Lecture 03: Visualizing Data

January 27 2020

Learning objectives for today:

Visualizing your data:

1. Making lovely plots using `ggplot` in R

- Visualization of categorical data: use `ggplot`'s `geom_bar()`
- Visualization of continuous data: use `ggplot`'s `geom_histogram()`

2. Describe distributions based on shape, centre, spread

Visualization of categorical data

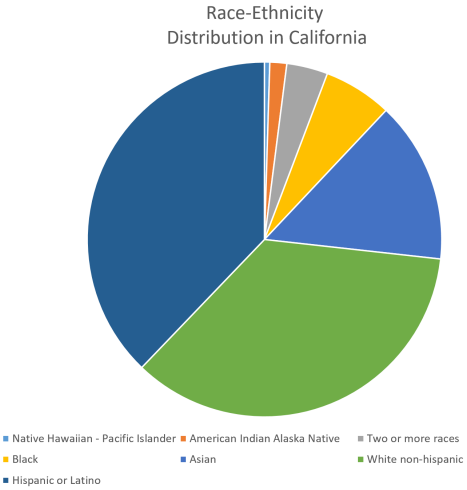
- What is the best way to visualize one categorical variable at a time?

Visualization of categorical data

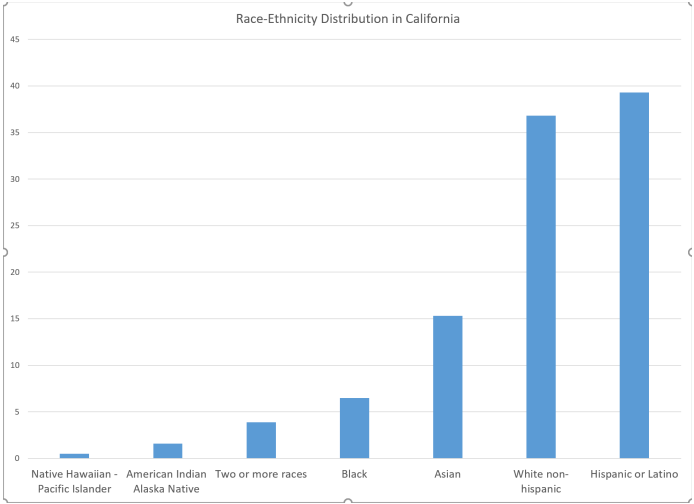
- Generally speaking, it is not a good idea to use pie charts

Visualziation of categorical data

Can you judge the area of the slices?



Visualziation of categorical data



Visualization of categorical data

- We prefer bar graphs (also called bar charts) for the display of categorical data.
- Bar charts display the number or percent of data for each level of the categorical variable being plotted

Example: infectious disease data

- Task: Make a bar chart of the percent of cases on infectious disease for each category of disease.
- First, read and view the infectious disease data from Baldi and Moore:

```
id_data <- read_csv("Ch01_ID-data.csv")
```

```
## Parsed with column specification:
## cols(
##   disease = col_character(),
##   type = col_character(),
##   number_cases = col_double(),
##   percent_cases = col_double()
## )
```

Example: infectious disease data

```
id_data
```

```
## # A tibble: 7 x 4
##   disease            type      number_cases percent_cases
##   <chr>              <chr>            <dbl>         <dbl>
## 1 Chlamydia          STI             174557         66.4
## 2 Gonorrhea          STI              44974         17.1
## 3 Pertussis          Pertussis        11219          4.27
## 4 Campylobacteriosis Foodborne         7919          3.01
## 5 Early syphilis     STI               7191          2.74
## 6 Salmonellosis      Foodborne         5361          2.04
## 7 Other              Other            11559          4.40
```

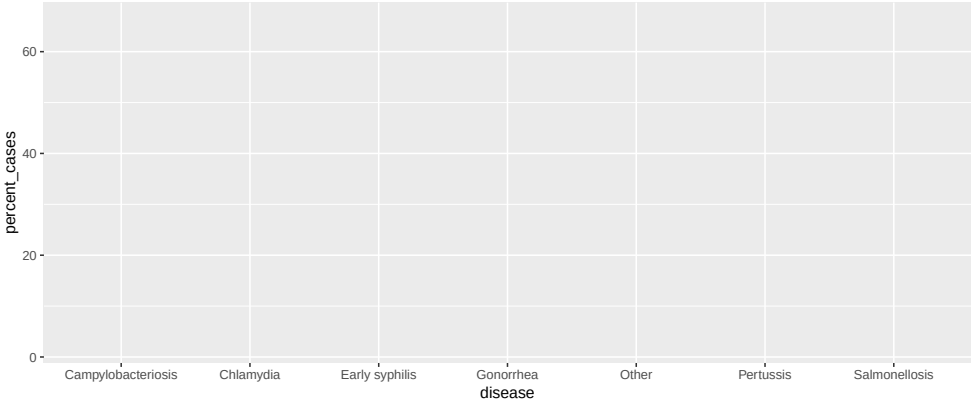
Example: infectious disease data

- Note the variables `number_cases` and `percent_cases`
- What do you want the bar chart to display? What is the x and y variables for a bar chart?

Introducing ggplot

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

- The first line of code below pulls in the ggplot package
- The second line of code below specifies the `data` set and what goes on the `x` and `y` axes



Next choose a function

- We will use a `geom_` function to create our chart

`ggplot()`'s `geom_bar()` makes a bar chart

Syntax for bar charts

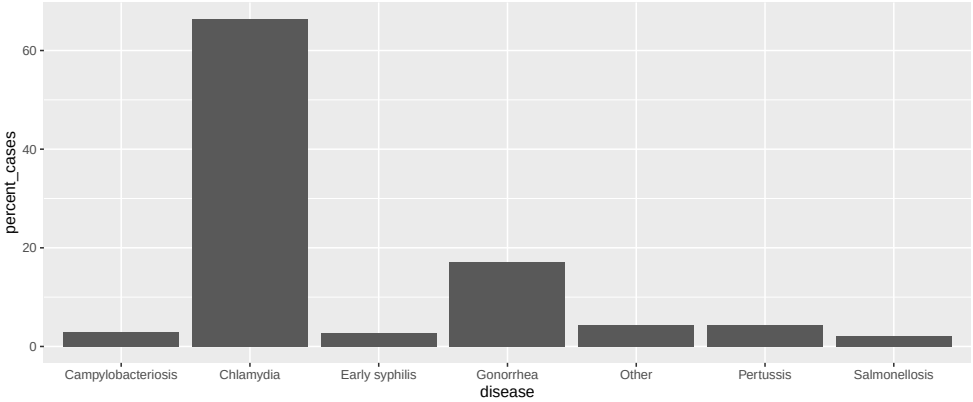
ggplot(id_data, aes(x = disease, y = percent_cases)) +

geom_bar(stat = "identity")

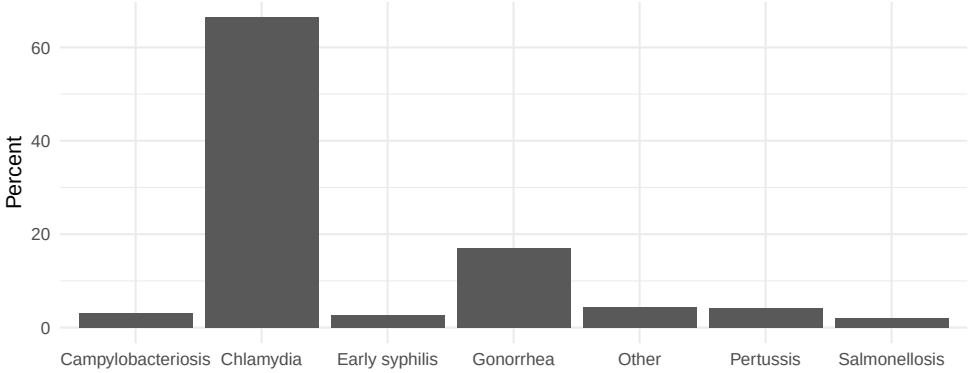
stat = "identity" tells geom_bar that we supplied a y variable that is exactly what we want to plot.

We do not need geom_bar() to calculate the number or percent for us.

ggplot()'s geom_bar() makes a bar chart



some additions to ggplot for style



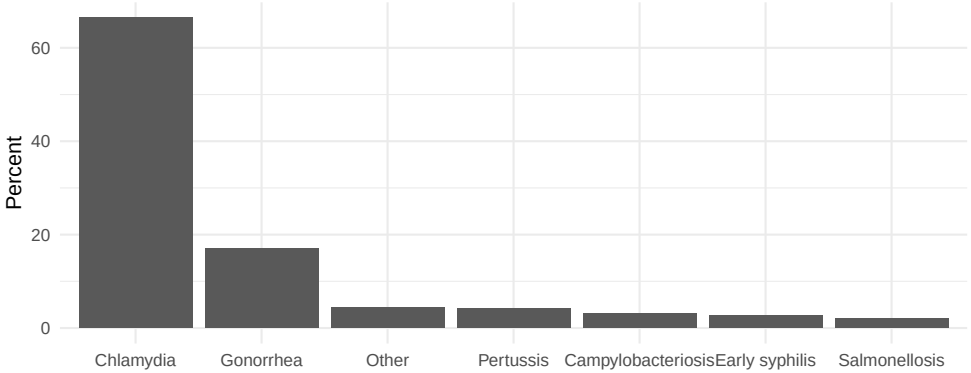
base_size controls the font size on these plots

theme_minimal affects the "look" of the plot it removes the grey background and adds grey gridlines

fct_reorder reorders disease according to value of percent_cases

```
id_data <- id_data %>%
  mutate(disease_ordered = fct_reorder(disease, percent_cases, .desc = T))
```

Re-ordered plot

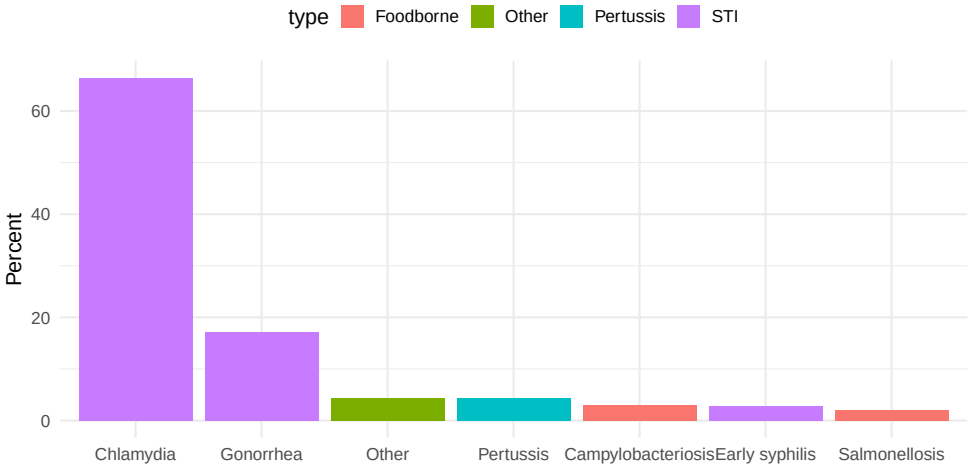


Use `aes(fill = type)` to link the bar's fill to the disease type

geom_bar(stat = "identity", aes(fill = type)) +

theme(legend.position = "top")

Use `aes(fill = type)` to link the bar's fill to the disease type



Visualizing quantitative variables

Visualize quantitative variables using histograms

- Histograms look a lot like bar charts, except that the bars touch because the underlying scale is continuous and the order of the bars matters
- In order to make a histogram, the underlying data needs to be binned into categories and the number or percent of data in each category becomes the height of each bar.
- the bins devide the entire range of data into a series of intervals and counts the number of observations in each interval
- the intervals must be consecutive and non-overlapping and are almost always chosen to be of equal size

Example: opioid state prescription rates

- The textbook gives an example using data from 2012.
- In the data folder, there is updated data from 2018. It came from the paper: "Opioid Prescribing Rates by Congressional Districts, United States, 2016", by Rolheiser et al. link

Example: opioid state prescription rates

Problem: To determine the extent to which opioid prescribing rates vary across US congressional districts.

Plan: In an observational cross-sectional framework using secondary data, they constructed 2016 congressional district–level opioid prescribing rate estimates using a population-weighted methodology.

Data: In the data structure we have State as the unit of analysis, and measured perscription rates as the variable of interest

Example: opioid state prescription rates

```r
opi_data <- read.csv("Ch01_opioid-data.csv")
head(opi_data)
```

```
##   Rank State   Mean Median    SD    Min    Max Num_Districts
## 1    1    AL 121.31 113.09 21.87 105.58 166.69             7
## 2    2    AR 115.22 115.13  8.59 104.80 125.79             4
## 3    3    TN 108.12 108.26 19.16  73.60 133.00             9
## 4    4    MS 105.64 106.25 17.36  83.90 126.14             4
## 5    5    LA  98.38  98.88 10.34  83.22 112.65             6
## 6    6    KY  98.13  85.76 26.72  77.62 147.00             6
```

- `Mean` provides the mean prescribing rate per 100 individuals. Thus, a mean of 121.31 implies that in Alabama, there were 121.31 opioid prescriptions per 100 persons, an average across the 7 congressional districts.

Histogram of opioid prescription rates

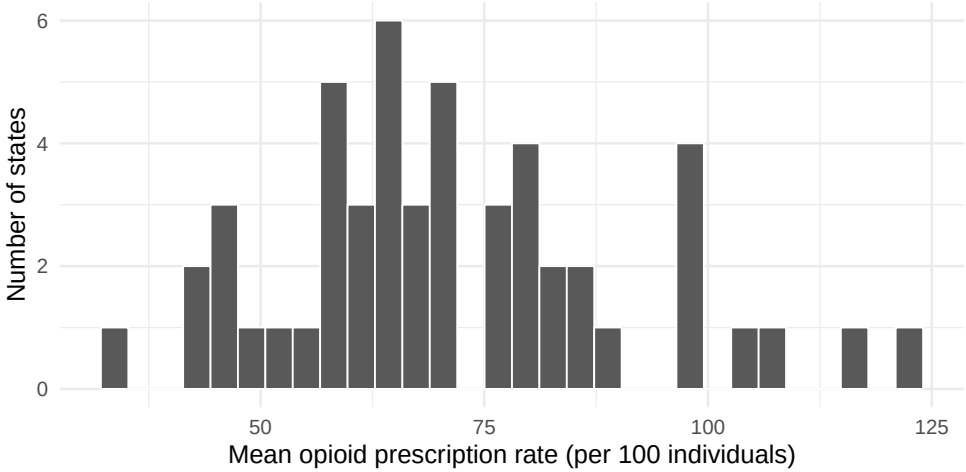
- Task: Make a histogram of the average prescribing rates across US states
- What is the x variable? What is the y variable?
- What `geom` should be used?

Histogram of opioid prescription rates - default is 30 bins

```
ggplot(data = opi_data, aes(x = Mean)) +
geom_histogram(col = "white") +
labs(x = "Mean opioid prescription rate (per 100 individuals)",
    y = "Number of states") +
theme_minimal(base_size = 15)
```

Histogram of opioid prescription rates

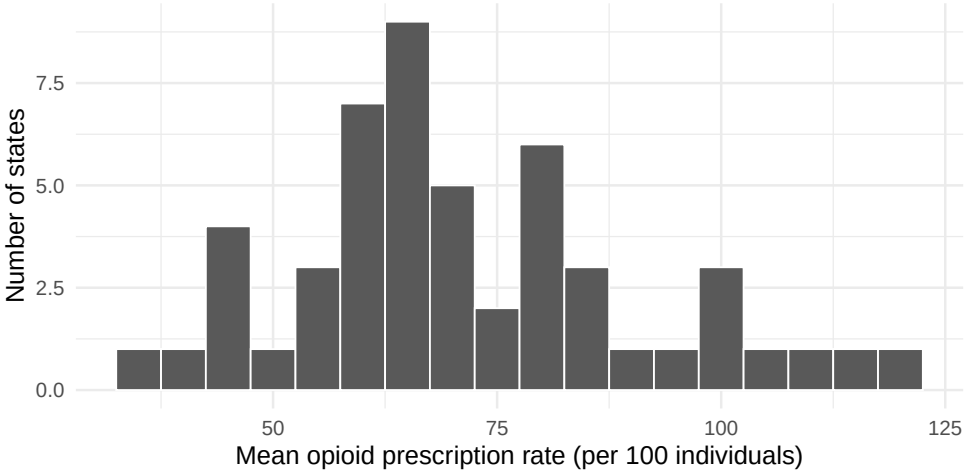
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



same graph, change the bins `geom_histogram(binwidth = 5)`

```
ggplot(data = opi_data, aes(x = Mean)) +
geom_histogram(col = "white", binwidth = 5) +
labs(x = "Mean opioid prescription rate (per 100 individuals)",
     y = "Number of states") +
theme_minimal(base_size = 15)
```

same graph, change the bins `geom_histogram(binwidth = 5)`



change the bins again `geom_histogram(binwidth = 10)`

ggplot(data = opi_data, aes(x = Mean)) +

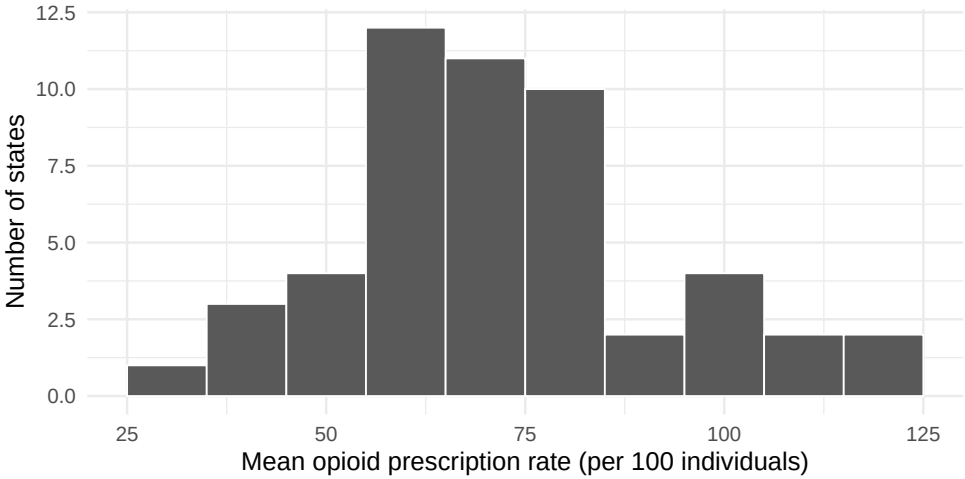
geom_histogram(col = "white", binwidth = 10) +

labs(x = "Mean opioid prescription rate (per 100 individuals)",

```
y = "Number of states") +
```

theme_minimal(base_size = 15)

change the bins again `geom_histogram(binwidth = 10)`

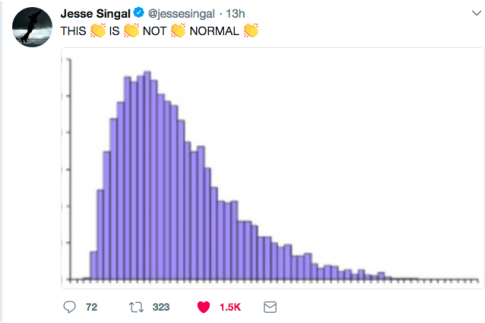


Describing your distribution based on shape, center and spread

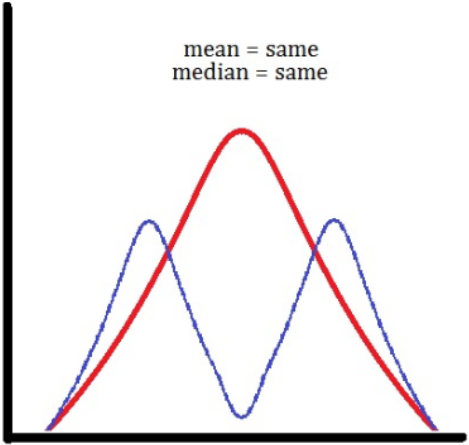
Shape, Center, Spread

- When we examine histograms, we can make comments on a distribution's:
  - Shape: Is the distribution symmetric or skewed to the left or right?
  - Center: Does the histogram have one peak (unimodal), or two (bimodal) or more?
  - Spread: How spread out are the values? What is the range of the data?
  - Outliers: Do any of the measurements fall outside of the range of most of the data points?

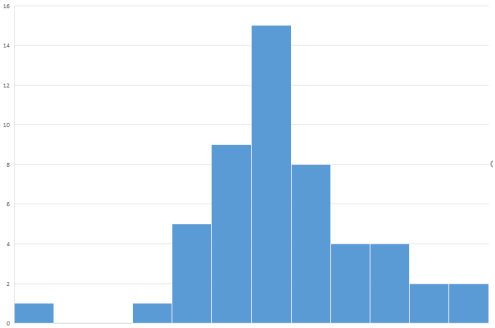
Is this skewed left or skewed right?



Center - one hump or two?

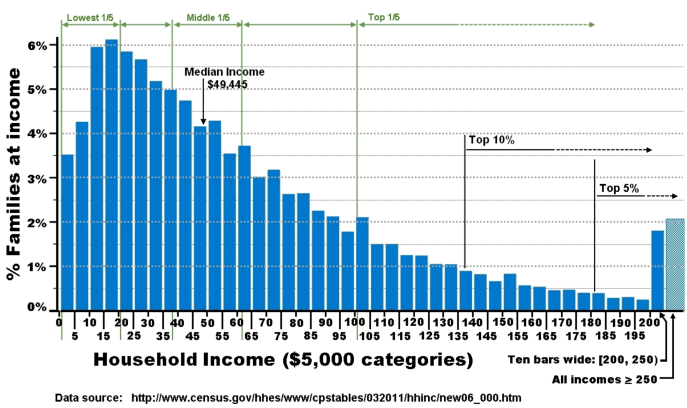


Outlier



Participation

Participation



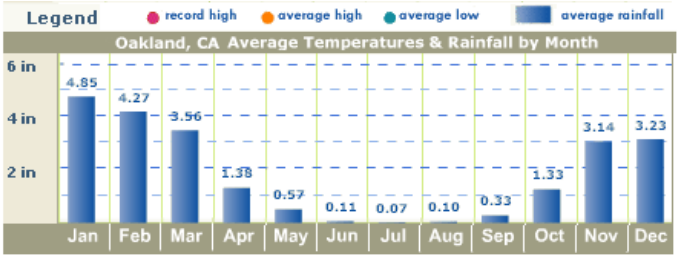
Time plots

Visualize quantitative variables over time using time plots

- Time plots are a specific subset of line plots where the x variable is time.
- Unlike the previous plots, the time plot shows a relationship between two variables:
  1. a quantitative variable
  2. time
- Often times, these plots can be used to look for cycles (e.g., seasonal patterns that recur each year) or trends (e.g., overall increases or decreases seen over time).

Time plot

- from *See California.com*, January 2019:



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 last lecture do we need?

- `mutate()`, `select()`, `filter()`, `rename()`, or `arrange()`?

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

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

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

Here we use `geom_point` to make a graph with dots

ggplot(data = wm_cali, aes(x = year, y = LE)) +

geom_point() +

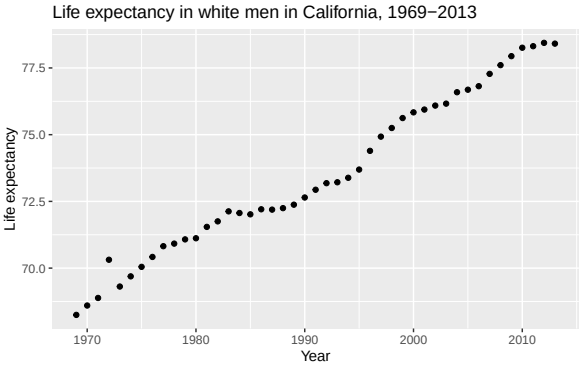
labs(title = "Life expectancy in white men in California, 1969-2013",

```
y = "Life expectancy",

x = "Year",

caption = "Data from Riddell et al. (2018)")
```

Here we use `geom_point` to make a graph with dots



Data from Riddell et al. (2018)

geom_line() to make a line plot

ggplot(data = wm_cali, aes(x = year, y = LE)) +

geom_line(col = "blue") +

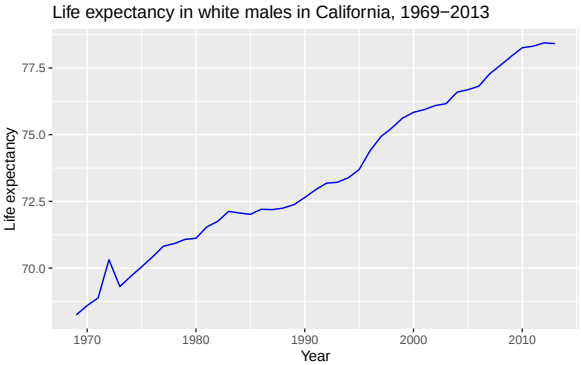
labs(title = "Life expectancy in white males in California, 1969-2013",

```
y = "Life expectancy",

x = "Year",

caption = "Data from Riddell et al. (2018)")
```

geom_line() to make a line plot



Data from Riddell et al. (2018)

R Recap: new code?

1. 'ggplot' to set up a canvas for graphics
2. `geom_bar(stat = "identity")` to make a bar chart when you specify the y variable
3. `geom_histogram()` to make a histogram for which ggplot needs to calculate the count
4. `fct_reorder(var1, var2)` to reorder a categorical variable (`var1`) by a numeric variable (`var2`)
   - from the `forcats` package
5. `geom_point()` to make a plot with dots
6. `geom_line()` to make a plot with lines

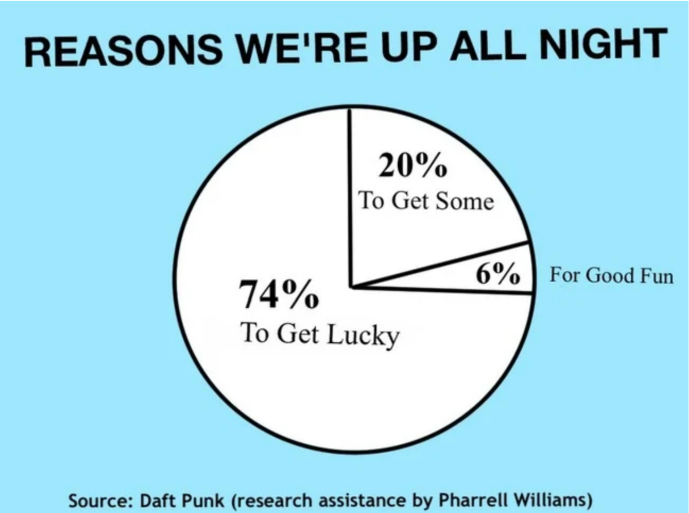
How to get help with code

- Ask questions during labs, GSI office hours, or on Piazza discussion forum. Use the appropriate thread!
- Develop your online search skills. For example if you have a `ggplot2` question, begin your google search with "r ggplot" and then describe your issues, e.g., "r ggplot how do I make separate lines by a second variable".
- The most common links that will appear are:
  - https://stackoverflow.com: Crowd-sourced answers that have been upvoted. The top answer is often the best one.
  - https://ggplot2.tidyverse.org/: The official ggplot2 webpage is very helpful.
  - https://community.rstudio.com/: The RStudio community page.
  - https://rpubs.com/: Web pages made by R users that often contain helpful tutorials.

We only skimmed the surface!

- Here is some extra material for those of you who love data visualization. This material won't be tested.
  - RStudio ggplot2 cheatsheet
  - Kieran Healy's data visualization book

Parting Humor



- from *Eric Tanoye Song Lyrics in Chart Form*