Module 3 Day 2 ROUGH-ROUGH DRAFT

```r
library(tidyverse)
library(lubridate)

library(stringr)

library(plotly)
library(gapminder)
library(maps)
library(animation)
library(scales)
library(stargazer)
```

Goals

Chat about Developmental Economics

Use data to learn some stylized facts about developing nations

Growth vs. Development

Use fixed effects

Use maps to visualize data

Create gifs to visualize dynamics in maps

```r
world_data <- gapminder
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333       779
## 2 Afghanistan Asia       1957    30.3  9240934       821
## 3 Afghanistan Asia       1962    32.0 10267083       853
## 4 Afghanistan Asia       1967    34.0 11537966       836
## 5 Afghanistan Asia       1972    36.1 13079460       740
## 6 Afghanistan Asia       1977    38.4 14880372       786
```

year - every 5 years

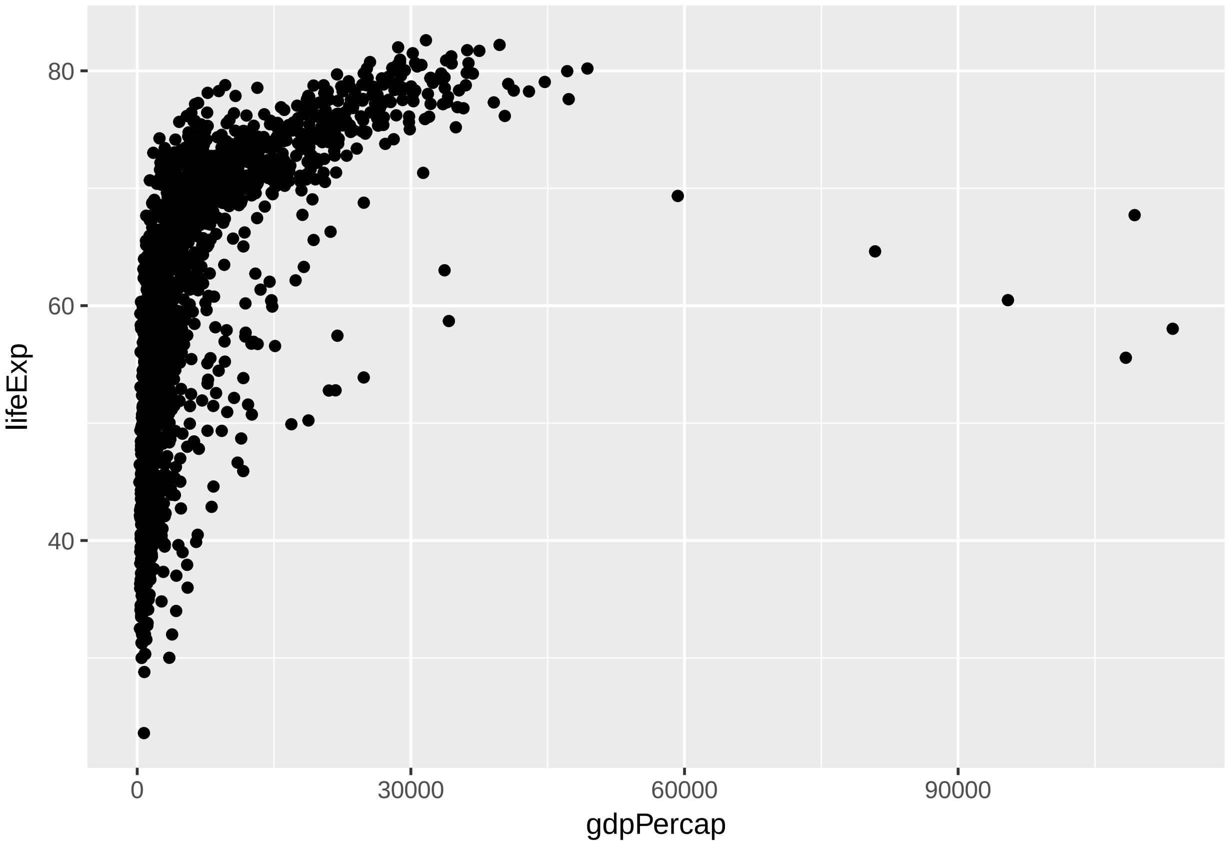
lifeExp - life expectancy

pop - population

gdpPercap - gdp percapita

Investigate the relationship between life expectancy and GDP percapita

```r
world_data %>%
  ggplot(aes(gdpPercap, lifeExp)) +
  geom_point()
```

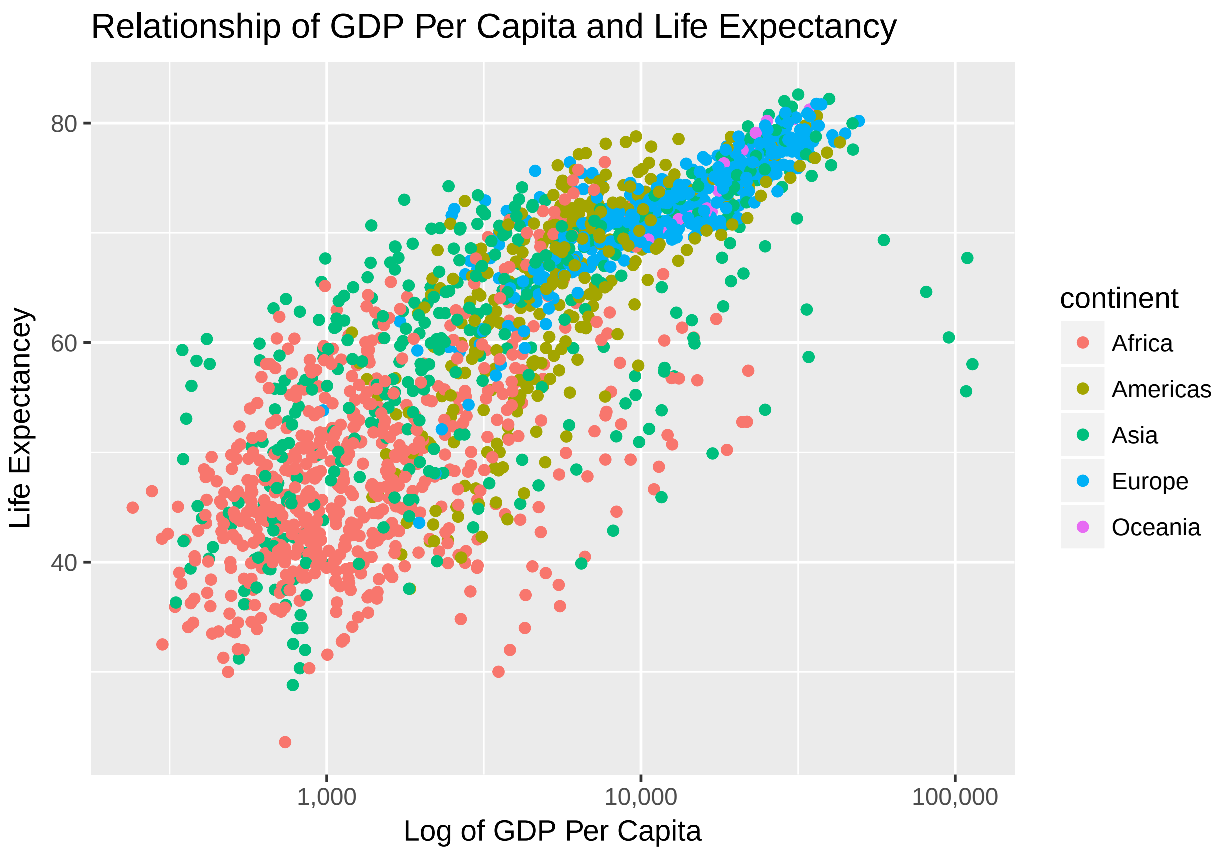


Class are now plotting pros!

In class exercise: Fix up the above plot and differentiate the observations by contitinet. I would suggest to use scale__x__log10(), if you do not know what this function does please google it or type ?scale__x__log10() into you counsle

Here is our attempt at a better plot!

```
world_data %>%
  ggplot(aes(gdpPercap, lifeExp, color = continent)) +
  geom_point() +
  scale_x_log10(labels = comma) +
  xlab("Log of GDP Per Capita") +
  ylab("Life Expectancey") +
  ggtitle("Relationship of GDP Per Capita and Life Expectancy")
```



Qucik digression about on about measurements of development and growth/wealth

Difference between economic growth and economic delevopment

How do we measure growth? Growth of GDP

How do we measure development? Education Attainment, Life Expectancy, Acess to Utilities, Stability of Infustructure, UN Development Index

Intertangled relationship between growth and development.

How much of economic growth explains development?

To anwser, or an idea of an anwser, we use regression analysis!

Here is our benchmark model

```
benchmark <- lm(lifeExp ~ gdpPercap, data = world_data)
summary(benchmark)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap, data = world_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.754  -7.758   2.176   8.225  18.426
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.396e+01  3.150e-01  171.29   <2e-16 ***
## gdpPercap   7.649e-04  2.579e-05   29.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.49 on 1702 degrees of freedom
## Multiple R-squared:  0.3407, Adjusted R-squared:  0.3403
## F-statistic: 879.6 on 1 and 1702 DF,  p-value: < 2.2e-16
```

Interpret results of benchmark model.

Evaluate the performance fo this model.

What tools did you just use? fit measures, significance of the coefficents, standard errors?

What about missing variables? Are adding variables always “better”?

Are there any missing factors that could effect Life Expectancy and are correlated with GDP per capita?

Our data has its limits, but can we do anything to imporve the estimates? Fixed Effects?

```r
better_reg <- lm(lifeExp ~ gdpPercap + factor(year), data = world_data)
summary(better_reg)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap + factor(year), data = world_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.880  -6.915   0.994   7.606  21.052
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.655e+01  8.161e-01  57.041  < 2e-16 ***
## gdpPercap        6.721e-04  2.442e-05  27.521  < 2e-16 ***
## factor(year)1957 2.064e+00  1.147e+00   1.799 0.072156 .
## factor(year)1962 3.879e+00  1.147e+00   3.381 0.000738 ***
## factor(year)1967 5.439e+00  1.148e+00   4.738 2.33e-06 ***
## factor(year)1972 6.543e+00  1.149e+00   5.693 1.47e-08 ***
## factor(year)1977 8.101e+00  1.150e+00   7.042 2.74e-12 ***
## factor(year)1982 9.926e+00  1.151e+00   8.626  < 2e-16 ***
## factor(year)1987 1.135e+01  1.152e+00   9.855  < 2e-16 ***
## factor(year)1992 1.212e+01  1.152e+00  10.523  < 2e-16 ***
## factor(year)1997 1.235e+01  1.154e+00  10.699  < 2e-16 ***
## factor(year)2002 1.248e+01  1.157e+00  10.783  < 2e-16 ***
## factor(year)2007 1.260e+01  1.163e+00  10.834  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.665 on 1691 degrees of freedom
## Multiple R-squared:  0.4441, Adjusted R-squared:  0.4402
## F-statistic: 112.6 on 12 and 1691 DF,  p-value: < 2.2e-16
```

Class Exercise: Run the baseline model with just time FEs, just continent FEs, and both. Put the baseline model and all three of the regressions, so a total of four regressions, into a stargazer table! Please exclude the coefficents reported on the fixed effects, but indicate what regression has which fixed effects.

```
benchmark <- lm(lifeExp ~ gdpPercap, data = world_data)
time_fe <- lm(lifeExp ~ gdpPercap + factor(year), data = world_data)
country_fe <- lm(lifeExp ~ gdpPercap + factor(continent), data = world_data)
both_fe <- lm(lifeExp ~ gdpPercap + factor(year) + factor(continent), data = world_data)


stargazer(benchmark, time_fe, country_fe, both_fe,
          type = "text",
          covariate.labels = c("GDP Per Capita"),
          omit = c("factor"),
          add.lines = list(c("Time Fixed Effects?", "No", "Yes", "No", "Yes"),
                           c("Continet Fixed Effects?", "No", "No", "Yes", "Yes"))
          )
```

```
##
## ====================================================================================================
##                                                                    Dependent variable:
##                         ----------------------------------------------------------------------------
##                                                                          lifeExp
##                                     (1)                       (2)                        (3)
## ----------------------------------------------------------------------------------------------------
## GDP Per Capita                   0.001***                   0.001***                  0.0004***
##                                  (0.00003)                (0.00002)                  (0.00002)
##
## Constant                         53.956***                46.554***                  47.889***
##                                   (0.315)                   (0.816)                    (0.340)
##
## ----------------------------------------------------------------------------------------------------
## Time Fixed Effects?                 No                        Yes                        No
## Continet Fixed Effects?             No                         No                        Yes
## Observations                       1,704                     1,704                      1,704
## R2                                 0.341                     0.444                      0.579
## Adjusted R2                        0.340                     0.440                      0.578
## Residual Std. Error        10.491 (df = 1702)         9.665 (df = 1691)          8.390 (df = 1698)
## F Statistic             879.577*** (df = 1; 1702) 112.578*** (df = 12; 1691) 467.712*** (df = 5; 169
## ====================================================================================================
## Note:
```

Why did we not suggest to do country and time fixed effects?

Think about the strucutre of this data?

How many observation for each country per year?

```
country_time_fe  <- lm(lifeExp ~ gdpPercap + factor(year) + factor(country), data = world_data)

stargazer(country_time_fe,
          type = "text",
          covariate.labels = c("GDP Per Capita"),
          omit = c("factor")
          )
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                lifeExp
## -----------------------------------------------
## GDP Per Capita              -0.0001***
##                              (0.00002)
##
## Constant                     26.852***
##                               (1.031)
##
## -----------------------------------------------
## Observations                   1,704
## R2                             0.936
## Adjusted R2                    0.929
## Residual Std. Error      3.438 (df = 1550)
## F Statistic         147.023*** (df = 153; 1550)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

Looks amazing!!!

Wait, there is a sign flip?

Due to overfitting? If we have dummies for every observation the coefficent infront of GDP per capita become meaningless

Now that we reviewed our plotting and regression skills lets learn something new!

Vizualizing data in maps is a powerful tool

An easy way to show "clustering" - like things are typically next to each other

Creating our first map - let's be ambitous! Lets map the world, but first let's look at the mapping data

```
world <- map_data("world")
head(world)
```

```
##        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>
```

Think of the data as a bunch of points where R is smart enough to just draw lines through the points

Ordering matters in this type of geospatial data - so don't go too crazy on it!!!

There are many types of way to store geographic data, and the type of data we are working with is the easiest.

Just be careful if you are wanting to do maps in the future, most of the time you will be given shape files which are its own special thing.

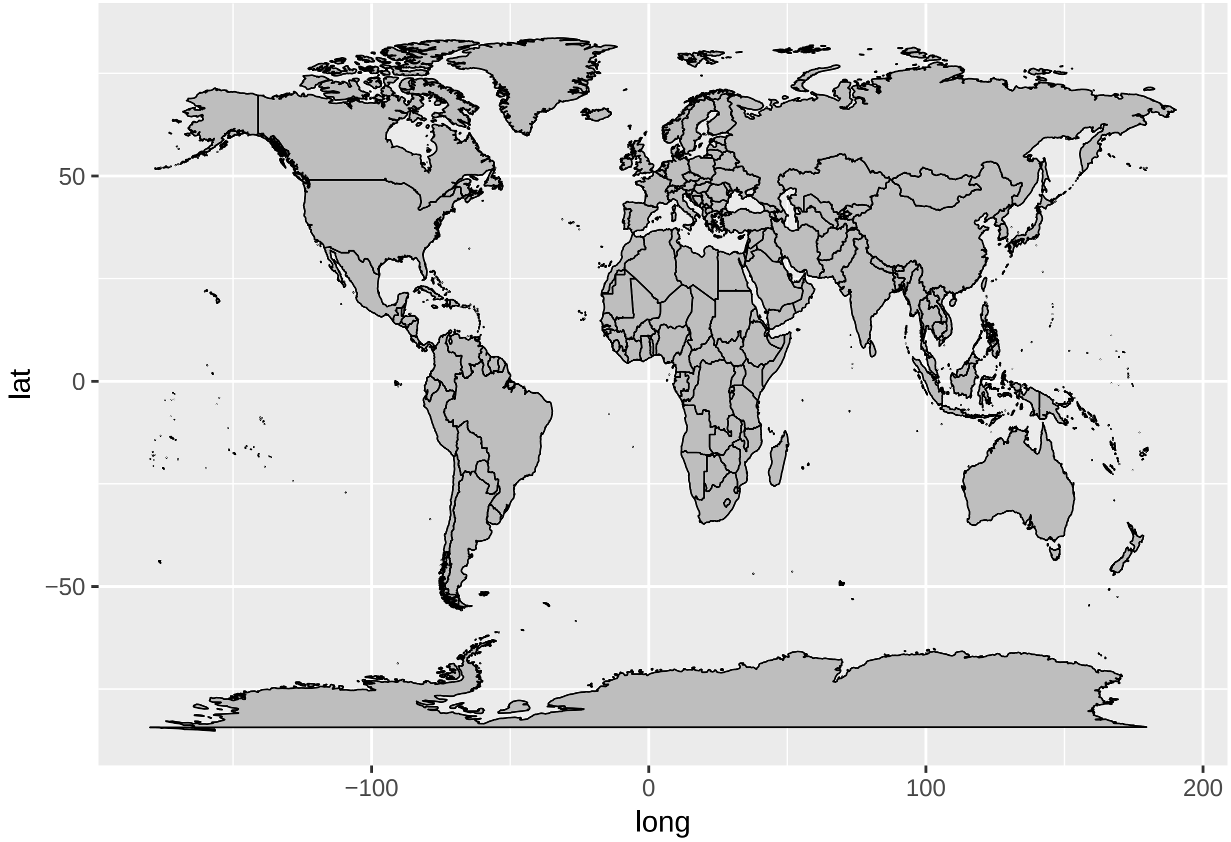
Mapping, in this lecture, works the exact same as a normal ggplot

There is a new "layer" called polygon

Note that the x variable is longitude and the y variable is lattitude

It common for people to say "latt, long" instead of "long, latt", either way to say it is fine, but when working with geographic data 90% of the time your x variable will be long and your y variable will be latt

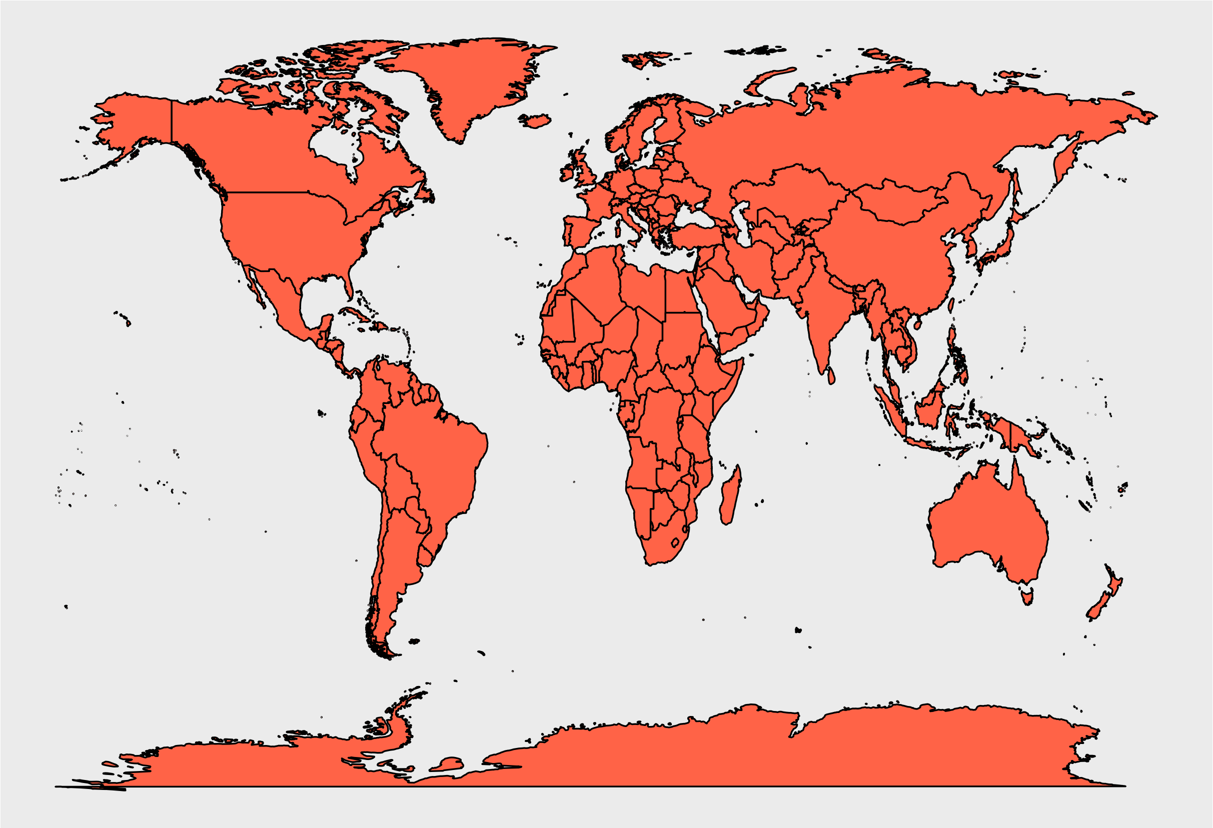
```
world %>%
    ggplot(aes(x = long, y = lat, group = group)) +
    geom_polygon(fill = "gray", color = "black", size = 0.3)
```



Let's get rid of the of the axes, lines, and change the countries to be the color "tomato"

```
no_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank())

world %>%
    ggplot(aes(x = long, y = lat, group = group)) +
    geom_polygon(fill = "tomato", color = "black", size = 0.3) +
    no_axes
```



Now lets merge our world_data with our maps to be able to plot maps that have data

Right now why don't we just look at Africa?

```
africa <- world_data %>%
  filter(continent == "Africa") %>%
  inner_join(world, by = c("country" = "region"))
```

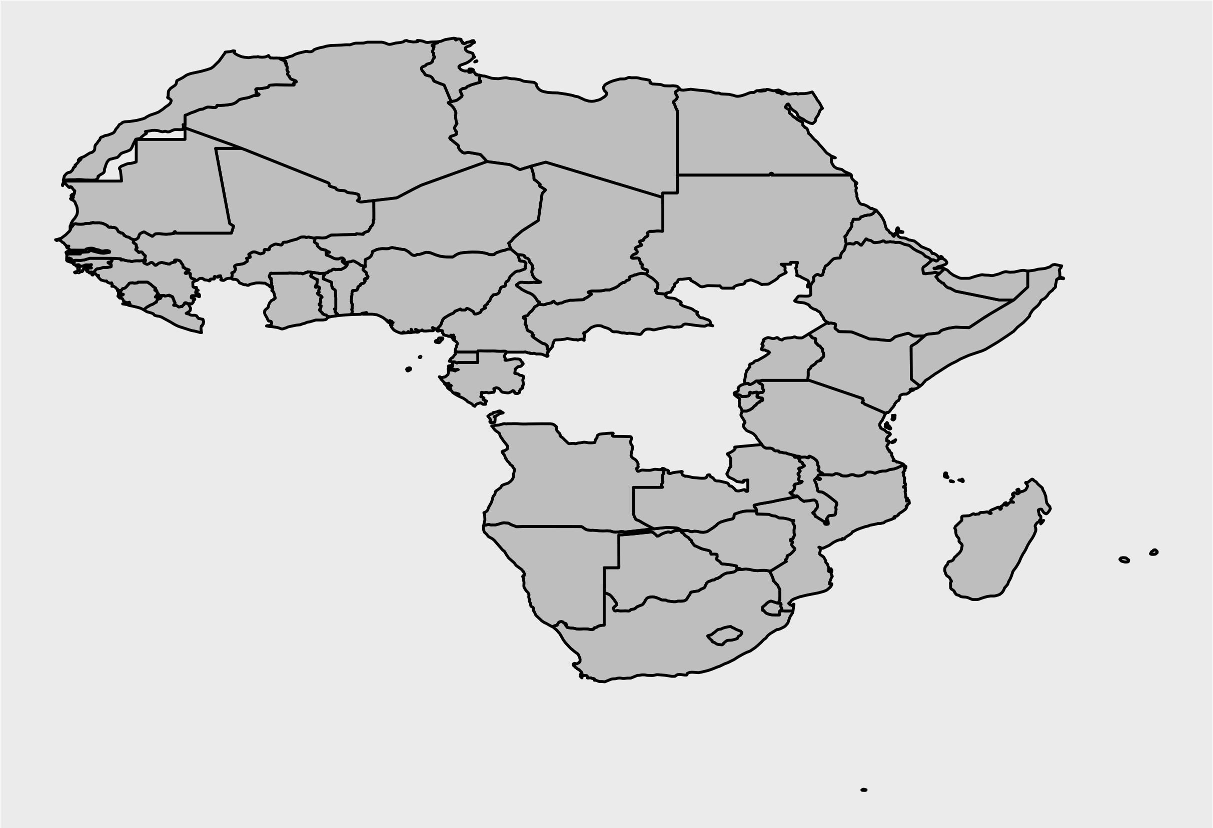
```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
## Warning: Column `country`/`region` joining factor and character vector,
## coercing into character vector
```

```
#need to make country a character
```

Now let's plot Africa, for the year 2007!

```
africa %>%
  filter(year == 2007) %>%
  ggplot() +
    geom_polygon(aes(long, lat, group = group), fill = "grey", color = "black") +
    no_axes
```



What is wrong? Why are there holes?

How many countries are in Africa?

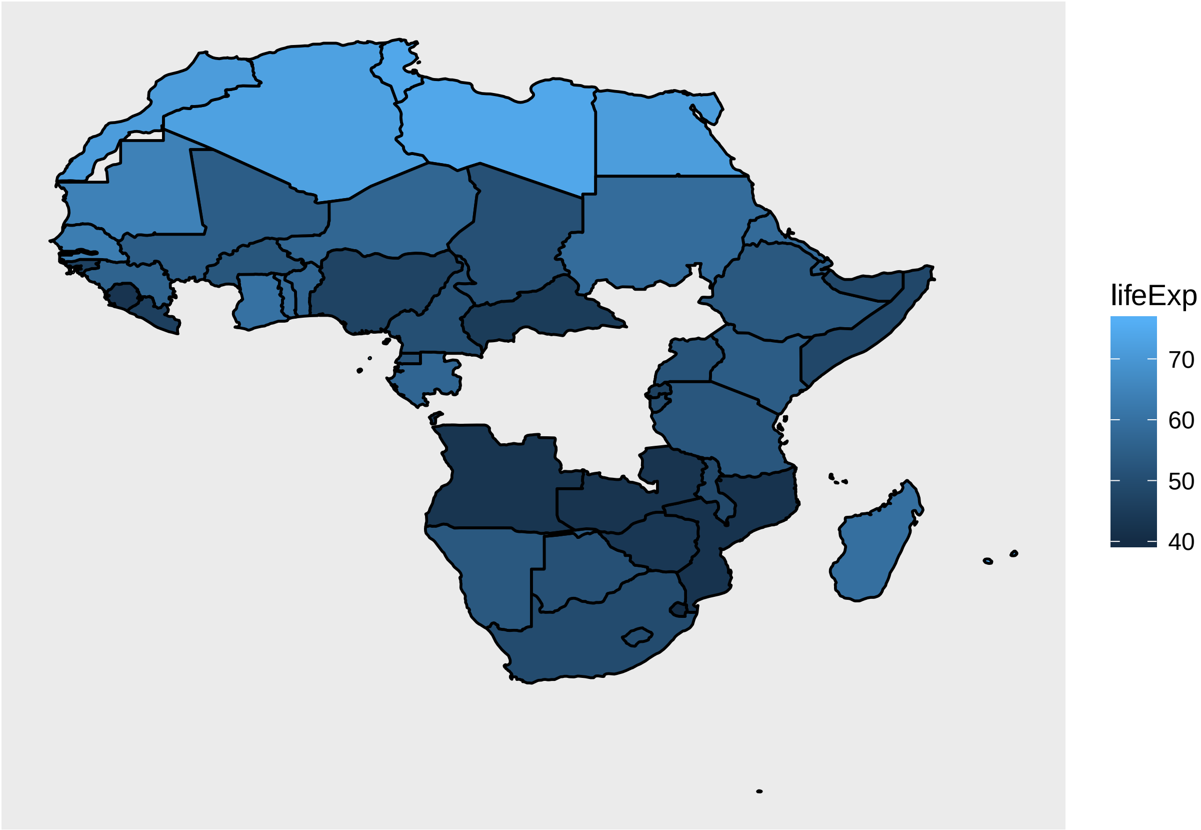
Did some get dropped in the merge?

Due to the data on GDP and Life Expentancy only covering countries that have been in exsitance since 1952, there is disagreement between the data sets on what should be the name of the countires. Thus durning the merge these countries were dropped

This is fine for now!

Now let us plot a heat map of the Life Expectancy of the countries in Africa durning the year of 2007! The brighter colors indicate a higher life expectancy

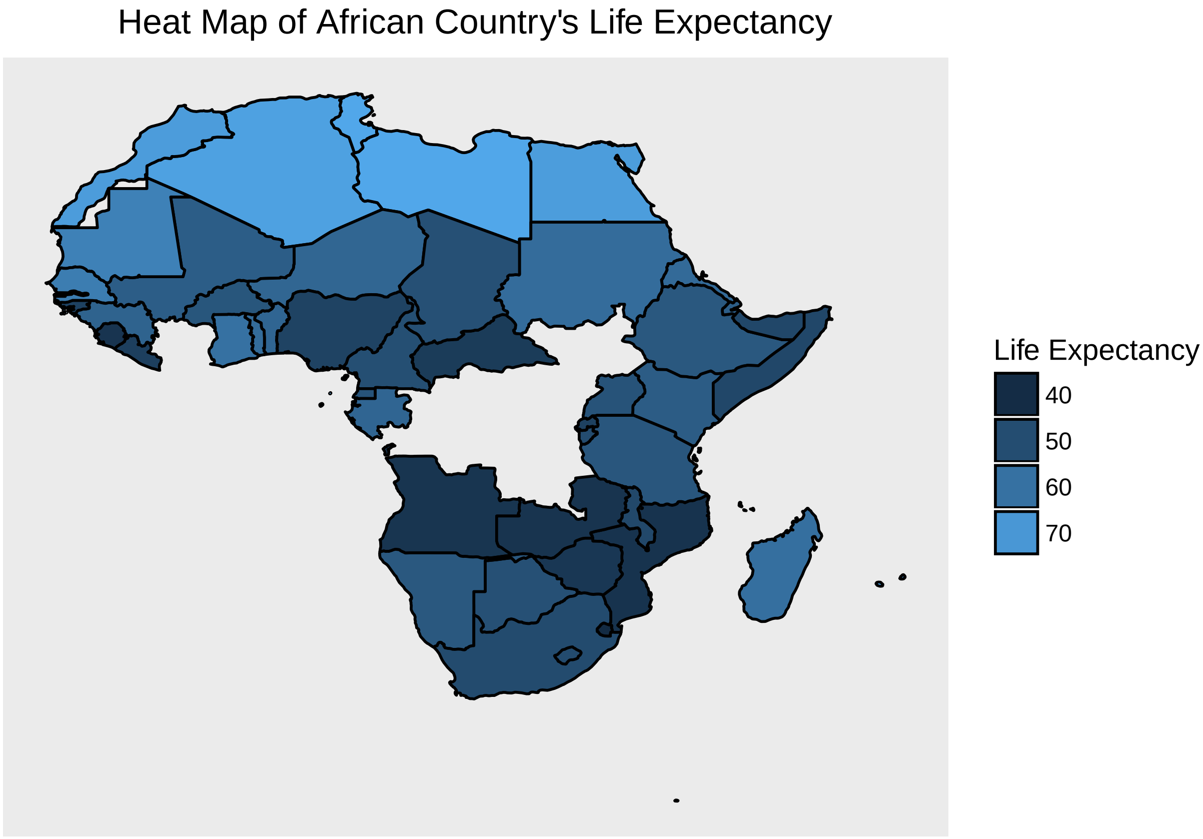
```
africa %>%
  filter(year == 2007) %>%
  ggplot() +
    geom_polygon(aes(long, lat, group = group, fill = lifeExp), color = "black") +
    no_axes
```



What improvements could be made?

Remember plotting maps with ggplot is similar to regular plots, so the same "fixes" apply

```
africa %>%
  filter(year == 2007) %>%
  ggplot() +
    geom_polygon(aes(long, lat, group = group, fill = lifeExp), color = "black") +
    no_axes +
    ggtitle("Heat Map of African Country's Life Expectancy") +
    theme(plot.title = element_text(hjust = 0.5)) +
    guides(fill = guide_legend(title = "Life Expectancy"))
```

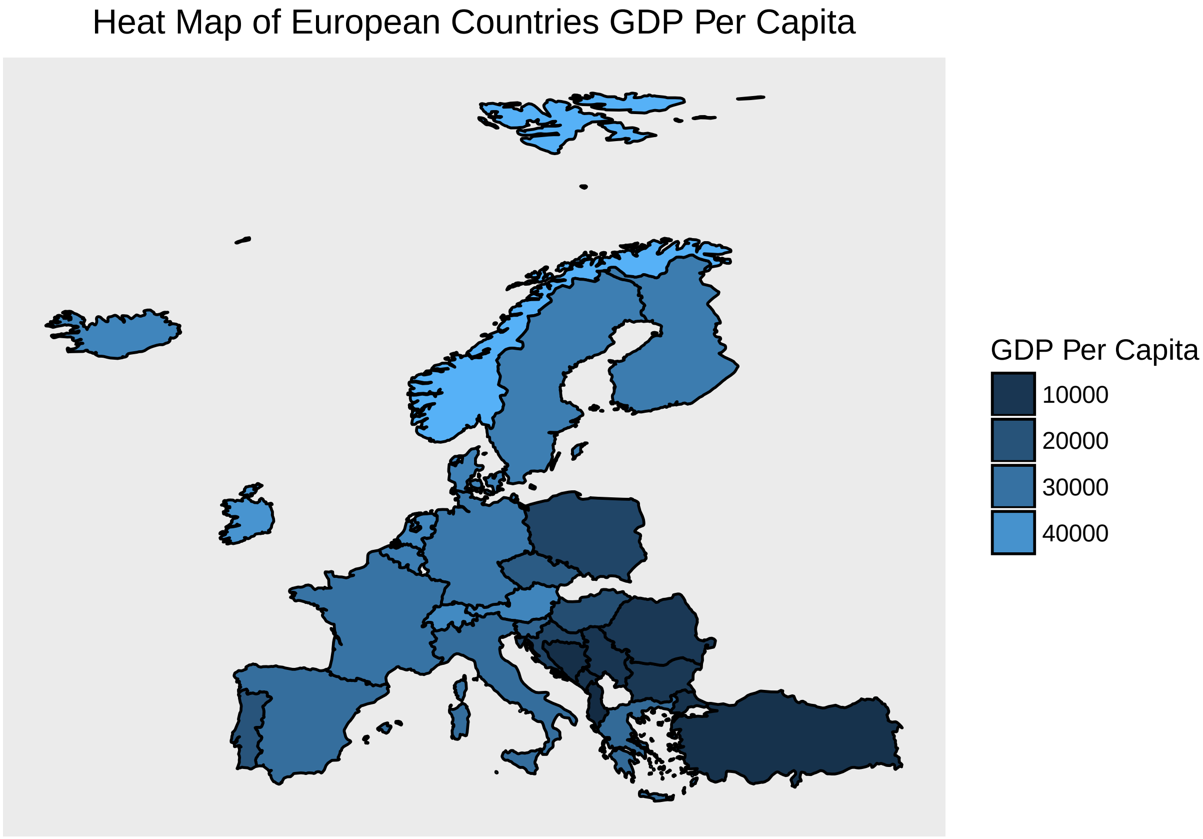


```
    #scale_fill_gradient(colours = jet.colors) #changing the color scale is being a bit weird
```

IN CLASS EXERCISE: Please plot a heat map of the European's Countries Per Capita GDP for the year 2007?

```
world_data %>%
  filter(continent == "Europe") %>%
  inner_join(world, by = c("country" = "region")) %>%
  filter(year == 2007) %>%
  ggplot() +
    geom_polygon(aes(long, lat, group = group, fill = gdpPercap), color = "black") +
    no_axes +
    ggtitle("Heat Map of European Countries GDP Per Capita") +
    theme(plot.title = element_text(hjust = 0.5)) +
    guides(fill = guide_legend(title = "GDP Per Capita"))
```

```
## Warning: Column `country`/`region` joining factor and character vector,
## coercing into character vector
```



```
#Need to fix GB, variable type, and color scale
```

Currently I am trying to see if there a more friendly way to do the animation, if not we can just plot the capitals of the nations.

```
#span <- seq(1952,2007,5)
#plot_list <- list(seq(1,12))
#for (i in seq(1,12)){
  #plot_list[i] <- africa %>%
  #filter(year == span[i]) %>%
  #ggplot() +
    #geom_polygon(aes(long, lat, group = group, fill = lifeExp), color = "black") +
    #ditch_the_axes
  #}
```