GGPLOT2 Tutorial

Carlos Yordan

6/3/2022

For this tutorial, We will be using three data sources:

- The metadata for my team's UN General Debate (UNGD) project.
- The `gapminder` database which includes a ton of economic and demographic variables
- Data on the ratification of the UN's core human rights instruments, collected by Jeehae and Julia

The Number of Democracies and Dictatorships in the UNGD (1970-2021)

```
setwd("E:/DHSI_2022/UNGD_1970_2021")


meta1 <- file.choose("merge_copy_undg_v2_june2022-merge_copy(1).csv")

meta1 <- read.csv(meta1)

head(meta1)
```

```
##   year session iso                                country            speaker
## 1 1970      25 ALB                                  Albania               Nase
## 2 1970      25 ARG                                Argentina Mr. DE PABLO PARDO
## 3 1970      25 AUS                                Australia        Mr. McMAHON
## 4 1970      25 AUT                                  Austria Mr. KIRCHSCHLAEGER
## 5 1970      25 BEL                                  Belgium         Mr. HARMEL
## 6 1970      25 BLR Byelorussian Soviet Socialist Republic     Mr. GURINOVICH
##        last_name                                post gender regional_group
## 1           Nase        Minister for Foreign Affairs   male            EES
## 2 De Pablo Pardo        Minister for Foreign Affairs   male         GRULAC
## 3        McMahon        Minister for Foreign Affairs   male           WEOG
## 4 Kirchschlaeger        Minister for Foreign Affairs   male           WEOG
## 5         Harmel        Minister for Foreign Affairs   male           WEOG
## 6     Gurinovich Second Minister for Foreign Affairs   male            EES
##          political_system head fm deputy_fm un_rep other vp_deputy_pm
## 1   Military dictatorship    0  1         0      0     0            0
## 2   Military dictatorship    0  1         0      0     0            0
## 3 Parliamentary democracy    0  1         0      0     0            0
## 4        Mixed democratic    0  1         0      0     0            0
## 5 Parliamentary democracy    0  1         0      0     0            0
## 6   Civilian dictatorship    0  1         0      0     1            0
##   dictatorship democracy  type_polsys  X nam unsc
## 1            1         0 dictatorship NA   0    0
## 2            1         0 dictatorship NA   0    0
## 3            0         1    democracy NA   0    0
## 4            0         1    democracy NA   0    0
## 5            0         1    democracy NA   0    0
## 6            1         0 dictatorship NA   0    0
```

This plot is similar to the one in the PowerPoint presentation which measured the number of male and female leaders delivering statements at the UNGD.

First, we need to wrangle the data using mostly `dplyr`.

I noticed that the year variable in the “meta” dataframe was coded as a “character” rather than numeric. Here is a quick way to change the format, using “base” `r`.

I also noticed that my gender variable needs to be changed to lowercase.

```
meta1$year <- as.numeric(meta1$year)

meta1$gender_lower <- tolower (meta1$gender)

head(meta1)
```

```
##   year session iso                                country            speaker
## 1 1970      25 ALB                                  Albania               Nase
## 2 1970      25 ARG                                Argentina Mr. DE PABLO PARDO
## 3 1970      25 AUS                                Australia        Mr. McMAHON
## 4 1970      25 AUT                                  Austria Mr. KIRCHSCHLAEGER
## 5 1970      25 BEL                                  Belgium         Mr. HARMEL
## 6 1970      25 BLR Byelorussian Soviet Socialist Republic     Mr. GURINOVICH
##        last_name                              post gender regional_group
## 1           Nase        Minister for Foreign Affairs   male            EES
## 2 De Pablo Pardo        Minister for Foreign Affairs   male         GRULAC
## 3        McMahon        Minister for Foreign Affairs   male           WEOG
## 4 Kirchschlaeger        Minister for Foreign Affairs   male           WEOG
## 5         Harmel        Minister for Foreign Affairs   male           WEOG
## 6     Gurinovich Second Minister for Foreign Affairs   male            EES
##          political_system head fm deputy_fm un_rep other vp_deputy_pm
## 1   Military dictatorship    0  1         0      0     0            0
## 2   Military dictatorship    0  1         0      0     0            0
## 3 Parliamentary democracy    0  1         0      0     0            0
## 4        Mixed democratic    0  1         0      0     0            0
## 5 Parliamentary democracy    0  1         0      0     0            0
## 6   Civilian dictatorship    0  1         0      0     1            0
##   dictatorship democracy  type_polsys  X nam unsc gender_lower
## 1            1         0 dictatorship NA   0    0         male
## 2            1         0 dictatorship NA   0    0         male
## 3            0         1    democracy NA   0    0         male
## 4            0         1    democracy NA   0    0         male
## 5            0         1    democracy NA   0    0         male
## 6            1         0 dictatorship NA   0    0         male
```

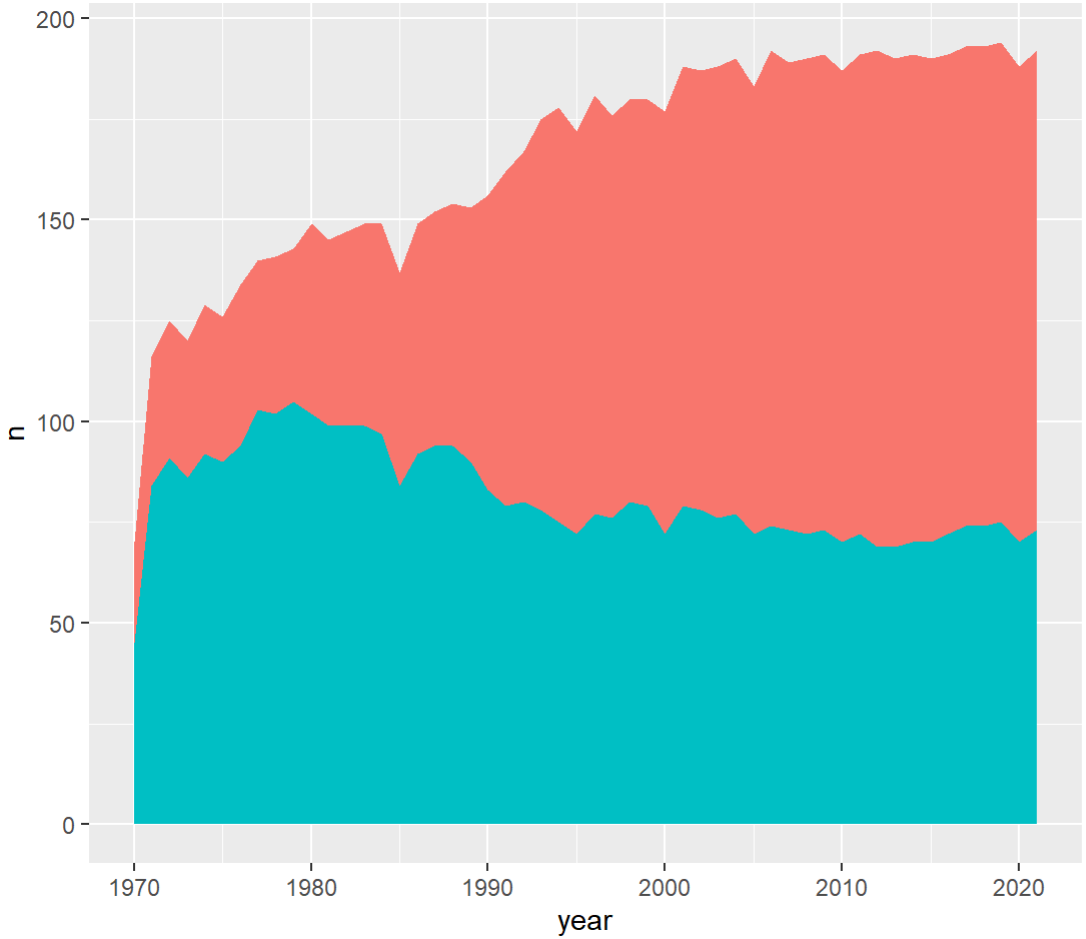
We will use ‘dplyr’ to count the number of democracies and dictatorships (see: “type_polsys”) participating in the UNDG per year. Note that we have some “n/a”s in our data and we will filter them out.

```
dem_dict <- meta1 %>%
  group_by (year) %>%
  count (type_polsys)%>%
  filter (type_polsys != "n/a")
```

Then we feed this data into `ggplot`

```
ggplot (dem_dict, aes (x=year, y=n, fill= type_polsys)) +
  geom_area (stat= "identity", condition = "stack")
```

```
## Warning: Ignoring unknown parameters: condition
```



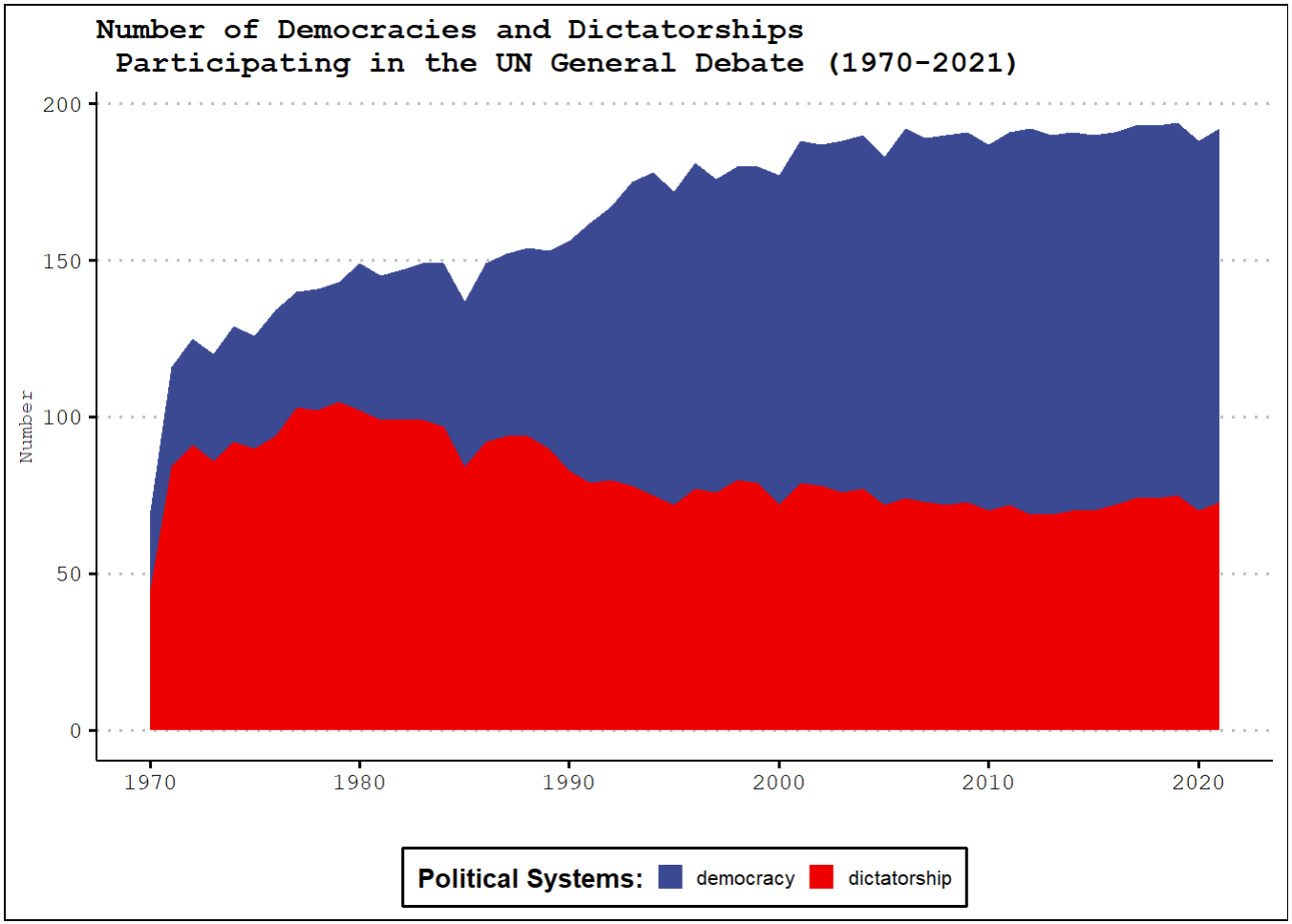
This is our basic graph. Let's go back to the PowerPoint. So far, we have added three layers:

- *data*
- *aes*
- *geom*

Now that we have the basic plot done we can start working on the theme. For this plot, we will NOT use *facet* or *stats* layers.

```
ggplot (dem_dict, aes (x=year, y=n, fill= type_polsys)) +
  geom_area (stat= "identity", condition = "stack") +
  theme_clean () +
  scale_fill_aaas() +
  labs (fill= "Political Systems:")+
  theme (text = element_text(family= "mono"))+
  theme (legend.position = "bottom",
         legend.title = element_text (size=10, face ="bold"),
         legend.text = element_text(size = 8),
         legend.key.size = unit(.75,"line"))+
  theme (plot.title = element_text(size=12, face="bold"),
         axis.title = element_text (size=8))+
  labs (title= "Number of Democracies and Dictatorships\n Participating in the UN General Debate (1970-2021)",
        x = "",
        y = "Number")
```

```
## Warning: Ignoring unknown parameters: condition
```



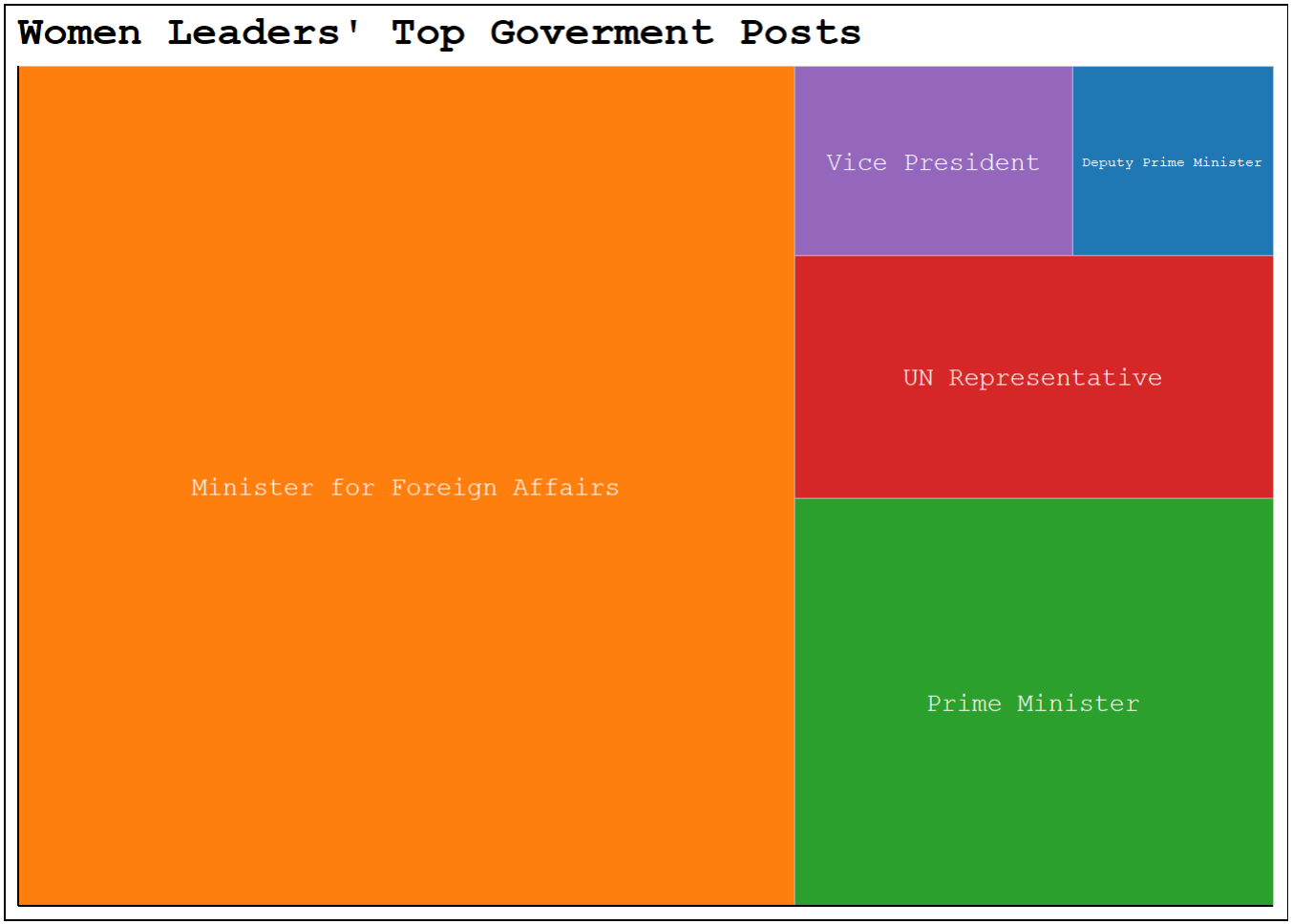
Treemap Plot

Using the UNGD metadata, we can try to figure out what governmental positions most of the female leaders hold when they address the UN General Debate. Note the code. We are interested only in women leaders. So we can filter the gender_lower to only women leaders. There are more than 20 positions listed in the dataframe. We will only look at the top 6 titles. Notice that we are looking at all the years in this plot.

```
fem_post2 <- meta1 %>%
  filter (gender_lower == "female") %>%
  count (post) %>%
  arrange (desc (n)) %>%
  filter (post %in% c ("President", "Vice President", "Prime Minister", "Deputy Prime Minister", "Minister for Foreign Affai
rs", "UN Representative"))
```

Let's feed the data to `ggplot` . Because we are plotting a treemap, we need to make sure that the `treemapify` package is already loaded - which I did so already. Note that the syntas used by the authors of this package is in line with `ggplot` 's grammar of graphics.

```
ggplot(fem_post2,
       aes(fill = post,
           area = n,
           label = post)) +
  geom_treemap() +
  theme_clean ()+
  geom_treemap_text(color ="white", place = "centre", family = "mono", size =10)+
  theme(legend.position = "none")+
  labs(title = "Women Leaders' Top Goverment Posts")+
  theme (plot.title = element_text(size=16, face="bold", family="mono"))+
  scale_fill_d3()
```



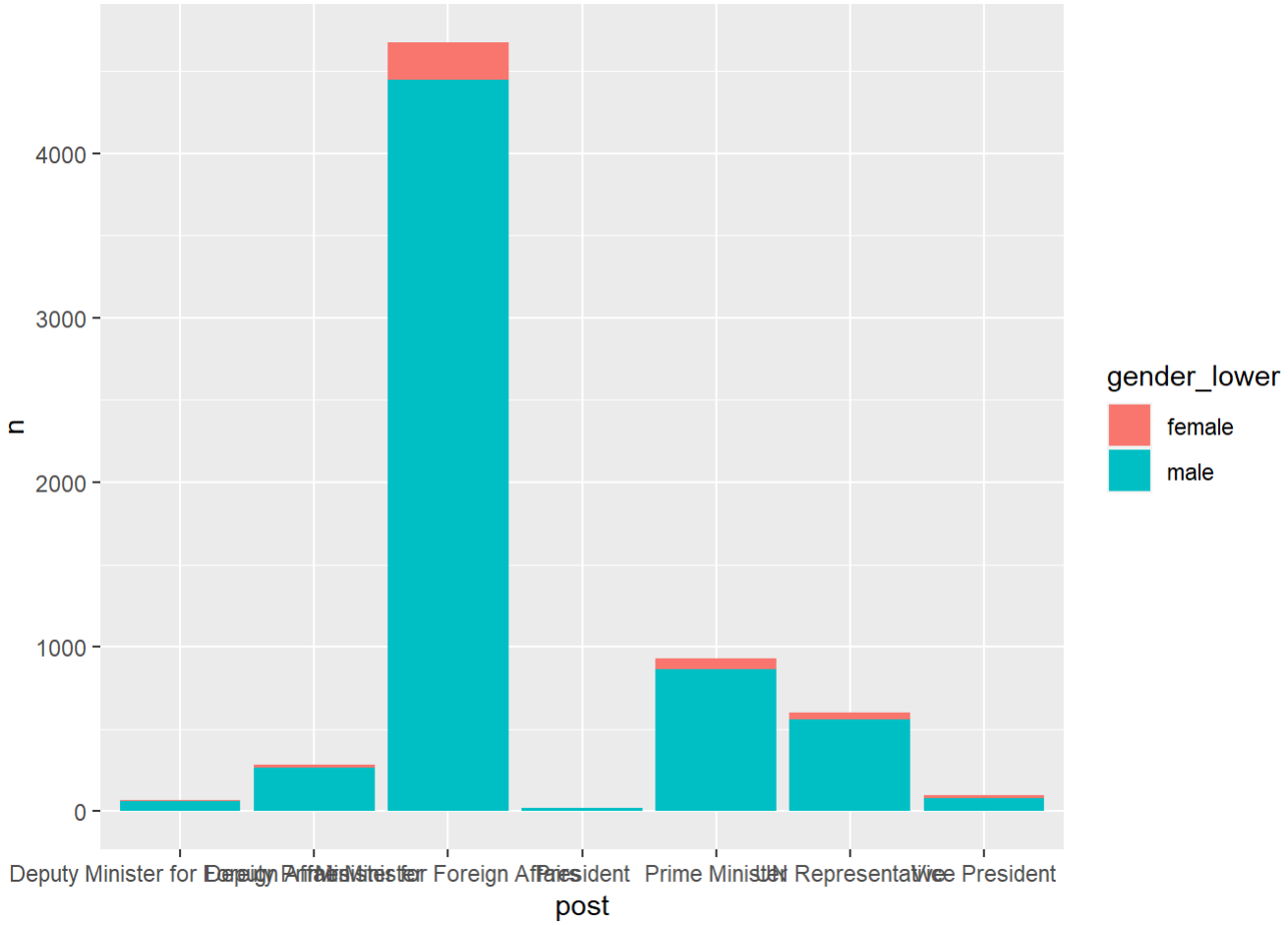
GEOM_COL Graph of Government Posts by Gender

Let's wrangle and prepare the data from the UNGD metadata.

```
gender_post <- meta1 %>%
  group_by (gender_lower) %>%
  count (post) %>%
  filter (gender_lower != "unknown") %>%
  filter (post %in% c ("President", "Vice President", "Prime Minister", "Deputy Prime Minister", "Minister for Foreign Affai
rs", "Deputy Minister for Foreign Affairs", "UN Representative"))
```

Let's run our first graph.

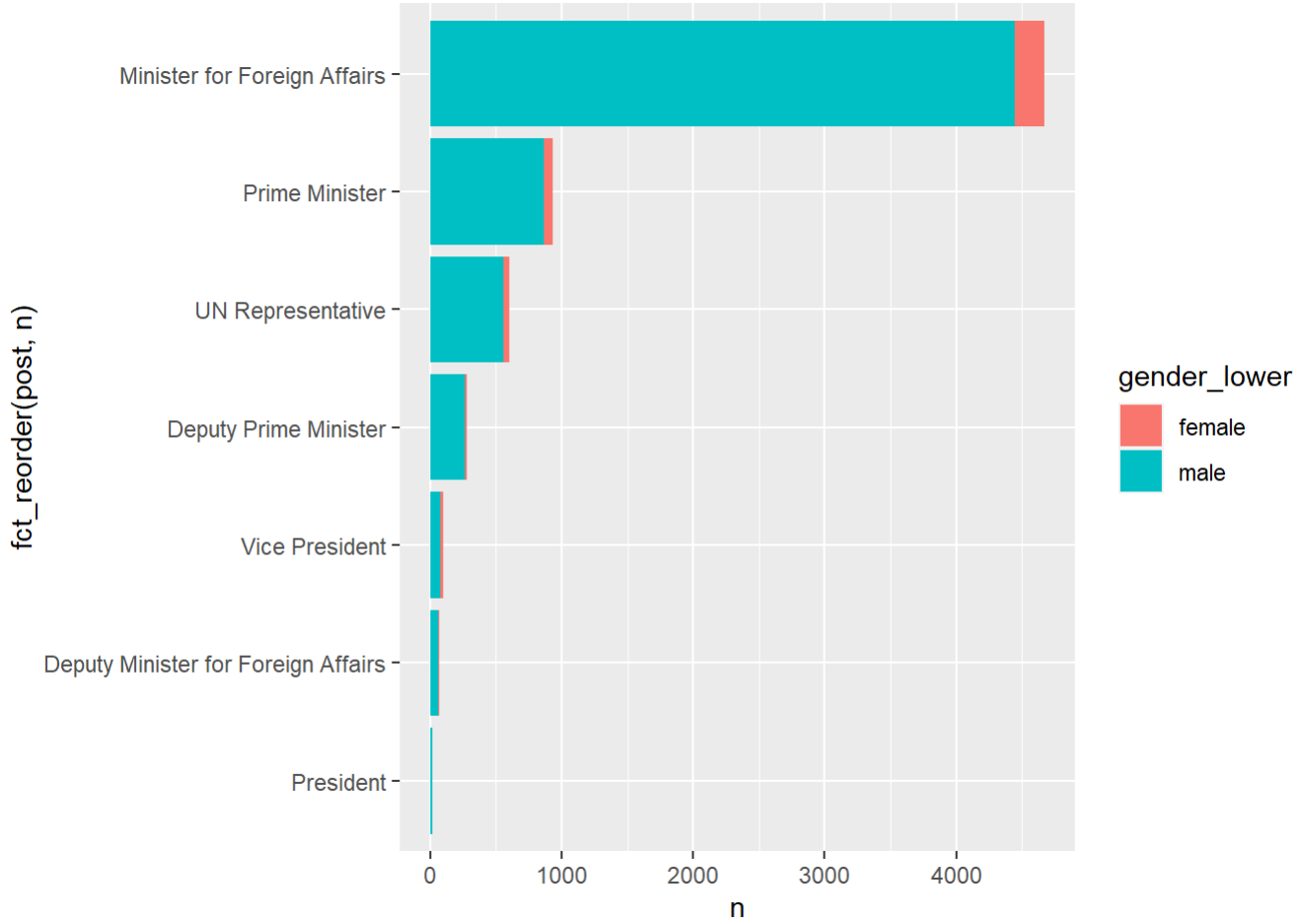
```
ggplot (gender_post, aes(x=post, y=n, fill=gender_lower))+
  geom_col ()
```



I prefer to reorder from large to small and also flip the graph. We will use “coord_flip” function which is a *coordinate* layer!

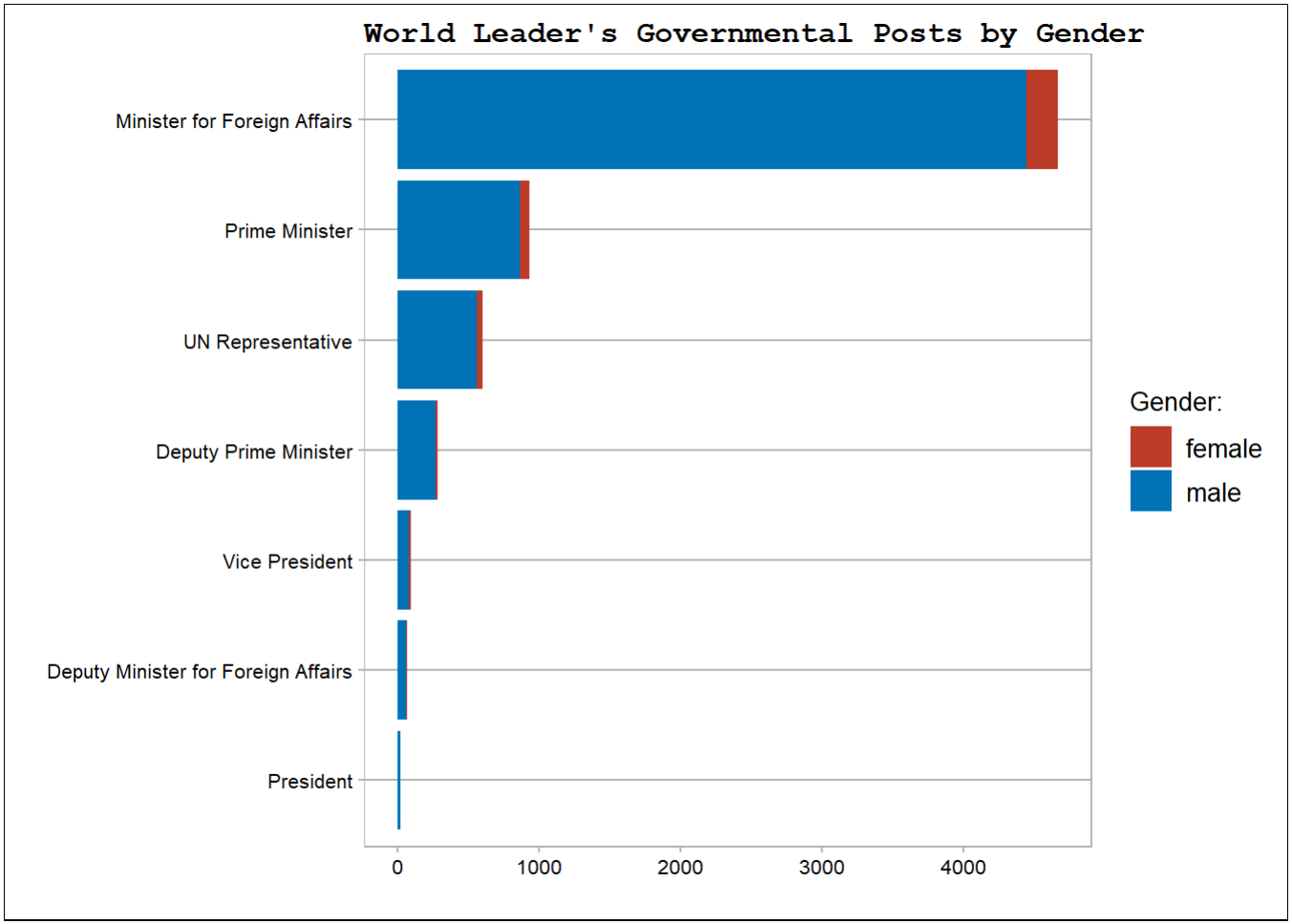
In addition, note in the *aes* the code snippet after “x=”. I added the fct_reorder command to reorder the posts from highest to lowest. This is technically a *stat* layer but I did so in the *aes* layer.

```
ggplot (gender_post, aes (x=fct_reorder (post, n), y=n, fill =gender_lower)) +
  geom_col () +
  coord_flip()
```



Then we can add *theme* layers.

```
ggplot (gender_post, aes (x=fct_reorder (post, n), y=n, fill =gender_lower)) +
  geom_col () +
  coord_flip()+
theme_calc ()+
  theme(legend.position = "right")+
  labs(title = "World Leader's Governmental Posts by Gender")+
  theme (plot.title = element_text(size=12, face="bold", family="mono", hjust= 0, vjust = 0))+
  labs (fill= "Gender:",
        x="",
        y="")+
  scale_fill_nejm()
```



Scatterplot using Gapminder

Let's see what the relationship is between GDP per capita and life expectancy. I don't have to wrangle the data as it is already organized in the `gapminder` package into a dataframe.

```
summary(gapminder)
```

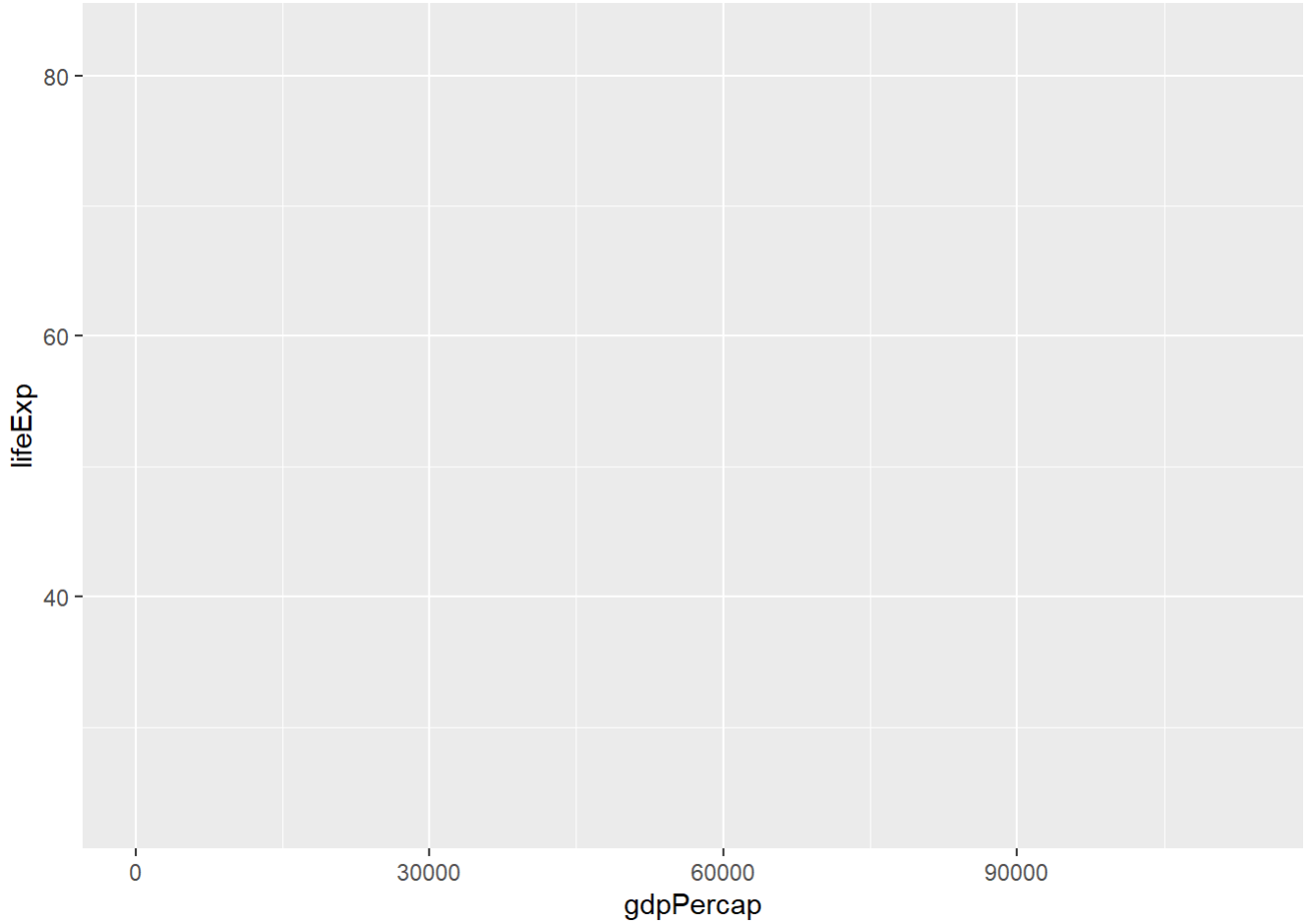
```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       pop              gdpPercap
##  Min.   :6.001e+04   Min.   :   241.2
##  1st Qu.:2.794e+06   1st Qu.:  1202.1
##  Median :7.024e+06   Median :  3531.8
##  Mean   :2.960e+07   Mean   :  7215.3
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5
##  Max.   :1.319e+09   Max.   :113523.1
##
```

Let's go back to `ggplot` 's layering system.

We will only feed `ggplot` the:

- *data*
- *aes*

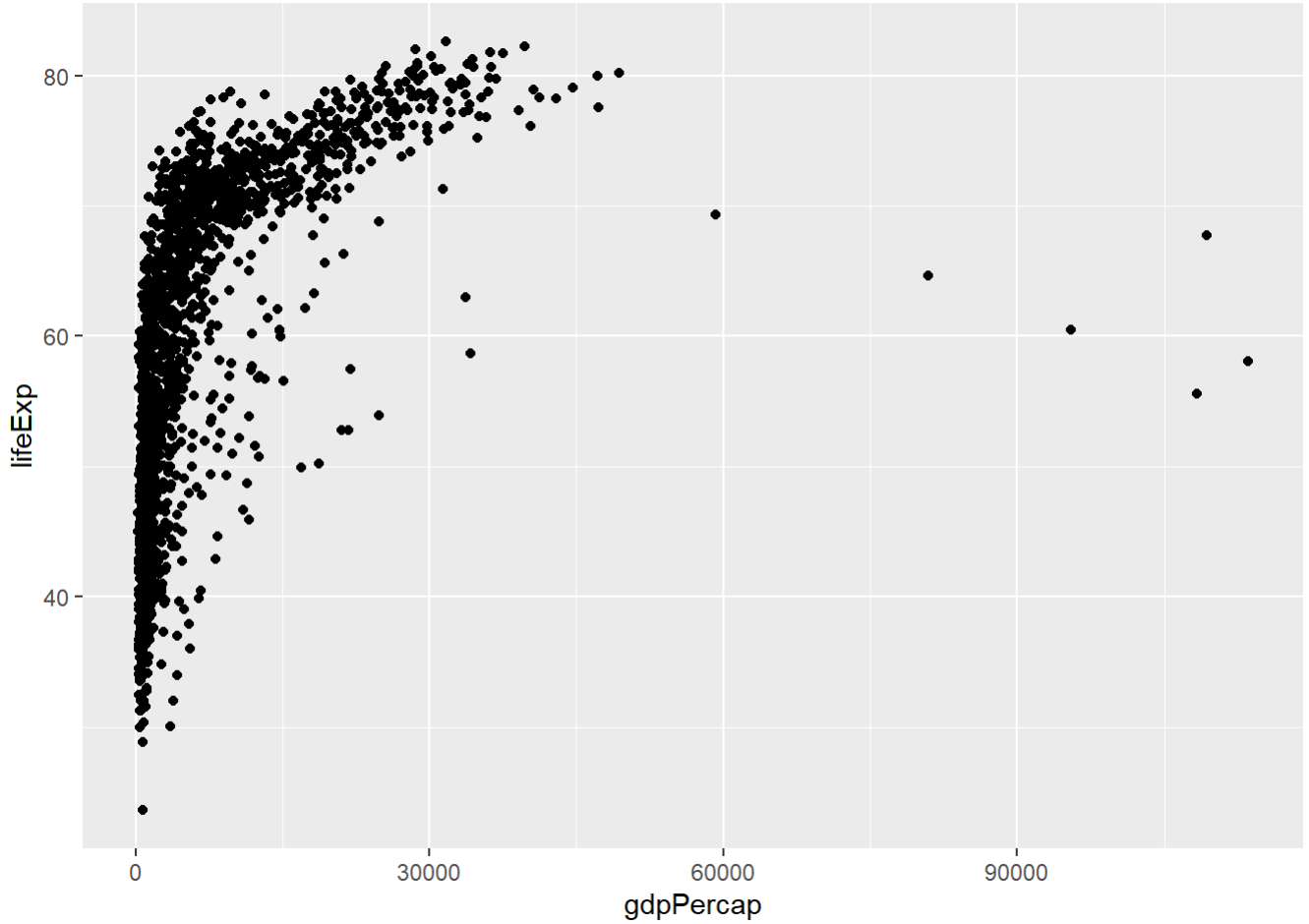
```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp))
```



This plot is empty because we did not

a *geom*, thus `ggplot` does not know HOW to plot the data.

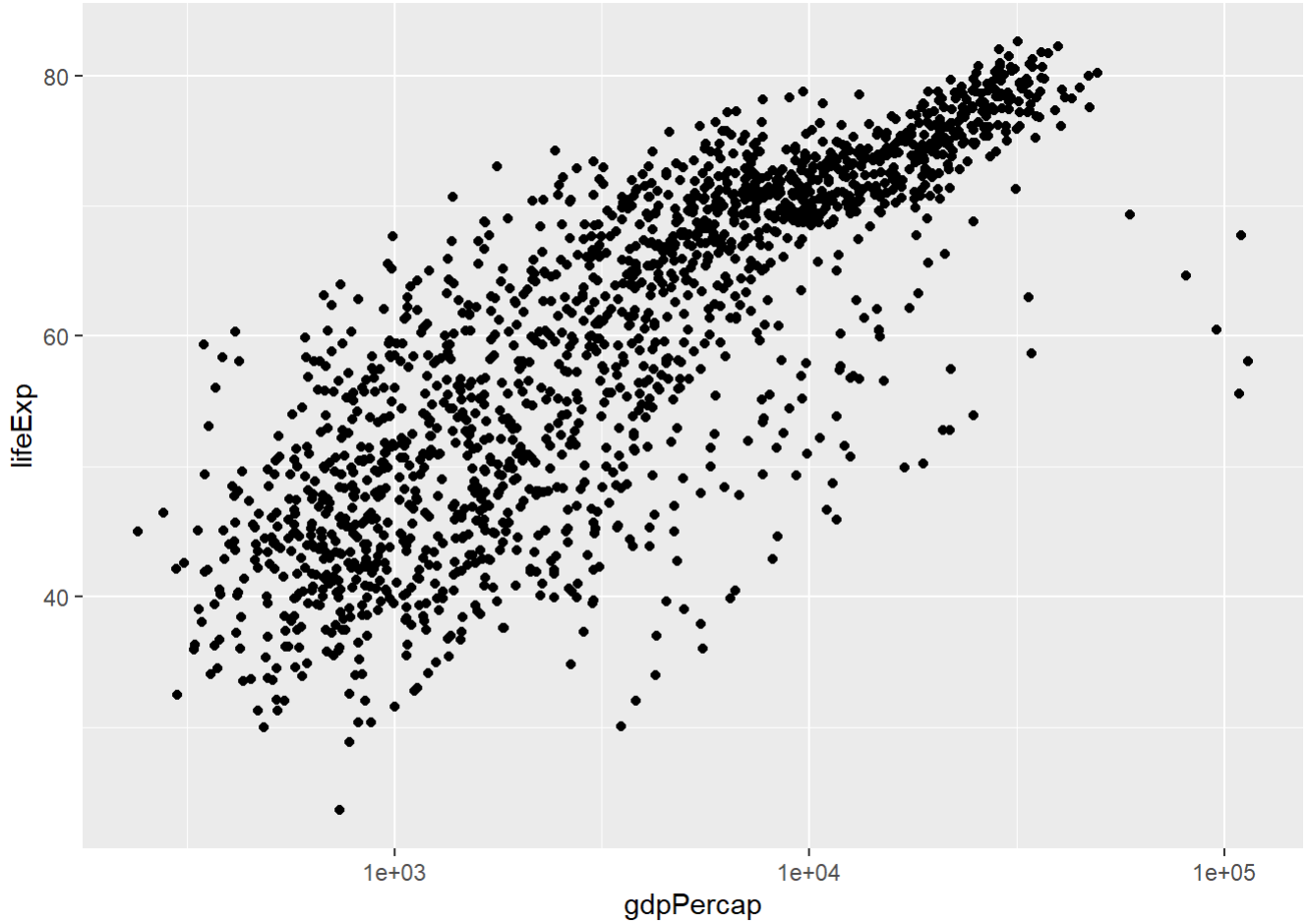
```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



Outliers in the “gdpPercap” variable are skewing the distribution of the data. Let’s transform the data by using a simple log transformation.

Now we are adding a *stat* layer to our plot!

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()+
   scale_x_log10()
```



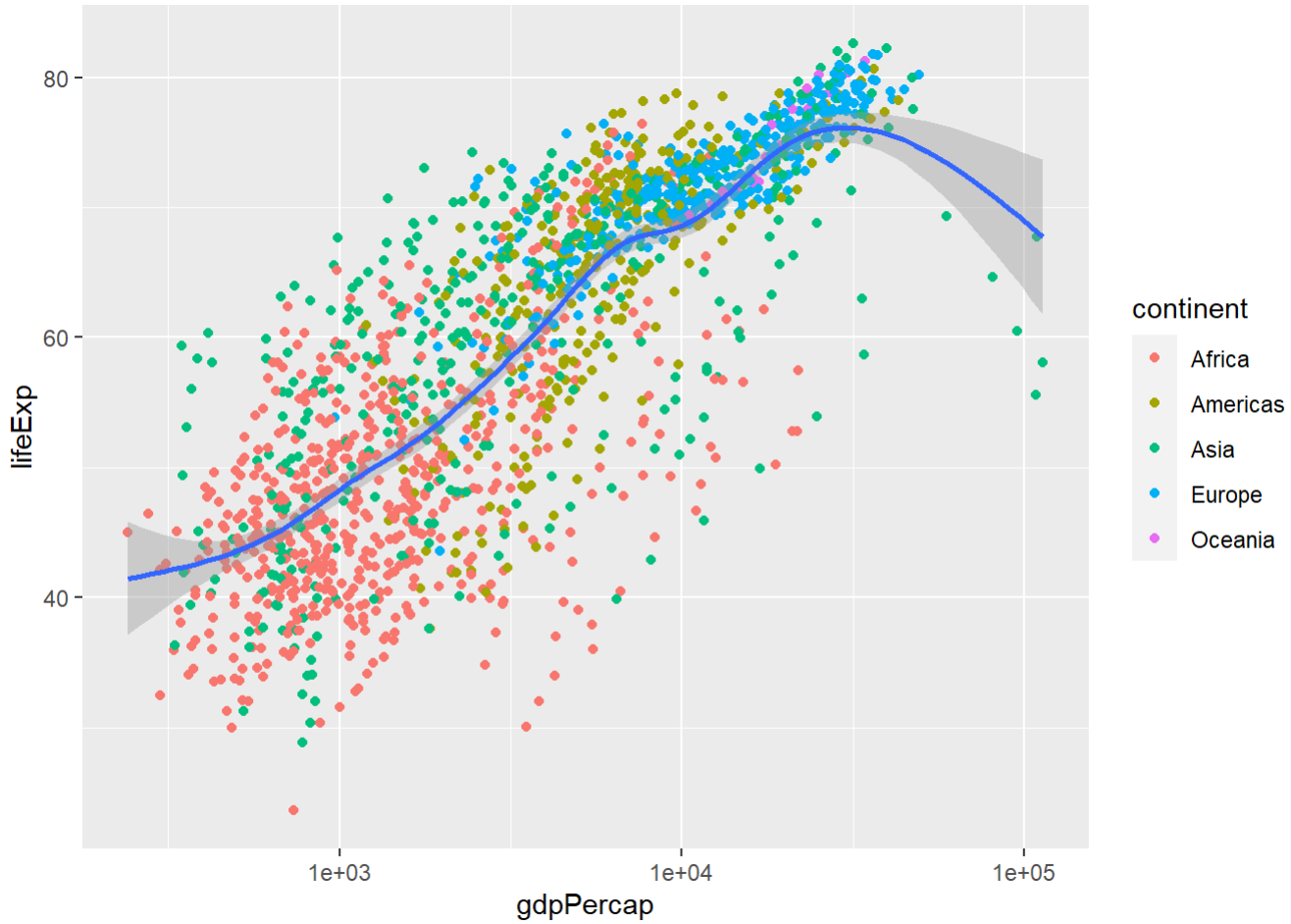
Could we do something else? We could color the dots by continent. This is an *aes* layer and we can put it in the “geom_point” function OR in the “aes” function, after the input for “y”.

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes (color=continent))+
   scale_x_log10()
```

We could also try to fit a line to make sense what type of relationship we have between both variables. Again, we are now entering a new *stat* layer.

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes (color=continent))+
   scale_x_log10()+
  geom_smooth()
```

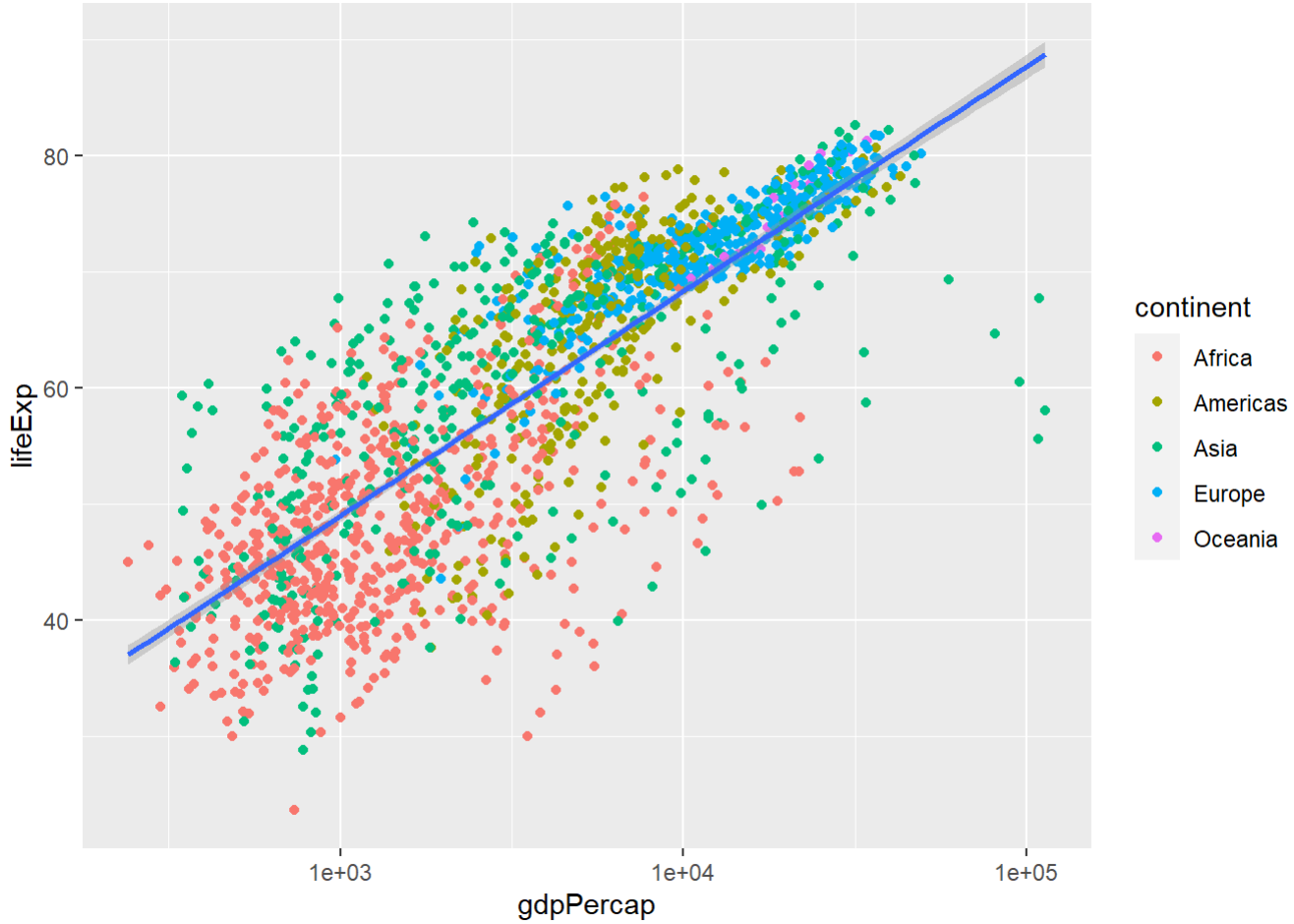
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



We can also enter a linear model (LM) instead of the smooth line.

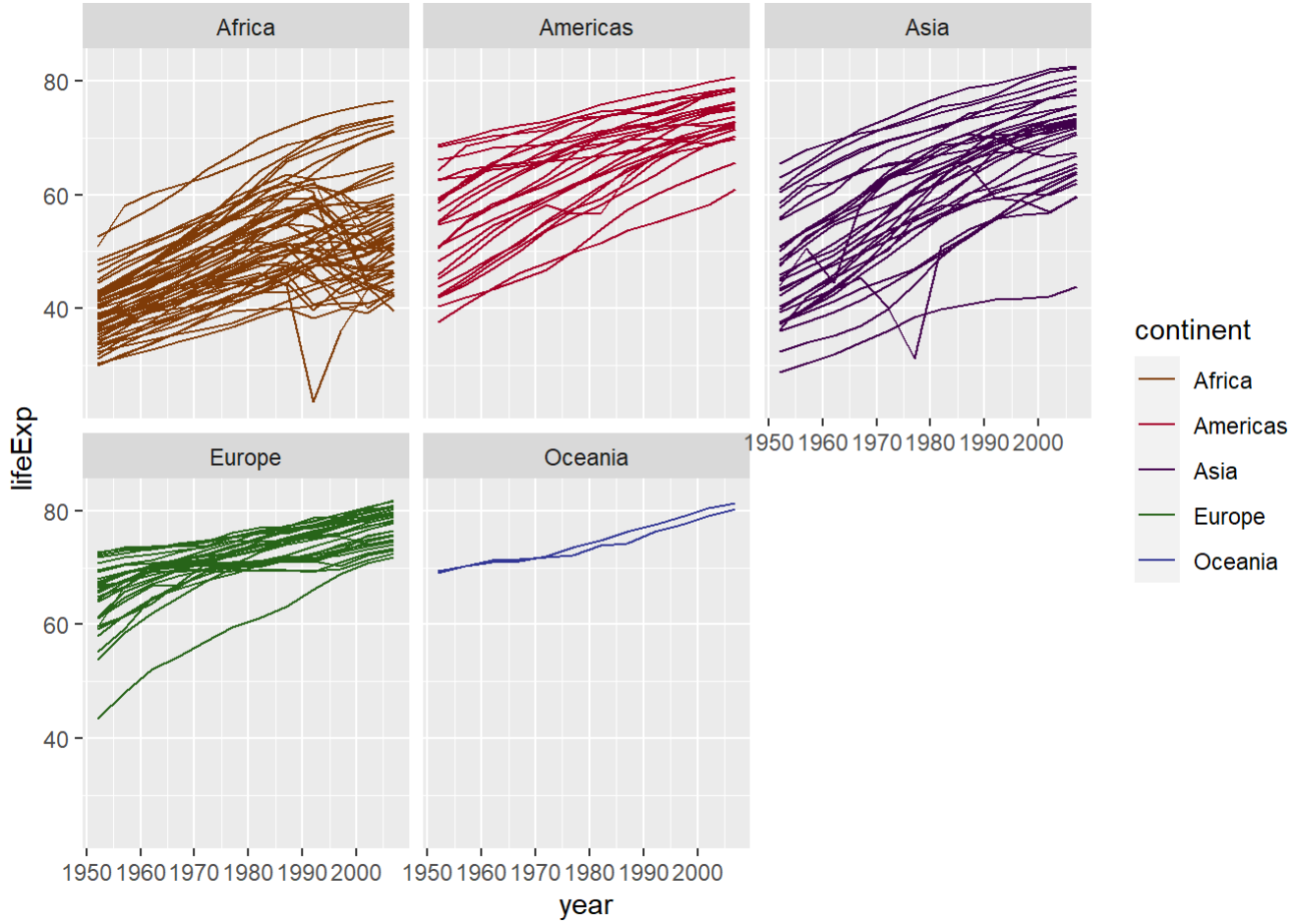
```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes (color=continent))+
   scale_x_log10()+
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Let's use *facet* layer. In this graph we will use a "geom_line" rather than "geom_point". Each line represents a country.

```
ggplot(gapminder, aes(x = year, y = lifeExp, group= country,
                      color = continent)) +
  geom_line ()+
  facet_wrap(~ continent) +
  scale_color_manual(values = continent_colors)
```

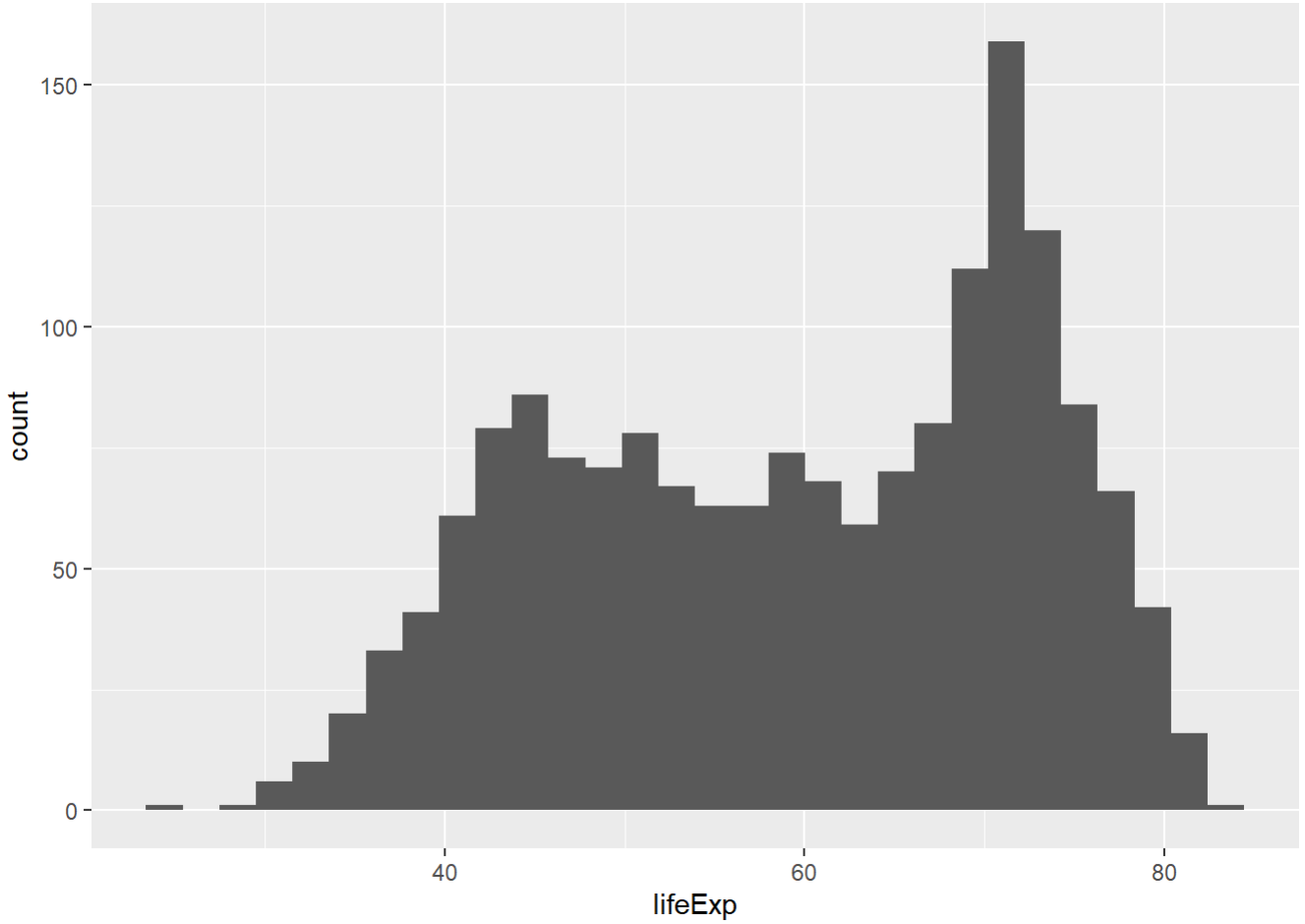


Of course, we can add *theme* layers to make these graphs look prettier.

Making a Histogram using Gapminder Data

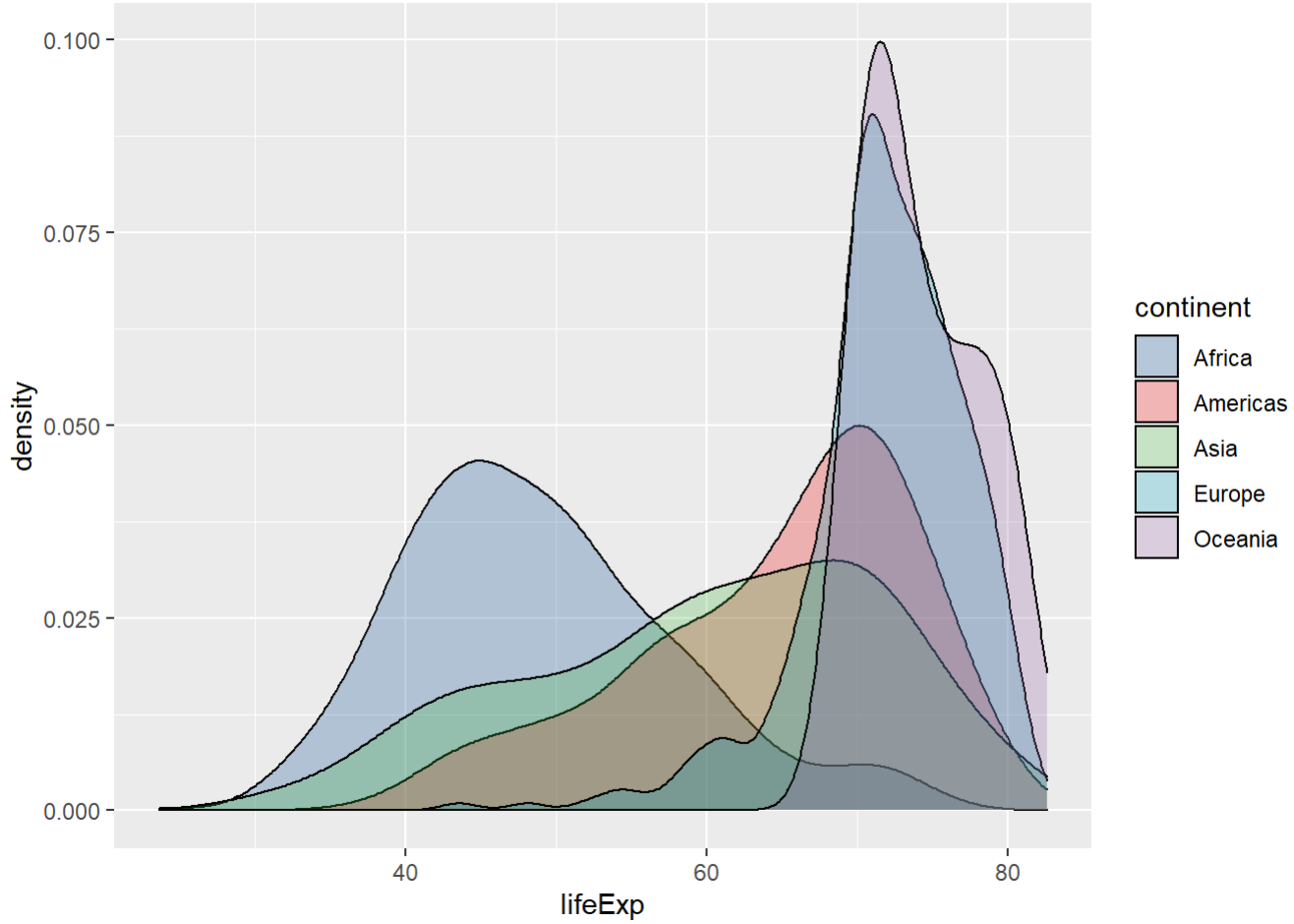
```
ggplot(gapminder, aes(lifeExp))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can also turn this histogram into a density plot and we can use the “continent” variable to map the density plot per continent. Notice that I used an “alpha” command to control for the transparency of the plot.

```
ggplot(gapminder, aes(lifeExp))+
 geom_density(aes(fill=continent), alpha= 0.25) +
  scale_fill_lancet()
```

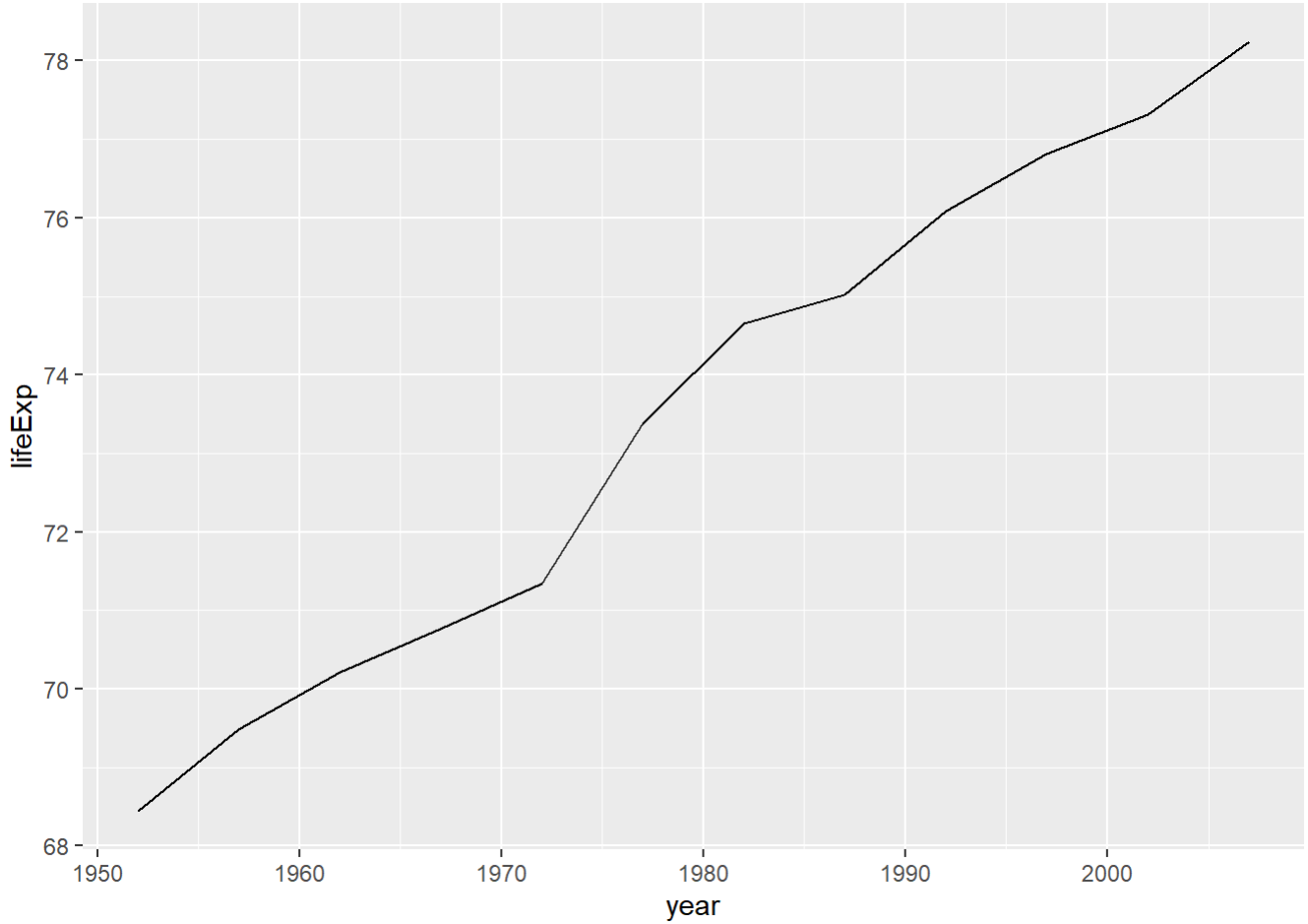


It is worth noting that distribution of the data is the same in the histogram and density plot!

Line graphs using the Gapminder Dataset

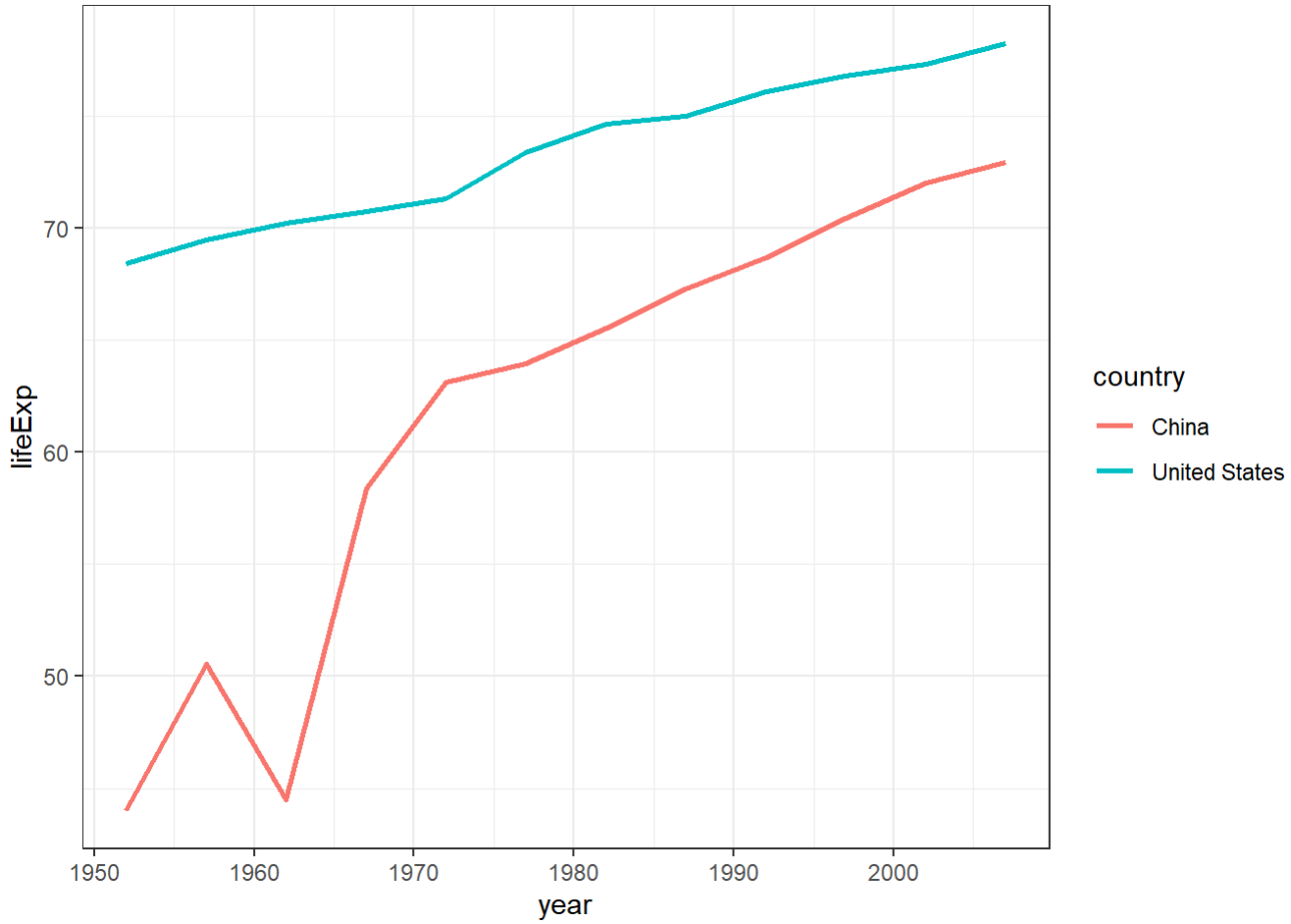
Let's look at life expectancy a bit more in depth. We will be using both `dplyr` and `ggplot` together in this example. Let's plot the life expectancy rate in the United States.

```
gapminder %>%
  filter(country == "United States") %>%
  ggplot(aes(x=year, y=lifeExp)) +
  geom_line()
```



How about if we want to compare a few countries. Note in “geom_line” I asked to make the line a bit thicker using the “size” command.

```
countries <- c("United States", "China")

gapminder %>% filter(country %in% countries) %>%
  ggplot(aes(x=year,y=lifeExp, color = country)) +
  geom_line(size = 1)+
  theme_bw()
```



Let’s Make Choropleth Map

Let’s make sure we have all the necessary packages.

```
library(tidyverse)
library(maps)
library(ggmap)
library(countrycode)
library(RColorBrewer)
library (ggthemes)
library (sf)
library (ggsci)
library (extrafont)
library (viridis)
```

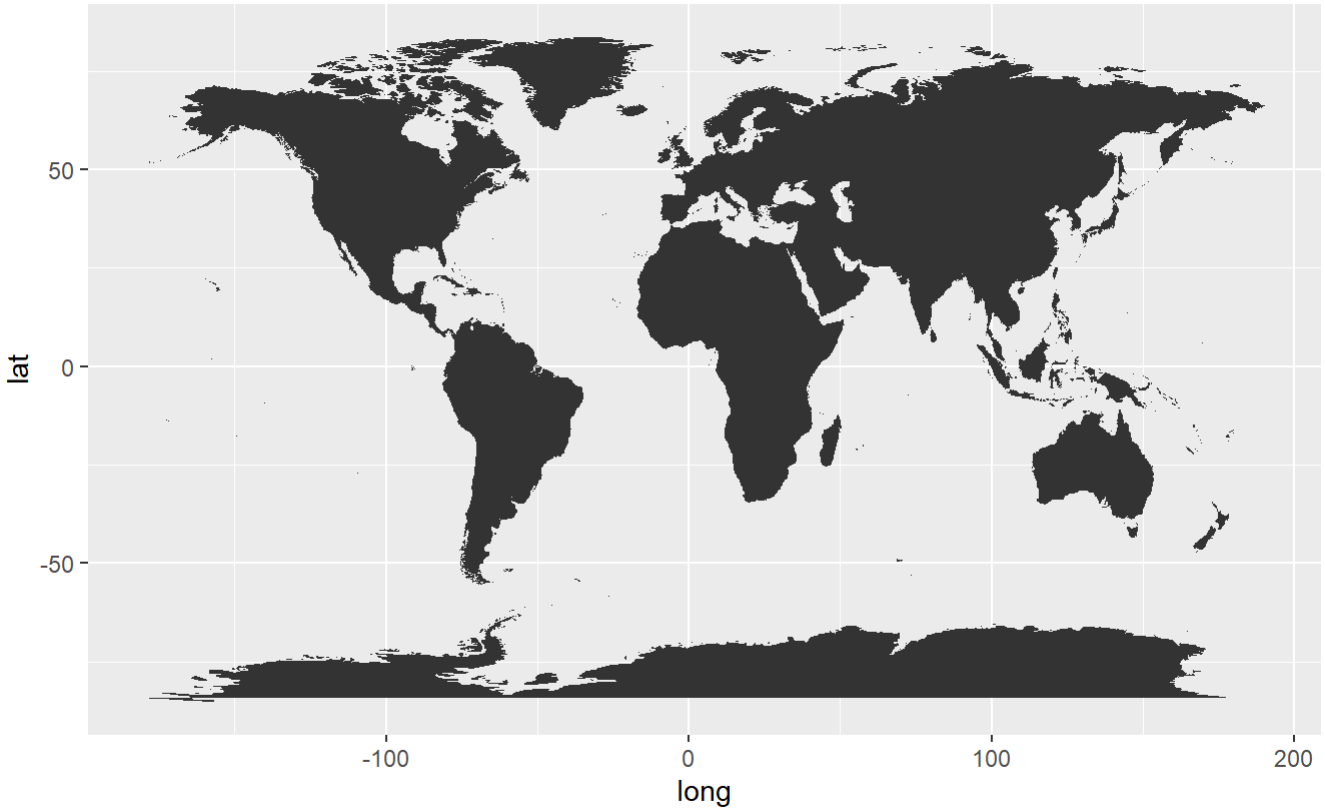
We will use the data collected by Jeehae and Julia on ICERD's ratification status.

We will wrangle and prepare the data.

```
# Read the data from the GitHub repository:

hr <- read.csv ("https://raw.githubusercontent.com/world-politics-datalab/humanrights-treaties-ratification-status/main/hr_status_icerd.csv")

#For mapping purposes, we will add Greenland to the dataset. Greenland is a Danish territory. So we will add the same information as Denmark's. This is the "dplyr" code

hr_gr <- hr %>%
  add_row(iso2 = "GL", member= "Greenland", state.party = 1, year.ratification.or.accession= 1971, signatory=0, no.action=0, status="state party")

# Let's merge the information on Greenland back to "hr" dataset:

hr <- hr_gr

# The ISO2 abbreviation for Namibia is NA, but R read the NA as n/a. We need to fix this. So I used the "countrycode" package to enter all the ISO2 abbreviations, fixing the problem.

hr$iso2 <- countrycode (hr$member, origin= "country.name", destination = "iso2c", )
```

We need to get a dataset with a world map, including longitude and latitude data. We need to use `ggplot` to see what we are working with.

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

# check the map using ggplot:

worldplot <- ggplot()+
  geom_polygon(data=wmap, aes(x=long, y=lat, group = group))+
  coord_fixed(1.3)

worldplot
```



We will use `dplyr` to combine the two datasets.

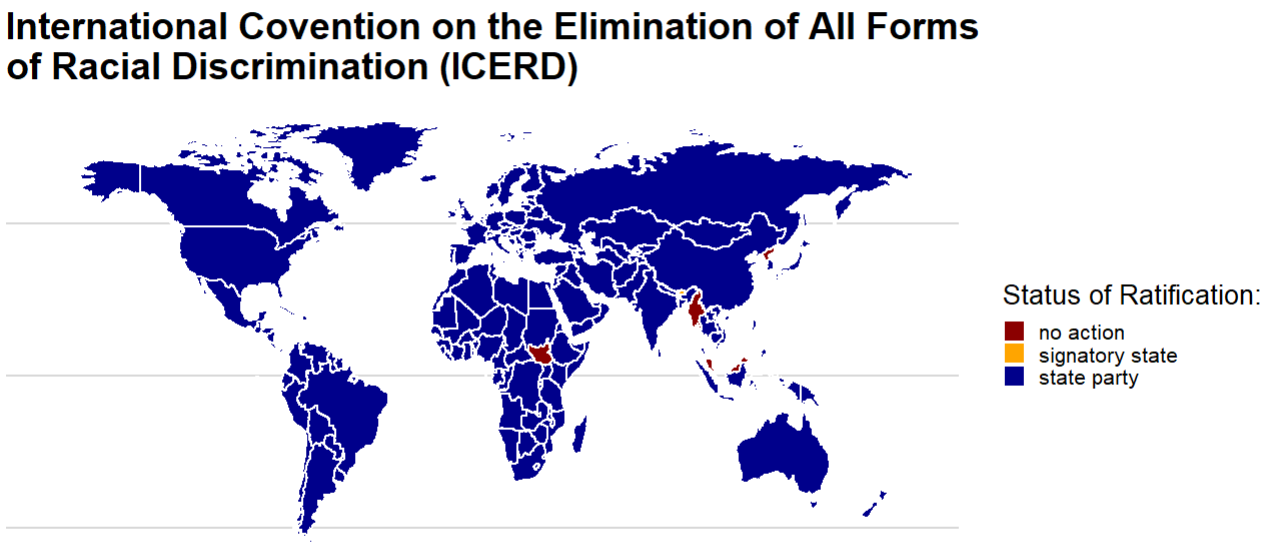
```
#let's add iso2

wmap$iso2 <- countrycode (wmap$region, origin= "country.name", destination = "iso2c", )
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some values were not matched
unambiguously: Ascension Island, Azores, Barbuda, Bonaire, Canary Islands, Chagos Archipelago, Grenadines, Heard Island, Kos
ovo, Madeira Islands, Micronesia, Saba, Saint Martin, Siachen Glacier, Sint Eustatius, Virgin Islands
```

```
#join the map with the hr data

wmap_hr <- wmap %>%
  left_join (hr, by = c("iso2" = "iso2"))

## filter out NAs

wmap_hr <- filter(wmap_hr, !is.na(status))
```

Now we can use `ggplot` to create the map.

```
ggplot(wmap_hr, aes(long, lat, group = group)) +
  coord_fixed(1.3)+
  geom_polygon(aes(fill = status))+
  geom_polygon(data =wmap_hr, colour = "white", fill = NA) +
  ggtitle("International Covention on the Elimination of All Forms\nof Racial Discrimination (ICERD)") +
  scale_y_continuous()+
  scale_fill_manual (values = c("darkred", "orange", "darkblue"))+
  #scale_fill_aaas ()+
  theme_hc()+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        rect = element_blank())+
  theme(plot.title = element_text(size = 14))+
    theme(plot.title = element_text(face = "bold"))+
    theme (legend.position = "right",
          legend.title = element_text(color = "black", size=10),
          legend.text = element_text (color = "black", size =8),
          legend.key.size = unit(0.3, 'cm'),
        legend.key.height = unit(0.3, 'cm'),
        legend.key.width = unit(0.3, 'cm'),
         )+
  labs (fill = "Status of Ratification:",
        x="",
        y= "")
```



#FINITO!!!

Feel free to email: cyordan@drew.edu (mailto:cyordan@drew.edu) with questions.