Creating Engaging Plots in ggplot2: Exercises

Angela Li

2023-02-24

Load Gapminder data

```
# Load any packages needed
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Set working directory (in fact, unnecssary when using R projects!) and read in data
gapminder <- read.csv("gapminder.csv")
```

We'll start with a few commands to explore our data. This data is from Gapminder, and has been edited for use in open-source Data Carpentry materials. (If you want to know more, there's a 2006 TED talk with this data.) Our research question is how life expectancy at birth compares across countries over time.

The codebook from the survey can be found online here.

Columns in the dataset include:

- `country`
- `continent`
- `year`
- `lifeExp` or life expectancy
- `pop` or population
- `gdpPercap` or GDP per capita

```
# Take a look at the data
head(gapminder)
```

```
##       country year      pop continent lifeExp gdpPercap
## 1 Afghanistan 1952  8425333      Asia  28.801  779.4453
## 2 Afghanistan 1957  9240934      Asia  30.332  820.8530
## 3 Afghanistan 1962 10267083      Asia  31.997  853.1007
```

```
## 4 Afghanistan 1967 11537966      Asia  34.020  836.1971
## 5 Afghanistan 1972 13079460      Asia  36.088  739.9811
## 6 Afghanistan 1977 14880372      Asia  38.438  786.1134
```

```
# Find unique countries represented in the data
unique(gapminder$country)
```

```
##   [1] "Afghanistan"              "Albania"
##   [3] "Algeria"                  "Angola"
##   [5] "Argentina"                "Australia"
##   [7] "Austria"                  "Bahrain"
##   [9] "Bangladesh"               "Belgium"
##  [11] "Benin"                    "Bolivia"
##  [13] "Bosnia and Herzegovina"   "Botswana"
##  [15] "Brazil"                   "Bulgaria"
##  [17] "Burkina Faso"             "Burundi"
##  [19] "Cambodia"                 "Cameroon"
##  [21] "Canada"                   "Central African Republic"
##  [23] "Chad"                     "Chile"
##  [25] "China"                    "Colombia"
##  [27] "Comoros"                  "Congo Dem. Rep."
##  [29] "Congo Rep."               "Costa Rica"
##  [31] "Cote d'Ivoire"            "Croatia"
##  [33] "Cuba"                     "Czech Republic"
##  [35] "Denmark"                  "Djibouti"
##  [37] "Dominican Republic"       "Ecuador"
##  [39] "Egypt"                    "El Salvador"
##  [41] "Equatorial Guinea"        "Eritrea"
##  [43] "Ethiopia"                 "Finland"
##  [45] "France"                   "Gabon"
##  [47] "Gambia"                   "Germany"
##  [49] "Ghana"                    "Greece"
##  [51] "Guatemala"                "Guinea"
##  [53] "Guinea-Bissau"            "Haiti"
##  [55] "Honduras"                 "Hong Kong China"
##  [57] "Hungary"                  "Iceland"
##  [59] "India"                    "Indonesia"
##  [61] "Iran"                     "Iraq"
##  [63] "Ireland"                  "Israel"
##  [65] "Italy"                    "Jamaica"
##  [67] "Japan"                    "Jordan"
##  [69] "Kenya"                    "Korea Dem. Rep."
##  [71] "Korea Rep."               "Kuwait"
##  [73] "Lebanon"                  "Lesotho"
##  [75] "Liberia"                  "Libya"
##  [77] "Madagascar"               "Malawi"
##  [79] "Malaysia"                 "Mali"
##  [81] "Mauritania"               "Mauritius"
##  [83] "Mexico"                   "Mongolia"
##  [85] "Montenegro"               "Morocco"
##  [87] "Mozambique"               "Myanmar"
##  [89] "Namibia"                  "Nepal"
##  [91] "Netherlands"              "New Zealand"
##  [93] "Nicaragua"                "Niger"
```

```
##  [95] "Nigeria"                 "Norway"
##  [97] "Oman"                    "Pakistan"
##  [99] "Panama"                  "Paraguay"
## [101] "Peru"                    "Philippines"
## [103] "Poland"                  "Portugal"
## [105] "Puerto Rico"             "Reunion"
## [107] "Romania"                 "Rwanda"
## [109] "Sao Tome and Principe"   "Saudi Arabia"
## [111] "Senegal"                 "Serbia"
## [113] "Sierra Leone"            "Singapore"
## [115] "Slovak Republic"         "Slovenia"
## [117] "Somalia"                 "South Africa"
## [119] "Spain"                   "Sri Lanka"
## [121] "Sudan"                   "Swaziland"
## [123] "Sweden"                  "Switzerland"
## [125] "Syria"                   "Taiwan"
## [127] "Tanzania"                "Thailand"
## [129] "Togo"                    "Trinidad and Tobago"
## [131] "Tunisia"                 "Turkey"
## [133] "Uganda"                  "United Kingdom"
## [135] "United States"           "Uruguay"
## [137] "Venezuela"               "Vietnam"
## [139] "West Bank and Gaza"      "Yemen Rep."
## [141] "Zambia"                  "Zimbabwe"
```

```r
# Filter to just one county for now
us <- filter(gapminder, country == "United States")
```

Your Turn

Let's learn about this data! Answer the following:

1. What's the year span represented in the country you chose?
2. What does each row mean? Is this "long" or "wide" data?

Your Turn - Bonus

Debugging is a large part of learning to code. Fix the following pieces of incorrect code! (Note - change eval = T so that the chunk runs when knitting the doc)

```r
# Fix the following pieces of incorrect code!
summary(gapmider)

read.cvs("gapminder.csv")

filter(gapminder, county = "United States")

ggplot(gapminder, aes(x = age)) %>%
  geom_histogram()
```

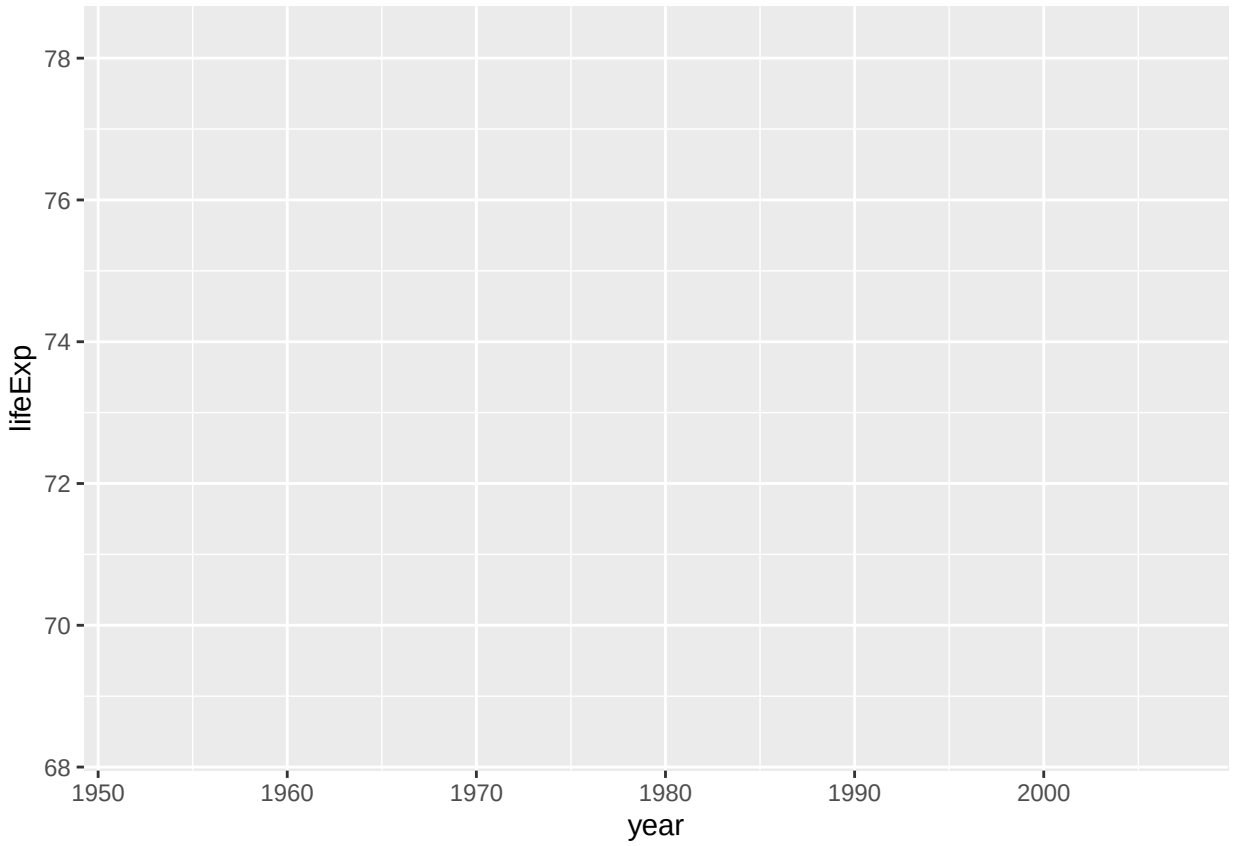
Visualizing our data with ggplot2

We'll start by looking at the trend in life expectancy for one country in our dataset before moving to multiple countries.

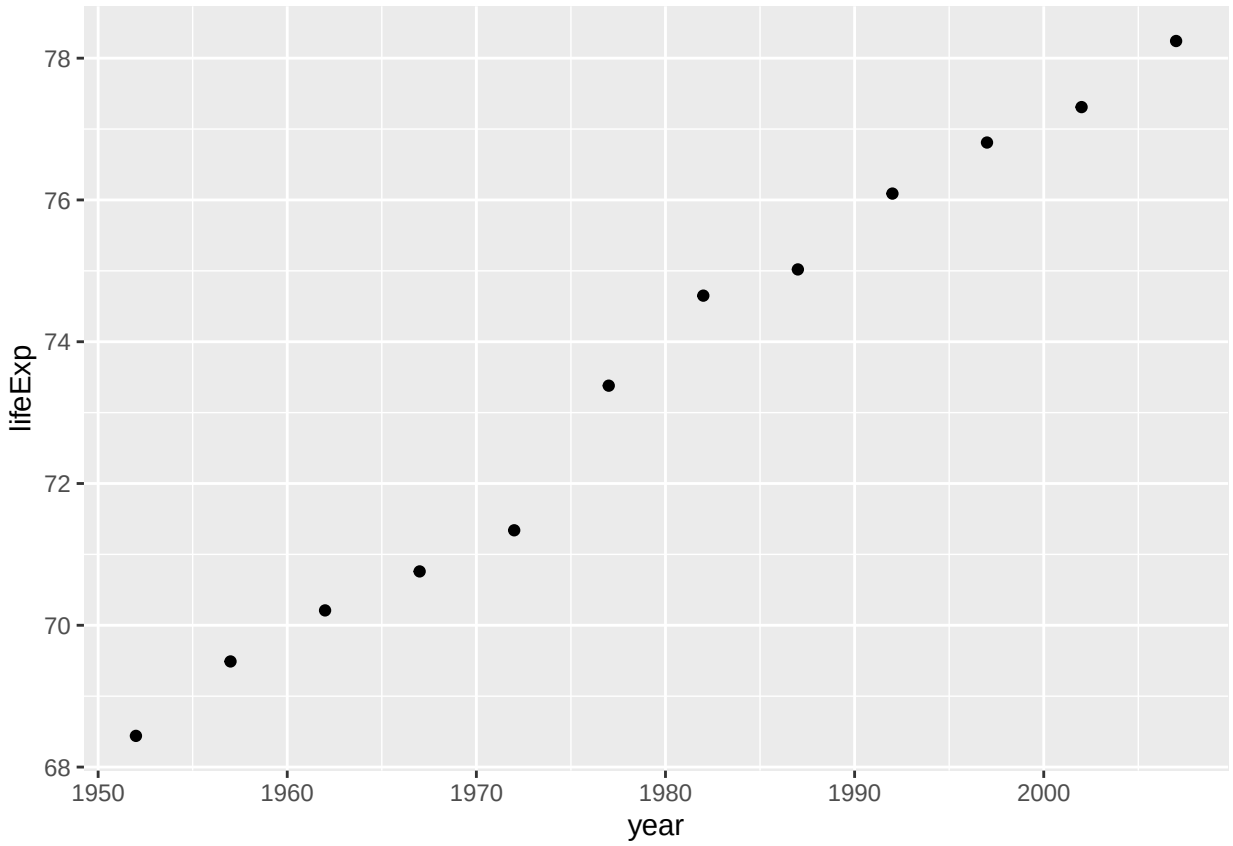
```
# We start by specifying our data
ggplot(data = us)
```



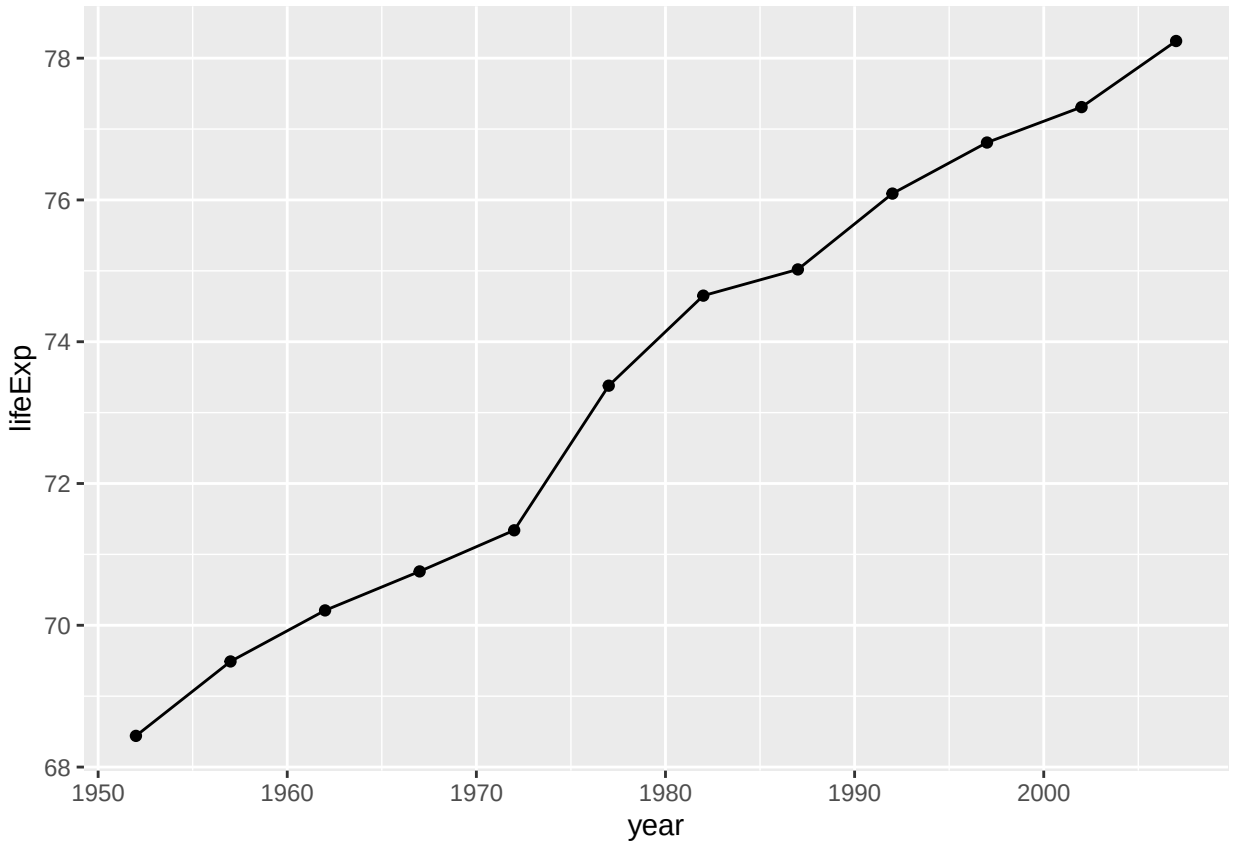
```
# We then map our *aes*thetics to axes
ggplot(data = us, aes(x = year, y = lifeExp))
```



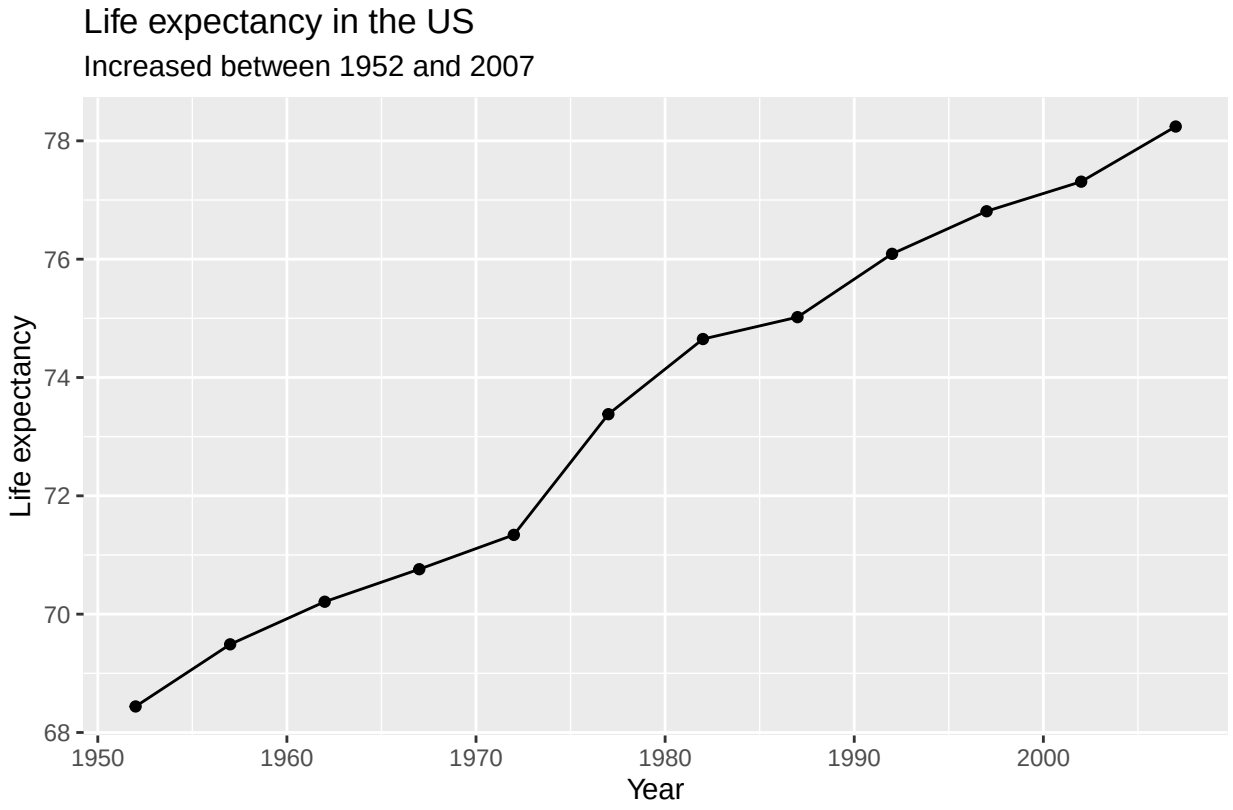

```
# Then add the **geom**etry (or geometries) - note the plus sign!
ggplot(data = us, aes(x = year, y = lifeExp)) +
  geom_point()
```



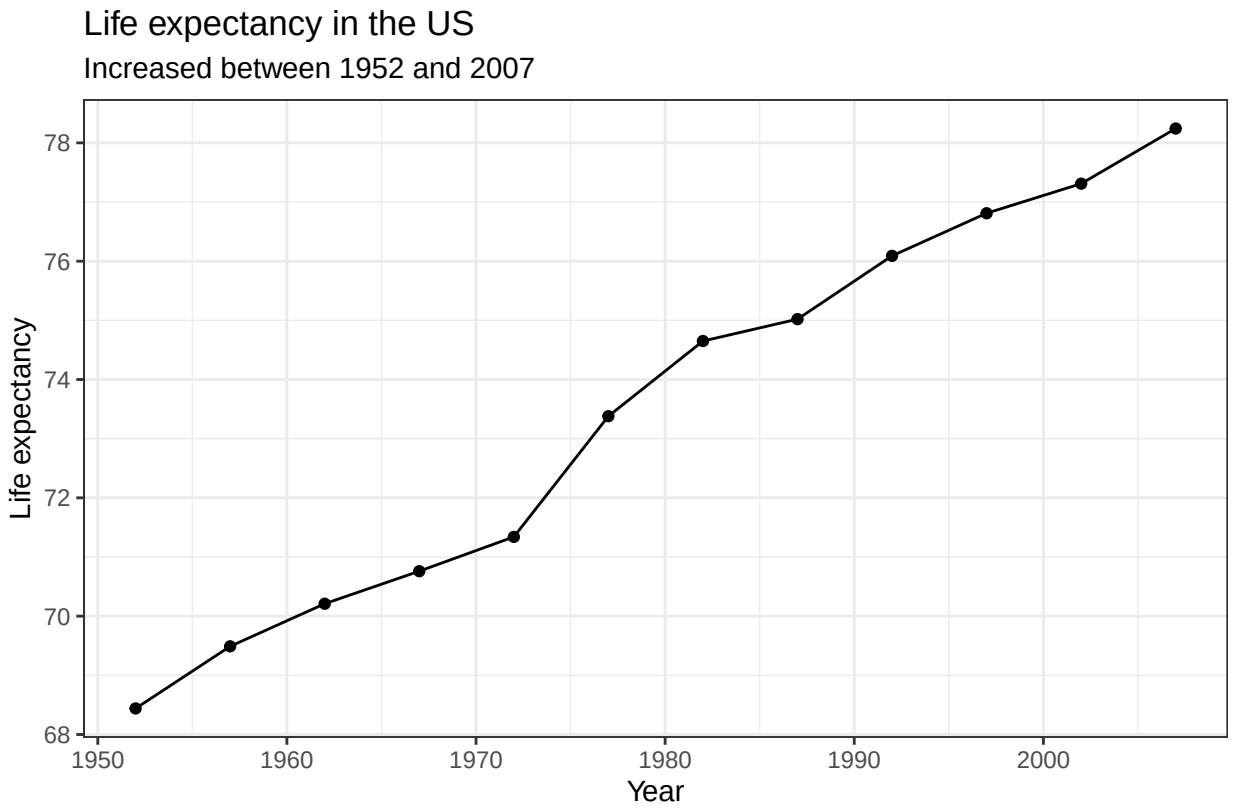

```
ggplot(data = us, aes(x = year, y = lifeExp)) +
  geom_point() +
  geom_line()
```



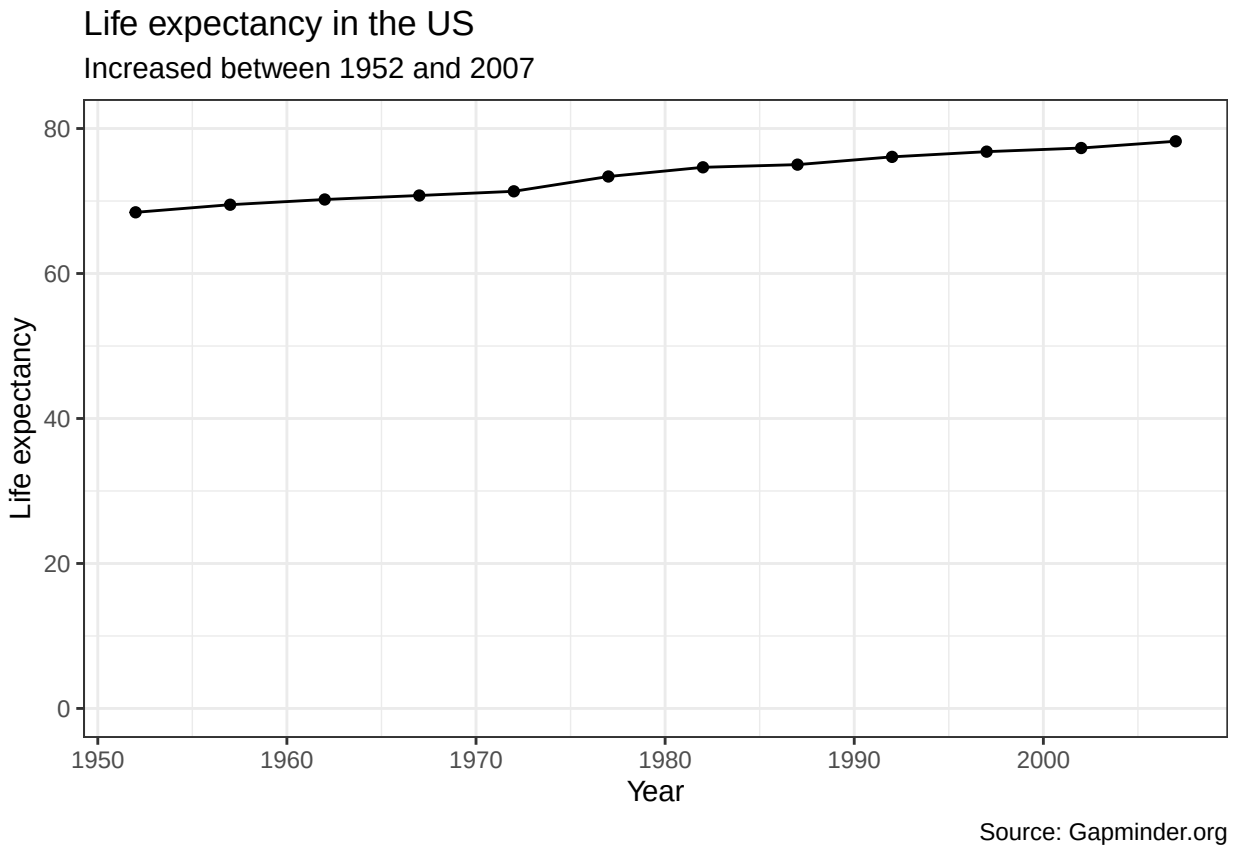

```
# Add informative labels (and maybe a subtitle or caption)
ggplot(data = us, aes(x = year, y = lifeExp)) +
  geom_point() +
  geom_line() +
  labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy in the US",
       subtitle = "Increased between 1952 and 2007",
       caption = "Source: Gapminder.org")
```




```
# Adjust the theme
ggplot(data = us, aes(x = year, y = lifeExp)) +
  geom_point() +
  geom_line() +
  labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy in the US",
       subtitle = "Increased between 1952 and 2007",
       caption = "Source: Gapminder.org") +
  theme_bw()
```



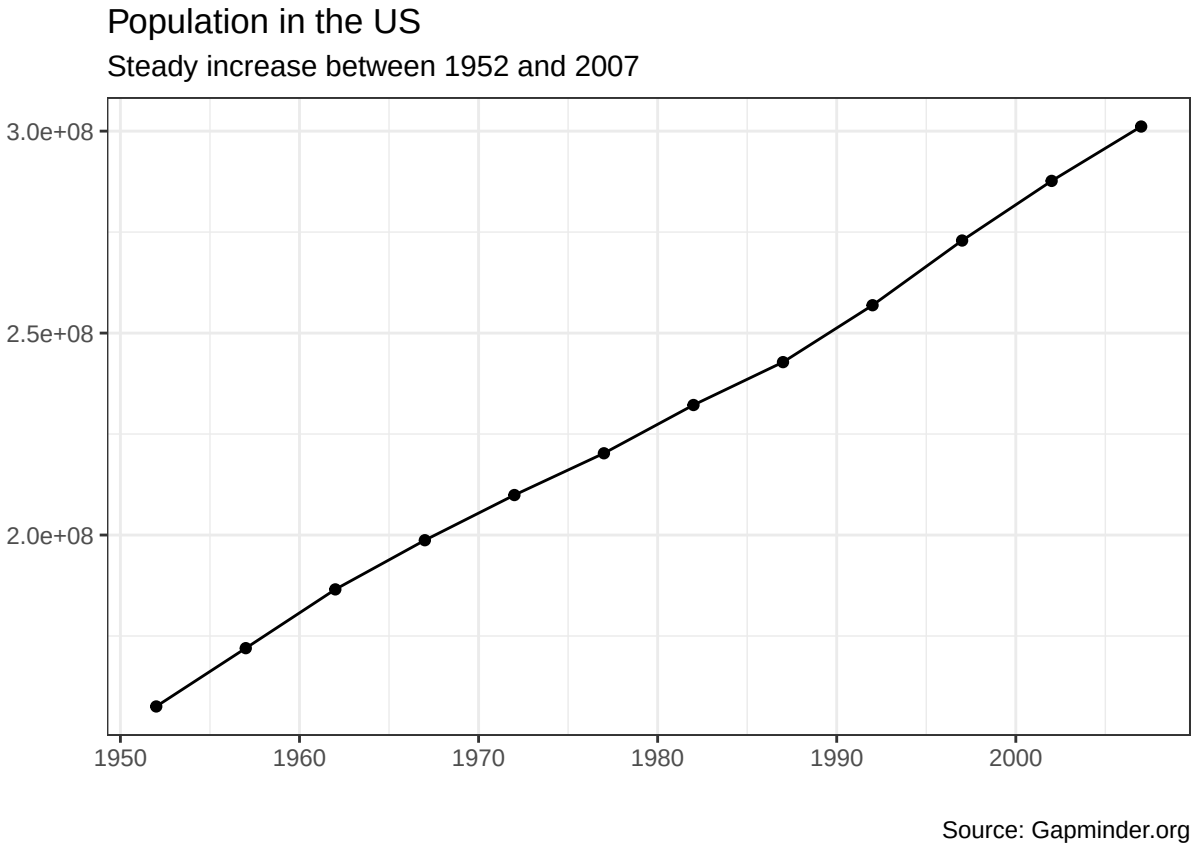

```
# Adjust limits
ggplot(data = us, aes(x = year, y = lifeExp)) +
  geom_point() +
  geom_line() +
  labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy in the US",
       subtitle = "Increased between 1952 and 2007",
       caption = "Source: Gapminder.org") +
  theme_bw() +
  ylim(0, 80)
```



Your Turn

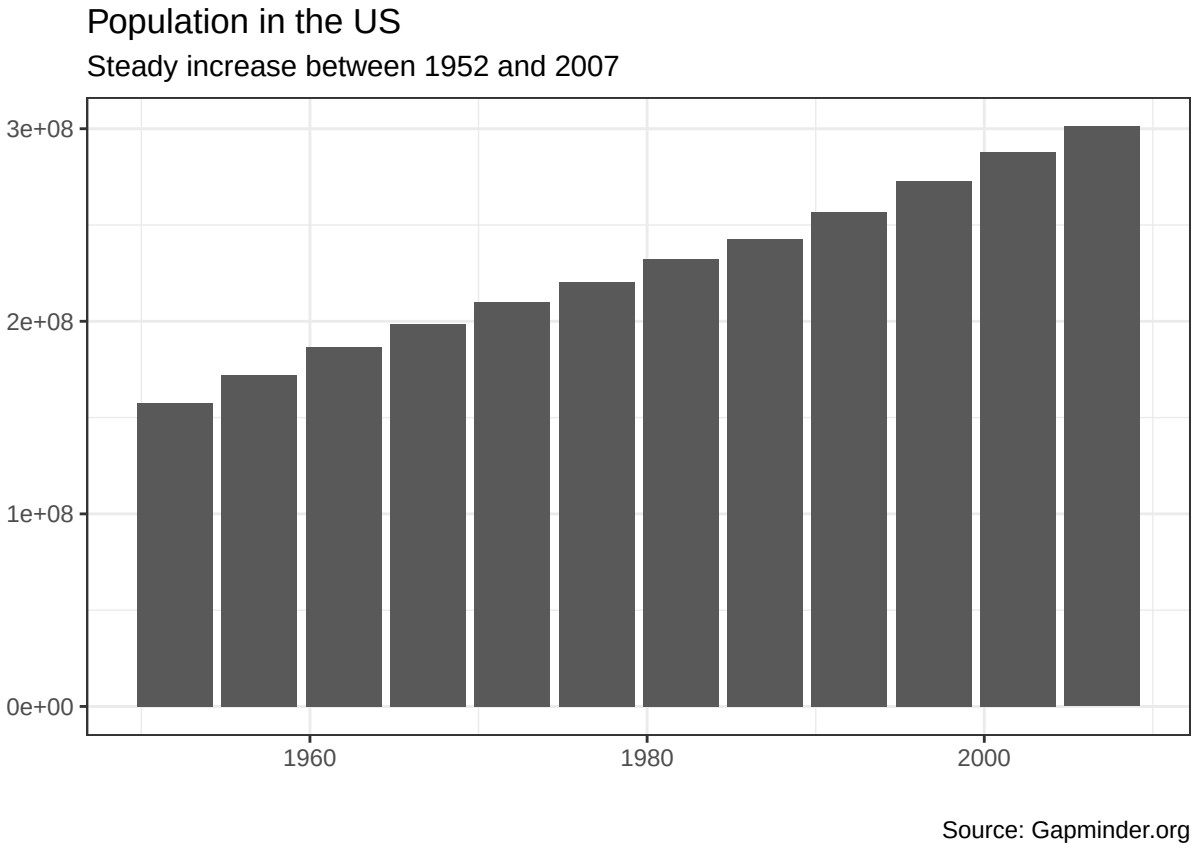
Visualize the change in **population** for your country of interest using ggplot2.

```
ggplot(data = us, aes(x = year, y = pop)) +
  geom_point() +
  geom_line() +
  labs(x = "",
       y = "",
       title = "Population in the US",
       subtitle = "Steady increase between 1952 and 2007",
       caption = "Source: Gapminder.org") +
  theme_bw()
```



Try making a bar chart instead of a line graph of population (hint: use `geom_col`). Is this the best visualization for this data? Why not?

```
ggplot(data = us, aes(x = year, y = pop)) +
  geom_col() +
  labs(x = "",
       y = "",
       title = "Population in the US",
       subtitle = "Steady increase between 1952 and 2007",
       caption = "Source: Gapminder.org") +
  theme_bw()
```



Plotting multiple countries

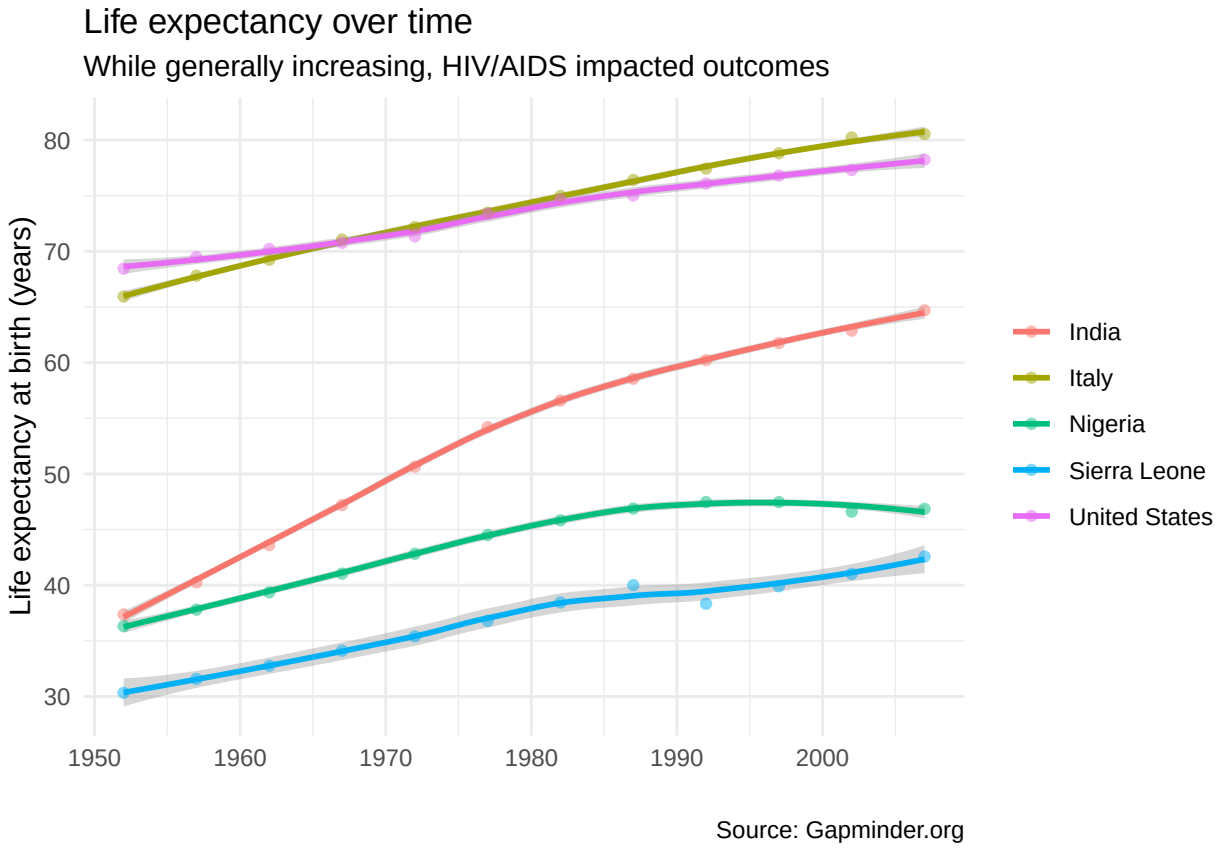
We want to compare multiple countries, so let's include 5 from multiple continents in our dataset.

```
gap5 <- gapminder %>%
  filter(country %in% c("Sierra Leone", "United States", "Italy", "Nigeria", "India"))
```

Then, visualize the subset of the data:

```
ggplot(data = gap5, aes(x = year, y = lifeExp, color = country)) +
  geom_smooth() +
  geom_point(alpha = 0.5) + # alpha makes points transparent
  labs(x = "",
       y = "Life expectancy at birth (years)",
       title = "Life expectancy over time",
       subtitle = "While generally increasing, HIV/AIDS impacted outcomes",
       caption = "Source: Gapminder.org") +
  theme_minimal() +
  theme(legend.title = element_blank()) +
  guides(color=guide_legend(override.aes=list(fill=NA)))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Your Turn

Create a plot for **population** over time for 5 countries in the dataset using ggplot2.

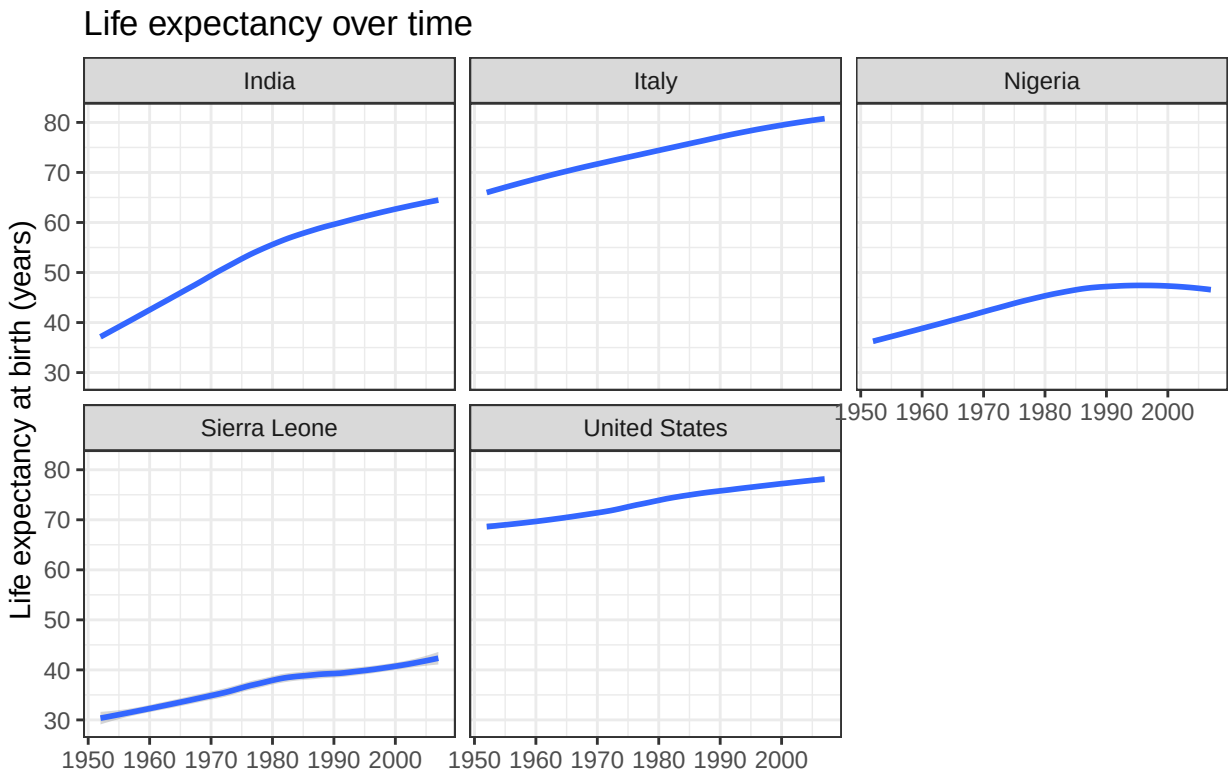
Your Turn - Bonus

You can also create small multiples of graphs, where each country is represented in its own graph.

Poke around the ggplot2 book on facetting and see if you can figure out how to make "small multiples" for the trend in each country.

```
ggplot(data = gap5, aes(x = year, y = lifeExp)) +
  geom_smooth() +
  labs(x = "",
       y = "Life expectancy at birth (years)",
       title = "Life expectancy over time",
       caption = "Source: Gapminder.org") +
  theme(legend.title = element_blank()) +
  theme_bw() +
  guides(color=guide_legend(override.aes=list(fill=NA))) +
  facet_wrap(~country)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

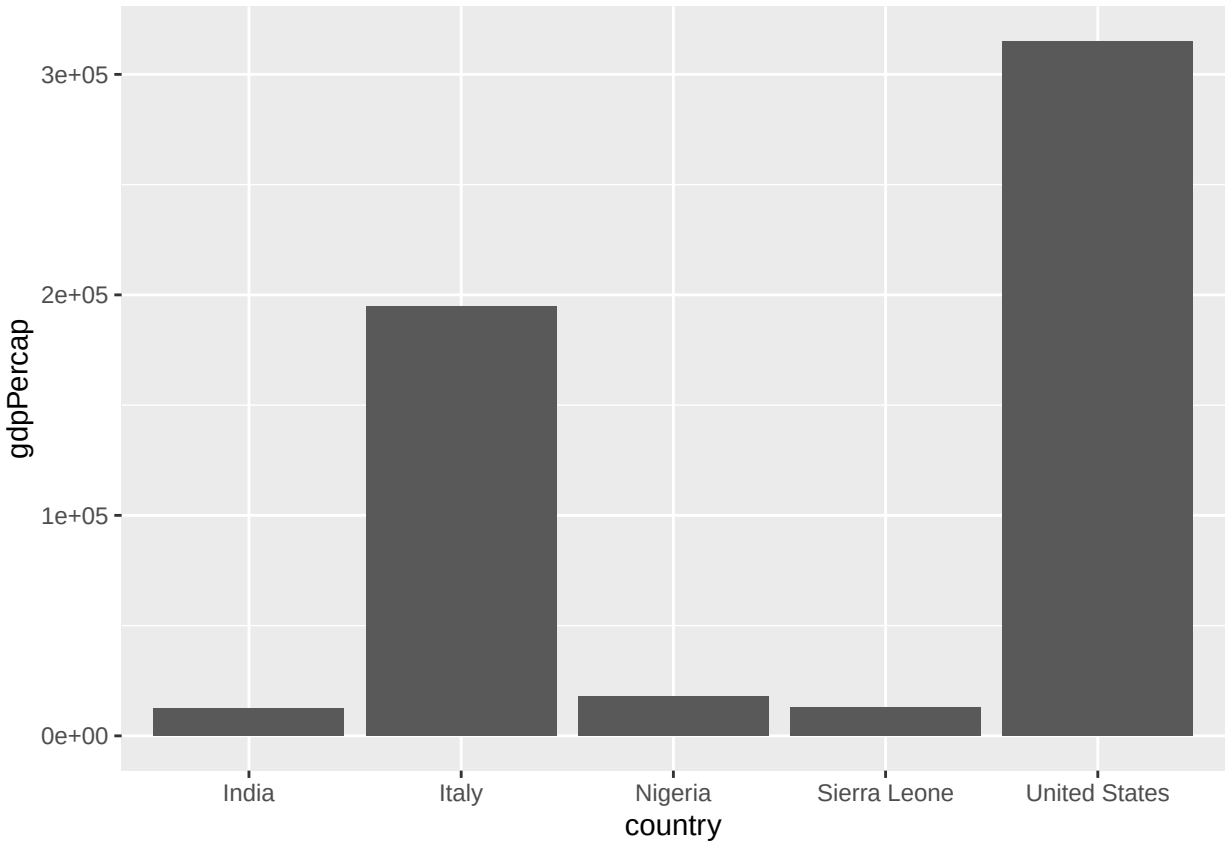


Source: Gapminder.org

Bar Charts

Now, we will subset to a given year and compare an outcome of our choice in 5 countries.

```
gap5 %>%
  ggplot(aes(x = country, y = gdpPercap)) +
  geom_col()
```

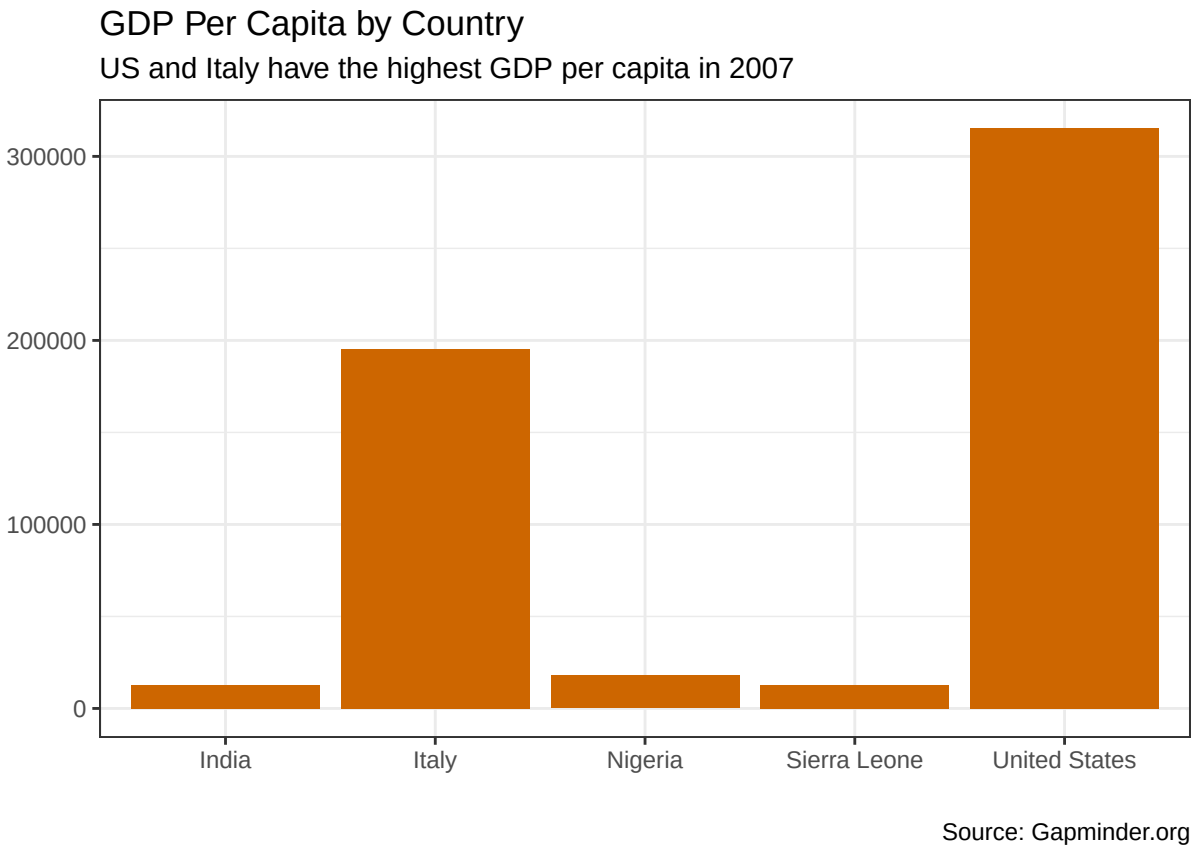


Your Turn

Improve on the plot we created to make it more engaging and informative to the viewer. If you want, try the `colors()` function in the R console to see the color names available. The R Graph Gallery also provides guidance.

```
options(scipen = 999)

gap5 %>%
  ggplot(aes(x = country, y = gdpPercap)) +
  geom_col(fill = "darkorange3") + # Tr
  labs(x = "",
       y = "",
       title = "GDP Per Capita by Country",
       subtitle = "US and Italy have the highest GDP per capita in 2007",
       caption = "Source: Gapminder.org") +
  theme_bw()
```



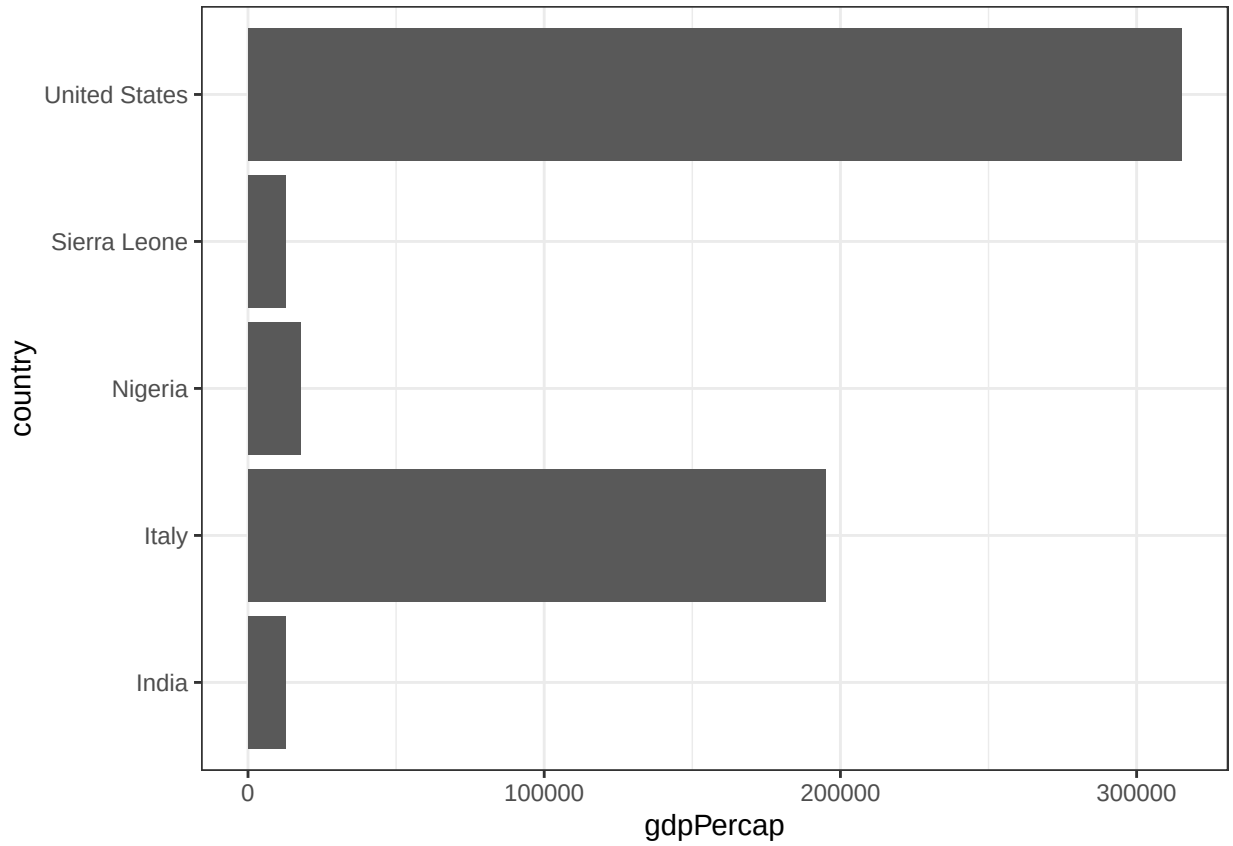
Your Turn - Bonus

The figure defaults in R don't order the bars in a meaningful way. There are two tips that will help with this:

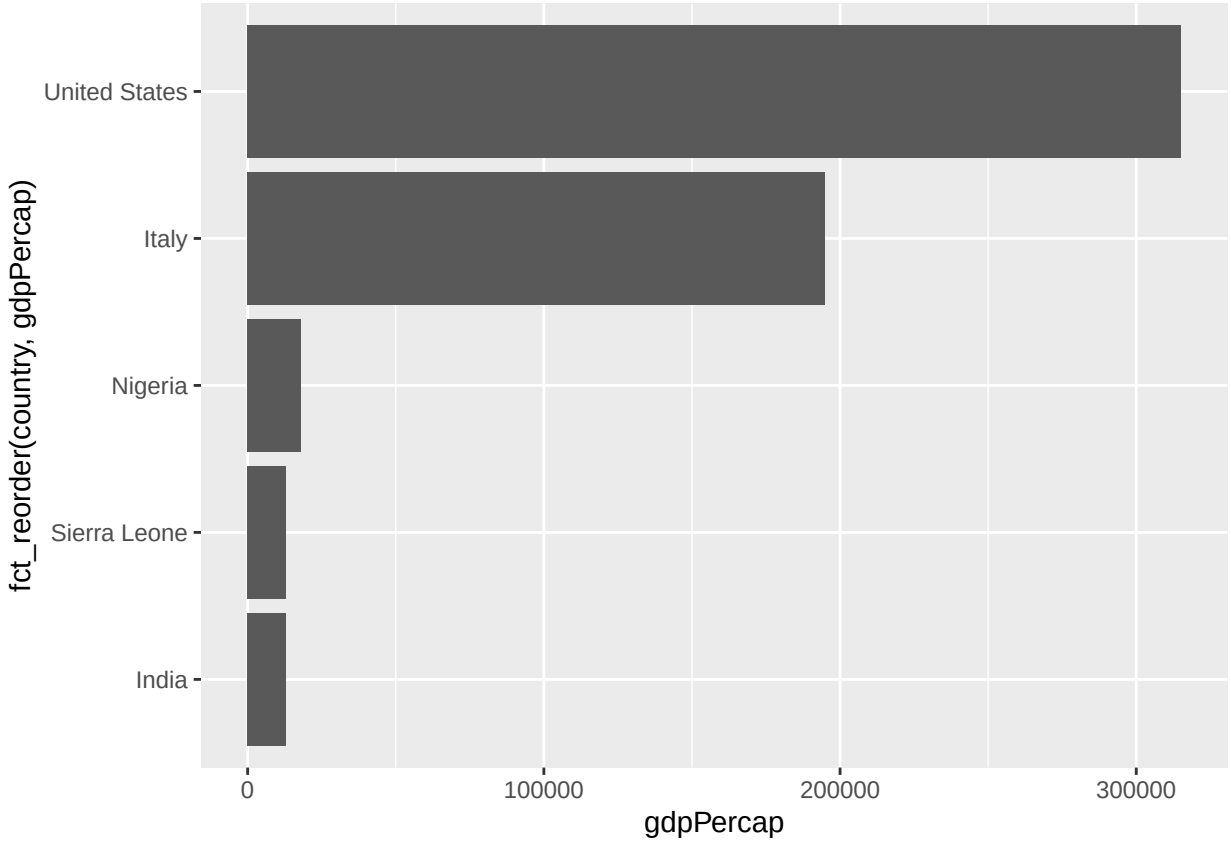
1. Flipping the x and y axes: `+ coord_flip()`
2. Ordering bars in a meaningful way, ie by the levels, not alphabetically `x = fct_reorder(<things to be reordered>, <thing to reorder it by>)`

Try implementing this to reorder the bars!

```
gap5 %>%
  ggplot(aes(x = country, y = gdpPercap)) +
  geom_col() +
  coord_flip() +
  theme_bw()
```




```
gap5 %>%
  ggplot(aes(x = fct_reorder(country, gdpPercap), y = gdpPercap)) +
  geom_col() +
  coord_flip()
```



There's much more to ggplot2, but for that, I direct you to some much better guides that have been written. . .

Your Turn: Explore Open Source Resources

Take 5 minutes and look through the resources listed below. Which one of these look the most useful for your work?

- Posit Cloud Primers - Visualize Data: Interactive online tutorials to learn data viz in R, from the makers of RStudio
- R Graphics Cookbook, 2nd edition: Gives you recipes for creating various graphs in R - https://r-graphics.org/
- ggplot2 book: Nitty gritty on how ggplot2 all works, from the creator of ggplot2
-
- Fundamentals of Data Visualization: Less specific to ggplot2, more on best practices to learn data viz
- Data Visualization course by Andrew Heiss: Designed as an entire course in R for policy school students

Your Turn - Bonus

Schedule in an hour on your calendar (possibly over break) to explore that resource.