Probability mass functions

```
library(tidyverse)
county_complete <- read_rds(
  path = url("http://spring18.cds101.com/files/datasets/county_complete.rds"))
```

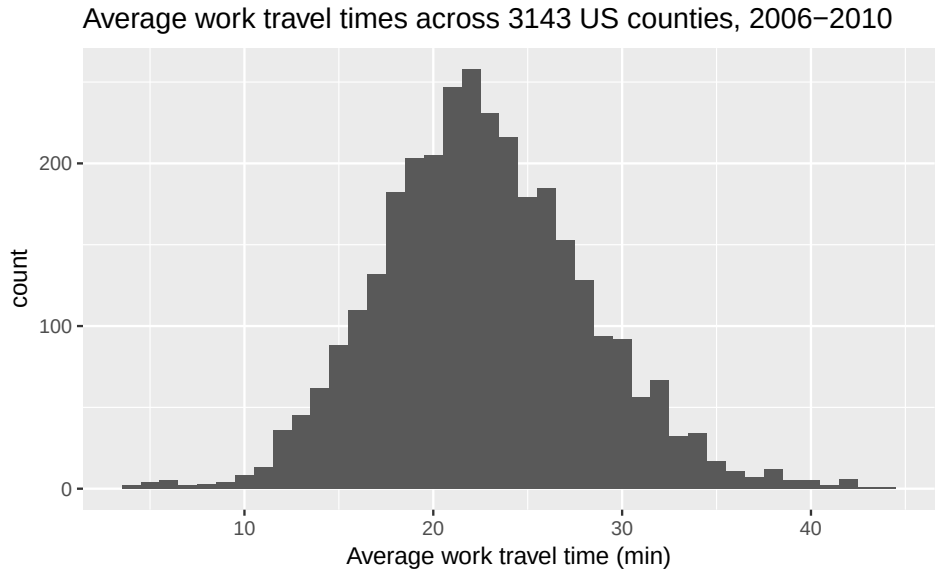
Statistical distributions

During the first part of the course, we learned how to use visualizations to explore a dataset. Now we will extend this approach using concepts from probability and statistics to build a scientific foundation for interpreting data distributions. We start with discussing univariate (single variable) distributions, which we've previously visualized as frequency histograms (by default, `geom_histogram()` sorts data into different bins and tells you how many end up in each one). Frequency histograms are useful for examining the particulars of a single variable, but have limited utility when directly comparing distributions that contain different numbers of observations. Here we introduce the normalized version of the frequency histogram, the **probability mass function** (PMF).

Example dataset

We use an example dataset of the average time it takes for people to commute to work across 3143 counties in the United States (collected between 2006-2010) to help illustrate the meaning and uses of the probability mass function. The frequency histogram for these times can be plotted using the following code snippet:

```
county_complete %>%
  ggplot(mapping = aes(x = mean_work_travel)) +
  geom_histogram(binwidth = 1)
```



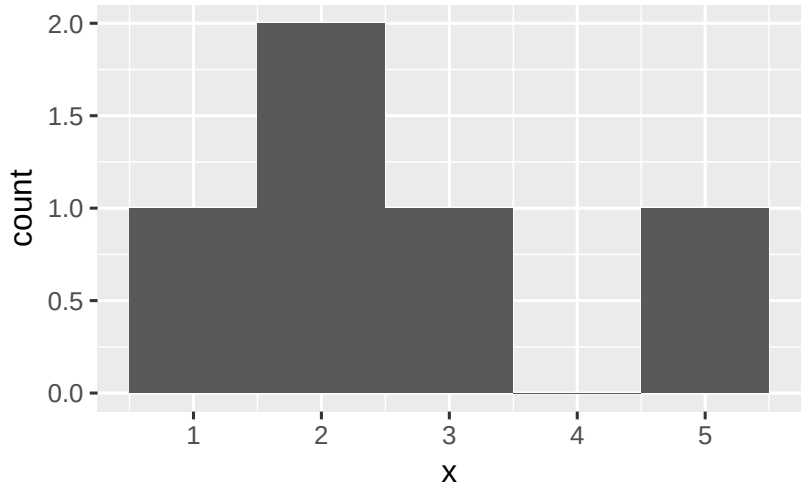
PMFs

The **probability mass function** (PMF) represents a distribution by sorting the data into bins (much like the frequency histogram) and then associates a probability with each bin in the distribution. A **probability** is a frequency expressed as a fraction of the sample size *n*. Therefore we can directly convert a frequency histogram to a PMF by dividing the count in each bin by the sample size *n*. This process is called **normalization**.

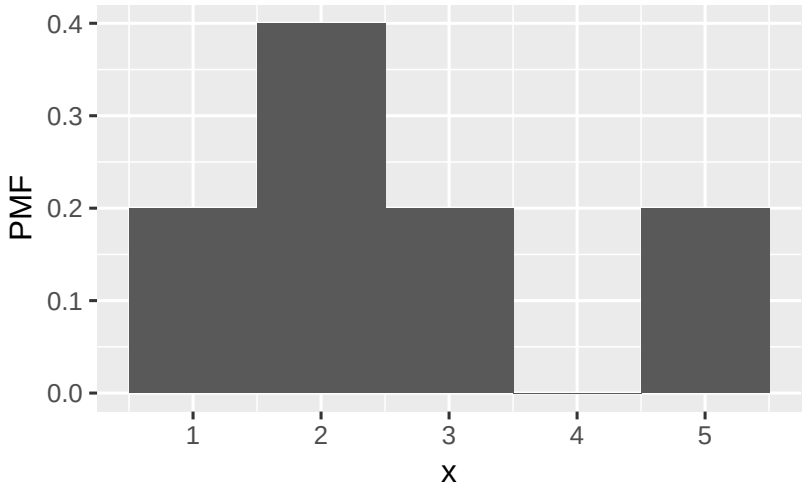
As an example, consider the following short sample,

1 2 2 3 5

If we choose a binwidth of 1, then we get a frequency histogram that looks like this:



There are 5 observations in this sample. So, we can convert to a PMF by dividing the count within each bin by 5, getting a histogram that looks like this:

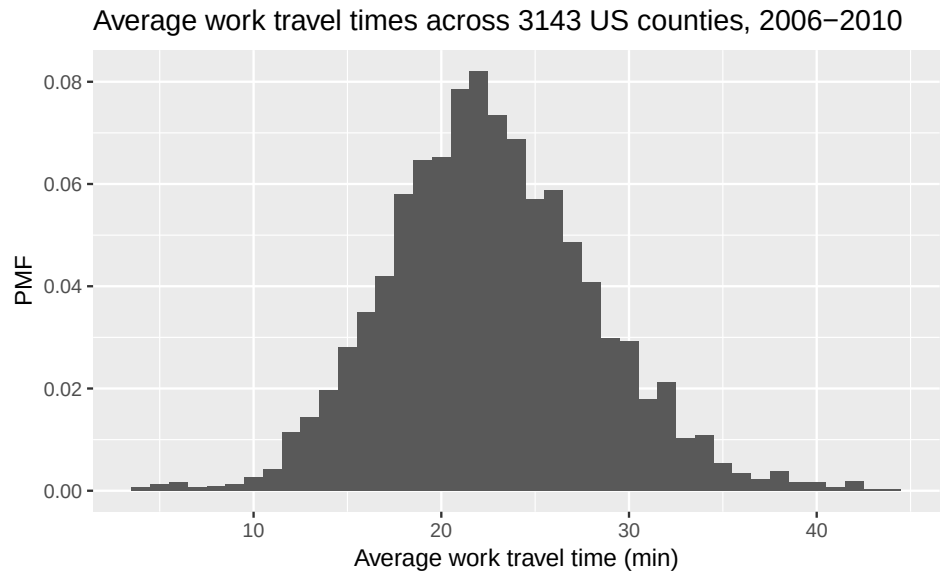


The relative shape stays the same, but compare the values along the vertical axis between the two figures. You'll find that they are no longer integers and are instead probabilities. The normalization procedure (dividing by 5) guarantees that adding together the probabilities of all bins will equal 1. For this example, we find that the probability of drawing the number 1 is 0.2, drawing 2 is 0.4, drawing 3 is 0.2, drawing 4 is 0, and drawing 5 is 0.2. That is the biggest difference between a frequency histogram and a PMF, the frequency histogram maps from values to integer counters, while the PMF maps from values to fractional probabilities.

Plotting PMFs

The syntax for plotting a PMF using `ggplot2` is nearly identical to what you would use to create a frequency histogram. The one modification is that you need to include `y = ..density..` inside `aes()`. As a simple example, let's take the full distribution of the average work travel times from earlier and plot it as a PMF:

```
county_complete %>%
  ggplot(mapping = aes(x = mean_work_travel, y = ..density..)) +
  geom_histogram(binwidth = 1)
```

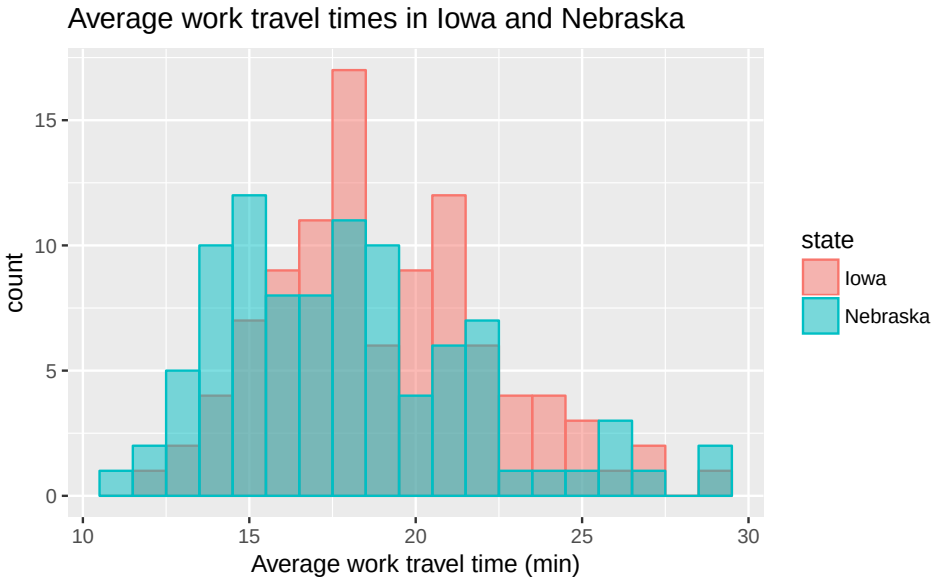


Let's do a comparison to show how one might use a PMF for analysis. For example, we could ask if two midwestern states such as Nebraska and Iowa have the same distribution of work travel times, or if there is a meaningful difference between the two. First, let's filter the dataset to only include these two states:

```
nebraska_iowa <- county_complete %>%
  filter(state == "Iowa" | state == "Nebraska")
```

Now let's plot the frequency histogram:

```
nebraska_iowa %>%
  ggplot() +
  geom_histogram(
    mapping = aes(x = mean_work_travel, fill = state, color = state),
    position = "identity", alpha = 0.5, binwidth = 1)
```



The `position = "identity"` input overlaps the two distributions (instead of stacking them) and `alpha = 0.5` makes the distributions translucent, so that you can see both despite the overlap. On our first glance, it looks like

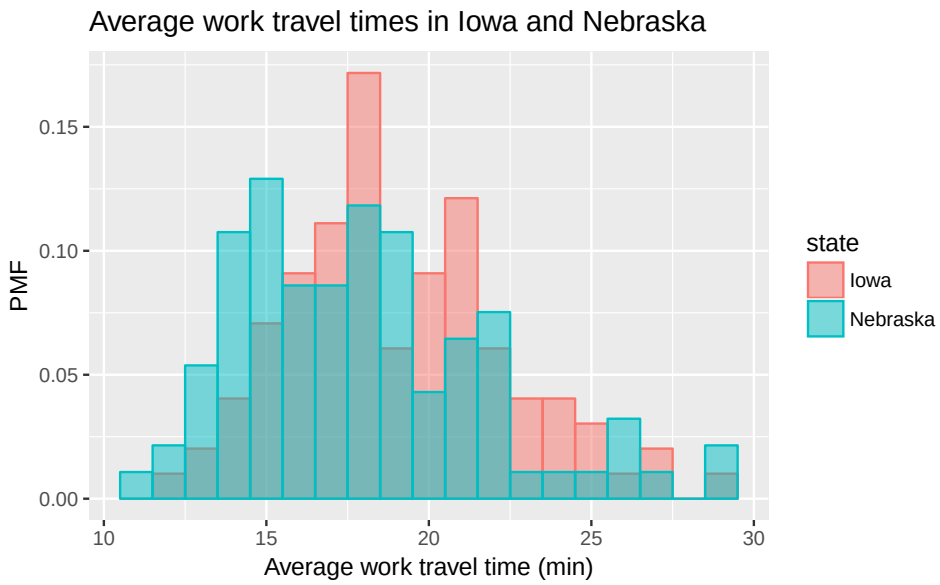
the center of the Nebraska times is lower than the center of the Iowa times, and that both have a long tail on the right-hand side. However, if we do a count summary,

```
nebraska_iowa %>%
  count(state)
```

state	n
Iowa	99
Nebraska	93

we find that the two states do not have the exact same number of counties, although they are close in this particular example. Nonetheless, any comparisons should be done using a PMF in order to account for differences in the sample size. We use the following code to create a PMF plot:

```
nebraska_iowa %>%
  ggplot() +
  geom_histogram(
    mapping = aes(x = mean_work_travel, y = ..density..,
                  fill = state, color = state),
    position = "identity", alpha = 0.5, binwidth = 1)
```



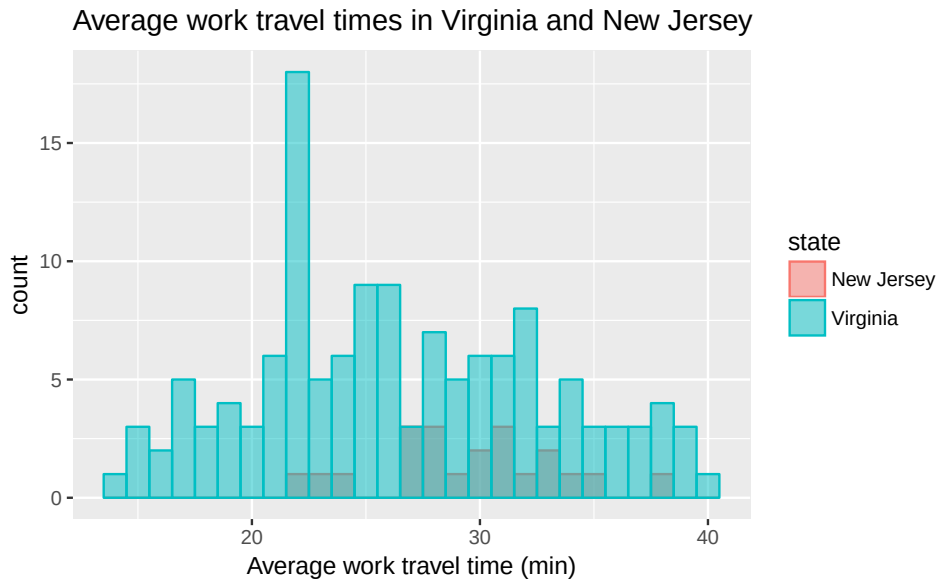
The trend that the center of the travel times in Nebraska is slightly smaller than in Iowa continues to hold even after converting to a PMF.

To provide an example where a PMF is clearly necessary, what if we compare New Jersey with Virginia? Virginia has many more counties than New Jersey:

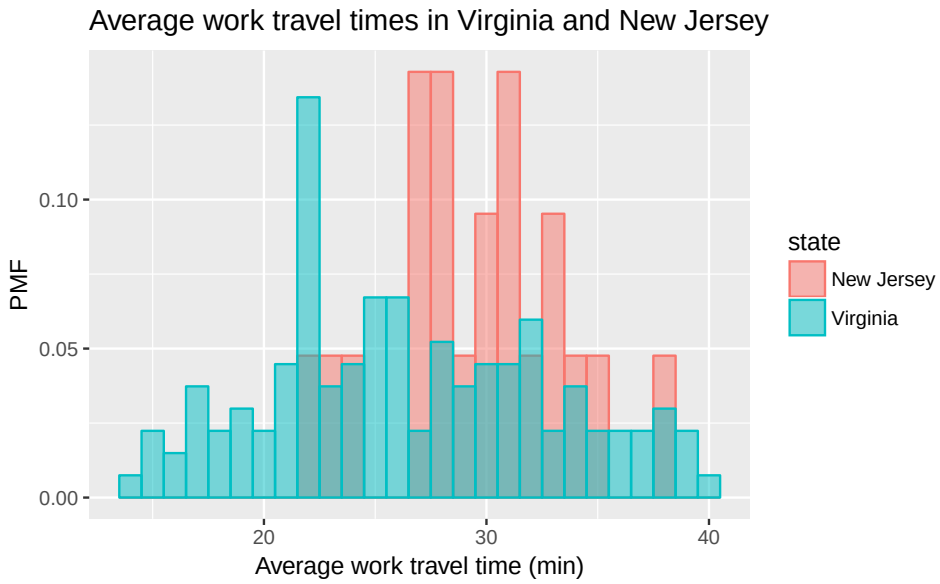
```
county_complete %>%
  filter(state == "New Jersey" | state == "Virginia") %>%
  count(state)
```

state	n
New Jersey	21
Virginia	134

As a result, comparing their frequency histograms gives you this:



The New Jersey distribution is dwarfed by the Virginia distribution and it makes it difficult to make comparisons. However, if we instead compare PMFs, we get this:



So, for example, we can now make statements like "a randomly selected resident in New Jersey is twice as likely as a randomly chosen resident in Virginia to have an average work travel time of 30 minutes." The PMF allows for an "apples-to-apples" comparison of the average travel times.

Credits