Data project 3: World happiness

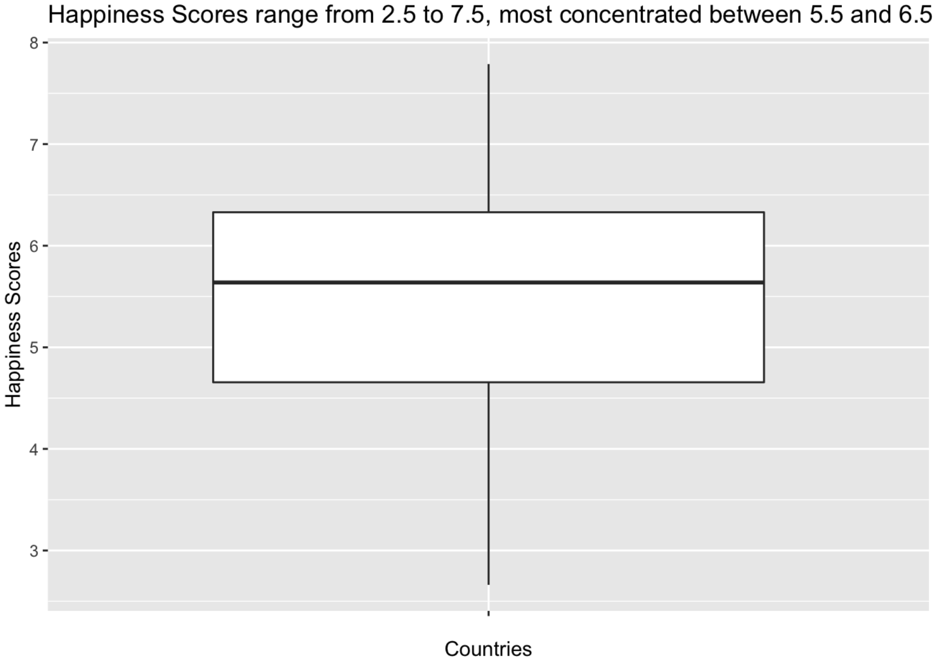
Jilli Violano

```
library(tidyverse)
happiness_train <- read_csv("happiness_train.csv")
happiness_test <- read_csv("happiness_test.csv")
```

Problem 1: Exploring and explaining the data

Part (a)

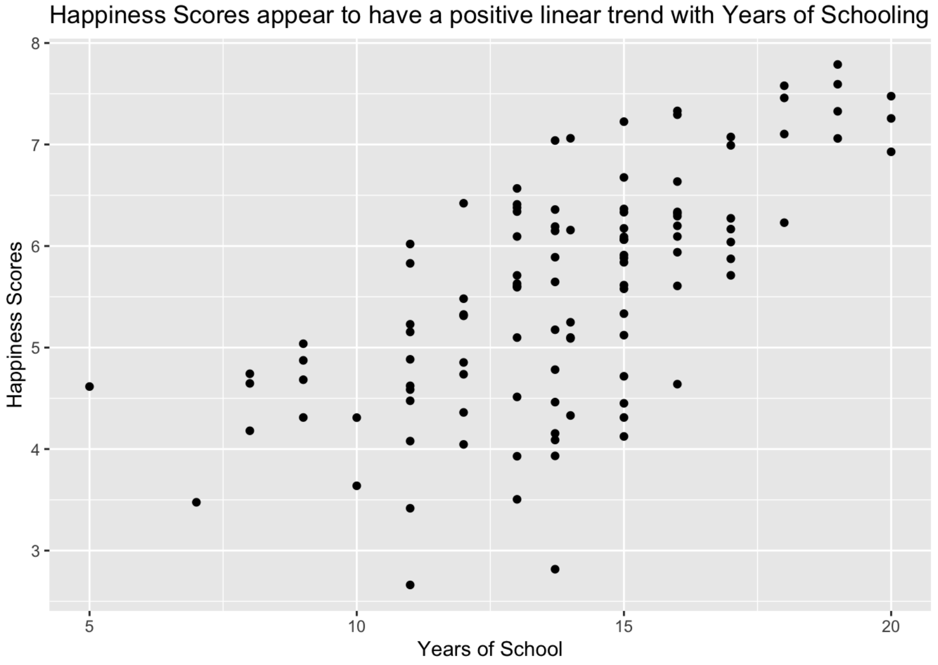
```
ggplot(happiness_train, aes(x='', y=world_happiness_score)) +
  geom_boxplot() +
  xlab('Countries') +
  ylab('Happiness Scores') +
  labs(title = 'Happiness Scores range from 2.5 to 7.5, most concentrated between 5.5 and 6.5')
```



From the box plot visualization of the happiness scores of all 110 countries, this viewer is able to see the range of values as well as where the data is most concentrated, giving them a good understanding of where their happiness score ranks in comparison to other countries.

Part (b)

```
ggplot(happiness_train, aes(x=school_years, y=world_happiness_score)) +
  geom_point() +
  xlab('Years of School') +
  ylab('Happiness Scores') +
  labs(title = 'Happiness Scores appear to have a positive linear trend with Years of Schooling')
```



The viewer should take away that there is a correlation between how many years an individual is in school for and their happiness score, as it has a generally positive trend. I think that these figures have a positive linear relationship, however due to the range of values, I expect there to be high error, especially in the 10-15 years of school range. This definitely makes sense though, since pre-University tends to last until 12th year, which explains the uptick in happiness (as they can get a job with that diploma), and further more, bachelor degrees typically last 4 years, so we see an uptick in the 16th year of schooling, once again, since it opens more doors to be taken with job opportunities.

Part (c)

~~Please refer to the analysis at the end of this document. It is going to take up a lot of space because I made a little EDA script, and it would take up a bit too much space during this document. (The EDA scripy is my own work made in an internship that I was in, and they granted me the rights to keep it. I attest that this is my own work and not any else's work. Originally written in python for Captiv8.io)~~

Some key takeaways that I noticed was that there were clear positive linear relationships with the happiness score, which include but are not limited to HDI index, sustainable economic development index, school year, political stability, government effectiveness, rule of law, regulatory quality, property rights score, and overall economic freedom score (referring to scatter plots). This surprised me especially since most have to do with having a government that is in control of its people, but not so much so it dictates what they spend and what they do with their own land. The idea that government gives you freedom of choice while also giving you the ability to live in peace with an effective law enforcement and judicial system. School year, as we mentioned in 1b as well gives us insight to the type of jobs we may see as well, since generally the more schooling one has, the higher the amount of money they make, allowing an individual to live more comfortably, as opposed to someone who is making a minimum wage job due to lack of education (if their country even has a set minimum wage, which some do not).

Another thing I also noticed was the population was very concentrated around 0-2 e8, with several countries going beyond that. This can give us insight on how gdp vs gdp per capita can be extrapolated, especially since they give us completely different values (overall wealth contributed by a country vs overall wealth on average contributed by each person in a country). Population can also affect rates of crime and property rights as if a smaller country (by area of inhabitable land).

Problem 2: Fitting linear regression models

Part (a)

The formula that we will use is the Least Squares formula for linear regression with multiple inputs. $\widehat{\beta} = (X^T X)^{-1} X^T Y$ After some calculation, we get the formula

$$\widehat{Y} = 2.02 + 4.81(\text{hdi\_index}) + 0.0002(\text{health\_expenditure\_per\_person}) - 0.044(\text{school\_years}) + 0.0004846084(\text{govt\_spending}) + 0.01(\text{women\_m}$$

```
X <- matrix(c(rep(1, length(happiness_train$country)), happiness_train$hdi_index, happiness_train$health_expendit
ure_per_person, happiness_train$school_years, happiness_train$government_spending_score, happiness_train$women_mp
s_pct), ncol = 6)
Y <- happiness_train$world_happiness_score
coefficents <- solve((t(X)%*%X))%*%(t(X))%*%(Y)
coefficents
```

```
##               [,1]
## [1,]  2.0201515740
## [2,]  4.8104636307
## [3,]  0.0001912149
## [4,] -0.0441369396
## [5,]  0.0004846084
## [6,]  0.0124361914
```

```
lmA=lm((world_happiness_score ~ hdi_index + health_expenditure_per_person + school_years + government_spending_sc
ore +women_mps_pct), data = happiness_train)
lmA
```

```
## 
## Call:
## lm(formula = (world_happiness_score ~ hdi_index + health_expenditure_per_person + 
##     school_years + government_spending_score + women_mps_pct), 
##     data = happiness_train)
## 
## Coefficients:
##                   (Intercept)                      hdi_index  
##                     2.0201516                      4.8104636  
## health_expenditure_per_person                   school_years  
##                     0.0001912                     -0.0441369  
##     government_spending_score                  women_mps_pct  
##                     0.0004846                      0.0124362
```

Part (b)

```
corl <- cor(select(happiness_train, -country))
for(i in seq(1, sqrt(length(corl)))) {
  for(j in seq(1, sqrt(length(corl)))){
    if((((corl[i,j]) >= 0.9 & corl[i,j] <= 1) | ((corl[i,j]) <= -0.9 & corl[i,j] >= -1)) & colnames(corl)[i] != c
olnames(corl)[j]){
      print('')
      print(colnames(corl)[i])
      print(rownames(corl)[j])
      print(corl[i,j])
    }
  }
}
```

```
## [1] ""
## [1] "government_spending_score"
## [1] "government_expenditure_pct_gdp"
## [1] -0.9632987
## [1] ""
## [1] "government_expenditure_pct_gdp"
## [1] "government_spending_score"
## [1] -0.9632987
## [1] ""
## [1] "political_rights_score"
## [1] "civil_liberties_score"
## [1] 0.9406639
## [1] ""
## [1] "civil_liberties_score"
## [1] "political_rights_score"
## [1] 0.9406639
## [1] ""
## [1] "government_effectiveness"
## [1] "regulatory_quality"
## [1] 0.9514886
## [1] ""
## [1] "government_effectiveness"
## [1] "rule_of_law"
## [1] 0.9615248
## [1] ""
## [1] "regulatory_quality"
## [1] "government_effectiveness"
## [1] 0.9514886
## [1] ""
## [1] "regulatory_quality"
## [1] "rule_of_law"
## [1] 0.9532886
## [1] ""
## [1] "rule_of_law"
## [1] "government_effectiveness"
## [1] 0.9615248
## [1] ""
## [1] "rule_of_law"
## [1] "regulatory_quality"
## [1] 0.9532886
```

```
remove = c('government_spending_score', 'civil_liberties_score', 'regulatory_quality', 'rule_of_law')
happiness_train2 = select(happiness_train, -remove)
```

```
## Note: Using an external vector in selections is ambiguous.
## ℹ Use `all_of(remove)` instead of `remove` to silence this message.
## ℹ See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

Upon this quick computation, we can see that government_expenditure_pct_gdp/government_spending_score, civil_liberties_score/political_rights_score, and regulatory_quality/rule_of_law/government_effectiveness are all correlated over .9 or under -.9 with each other in the four pairs. I will keep govt expenditure of gdp percentage, political rights score, and government effectiveness, as they tend to encapsulate their correlated pair within its description.

The reason we want to reduce co-linearity as much as we can because this can really mess with our hypothesis tests and receive a lot of redundant information. This is usually due to a lack of data, combination of existing data, or scaled data of an existing variable.

Part (c)

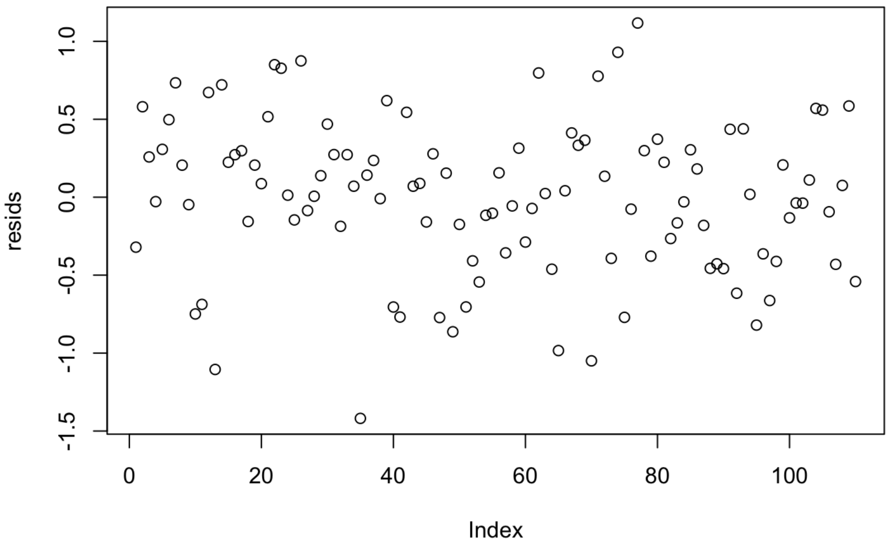
```
coeff = lm(world_happiness_score ~ ., data = happiness_train2[-1])
X <- data.matrix(select(happiness_train2, c(-country, -world_happiness_score)))
Y <- happiness_train2$world_happiness_score
np = length(Y) - 26
np
```

```
## [1] 84
```

```
for(i in seq(1, 26)){
  top = as.numeric(coeff$coef[i+1])
  bottom = norm(Y-(X[,i]*as.numeric(coeff$coef[i+1])),  type="2")**2/np
  test =  top/sqrt(bottom)
  p = 2*pt(abs(test), df = np, lower.tail=FALSE)
  if (p < .10) {
  print(coeff$coef[i+1])
  print(test)
  print(p) }
}
```

Part (d)

```
resids = resid(coeff)
plot(resids)
```



The homoskedasticity assessment of these coefficients seems to hold no decernable pattern (being as that is what we want to see), which tells us how important these values are overall to the hypothesis testing and thus validates our findings in C.

Part (e)

```
switzA = select(filter(happiness_test, country == 'Switzerland'), c(hdi_index, health_expenditure_per_person, sch
ool_years, government_spending_score, women_mps_pct))

scoreA = as.numeric(lmA$coef[1])
for(i in seq(1, 5)){
  scoreA = scoreA + as.numeric(lmA$coef[i+1]*switzA[i])
}
print("Score if we use part A")
```

```
## [1] "Score if we use part A"
```

```
scoreA
```

```
## [1] 7.740716
```

```
switzC = select(filter(happiness_test, country == 'Switzerland'), c(-country, -world_happiness_score, -remove))
scoreC = as.numeric(coeff$coef[1])
for(i in seq(1, 26)){
  scoreC = scoreC + as.numeric(coeff$coef[i+1]*switzC[i])
}
print("Score if we use part C")
```

```
## [1] "Score if we use part C"
```

```
scoreC
```

```
## [1] 7.50421
```

Part (f)

```
predictedA = 0
for(j in happiness_test$country){
countr = select(filter(happiness_test, country == j), c(hdi_index, health_expenditure_per_person, school_years, government_spending_score, women_mps_pct))
scoref = as.numeric(lmA$coef[1])
for(i in seq(1, 5)){
  scoref = scoref + as.numeric(lmA$coef[i+1])*as.numeric(countr[i])
}
predictedA = c(predictedA, scoref)
}
print("average error for A")
```

```
## [1] "average error for A"
```

```
print(sum(abs(predictedA[-1] - happiness_test$world_happiness_score))/length(happiness_test$country))
```

```
## [1] 0.5599476
```

```
predictedC = 0
for(j in happiness_test$country){
countr = select(filter(happiness_test, country == j), c(-country, -world_happiness_score, -remove))
scoref = as.numeric(coeff$coef[1])
for(i in seq(1, 26)){
  scoref = scoref + as.numeric(coeff$coef[i+1]*countr[i])
}
predictedC = c(predictedC, scoref)
}
print("average error for C")
```

```
## [1] "average error for C"
```

```
print(sum(abs(predictedC[-1] - happiness_test$world_happiness_score))/length(happiness_test$country))
```

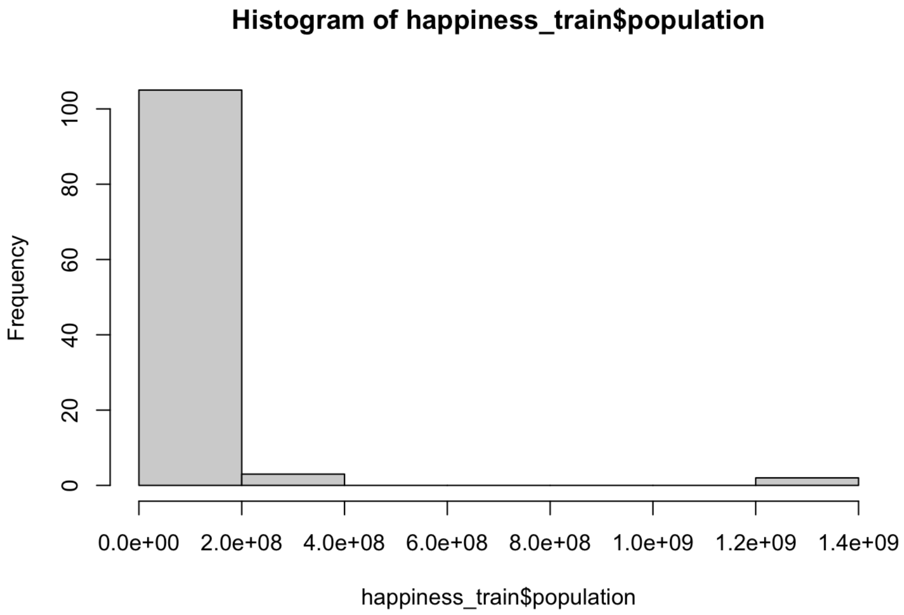
```
## [1] 0.5161896
```

Upon looking at these values, since i took the absolute sum of all the countries, then divided then by the amount of countries, I was able to find the average error of both part A's and part C's. C's preformed slightly better, about .05 avg error in comparison to A. Being off about half a point regardless is not too bad, especially if we plan on predicting other countries with this model or even future years of the countries to see if much is expected to change based on said value.

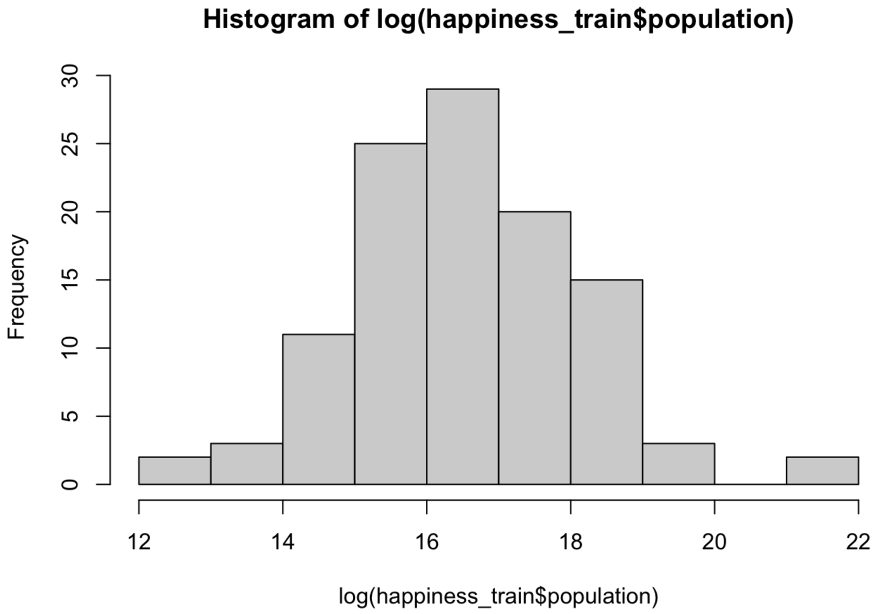
Part (g)

From evaluating the individual histograms, I was considering log the population, gdp, area, and education expenditure per person, to get a more accurate distribution that is not as tight. If we examine the two graphs below, they both give the same information, but give us a better and more distinct distribution that we can take advantage of.

```
hist(happiness_train$population)
```



```
hist(log(happiness_train$population))
```



```
logs = c("population", "gdp_billions", "area", 'education_expenditure_per_person')

happiness_train2$populationL=log(happiness_train2$population)
happiness_train2$gdp_billionsL=log(happiness_train2$gdp_billions)
happiness_train2$areaL=log(happiness_train2$area)
happiness_train2$education_expenditure_per_personL=log(happiness_train2$education_expenditure_per_person)

happiness_train3 = select(happiness_train2, -logs)
```

```
## Note: Using an external vector in selections is ambiguous.
## ℹ Use `all_of(logs)` instead of `logs` to silence this message.
## ℹ See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
lmG = lm(world_happiness_score ~ ., data = happiness_train3[-1])

happiness_test$populationL=log(happiness_test$population)
happiness_test$gdp_billionsL=log(happiness_test$gdp_billions)
happiness_test$areaL=log(happiness_test$area)
happiness_test$education_expenditure_per_personL=log(happiness_test$education_expenditure_per_person)

happiness_test2 = select(happiness_test, -logs)


predictedG = 0
for(j in happiness_test2$country){
countr = select(filter(happiness_test2, country == j), c(-country, -world_happiness_score, -remove))
scoreg = as.numeric(lmG$coef[1])
for(i in seq(1, 26)){
  scoreg = scoreg + as.numeric(lmG$coef[i+1]*countr[i])
}
predictedG = c(predictedG, scoreg)
}
print("average error for G")
```

```
## [1] "average error for G"
```

```
print(sum(abs(predictedG[-1] - happiness_test2$world_happiness_score))/length(happiness_test2$country))
```

```
## [1] 0.5020854
```

By adjusting both the training and test sets to accept log of population, gdp, area, and education expenditure per person rather than their original values, we were able to train our set to have an average of .50 error average, and although its not much better, it did help improve the value.

Problem 3: The distribution of β

Part (a)

```
lm3a = lm(world_happiness_score ~ hdi_index, data = happiness_train )
lm3a
```

```
##
## Call:
## lm(formula = world_happiness_score ~ hdi_index, data = happiness_train)
##
## Coefficients:
## (Intercept)    hdi_index
##       1.176        5.965
```

```
x = happiness_train$hdi_index
vB0 = (var(x)* sum(x**2)) / (length(x)*sum(x**2) - ((sum(x))**2))
print("variance of B0")
```

```
## [1] "variance of B0"
```

```
vB0
```

```
## [1] 0.00516312
```

```
vB1 = (var(x)*length(n)) / (length(x)*sum(x**2) - ((sum(x))**2))
print("variance of B1")
```

```
## [1] "variance of B1"
```

```
vB1
```

```
## [1] 8.340284e-05
```