Lab 5 Graphing Data (Part 2)

B. Sosnovski

10/18/2021

Graphing Data (Part 2)

This is a continuation of the practice of the skills presented in Lab 4, Graphing Data.

General goals

Our general goals for this lab are to do the following:

1. Load some data into R
2. Make graphs of the quantitative data.

R

Gapminder data

https://www.gapminder.org (https://www.gapminder.org) is an organization that collects some interesting worldwide data. They also make cool visualization tools for looking at the data. There are many good examples, and they have visualization tools built right into their website that you can play around with https://www.gapminder.org/tools/ (https://www.gapminder.org/tools/). That's fun; check it out.

Load the data to R

The data set needed for this lab is included together with these instructions.

Use the following commands to load the data.

```
library("gapminder")
gapminder_df<-gapminder
```

If you haven't installed `gapminder` in your project, run the code below.

```
library(data.table)
gapminder_df <-fread("gapminder.csv")
```

Look at the data frame from gapminder

You can look at the data to see what is in it.

```
colnames(gapminder_df)
```

```
## [1] "country"   "continent" "year"      "lifeExp"   "pop"       "gdpPercap"
```

Let's check how big is this data frame.

```
data_size <- dim(gapminder_df)
data_size
```

```
## [1] 1704    6
```

There are 1704 rows of data and 6 variables. We see columns for the country, continent, year, life expectancy, population, and GDP per capita.

Example

Enter a code chunk below that displays the data's first and last rows in gapminder.

```
 Enter your code below:
```

```
head(gapminder_df)
```

```
## # A tibble: 6 × 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.
```

```
tail(gapminder_df)
```

```
## # A tibble: 6 × 6
##   country  continent  year lifeExp      pop gdpPercap
##   <fct>    <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Zimbabwe Africa     1982    60.4  7636524      789.
## 2 Zimbabwe Africa     1987    62.4  9216418      706.
## 3 Zimbabwe Africa     1992    60.4 10704340      693.
## 4 Zimbabwe Africa     1997    46.8 11404948      792.
## 5 Zimbabwe Africa     2002    40.0 11926563      672.
## 6 Zimbabwe Africa     2007    43.5 12311143      470.
```

Asking Questions with the gapminder data

We will show you how to graph some of the data to answer a few different kinds of questions. Then you will form your questions and see if you can answer them with ggplot2 yourself. You will need to copy and paste the following examples and change them up a little bit.

Life expectancy histogram

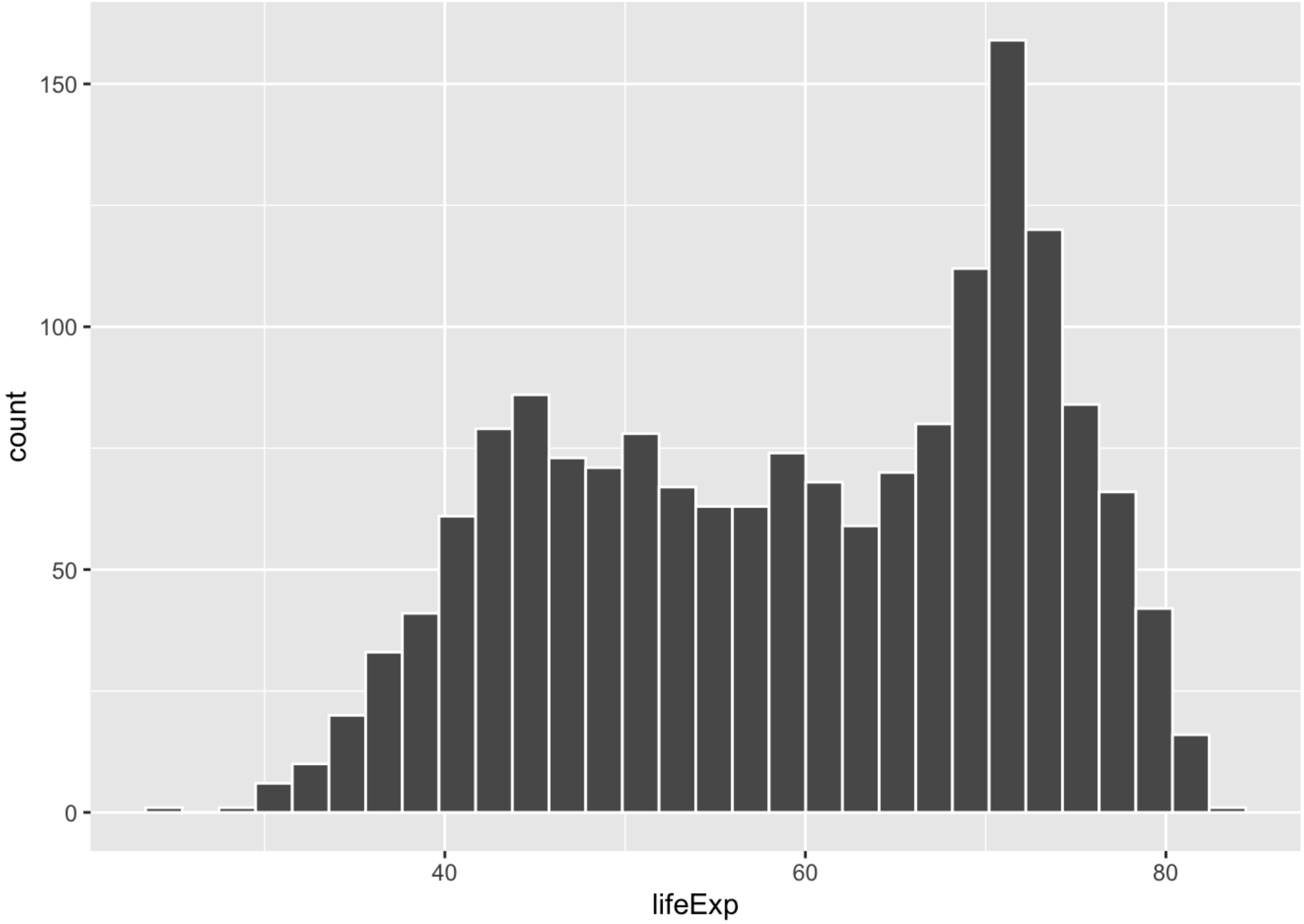
How long are people living all around the world according to this data set?

There are many ways we could plot the data to find out. The first way is a histogram. We have many numbers for life expectancy in the column `lifeExp` . This is a big sample, full of numbers for 142 countries across many years.

It’s easy to make a histogram in ggplot to view the distribution. Most of the code should be familiar to you since you already encountered most of it in lab 3.

```
library(ggplot2)
ggplot(gapminder_df, aes(x = lifeExp))+
  geom_histogram(color="white")
```

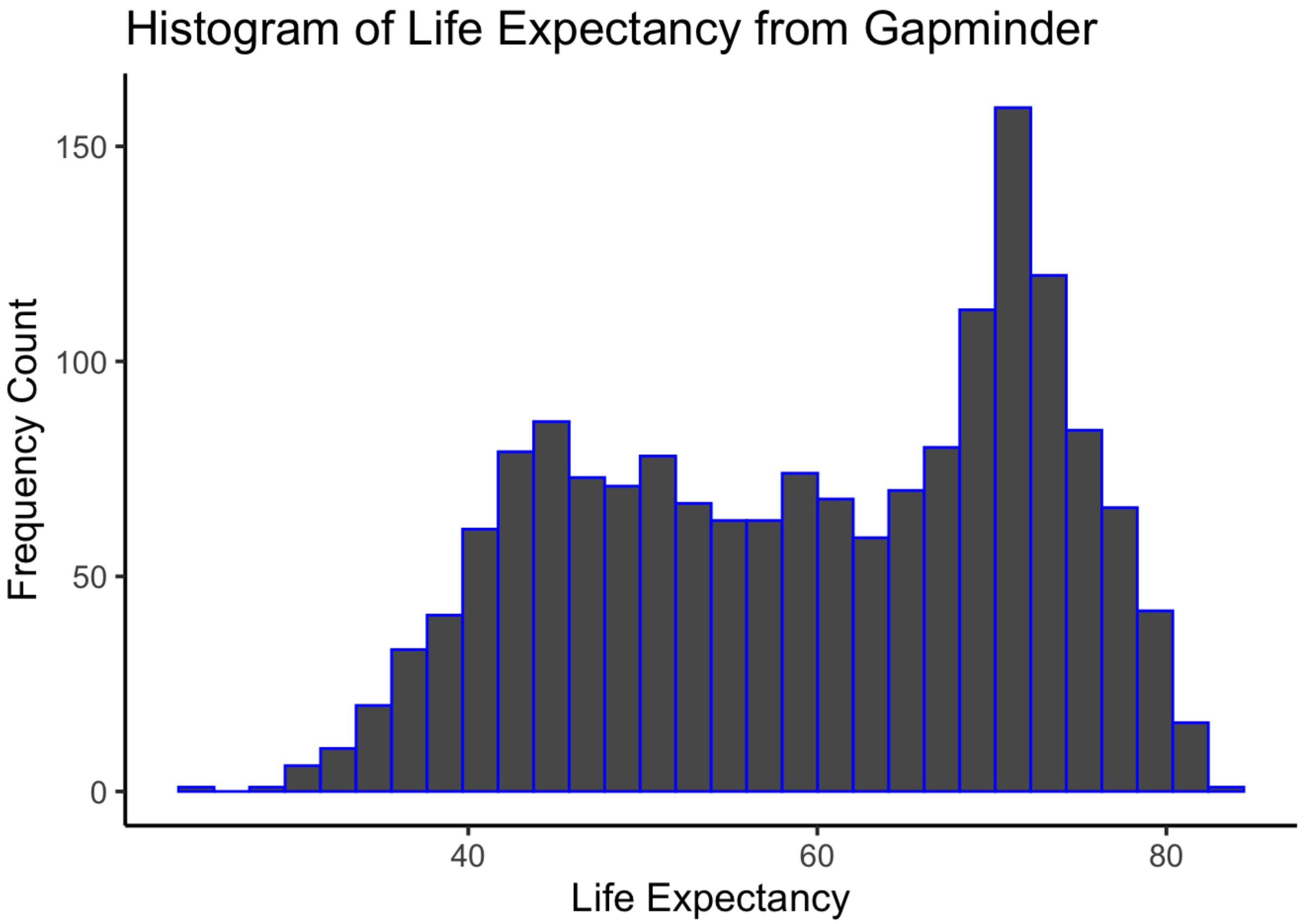
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



See, that was easy. Next is a code chunk that adds more layers and settings if you want to modify parts of the graph:

```
ggplot(gapminder_df, aes(x=lifeExp)) +
  geom_histogram(color="blue")+
  theme_classic(base_size = 15) +
  xlab("Life Expectancy") +
  ylab("Frequency Count") +
  ggtitle("Histogram of Life Expectancy from Gapminder")
```

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

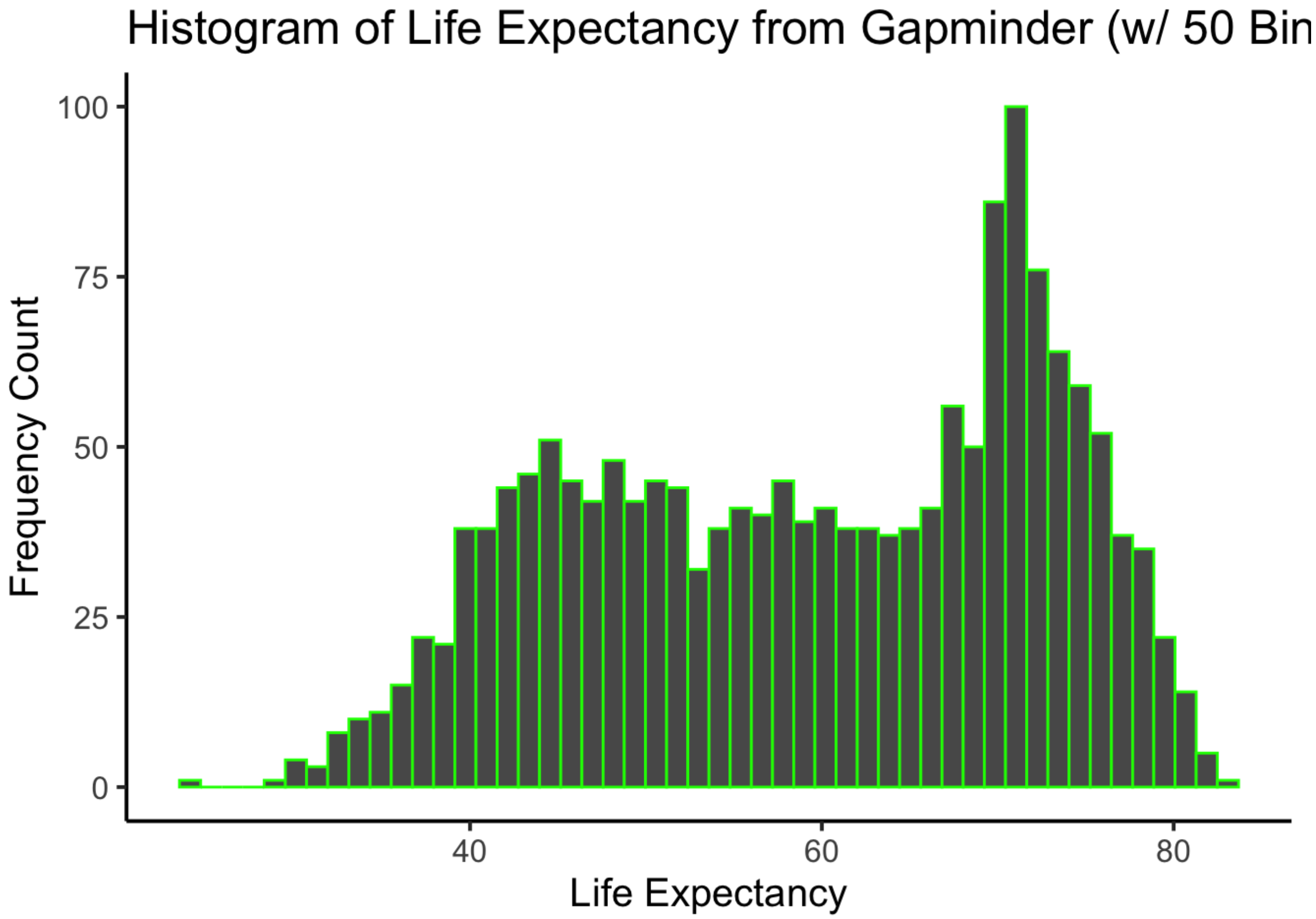


Did you notice the changes from the examples presented in lab 3?

The histogram shows a wide range of life expectancies, from below 40 to just over 80. Histograms are helpful; they show you what values happen more often than others.

One final thing about histograms in ggplot is that you may want to change the bin size. That controls how wide or narrow, or the number of bars (how they split across the range) in the histogram. You must set the `bins=` option in `geom_histogram()` .

```
ggplot(gapminder_df, aes(x = lifeExp)) +
  geom_histogram(color="green", bins=50)+
  theme_classic(base_size = 15) +
  xlab("Life Expectancy") +
  ylab("Frequency Count") +
  ggtitle("Histogram of Life Expectancy from Gapminder (w/ 50 Bins)")
```

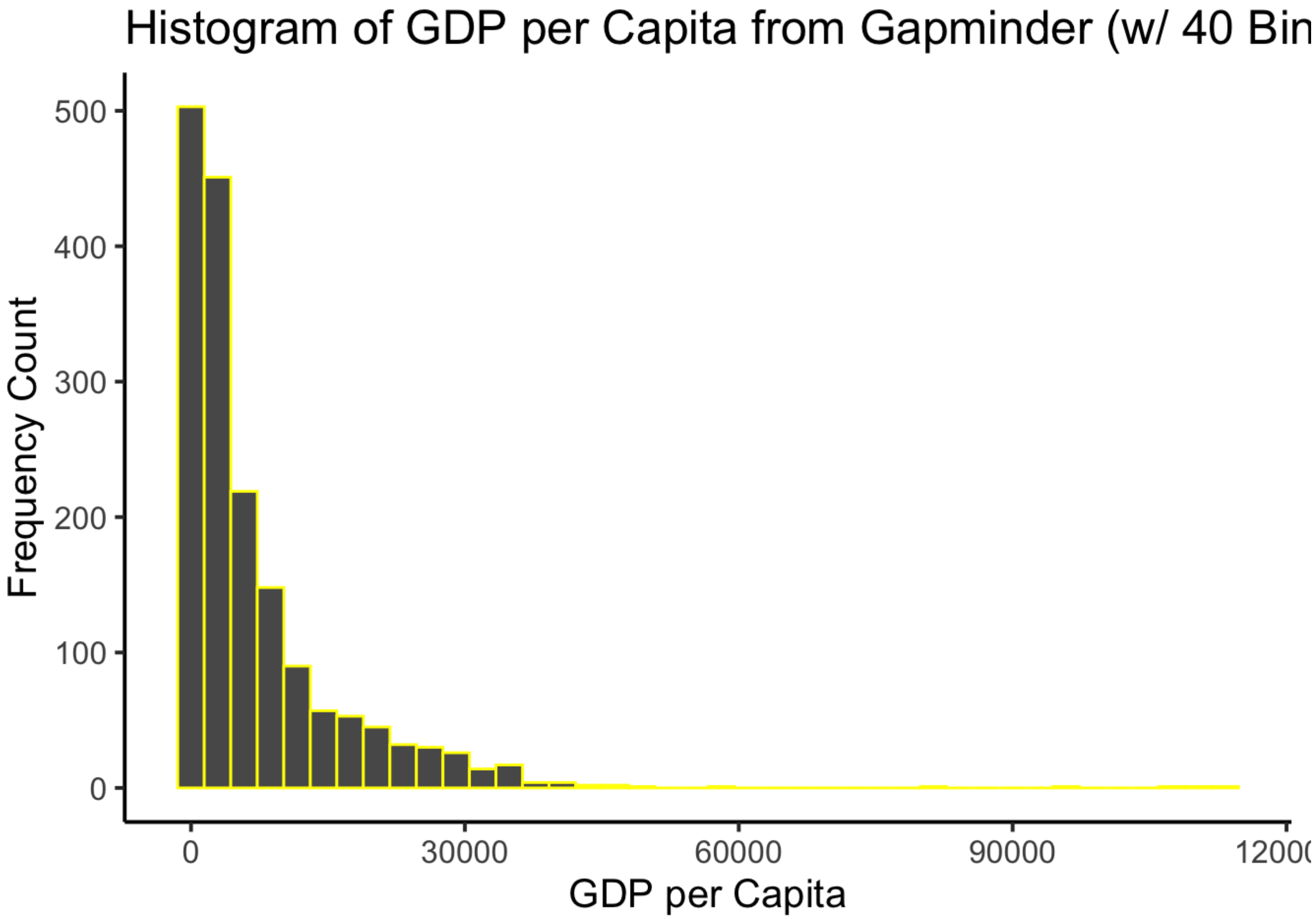


Example

Use the code chunk above to create a histogram with different colors and 40 bins displaying the variable corresponding to the GDP per Capita variable in `gapminder`.

```
Enter your code below:
```

```
ggplot(gapminder_df, aes(x = gdpPercap)) +
  geom_histogram(color="yellow", bins=40)+
  theme_classic(base_size = 15) +
  xlab("GDP per Capita") +
  ylab("Frequency Count") +
  ggtitle("Histogram of GDP per Capita from Gapminder (w/ 40 Bins)")
```



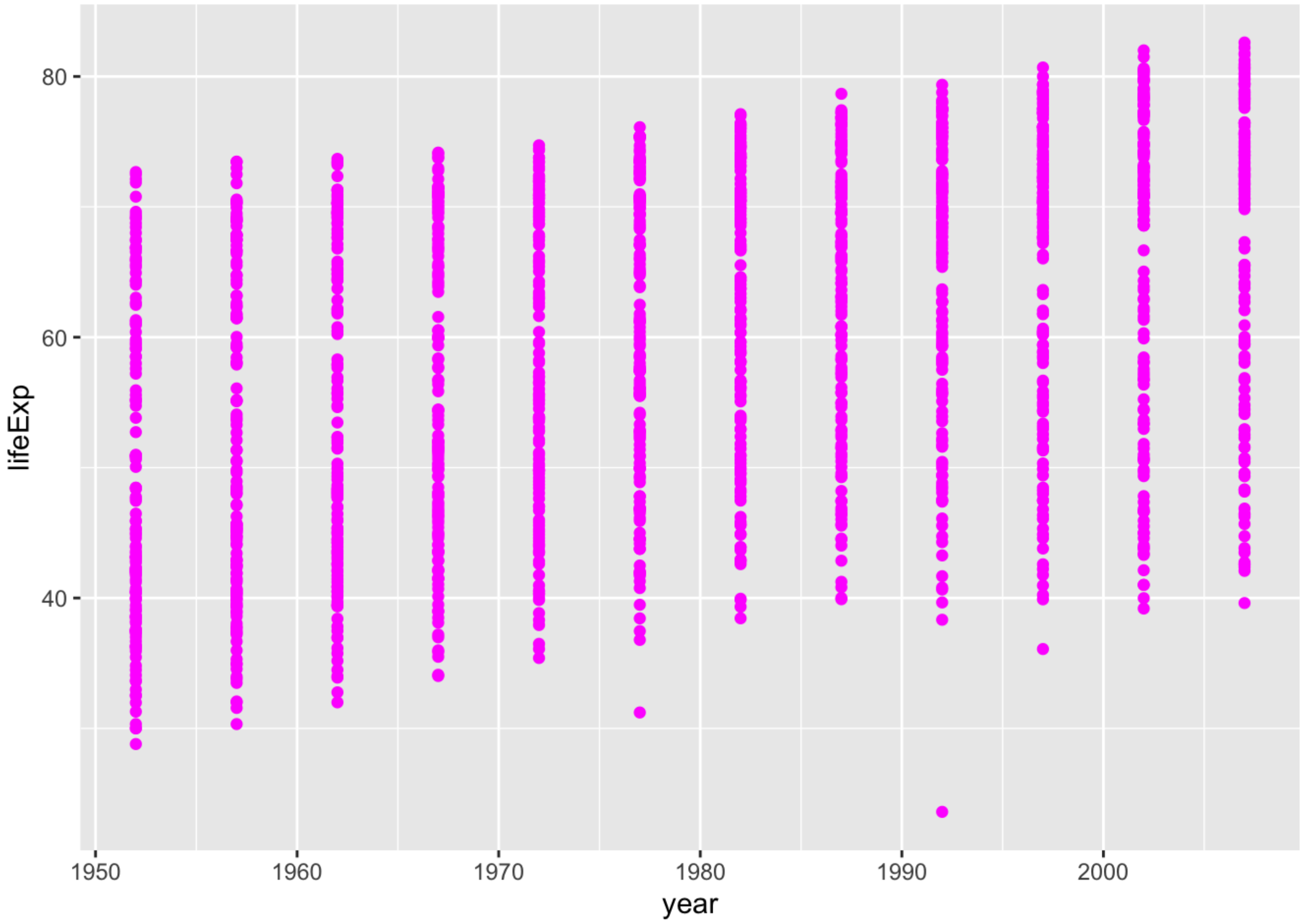
Note: See, same basic shape, but now breaking up the range into 40 little equal-sized bins, rather than 30, which is the default. You get to choose what you want to do.

Life expectancy by year scatterplot

We can see that we have data for life expectancy and different years. So, does worldwide life expectancy change across the years in the data set? As we go into the future, are people living longer?

Let’s look at this by using a scatter plot. We can set the x-axis as the years and the y-axis as life expectancy. Then we can use `geom_point()` to display a whole bunch of dots and look at them. Here’s the simple code:

```
ggplot(gapminder_df, aes( x= year, y= lifeExp))+
  geom_point(color="magenta")
```



Whoa, that's a lot of dots! Remember that each country is measured each year. So, the bands of dots you see show the life expectancies for the whole range of countries within each year of the database. There is a big spread inside each year. But it looks like groups of dots slowly go up over the years.

One country's life expectancy by year

I'm (B. Sosnovski), born and raised in Brazil, so maybe I want to know if life expectancy for Brazilians has been going up over the years. To find out the answer for one country, we first need to split the complete data set into another smaller one that only contains Brazil data. In other words, we want only the rows where the word "Brazil" is found in the `country` column. We will use the `filter` function from `dplyr` for this:

```
# filter rows to contain Brazil
library(dplyr)
```

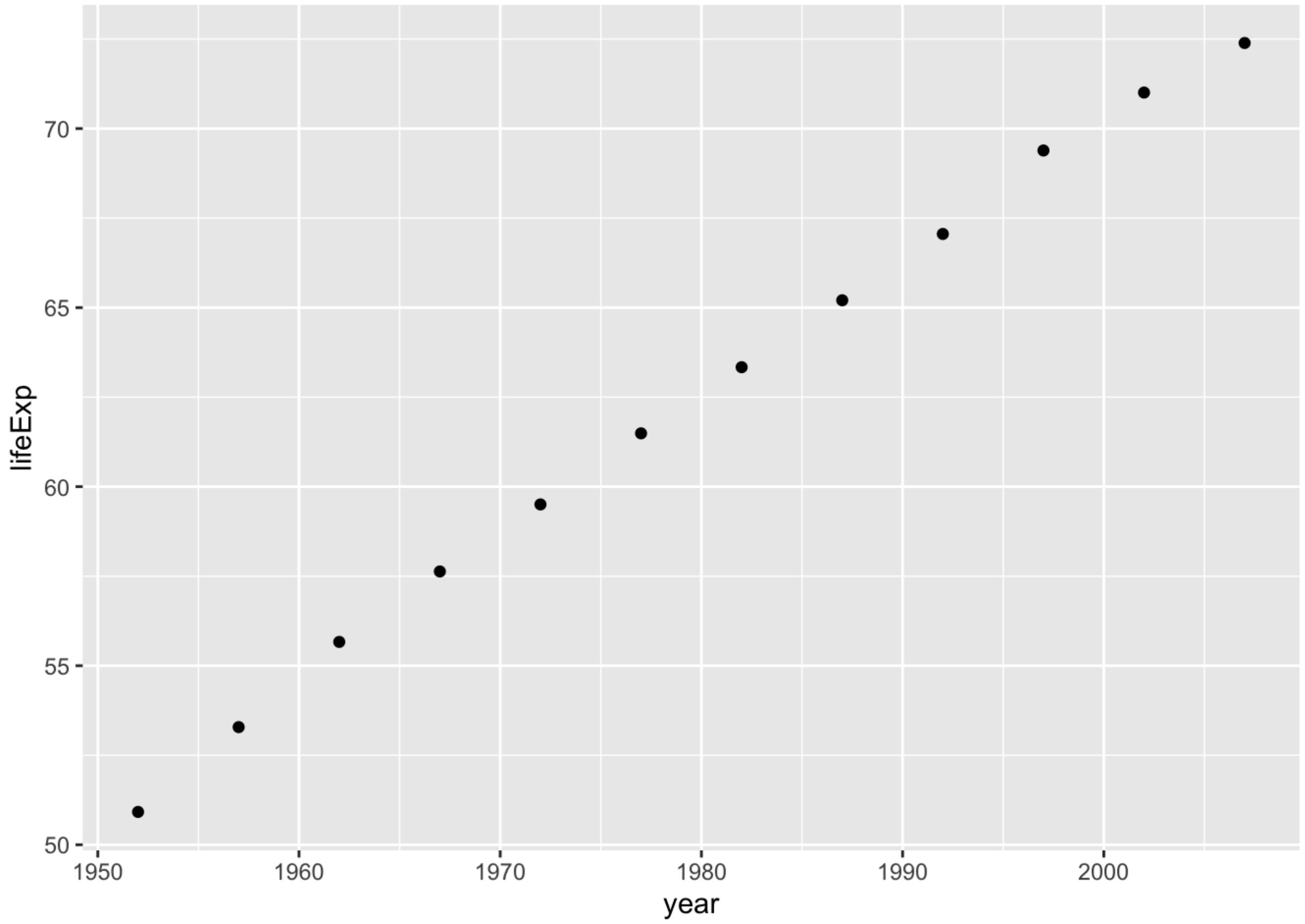
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
smaller_df <- gapminder_df %>%
                 filter(country == "Brazil")
head(smaller_df)
```

```
## # A tibble: 6 × 6
##   country continent  year lifeExp       pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Brazil  Americas   1952    50.9  56602560     2109.
## 2 Brazil  Americas   1957    53.3  65551171     2487.
## 3 Brazil  Americas   1962    55.7  76039390     3337.
## 4 Brazil  Americas   1967    57.6  88049823     3430.
## 5 Brazil  Americas   1972    59.5 100840058     4986.
## 6 Brazil  Americas   1977    61.5 114313951     6660.
```

```
# plot the new data contained in smaller_df
ggplot(smaller_df, aes( x= year, y= lifeExp))+
  geom_point(color="black")
```



I would say things are looking good for Brazilians 🙏 ; their life expectancy has increased over the years!

Example

Enter a code chunk below that shows the life expectance for another country of your choice. Also, add a title and adjust labels for horizontal and vertical axes.

- Here is a list of the countries in the data frame:

```
listOfCountries <- unique(gapminder_df[,1])
listOfCountries$country
```

```
##   [1] Afghanistan              Albania                  Algeria                 
##   [4] Angola                   Argentina                Australia               
##   [7] Austria                  Bahrain                  Bangladesh              
##  [10] Belgium                  Benin                    Bolivia                 
##  [13] Bosnia and Herzegovina   Botswana                 Brazil                  
##  [16] Bulgaria                 Burkina Faso             Burundi                 
##  [19] Cambodia                 Cameroon                 Canada                  
##  [22] Central African Republic Chad                     Chile                   
##  [25] China                    Colombia                 Comoros                 
##  [28] Congo, Dem. Rep.         Congo, Rep.              Costa Rica              
##  [31] Cote d'Ivoire            Croatia                  Cuba                    
##  [34] Czech Republic           Denmark                  Djibouti                
##  [37] Dominican Republic       Ecuador                  Egypt                   
##  [40] El Salvador              Equatorial Guinea        Eritrea                 
##  [43] Ethiopia                 Finland                  France                  
##  [46] Gabon                    Gambia                   Germany                 
##  [49] Ghana                    Greece                   Guatemala               
##  [52] Guinea                   Guinea-Bissau            Haiti                   
##  [55] Honduras                 Hong Kong, China         Hungary                 
##  [58] Iceland                  India                    Indonesia               
##  [61] Iran                     Iraq                     Ireland                 
##  [64] Israel                   Italy                    Jamaica                 
##  [67] Japan                    Jordan                   Kenya                   
##  [70] Korea, Dem. Rep.         Korea, Rep.              Kuwait                  
##  [73] Lebanon                  Lesotho                  Liberia                 
##  [76] Libya                    Madagascar               Malawi                  
##  [79] Malaysia                 Mali                     Mauritania              
##  [82] Mauritius                Mexico                   Mongolia                
##  [85] Montenegro               Morocco                  Mozambique              
##  [88] Myanmar                  Namibia                  Nepal                   
##  [91] Netherlands              New Zealand              Nicaragua               
##  [94] Niger                    Nigeria                  Norway                  
##  [97] Oman                     Pakistan                 Panama                  
## [100] Paraguay                 Peru                     Philippines             
## [103] Poland                   Portugal                 Puerto Rico             
## [106] Reunion                  Romania                  Rwanda                  
## [109] Sao Tome and Principe    Saudi Arabia             Senegal                 
## [112] Serbia                   Sierra Leone             Singapore               
## [115] Slovak Republic          Slovenia                 Somalia                 
## [118] South Africa             Spain                    Sri Lanka               
## [121] Sudan                    Swaziland                Sweden                  
## [124] Switzerland              Syria                    Taiwan                  
## [127] Tanzania                 Thailand                 Togo                    
## [130] Trinidad and Tobago      Tunisia                  Turkey                  
## [133] Uganda                   United Kingdom           United States           
## [136] Uruguay                  Venezuela                Vietnam                 
## [139] West Bank and Gaza       Yemen, Rep.              Zambia                  
## [142] Zimbabwe                
## 142 Levels: Afghanistan Albania Algeria Angola Argentina Australia ... Zimbabwe
```

Enter your code below:

Multiple countries scatterplot

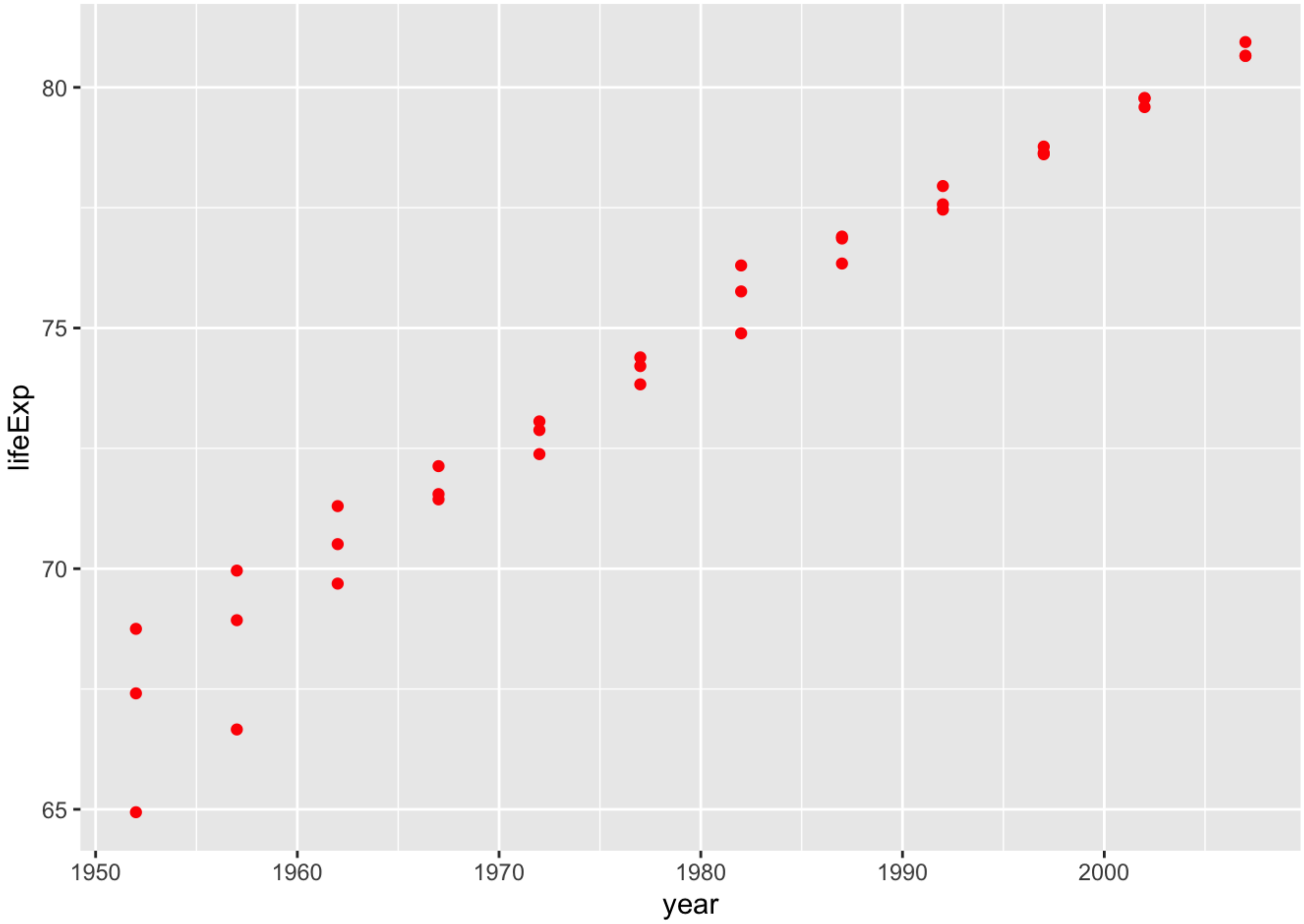
What if we want to look at a few countries altogether?

We can do this too. We change how we filter the data so that more than one country is allowed, then plot the data. We will also add some nicer color options and make the plot look pretty.

First, the simple code that shows the data for the countries Canada, France, and Spain:

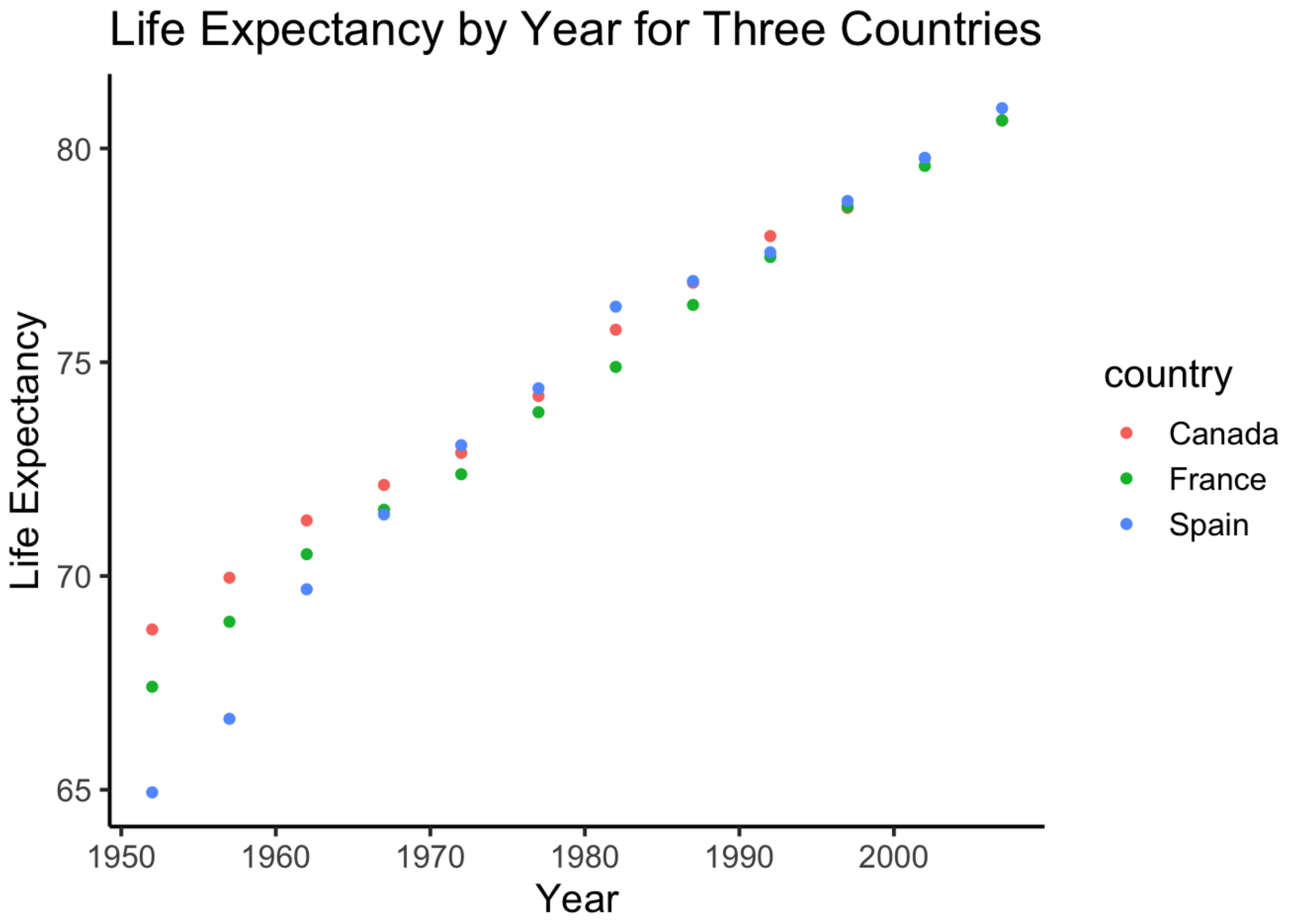
```
# filter rows to contain countries of choice
smaller_df <- gapminder_df %>%
               filter(country %in% c("Canada","France","Spain") == TRUE)

# plot the new data contained in smaller_df
ggplot(smaller_df, aes(x= year, y= lifeExp, group= country))+
  geom_point(color="red")
```



Nice, we can now see three sets of dots, but which countries do they represent? Let’s add a legend and make the graph better looking.

```
ggplot(smaller_df,aes( x= year, y= lifeExp,
                       group= country, color = country)) +
  geom_point()+
  theme_classic(base_size = 15) +
  xlab("Year") +
  ylab("Life Expectancy") +
  ggtitle("Life Expectancy by Year for Three Countries")
```

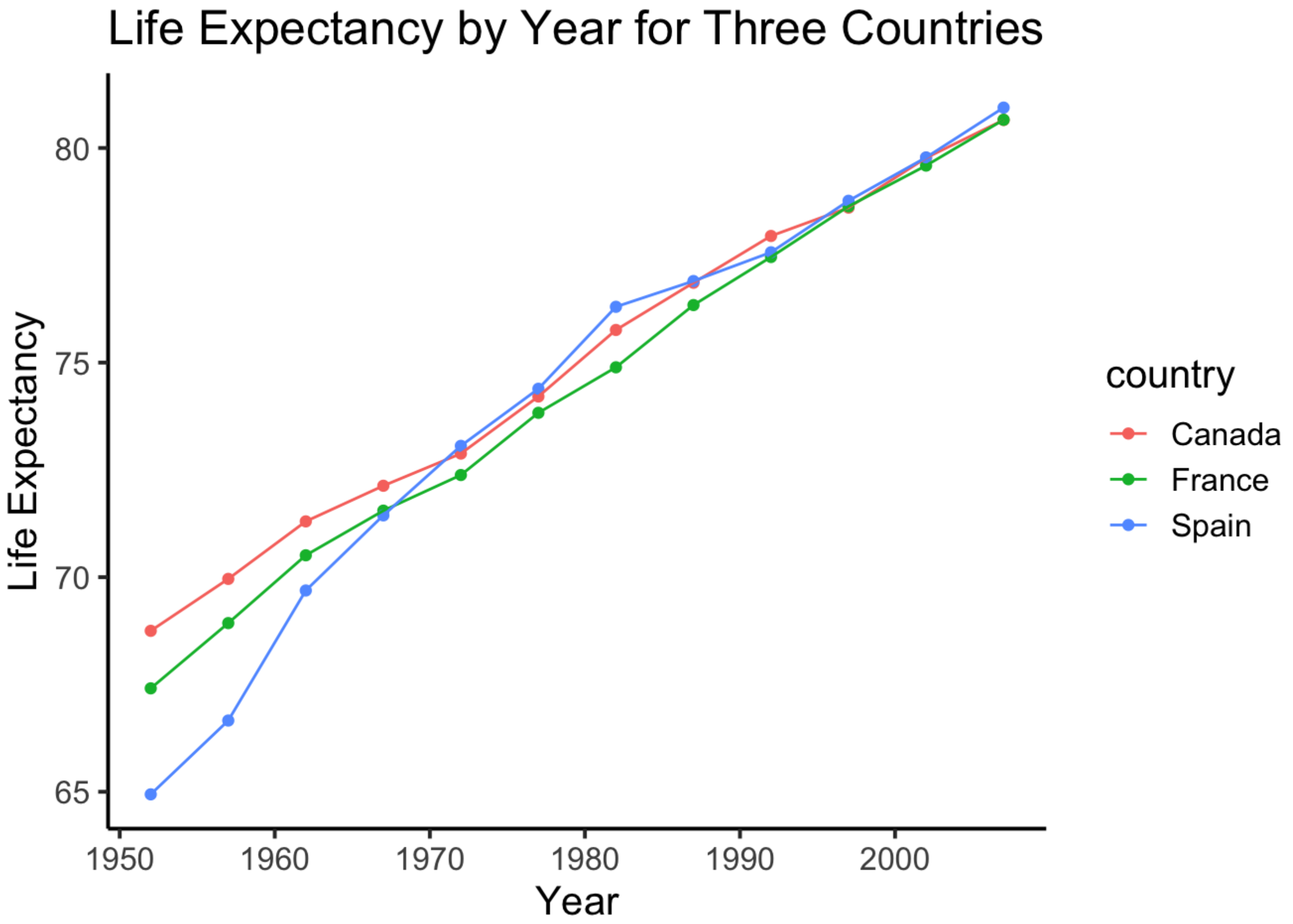


geom_line() connecting the dots

We might also want to connect the dots with a line!

To make it easier to see the connection! Remember, ggplot2 draws layers on top of layers. So, we add a new `geom_line()` layer.

```
ggplot(smaller_df,aes( x= year, y= lifeExp,
                       group= country, color = country)) +
  geom_point()+
  geom_line()+
  theme_classic(base_size = 15) +
  ylab("Life Expectancy") +
  xlab("Year") +
  ggtitle("Life Expectancy by Year for Three Countries")
```



Example

Utilizing the list of countries above, create a plot with lines connecting the points to display the life expectancy for 2 countries of your choice that haven’t been featured in the lab so far.

```
Enter your code below:
```

References

The material used in this document contains excerpts and modifications from:

- Matthew J. C. Crump, Anjali Krishnan, Stephen Volz, and Alla Chavarga (2018) “Answering questions with data: Lab Manual”. Last compiled on 2019-04-06. https://www.crumplab.com/statisticsLab/ (https://www.crumplab.com/statisticsLab/)