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. We accessed the dataset from Kaggle where it was published directly by Kaggle. 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:

What indicators have the greatest impact on a country's economic production?

There are massive inequalities in the economic health of countries worldwide, ranging from superpowers like the United States and China to developing countries like Somalia and Venezuela. We wanted to find out what specific factors are most responsible for these inequalities. Answering this question will not directly fix inequality, but it is important to identify the primary causes so that more attention can be spent addressing relevant issues in poor or developing countries.

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

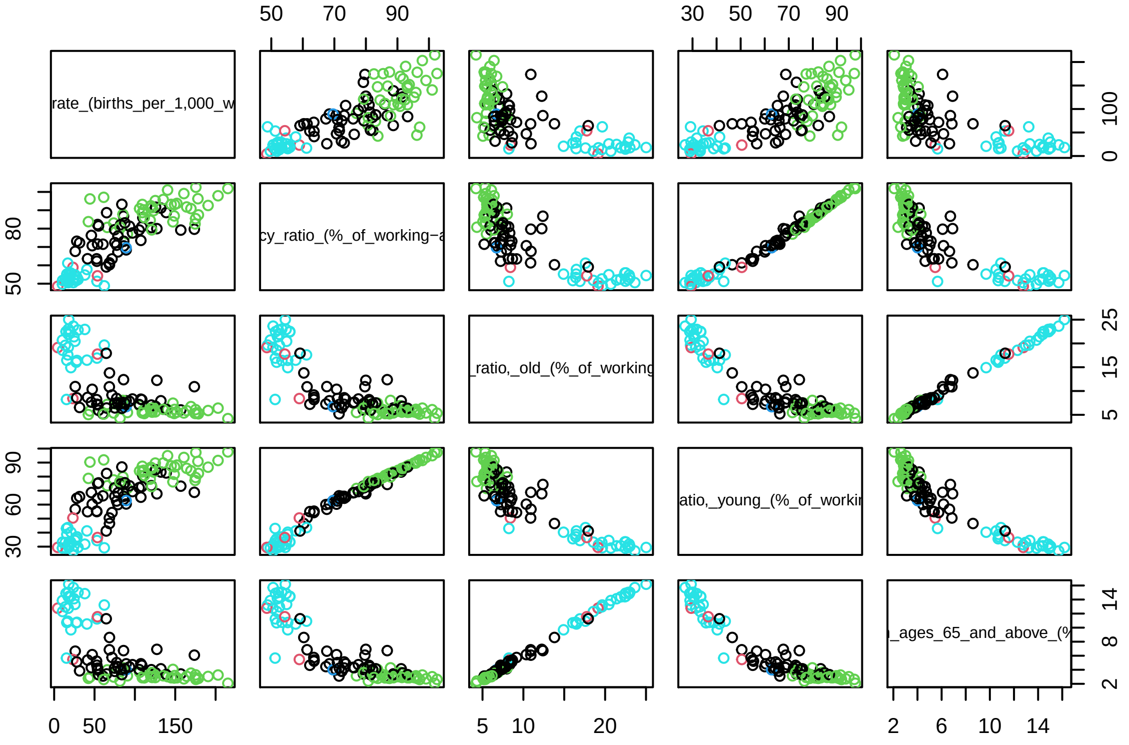
Doing PCA and kmeans

```
# 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)
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

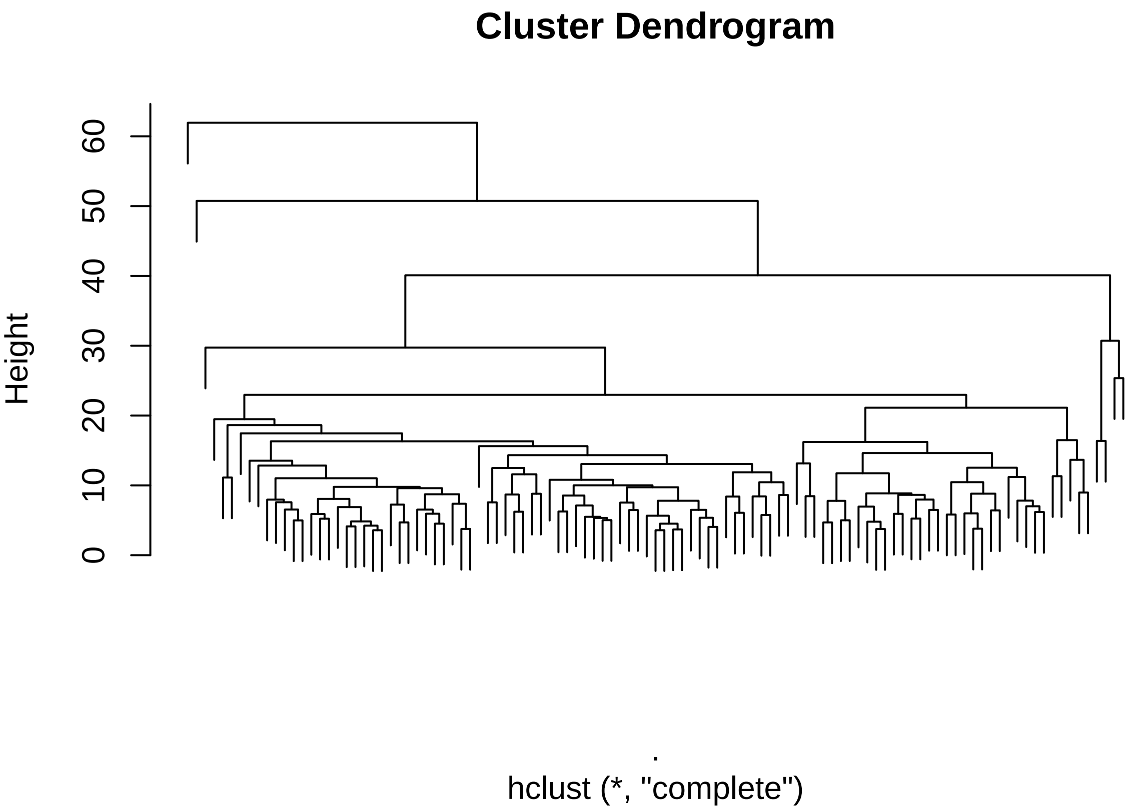
Pairs plot with arbitrarily chosen indicators to show clustering:

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



Hierarchical clustering

```
# heir clustering
foo[, 5:142] %>% 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()
```

```
##
## Call:
## lm(formula = gdp_usd ~ PC1 + PC2 + PC3 + PC4 + PC5, data = goo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8637.1 -2954.7  -292.3  2005.6 19373.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5347.28     405.73  13.179  < 2e-16 ***
## PC1          -895.51      66.42 -13.483  < 2e-16 ***
## PC2           718.10      91.70   7.831 4.97e-12 ***
## PC3          -350.91      95.54  -3.673 0.000385 ***
## PC4          -552.49     141.75  -3.898 0.000175 ***
## PC5          -152.56     160.58  -0.950 0.344351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4197 on 101 degrees of freedom
## Multiple R-squared:  0.7297, Adjusted R-squared:  0.7164
## F-statistic: 54.54 on 5 and 101 DF,  p-value: < 2.2e-16
```

Model/test 2

Paragraph on model/test 2

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

```
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
```

```r
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)

# Calculating proportion of missing values:

p_missing_ <- function(d) {
  mvt <- table(is.na(d))
  mvt[2] / (mvt[1] + mvt[2])
}

# 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)
}
```

```
# After all that, we have a new dataset that has much less missing values.
tibble(
  p_missing_before = p_missing_(ind2),
  p_missing_after = p_missing_(ind4)
)
```

```
## # A tibble: 1 x 2
##   p_missing_before p_missing_after
##              <dbl>           <dbl>
## 1            0.687           0.107
```