HW 4

Enter your name and EID here: Austine Do (ahd589)

You will submit this homework assignment as a pdf file on Gradescope.

For all questions, include the R commands/functions that you used to find your answer (show R chunk). Answers without supporting code will not receive credit. Write full sentences to describe your findings.

Part 1

The dataset `world_bank_pop` is a built-in dataset in `tidyverse`. It contains information about the total population and population growth, overall and more specifically in urban areas, for countries around the world.

Question 1: (2 pts)

Why is the `world_bank_pop` dataset not tidy? What shall we do to make it tidy?

```
# Displaying the head of world_bank_pop
head(world_bank_pop)
```

```
## # A tibble: 6 x 20
##   country indicator       `2000`  `2001`  `2002`  `2003`  `2004`  `2005`  `2006`
##   <chr>   <chr>            <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 ABW     SP.URB.TOTL     4.16e4 4.20e+4 4.22e+4 4.23e+4 4.23e+4 4.24e+4 4.26e+4
## 2 ABW     SP.URB.GROW     1.66e0 9.56e-1 4.01e-1 1.97e-1 9.46e-2 1.94e-1 3.67e-1
## 3 ABW     SP.POP.TOTL     8.91e4 9.07e+4 9.18e+4 9.27e+4 9.35e+4 9.45e+4 9.56e+4
## 4 ABW     SP.POP.GROW     2.54e0 1.77e+0 1.19e+0 9.97e-1 9.01e-1 1.00e+0 1.18e+0
## 5 AFE     SP.URB.TOTL     1.16e8 1.20e+8 1.24e+8 1.29e+8 1.34e+8 1.39e+8 1.44e+8
## 6 AFE     SP.URB.GROW     3.60e0 3.66e+0 3.72e+0 3.71e+0 3.74e+0 3.81e+0 3.81e+0
## # i 11 more variables: `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>,
## #   `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>,
## #   `2016` <dbl>, `2017` <dbl>
```

The dataset is not tidy because the years that are currently columns should just be a single variable column called 'year' and the values for 'indicator' column should each have their own column.

Use one of the `pivot` functions on `world_bank_pop` to create a new dataset with the years 2000 to 2017 appearing as variable `year`, and the different values for the indicator variable are in a variable called `indicator_value`. Double check that the `year` variable appears as a **numeric** variable. Continue tidying `world_bank_pop` with another `pivot` function, with the different categories for the `indicator` variable appearing as their own variables. Is the data tidy now? It should be! Save the resulting dataset as `myworld`.

```r
# Get the column names of the dataset
world_bank_pop_col_names <- colnames(world_bank_pop)

# Pivot longer to get all the year columns into one column named 'year' and get the values as populatio
myworld <- world_bank_pop |>
                pivot_longer(cols = world_bank_pop_col_names[3:20],
                             names_to = 'year',
                             values_to = 'population') |>
                mutate_at('year', as.numeric)

# Pivot wider to get all the variables from 'indicator' column into their own column referencing the po
myworld <- myworld |>
                pivot_wider(names_from = indicator,
                            values_from = population)
myworld
```

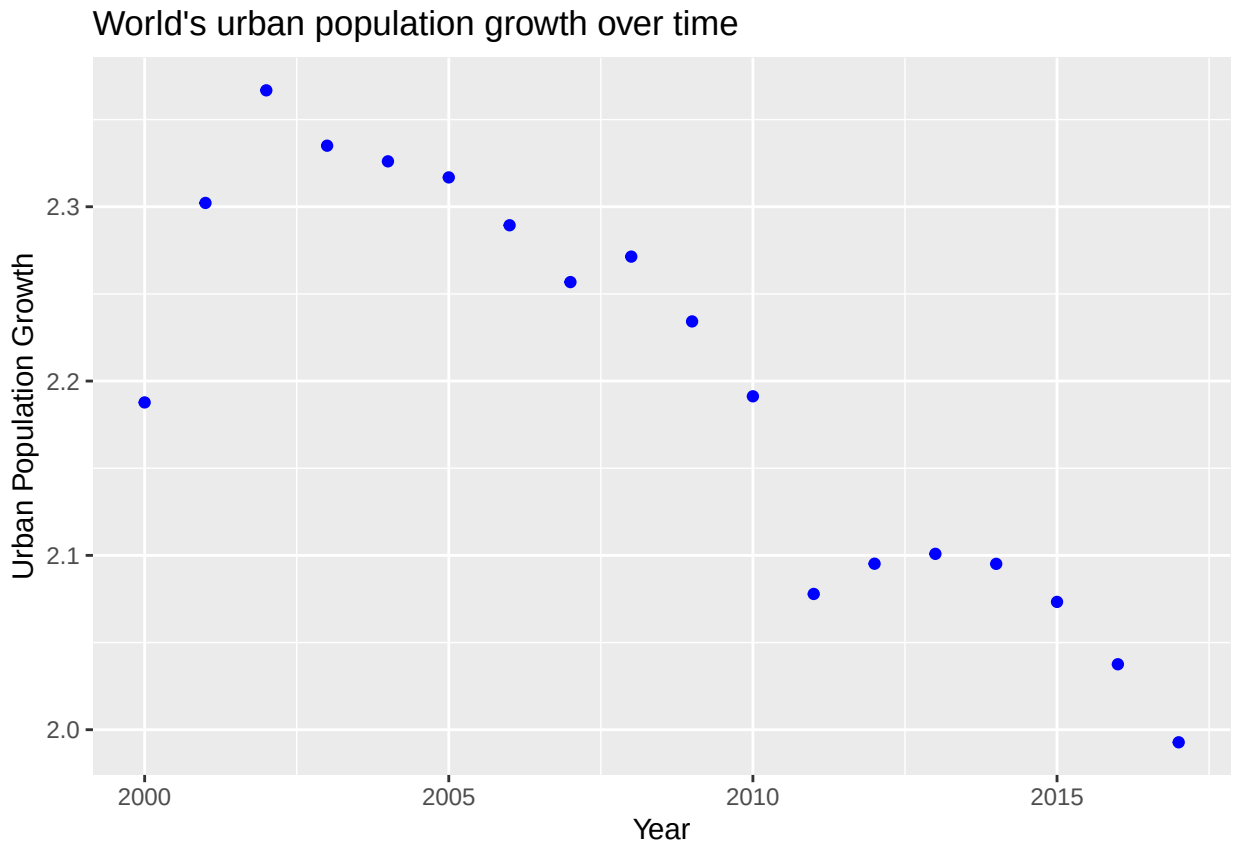
```
## # A tibble: 4,788 x 6
##    country  year SP.URB.TOTL SP.URB.GROW SP.POP.TOTL SP.POP.GROW
##    <chr>   <dbl>       <dbl>       <dbl>       <dbl>       <dbl>
##  1 ABW      2000       41625      1.66         89101       2.54
##  2 ABW      2001       42025      0.956        90691       1.77
##  3 ABW      2002       42194      0.401        91781       1.19
##  4 ABW      2003       42277      0.197        92701       0.997
##  5 ABW      2004       42317      0.0946       93540       0.901
##  6 ABW      2005       42399      0.194        94483       1.00
##  7 ABW      2006       42555      0.367        95606       1.18
##  8 ABW      2007       42729      0.408        96787       1.23
##  9 ABW      2008       42906      0.413        97996       1.24
## 10 ABW      2009       43079      0.402        99212       1.23
## # i 4,778 more rows
```

The dataset is now much more tidy and makes more sense when looking at the format of the dataset.

Question 2: (2 pts)

Create a `ggplot` to display how the world's urban population growth has changed over the years. *Note: the country code `WLD` represents the entire world.* Why does this graph not contradict the fact that the urban population worldwide is increasing over the years?

```r
# Scatter plot of the world's urban population growth
myworld |>
    filter(country == 'WLD') |>
    ggplot() +
    geom_point(aes(x = year, y = SP.URB.GROW), color = 'blue') +
    labs(title = 'World\'s urban population growth over time',
         x = 'Year',
         y = 'Urban Population Growth')
```



This graph doesn't contradict the fact the world's urban population has increased over time because this doesn't display any values that are negative or zero values which would indicate a stagnation or decline in population growth.

Which country code in `myworld` had the highest population growth in 2017?

```
# Find the country with the highest population growth in 2017
myworld |>
    filter(year == 2017) |>
    slice_max(n = 10, SP.POP.GROW)
```

```
## # A tibble: 10 x 6
##    country  year SP.URB.TOTL SP.URB.GROW SP.POP.TOTL SP.POP.GROW
##    <chr>   <dbl>       <dbl>       <dbl>       <dbl>       <dbl>
##  1 QAT      2017     2686753        4.46     2711755        4.39
##  2 TCA      2017       36982        4.42       39844        4.09
##  3 MDV      2017      186048        5.01      472442        3.93
##  4 SOM      2017     6598376        5.22    14864221        3.92
##  5 NER      2017     3554150        4.19    21737922        3.83
##  6 GNQ      2017     1039364        4.35     1450694        3.63
##  7 AGO      2017    19586972        4.62    30208628        3.55
##  8 UGA      2017     9307879        5.99    40127085        3.50
##  9 COD      2017    36983500        4.76    84283273        3.44
## 10 TZA      2017    18597942        5.57    56267032        3.37
```

The country with the highest population growth in 2017 was the country with the country code QAT.

Question 3: (2 pts)

When answering the previous question, we can only report the three-letter code and (probably) have no idea what the actual country is. Let's use the package `countrycode` to join some relevant information such as the country name:

```
# Install the package (only needed once)
install.packages("countrycode")
```

This package contains a built-in dataset called `codelist` that has information about the coding system used by the World bank (and many other coding systems):

```
# Call the countrycode package
library(countrycode)

# Take a look at the dataset
head(codelist)
```

Create a list of country codes 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 and missing `continent`. Save the resulting dataset as `mycodes`.

```
# Selects only 3 columns and filters out rows with NA/missing values for 'wb' and 'continent' column
mycodes <- codelist |>
                select(continent, wb, country.name.en) |>
                filter(!is.na(wb) & !is.na(continent))
```

How many distinct country codes are there in `mycodes`?

```
# Finds the unique number of country codes in the dataset 'mycodes'
length(unique(mycodes$wb))
```

```
## [1] 216
```

There are 216 distinct country codes in the 'mycodes' dataset.

Question 4: (2 pts)

Is there the same number of distinct country codes in `myworld` than there were in `mycodes`? Why or why not?

```
# Finds the number of distinct country codes in the `myworld`
length(unique(myworld$country))
```

```
## [1] 266
```

There are more country codes in 'myworld' than in `mycodes` this is likely because we removed rows with missing country code and continent values from the `mycodes`.

Use the `inner_join()` function to add the information of the country names **to** `myworld` dataset, matching the two datasets based on the World Bank code. Save the resulting dataset as `mycountries`.

```
# Using inner_join() to join 'mycodes' to 'myworld'
mycountries <- inner_join(myworld, mycodes, by = c('country' = 'wb'))
```

Now, which country code in `mycountries` had the highest population growth in 2017?

```
# Displays the top 10 countries with the highest population growth in 2017
mycountries |>
    filter(year == 2017) |>
    slice_max(n = 10, SP.POP.GROW)
```

```
## # A tibble: 10 x 8
##    country  year SP.URB.TOTL SP.URB.GROW SP.POP.TOTL SP.POP.GROW continent
##    <chr>   <dbl>       <dbl>       <dbl>       <dbl>       <dbl> <chr>
##  1 QAT      2017     2686753        4.46     2711755        4.39 Asia
##  2 TCA      2017       36982        4.42       39844        4.09 Americas
##  3 MDV      2017      186048        5.01      472442        3.93 Asia
##  4 SOM      2017     6598376        5.22    14864221        3.92 Africa
##  5 NER      2017     3554150        4.19    21737922        3.83 Africa
##  6 GNQ      2017     1039364        4.35     1450694        3.63 Africa
##  7 AGO      2017    19586972        4.62    30208628        3.55 Africa
##  8 UGA      2017     9307879        5.99    40127085        3.50 Africa
##  9 COD      2017    36983500        4.76    84283273        3.44 Africa
## 10 TZA      2017    18597942        5.57    56267032        3.37 Africa
## # i 1 more variable: country.name.en <chr>
```

QAT had the highest population growth in 2017.

Question 5: (2 pts)

Compare the average urban population growth per `continent` over the years using `mycountries`. Which continent had constantly the highest average urban population growth over the years? the lowest?

```
# Grouping by continent and year and finding the average urban population growth
mycountries |>
    group_by(continent, year) |>
    summarize(avg.urb.pop.grow = sum(SP.URB.GROW) / n()) |>
    arrange(desc(avg.urb.pop.grow))
```

```
## # A tibble: 90 x 3
## # Groups:   continent [5]
##    continent  year avg.urb.pop.grow
##    <chr>     <dbl>            <dbl>
##  1 Africa     2008             3.82
##  2 Africa     2002             3.80
```

```
##  3 Africa     2010            3.79
##  4 Africa     2009            3.76
##  5 Africa     2001            3.74
##  6 Africa     2006            3.69
##  7 Africa     2005            3.66
##  8 Africa     2000            3.63
##  9 Africa     2003            3.62
## 10 Africa     2004            3.61
## # i 80 more rows
```

The continent with the highest average urban population growth over the years is Africa while the continent with the lowest average urban population growth is Europe. It is worth noting that Americas had no value for average urban population growth for any year so the previous conclusion is drawn purely from the calculated data that did not have missing values.

Let's focus on countries in Africa for the year of 2017 from now on. Save the resulting dataset as `myafrica2017`.

```
# Subset `mycountries` to only contain countries in Africa for 2017
myafrica2017 <- mycountries |>
                    filter(continent == 'Africa' & year == 2017)
```

Question 6: (2 pts)

When dealing with spatial data, we can actually visualize information on a map if we have geographic information such as latitude and longitude. Let's use a function called `map_data()` to get geographic coordinates about countries in the world from the `maps` package:

```
# Install package (only needed once)
install.packages("maps")
```

Take a look at the built-in dataset `mapWorld`:

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

# Take a quick look
head(mapWorld)
```

```
##        long      lat group order region subregion
## 1 -69.89912 12.45200     1     1  Aruba      <NA>
## 2 -69.89571 12.42300     1     2  Aruba      <NA>
## 3 -69.94219 12.43853     1     3  Aruba      <NA>
## 4 -70.00415 12.50049     1     4  Aruba      <NA>
## 5 -70.06612 12.54697     1     5  Aruba      <NA>
## 6 -70.05088 12.59707     1     6  Aruba      <NA>
```

Inner join `mapWorld` with `myafrica2017`. What variable in each dataset should we use to join? *Note: the variables do not have the same name for each dataset but they contain the same information.* Save the resulting dataset as `mymap`.

We should use 'region' from `mapWorld` and 'country.name.en' from `myafrica2017`

```r
# Joins `mapWorld` and `myafrica2017` togoether
mymap <- inner_join(myafrica2017, mapWorld, by = c('country.name.en' = 'region'))
```

Question 7: (2 pts)

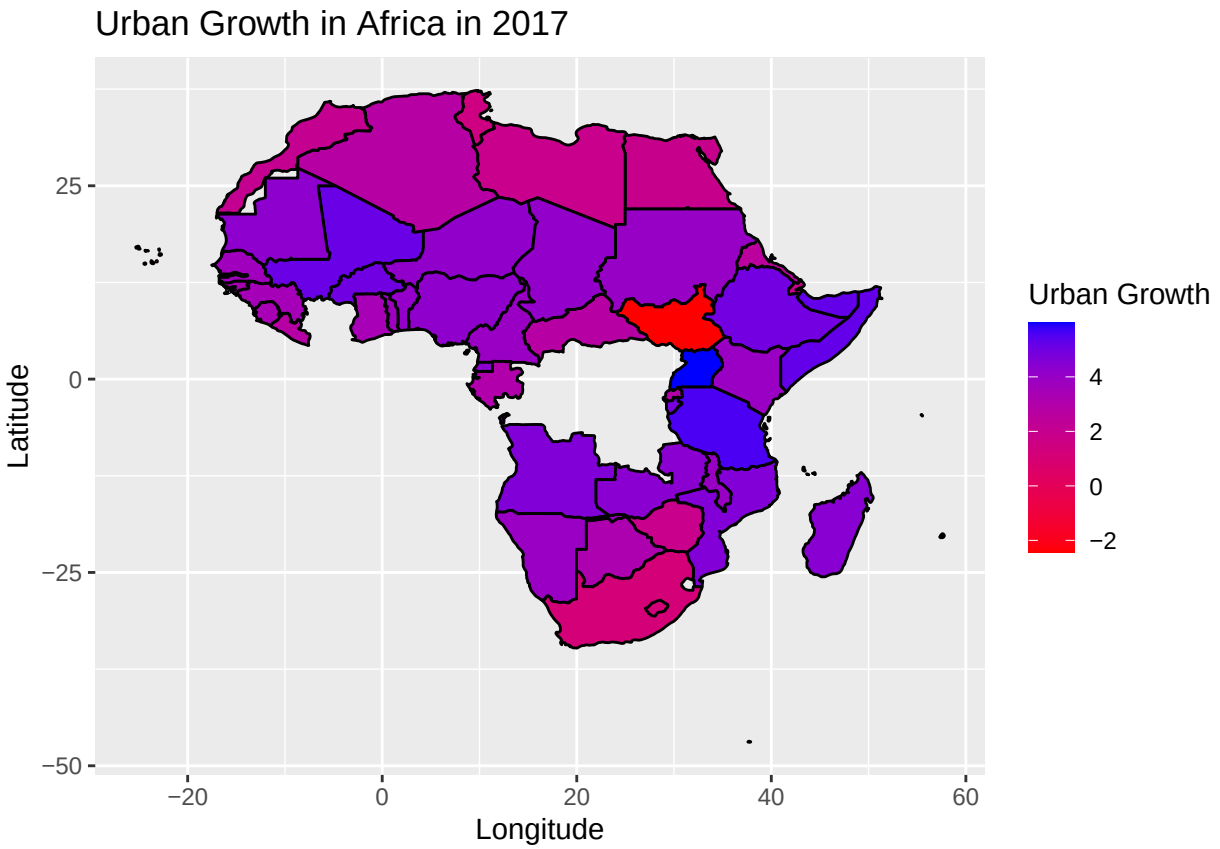
Let's visualize how urban population growth varied across African countries in 2017 using the `ggmap` package:

```r
# Install package (only needed once)
install.packages("ggmap")
```

Use the R code provided below to make a map. 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.* Once your code works, get rid of the option `eval=FALSE` so it will actually run this code chunk when knitting!

```r
# Upload the ggmap package
library(ggmap)

# Build a map!
mymap |>
  # creates the plot where x is longitude, y is latitude, group is the country group,
  # and fills the country that is drawn with its corresponding SP.URB.GROW value
  ggplot(aes(x = long, y = lat, group = group, fill = SP.URB.GROW)) +
  # sets the outline of the shape of the country to be black
  geom_polygon(colour = "black") +
  # changes the color gradient for the range of values found in SP.URB.GROW column
  scale_fill_gradient(low = "red", high = "blue") +
  # Makes the labels for the plot
  labs(fill = "Urban Growth",
       title = "Urban Growth in Africa in 2017",
       x ="Longitude",
       y ="Latitude")
```



Comment on the distribution of urban population growth across African countries in 2017:

We can see from the map that the countries with the highest urban growth are generally in the more tropical and temperate climate regions of Africa whereas the countries with the least urban population growth are on the most Northern and Southern parts of Africa which is where the more extreme climates and regions are like the Deserts of North Africa.

Question 8: (1 pt)

Did you notice that there was some missing data for some of these countries? Check if any information from `myafrica2017` was not contained in `mapWorld`, meaning that there might not be a match for a country in `mapWorld` for some African countries in `myafrica2017` and only display the names of countries for which it might be the case:

```
# Finds and displays the countries in `myafrica2017` that were not contained in `mapworld`
anti_join(myafrica2017, mapWorld, by = c('country.name.en' = 'region'))
```

```
## # A tibble: 5 x 8
##   country  year SP.URB.TOTL SP.URB.GROW SP.POP.TOTL SP.POP.GROW continent
##   <chr>   <dbl>       <dbl>       <dbl>       <dbl>       <dbl> <chr>
## 1 CIV      2017    12505013        3.47    24848016        2.59  Africa
## 2 COD      2017    36983500        4.76    84283273        3.44  Africa
## 3 COG      2017     3530528        3.08     5312340        2.39  Africa
```

```
## 4 STP      2017      149719        2.87      208036       1.65  Africa
## 5 SWZ      2017      272016        1.48     1151390       0.773 Africa
## # i 1 more variable: country.name.en <chr>
```

You should find that some countries did not have a match. Why do you think this happened? *Note: This question can be challenging! You will have to do some research about each of these countries: this is pretty typical for a data scientist though! We need to get more knowledge about the context to make sense of the data.*

I believe this happened because some of these countries might have been aggregated under a larger, more well-known territory, might have emerged in recent years, or have had a name change that `mapWorld` might have not contained or been updated to contain.

Using the `str_detect()` function, find the distinct country names in `mapWorld` that maybe be potential matches for countries in `myafrica2017`:

```
# Using string functions to potentially find matches between the missing countries
# and the countries that exist in `mapWorld`

missing_country_names <- 'Ivory|Coast|Congo|Kinshasa|Brazzaville|Sao|Tome|Principe|Eswatini|Swaziland'
unique(mapWorld[str_detect(mapWorld$region, missing_country_names), 'region'])
```

```
## [1] "Ivory Coast"                       "Democratic Republic of the Congo"
## [3] "Republic of Congo"                 "Sao Tome and Principe"
## [5] "Swaziland"
```

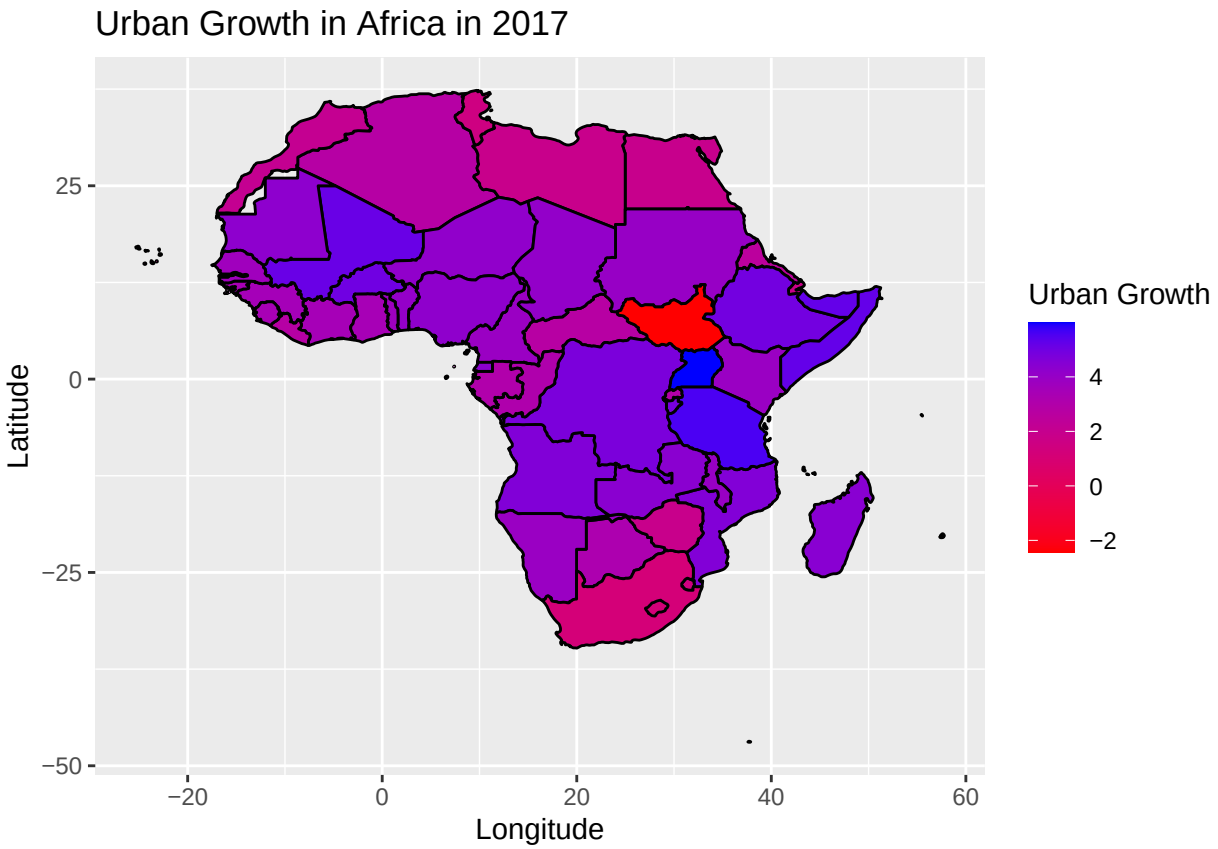
Recode the country names in `myafrica2017` so that the 5 countries with no previous match will now have a match. *Hint: use `recode()` inside `mutate()` as described in our WS10 or in this article https://www.statology.org/recode-dplyr/.* Then add a pipe and joining function to add the geographic information in `mapWorld` to the countries in `myafrica2017`. Add another pipe and update the map from the previous question!

```
# Recode and update the `myafrica2017` dataset with the matched country names
# and add the country data from `mapWorld` to `myafrica2017`

recode_country_names <- c('Côte d'Ivoire' = 'Ivory Coast',
                          'Congo - Kinshasa' = 'Democratic Republic of the Congo',
                          'Congo - Brazzaville' = 'Republic of Congo',
                          'São Tomé & Príncipe' = 'Sao Tome and Principe',
                          'Eswatini' = 'Swaziland')

myafrica2017 <- myafrica2017 |>
                    mutate(country.name.en.updated = recode(country.name.en, !!!recode_country_names)) |>
                    inner_join(mapWorld, by = c('country.name.en.updated' = 'region'))

# Using the updated `myafrica2017` dataset to update the map plot
myafrica2017 |> ggplot(aes(x = long, y = lat, group = group, fill = SP.URB.GROW)) +
                      geom_polygon(colour = "black") +
                      scale_fill_gradient(low = "red", high = "blue") +
                      labs(fill = "Urban Growth",
                           title = "Urban Growth in Africa in 2017",
                           x ="Longitude",
                           y ="Latitude")
```



Part 2

Recall the context about the Internet clothing retailer Stitch Fix wanting to develop a new model for selling clothes to people online (see HW 1 and HW2). Their basic approach is to send people a box of 5–6 items of clothing and allow them to try the clothes on. Customers keep (and pay for) what they like while mailing back the remaining clothes. Stitch Fix then sends customers a new box of clothes a month later.

You built an intake survey distributed to customers when they first sign up for the service. You are now analyzing the results of this survey to choose some variables for predicting what types of clothes each customer would be more likely to keep.

Question 9: (2 pts)

When analyzing the results of the survey, you noticed that some customers left their demographic information (for example: age, location, ...) blank. Why did that occur? What could be some potential issues on the analysis?

A couple of reasons why the customers may have left their demographic information blank is for privacy reason or perhaps the survey design was confusing or not inclusive enough for the customer. One potential issue that may result is a smaller sample size of data since the amount of complete/accurate data for analysis will be reduced and another issue would be limitations on the generalization of the analysis since the analysis may be less representative of the customer population.

Question 10: (2 pts)

When analyzing the hip size (in cm), waist size (in cm), and the size for skirts, you noticed that a customer reported a hip size of 38, waist size of 28, and a size of L. What could be some potential issues related to these values and what could you do about it?

Issues that may be related to the inputted values are ensuring that both numerical measures (hip size and waist size) are in the same and correct unit, inaccurate data that may not represent the customer's actual measurements (which can happen if they estimated or guessed), and inconsistency within the categorical variable 'size' as S, M, L, etc. may differ across brands, countries, and regions. There are many ways we can address these issues. One thing we can do is use data validation techniques (such as checking for outliers, patterns of error, and unlikely values) and another thing we can do is convert the 'size' variable to a numerical variable that is based on a specific sizing chart to better standardize the analysis.

Formatting: (1 pt)

Knit your file! You can knit into html and once it knits in html, click on `Open in Browser` at the top left of the window that pops out. **Print** your html file into pdf from your browser.

Is it working? If not, try to decipher the error message: look up the error message, consult websites such as stackoverflow or crossvalidated.

Finally, remember to select pages for each question when submitting your pdf to Gradescope.