Joining/Merging

Author: Melina Padron

Question 1:

The dataset `world_bank_pop` is a built-in dataset in `tidyverse`. It contains information about total population and population growth, overall and more specifically in urban areas, for countries around the world. Take a look at it with `head()`. Is the data tidy? Why or why not?

```
# Call tidyr, dplyr and ggplot2 packages within tidyverse
library(tidyverse)

# Take a look!
head(world_bank_pop)
```

```
## # A tibble: 6 x 20
##   country indic~1 ‘2000‘  ‘2001‘  ‘2002‘  ‘2003‘  ‘2004‘  ‘2005‘  ‘2006‘  ‘2007‘
##   <chr>   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 ABW     SP.URB~ 4.16e4 4.20e+4 4.22e+4 4.23e+4 4.23e+4 4.24e+4 4.26e+4 4.27e+4
## 2 ABW     SP.URB~ 1.66e0 9.56e-1 4.01e-1 1.97e-1 9.46e-2 1.94e-1 3.67e-1 4.08e-1
## 3 ABW     SP.POP~ 8.91e4 9.07e+4 9.18e+4 9.27e+4 9.35e+4 9.45e+4 9.56e+4 9.68e+4
## 4 ABW     SP.POP~ 2.54e0 1.77e+0 1.19e+0 9.97e-1 9.01e-1 1.00e+0 1.18e+0 1.23e+0
## 5 AFE     SP.URB~ 1.16e8 1.20e+8 1.24e+8 1.29e+8 1.34e+8 1.39e+8 1.44e+8 1.49e+8
## 6 AFE     SP.URB~ 3.60e0 3.66e+0 3.72e+0 3.71e+0 3.74e+0 3.81e+0 3.81e+0 3.61e+0
## # ... with 10 more variables: ‘2008‘ <dbl>, ‘2009‘ <dbl>, ‘2010‘ <dbl>,
## #   ‘2011‘ <dbl>, ‘2012‘ <dbl>, ‘2013‘ <dbl>, ‘2014‘ <dbl>, ‘2015‘ <dbl>,
## #   ‘2016‘ <dbl>, ‘2017‘ <dbl>, and abbreviated variable name 1: indicator
```

Answer: The data is not tidy as the values in the indicators column should be their own columns. Moreover, the data would be tidy if the years were under one column and the values would be the individual years. Overall, a tidy dataset would have all the variables as columns, observations as rows, and values in cells.

Question 2:

Using `dplyr` functions on `world_bank_pop`, count how many distinct countries there are in the dataset. Does this makes sense? Why or why not?

```
# Count the number of distinct countries in the dataset
world_bank_pop %>%
  distinct(country)
```

```
## # A tibble: 266 x 1
##    country
##    <chr>
##  1 ABW
##  2 AFE
##  3 AFG
##  4 AFW
##  5 AGO
##  6 ALB
##  7 AND
##  8 ARB
##  9 ARE
## 10 ARG
## # ... with 256 more rows
```

Answer: There are 266 distinct countries in the dataset. This does not make sense as the UN only recognizes 193 member states and 2 non-member observer states

---

Question 3:

Use one of the `pivot` functions on `world_bank_pop` to create a new dataset with the years 2000 to 2017 appearing as a *numeric* variable `year`, and the different values for the indicator variable are in a variable called `value`. Save this new dataset in your environment as `myworld1`.

```r
# Pivot the dataset to have year and value variables
myworld1 <- world_bank_pop %>%
  pivot_longer(cols = starts_with("20"),
               names_to = "year",
               values_to = "value") %>%
  mutate(year = as.numeric(year))
```

How many rows are there per country? Why does it make sense?

```r
# Count the number of rows per country
myworld1 %>%
  count(country)
```

```
## # A tibble: 266 x 2
##    country     n
##    <chr>   <int>
##  1 ABW        72
##  2 AFE        72
##  3 AFG        72
##  4 AFW        72
##  5 AGO        72
##  6 ALB        72
##  7 AND        72
##  8 ARB        72
##  9 ARE        72
## 10 ARG        72
## # ... with 256 more rows
```

Answer: There are 72 rows in each country. This answer makes sense as there are 18 years total from 200-2017 and four indicators recorded. Thus, it is 18*4 = 72.

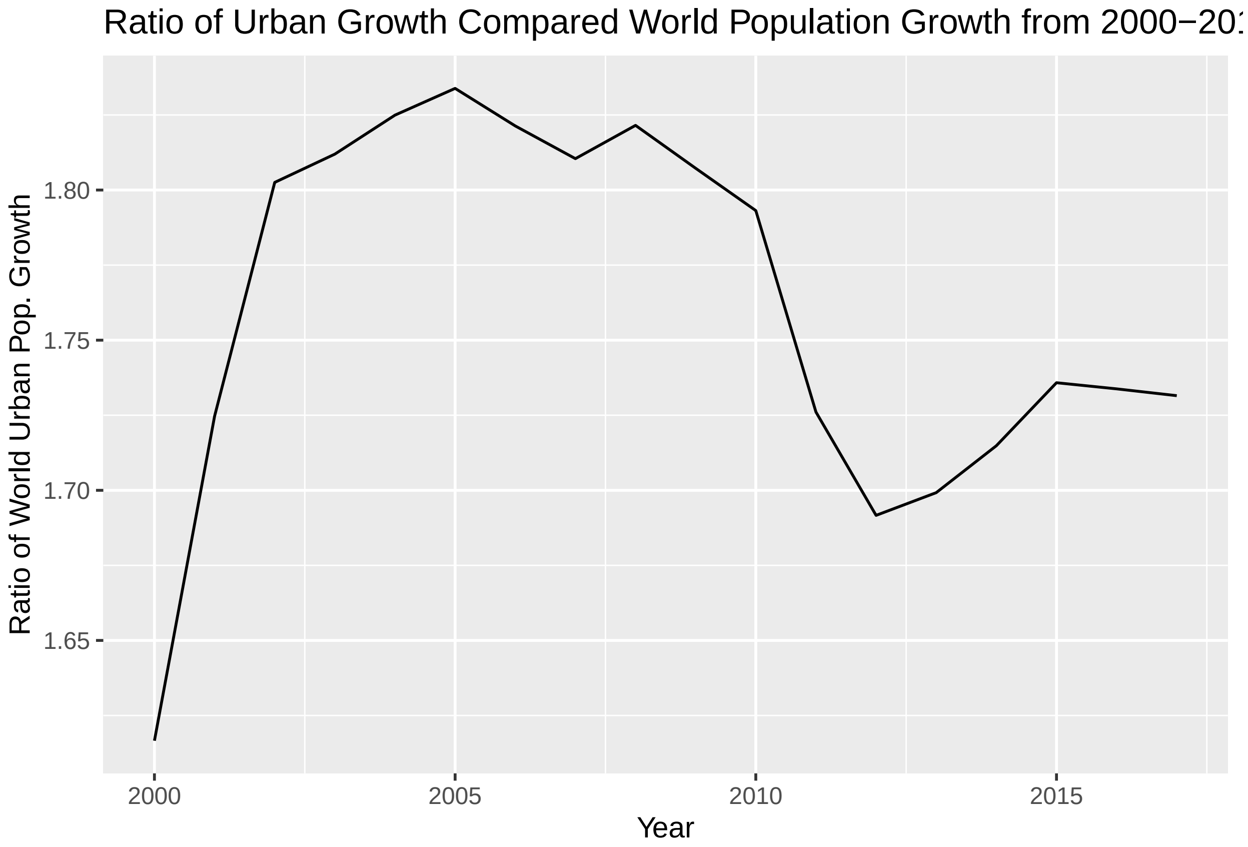
Question 4:

Use another `pivot` function on `myworld1` to create a new dataset, `myworld2`, with the different categories for the indicator variable appearing as their own variables. Use `dplyr` functions to rename `SP.POP.GROW` and `SP.URB.GROW`, as `pop_growth` and `pop_urb_growth` respectively.

```
# Pivot the dataset to have indicator variables as columns and rename them
myworld2 <- myworld1 %>%
  pivot_wider(names_from = indicator,
              values_from = value) %>%
  rename(pop_growth = SP.POP.GROW,
         pop_urb_growth = SP.URB.GROW)
```

Using `dplyr` functions, find the ratio of urban growth compared to the population growth in the world for each year. *Hint: the country code `WLD` represents the entire world.* Create a `ggplot` to display how the percentage of urban population growth has changed over the years. Why does your graph not contradict the fact that the urban population worldwide is increasing over the years?

```
# Create a ggplot to display the percentage of urban population growth over time
myworld2 %>%
  filter(country == "WLD") %>%
  mutate(growth_ratio = pop_urb_growth / pop_growth) %>%
  ggplot(aes(x = year, y = growth_ratio)) +
  geom_line() +
  labs( x = "Year",
        y = "Ratio of World Urban Pop. Growth",
        title ="Ratio of Urban Growth Compared World Population Growth from 2000-2017")
```



Answer: The graph does not contradict the fact that the urban population worldwide is increasing over the years because while the absolute number of people living in urban areas is increasing, so is the total global population, which means that the urban population growth rate is slower than the overall population growth rate. Thus, the plot shows that although the ratio of urban growth compared to the population growth in the world fluctuates from year to year, it has generally been increasing over time.

Question 5:

In `myworld2`, which country code had the highest population growth in 2017? *Hint: Use the* `arrange()` *function here.*

```
# Find which country code had the highest population growth in 2017
myworld2 %>%
  filter(year == 2017) %>%
  arrange(desc(pop_growth)) %>%
  slice(1) %>%
  select(country)
```

```
## # A tibble: 1 x 1
##   country
##   <chr>
## 1 QAT
```

Answer: The country code that had the highest population growth in 2017 was QAT, which is Qatar.

Question 6:

When answering the previous, we only reported the three-letter code and (probably) have no idea what the actual country is. We will now use the package `countrycode` with a built-in dataset called `codelist` that has information about the coding system used by the World bank:

Using `dplyr` functions, modify `mycodes` above to only keep the variables `continent`, `wb` (World Bank code), and `country.name.en` (country name in English). Then remove countries with missing `wb` code.

```r
# Paste and run the following into your console (NOT HERE): install.packages("countrycode")

# Call the countrycode package
library(countrycode)

# Create a list of codes with matching country names
mycodes <- codelist #  Cleaned mycodes to only keep certain variables
mycodes <- mycodes %>%
  select(continent, wb, country.name.en) %>%
  filter(!is.na(wb))
```

How many countries are there in `mycodes`?

```r
# Count how many countries there are
mycodes %>%
  distinct(country.name.en)
```

```
## # A tibble: 218 x 1
##    country.name.en
##    <chr>
##  1 Afghanistan
##  2 Albania
##  3 Algeria
##  4 American Samoa
##  5 Andorra
##  6 Angola
##  7 Antigua & Barbuda
##  8 Argentina
##  9 Armenia
## 10 Aruba
## # ... with 208 more rows
```

Answer: There are 218 countries in “mycodes”.

Question 7:

Use a `left_join()` function to add the information of the country codes **to** `myworld2` dataset. Match the two datasets based on the World Bank code. *Note: the column containing the World Bank code does not have the same name in each dataset.* Using `dplyr` functions, only keep the data available for Europe and for the year 2017. Save this new dataset as `myeurope`.

```
# Join myworld2 and mycodes
# Keep only data available for Europe and for the year 2017
myeurope <- left_join(myworld2,
                      mycodes,
                      by = c("country" = "wb"))%>%
  filter(continent == "Europe", year == 2017)
```

How many rows are there in this new dataset `myeurope`? What does each row represent?

```
# Count rows
count(myeurope)
```

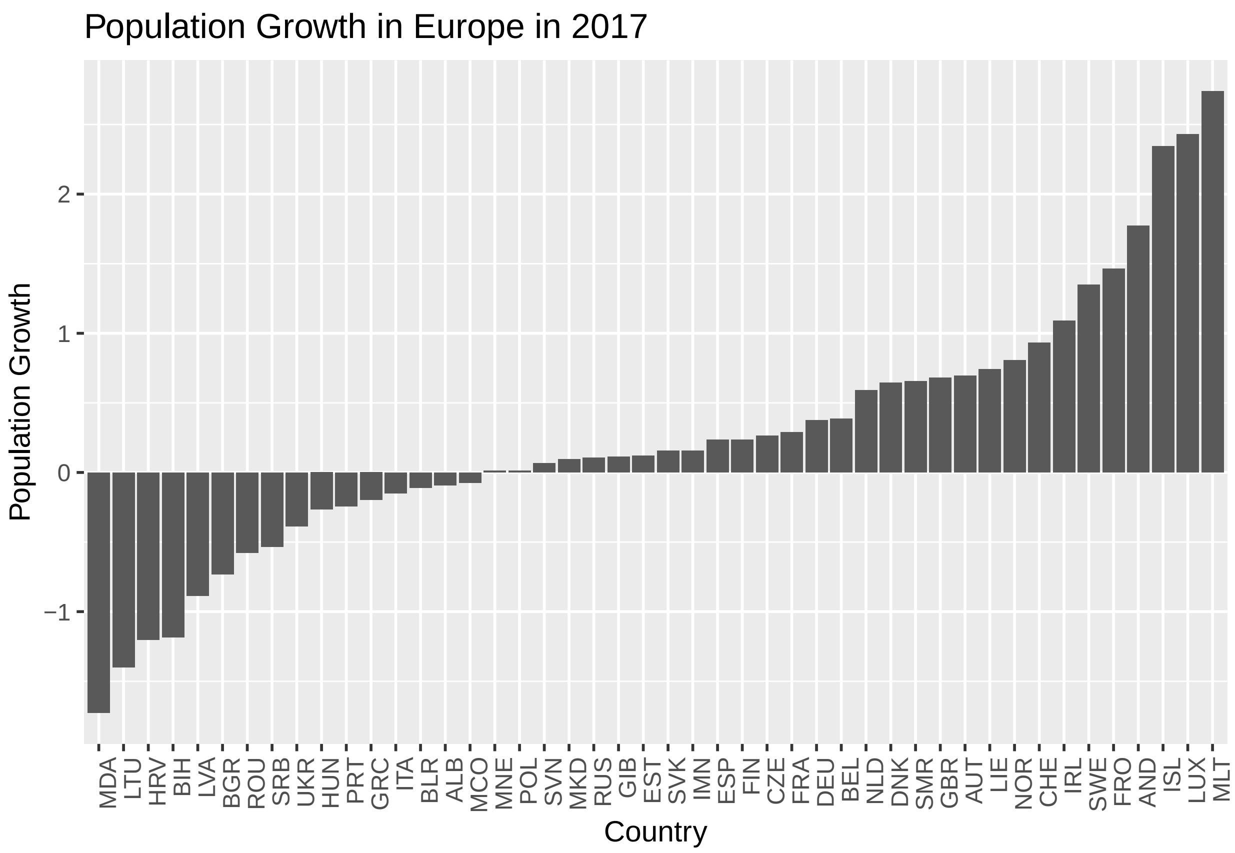
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    46
```

Answer: There are 46 rows in this new dataset "myeurope". Each row in this dataset represents a country in Europe with available data for the year 2017, along with its corresponding population growth, total population, urban population growth, and total urban population.

Question 8:

Using `dplyr` functions on `myeurope`, only keep information for the population growth in 2017 then compare the population growth per country with `ggplot` using `geom_bar()`. Use the `reorder()` function to order countries in order of population growth. Which country in Europe had the lowest population growth in 2017?

```
# Created a plot to compare the population growth per country
myeurope %>%
  filter(year == 2017) %>%
  ggplot(aes(x = reorder(country, pop_growth),
             y = pop_growth)) +
  geom_bar(stat = "identity")+
  labs(x = "Country",
       y = "Population Growth",
       title = "Population Growth in Europe in 2017") +
  theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust = 1))
```



Answer: The country in Europe that had the lowest population growth in 2017 is MDA, which is Moldova.

Question 9:

When dealing with location data, we can actually visualize information on a map if we have geographic information such as latitude and longitude. Next, we will use a built-in function called `map_data()` to get geographic coordinates about countries in the world (see below). Take a look at the dataset `mapWorld`. What variables could we use to join `mapWorld` and `myeurope`? *Note: the variables do not have the same name in each dataset but they contain the same information.*

```
# Geographic coordinates about countries in the world
mapWorld <- map_data("world") %>%
        as_tibble()
```

Answer: We could use the variable "region" in mapWorld and the variable "country.name.en." in myeurope to join the two datasets. Both variables represent the World Bank code for each country, which is a unique identifier.

Question 10:

Use a joining function to check if any information from `myeurope` is not contained in `mapWorld`, matching the two datasets based on the country name.

```
# Check if any information from myeurope is not contained in mapWorld
anti_join(myeurope, mapWorld, by = c("country.name.en" = "region"))
```

```
## # A tibble: 4 x 8
##   country  year SP.URB.TOTL pop_urb_growth SP.POP.TOTL pop_gro~1 conti~2 count~3
##   <chr>   <dbl>       <dbl>          <dbl>       <dbl>     <dbl> <chr>   <chr>
## 1 BIH      2017     1646947         -0.433     3440027    -1.18  Europe  Bosnia~
## 2 CZE      2017     7805452          0.408    10594438     0.266 Europe  Czechia
## 3 GBR      2017    54923317          0.989    66058859     0.679 Europe  United~
## 4 GIB      2017       32602          0.114       32602     0.114 Europe  Gibral~
## # ... with abbreviated variable names 1: pop_growth, 2: continent,
## #   3: country.name.en
```

Some countries such as United Kingdom did not have a match. Why do you think this happened? *Hint: find the distinct country names in `mapWorld`, arrange them in alphabetical order, and scroll through the names. Can you find any of these countries with no match in a slightly different form?* If you need to print more output from a tibble, you can use `print(n = X)` where `X` is the number of lines to print out.

```
# Find the distinct country names in mapWorld
mapWorld %>%
  distinct(region) %>%
  arrange(region)
```

```
## # A tibble: 252 x 1
##    region
##    <chr>
##  1 Afghanistan
##  2 Albania
##  3 Algeria
##  4 American Samoa
##  5 Andorra
##  6 Angola
##  7 Anguilla
##  8 Antarctica
##  9 Antigua
## 10 Argentina
## # ... with 242 more rows
```

Answer: Some countries in mapWorld did not have a match in myeurope as there are different names for certain countries in mapWorld. For example, Bosnia and Herzegovina show up as "Bosnia and Herzegovina" in mapWorld but "Bosnia & Herzegovina" in myeurope.

Question 11:

Consider the `myeurope` dataset. Recode some of the country names so that the countries with no match from the previous question (with the exception of Gibraltar which is not technically a country anyway) will have a match.

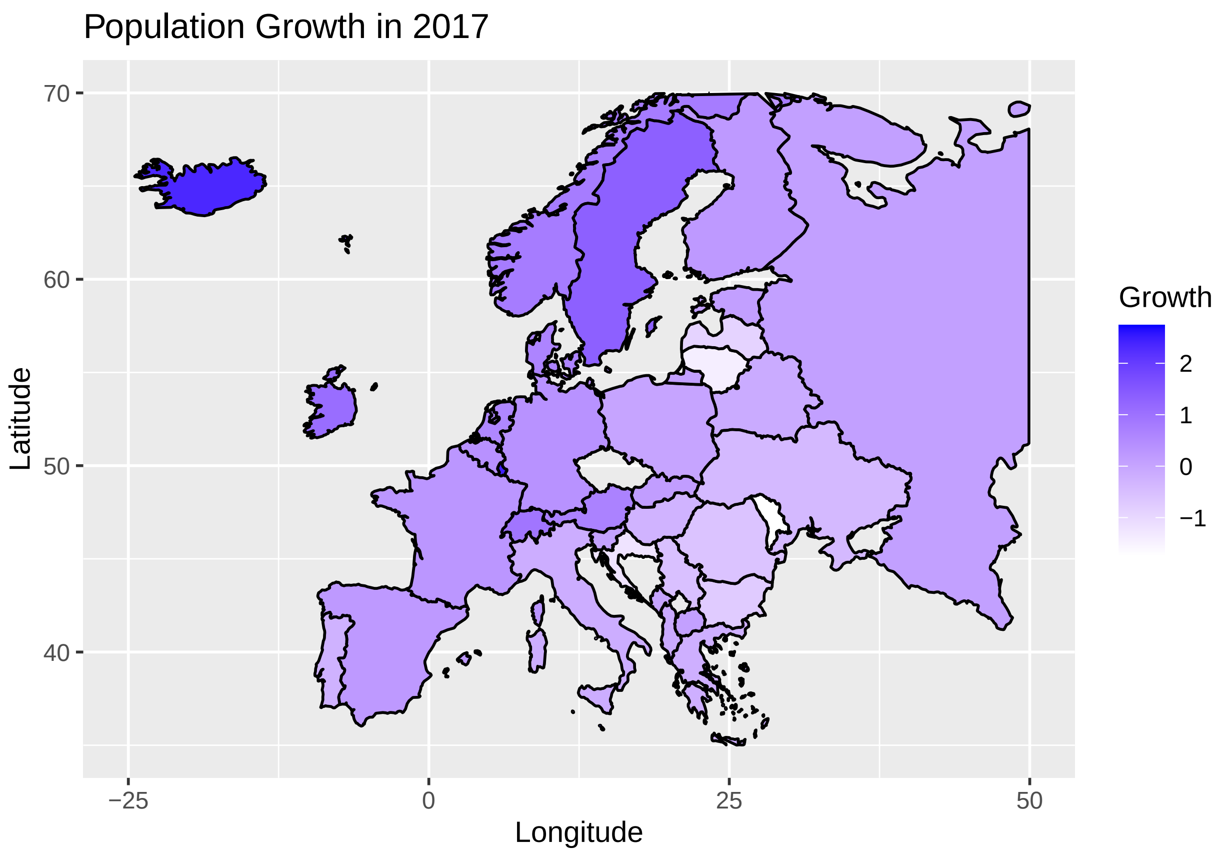
Hint: use `recode()` *inside* `mutate()` *as described in this article https://www.statology.org/recode-dplyr/.* Then add a pipe and use a `left_join()` function to add the geographic information in `mapWorld` to the countries in `myeurope`. Save this new dataset as `mymap`.

```r
# Recode country names to match with mapWorld
mymap <- myeurope %>%
  mutate(country = recode(country,
                          "Czech Republic" = "Czechia",
                          "Macedonia" = "North Macedonia",
                          "Bosnia & Herzegovina" = "Bosnia and Herzegovina",
                          "United Kingdom" = "UK")) %>%
  left_join(mapWorld, by = c("country.name.en" = "region"))
```

Question 12:

Let's visualize how population growth varies across European countries in 2017 with a map. Use the R code provided below. Add a comment after each `#` to explain what each component of this code does. *Note: it would be a good idea to run the code piece by piece to see what each layer adds to the plot.*

```r
# Build a map!
mymap %>%
  # Specify the mapping of data to aesthetics of the plot
  ggplot(aes(x = long, y = lat, group = group, fill = pop_growth)) +
  # Add a layer for polygons with a black border
  geom_polygon(colour = "black") +
  # Specify the color scale for fill colors
  scale_fill_gradient(low = "white", high = "blue") +
  # Add axis and legend labels
  labs(fill = "Growth" ,title = "Population Growth in 2017",
       x ="Longitude", y ="Latitude") +
  # Set the limits for the x and y axis
  xlim(-25,50) + ylim(35,70)
```



Which country had the highest population growth in Europe in 2017? *Hint: it's very tiny! You can refer to this map for European geography: https://www.wpmap.org/europe-map-hd-with-countries/*

Answer: Malta had the highest population growth in Europe in 2017.