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
library(GGally)          # For ggpairs() correlation matrix

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")

# 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")

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:

```
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()`:

```
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. Again, for brevity, those steps are not shown here, but are available on the project GitHub site (Prioli 2018).

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 %>%
  mutate(country = factor(country)) %>%
  mutate(country3 = factor(country3)) %>%
  mutate(US = case_when(
    country == "United States" ~ "US",
    TRUE                       ~ "Non US"
  )) %>%
  mutate(color = case_when(
    country == "United States" ~ "red",
    TRUE                       ~ "#545454"
  ))

alldata <- alldata[c(1:2, 13:14, 6, 12, 3:5, 7:11)]

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

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

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

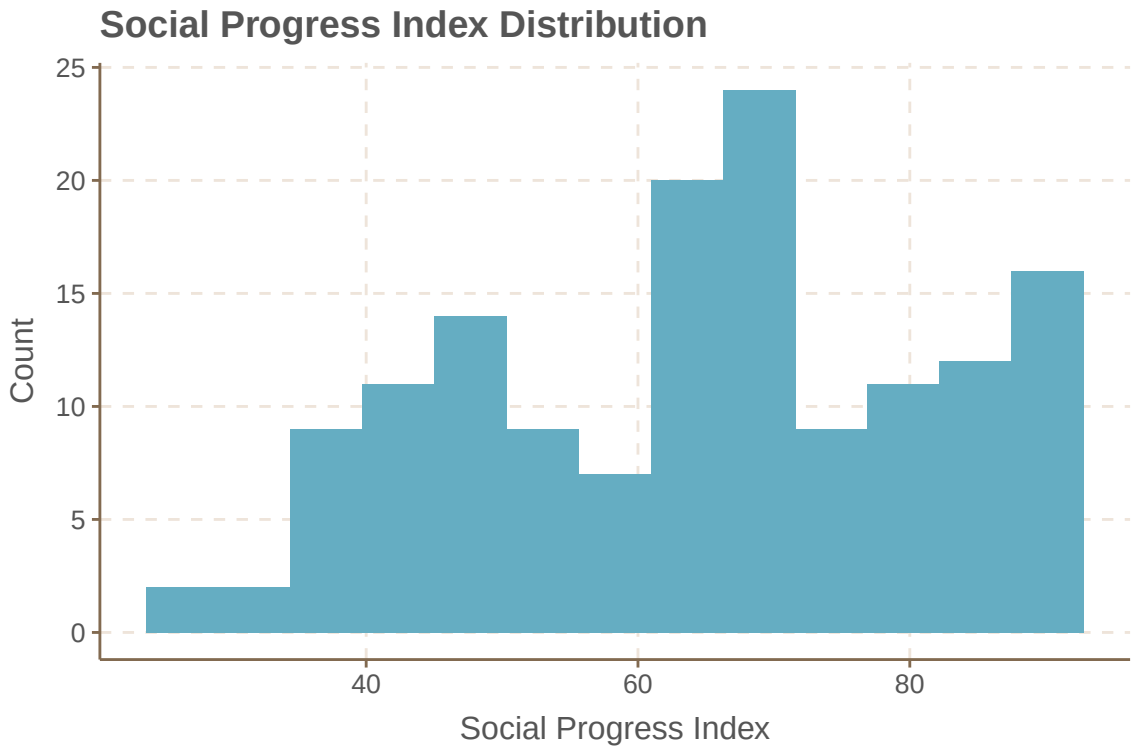
Source	Variable Name	Description
Social Progress Imperative (2018)	SPI	Social Progress Index value (scale of 0:100)
The World Bank (2018)	GDP_USD_2018	2016 Gross Domestic Product (valued in $US 2018)
The United Nations Development Programme (2018)	HDIrank	Human Development Index ranking
The United Nations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

Visualizations

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

Exploring the Social Progress Index data:

```
SPI_hist <- ggplot(data = alldata, aes(x = SPI)) +
  geom_histogram(bins = ceiling(sqrt(len - sum(is.na(alldata$SPI))))) +
  xlab("Social Progress Index") +
  ylab("Count") +
  ggtitle("Social Progress Index Distribution")
SPI_hist
```



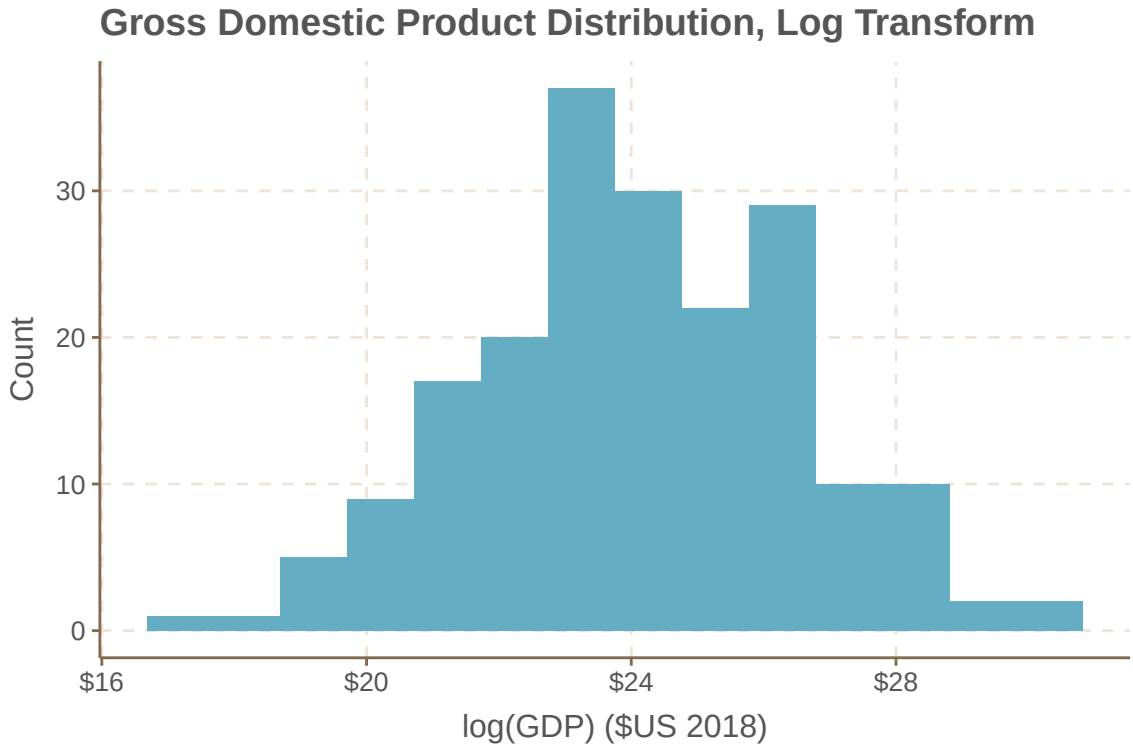
Next, exploring GDP by summary statistics:

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

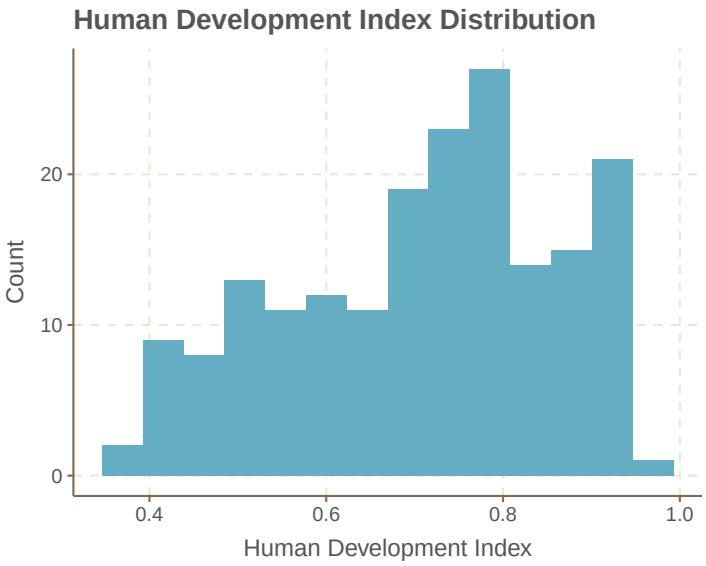
```
GDPsumm
```

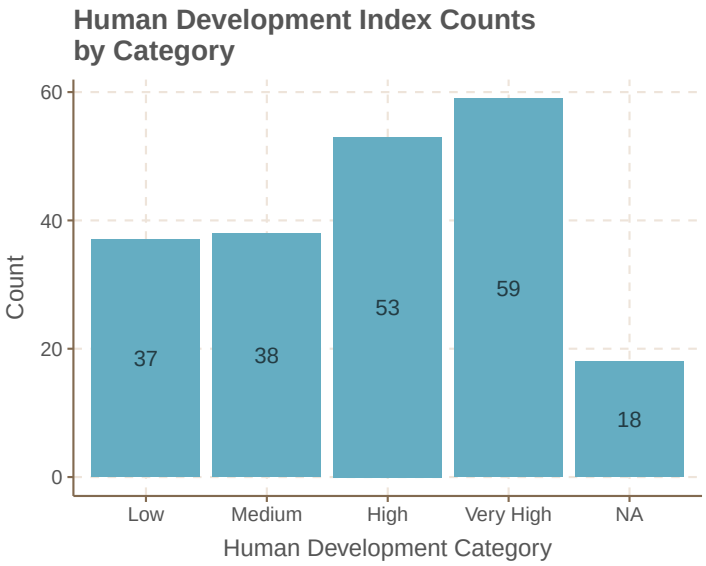
minimum	q1	median	mean	q3	maximum	na
36.5726	6734.07	27424.07	383069.6	190463	18624500	10

Taking the log transform and plotting:



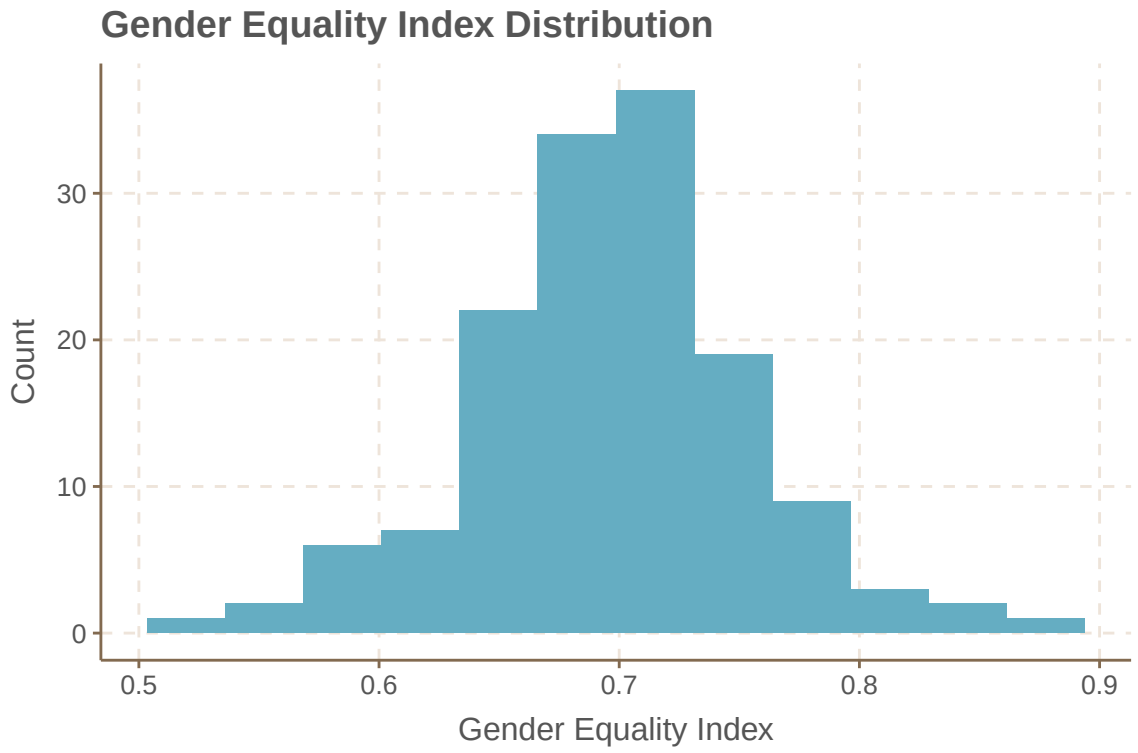
Exploring the Human Development Index variables:



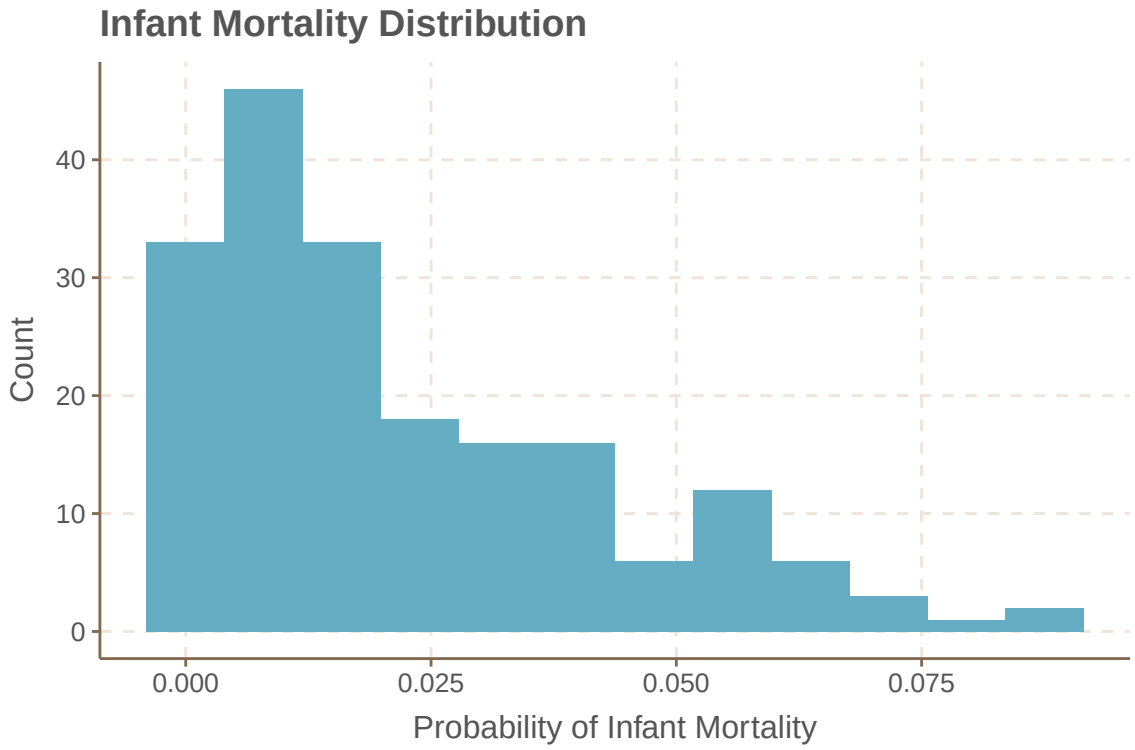


Exploring the World Happiness Report data:

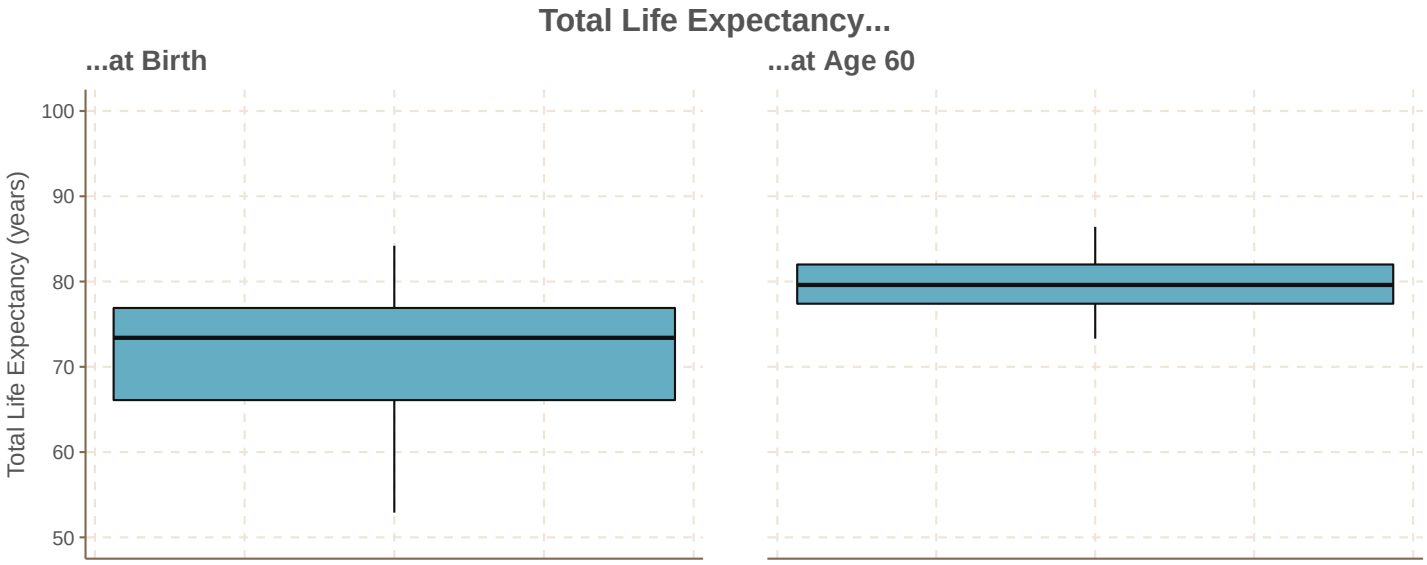
Exploring the gender equality index data:



Exploring the WHO infant mortality rate data:

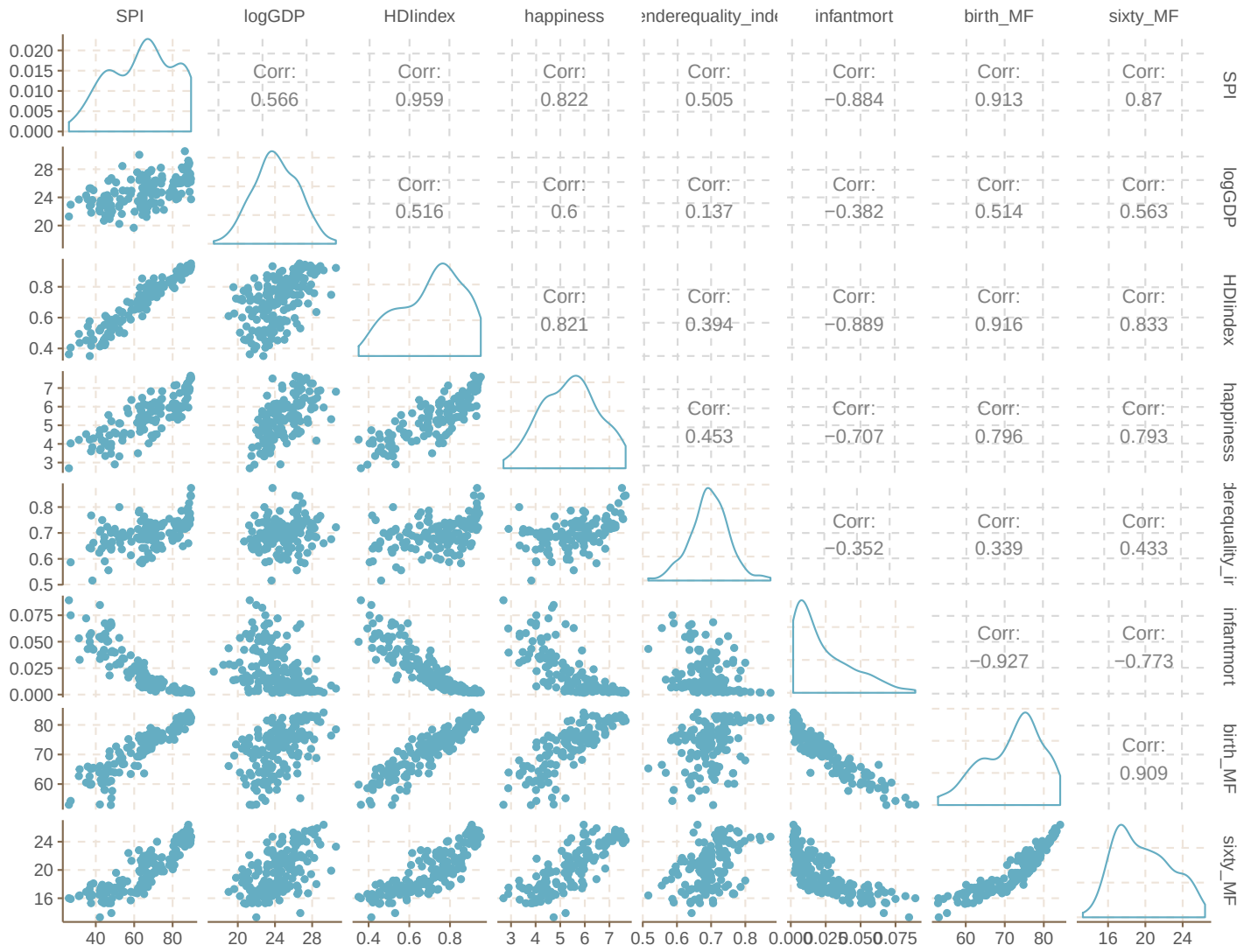


Exploring the WHO life expectancy data:



Investigating pairwise relationships between continuous variables:

```
alldata <- alldata %>% mutate(logGDP = log(GDP_USD_2018))

corrplot <- ggpairs(data = alldata, columns = c(5, 15, 8, 10:14),
                    title = "Correlation Matrix, Continuous Variables")
corrplot
```



Strong positive linear relationships are seen between `HDIindex` and `SPI`, `happiness`, and `birth_MF`; between `SPI` and `happiness`, `birth_MF`, and `sixty_MF`; and between `happiness` and `sixty_MF`. Additionally, strong positive relationships that are possibly nonlinear are seen between `HDI_index` and `sixty_MF`, and between `birth_MF` and `sixty_MF`.

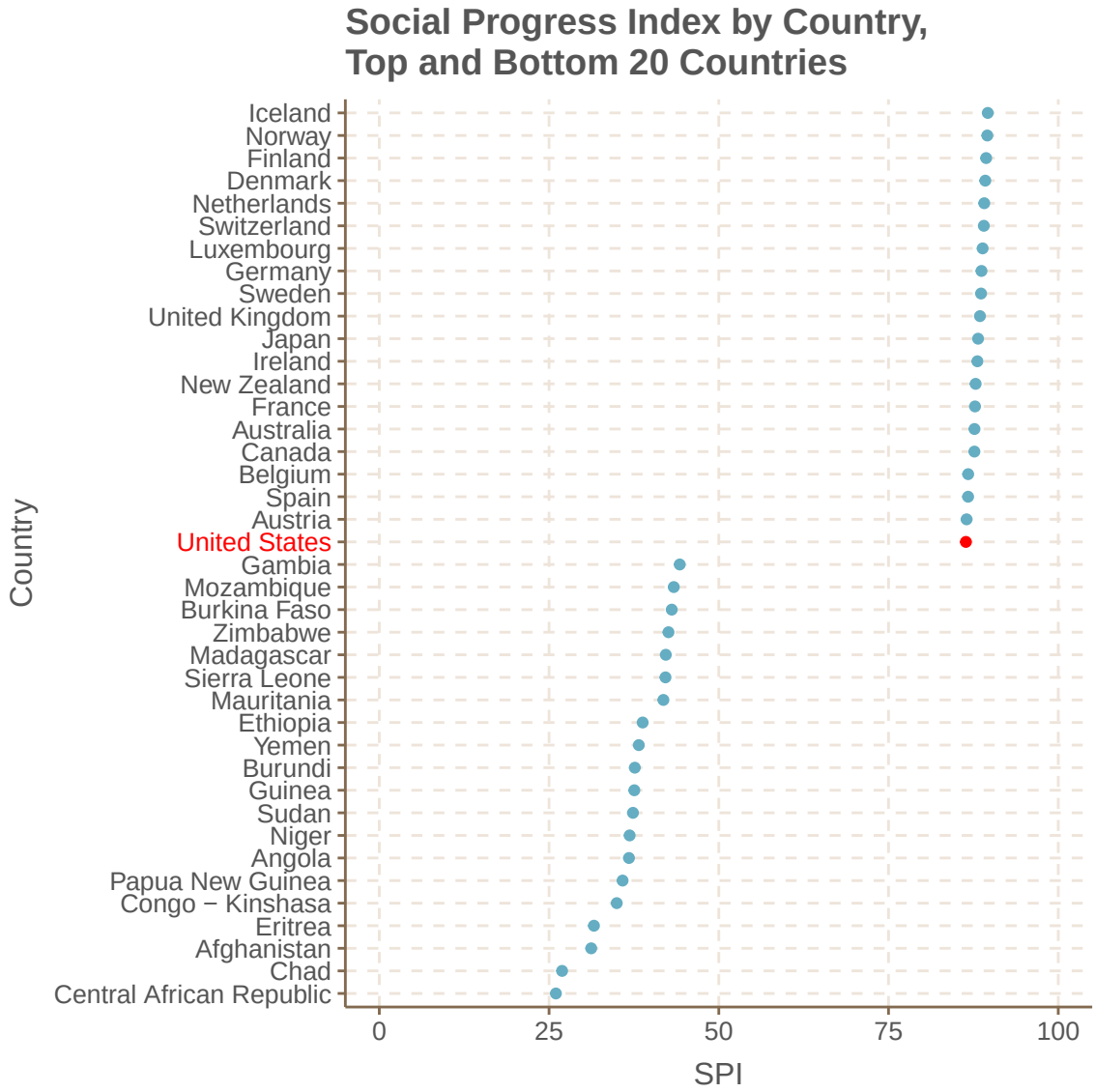
Strong negative relationships are seen between `infantmort` and `birth_MF`, between `HDIindex` and `infantmort`, and between `SPI` and `infantmort`, though the latter two of these may not necessarily be linear. A strong negative nonlinear relationship is seen between `infantmort` and `sixty_MF`.

Since the goal of this analysis is to compare countries with particular focus on the United States, factor-ordered bivariate plots were generated to explore how the countries compare across the variables of interest, with the United States denoted in red.

First, the top and bottom 20 countries were compared by Social Progress Index:

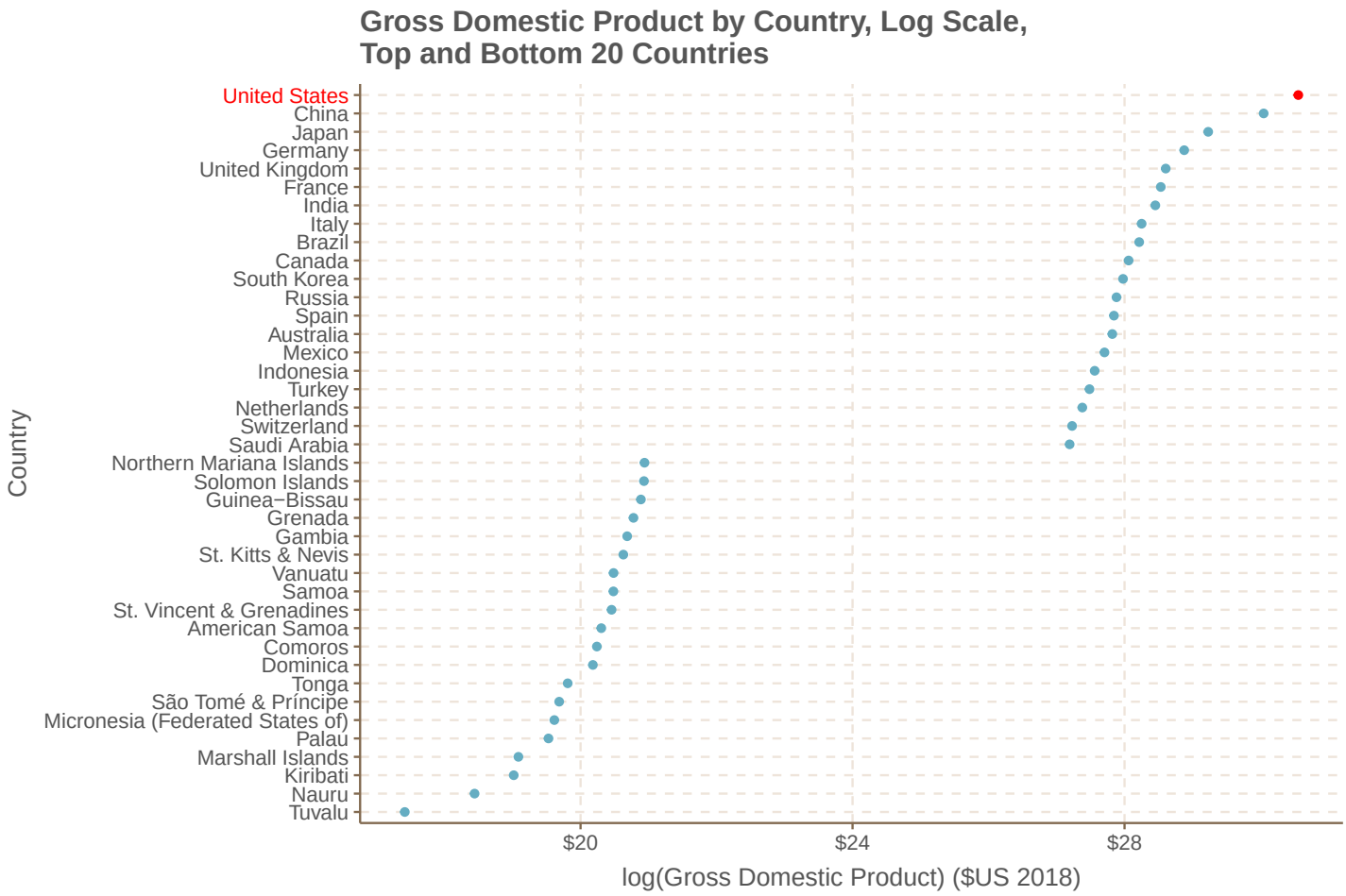
```
alldata_SPI <- alldata %>%
  filter(!is.na(SPI) == TRUE) %>%
  arrange(desc(SPI)) %>%
  select(SPI, country, US, color)

alldata_SPI_top20 <- alldata_SPI %>% head(20)
alldata_SPI_bot20 <- alldata_SPI %>% tail(20)
alldata_SPI40 <- bind_rows(alldata_SPI_top20, alldata_SPI_bot20)

colors <- alldata_SPI40$color[order(alldata_SPI40$SPI)]

SPI_country_point <- ggplot(data = alldata_SPI40, aes(x = SPI,
```

```
                                                y = fct_reorder(country, SPI), color = US)) +
  geom_point() +
  scale_color_manual(values = c("US" = "red", "Non US" = "#65ADC2")) +
  theme(axis.text.y = element_text(color = colors)) +
  guides(color = FALSE) +
  xlim(0, 100) +
  xlab("SPI") +
  ylab("Country") +
  ggtitle("Social Progress Index by Country, \nTop and Bottom 20 Countries")
SPI_country_point
```



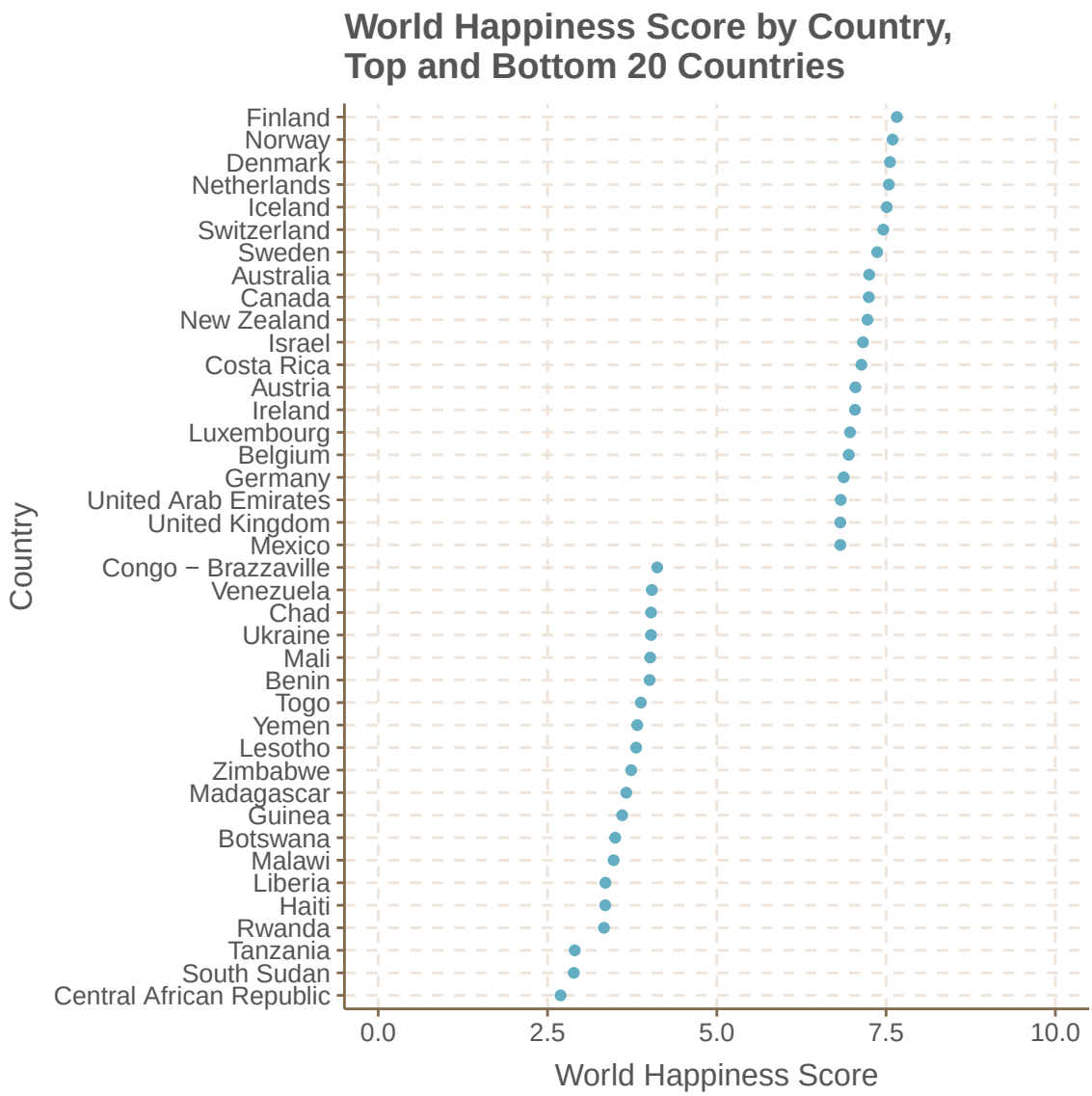
The United States ranks twentieth in social progress.

Next, exploring GDP by country (code for this and subsequent country-level plots not shown for brevity):



The United States has the world's largest GDP.

Next, World Happiness Score:

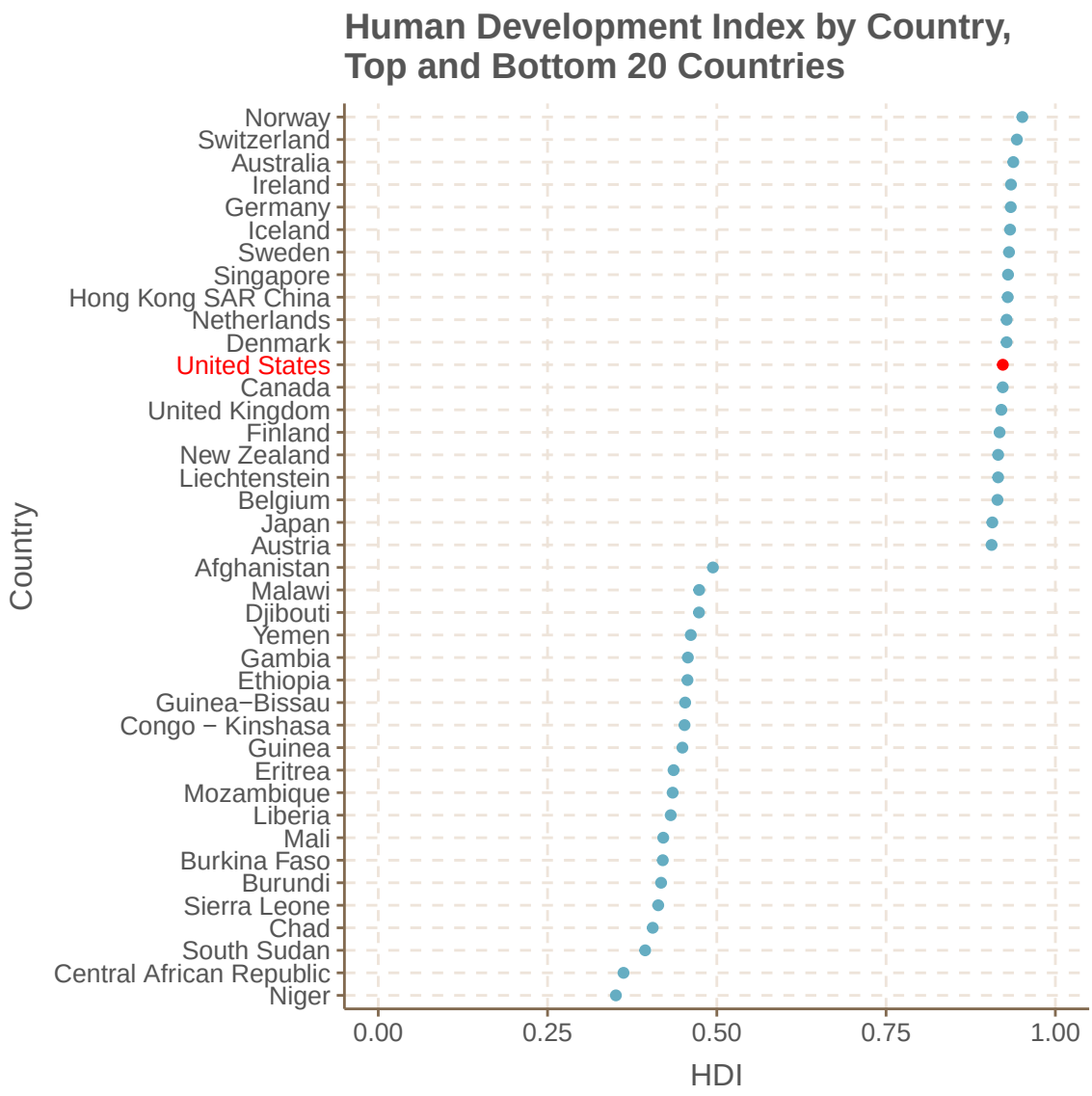



```
which(alldata_WHR$country == "United States")
```

```
## [1] 21
```

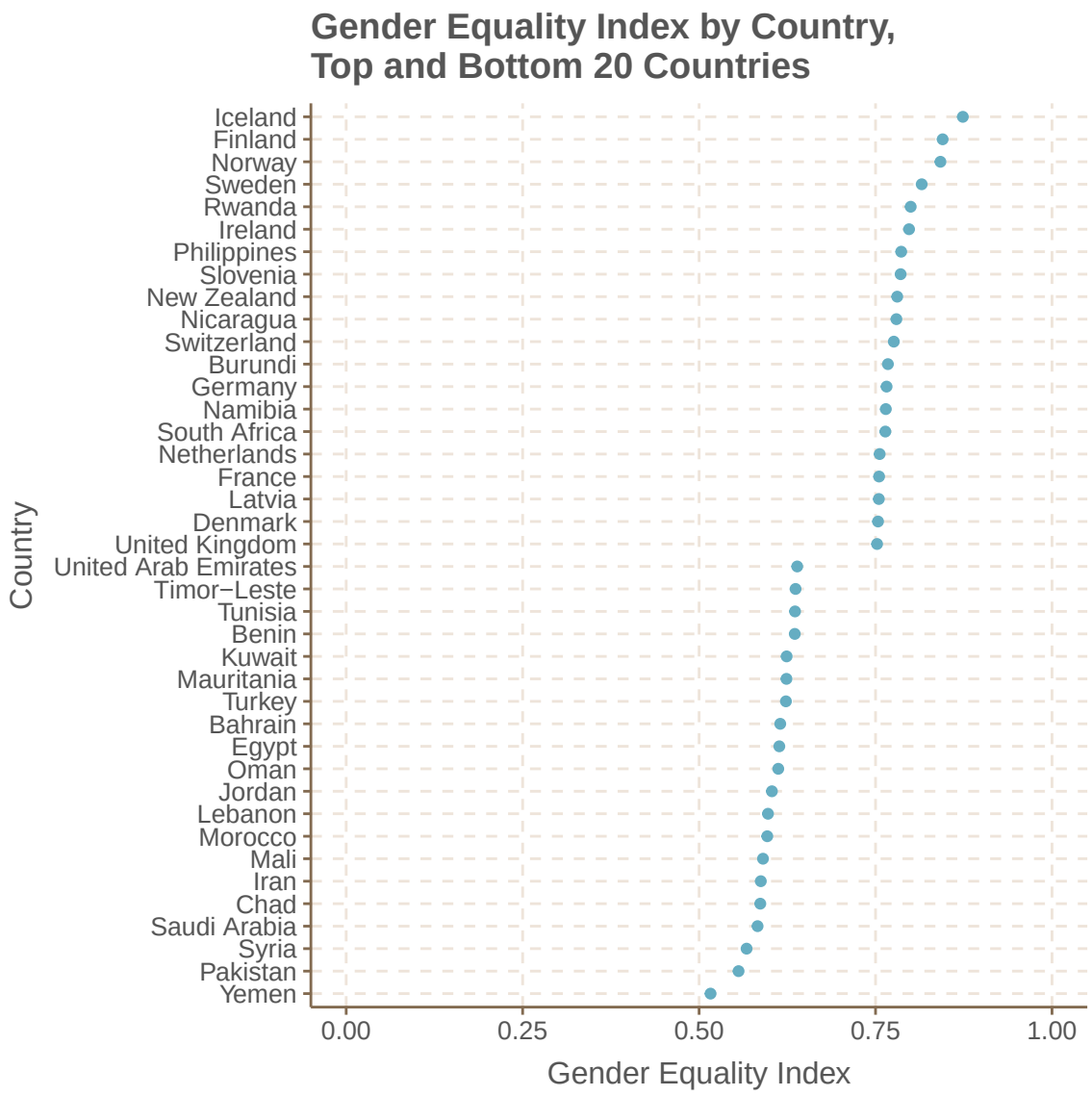
The United States is not among the top 20 countries in terms of happiness; it ranks 21st.

Next, the Human Development Index:



The United States ranks twelfth by HDI.

Exploring gender equality:

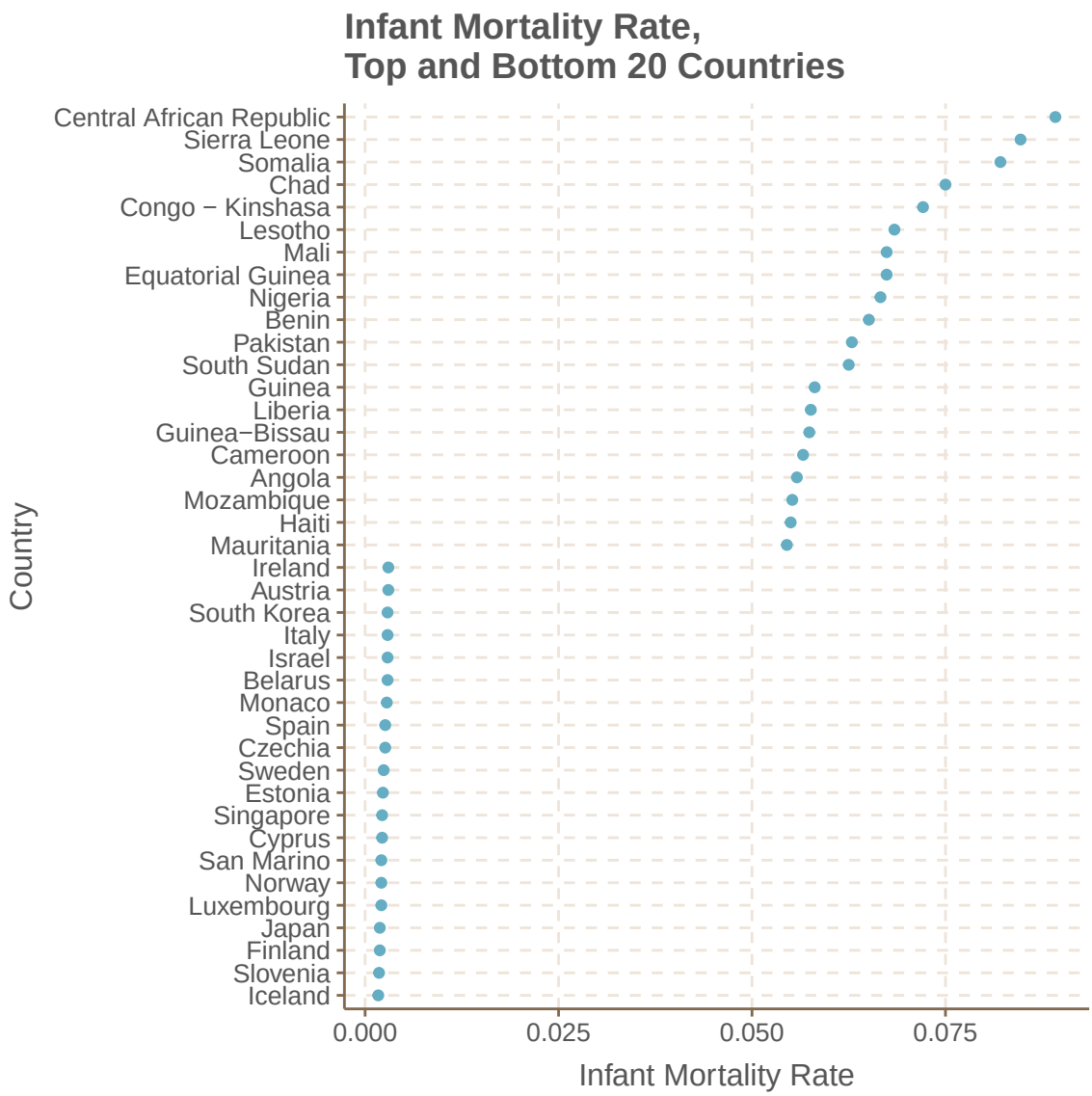



```
which(alldata_gender$country == "United States")
```

```
## [1] 45
```

The United States is not among the top 20 countries in terms of gender equality; it ranks 45th.

Examining infant mortality:

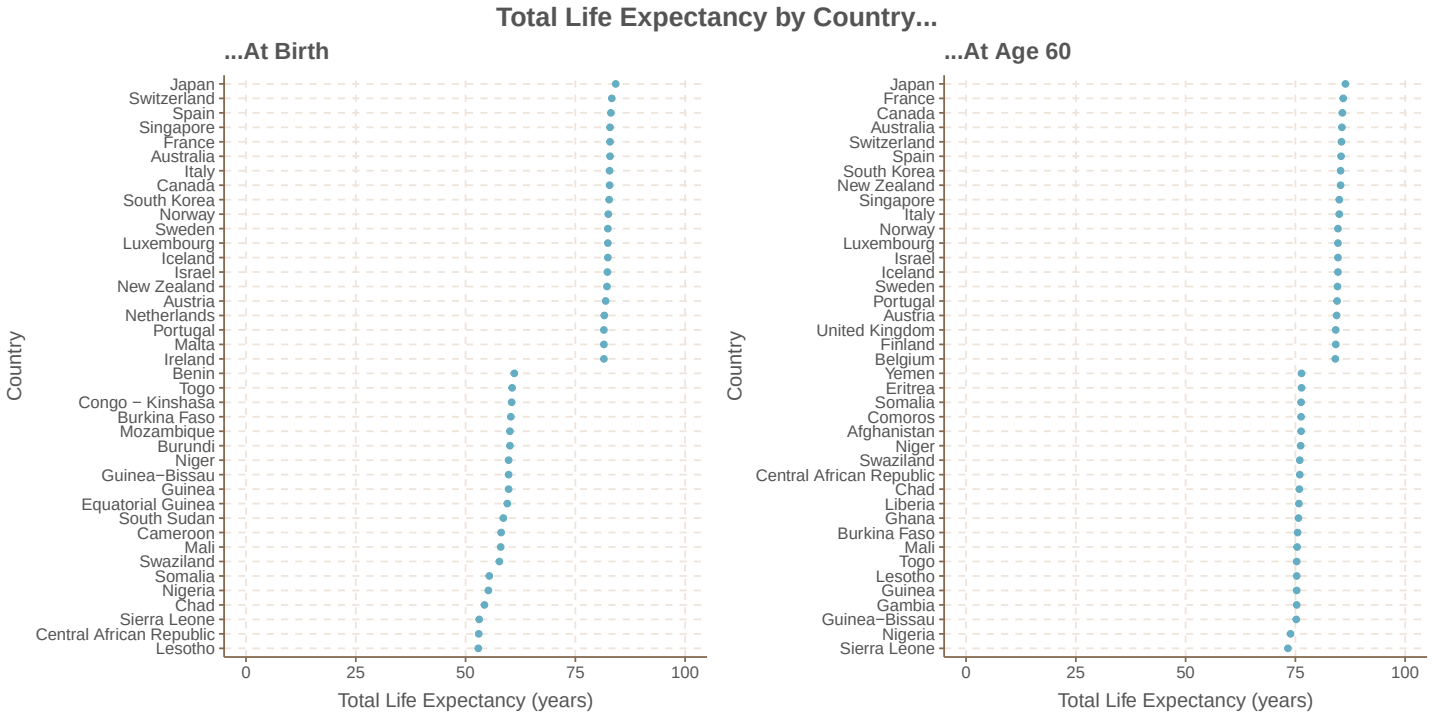



```
alldata_infantmort_asc <- alldata_infantmort %>% arrange(infantmort)
which(alldata_infantmort_asc$country == "United States")
```

```
## [1] 46
```

The United States has the world's 46th lowest infant mortality rate.

Finally, exploring life expectancy:




```
which(alldata_lifeexp_birth$country == "United States")
```

```
## [1] 34
```

```
which(alldata_lifeexp_sixty$country == "United States")
```

```
## [1] 31
```

Once again, the United States is not among the top 20 countries for life expectancy, ranking 34th and 31st respectively for life expectancy at birth and at 60 years of age.

Clustering Analysis

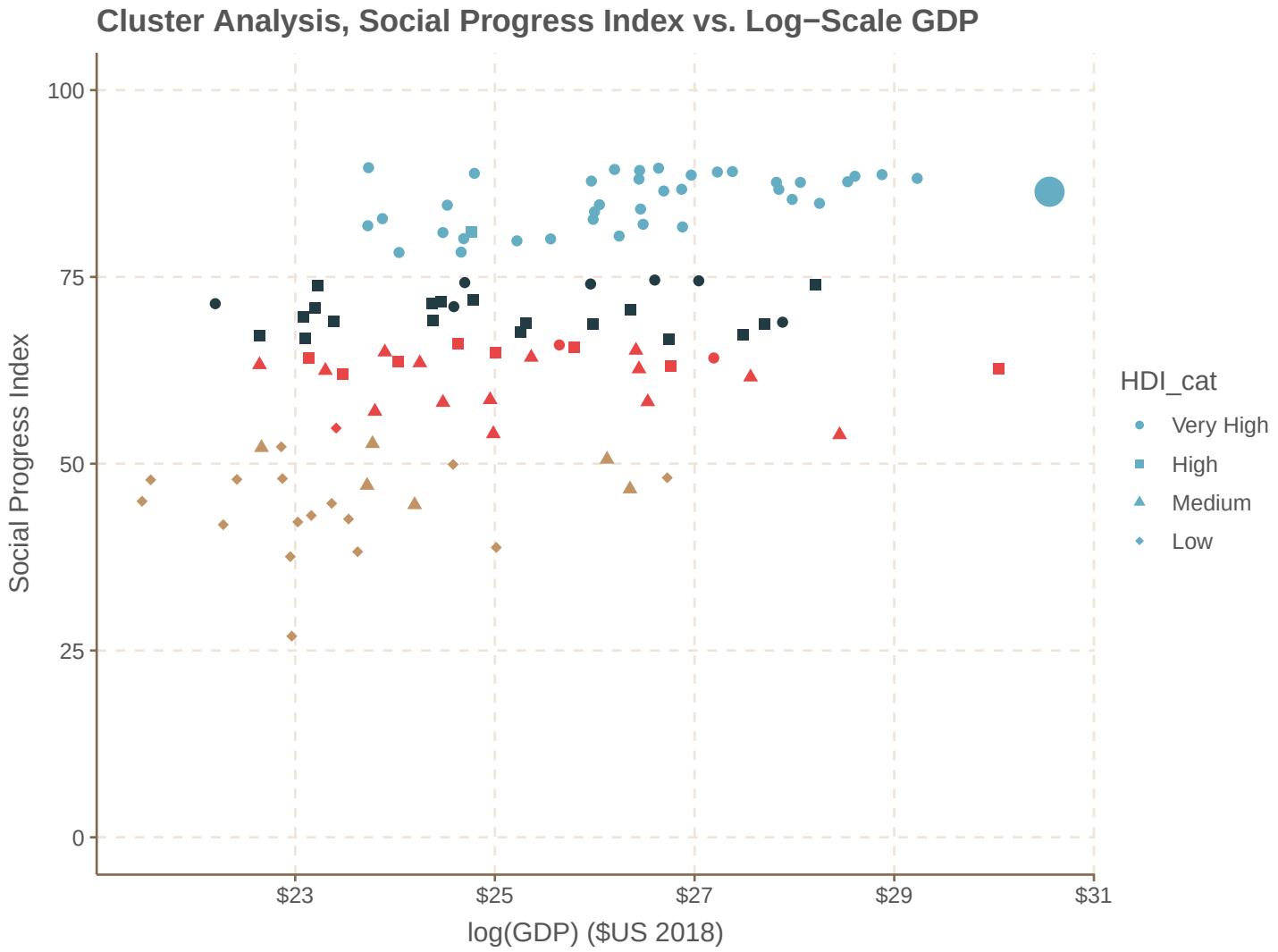
The Human Development Index categorizes the world's countries into four developmental levels (low, medium, high, and very high); thus k-means clustering analysis was performed assuming 4 clusters. Additionally, missing values were excluded to ensure the clustering algorithm would run, and a function was written to subset the clustering dataset (named `clusterdata`) to the variables of interest for each k-means analysis. On each output plot, countries in the "Very High" HDI category are denoted with an empty `geom_point()`, and the United States was identified by an enlarged point.

```
kmdf <- function(data, x, y, z){
  kmdata <- data %>%
    select(x, y, z)
  kmdata <- return(kmdata)
}
```

Clustering happiness versus log GDP:

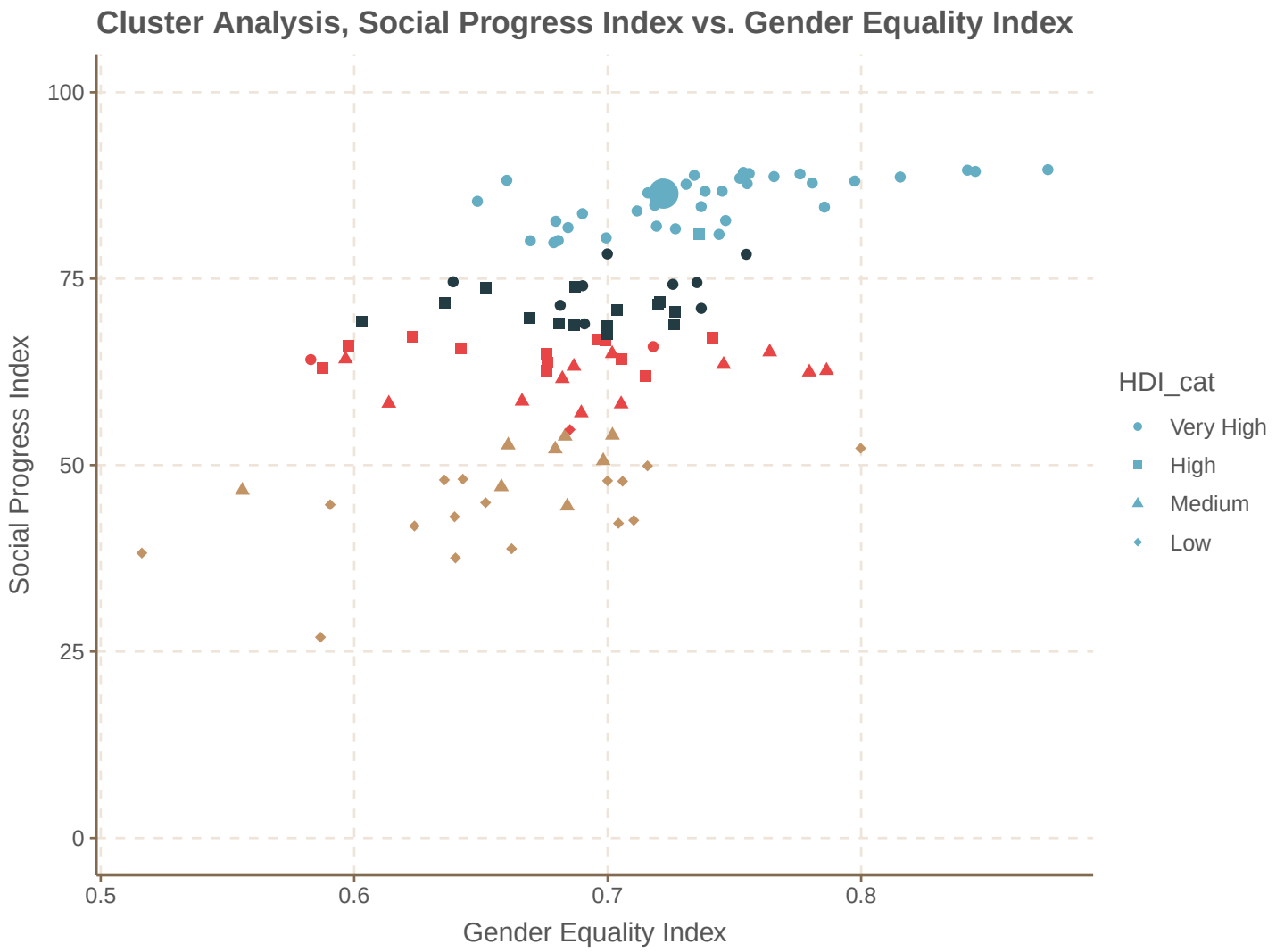
```
kmdata <- kmdf(clusterdata, "country", "SPI", "logGDP")

set.seed(19811221)
km_SPI_GDP <- kmeans(kmdata[, 2:3], 4)
km_SPI_GDP_cluster <- as.factor(km_SPI_GDP$cluster)

clusterdata1 <- cbind(clusterdata, km_SPI_GDP_cluster)

km_SPI_GDP_plot <- ggplot(data = clusterdata1,
                          aes(x = logGDP, y = SPI,
                              color = km_SPI_GDP_cluster,
                              size = US,
```

```
                          shape = HDI_cat)) +
  geom_point() +
  ylim(0, 100) +
  scale_shape_manual(values = c(18, 17, 15, 16)) +
  scale_x_continuous(labels = scales::dollar_format(prefix = "$")) +
  guides(color = FALSE, size = FALSE, shape = guide_legend(reverse = TRUE)) +
  xlab("log(GDP) ($US 2018)") +
  ylab("Social Progress Index") +
  ggtitle("Cluster Analysis, Social Progress Index vs. Log-Scale GDP")
km_SPI_GDP_plot
```

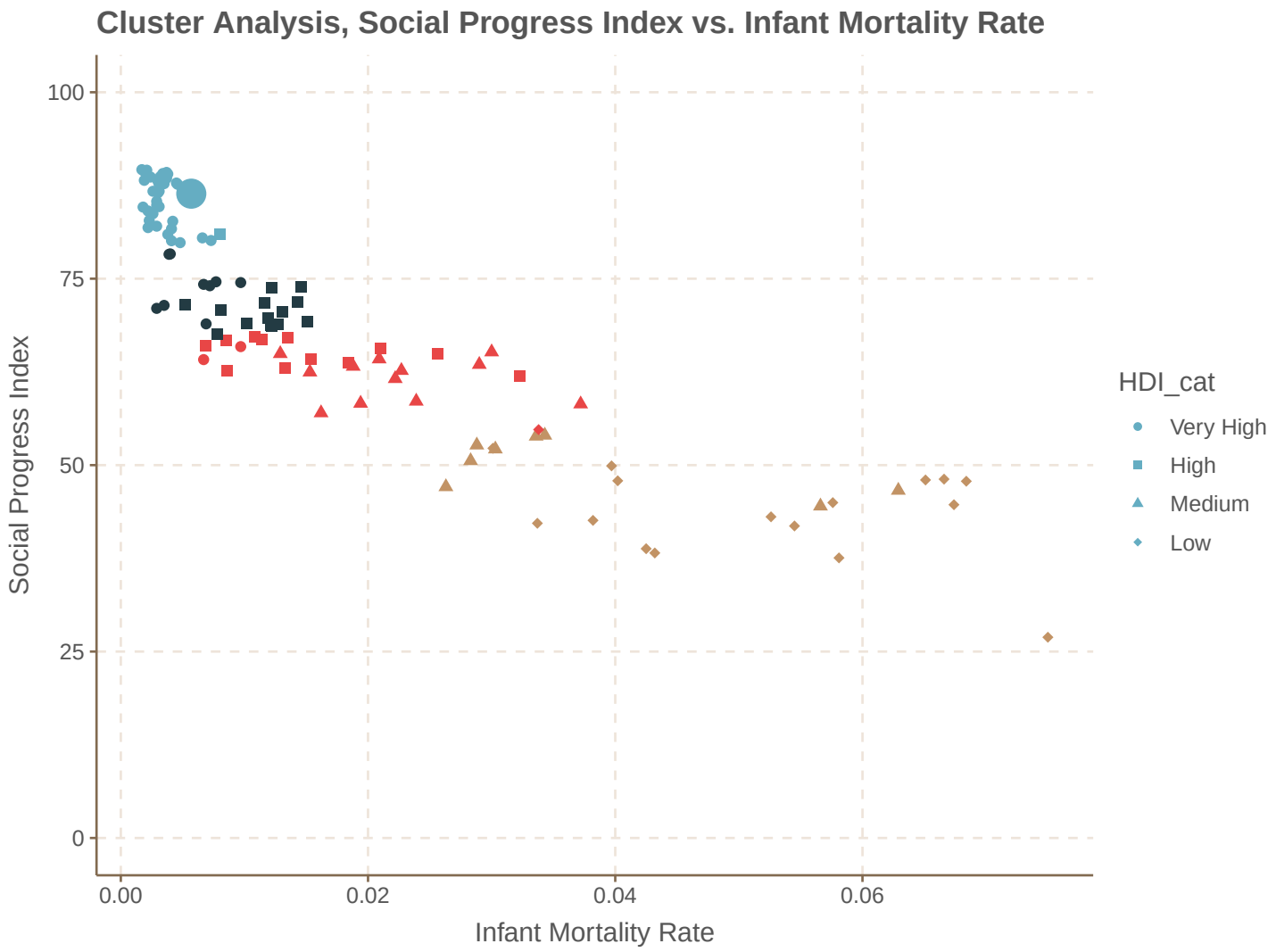


Clustering social progress versus happiness (code for this and subsequent clustering not shown for brevity):

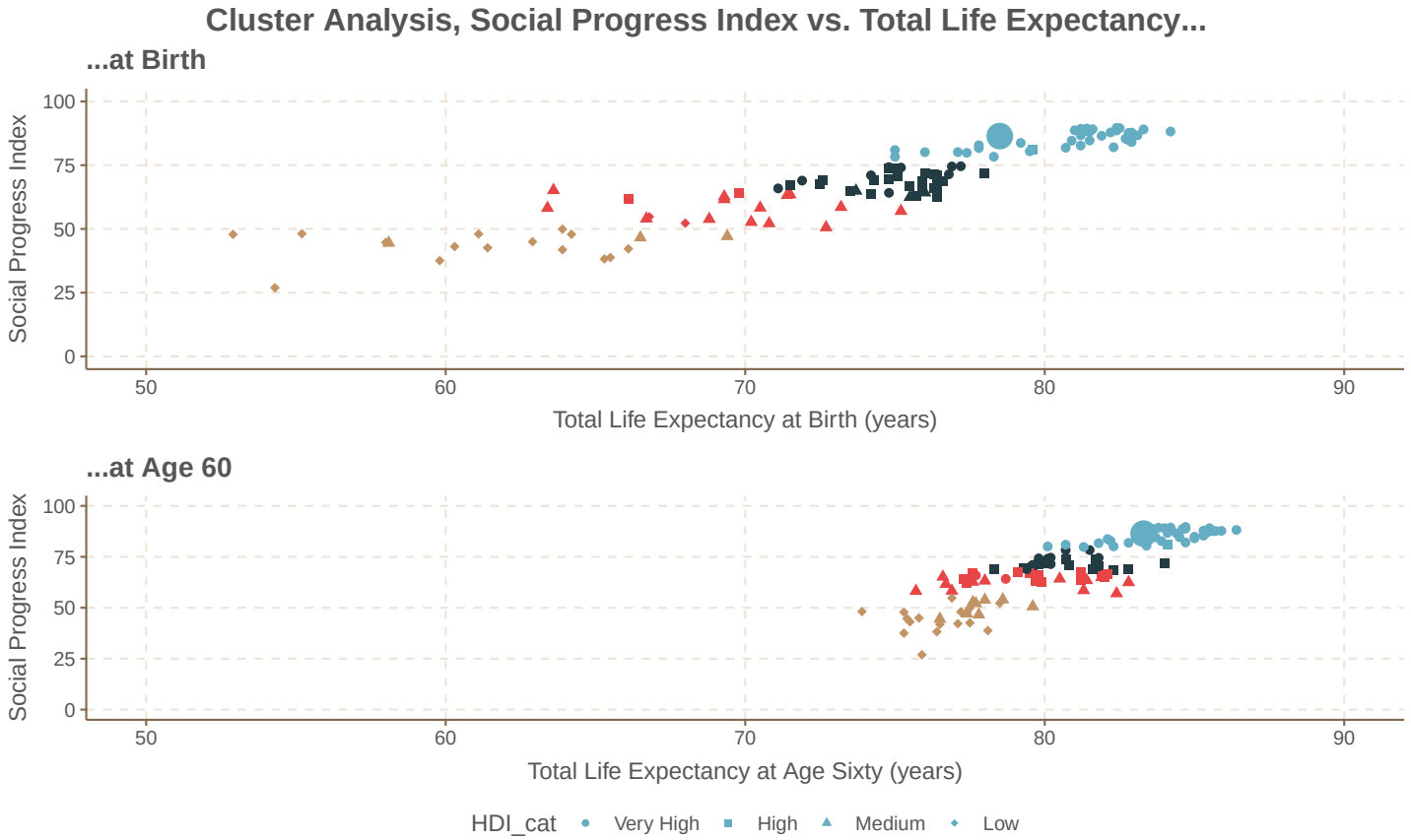
Clustering social progress versus gender equality:



Clustering social progress versus infant mortality:



Clustering social progress versus life expectancy:



Results

Discussion

Limitations

Conclusion

References

Prioli, Katherine M. 2018. "MAT_8790_Final_Project." https://github.com/kmprioliPROF/MAT_8790_Final_Project.

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.

World Economic Forum. 2016. "Gender Equality." http://reports.weforum.org/global-gender-gap-report-2016/rankings/.

World Happiness Report. 2018. "World Happiness Report." http://worldhappiness.report/ed/2018/.

World Health Organization. 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.