Final Exam

Connor Phillips

May 08, 2024

Question 1 [20 points]

I am interested in the average attendance at each World Cup Soccer game dependent on the host country. This data is available as WorldCup.xlsx within the Final Exam assignment page.

```r
library(tidyverse)
library(readxl)
library(ggplot2)
library(lubridate)
library(cowplot)
```

a. Load the data frame. Be sure to load the correct sheet and skip any uninformative lines.

```r
worldCup_df <- data.frame(read_excel(path = "data-raw/WorldCup-1.xlsx",
                                     sheet = "worldcup", range = 'A3:I29'))
```

b. Clean the data you have loaded to include the following columns: Year, Hosts, Matches, Totalattendance, and Averageattendance. You will either need to make your own column names or properly clean the strings given for the column names (they contain special characters that should not be retained). Remove commas from numerical values and ensure the Attendance columns are properly formatted as numerical data. Keep the Year variable as strings. Remove data related to any World Cups that have not occurred and the Overall statistics. Do all cleaning within R. Do not edit the excel file.

```r
worldcup_cleaned <- worldCup_df %>%
  select(Year, Hosts, Matches, Totalattendance, Averageattendance) %>%
  mutate(Year = str_remove(Year, pattern = '\\[n 1]')) %>%
  mutate(Totalattendance = as.numeric(str_remove_all(Totalattendance, pattern = ','))) %>%
  mutate(Averageattendance = as.numeric(str_remove_all(Averageattendance, pattern = ','))) %>%
  filter(Year != 'Overall') %>%
  filter(as.numeric(Year) >= 1930 & as.numeric(Year) <= 2022)
```

c. Some countries have hosted multiple World Cups. Make unique identifiers for each World Cup by pasting together the Host and Year. Create a new column named worldcup that contains these unique identifiers (i.e. Uruguay1930). Remove any remaining spaces in the worldcup names. Remove the Hosts and Year columns when finished.

```r
worldcup_cleaned = worldcup_cleaned %>%
  mutate(worldcup = paste(Hosts, Year, sep = "")) %>%
  mutate(worldcup = str_remove_all(worldcup, pattern = " ")) %>%
  select(-Year, -Hosts)
```

d. Display the first ten rows of the data frame using the head command.

```
head(worldcup_cleaned, n = 10)
```

```
##    Matches Totalattendance Averageattendance        worldcup
## 1       18          590549             32808     Uruguay1930
## 2       17          363000             21353       Italy1934
## 3       18          375700             20872      France1938
## 4       22         1045246             47511      Brazil1950
## 5       26          768607             29562 Switzerland1954
## 6       35          819810             23423      Sweden1958
## 7       32          893172             27912       Chile1962
## 8       32         1563135             48848     England1966
## 9       32         1603975             50124      Mexico1970
## 10      38         1865753             49099 WestGermany1974
```

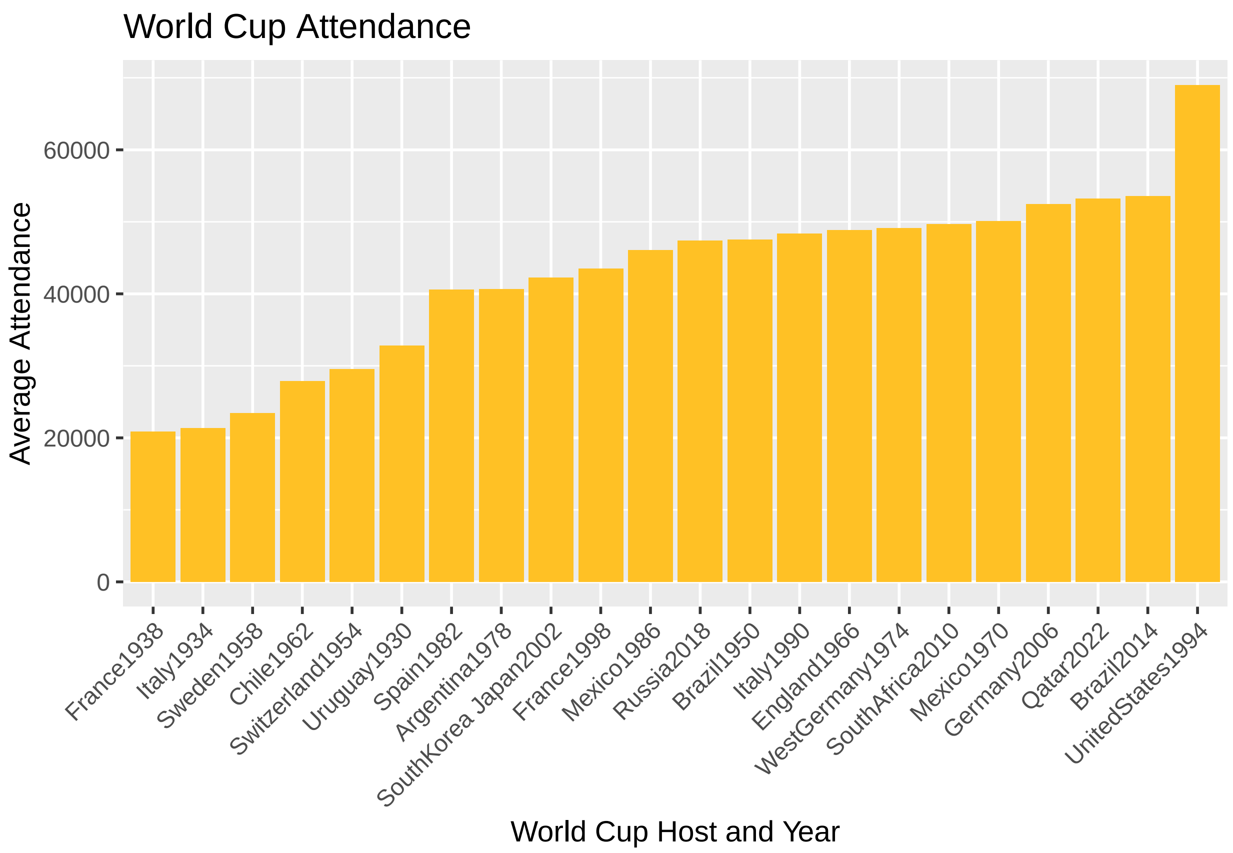
e. Display the structure of the data frame.

```
str(worldcup_cleaned)
```

```
## ’data.frame’:    22 obs. of  4 variables:
##  $ Matches          : num  18 17 18 22 26 35 32 32 32 38 ...
##  $ Totalattendance  : num  590549 363000 375700 1045246 768607 ...
##  $ Averageattendance: num  32808 21353 20872 47511 29562 ...
##  $ worldcup         : chr  "Uruguay1930" "Italy1934" "France1938" "Brazil1950" ...
```

f. Create a column graph displaying worldcup against the Averageattendance. Arrange the graph such that the bars are ordered by average attendance. Make sure the worldcup identifiers are visible on the graph (i.e. you can read them). Clean up the axes such that they read World Cup Host and Year and Average Attendance.

```
ggplot(worldcup_cleaned) +
  geom_col(aes(x = reorder(worldcup, Averageattendance), y = Averageattendance),
           fill='goldenrod1') +
  labs(title = 'World Cup Attendance',
       x = 'World Cup Host and Year',
       y = 'Average Attendance') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



`Question 2 [20 points]`

Considering the average attendance at World Cup matches got me thinking about world population. I was able to find an excel file from the United Nations tracking estimated populations for all countries that are part of the UN. This data is available as World_Populations.xlsx within the Final Exam assignment page.

a. Load the data frame the ESTIMATES tab. Be sure to skip any uninformative lines.

```r
worldPop_df <- data.frame(read_excel(path = "data-raw/World_Population.xlsx",
                                     sheet = "ESTIMATES", range = 'A17:BZ306'))
```

b. Using regular expressions and tidyverse commands, clean the data to include only population information from 1950 to 2020 for all countries. Remove all extra information regarding regions, subregions, income, etc. Retain only the Country Name and population estimates for years 1950 to 2020. Name this data.frame WorldPopulation. Do all the data cleaning within R. Do not edit the excel file.

```r
WorldPopulation = worldPop_df %>%
  mutate(Country = Region..subregion..country.or.area..) %>%
  filter(Type == "Country/Area") %>%
  select(Country, starts_with("X")) %>%
  pivot_longer(X1950:X2020,
               names_to = "Year",
               values_to = "Population") %>%
  mutate(Year = as.numeric(str_sub(Year, start = 2))) %>%
  mutate(Year = make_date(year = Year)) %>%
  mutate(Population = as.integer(Population))
```

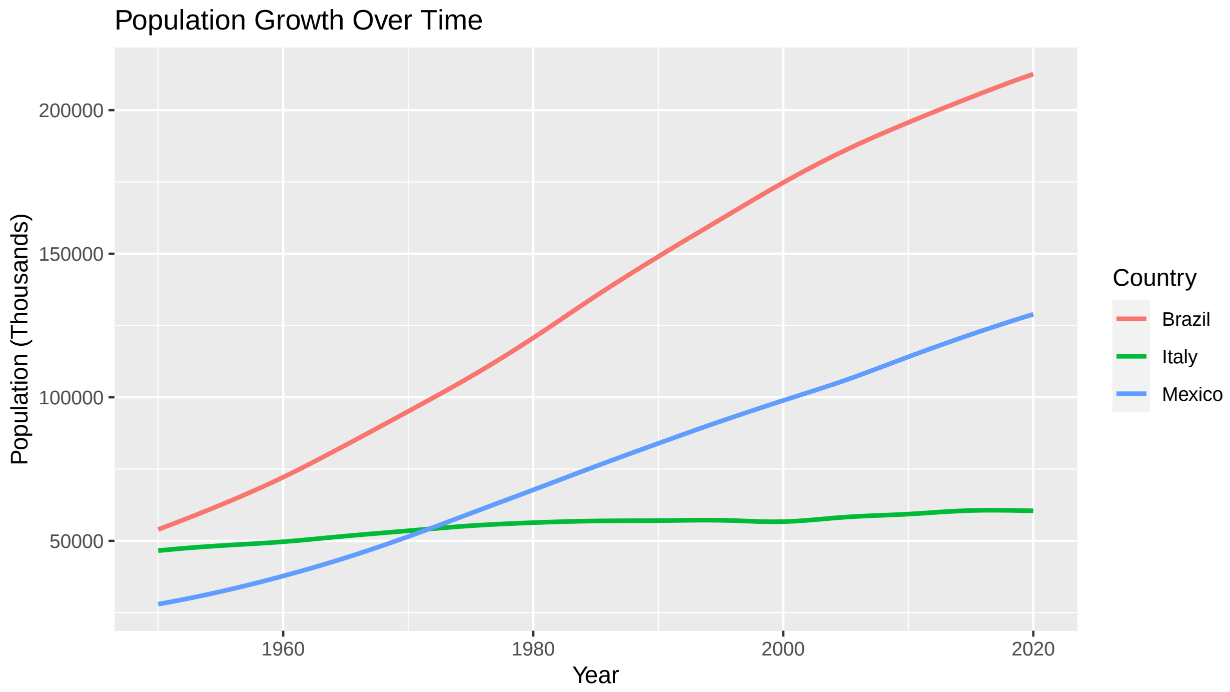
c. Create a single panel graph displaying Year against Population for Brazil, Mexico, and Italy. Use different colors for the three countries. Properly label the axes.

```
threeCountries <- WorldPopulation %>%
  filter(Country %in% c("Italy", "Mexico", "Brazil"))

countriesPlotted <- ggplot(threeCountries, aes(x=Year, y = Population, group = Country, color = Country)
  geom_line(size = 1) +
  labs(title = "Population Growth Over Time",
       y = "Population (Thousands)",
       color = "Country")
```

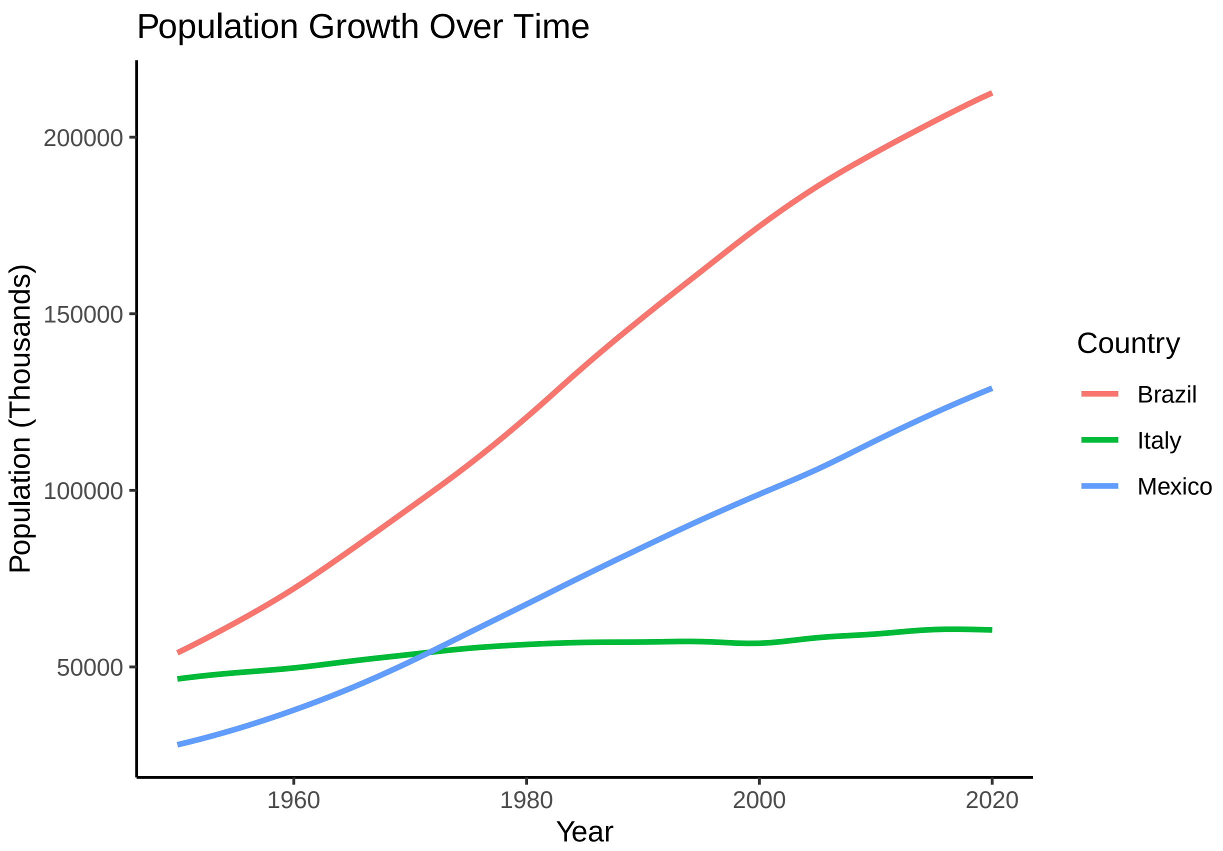
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
countriesPlotted
```



d. Apply a theme of your choice to the graph in part c..

```
countriesPlotted +
  theme_classic()
```



Question 3 [25 points]

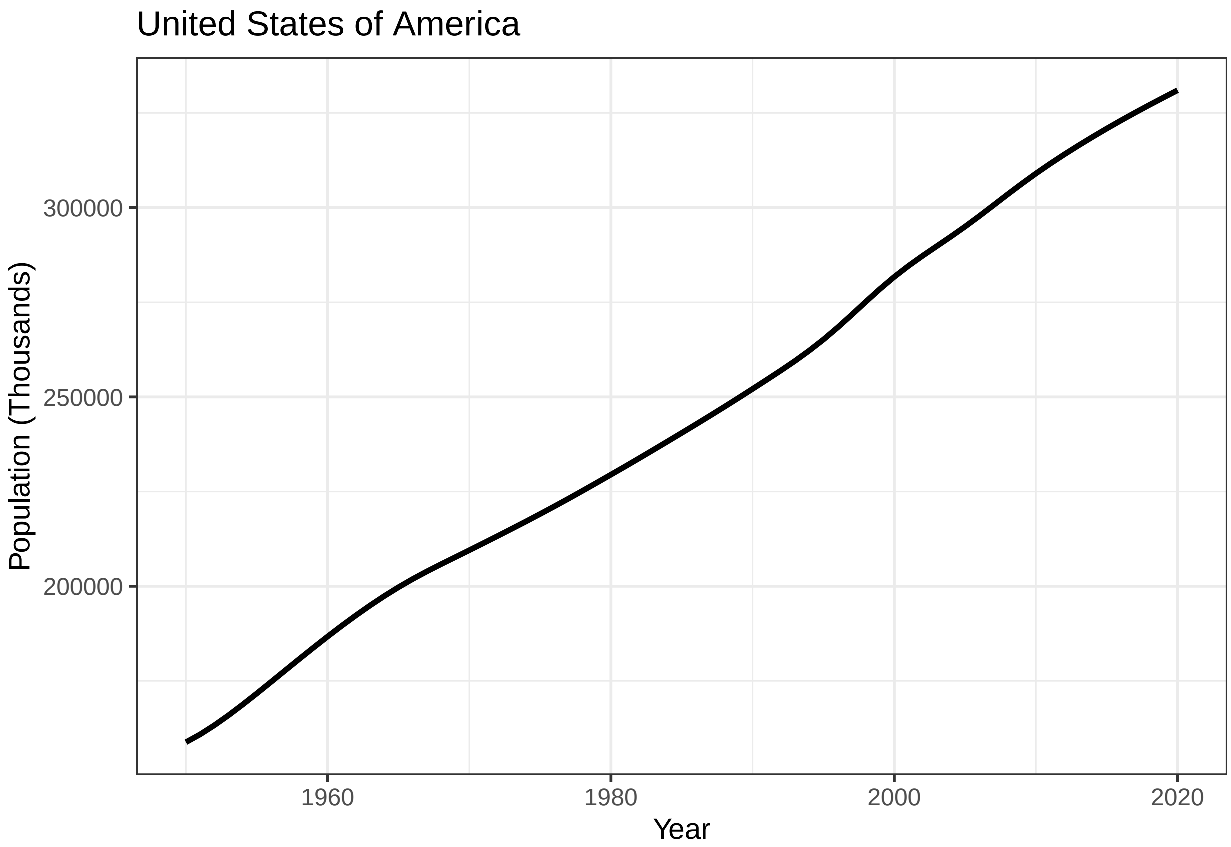
I want to be able to easily graph any of the UN countries given in the Excel file for Question 2. My preference would be to just enter a country name and obtain a graph of the population from 1950 to 2020.

a. Produce a function that is hard coded to use the data.frame produced in Question 2 part b. The function should take as input a country name (as a string) and return the population against year graph for that country. Name this function CountryPopulation. Hint: Wrap up what you did Question 2c into a function that returns an object that is a ggplot. Remove any options for color. Add an option for title that uses the string that is input. This should produce a black and white graph with the name of the country at the top.

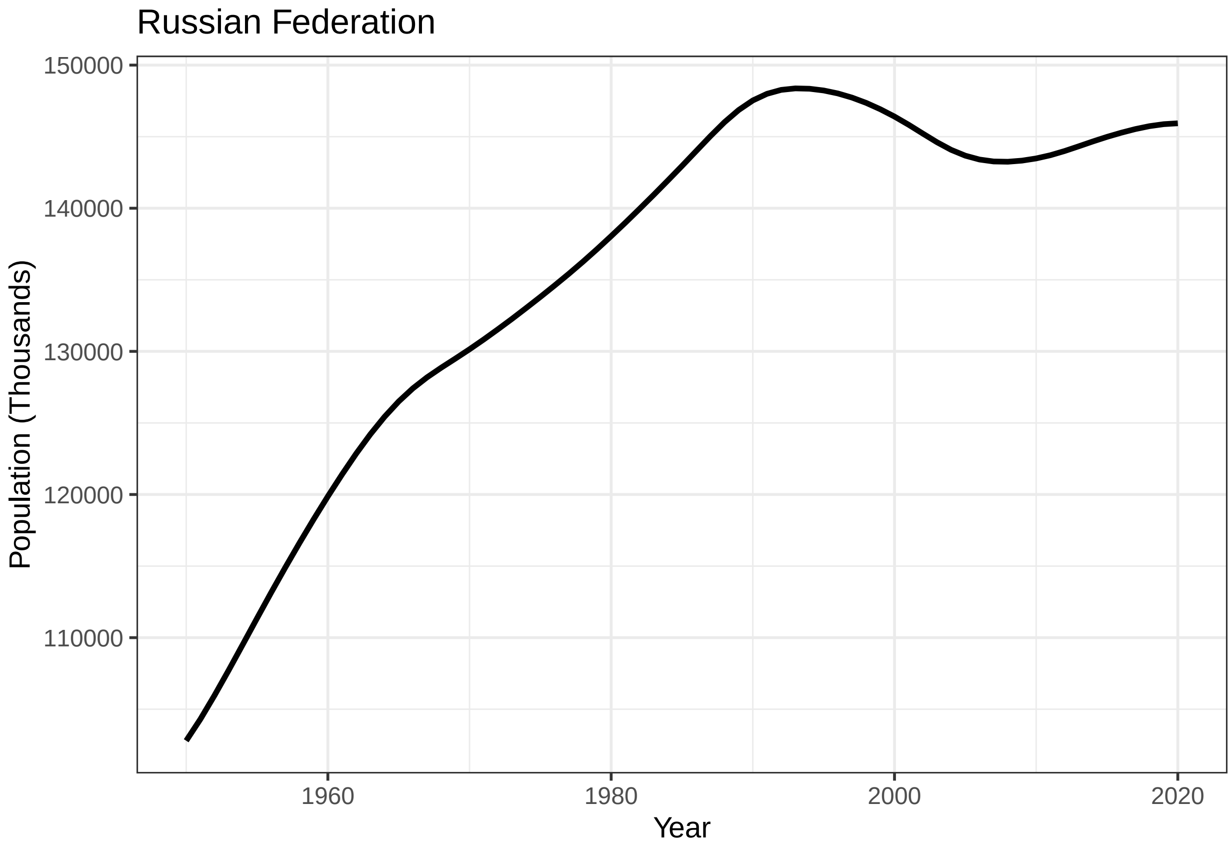
```
CountryPopulation <- function(country){
  country <- as.character(country)
  countryFilt <- WorldPopulation %>%
    filter(Country == country)
  ggplot(countryFilt, aes(x=Year, y = Population)) +
  geom_line(size = 1) +
  labs(title = country,
       y = "Population (Thousands)") +
  theme_bw()
}
```

b. Produce graphs for United States of America, Russian Federation, China, and United Kingdom using your function. You may store these as objects to be used in part c.. Display at least one of the four graphs produced.

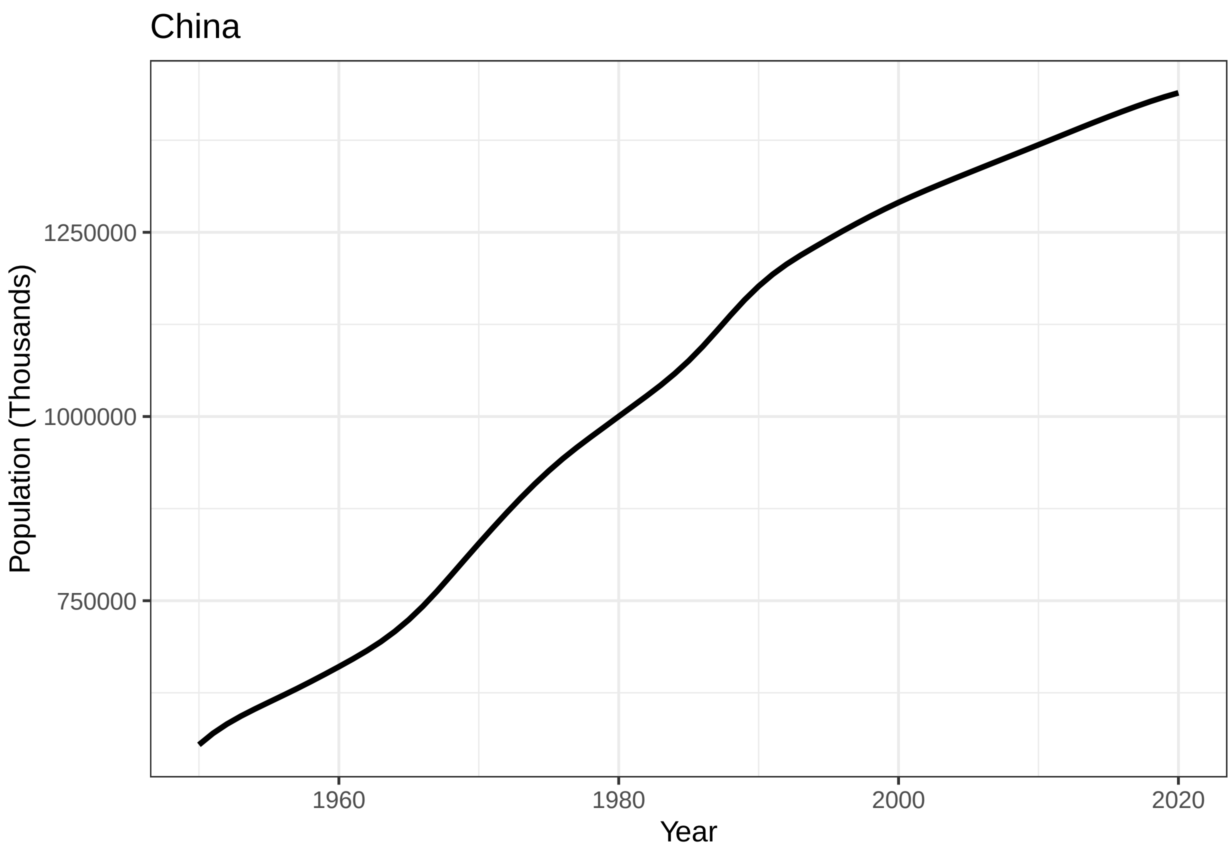
```
usGraph <- CountryPopulation("United States of America")
usGraph
```



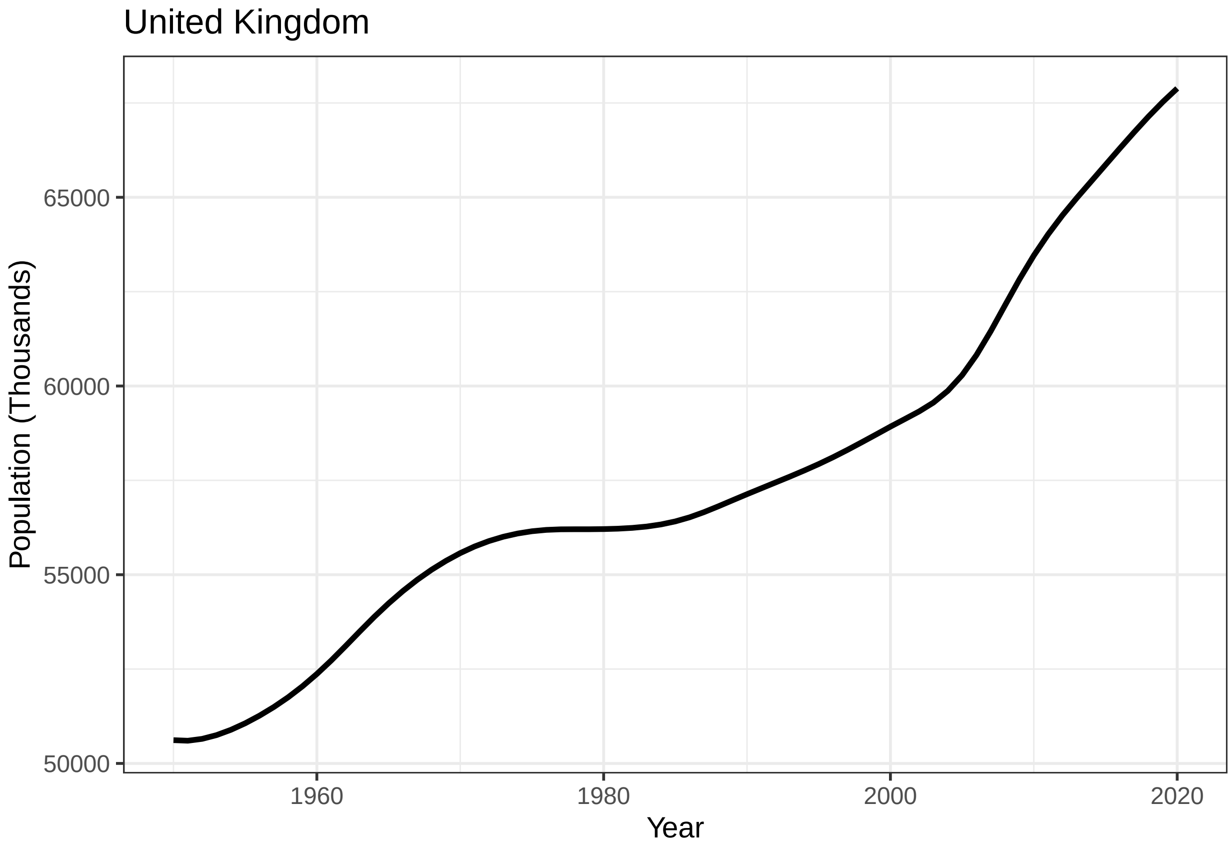
```
rfGraph <- CountryPopulation("Russian Federation")
rfGraph
```



```
chinaGraph <- CountryPopulation("China")
chinaGraph
```

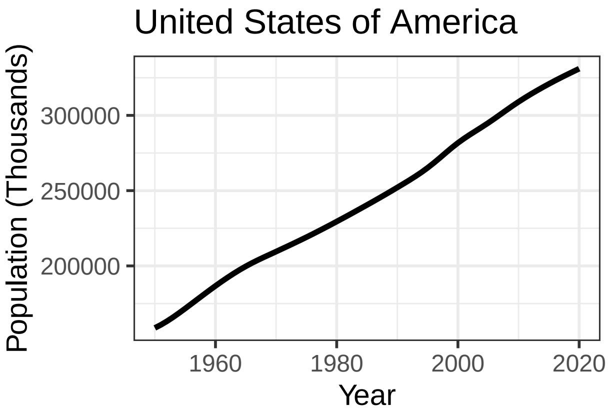


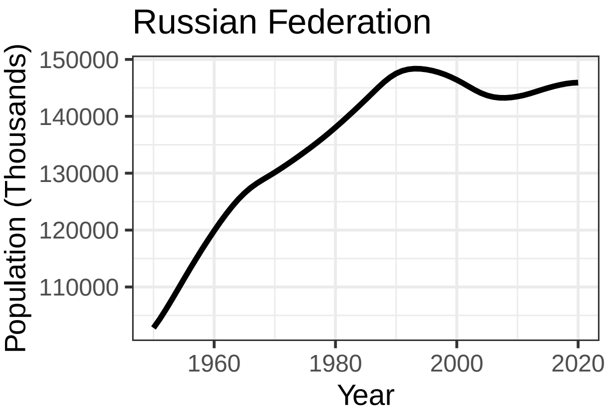
```
ukGraph <- CountryPopulation("United Kingdom")
ukGraph
```

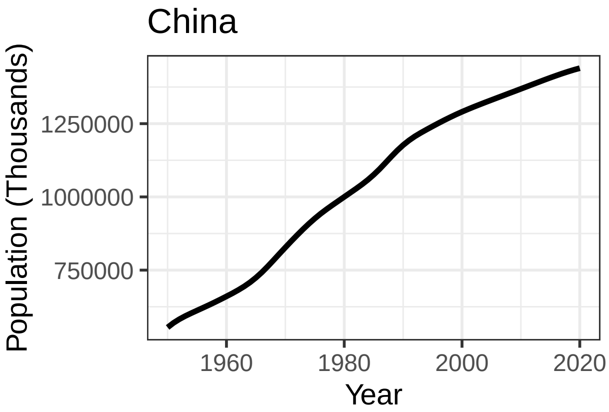


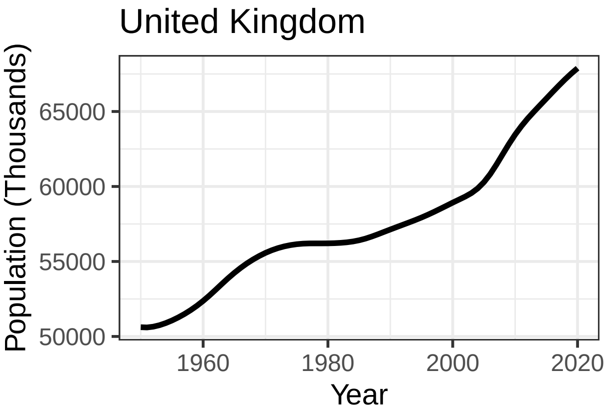
c. Using cowplot or 'patchwork' combine the four graphs from part b. into a single panel.

```
plot_grid(usGraph, rfGraph, chinaGraph, ukGraph)
```









Question 4 [25 points]

This problem should be listed on the submitted PDF and provide a link to your GitHub package. We now have some really interesting population data and a function that allows us to view any population graphs of UN countries. Let's package this up with some additional troubleshooting. Follow the steps below and ensure you upload the package to GitHub.

a. Initialize a new package named YourLastNameWorldPopulation.

b. Add the World_Population.xlsx file to the data-raw folder.

c. Using your cleaning script from Question 2b, add the cleaned version of the world population data to the package. Call this data set WorldPopulation, ensure it gives only the columns as given in Question 2b. Make sure to document the data set.

d. Add to your package the function CountryPopulation. Be sure to include a description for the documentation.

e. Add a unit test to the :package to check if a country name entered is in the cleaned data file WorldPopulation. If the country is not present, then the function CountryPopulation should return an error.

f. Compile your package and upload to GitHub within the repository YourLastNameWorldPopulation. As a solution to Question 4, provide the link to your GitHub package.

Here is the link: https://github.com/connorphillips10/PhillipsWorldPopulation