Quality of Life by Country: A Clustering Analysis

Katherine M. Prioli

December 22, 2018

Background

Methods

Loading libraries

```
library(tidyverse)
library(readxl)          # For importing .xls(x) datasets
library(lazyeval)        # For renaming columns in function
library(countrycode)     # For establishing uniform country identifiers
library(ggthemr)         # For prettifying output
library(gridExtra)       # For grid.arrange()
library(grid)            # For textGrob() to annotate grid.arrange() elements
library(kableExtra)      # For nicer output tables

ggthemr("fresh")
```

Establishing a crosswalk for country names and 3-letter codes

```
countries_full <- codelist_panel %>%
  select(country.name.en, year, genc3c, iso3c, wb_api3c) %>%
  group_by(country.name.en) %>%
  mutate(maxyr = max(year)) %>%
  ungroup %>%
  mutate(maxyr = case_when(
    maxyr == year ~ 1,
    TRUE ~ 0
  )) %>%
  filter(maxyr == 1) %>%
  select(-maxyr) %>%
  distinct()

countries_full <- countries_full %>%
  mutate(country3 = case_when(
    iso3c == genc3c & iso3c == wb_api3c ~ iso3c,
    is.na(iso3c) == FALSE ~ iso3c,
    is.na(iso3c) == TRUE & is.na(genc3c) == FALSE ~ genc3c,
    is.na(iso3c) == TRUE & is.na(genc3c) == TRUE & is.na(wb_api3c) == FALSE ~ wb_api3c
  )) %>%
  rename(country = country.name.en) %>%
  arrange(country)

countries <- countries_full %>%
  select(country, country3)
```

Importing and wrangling each data file, and standardizing country names

Each datafile was imported and wrangled to subset to the variable(s) of interest for 2016. Next, country identifiers in each dataset were compared to the `countries` table, and a `mutate()` statement was used to correct mismatches. In the interest of brevity, these steps are demonstrated for the Human Development Index (HDI) data below.

First, importing and wrangling the HDI data:

```
# Importing raw data

HDIraw <- read_xlsx("data/HDIdata2018.xlsx", sheet = "Table 2")
HDIraw
```

```
## # A tibble: 240 x 27
##    X__1  `Table 2. Human~ X__2  X__3  X__4  X__5  X__6  X__7  X__8  X__9
##    <chr> <chr>            <chr> <lgl> <chr> <lgl> <chr> <lgl> <chr> <lgl>
##  1 <NA>  <NA>             <NA>  NA    <NA>  NA    <NA>  NA    <NA>  NA
##  2 <NA>  <NA>             Huma~ NA    <NA>  NA    <NA>  NA    <NA>  NA
##  3 HDI ~ Country          Value NA    <NA>  NA    <NA>  NA    <NA>  NA
##  4 <NA>  <NA>             1990  NA    2000  NA    2010  NA    2012  NA
##  5 <NA>  VERY HIGH HUMAN~ <NA>  NA    <NA>  NA    <NA>  NA    <NA>  NA
##  6 1     Norway           0.85~ NA    0.91~ NA    0.94~ NA    0.94~ NA
##  7 2     Switzerland      0.83~ NA    0.88~ NA    0.93~ NA    0.93~ NA
##  8 3     Australia        0.86~ NA    0.89~ NA    0.92~ NA    0.92~ NA
##  9 4     Ireland          0.76~ NA    0.85~ NA    0.90~ NA    0.90~ NA
## 10 5     Germany          0.80~ NA    0.86~ NA    0.92~ NA    0.92~ NA
## # ... with 230 more rows, and 17 more variables: X__10 <chr>, X__11 <lgl>,
## #   X__12 <chr>, X__13 <lgl>, X__14 <chr>, X__15 <lgl>, X__16 <chr>,
## #   X__17 <lgl>, X__18 <chr>, X__19 <chr>, X__20 <chr>, X__21 <lgl>,
## #   X__22 <chr>, X__23 <lgl>, X__24 <chr>, X__25 <lgl>, X__26 <chr>
```

```
# Selecting columns of interest

HDIdata <- HDIraw %>%
  select(1:2, X__14)

# Assigning sensible column names

HDIcolnm <- c(HDIdata[[3,1]], HDIdata[[3,2]], HDIdata[[4,3]])
colnames(HDIdata) <- HDIcolnm

# Determining boundaries for human development levels in the data
# and using these to create one dataframe for each level

vhhd_st <- which(HDIdata$Country == "VERY HIGH HUMAN DEVELOPMENT") + 1
vhhd_end <- which(HDIdata$Country == "HIGH HUMAN DEVELOPMENT") - 1

hhd_st <- which(HDIdata$Country == "HIGH HUMAN DEVELOPMENT") + 1
hhd_end <- which(HDIdata$Country == "MEDIUM HUMAN DEVELOPMENT") - 1

mhd_st <- which(HDIdata$Country == "MEDIUM HUMAN DEVELOPMENT") + 1
mhd_end <- which(HDIdata$Country == "LOW HUMAN DEVELOPMENT") - 1

lhd_st <- which(HDIdata$Country == "LOW HUMAN DEVELOPMENT") + 1
lhd_end <- which(HDIdata$Country == "OTHER COUNTRIES OR TERRITORIES") - 1

oth_st <- which(HDIdata$Country == "OTHER COUNTRIES OR TERRITORIES") + 1
oth_end <- which(HDIdata$Country == "Human development groups") - 2

HDI_vhhd <- HDIdata %>%
  slice(vhhd_st:vhhd_end) %>%
  mutate(HDI_cat = "Very High")

HDI_hhd <- HDIdata %>%
  slice(hhd_st:hhd_end) %>%
  mutate(HDI_cat = "High")
```

```r
HDI_mhd <- HDIdata %>%
  slice(mhd_st:mhd_end) %>%
  mutate(HDI_cat = "Medium")

HDI_lhd <- HDIdata %>%
  slice(lhd_st:lhd_end) %>%
  mutate(HDI_cat = "Low")

HDI_oth <- HDIdata %>%
  slice(oth_st:oth_end) %>%
  mutate(HDI_cat = NA)

# Combining the dataframes into one

HDIdata <- bind_rows(HDI_vhhd, HDI_hhd, HDI_mhd, HDI_lhd, HDI_oth) %>%
  rename(HDIrank = `HDI rank`) %>%
  rename(country = Country) %>%
  rename(HDIindex = `2016`) %>%
  mutate(HDI_cat = factor(HDI_cat, levels = c("Low", "Medium", "High", "Very High"))) %>%
  mutate(HDIrank = case_when(
    HDIrank == ".." ~ as.numeric(NA),
    TRUE ~ as.numeric(HDIrank)
  )) %>%
  mutate(HDIindex = case_when(
    HDIindex == ".." ~ as.numeric(NA),
    TRUE ~ as.numeric(HDIindex)
  ))
HDIdata <- HDIdata[c(2, 1, 3:4)]
```

Next, standardizing country names by using `anti_join()` to see which countries in `HDIdata` don't have a match in the `countries` dataframe, and correcting those for which an inexact match exists:

```r
HDIanti <- HDIdata %>%
  anti_join(countries, by = "country") %>%
  select(country) %>%
  arrange(country)
dim(HDIanti)
```

```
## [1] 28  1
```

There are 28 countries in `HDIdata` without an exact match in `countries`. Correcting using `mutate()`:

```r
HDIdata <- HDIdata %>%
  mutate(country = case_when(
    country == "Antigua and Barbuda"                       ~ "Antigua & Barbuda",
    country == "Bolivia (Plurinational State of)"          ~ "Bolivia",
    country == "Bosnia and Herzegovina"                    ~ "Bosnia & Herzegovina",
    country == "Brunei Darussalam"                         ~ "Brunei",
    country == "Cabo Verde"                                ~ "Cape Verde",
    country == "Congo"                                     ~ "Congo - Brazzaville",
    country == "Congo (Democratic Republic of the)"        ~ "Congo - Kinshasa",
    country == "Eswatini (Kingdom of)"                     ~ "Swaziland",
    country == "Hong Kong, China (SAR)"                    ~ "Hong Kong SAR China",
    country == "Iran (Islamic Republic of)"                ~ "Iran",
    country == "Korea (Democratic People's Rep. of)"       ~ "North Korea",
    country == "Korea (Republic of)"                       ~ "South Korea",
    country == "Lao People's Democratic Republic"          ~ "Laos",
    country == "Moldova (Republic of)"                     ~ "Moldova",
    country == "Myanmar"                                   ~ "Myanmar (Burma)",
    country == "Palestine, State of"                       ~ "Palestinian Territories",
    country == "Russian Federation"                        ~ "Russia",
```

```
    country == "Saint Kitts and Nevis"                       ~ "St. Kitts & Nevis",
    country == "Saint Lucia"                                 ~ "St. Lucia",
    country == "Saint Vincent and the Grenadines"            ~ "St. Vincent & Grenadines",
    country == "Syrian Arab Republic"                        ~ "Syria",
    country == "Tanzania (United Republic of)"               ~ "Tanzania",
    country == "The former Yugoslav Republic of Macedonia" ~ "Macedonia",
    country == "Trinidad and Tobago"                         ~ "Trinidad & Tobago",
    country == "Venezuela (Bolivarian Republic of)"          ~ "Venezuela",
    country == "Viet Nam"                                    ~ "Vietnam",
    country == "Côte d'Ivoire"                               ~ as.character(NA),   # UTC-8
    country == "Sao Tome and Principe"                       ~ as.character(NA),   # conflicts
    TRUE                                                     ~ as.character(country)
  )) %>%
  filter(!is.na(country))

HDIanti <- HDIdata %>%
  anti_join(countries, by = "country") %>%
  select(country) %>%
  arrange(country)
dim(HDIanti)
```

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

Now there are no countries in `HDIdata` without an exact match in `countries`.

This process of importing, wrangling, and testing against the `countries` dataframe was largely the same for all other datasets of interest, with minor differences depending on the native structure of the data.

Combining individual data files into one dataframe

All datasets were merged into a single dataframe using serial `join()` statements, and the resulting dataset was filtered to omit countries without data.

```
joindata_1 <- full_join(countries, HDIdata, by = "country")
joindata_2 <- left_join(joindata_1, SPIdata, by = "country3")
joindata_3 <- left_join(joindata_2, WHRdata, by = "country")
joindata_4 <- left_join(joindata_3, genderdata, by = "country")
joindata_5 <- left_join(joindata_4, infantmortdata, by = "country")
joindata_6 <- left_join(joindata_5, lifeexpdata, by = "country")
joindata_7 <- left_join(joindata_6, GDPdata, by = "country3")

joinsub <- joindata_7 %>%
  arrange(country) %>%
  mutate(exclude_flag = case_when(
    is.na(HDIrank) == TRUE &
      is.na(HDIindex) == TRUE &
      is.na(HDI_cat) == TRUE &
      is.na(SPI) == TRUE &
      is.na(happiness) == TRUE &
      is.na(genderequality_index) == TRUE &
      is.na(infantmort) == TRUE &
      is.na(birth_MF) == TRUE &
      is.na(sixty_MF) == TRUE &
      is.na(GDP_USD_2018) == TRUE             ~ TRUE,
    TRUE                                      ~ FALSE
  )) %>%
  filter(exclude_flag == FALSE) %>%
  select(-exclude_flag)

alldata <- joinsub
len <- dim(alldata)[[1]]
```

```r
# write_csv(alldata, paste0("data/alldata_", lubridate::today(),".csv"))  # Uncomment to export data
```

The final dataframe, titled `alldata`, contains the following:

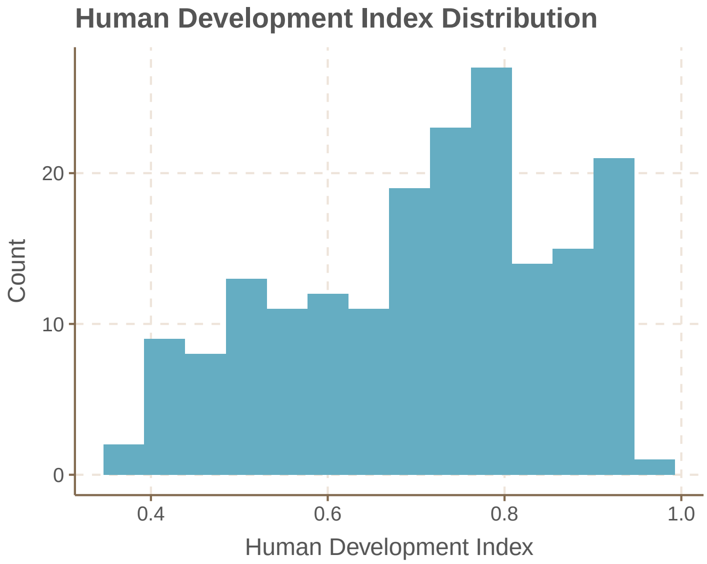
Source	Variable Name	Description
The United Nations Development Programme (2018)	HDIrank	Human Development Index ranking
The United Nations Development Programme (2018)	HDIindex	HDI index value (scale of 0:1)
The United Nations Development Programme (2018)	HDI_cat	HDI index category (5 levels)
Social Progress Imperative (2018)	SPI	Social Progress Index value (scale of 0:100)
World Happiness Report (2018)	happiness	World Happiness Score (scale of 0:10)
WEF (2016)	genderequality_index	Gender Equality Index (scale of 0:1)
WHO (2018b)	infantmort	Infant mortality rate
WHO (2018a)	birth_MF	Life expectancy at birth, males & females
WHO (2018a)	sixty_MF	Life expectancy at 60 years, males & females
The World Bank (2018)	GDP_USD_2018	2016 Gross Domestic Product (valued in $US 2018)

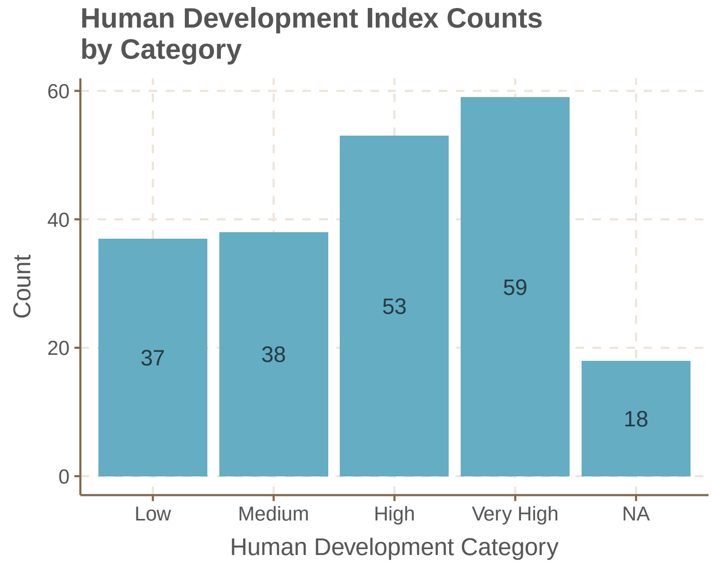
Visualizations

Univariate and sensible bivariate analyses were generated to explore the data.

Exploring the Human Development Index variables:

```r
HDIindex_hist <- ggplot(data = alldata, aes(x = HDIindex)) +
  geom_histogram(bins = ceiling(sqrt(203 - sum(is.na(alldata$HDIindex))))) +
  # geom_density(aes(y = (max(alldata$HDIindex, na.rm = TRUE) - min(alldata$HDIindex, na.rm = TRUE)) /
  #                      ceiling(sqrt(203 - sum(is.na(alldata$HDIindex)))) * ..count..), color = "#233b43") +
  xlab("Human Development Index") +
  ylab("Count") +
  ggtitle("Human Development Index Distribution")
#HDIindex_hist

HDIcat_bar <- ggplot(data = alldata, aes(x = HDI_cat)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), color = "#233b43",
            hjust = 0.5, position = position_stack(vjust = 0.5)) +
  xlab("Human Development Category") +
  ylab("Count") +
  ggtitle("Human Development Index Counts \nby Category")
#HDIcat_bar

grid.arrange(HDIindex_hist, HDIcat_bar, nrow = 1)
```





The top 5 countries by HDI rank are:

```
HDIsub <- alldata %>%
  select(country, HDIrank, HDIindex) %>%
  arrange(HDIrank) %>%
  filter(!is.na(HDIindex) == TRUE) %>%
  mutate(HDIindex = round(HDIindex, digits = 4))
HDItop <- head(HDIsub, 5) %>% kable(format = "markdown")
HDItop
```

country	HDIrank	HDIindex
Norway	1	0.9512
Switzerland	2	0.9432
Australia	3	0.9379
Ireland	4	0.9345
Germany	5	0.9342

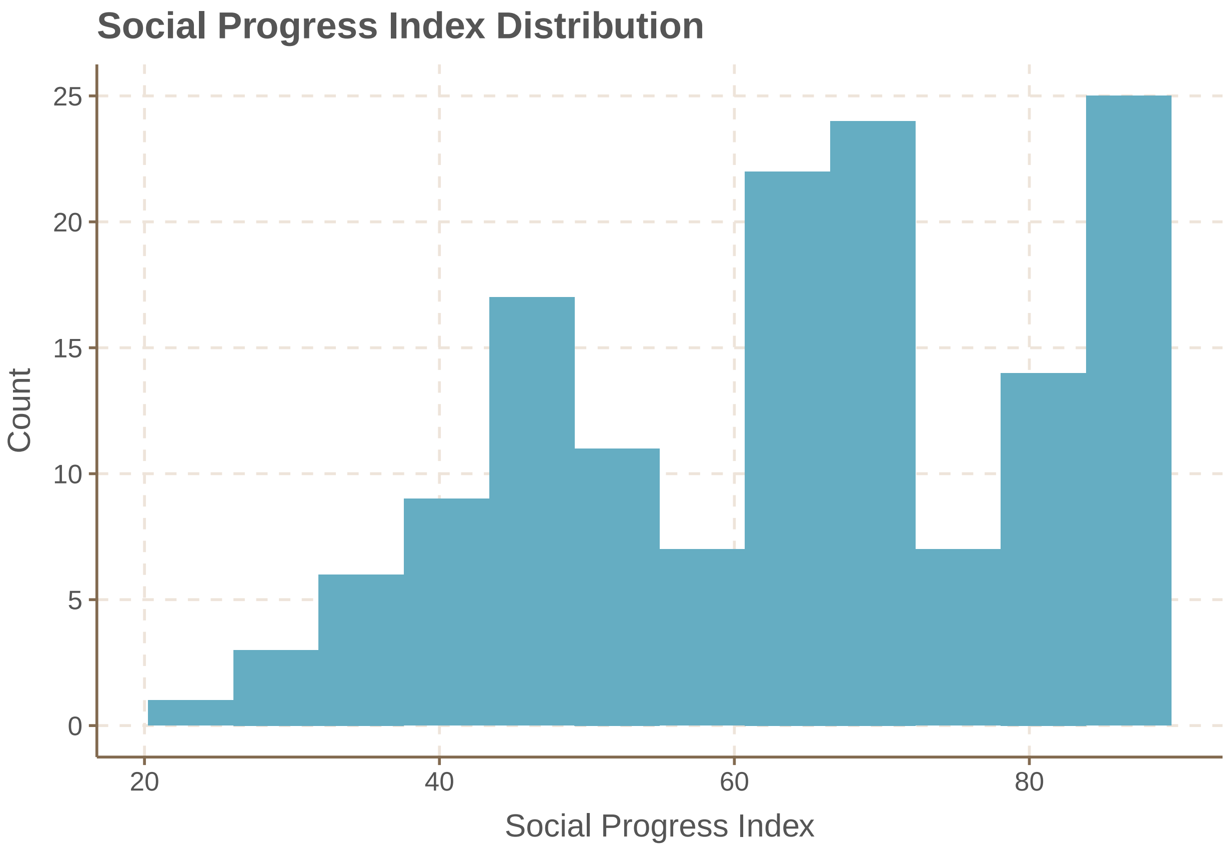
and the bottom 5 are:

```
HDIbot <- tail(HDIsub, 5) %>% kable(format = "markdown")
HDIbot
```

country	HDIrank	HDIindex
Burundi	185	0.4177
Chad	186	0.4053
South Sudan	187	0.3942
Central African Republic	188	0.3622
Niger	189	0.3509

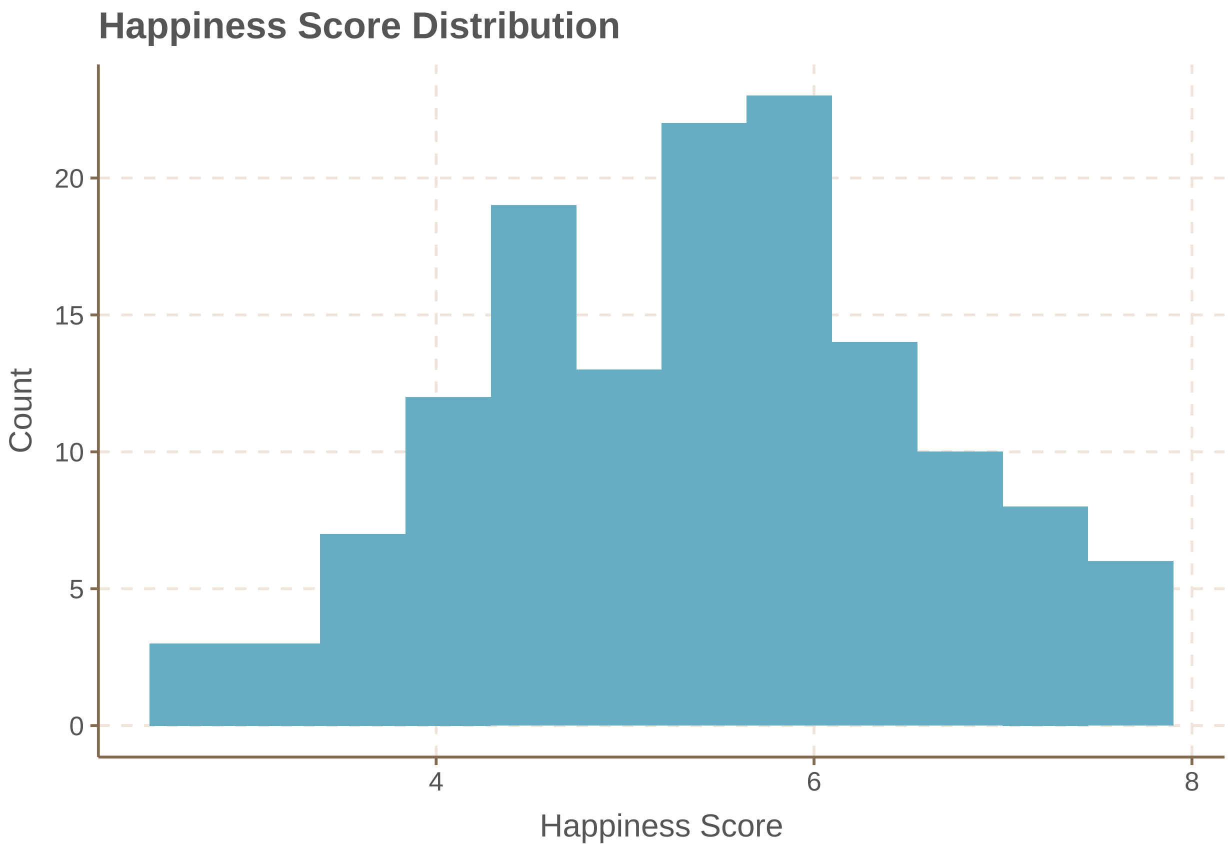
Exploring the Social Progress Index data:

```
SPI_hist <- ggplot(data = alldata, aes(x = SPI)) +
  geom_histogram(bins = ceiling(sqrt(203 - sum(is.na(alldata$SPI))))) +
  # geom_density(aes(y = (max(alldata$SPI, na.rm = TRUE) - min(alldata$SPI, na.rm = TRUE)) /
  #                    ceiling(sqrt(203 - sum(is.na(alldata$SPI)))) * ..count..), color = "#233b43") +
  xlab("Social Progress Index") +
  ylab("Count") +
  ggtitle("Social Progress Index Distribution")
SPI_hist
```



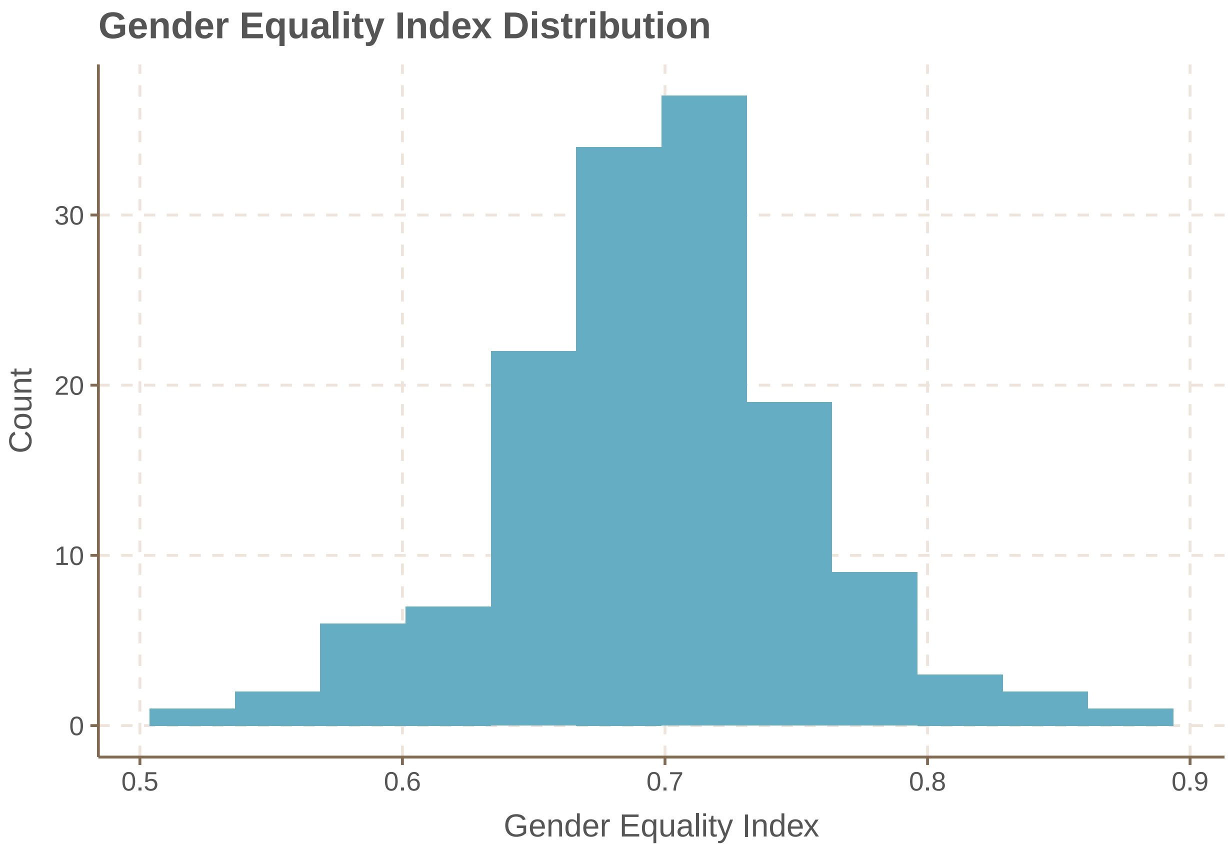
Exploring the World Happiness Report data:

```
happiness_hist <- ggplot(data = alldata, aes(happiness)) +
  geom_histogram(bins = ceiling(sqrt(203 - sum(is.na(alldata$happiness))))) +
  # geom_density(aes(y = (max(alldata$happiness, na.rm = TRUE) - min(alldata$happiness, na.rm = TRUE)) /
  #                 ceiling(sqrt(203 - sum(is.na(alldata$happiness)))) * ..count..), color = "#233b43") +
  xlab("Happiness Score") +
  ylab("Count") +
  ggtitle("Happiness Score Distribution")
happiness_hist
```



Exploring the gender equality index data:

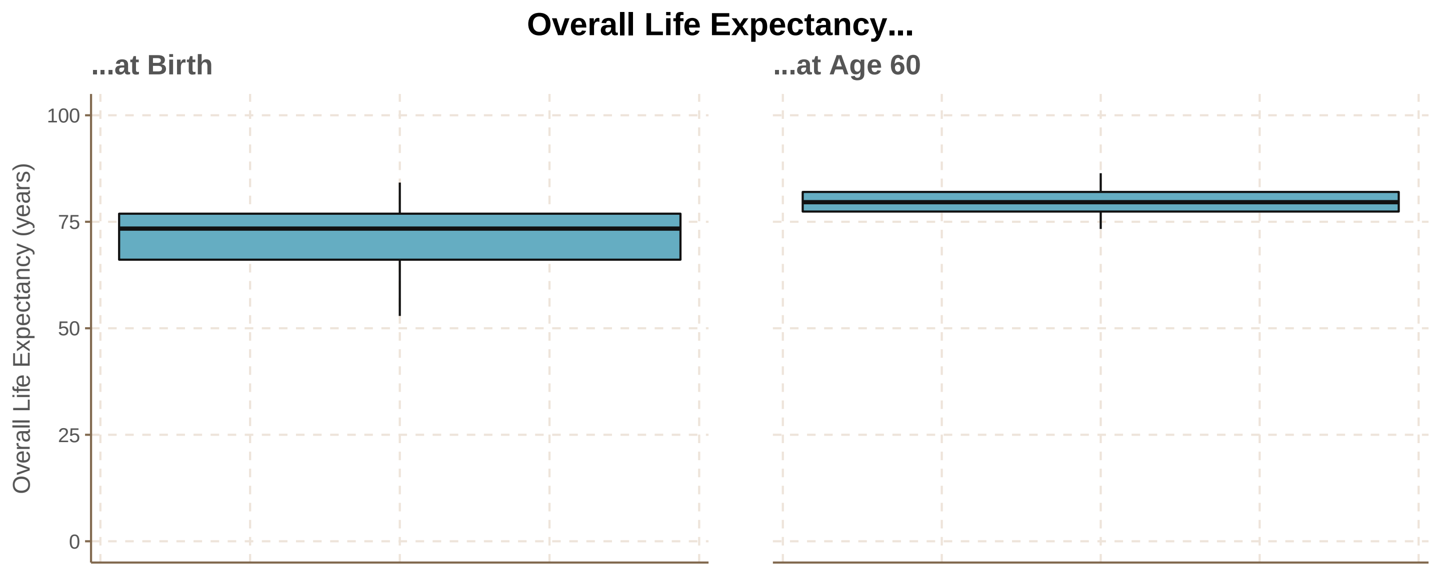
```
gender_hist <- ggplot(data = alldata, aes(x = genderequality_index)) +
  geom_histogram(bins = ceiling(sqrt(203 - sum(is.na(alldata$genderequality_index))))) +
  xlab("Gender Equality Index") +
  ylab("Count") +
  ggtitle("Gender Equality Index Distribution")
gender_hist
```



Exploring the WHO infant mortality rate data:

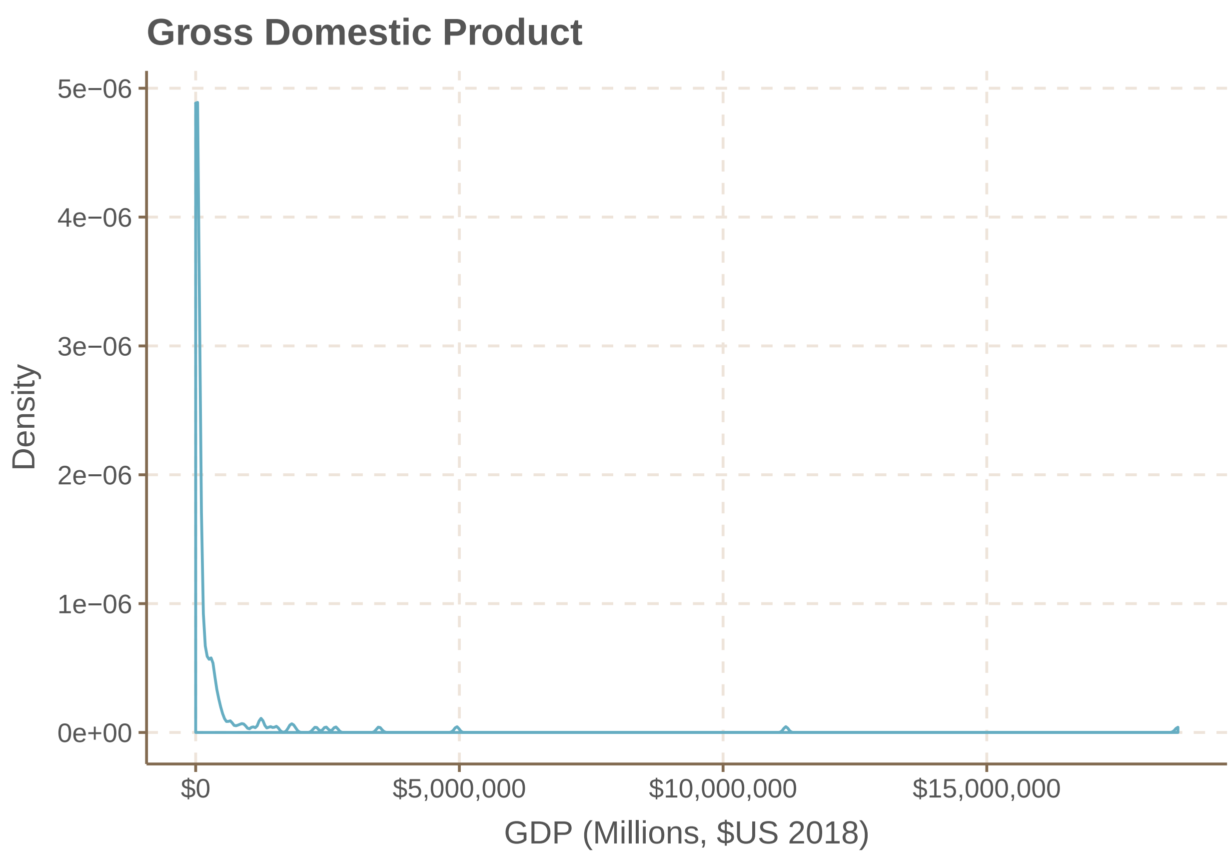
Exploring the WHO life expectancy data:

```
lifeexp_birth_box <- ggplot(data = alldata, aes(y = birth_MF)) +
  geom_boxplot() +
  ylim(0, 100) +
  ylab("Overall Life Expectancy (years)") +
  ggtitle("...at Birth") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

lifeexp_sixty_box <- ggplot(data = alldata, aes(y = 60 + sixty_MF)) +
  geom_boxplot() +
  ylim(0, 100) +
  ylab("") +
  ggtitle("...at Age 60") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.line.y.left = element_blank())

grid.arrange(lifeexp_birth_box, lifeexp_sixty_box, nrow = 1,
             top = textGrob("Overall Life Expectancy...",
                            gp = gpar(fontsize = 16, fontface = "bold")))
```



Exploring the GDP data:

```
GDP_dens <- ggplot(data = alldata, aes(x = GDP_USD_2018 / 1000000)) +
  geom_density() +
  xlab("GDP (Millions, $US 2018)") +
  ylab("Density") +
  ggtitle("Gross Domestic Product") +
  scale_x_continuous(labels = scales::dollar_format(prefix = "$"))
GDP_dens
```



The summary statistics (in millions) are:

```
GDPsumm <- broom::tidy(round(summary(alldata$GDP_USD_2018 / 1000000), digits = 4)) %>%
  kable(format = "markdown")
GDPsumm
```

minimum	q1	median	mean	q3	maximum	na
36.5726	6734.07	27424.07	383069.6	190463	18624500	10

Results

Discussion

Limitations

Conclusion

References

Social Progress Imperative. 2018. “Social Progress Index.” https://www.socialprogress.org/?tab=4.

The United Nations Development Programme. 2018. “Human Development Index.” http://hdr.undp.org/en/data.

The World Bank. 2018. “Gross Domestic Product.” https://data.worldbank.org/indicator/ny.gdp.mktp.cd?view=map&year_high_desc=true.

WEF. 2016. “Gender Equality.” http://reports.weforum.org/global-gender-gap-report-2016/rankings/.

WHO. 2018a. “Life Expectancy.” http://apps.who.int/gho/data/view.main.SDG2016LEXv?lang=en.

———. 2018b. “Probability of Dying Per 1000 Live Births.” http://apps.who.int/gho/data/view.main.182?lang=en.

World Happiness Report. 2018. “World Happiness Report.” http://worldhappiness.report/ed/2018/.