

Visualizing epidemiologic data in R and RStudio

Corinne Riddell

November 19, 2020

Learning objectives:

1. To put to use the `dplyr` commands from the first session
2. To make beautiful plots using the `ggplot2` package

Life expectancy in the United States by race and gender, 1969-2013

These data are partial results from a study that I did on the difference in life expectancy between non-Hispanic Black and White men and women in the United States over time.

A subset of the results have been stored in the `data/` folder as a CSV file.

Do you remember which function to use to import CSV data into R?

`readr`'s `read_csv()` to import these data

```
library(readr) #readr is part of the tidyverse
le_data <- read_csv("./data/Life-expectancy-by-state-long.csv")
```

```
## Parsed with column specification:
## cols(
##   state = col_character(),
##   stabbrs = col_character(),
##   year = col_double(),
##   sex = col_character(),
##   Census_Region = col_character(),
##   Census_Division = col_character(),
##   LE = col_double(),
##   race = col_character()
## )
```

Five functions to get to know your dataset

Function 1

```
head(le_data)
```

```
## # A tibble: 6 x 8
##   state  stabbrs  year sex  Census_Region Census_Division    LE race
##   <chr>  <chr>    <dbl> <chr>  <chr>         <chr>         <dbl> <chr>
## 1 Alabama AL      1969 Female South      East South Central  75.8 white
## 2 Alabama AL      1969 Male   South      East South Central  66.6 white
## 3 Alabama AL      1970 Female South      East South Central  75.9 white
## 4 Alabama AL      1970 Male   South      East South Central  66.7 white
## 5 Alabama AL      1971 Female South      East South Central  76.2 white
```

```
## 6 Alabama AL      1971 Male   South      East South Central 66.9 white
```

Five functions to get to know your dataset

Function 2

```
dim(le_data)
```

```
## [1] 7200    8
```

Five functions to get to know your dataset

Function 3

```
names(le_data)
```

```
## [1] "state"      "stabbrs"    "year"       "sex"
## [5] "Census_Region" "Census_Division" "LE"        "race"
```

Five functions to get to know your dataset

Function 4

```
str(le_data)
```

```
## tibble [7,200 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ state      : chr [1:7200] "Alabama" "Alabama" "Alabama" "Alabama" ...
## $ stabbrs    : chr [1:7200] "AL" "AL" "AL" "AL" ...
## $ year       : num [1:7200] 1969 1969 1970 1970 1971 ...
## $ sex        : chr [1:7200] "Female" "Male" "Female" "Male" ...
## $ Census_Region : chr [1:7200] "South" "South" "South" "South" ...
## $ Census_Division: chr [1:7200] "East South Central" "East South Central" "East South Central" "East South Central" ...
## $ LE         : num [1:7200] 75.8 66.6 75.9 66.7 76.2 ...
## $ race       : chr [1:7200] "white" "white" "white" "white" ...
## - attr(*, "spec")=
## .. cols(
## ..   state = col_character(),
## ..   stabbrs = col_character(),
## ..   year = col_double(),
## ..   sex = col_character(),
## ..   Census_Region = col_character(),
## ..   Census_Division = col_character(),
## ..   LE = col_double(),
## ..   race = col_character()
## .. )
```

Five functions to get to know your dataset

Function 5

```
View(le_data)
```

To RStudio!

Summary: Five functions to get to know your dataset

1. `head()`: prints the first 6 lines of a data frame
2. `dim()`: prints the # rows and # columns
3. `names()`: prints the variable names

4. `str()`: shows the type of each variable and some values
5. `View()`: opens the viewer pane in RStudio

Life expectancy for White men in California

Make a scatter plot of the life expectancy for White men in California over time.

Since the dataset contains 39 states across two genders and two races, first use a function to subset the data to contain only White men in California.

Which function from Malcolm's lesson do we need?

`dplyr`'s `filter()` to select a subset of rows

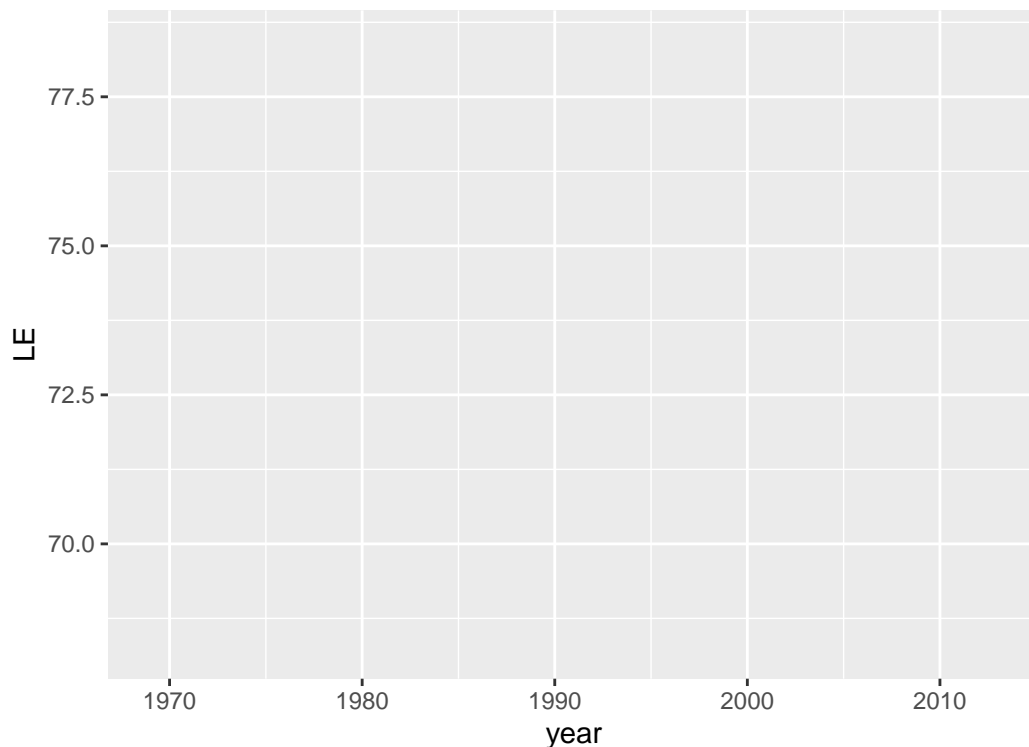
```
library(dplyr)
wm_california <- le_data %>% filter(state == "California",
                                   sex == "Male",
                                   race == "white")

#this is equivalent:
wm_california <- le_data %>% filter(state == "California" & sex == "Male" & race == "white")
```

First step to building a `ggplot()`: set up a canvas

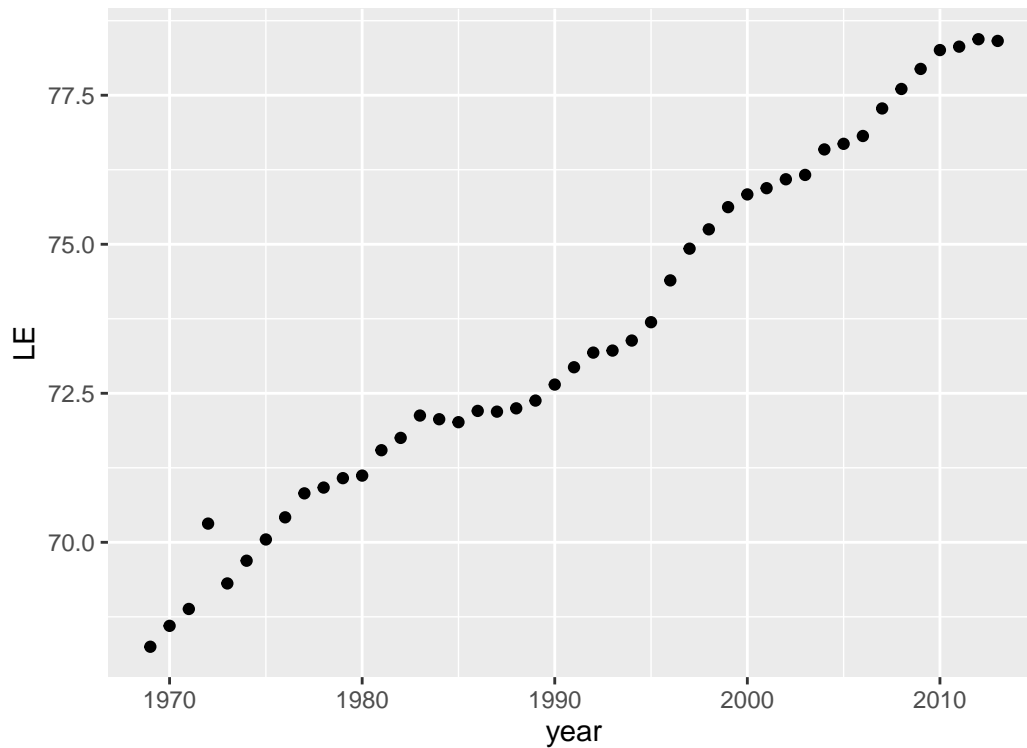
- The line of code specified the data set and what goes on the x and y axes

```
library(ggplot2)
ggplot(data = wm_california, aes(x = year, y = LE))
```



Second step to building a `ggplot()`: tell `ggplot` how to plot the data

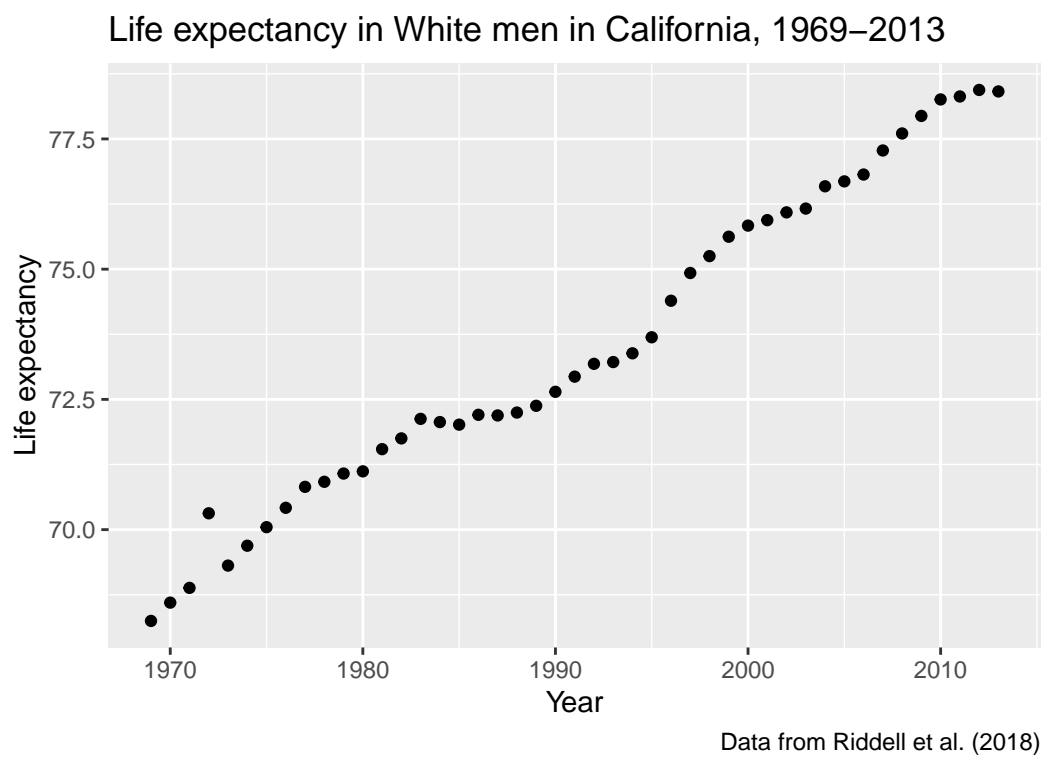
```
ggplot(data = wm_cal, aes(x = year, y = LE)) + geom_point()
```



- `geom_point()` tells `ggplot` to use points to plot these data

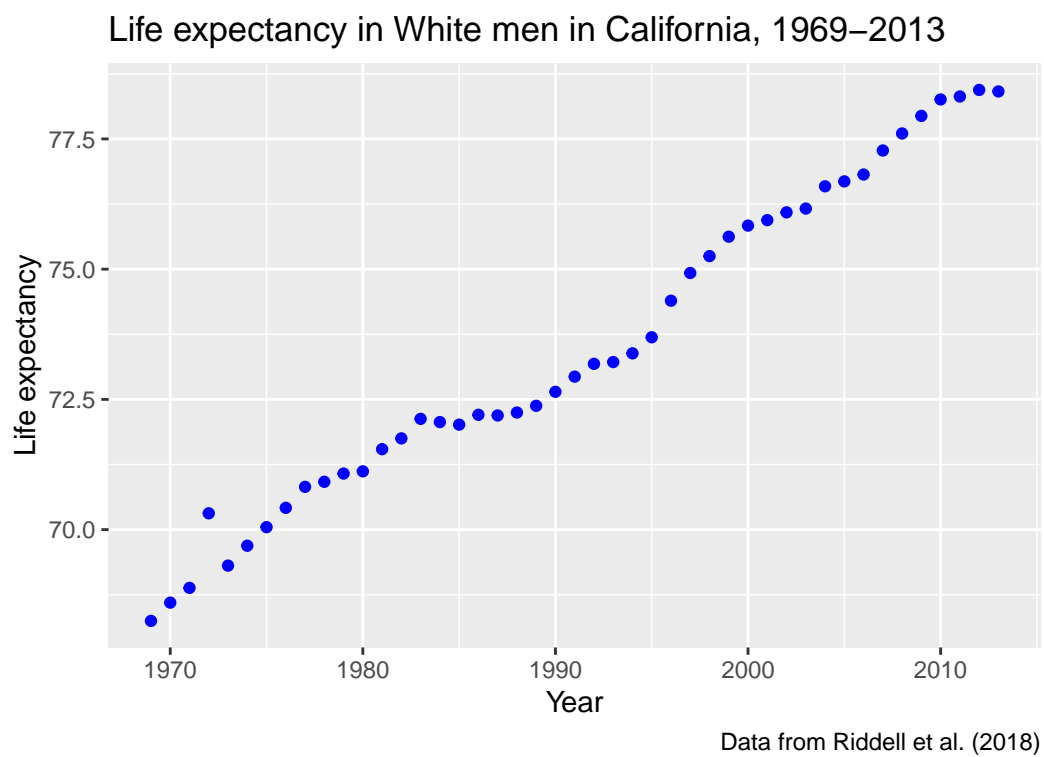
`labs()` to add a title, a caption, and modify x and y axes titles

```
ggplot(data = wm_cal, aes(x = year, y = LE)) + geom_point() +  
  labs(title = "Life expectancy in White men in California, 1969-2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



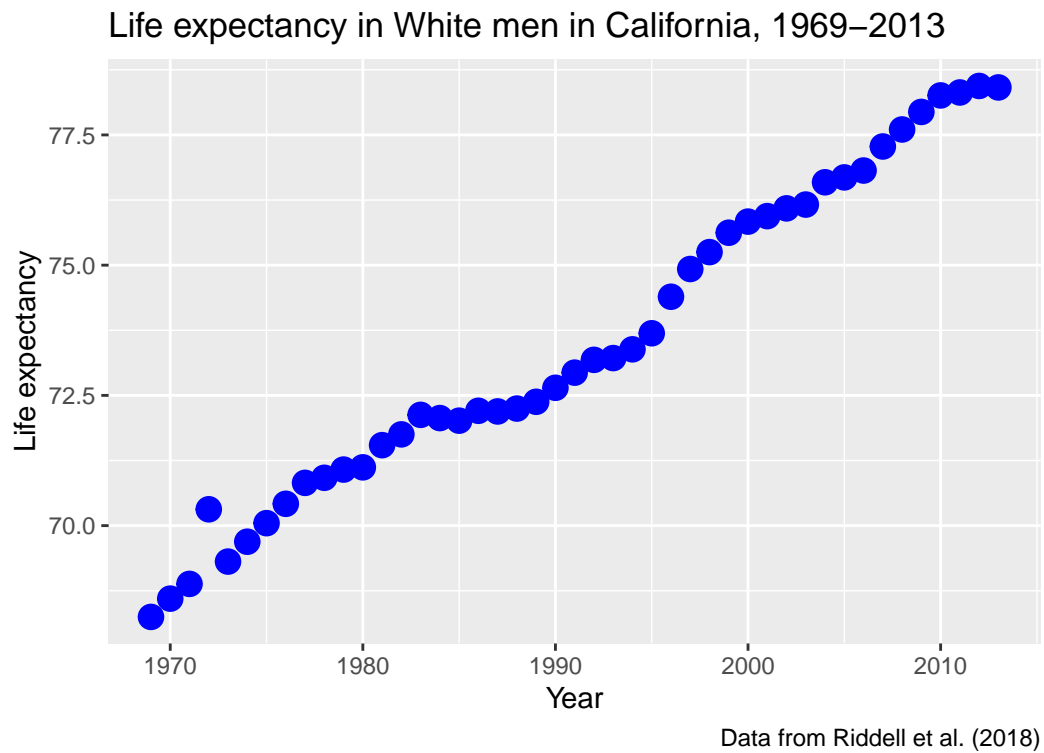
col controls the color of geom_point()

```
ggplot(data = wm_cal, aes(x = year, y = LE)) + geom_point(col = "blue") +  
  labs(title = "Life expectancy in White men in California, 1969-2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



size controls the size of geom_point()

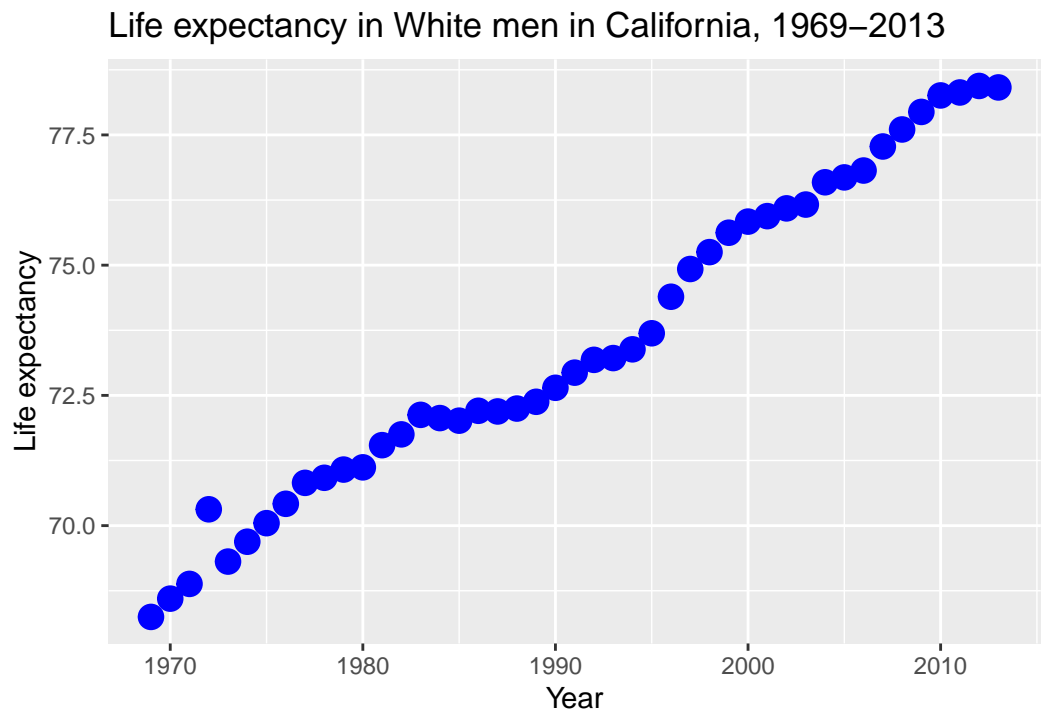
```
ggplot(data = wm_cal, aes(x = year, y = LE)) + geom_point(col = "blue", size = 4) +  
  labs(title = "Life expectancy in White men in California, 1969-2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



Line plot rather than scatter plot

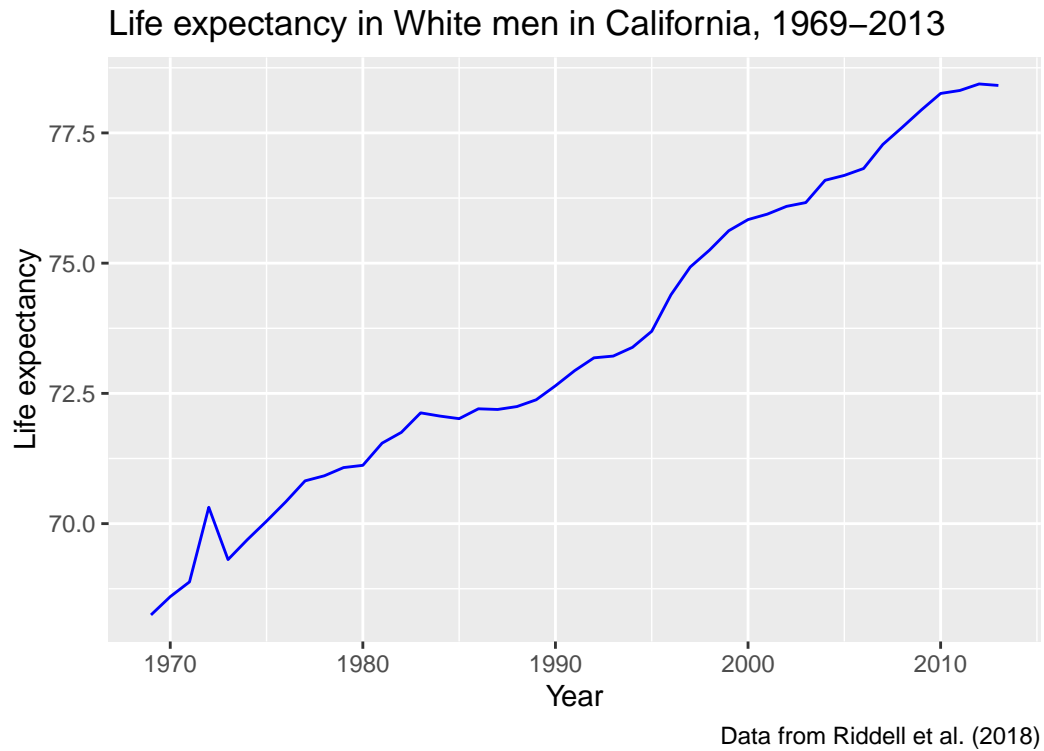
What if we wanted to make these data into a line plot instead. What part of the code should change?

```
ggplot(data = wm_cali, aes(x = year, y = LE)) +  
  geom_point(col = "blue", size = 4) +  
  labs(title = "Life expectancy in White men in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



`geom_line()` to make a line plot

```
ggplot(data = wm_cal, aes(x = year, y = LE)) + geom_line(col = "blue") +  
  labs(title = "Life expectancy in White men in California, 1969-2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```

Life expectancy for White and Black men in California

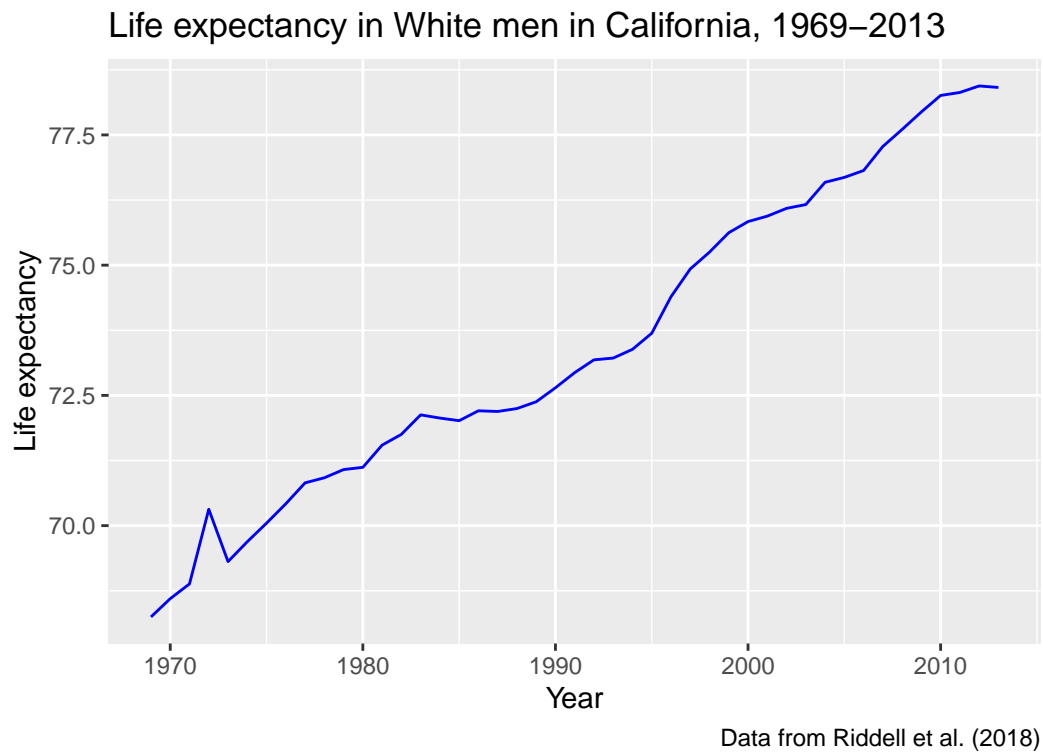
What do we need to change to make a separate line for both Black and White men?

First, update the filter()

```
wbm_cali <- le_data %>% filter(state == "California",  
                               sex == "Male")
```

Look at the previous code and output first:

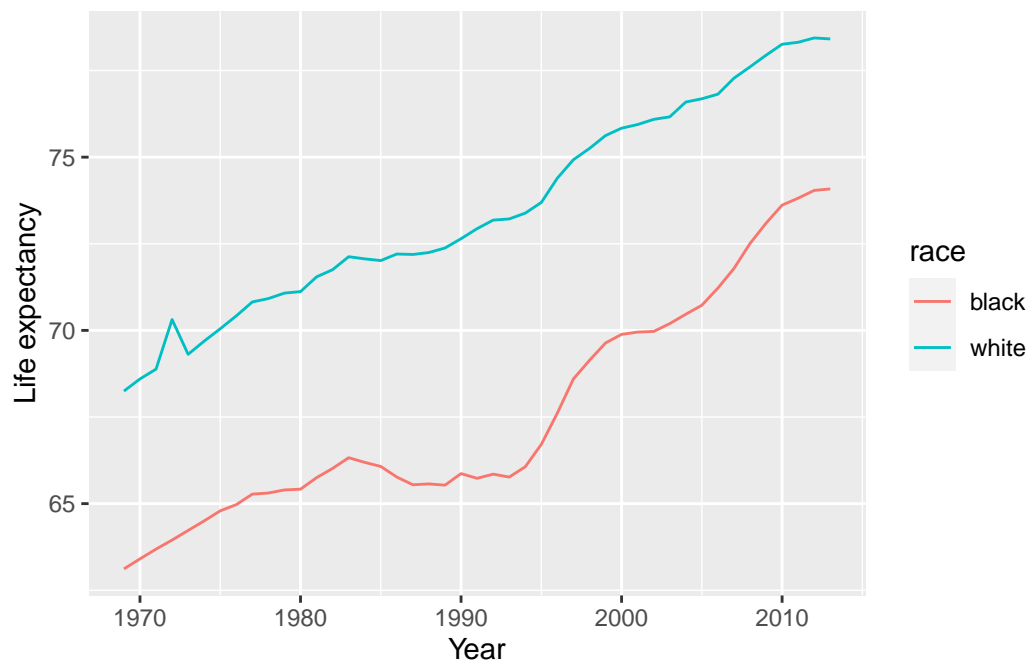
```
ggplot(data = wbm_cali, aes(x = year, y = LE)) + geom_line(col = "blue") +  
  labs(title = "Life expectancy in White men in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



And change it to *link* color to race

```
ggplot(data = wbm_cali, aes(x = year, y = LE)) + geom_line(aes(col = race)) +  
  labs(title = "Life expectancy in Black and White men in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```

Life expectancy in Black and White men in California, 1969–2013



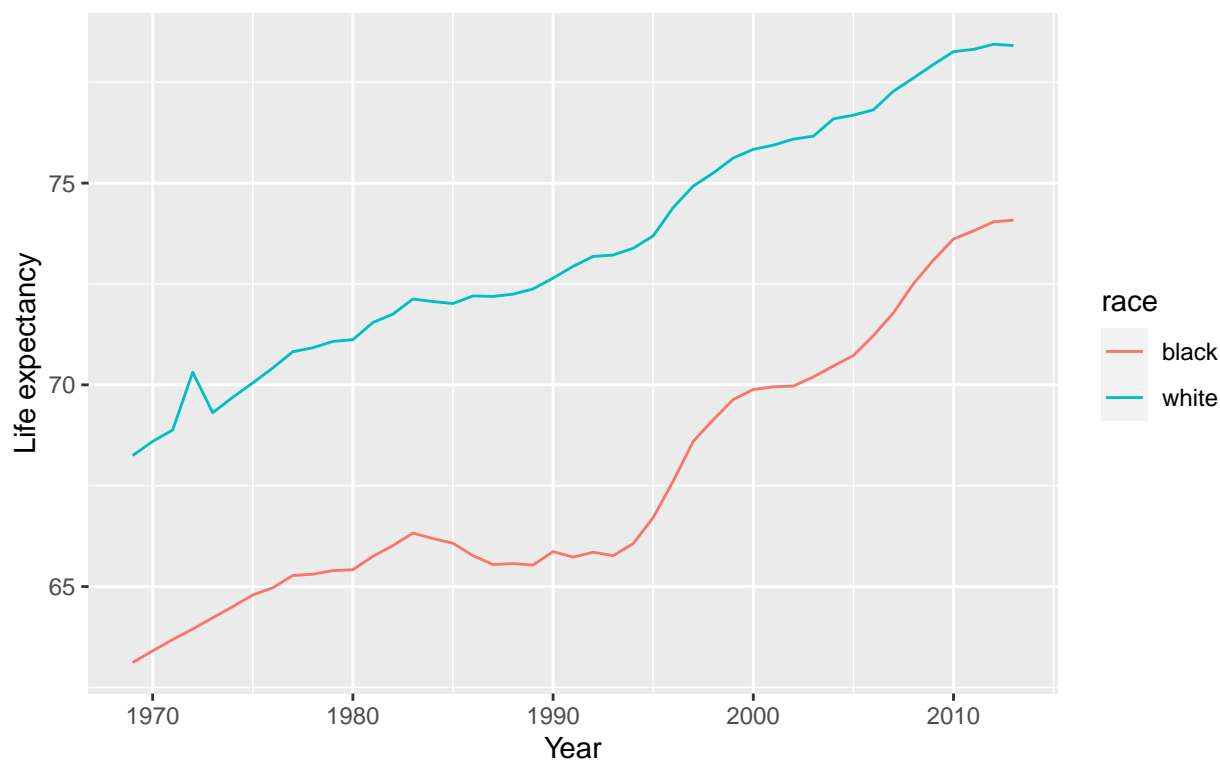
Data from Riddell et al. (2018)

Always use the `aes()` function to link a plot feature to a variable in your data frame

The operative word is *link*. Whenever you want to link something about how the plot looks to a variable in the data frame, you need to *link* these items inside the `aes()` function:

```
ggplot(data = wbm_california, aes(x = year, y = LE)) + geom_line(aes(col = race)) +  
  labs(title = "Life expectancy in Black and White men in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```

Life expectancy in Black and White men in California, 1969–2013



Data from Riddell et al. (2018)

The `aes()` function

- What else was added to the plot when you used the `aes()` function?

The `aes()` function

- What else was added to the plot when you used the `aes()` function?
 - A legend was added showing the link between the line color and the data frame's `race` variable

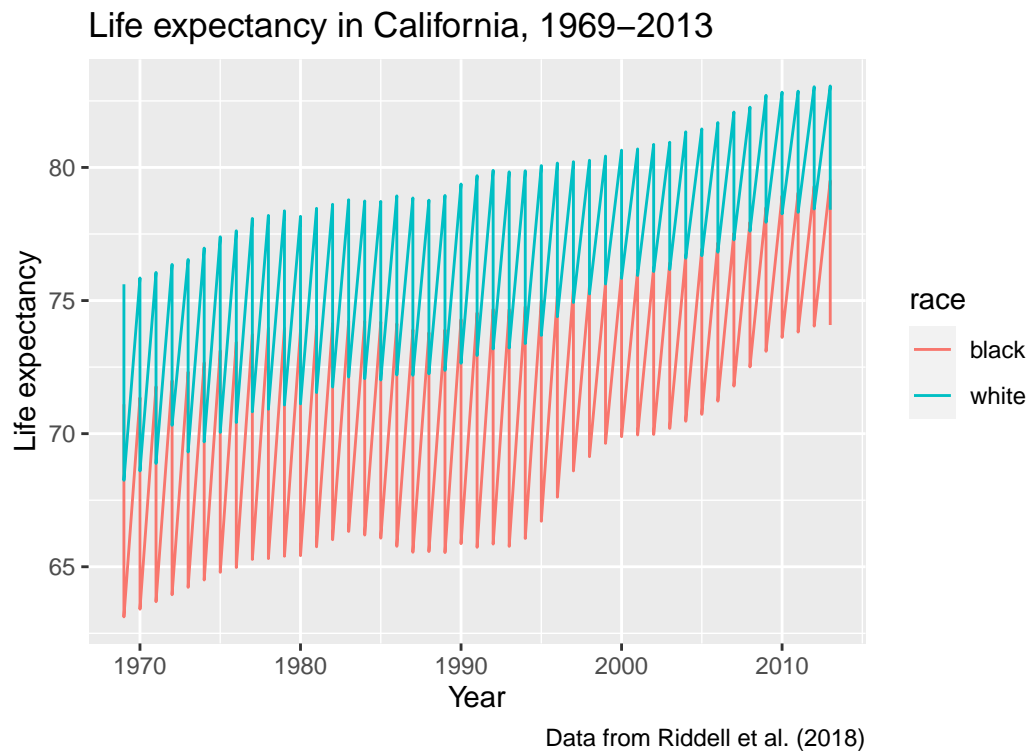
What if we also wanted to look at women?

What if we also wanted to look at women?

```
cali_data <- le_data %>% filter(state == "California")
```

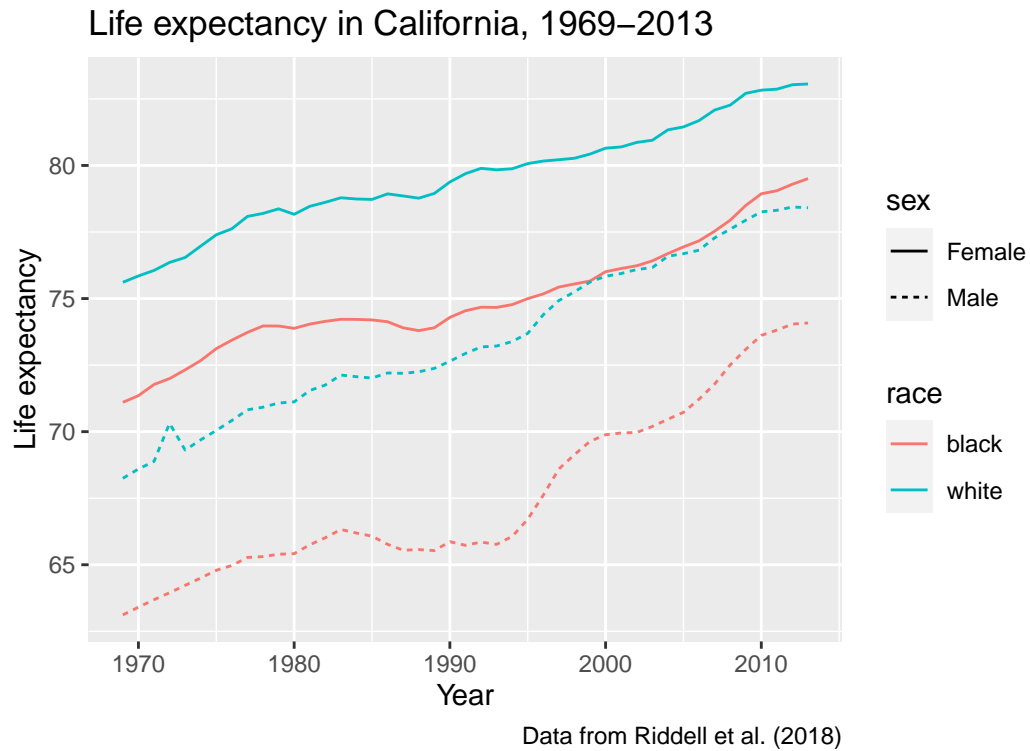
What is wrong with this plot?

```
ggplot(data = cali_data, aes(x = year, y = LE)) + geom_line(aes(col = race)) +  
  labs(title = "Life expectancy in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



Use `lty()` to link line type to sex

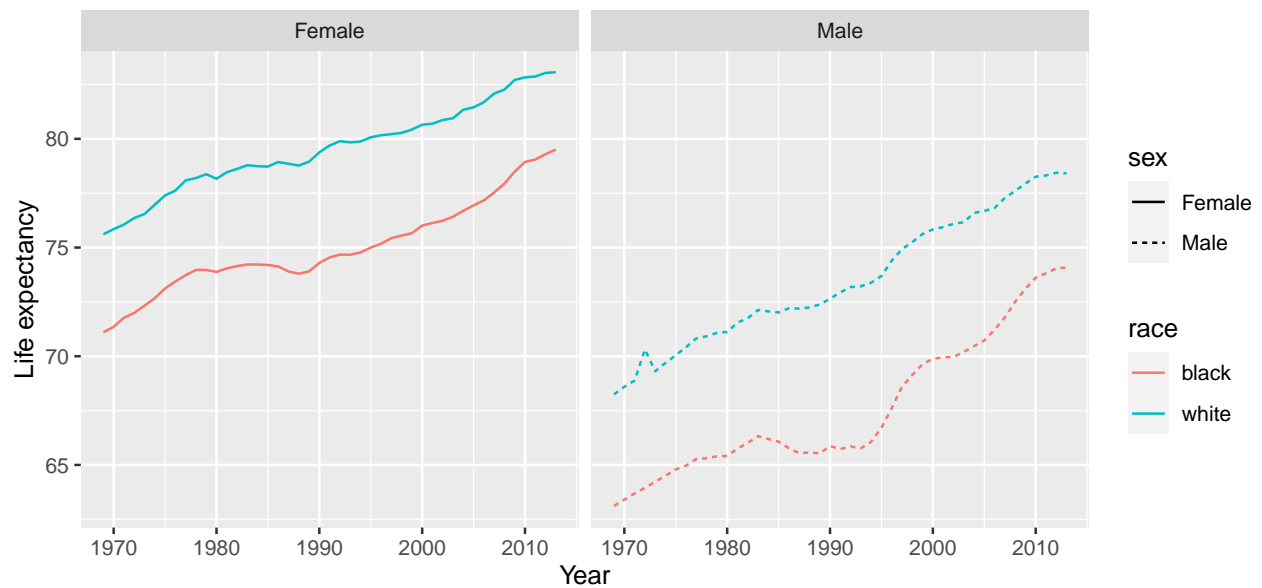
```
ggplot(data = cali_data, aes(x = year, y = LE)) + geom_line(aes(col = race, lty = sex)) +  
  labs(title = "Life expectancy in California, 1969-2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)")
```



Use `facet_wrap()` to make separate plots for a specified variable

```
ggplot(data = cali_data, aes(x = year, y = LE)) +  
  geom_line(aes(col = race, lty = sex)) +  
  labs(title = "Life expectancy in California, 1969–2013",  
        y = "Life expectancy",  
        x = "Year",  
        caption = "Data from Riddell et al. (2018)") +  
  facet_wrap(~ sex)
```

Life expectancy in California, 1969–2013



Data from Riddell et al. (2018)

Compare two states

How do we update the `filter` to include data from California and New York?

Compare two states

```
updated_data <- le_data %>% filter(state %in% c("California", "New York"))
```

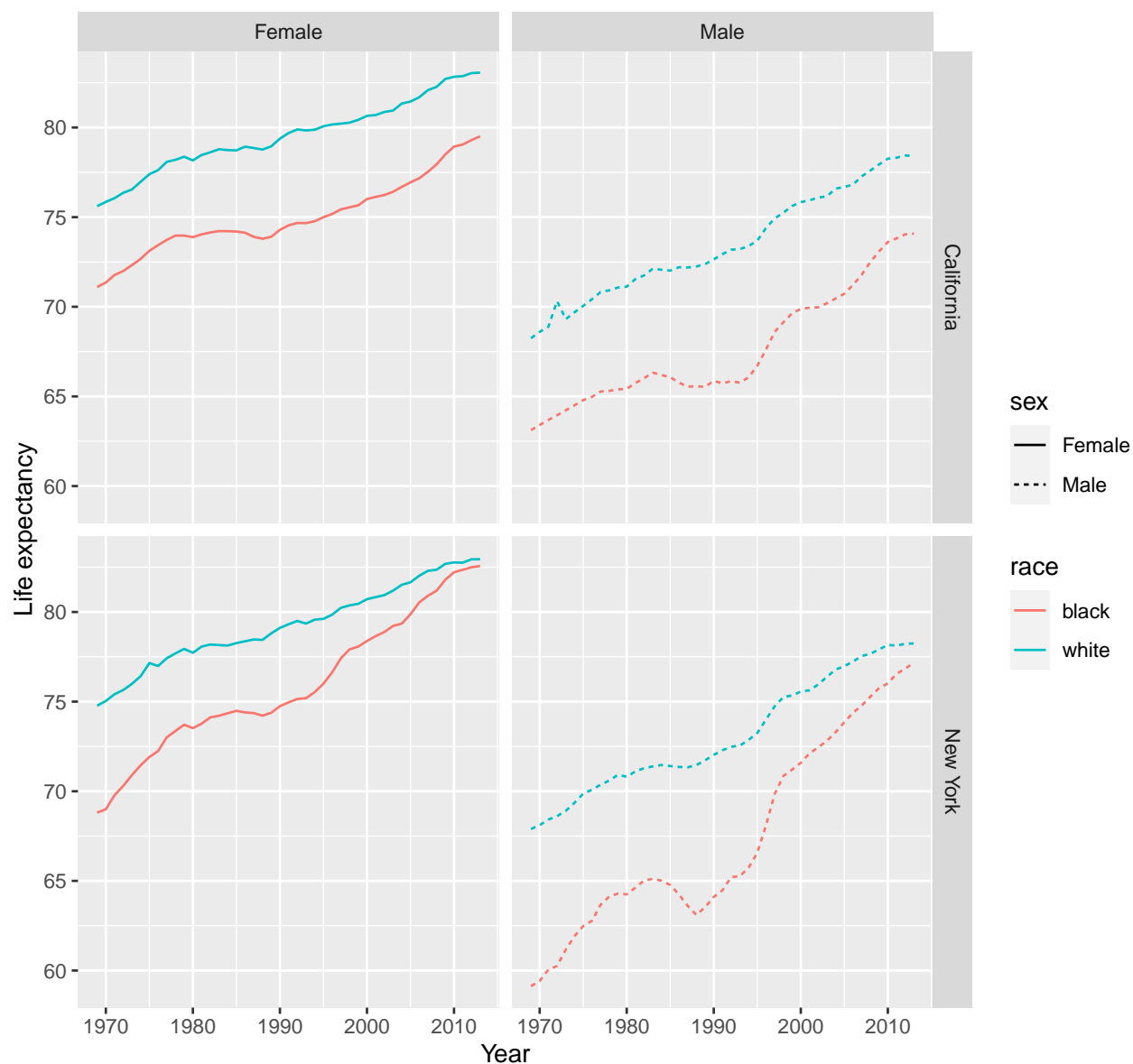
Let's write the code together

#to fill in during class

Let's write the code together

```
ggplot(data = updated_data, aes(x = year, y = LE)) +
  geom_line(aes(col = race, lty = sex)) +
  labs(title = "Life expectancy in California and New York, 1969-2013",
       y = "Life expectancy",
       x = "Year",
       caption = "Data from Riddell et al. (2018)") +
  facet_grid(state ~ sex)
```

Life expectancy in California and New York, 1969–2013



Data from Riddell et al. (2018)

Question

What is the difference between `facet_wrap()` and `facet_grid()`?

So far

- `geom_point()` to make scatter plots
- `geom_line()` to make line plots
- `col = "blue", size = 2, lty = 2`, to change color, size and line type of the geom
- `aes(col = race)` to *link* color to race
- `aes(lty = sex)` to *link* line type to sex
- `facet_wrap(~ var1)` to make separate plots for different levels of one variable
- `facet_grid(var1 ~ var2)` to make separate plots for combinations of levels of two variables

What if we wanted to make a histogram...

...of life expectancy of White men in 2013?

Before you code, try and visualize what the histogram will show

- What is on the x axis?
- What is on the y axis?

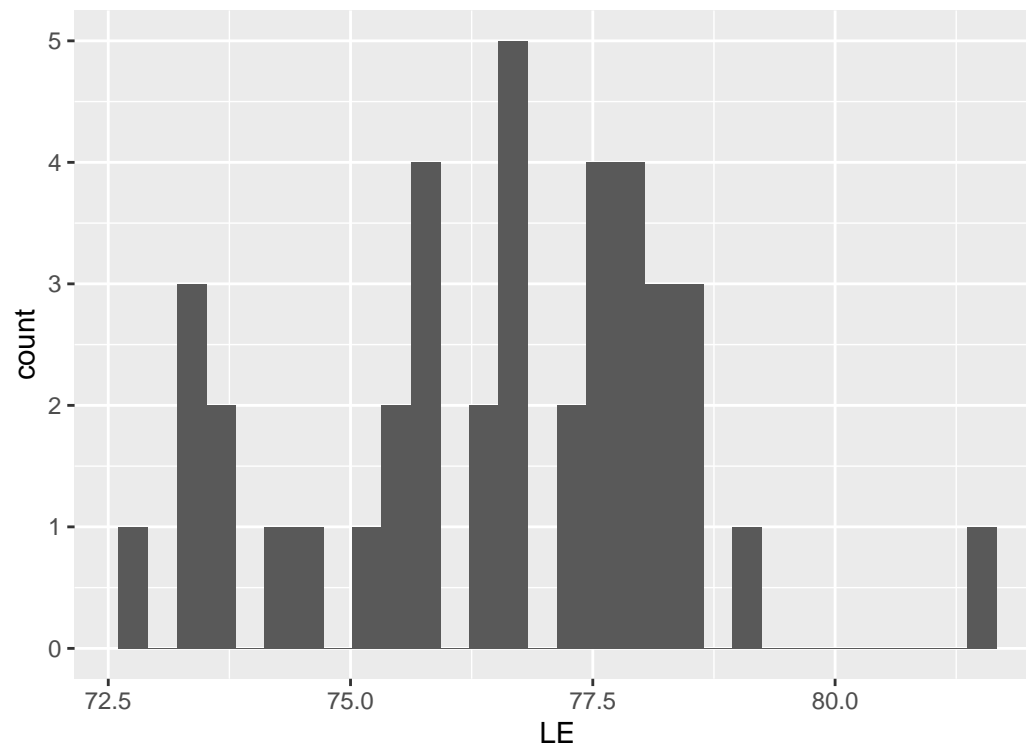
Update the filter

```
wm_data <- le_data %>% filter(year == 2013, sex == "Male", race == "white")
```

geom_histogram() to make histograms

```
ggplot(dat = wm_data, aes(x = LE)) + geom_histogram()
```

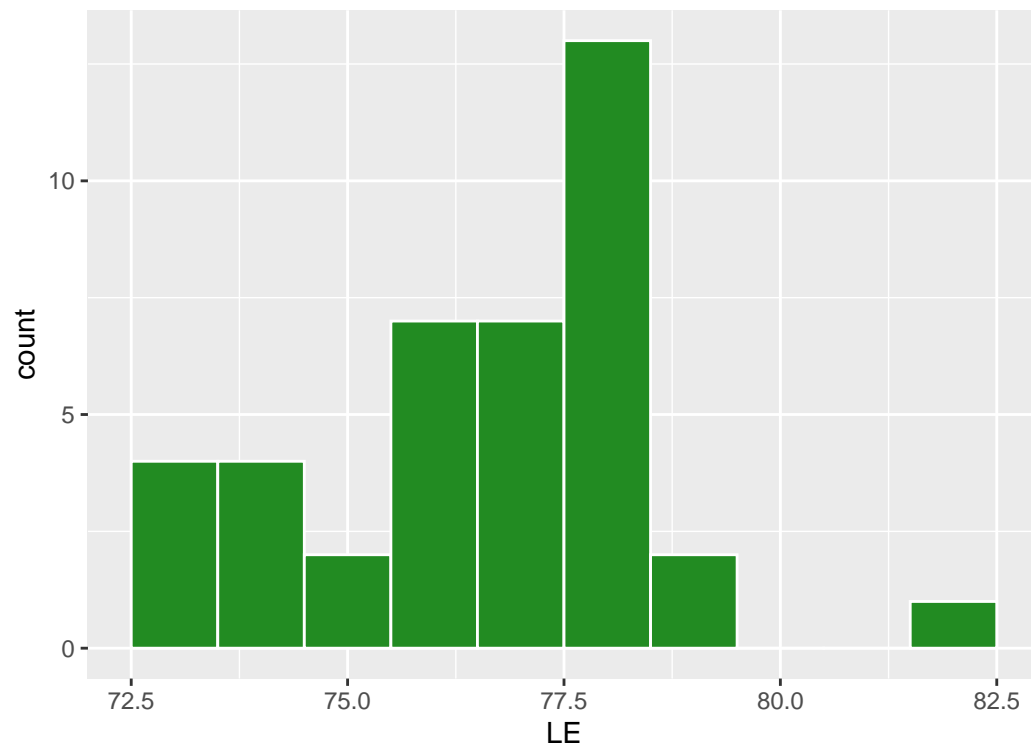
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



use fill to change the *fill* of the histogram and binwidth to specify the bin width

```
wm_data <- le_data %>% filter(year == 2013, sex == "Male", race == "white")
```

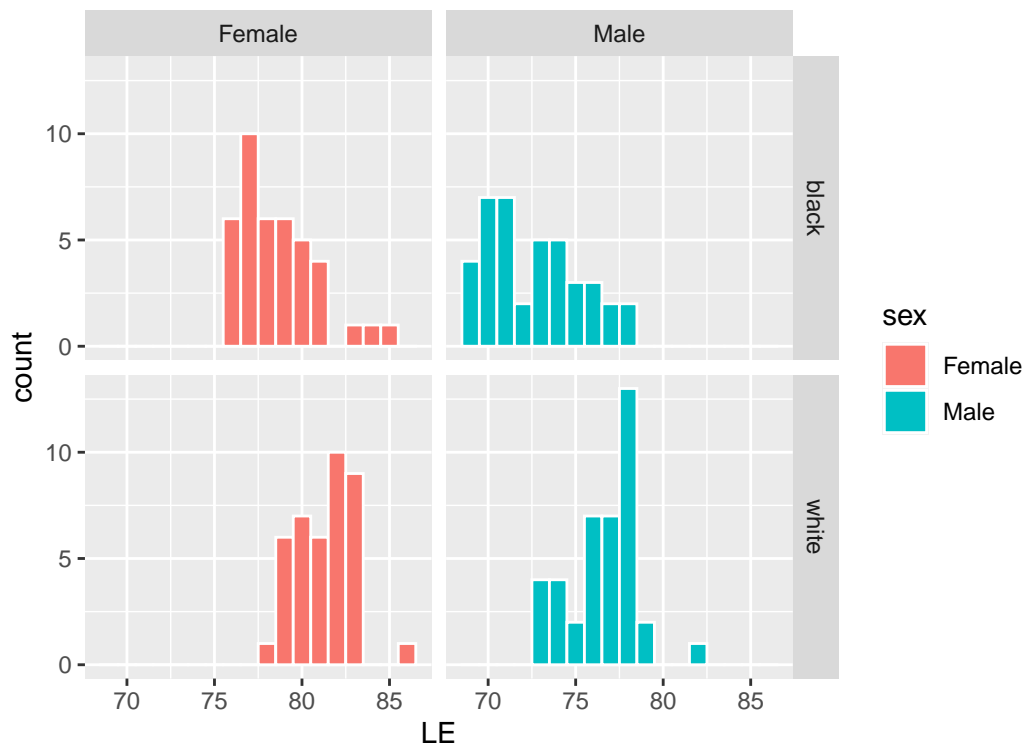
```
ggplot(dat = wm_data, aes(x = LE)) +  
  geom_histogram(binwidth = 1, col = "white", fill = "forest green")
```



Apply some of our new skills

```
data_2013 <- le_data %>% filter(year == 2013)

ggplot(dat = data_2013, aes(x = LE)) +
  geom_histogram(binwidth = 1, col = "white", aes(fill = sex)) +
  facet_grid(race ~ sex)
```



Recap: What functions did we learn?

1. `ggplot()`
 - `geom_scatter()`
 - `geom_line()`
 - `geom_histogram()`
 - `aes()` to link aesthetics to variables in our data frame
 - `facet_wrap(~ var1)`, `facet_grid(var1 ~ var2)`
 - `labs(title = "Main", y = "y axis", x = "x axis", caption = "below plot")`

Recap: What arguments were useful?

2. `ggplot()`
 - `col`
 - `size`
 - `lty`

We only skimmed the surface!

- You now have a sense of how `ggplot` works, but you might be itching to learn more.
 - Kieran Healy's data visualization book
 - RStudio `ggplot2` cheatsheet

Where to ask `ggplot2` questions

- The RStudio community page
- Stack Overflow
- On Twitter using the `#rstats` hashtag