STAT 340 Final Report

12/18/2021

Contents

Names/NetID:

- Justin Chan: jachan
- Tambre Hu: thu53
- Oat Sukcharoenyingyong: sukcharoenyi
- James Ma: yma255
- Shawn Riemer: seriemer

Abstract

An abstract of at most 300 words giving an overview of your data, the question(s) you tried to answer, and a brief summary of your findings.

The data that we analyzed throughout the semester is the World Development Indicators dataset, collected by the World Bank and published to Kaggle. This dataset includes 1,344 different indicators for 247 countries between 1960 to 2015, although much of the data is missing. The indicators cover a wide range of economic, health, and assorted other metrics. We chose to...

(if focusing on statistical question 1) ...investgate these indicators to find how they impact a country's economic strength, which we defined as GDP per capita. We first refined the data we woked with to only include _____ countries, _____ indicators, and _____ years. We selected those subsets so that we could focus on _____ without missing too much data. The models that we used were _____ for **,** for ***, and*** for _____. We found that...

Dataset

> A brief introduction describing your data set: where it came from, who gathered it and a brief description of why your group found this data interesting and why your reader should care. If you obtained your data set from Kaggle or a similar repository, it is not enough to simply say "This was collected by user xyz and posted to Kaggle". You should be able to explain the specific source of your data.

Our analysis used the World Development Indicators dataset (link). We accessed the dataset from Kaggle which included a cleaned version of the raw data from The World Bank (link). It includes two files. The first is Country.csv, which includes a row for each of the 247 countries present in the dataset, along with the country's name, code, currency, and region. The second file, Indicators.csv, contains a comprehensive list of over 1,300 indicators related to economic development, health, and more. Some of these indicators include GDP per capita, unemployment rate, adolescent fertility rate, and urban population. Indicators for years between 1960 and 2015 are present, although not every combination of country, indicator, and year is available. More than 70% of the total data is missing, so careful consideration will be taken when choosing what countries, indicators, and years to analyze.

The data was compiled by the World Bank, an international financial organization that provides support for lower-income countries. They track the indicators found in this dataset and compile them for anayone to use. Our group was drawn to this dataset because of its importance and relevance. The indicators included comprise some of the most discussed and debated inequalities between countries. While we cannot solve these inequalities just by looking at data, we are excited to investigate and identify prominent relationships between features.

Variables

> A description of the variables available in your data set, with more details for the variables that are likely to be relevant to your question (you may copy-paste this from your previous drafts, if you wish).

With 1,344 total unique indicators there are too many to name them all, but we will focus on a handful of them. For our economic analysis, we will use `GDP_per_capita_(current_US$)`, which is a country's economic output per citizen adjusted to the United States Dollar currency. There are many variations of GDP in the dataset, but GDP per capita is commonly used to compare the economic states of different countries, so we will use that one. A couple of variables that we expect impact a country's economic strenght are unemployment rate, `Unemployment,_total_(%_of_total_labor_force)`, and relative military expenditure, `Military_expenditure_(%_of_GDP)`.

Of the health indicators, one variable we will look at is `Adolescent_fertility_rate_(births_per_1,000_women_ages_15-19` Adolescent fertility rate is the rate of births per 1,000 women aged 15-19 years old, and is often used as a healthcare metric, so we expect it to be related to GDP per capita. Another indicator that we expect to impact GDP per capita is `Population_ages_65_and_above_(%_of_total)`, the percent percent of citizens 65 years or older. Other metrics related to health that we expect to find important include those related to health expenditure, `Health_expenditure,_total_(%_of_GDP)`, and life expectancy, `Life_expectancy_at_birth,_total_(years)`.

Many other assorted metrics are included in the dataset. Environmental metrics such as CO2 emmisions, `CO2_emissions_(metric_tons_per_capita)`, will be interesting to investigate against economic metrics. Other indicators we expect to correlate with economic strength are internet users, `Internet_users_(per_100_people)`, and urban population, `Urban_population_(%_of_total)`, although plenty of indicators not mentioned here will also be analyzed.

Statistical Question

> A discussion of your statistical question of interest.

The vast amount of data meant that there was a lot to explore, but we decided to focus on answering the following question:

---

1. What indicators have the greatest impact on a country's GDP per capita?

a. Are there relationships between those indicators?

2. Can we use these indicators to predict a country's GDP per capita?

a. How does that compare with other prediction methods?

For our first statistical question, given the various metrics we have in our data, we thought it would be interesting to see how different factors impact a country's GDP per capita since that is a relatively prominent indicator of a country's economic health. We chose to use GDP as the deciding metric for comparison since it gives information about the size of the economy and how an economy is performing. The growth rate of GDP is often used as an indicator of the general health of the economy, so in a broader sense, an increase in real GDP is interpreted as a sign that the economy is doing well. We also thought figuring out if there are any relationships between said metrics would be interesting because along with knowing what indicators have the greatest impact on a country's GDP, being able to pinpoint how those indicators are related and if they're all part of a certain category, for instance, if the most impactful metrics were all related to healthcare, we would be able to gain a more general insight on what general system of a country impacts it's GDP and economy.

Our second statistical question delves into the relationship between the indicators we are looking for and GDP. We wanted to find out if we could use the indicators we found to predict a country's GDP because, hypothetically, we could be able to use the most impactful metrics that impact a country's economy to predict how much, and to what extent that impact would be. Along with this, since the end goal of us asking this question is to derive a successful GDP model predictor, we also wanted to look into if using the most impactful indicators is the best metric for prediction by comparing the results of that model with other prediction methods.

Analysis

> A summary, including as many plots, tables, R model outputs, etc as necessary to show how you tried to answer your statistical question. You should explain the reasoning behind your decisions to use particular methods, models or tests. This summary need not include only successes– if you tried something that turned out not to work, take some time to discuss it, why it might not have worked, and how your might fix the problem given more time.

The first step of our analysis was refining our dataset to get it into a form where we could easily implement our models. *Paragraph explaining how we refine dataset*

Paragraph on model/test 1

PCA, kmeans, clustering

Doing PCA and kmeans

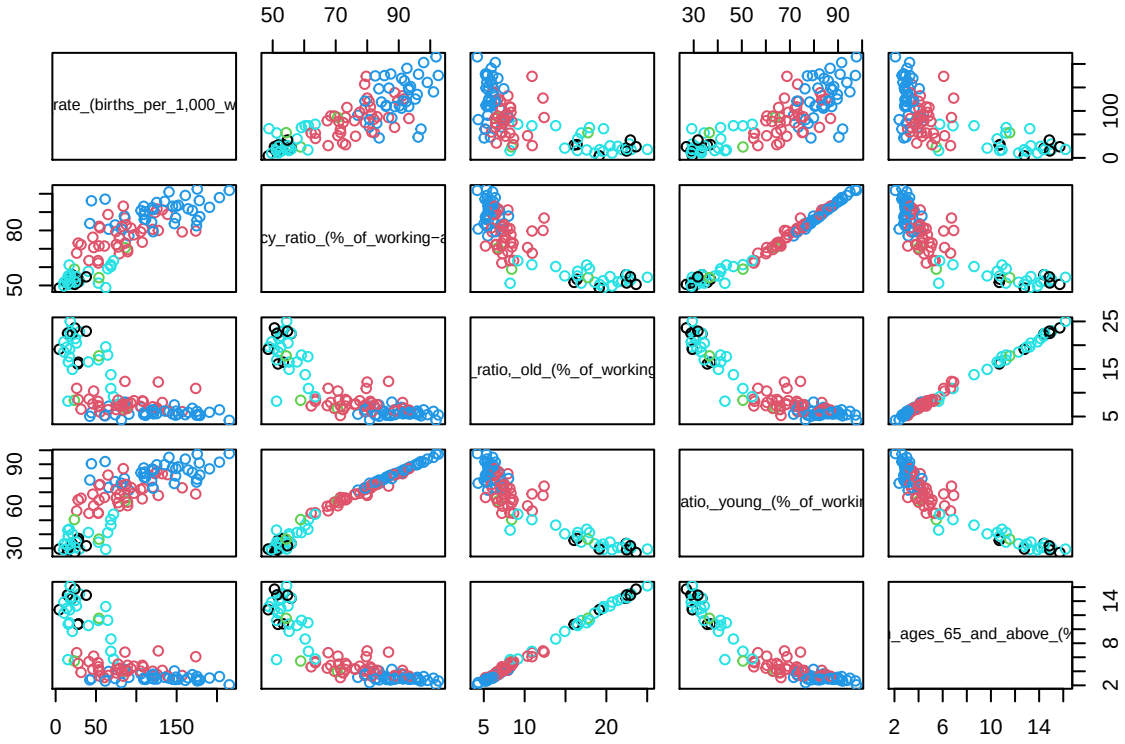
```
# At this point the data frame we want to use is called `ind5`. Doing PCA and kmeans clustering.
gdp_per_capita_current_US <- ind5 %>%
  select("GDP_per_capita_(current_US$)")
predictors <- ind5 %>%
  select(-"GDP_per_capita_(current_US$)", -"CountryName")

pc3 <- prcomp(predictors, scale = TRUE, rank. = 2)
```

```r
km3 <- kmeans(pc3$x, centers = 5, nstart = 10)

pcs <- pc3$x %>% data.frame()

foo <-
  cbind(ind5, pcs) %>% select(CountryName, PC1, PC2, everything())

foo$Cluster = km3$cluster

# foo %>%
#   select(contains("GDP"),
#          contains("GNI"))

foo <- foo %>% select(CountryName, Cluster, PC1, PC2, everything())
```

Pairs plot

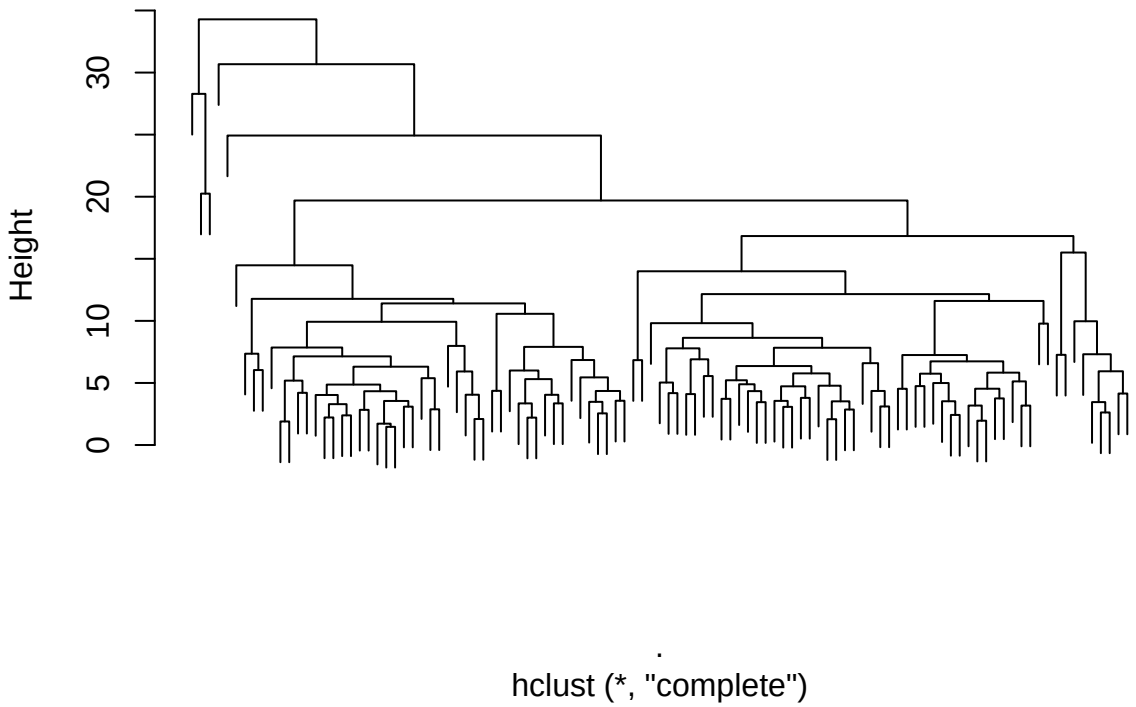
Pairs plot with arbitrarily chosen indicators to show clustering:

```r
# Pairs plot
pairs(foo[, 5:9], col = foo$Cluster)
```



Hierarchical clustering

```r
# heir clustering
foo[, 5:75] %>% scale() %>% data.frame() %>% dist() %>% hclust() %>% plot(labels = FALSE)
```



Linear model on PCA

```
# Linear model using PCA
y <- gdp_per_capita_current_US
x <- prcomp(predictors, scale = TRUE, rank. = 5)$x %>% data.frame()
goo <- cbind(y, x) %>% rename(gdp_usd = `GDP_per_capita_(current_US$)`)
model <- lm(gdp_usd ~ PC1 + PC2 + PC3 + PC4 + PC5, goo)
# model %>% summary()
```

```
n = 107
k = 5
grp_sz <- floor(n / k)

# want to split x and y into grp_sz
x <- prcomp(predictors, scale = TRUE, rank. = 5)$x %>% data.frame()
y <- gdp_per_capita_current_US

## FUNCTION (x, y)
k_ = 1
xs <- list()
ys <- list()
for (i in 1:n) {
  if ((i-1) %% grp_sz == 0) {
    if ((i + (grp_sz*2)) >= n) {
      gr_x <- x[i:(n),]
      gr_y <- y[i:(n),]
    } else {
      gr_x <- x[i:(i+grp_sz),]
```

```
      gr_y <- y[i:(i+grp_sz),]
    }
    xs[[k_]] <- gr_x
    ys[[k_]] <- gr_y
    k_ <- k_ + 1
  }
}

xs <- xs[1:length(xs)-1]
ys <- ys[1:length(ys)-1]

# for (gr in xs[2:length(xs)]) {
models <- list()
for (i in 1:length(xs)) {
  xs_ <- data.frame()
  ys_ <- data.frame()
  for (j in 1:length(xs)) {
    if (i != j) {
      xs_ <- bind_rows(xs_, xs[[j]])
      ys_ <- bind_rows(ys_, ys[[j]])
    }
  }
  x_ <- xs_
  y_ <- ys_
  goo <- cbind(y_, x_) %>% rename(gdp_usd = `GDP_per_capita_(current_US$)`)
  model <- lm(gdp_usd ~ PC1 + PC2 + PC3 + PC4 + PC5, goo)
  models[[i]] <- model
}
sqr <- function (x) {
  x * x
}

accr2 <- 0
acc <- 0
for (i in 1:length(xs)) {
  m <- models[[i]]
  foo <- cbind(predict(m, xs[[i]]) %>% data.frame(), ys[[i]]) %>%
  mutate(
    res = . - `GDP_per_capita_(current_US$)`,
    rse = res^2,
    rss = res ^ 2,
    tss = (. - mean(`GDP_per_capita_(current_US$)`))^2
  )
  r2 <- sum(foo$rss) / sum(foo$tss)
  acc <- acc + (foo$rse %>% mean())
  accr2 <- accr2 + r2

}

tibble(
  r2 = (r2 <- (accr2 / (models %>% length()))),
  mse = (acc / (models %>% length())),
  rmse = (acc / (models %>% length())) %>% sqrt()
```

```
)
```

```
## # A tibble: 1 x 3
##      r2       mse  rmse
##   <dbl>     <dbl> <dbl>
## 1 0.501 22776755. 4772.
```

Model/test 2

Paragraph on model/test 2

```
# grid <- 10 ^ seq(10,-2, length = 100)
grid <- c(0, 1, 2, 5, 10, 100, 500, 1000)

results <- tibble(
  lambda=-1, mse=-1, rmse=-1, r2=-1
)

for (lambda in grid) {
  # lambda = 5
  n = 107
  k = 5
  grp_sz <- floor(n / k)

  # want to split x and y into grp_sz
  x <- prcomp(predictors, scale = TRUE, rank. = 5)$x %>% data.frame()
  y <- gdp_per_capita_current_US

  ## FUNCTION (x, y)
  k_ = 1
  xs <- list()
  ys <- list()
  for (i in 1:n) {
    if ((i - 1) %% grp_sz == 0) {
      if ((i + (grp_sz * 2)) >= n) {
        gr_x <- x[i:(n), ]
        gr_y <- y[i:(n), ]
      } else {
        gr_x <- x[i:(i + grp_sz), ]
        gr_y <- y[i:(i + grp_sz), ]
      }
      xs[[k_]] <- gr_x
      ys[[k_]] <- gr_y
      k_ <- k_ + 1
    }
  }

  xs <- xs[1:length(xs) - 1]
  ys <- ys[1:length(ys) - 1]

  # for (gr in xs[2:length(xs)]) {
  models <- list()
  for (i in 1:length(xs)) {
    xs_ <- data.frame()
```

```
    ys_ <- data.frame()
    for (j in 1:length(xs)) {
      if (i != j) {
        xs_ <- bind_rows(xs_, xs[[j]])
        ys_ <- bind_rows(ys_, ys[[j]])
      }
    }
    x_ <- xs_
    y_ <- ys_
    goo <-
      cbind(y_, x_) %>% rename(gdp_usd = `GDP_per_capita_(current_US$)`)
    model <-
      glmnet(x_ %>% as.matrix(),
             y_ %>% as.matrix(),
             alpha = 1,
             lambda = lambda)
    models[[i]] <- model
  }
  sqr <- function (x) {
    x * x
  }

  acc <- 0
  accr2 <- 0
  for (i in 1:length(xs)) {
    m <- models[[i]]
    foo <-
      cbind(predict(m, s = lambda, newx = xs[[i]] %>% as.matrix()) %>% data.frame(),
            ys[[i]]) %>%
      mutate(
        res = s1 - `GDP_per_capita_(current_US$)`,
        rse = res ^ 2,
        rss = res ^ 2,
        tss = (s1 - mean(`GDP_per_capita_(current_US$)`))^2
        )
    r2 <- sum(foo$rss) / sum(foo$tss)
    rse <- (foo$rse %>% mean())
    accr2 <- accr2 + r2
    acc <- acc + rse
  }
  mse <- (acc / (models %>% length()))
  rmse <- (acc / (models %>% length())) %>% sqrt()
  r2 <- (accr2 / (models %>% length()))

  results <- results %>% rbind(
    c(lambda, rmse, mse, r2)
  )
}
results <- results %>% filter(lambda != -1)
results %>% arrange(-mse)
```

```
## # A tibble: 8 x 4
##   lambda   mse      rmse    r2
##    <dbl> <dbl>     <dbl> <dbl>
```

```
## 1   1000 4985. 24849698. 0.948
## 2    500 4797. 23007333. 0.680
## 3      0 4773. 22778879. 0.501
## 4      1 4772. 22774375. 0.502
## 5      2 4772. 22769898. 0.502
## 6      5 4770. 22756628. 0.503
## 7     10 4768. 22735046. 0.504
## 8    100 4739. 22462755. 0.528
```

```
mse_results <- results %>% arrange(-mse) %>% tail(1)
r2_results <- results %>% arrange(r2) %>% tail(1)

mse_results
```

```
## # A tibble: 1 x 4
##   lambda   mse      rmse    r2
##    <dbl> <dbl>     <dbl> <dbl>
## 1    100 4739. 22462755. 0.528
```

```
r2_results
```

```
## # A tibble: 1 x 4
##   lambda   mse      rmse    r2
##    <dbl> <dbl>     <dbl> <dbl>
## 1   1000 4985. 24849698. 0.948
```

```
x <- ind5 %>% select(-`GDP_per_capita_(current_US$)`, -`CountryName`) %>% as.matrix()
y <- ind5$`GDP_per_capita_(current_US$)` %>% as.matrix()
out <- glmnet(x, y, alpha=1, lambda=mse_results$lambda)
lasso.coef <- predict(out, type="coefficients", s=mse_results$lambda)
lasso.coef2 <- predict(out, type="coefficients", s=r2_results$lambda)
lasso.coef
```

```
## 71 x 1 sparse Matrix of class "dgCMatrix"
##                                                                                    s
## (Intercept)                                                                -1.186254e+04
## Adolescent_fertility_rate_(births_per_1,000_women_ages_15-19)              -1.416810e+01
## Age_dependency_ratio_(%_of_working-age_population)                          .
## Age_dependency_ratio,_old_(%_of_working-age_population)                     5.466263e+01
## Age_dependency_ratio,_young_(%_of_working-age_population)                   .
## Population_ages_65_and_above_(%_of_total)                                   4.501267e+01
## Population,_ages_0-14_(%_of_total)                                          .
## Population,_ages_15-64_(%_of_total)                                         .
## Population,_female_(%_of_total)                                             .
## Population,_total                                                           .
## Rural_population                                                            .
## Rural_population_(%_of_total_population)                                    .
## Urban_population                                                            .
## Urban_population_(%_of_total)                                               .
## Urban_population_growth_(annual_%)                                          5.305003e+01
## Population_growth_(annual_%)                                                8.211135e+01
## Rural_population_growth_(annual_%)                                          2.339027e+01
## Merchandise_exports_(current_US$)                                           5.353124e-09
## Merchandise_imports_(current_US$)                                           .
## Birth_rate,_crude_(per_1,000_people)                                        .
## Death_rate,_crude_(per_1,000_people)                                        3.734714e+01
```

```
## Life_expectancy_at_birth,_female_(years)                                                .
## Life_expectancy_at_birth,_male_(years)                                                  .
## Life_expectancy_at_birth,_total_(years)                                                 .
## Survival_to_age_65,_female_(%_of_cohort)                                                .
## Survival_to_age_65,_male_(%_of_cohort)                                                  .
## Fertility_rate,_total_(births_per_woman)                                                .
## Land_area_(sq._km)                                                                      .
## Population_density_(people_per_sq._km_of_land_area)                                     1.786434e-0
## Surface_area_(sq._km)                                                                   .
## Mortality_rate,_adult,_female_(per_1,000_female_adults)                                 .
## Mortality_rate,_adult,_male_(per_1,000_male_adults)                                     .
## Mortality_rate,_infant_(per_1,000_live_births)                                          .
## Mortality_rate,_under-5_(per_1,000)                                                     .
## Number_of_infant_deaths                                                                 .
## Number_of_under-five_deaths                                                             4.544230e-04
## DEC_alternative_conversion_factor_(LCU_per_US$)                                         .
## Agricultural_land_(%_of_land_area)                                                     -1.844513e+0
## Agricultural_land_(sq._km)                                                              .
## Arable_land_(%_of_land_area)                                                           -1.087259e+0
## Arable_land_(hectares)                                                                  .
## Arable_land_(hectares_per_person)                                                       9.221247e+0
## Crop_production_index_(2004-2006_=_100)                                                 2.214619e+0
## Food_production_index_(2004-2006_=_100)                                                 .
## Livestock_production_index_(2004-2006_=_100)                                            .
## Cereal_production_(metric_tons)                                                         .
## Cereal_yield_(kg_per_hectare)                                                           7.059605e-0
## Land_under_cereal_production_(hectares)                                                 .
## Permanent_cropland_(%_of_land_area)                                                    -8.149958e+0
## Official_exchange_rate_(LCU_per_US$,_period_average)                                    .
## GDP_(current_LCU)                                                                       .
## GDP_per_capita_(current_LCU)                                                            .
## CO2_emissions_(kt)                                                                      .
## CO2_emissions_(metric_tons_per_capita)                                                  6.637975e+0
## CO2_emissions_from_solid_fuel_consumption_(kt)                                         -1.283613e-0
## CO2_emissions_from_solid_fuel_consumption_(%_of_total)                                  .
## CO2_emissions_from_liquid_fuel_consumption_(%_of_total)                                 3.226148e+0
## CO2_emissions_from_liquid_fuel_consumption_(kt)                                         .
## CO2_emissions_from_gaseous_fuel_consumption_(kt)                                        .
## CO2_emissions_from_gaseous_fuel_consumption_(%_of_total)                               -2.036777e+0
## GDP_at_market_prices_(current_US$)                                                      .
## Merchandise_trade_(%_of_GDP)                                                           -9.694197e-0
## Total_reserves_(includes_gold,_current_US$)                                             .
## Total_reserves_minus_gold_(current_US$)                                                 .
## GDP_(constant_LCU)                                                                      .
## GDP_per_capita_(constant_LCU)                                                           .
## GDP_deflator_(base_year_varies_by_country)                                              .
## Merchandise_exports_by_the_reporting_economy_(current_US$)                              .
## Merchandise_exports_by_the_reporting_economy,_residual_(%_of_total_merchandise_exports) -1.881005e+0
## GNI_(current_LCU)                                                                       .
## GNI_per_capita_(current_LCU)                                                            .
```

```
lasso.coef2
```

```
## 71 x 1 sparse Matrix of class "dgCMatrix"
##                                                                                          s
```

```
## (Intercept)                                                                  -1.186254e+04
## Adolescent_fertility_rate_(births_per_1,000_women_ages_15-19)                -1.416810e+01
## Age_dependency_ratio_(%_of_working-age_population)                            .
## Age_dependency_ratio,_old_(%_of_working-age_population)                       5.466263e+01
## Age_dependency_ratio,_young_(%_of_working-age_population)                     .
## Population_ages_65_and_above_(%_of_total)                                     4.501267e+01
## Population,_ages_0-14_(%_of_total)                                            .
## Population,_ages_15-64_(%_of_total)                                           .
## Population,_female_(%_of_total)                                               .
## Population,_total                                                             .
## Rural_population                                                              .
## Rural_population_(%_of_total_population)                                      .
## Urban_population                                                              .
## Urban_population_(%_of_total)                                                 .
## Urban_population_growth_(annual_%)                                            5.305003e+01
## Population_growth_(annual_%)                                                  8.211135e+01
## Rural_population_growth_(annual_%)                                            2.339027e+01
## Merchandise_exports_(current_US$)                                             5.353124e-09
## Merchandise_imports_(current_US$)                                             .
## Birth_rate,_crude_(per_1,000_people)                                          .
## Death_rate,_crude_(per_1,000_people)                                          3.734714e+01
## Life_expectancy_at_birth,_female_(years)                                      .
## Life_expectancy_at_birth,_male_(years)                                        .
## Life_expectancy_at_birth,_total_(years)                                       .
## Survival_to_age_65,_female_(%_of_cohort)                                      .
## Survival_to_age_65,_male_(%_of_cohort)                                        .
## Fertility_rate,_total_(births_per_woman)                                      .
## Land_area_(sq._km)                                                            .
## Population_density_(people_per_sq._km_of_land_area)                           1.786434e-01
## Surface_area_(sq._km)                                                         .
## Mortality_rate,_adult,_female_(per_1,000_female_adults)                       .
## Mortality_rate,_adult,_male_(per_1,000_male_adults)                           .
## Mortality_rate,_infant_(per_1,000_live_births)                                .
## Mortality_rate,_under-5_(per_1,000)                                           .
## Number_of_infant_deaths                                                       .
## Number_of_under-five_deaths                                                   4.544230e-04
## DEC_alternative_conversion_factor_(LCU_per_US$)                               .
## Agricultural_land_(%_of_land_area)                                           -1.844513e+01
## Agricultural_land_(sq._km)                                                    .
## Arable_land_(%_of_land_area)                                                 -1.087259e+01
## Arable_land_(hectares)                                                        .
## Arable_land_(hectares_per_person)                                             9.221247e+01
## Crop_production_index_(2004-2006_=_100)                                       2.214619e+01
## Food_production_index_(2004-2006_=_100)                                       .
## Livestock_production_index_(2004-2006_=_100)                                  .
## Cereal_production_(metric_tons)                                               .
## Cereal_yield_(kg_per_hectare)                                                 7.059605e-01
## Land_under_cereal_production_(hectares)                                       .
## Permanent_cropland_(%_of_land_area)                                          -8.149958e+01
## Official_exchange_rate_(LCU_per_US$,_period_average)                          .
## GDP_(current_LCU)                                                             .
## GDP_per_capita_(current_LCU)                                                  .
## CO2_emissions_(kt)                                                            .
## CO2_emissions_(metric_tons_per_capita)                                        6.637975e+01
```

```
## CO2_emissions_from_solid_fuel_consumption_(kt)                                    -1.283613e-03
## CO2_emissions_from_solid_fuel_consumption_(%_of_total)                             .
## CO2_emissions_from_liquid_fuel_consumption_(%_of_total)                            3.226148e+01
## CO2_emissions_from_liquid_fuel_consumption_(kt)                                    .
## CO2_emissions_from_gaseous_fuel_consumption_(kt)                                   .
## CO2_emissions_from_gaseous_fuel_consumption_(%_of_total)                          -2.036777e+01
## GDP_at_market_prices_(current_US$)                                                 .
## Merchandise_trade_(%_of_GDP)                                                      -9.694197e-01
## Total_reserves_(includes_gold,_current_US$)                                        .
## Total_reserves_minus_gold_(current_US$)                                            .
## GDP_(constant_LCU)                                                                 .
## GDP_per_capita_(constant_LCU)                                                      .
## GDP_deflator_(base_year_varies_by_country)                                         .
## Merchandise_exports_by_the_reporting_economy_(current_US$)                         .
## Merchandise_exports_by_the_reporting_economy,_residual_(%_of_total_merchandise_exports) -1.881005e+01
## GNI_(current_LCU)                                                                  .
## GNI_per_capita_(current_LCU)                                                       .
```

Model/test 3

Paragraph on model/test 3

Conclusion

A brief summary of your findings and potential future research questions

Paragraph summarizing our findings

Appendix

Data Cleaning

The problem

One challenge we faced was cleaning our data. The data was provided by The World Bank, which aggregated this data from multiple sources such as global studies or national surveys. Due to the nature of how our data was collected, there were many missing values in our dataset. We assume this is because national surveys are not standardized across countries and time, and most of the data sources didn't exist for the entirety of the time range of 1960 to 2015.

Take for example this code snippet. We look at our original data and find that given that each observation was uniquely identified by a `Country`, `Indicator`, `Year` combination, we have `68.68254` percent of our data is missing values. We didn't notice this at first because the missing data was implicit in the original dataset, there simply wouldn't be a `Country`, `Indicator`, `Year` combination if that value didn't exist for some year or country.

```r
# Calculating proportion of missing values:
p_missing_ <- function(d) {
  mvt <- table(is.na(d))
  mvt[2] / (mvt[1] + mvt[2])
}

ind2 <- ind2_orig
tibble(proportion_missing_values = p_missing_(ind2))
```

```
## # A tibble: 1 x 1
##   proportion_missing_values
##                       <dbl>
## 1                     0.687
```

In addition, almost all the indicators had this issue. Of the 1344 indicators, only 29 had less than 10% missing values and only 652 had less than 50% missing values.

```
max_percent_missing <- 0.1
n_ind_lt_10 <- sapply(ind2, function(x) sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
  arrange(-desc(.)) %>%
  filter(. / 13823 < max_percent_missing) %>% nrow()

ind2 <- ind2_orig
max_percent_missing <- 0.5
n_ind_lt_50 <- sapply(ind2, function(x) sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
  arrange(-desc(.)) %>%
  filter(. / 13823 < max_percent_missing) %>% nrow()

tibble(
  max_percent_missing = c(10, 50),
  num_indicators = c(n_ind_lt_10, n_ind_lt_50),
  proportion_of_indicators = c(n_ind_lt_10 / ind2 %>% nrow() * 10, n_ind_lt_50 / ind2 %>% nrow() * 10),
)
```

```
## # A tibble: 2 x 3
##   max_percent_missing num_indicators proportion_of_indicators
##                 <dbl>          <int>                    <dbl>
## 1                  10             29                   0.0285
## 2                  50            652                   0.640
```

No country had less than 50% of missing values.

```
n_year <- ind2$Year %>% unique() %>% length()
n_ind <- (ind2 %>% ncol() - 2) # CountryName and Year don't count
n_d <- n_ind * n_year

ind2 <- ind2_orig

obs_counts <- ind %>%
  group_by(CountryName) %>%
  summarise(
    n = n()
  )

ind2 %>%
  group_by(CountryName) %>%
  select(-Year) %>%
  summarise_all(funs(sum(is.na(.)))) %>%
  mutate(
    p_missing = select(., -CountryName) %>% rowSums() / n_d
  ) %>% select(
```

```
    CountryName,
    p_missing
  ) %>%
  full_join(obs_counts) %>%
  arrange(-desc(p_missing))
```

```
## Joining, by = "CountryName"
```

```
## # A tibble: 182 x 3
##    CountryName p_missing     n
##    <chr>           <dbl> <int>
##  1 Mexico          0.505 37244
##  2 Colombia        0.505 37227
##  3 Philippines     0.510 36912
##  4 Peru            0.511 36815
##  5 Costa Rica      0.516 36457
##  6 Thailand        0.517 36355
##  7 Morocco         0.518 36275
##  8 Indonesia       0.518 36252
##  9 Malaysia        0.523 35874
## 10 Turkey          0.524 35819
## # ... with 172 more rows
```

The solution

Our goal was to reduce the proportion of our data that was missing values while still keeping a significant number of observations. Recall that each observation was uniquely identified by a `Country`, `Indicator`, `Year` combination. Because of this, we decided to remove a combination of countries and indicators that would remove the most missing values.

We had decided not to remove Years because we wanted to maintain that temporal data and thought there might be interesting insights based on time. We ended up not going down this route. Therefore, if we had more time we would also remove some range of Years if it would help us maintain more indicators or countries as a result.

Also in this step we filtered our countries to only be those included in another external list of country names because there were a few territories included in this list

```
country <- read_csv("data/Country.csv", col_types = cols())
ind <- read_csv("data/Indicators.csv", col_types = cols())

# Each observation is uniquely identified by CountryName/CountryCode, IndicatorName/IndicatorCode, Year

names <- read_tsv("world-country-names.tsv", col_types = cols())
data_names <- names$name
ind_names <- ind$CountryName %>% unique()
overlap_names <- intersect(ind_names, data_names)

ind <- ind %>%
  mutate(IndicatorName = str_replace_all(IndicatorName, " ", "_"))

# remove Channel Islands explicitly because it has a missing year
ind <- ind %>% filter(CountryName != "Channel Islands") %>% filter(CountryName %in% overlap_names)

# pivot wider to add missing values
ind2 <- ind %>%
```

```
  select(CountryName, IndicatorName, Year, Value) %>%
  pivot_wider(names_from = IndicatorName,
              values_from = Value,
              values_fill = NA)

# Number of missing values for each indicator
max_percent_missing <- 0.2

# Filter to only have indicators that have less than max_percent_missing percent of missing values
sapply(ind2, function(x)
  sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
  mutate(p_missing = . / 13823) %>%
  arrange(-desc(.)) %>%
  filter(p_missing < max_percent_missing) %>%
  invisible()

filtered_indicators <- sapply(ind2, function(x)
  sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
  mutate(p_missing = . / 13823) %>%
  arrange(-desc(.)) %>%
  filter(p_missing < 0.5)

filtered_indicators <- filtered_indicators %>% row.names()
filtered_indicators <-
  filtered_indicators[filtered_indicators != "Year"]

# Number of filtered indicators
# filtered_indicators %>% length()
# 651

# Number of missing values per country
total_num_data_ideally <-
  (ind2 %>% colnames() %>% length() - 2) * 56 # CountryName and Year don't count
max_percent_missing <- 0.1

good_countries <- ind2 %>%
  group_by(CountryName) %>%
  select(-Year) %>%
  summarise_all(funs(sum(is.na(.)))) %>%
  mutate(p_missing = select(.,-CountryName) %>% rowSums() / total_num_data_ideally) %>% select(CountryNa
  filter(p_missing < 0.7)

ind3 <- ind2 %>%
  filter(CountryName %in% good_countries$CountryName)

my_row_names <- sapply(ind3, function(x)
  sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
```

```
  row.names()

chosen_indicators_ <- sapply(ind3, function(x)
  sum(is.na (x))) %>%
  as.data.frame() %>%
  select(., everything()) %>%
  mutate(names = my_row_names) %>%
  mutate(p_missing = . / 13823) %>% # removes rownames on oat
  arrange(-desc(.)) %>%
  filter(p_missing < max_percent_missing)
chosen_indicators <- chosen_indicators_$names
```

```
ind4 <- ind3 %>%
  select(CountryName, Year, all_of(chosen_indicators))

# Now the proportion of missing values is much less at `r p_missing_(ind4)`.
# We want to have some fraction of countries we remove and some fraction of indicators we remove such t

ind5 <- ind4 %>% group_by(CountryName) %>%
  select(-Year) %>%
  summarise_all(funs(mean(., na.rm = TRUE)))

# impute values
for (i in 2:ncol(ind5)) {
  ind5[is.na(ind5[, i]), i] <- lapply(ind5[, i], mean, na.rm = TRUE)
}
```

The following table summarizes the results of our data cleaning (excluding imputation). We were able to reduce the number of missing values but still keep a significant amount of observations. We went from `68.68254` percent of the data being missing values to only `10.68711` percent, while still maintaining `5992` (58.7912088 percent) of the observations.

While we deemed this satisfactory and moved on with our analysis, we know it probably wasn't the optimal choice of countries and indicators dropped. We did a bunch of testing and ended up first dropping all indicators with more than 20% missing values and then countries with more than 10% missing values. The optimal solution would probably to have alternated between dropping indicators and countries by dropping the indicator or country that contained the most missing values and alternating until some threshold of missing values or number of observations were met. Also, as mentioned before, we could have considered dropping some years as well. Ideally, given more time, we would have implemented this.

```
# After all that, we have a new dataset that has much less missing values.
tibble(
  label = c("Before", "After"),
  p_missing = c(p_missing_(ind2), p_missing_(ind4)),
  n_obs = c(ind2 %>% nrow(), ind4 %>% nrow())
)
```

```
## # A tibble: 2 x 3
##   label  p_missing n_obs
##   <chr>      <dbl> <int>
## 1 Before    0.687  10192
## 2 After     0.0576  5992
```