Chapter 1: Picturing Distributions with Graphs

Corinne Riddell

Aug 31, 2022

Learning objectives for today:

1. What are the four types of variables?
2. Visualization of categorical data: use `ggplot`'s `geom_bar()`
3. Visualization of continuous data: use `ggplot`'s `geom_histogram()`
4. Describe distributions based on their shape, center, and spread.

Readings

- Chapter 1 of Baldi and Moore
- General `ggplot` R code Resources
  - `geom_bar()`(See 2.8.1)
  - `geom_histogram()`(2.5.1-2.5.2)

Types of variables

- **Categorical** variable: A variable that has grouping levels. Mathematically you can calculate the proportion (%) of individuals in each level of the category.
  - **Nominal** variables: have no underlying order or rank. E.g., hospital ID, HIV status (yes/no variables)

  - **Ordinal** variables: can be ordered or ranked. E.g., socio-economic status
- **Quantitative** variable: A continuous, numeric variable that you can perform mathematical operations on. Mathematically, we can you take the median or average of these variables
  - **Discrete** variables: can be counted. E.g., number of brain lesions
  - **Continuous** variables: can be measured precisely, with a ruler or scale. E.g, blood alcohol content, gestational age at birth

Examples

Categorize each variable as nominal, ordinal, discrete, or continuous:

- `BMI category`:
- `Race`:
- `Income`:
- `Number of previous births`:

Examples

Categorize each variable as nominal, ordinal, discrete, or continuous:

- `BMI category`: ordinal - there is an underlying order to this categorical variable
- `Race`:
- `Income`:
- `Number of previous births`:

Examples

Categorize each variable as nominal, ordinal, discrete, or continuous:

- `BMI category`: ordinal - there is an underlying order to this categorical variable
- `Race`: nominal - no underlying order to racial categories.
- `Income`:
- `Number of previous births`:

Examples

Categorize each variable as nominal, ordinal, discrete, or continuous:

- `BMI category`: ordinal - there is an underlying order to this categorical variable
- `Race`: nominal - no underlying order to racial categories.
- `Income`: continuous variable - can measure income to the nearest dollar or cent
- `Number of previous births`:

Examples

Categorize each variable as nominal, ordinal, discrete, or continuous:

- `BMI category`: ordinal - there is an underlying order to this categorical variable
- `Race`: nominal - no underlying order to racial categories.
- `Income`: continuous variable - can measure income to the nearest dollar or cent
- `Number of previous births`: discrete - can count the number of previous births a person has

Visualization of categorical data

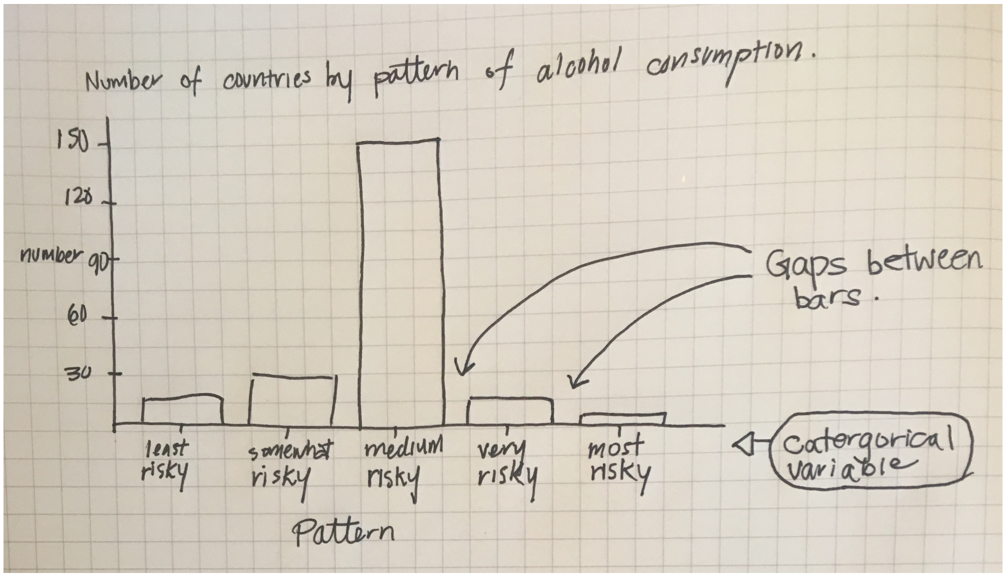
- What is the best way to visualize one categorical variable at a time?

Visualization of categorical data

- Generally speaking, **it is not a good idea to use pie charts**
  - Difficult to judge the area of the smallest slices
  - Areas of the smallest (largest) slices are under-judged (over-judged)
- We will not use pie charts in this class

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



An example using 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("./data/Ch01_ID-data.csv")
```

```
## Rows: 7 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): disease, type
## dbl (2): number_cases, percent_cases
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
id_data
```

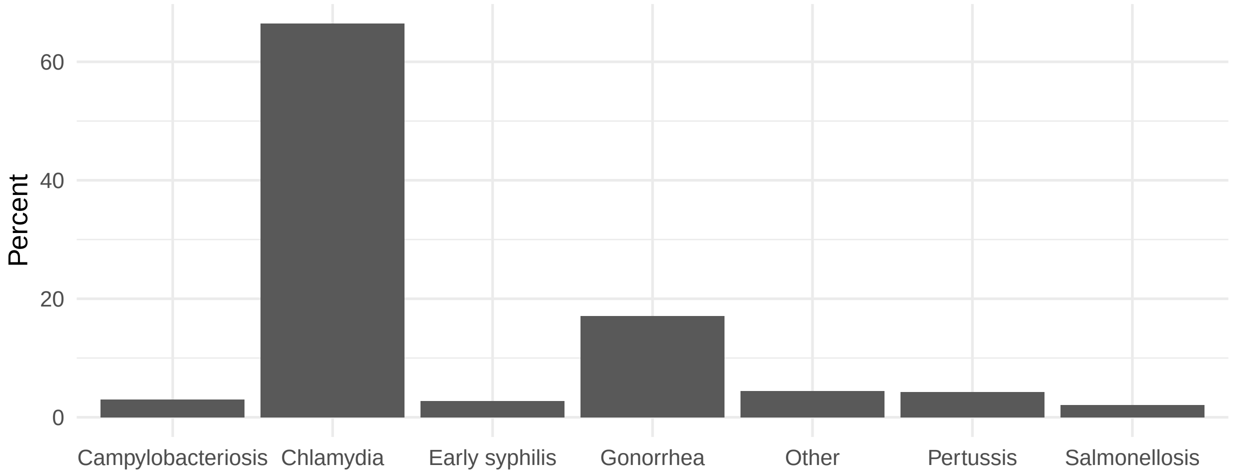
```
## # A tibble: 7 x 4
##   disease             type      number_cases percent_cases
##   <chr>               <chr>            <dbl>         <dbl>
## 1 Chlamydia           STI             174557         66.4
## 2 Gonorrhea           STI              44974         17.1
## 3 Pertussis           Pertussis        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
```

An example using 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?
- What `geom_` should we use? (It is one we have not learnt yet)

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

```
ggplot(id_data, aes(x = disease, y = percent_cases)) +
  geom_bar(stat = "identity") +
  labs(y = "Percent", x = "") +
  theme_minimal(base_size = 15)
```

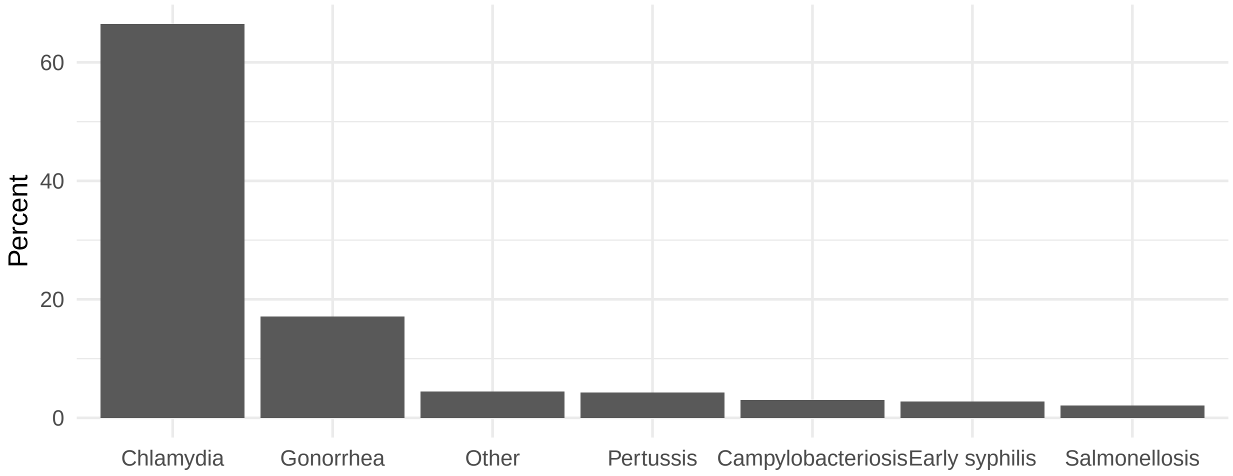


- `stat = "identity"` tells geom_bar that we supplied a y variable that is exactly what we want to plot. We do not need geom_bar() to calculate the number or percent for us.
- `base_size` controls the font size on these plots
- `theme_minimal()` affects the "look" of the plot. It removes the grey background and adds grey gridlines

`fct_reorder` reorders `disease` according to value of `percent_cases`

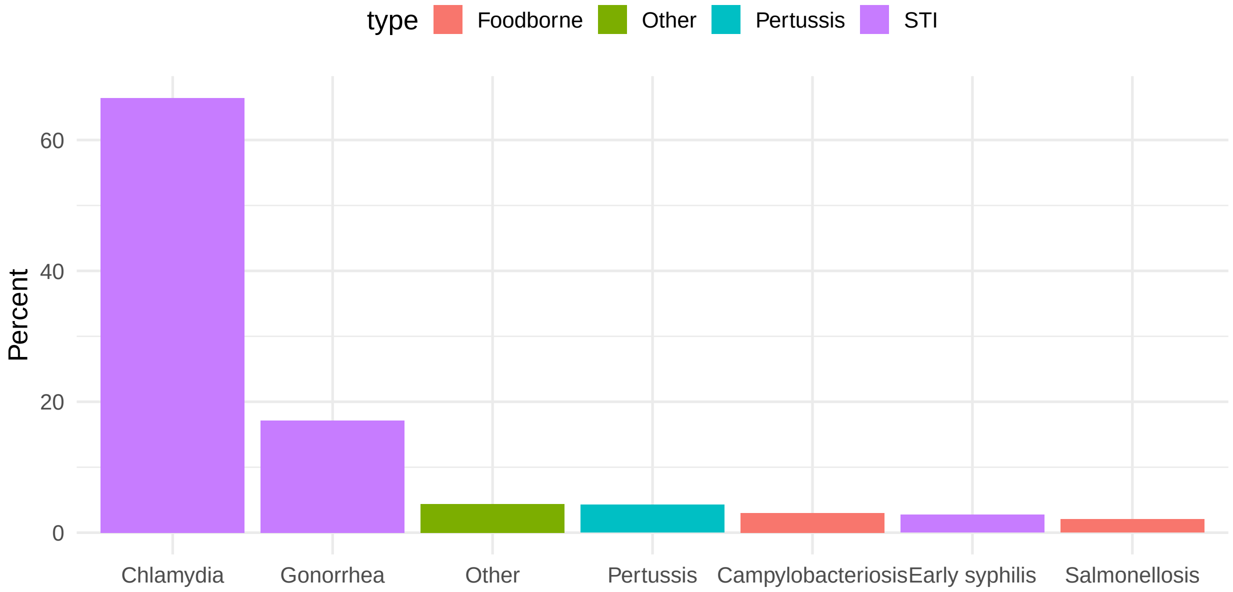
```
id_data <- id_data %>%
  mutate(disease_ordered = fct_reorder(disease, percent_cases, .desc = T))

ggplot(id_data, aes(x = disease_ordered, y = percent_cases)) +
  geom_bar(stat = "identity") +
  labs(y = "Percent", x = "") +
  theme_minimal(base_size = 15)
```



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

```r
ggplot(id_data, aes(x = disease_ordered, y = percent_cases)) +
  geom_bar(stat = "identity", aes(fill = type)) +
  labs(y = "Percent", x = "") +
  theme_minimal(base_size = 15) +
  theme(legend.position = "top")
```

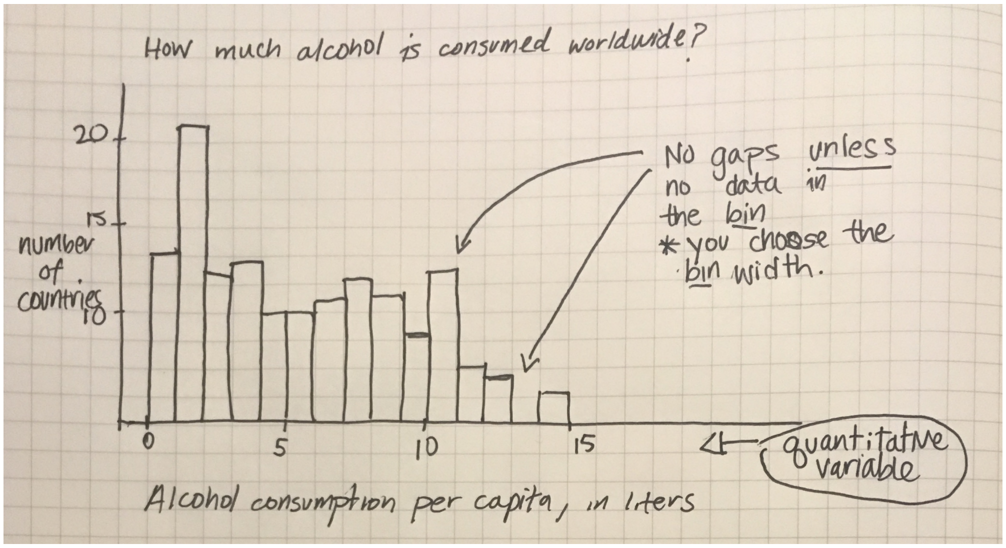


- Why do we use `fill` not `col` to shade the bars?

Visualization of quantitative data

- Histograms look a lot like bar charts, except that the bars touch because the underlying scale is numeric.

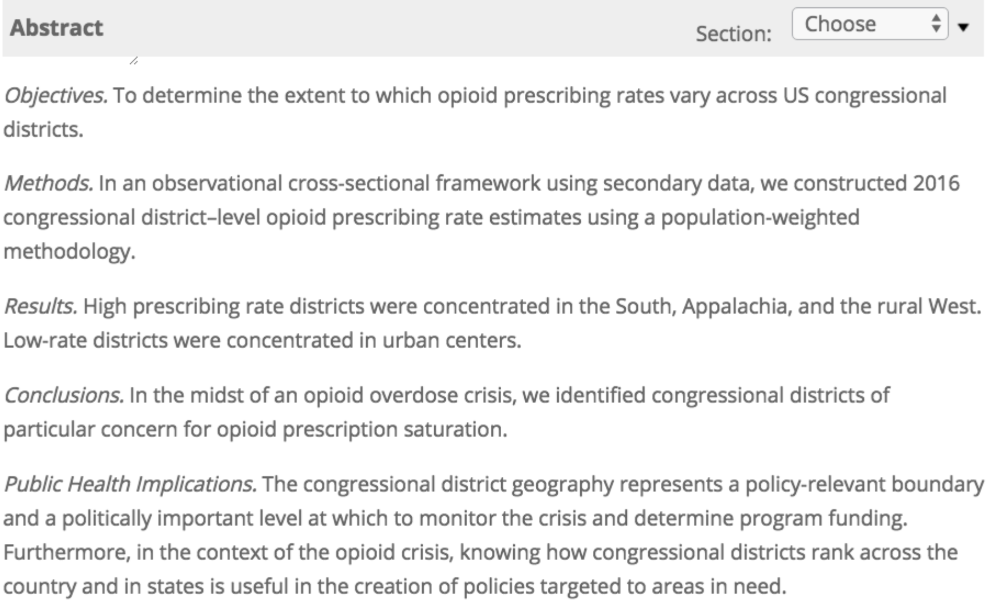
- In order to make a histogram, the underlying data needs to be **binned** into categories and the number or percent of data in each category becomes the height of each bar.



Example of opioid state prescription rates

- The data folder contains updated data from 2018. It came from the paper: "Opioid Prescribing Rates by Congressional Districts, United States, 2016", by Rolheiser et al. link

Example of opioid state prescription rates



Abstract

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

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

Results. High prescribing rate districts were concentrated in the South, Appalachia, and the rural West. Low-rate districts were concentrated in urban centers.

Conclusions. In the midst of an opioid overdose crisis, we identified congressional districts of particular concern for opioid prescription saturation.

Public Health Implications. The congressional district geography represents a policy-relevant boundary and a politically important level at which to monitor the crisis and determine program funding. Furthermore, in the context of the opioid crisis, knowing how congressional districts rank across the country and in states is useful in the creation of policies targeted to areas in need.

Example of opioid state prescription rates

```
opi_data <- read_csv("./data/Ch01_opioid-data.csv")
```

```
## Rows: 51 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): State
## dbl (7): Rank, Mean, Median, SD, Min, Max, Num_Districts
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(opi_data)
```

```
## # A tibble: 6 x 8
##    Rank State  Mean Median    SD   Min   Max Num_Districts
##   <dbl> <chr> <dbl>  <dbl> <dbl> <dbl> <dbl>         <dbl>
## 1     1 AL    121.   113.  21.9  106.   167.             7
## 2     2 AR    115.   115.   8.59 105.   126.             4
## 3     3 TN    108.   108.  19.2   73.6  133              9
## 4     4 MS    106.   106.  17.4   83.9  126.             4
## 5     5 LA     98.4   98.9 10.3   83.2  113.             6
## 6     6 KY     98.1   85.8 26.7   77.6  147              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.

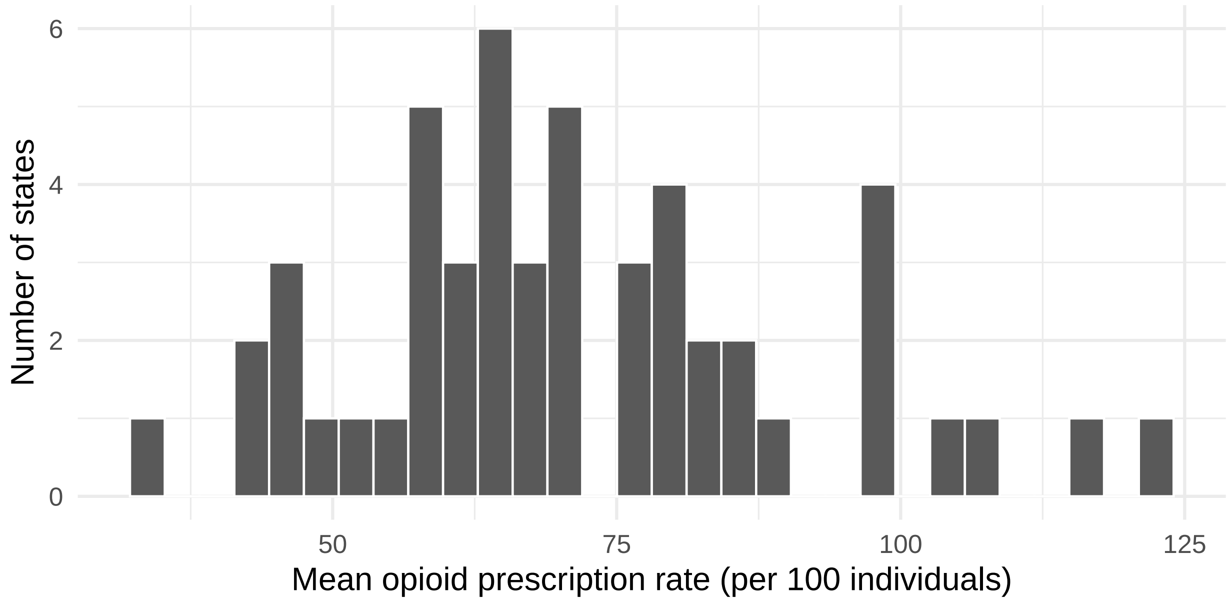
Histogram of opioid prescription rates

- 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

```
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)
```

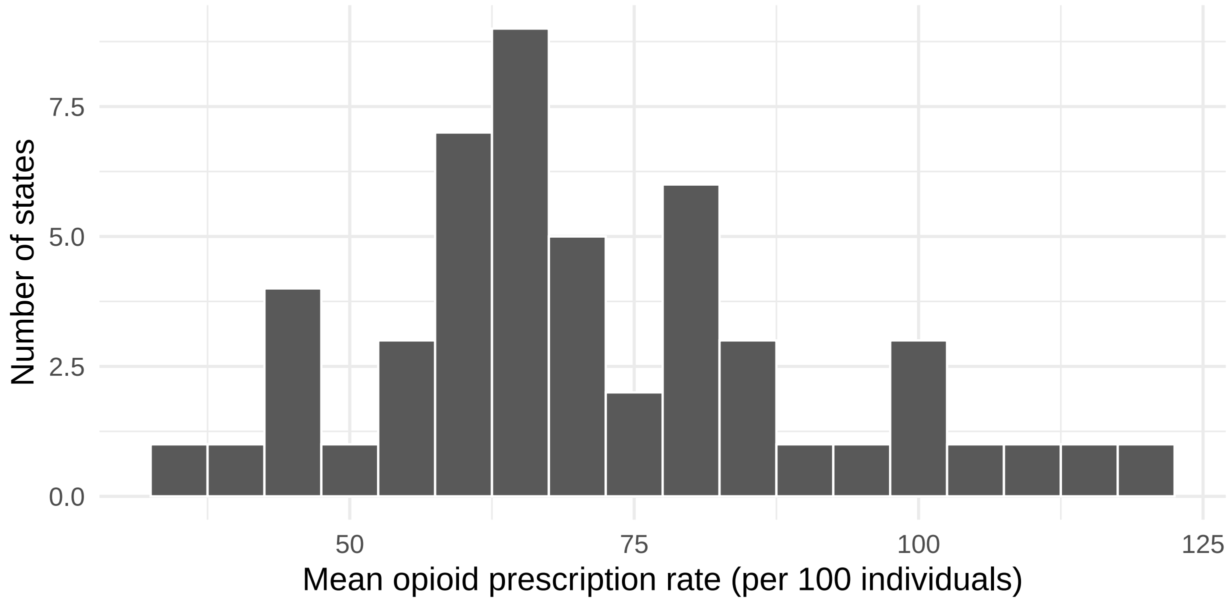
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Remember: the default number of bins is 30. R prints a message about this and reminds us to pick a `binwidth` instead.
- The `binwidth` is the number of units according to the x axis that each bin covers.
- Try making plots of different binwidths to see how it affects the display of data

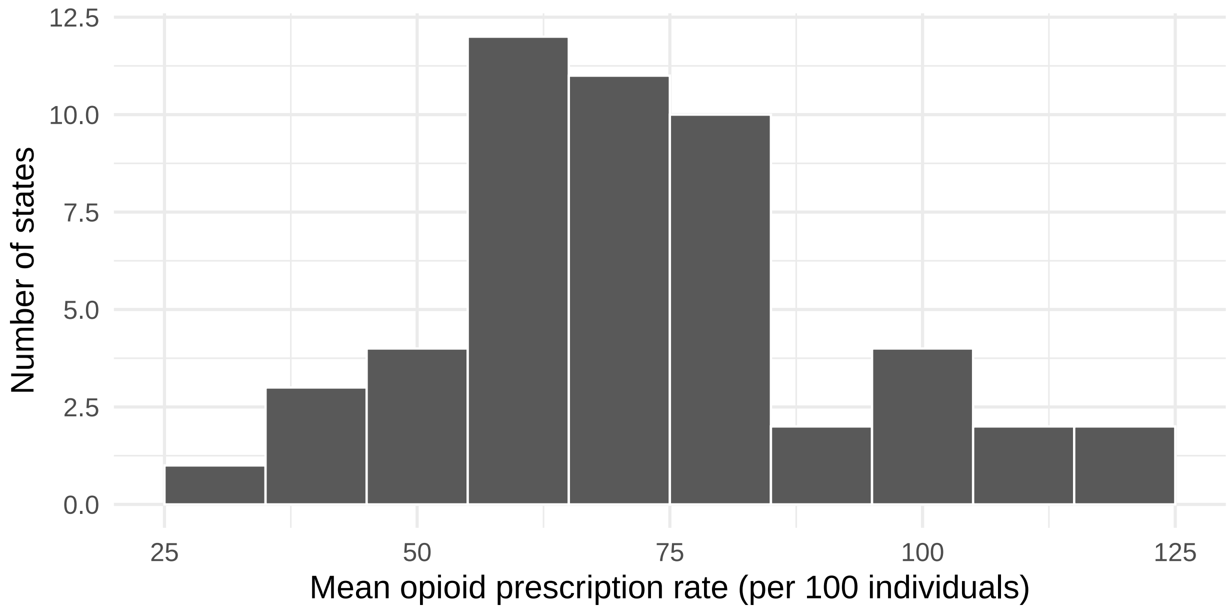
`geom_histogram(binwidth = 5)`

```r
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)
```



geom_histogram(binwidth = 10)

```r
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)
```

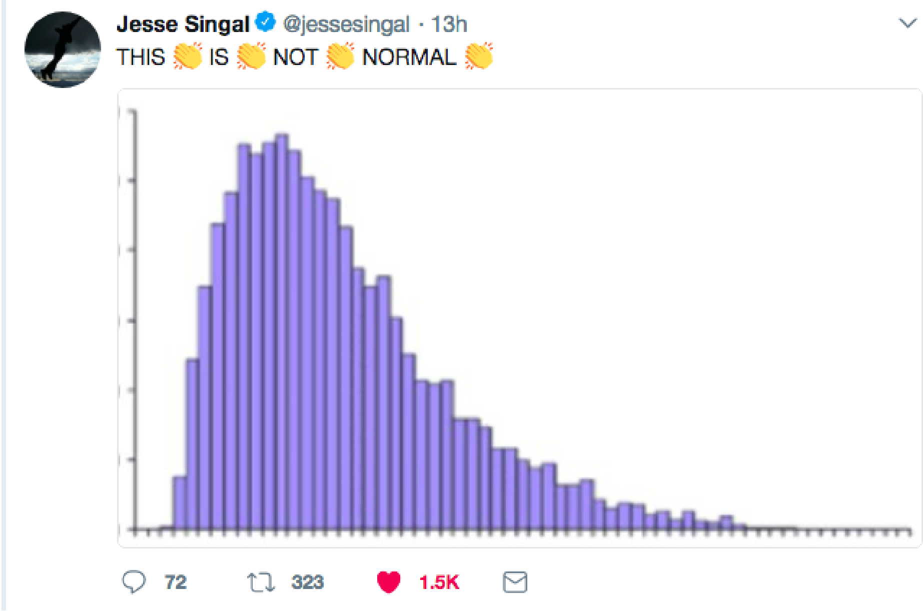


Shape, Center, Spread

- When we examine histograms, we can make comments on a distribution's:

- **Shape**: Is the distribution **symmetric** or **skewed** to the left or right? Be careful! People often confused distributions that are skewed left with skewed right and vice versa.
- **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?



Jesse Singal @jessesingal · 13h

THIS 👏 IS 👏 NOT 👏 NORMAL 👏

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 discussed today, the time plot shows a relationship between two variables:

  i) a quantitative variable
  ii) time

- Often, these plots are used to look for cycles (e.g., seasonal patterns that recur each year) or trends (e.g., overall increases or decreases seen over time).
- The life expectancy trends we plotted last lecture are time plots.

Recap: What new functions did we use?

1. `geom_bar(stat = "identity")` to make a bar chart when you specify the y variable
2. `geom_histogram()` to make a histogram for which ggplot needs to calculate the count
3. `fct_reorder(var1, var2)` to reorder a categorical variable (`var1`) by a numeric variable (`var2`)

   - from the `forcats` package