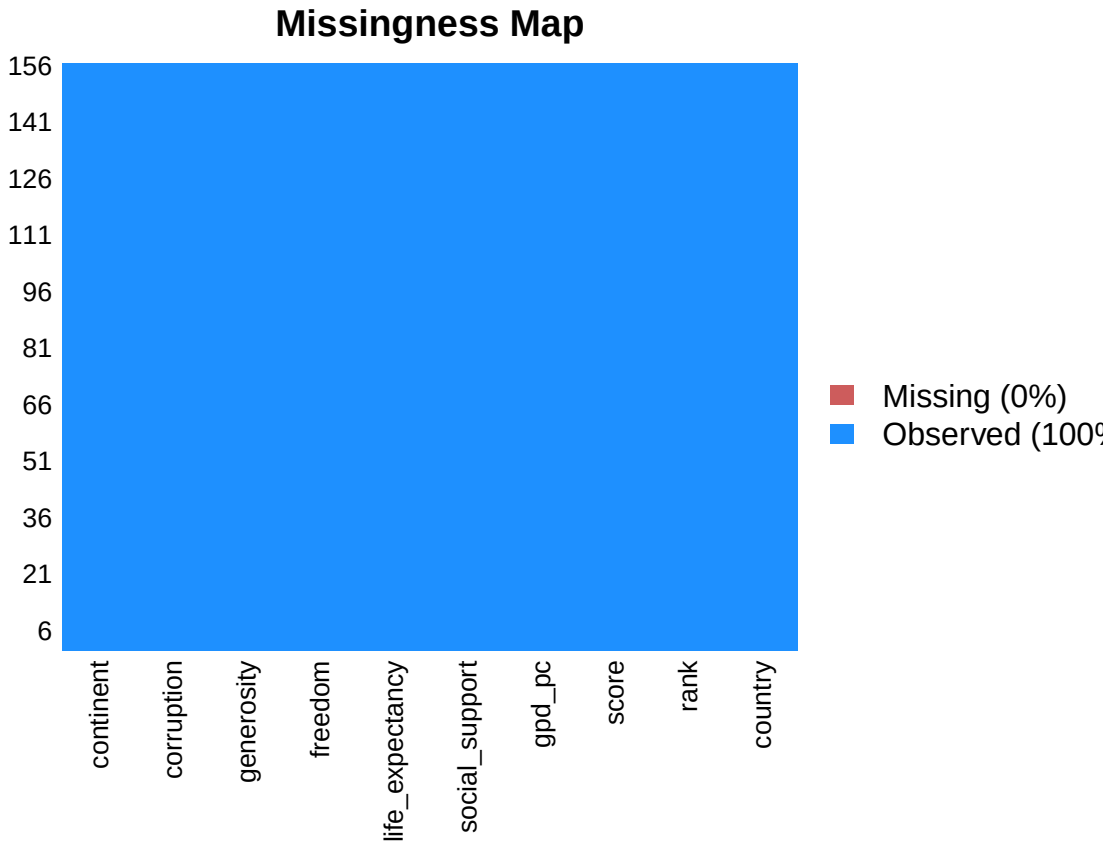
```
## # A tibble: 7 x 2
##   continent         n
##   <chr>         <int>
## 1 Africa           59
## 2 Europe_CEE       30
## 3 Asia             22
## 4 South America    21
## 5 Europe_WE        20
## 6 Australasia       2
## 7 North America     2
```

```r
#Drop region since
d <- subset(d, select = -c(region))
```

Missing Values

A key part is validating if there are any missing values in the information. If there is we need to address this. A nice option is the missing values plot from the Amelia package. Fortunately there are not missing values in our data.

```r
#Plot missing values
missmap(d)
```



Univariate Data Exploration

It is very important to understand what each column of data is and what type of data we are dealing with. So here we double click on each variable to understand it with summary statistics and plots.

I like to get a list of the variables and it's type and then explore 1x1 since it's a small dataset.

```r
str(d)
```

```
## 'data.frame':    156 obs. of  10 variables:
##  $ country        : chr  "Afghanistan" "Albania" "Algeria" "Argentina" ...
```

```
##  $ rank           : int  154 107 88 47 116 11 10 90 37 125 ...
##  $ score          : num  3.2 4.72 5.21 6.09 4.56 ...
##  $ gpd_pc         : num  0.35 0.947 1.002 1.092 0.85 ...
##  $ social_support : num  0.517 0.848 1.16 1.432 1.055 ...
##  $ life_expectancy: num  0.361 0.874 0.785 0.881 0.815 ...
##  $ freedom        : num  0 0.383 0.086 0.471 0.283 0.557 0.532 0.351 0.536 0.527 ...
##  $ generosity     : num  0.158 0.178 0.073 0.066 0.095 0.332 0.244 0.035 0.255 0.166 ...
##  $ corruption     : num  0.025 0.027 0.114 0.05 0.064 0.29 0.226 0.182 0.11 0.143 ...
##  $ continent      : chr  "Asia" "Europe_CEE" "Africa" "South America" ...
```

a) country: the country name should be unique so let's double check no duplicates exist.

```
sum(duplicated(d$country))
```

```
## [1] 0
```

b) rank: the country rank should go from 1 to 156 and should be unique, let's double check.

```
summary(d$rank)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   39.75   78.50   78.50  117.25  156.00
```

```
sum(duplicated(d$rank))
```

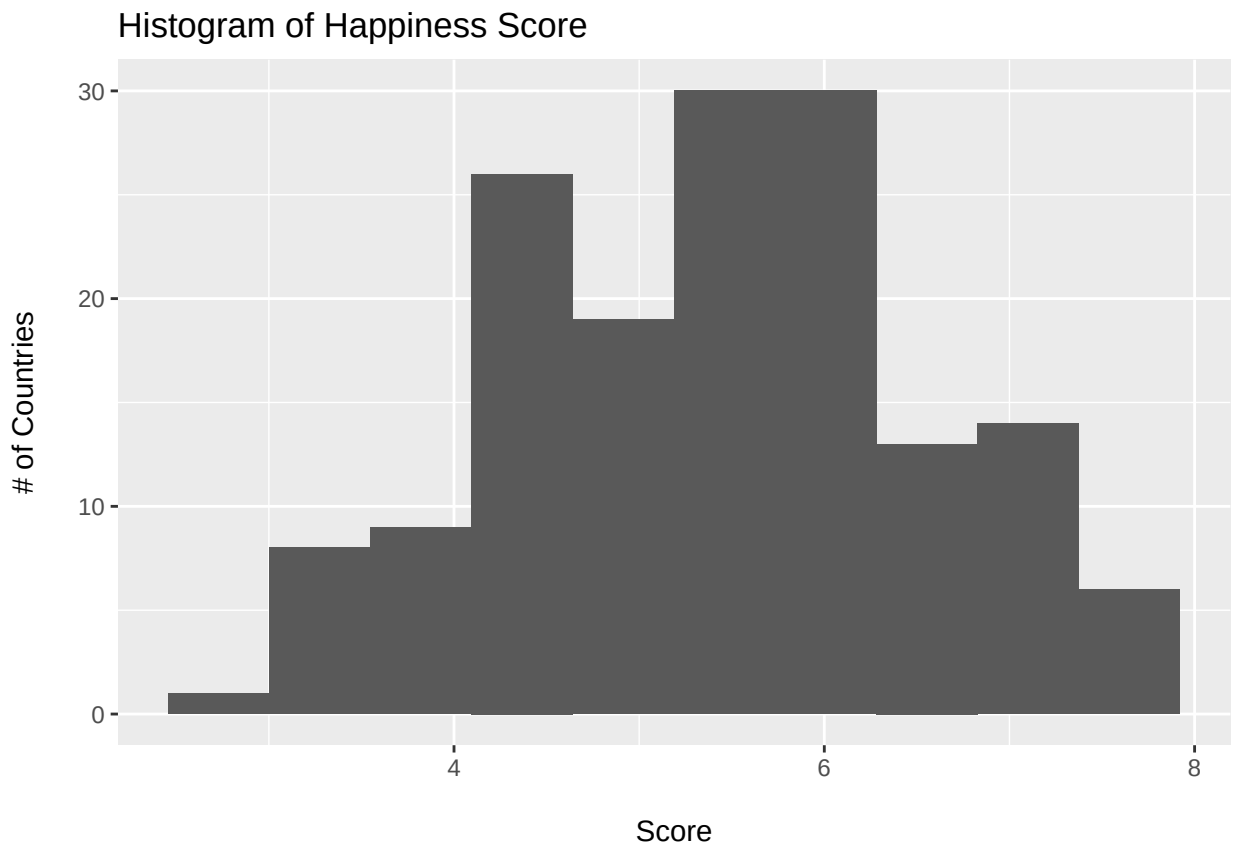
```
## [1] 0
```

c) score: this is the happiness score and the main variable of interest. It ranges from 2.853 to 7.769 and has a very normal distribution, which is really a good thing since we will predict this variable later using Machine Learning algorithms. And some of them, like linear regression performs better on "normal" data.

```
summary(d$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.853   4.545   5.380   5.407   6.184   7.769
```

```
ggplot(d, aes(score)) +
  geom_histogram(bins = 10) +
  ggtitle("Histogram of Happiness Score") +
  xlab("\nScore") +
  ylab("# of Countries\n")
```



#VOY $ gpd_pc : num 0.35 0.947 1.002 1.092 0.85 ... $ social_support : num 0.517 0.848 1.16 1.432 1.055 ... $ life_expectancy: num 0.361 0.874 0.785 0.881 0.815 ... $ freedom : num 0 0.383 0.086 0.471 0.283 0.557 0.532 0.351 0.536 0.527 ... $ generosity : num 0.158 0.178 0.073 0.066 0.095 0.332 0.244 0.035 0.255 0.166 ... $ corruption : num 0.025 0.027 0.114 0.05 0.064 0.29 0.226 0.182 0.11 0.143 ... $ continent : chr "Asia" "Europe_CEE" "Africa" "South America" ...

```
#kdepairs(d[,3:9])

#describe(d)

#summary(d)
```

Correlations

Methods

Results

list top and bottom 10 countries happines by continent plot

Conclusion

References

https://datahub.io/JohnSnowLabs/country-and-continent-codes-list#resource-country-and-continent-codes-list-csv

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

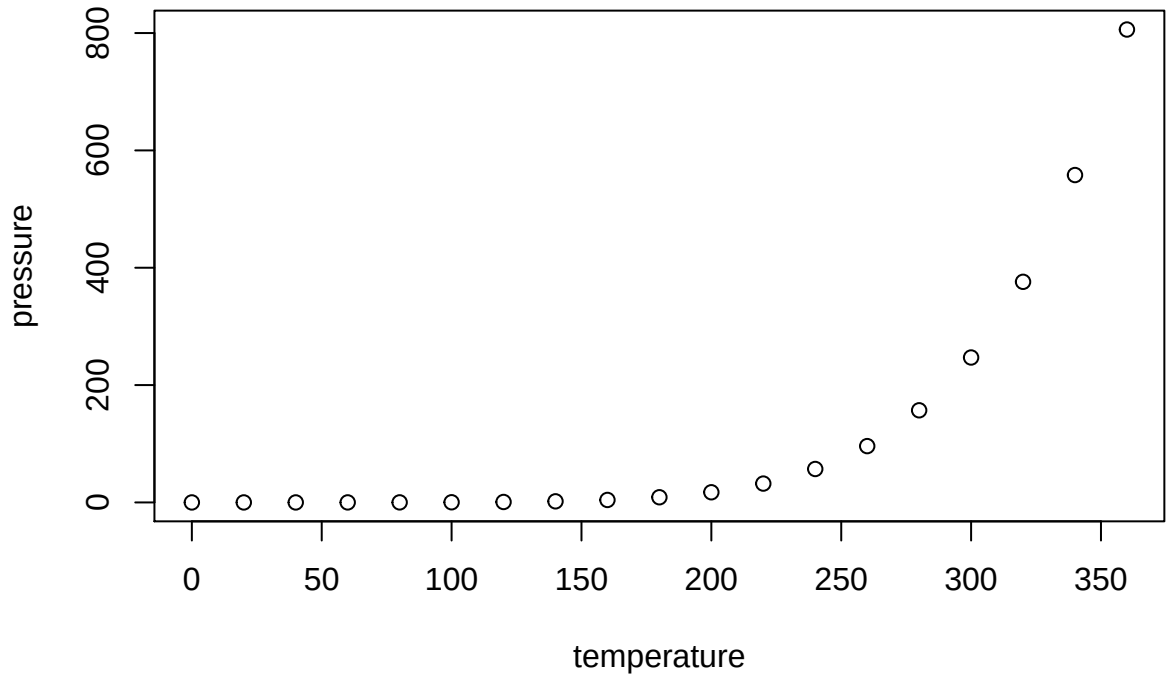
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.