Chapter 4

Summarizing data

In this Chapter we will discuss why and how to summarize data.

4.1 Why summarize data?

When we summarize data, we are necessarily throwing away information, and there are many conceivable objections to this. As an example, let's go back to the PURE study that we discussed in Chapter 1. Are we not supposed to believe that all of the details about each individual matter, beyond those that are summarized in the dataset? What about the specific details of how the data were collected, such as the time of day or the mood of the participant? All of these details are lost when we summarize the data.

We summarize data in general because it provides us with a way to *generalize* - that is, to make general statements that extend beyond specific observations. The importance of generalization was highlighted by the writer Jorge Luis Borges in his short story "Funes the Memorious", which describes an individual who loses the ability to forget. Borges focuses in on the relation between generalization (i.e. throwing away data) and thinking: "To think is to forget a difference, to generalize, to abstract. In the overly replete world of Funes, there were nothing but details."

Psychologists have long studied all of the ways in which generalization is central to thinking. One example is categorization: We are able to easily recognize different examples of the category of "birds" even though the individual examples may be very different in their surface features (such as an ostrich, a robin, and a chicken). Importantly, generalization lets us make predictions about these individuals – in the case of birds, we can predict that they can fly and eat worms, and that they probably can't drive a car or speak English. These predictions won't always be right, but they are often good enough to be useful in the world.

4.2 Summarizing data using tables

A simple way to summarize data is to generate a table representing counts of various types of observations. This type of table has been used for thousands of years (see Figure 4.1).

Let's look at some examples of the use of tables, again using the NHANES dataset. Type the command `help(NHANES)` in the RStudio console, and scroll through the help page, which should open within the Help panel if you are using RStudio. This page provides some information about the dataset as well as a listing of all of the variables included in the dataset. Let's have a look at a simple variable, called "PhysActive" in the dataset. This variable contains one of three different values: "Yes" or "No" (indicating whether or not the person reports doing "moderate or vigorous-intensity sports, fitness or recreational activities"), or "NA" if the data are missing for that individual. There are different reasons that the data might be missing; for

Figure 4.1: A Sumerian tablet from the Louvre, showing a sales contract for a house and field. Public domain, via Wikimedia Commons.

example, this question was not asked of children younger than 12 years of age, while in other cases an adult may have declined to answer the question during the interview.

4.2.1 Frequency distributions

Let's look at how many people fall into each of these categories. Don't worry right now about exactly how R is doing this; we will come back to that later.

```
# summarize physical activity data

PhysActive_table <- NHANES %>%
  dplyr::select(PhysActive) %>%
  group_by(PhysActive) %>%
  summarize(AbsoluteFrequency = n())

pander(PhysActive_table)
```

PhysActive	AbsoluteFrequency
No	2473
Yes	2972
NA	1334

The R code in this cell generates a table showing the frequencies of each of the different values; there were 2473 individuals who responded "No" to the question, 2972 who responded "Yes", and 1334 for whom no response was given. We call this a *frequency distribution* because it tells us how each of the values is distributed across the sample.

Since we only want to work with people who gave an answer to the question, let's filter the dataset to only include individuals who responded to this question.

```
# summarize physical activity data after dropping NA values using drop_na()

NHANES %>%
  drop_na(PhysActive) %>%
  dplyr::select(PhysActive) %>%
  group_by(PhysActive) %>%
  summarize(AbsoluteFrequency = n()) %>%
  pander()
```

```
# compute relative frequency of physical activity categories

NHANES %>%
  drop_na(PhysActive) %>%
  dplyr::select(PhysActive) %>%
  group_by(PhysActive) %>%
  summarize(AbsoluteFrequency = n()) %>%
  mutate(RelativeFrequency = AbsoluteFrequency / sum(AbsoluteFrequency)) %>%
  pander()
```

PhysActive	AbsoluteFrequency	RelativeFrequency
No	2473	0.454
Yes	2972	0.546

The relative frequency provides a much easier way to see how big the imbalance is. We can also interpret the relative frequencies as percentages by multiplying them by 100:

```
# compute percentages for physical activity categories

PhysActive_table_filtered <- NHANES %>%
  drop_na(PhysActive) %>%
  dplyr::select(PhysActive) %>%
  group_by(PhysActive) %>%
  summarize(AbsoluteFrequency = n()) %>%
  mutate(
    RelativeFrequency = AbsoluteFrequency / sum(AbsoluteFrequency),
    Percentage = RelativeFrequency * 100
  )

pander(PhysActive_table_filtered)
```

PhysActive	AbsoluteFrequency	RelativeFrequency	Percentage
No	2473	0.454	45.418
Yes	2972	0.546	54.582

This lets us see that 45.42 percent of the individuals in the NHANES sample said “No” and 54.58 percent said “Yes”.

4.2.2 Cumulative distributions

The PhysActive variable that we examined above only had two possible values, but often we wish to summarize data that can have many more possible values. When those values are at least ordinal, then one useful way to summarize them is via what we call a *cumulative* frequency representation: rather than asking how many observations take on a specific value, we ask how many have a value of *at least* some specific value.

Let’s look at another variable in the NHANES dataset, called SleepHrsNight which records how many hours the participant reports sleeping on usual weekdays. Let’s create a frequency table as we did above, after removing anyone who didn’t provide a response to the question.

```
# create summary table for relative frequency of different
# values of SleepHrsNight
```

```
NHANES %>%
  drop_na(SleepHrsNight) %>%
  dplyr::select(SleepHrsNight) %>%
  group_by(SleepHrsNight) %>%
  summarize(AbsoluteFrequency = n()) %>%
  mutate(
    RelativeFrequency = AbsoluteFrequency / sum(AbsoluteFrequency),
    Percentage = RelativeFrequency * 100
  ) %>%
  pander()
```

SleepHrsNight	AbsoluteFrequency	RelativeFrequency	Percentage
2	9	0.002	0.179
3	49	0.01	0.973
4	200	0.04	3.972
5	406	0.081	8.064
6	1172	0.233	23.277
7	1394	0.277	27.686
8	1405	0.279	27.905
9	271	0.054	5.382
10	97	0.019	1.927
11	15	0.003	0.298
12	17	0.003	0.338

We can already begin to summarize the dataset just by looking at the table; for example, we can see that most people report sleeping between 6 and 8 hours. Let's plot the data to see this more clearly. To do this we can plot a *histogram* which shows the number of cases having each of the different values; see left panel of Figure 4.2. The ggplot2() library has a built in histogram function (`geom_histogram()`) which we will often use. We can also plot the relative frequencies, which we will often refer to as *densities* - see the right panel of Figure 4.2.

What if we want to know how many people report sleeping 5 hours or less? To find this, we can compute a *cumulative distribution*:

$$cumulative\ frequency_j = \sum_{i=1}^{j} absolute\ frequency_i$$

That is, to compute the cumulative frequency for some value j, we add up the frequencies for all of the values up to and including j. Let's do this for our sleep variable, first for the absolute frequency:

```
# create cumulative frequency distribution of SleepHrsNight data

SleepHrsNight_cumulative <-
  NHANES %>%
  drop_na(SleepHrsNight) %>%
  dplyr::select(SleepHrsNight) %>%
  group_by(SleepHrsNight) %>%
  summarize(AbsoluteFrequency = n()) %>%
  mutate(CumulativeFrequency = cumsum(AbsoluteFrequency))

pander(SleepHrsNight_cumulative)
```

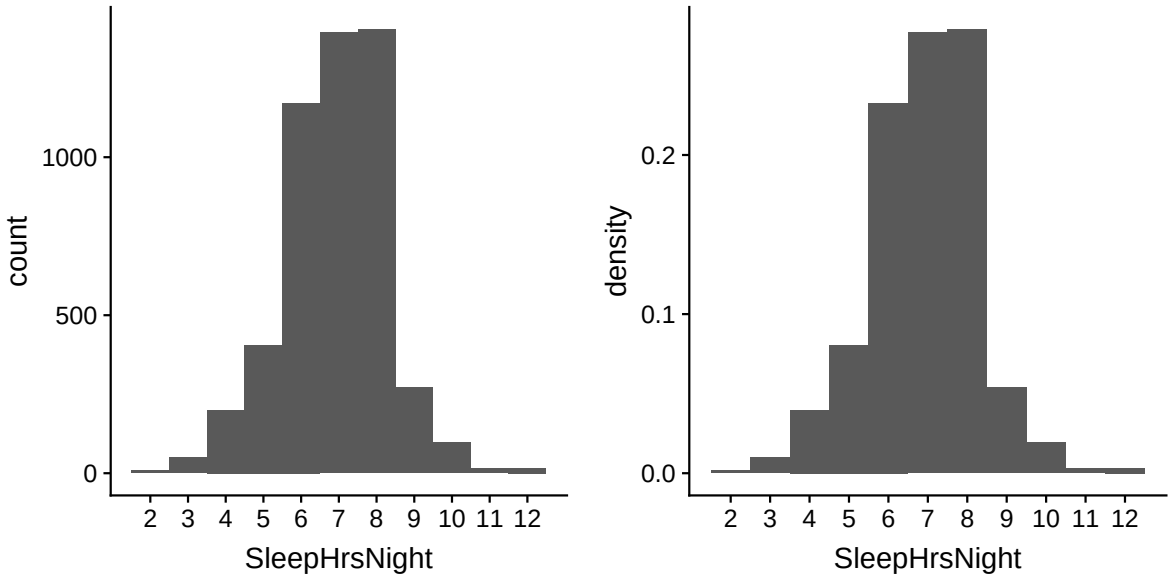

Figure 4.2: Left: Histogram showing the number (left) and proportion (right) of people reporting each possible value of the SleepHrsNight variable.

SleepHrsNight	AbsoluteFrequency	CumulativeFrequency
2	9	9
3	49	58
4	200	258
5	406	664
6	1172	1836
7	1394	3230
8	1405	4635
9	271	4906
10	97	5003
11	15	5018
12	17	5035

In the left panel of Figure 4.3 we plot the data to see what these representations look like; the absolute frequency values are plotted in red, and the cumulative frequencies are plotted in blue. We see that the cumulative frequency is *monotonically increasing* – that is, it can only go up or stay constant, but it can never decrease. Again, we usually find the relative frequencies to be more useful than the absolute; those are plotted in the right panel of Figure 4.3.

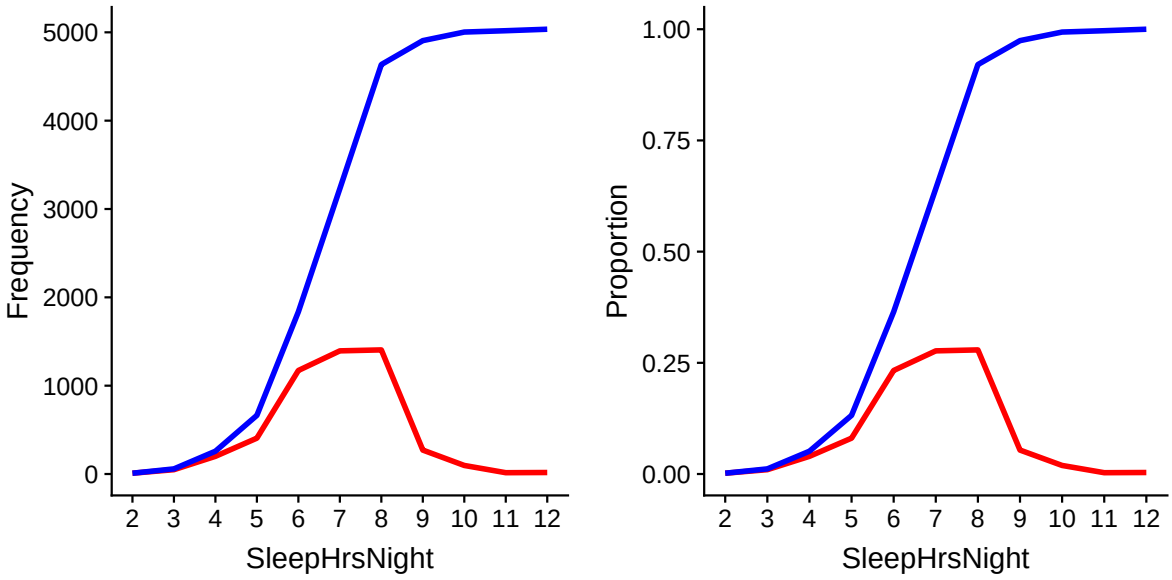

Figure 4.3: A plot of the relative (red) and cumulative relative (blue) values for frequency (left) and proportion (right) for the possible values of SleepHrsNight.

4.2.3 Plotting histograms

The variables that we examined above were fairly simple, having only a few possible values. Now let's look at a more complex variable: Age. First let's plot the Age variable for all of the individuals in the NHANES dataset (see left panel of Figure 4.4). What do you see there? First, you should notice that the number of individuals in each age group is declining over time. This makes sense because the population is being randomly sampled, and thus death over time leads to fewer people in the older age ranges. Second, you probably notice a large spike in the graph at age 80. What do you think that's about?

If you look at the help function for the NHANES dataset, you will see the following definition: "Age in years at screening of study participant. Note: Subjects 80 years or older were recorded as 80." The reason for this is that the relatively small number of individuals with very high ages would make it potentially easier to identify the specific person in the dataset if you knew their exact age; researchers generally promise their participants to keep their identity confidential, and this is one of the things they can do to help protect their research subjects. This also highlights the fact that it's always important to know where one's data have come from and how they have been processed; otherwise we might interpret them improperly.

Let's look at another more complex variable in the NHANES dataset: Height. The histogram of height values is plotted in the right panel of Figure 4.4. The first thing you should notice about this distribution is that most of its density is centered around about 170 cm, but the distribution has a "tail" on the left; there are a small number of individuals with much smaller heights. What do you think is going on here?

You may have intuited that the small heights are coming from the children in the dataset. One way to examine this is to plot the histogram with separate colors for children and adults (left panel of Figure 4.5). This shows that all of the very short heights were indeed coming from children in the sample. Let's create a new version of NHANES that only includes adults, and then plot the histogram just for them (right panel of Figure 4.5). In that plot that the distribution looks much more symmetric. As we will see later, this is a nice example of a *normal* (or *Gaussian*) distribution.

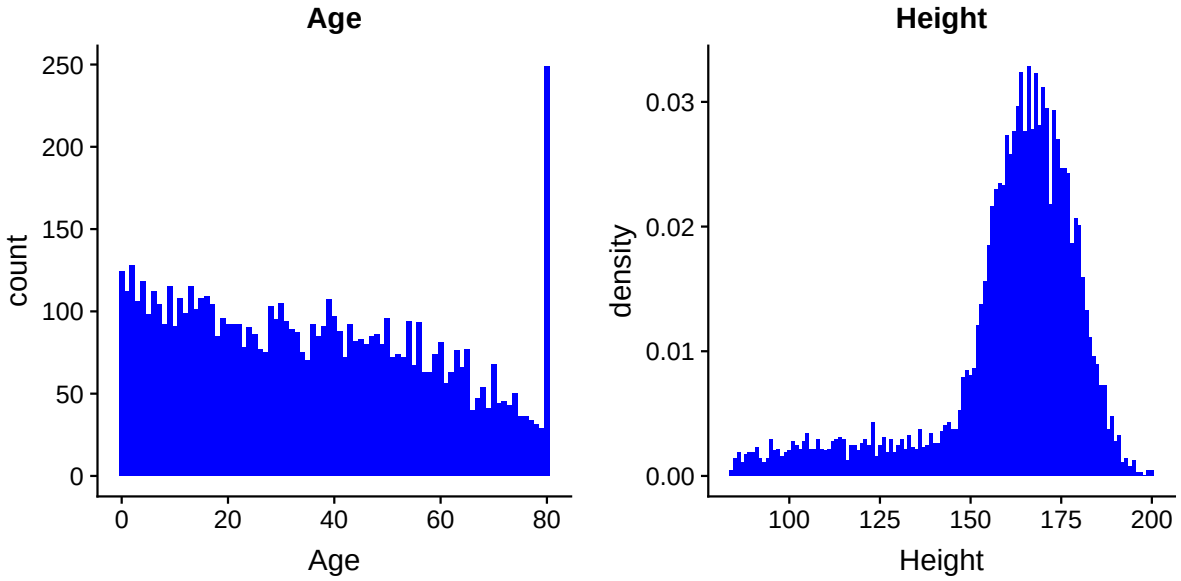

Figure 4.4: A histogram of the Age (left) and Height (right) variables in NHANES.

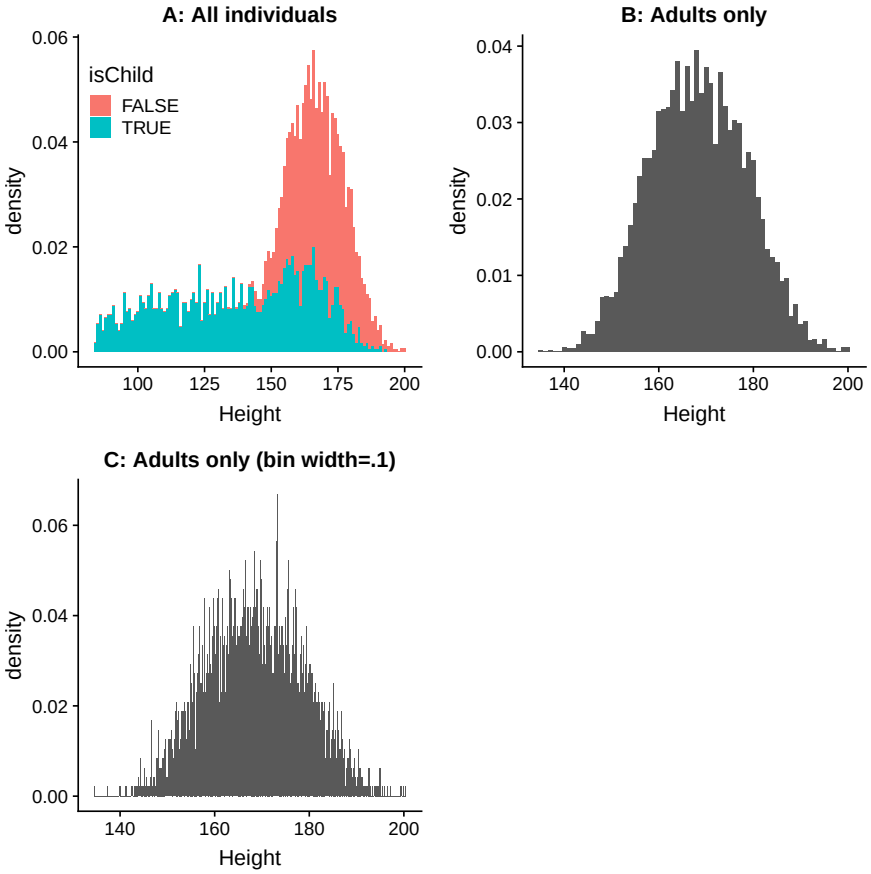

Figure 4.5: Histogram of heights for NHANES. A: values plotted separately for children (blue) and adults (red). B: values for adults only. C: Same as B, but with bin width = 0.1

4.2.4 Histogram bins

In our earlier example with the sleep variable, the data were reported in whole numbers, and we simply counted the number of people who reported each possible value. However, if you look at a few values of the Height variable in NHANES, you will see that it was measured in centimeters down to the first decimal place:

```
# take a slice of a few values from the full data frame
NHANES_adult %>%
  dplyr::select(Height) %>%
  slice(45:50) %>%
  pander()
```

Height
169.6
169.8
167.5
155.2
173.8
174.5

Panel C of Figure 4.5 shows a histogram that counts the density of each possible value. That histogram looks really jagged, which is because of the variability in specific decimal place values. For example, the value 173.2 occurs 32 times, while the value 173.3 only occurs 15 times. We probably don't think that there is really such a big difference between the prevalence of these two weights; more likely this is just due to random variability in our sample of people.

In general, when we create a histogram of data that are continuous or where there are many possible values, we will *bin* the values so that instead of counting and plotting the frequency of every specific value, we count and plot the frequency of values falling within a specific range. That's why the plot looked less jagged above in Panel B of 4.5; if you look at the `geom_histogram` command you will see that we set "binwidth = 1" which told the command to compute the histogram by combining values within bins with a width of one; thus, the values 1.3, 1.5, and 1.6 would all count toward the frequency of the same bin, which would span from values equal to one up through values less than 2.

Note that once the bin size has been selected, then the number of bins is determined by the data:

$$number\,of\,bins = \frac{range\,of\,scores}{bin\,width}$$

There is no hard and fast rule for how to choose the optimal bin width. Occasionally it will be obvious (as when there are only a few possible values), but in many cases it would require trial and error. There are methods that try to find an optimal bin size, such as the Freedman-Diaconis method that is implemented within the `nclass.FD()` function in R; we will use this function in some of the examples below.

4.3 Idealized representations of distributions

Datasets are like snowflakes, in that every one is different, but nonetheless there are patterns that one often sees in different types of data. This allows us to use idealized representations of the data to further summarize them. Let's take the adult height data plotted in 4.5, and plot them alongside a very different variable: pulse rate (heartbeats per minute), also measured in NHANES (see Figure 4.6).

While these plots certainly don't look exactly the same, both have the general characteristic of being relatively symmetric around a rounded peak in the middle. This shape is in fact one of the commonly observed shapes

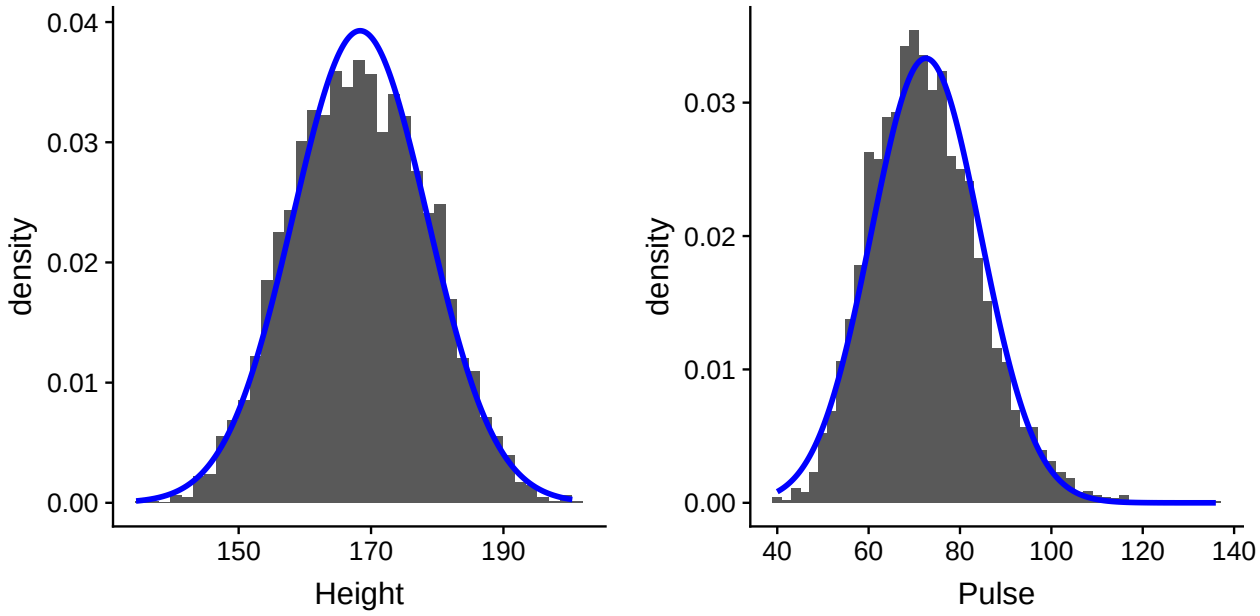

Figure 4.6: Histograms for height (left) and pulse (right) in the NHANES dataset, with the normal distribution overlaid for each dataset.

of distributions when we collect data, which we call the *normal* (or *Gaussian*) distribution. This distribution is defined in terms of two values (which we call *parameters* of the distribution): the location of the center peak (which we call the *mean*) and the width of the distribution (which is described in terms of a parameter called the *standard deviation*). Figure 4.6 shows the appropriate normal distribution plotted on top of each of the histrograms.You can see that although the curves don't fit the data exactly, they do a pretty good job of characterizing the distribution – with just two numbers!

As we will see later in the course when we discuss the central limit theorem, there is a deep mathematical reason why many variables in the world exhibit the form of a normal distribution.

4.3.1 Skewness

The examples in Figure 4.6 followed the normal distribution fairly well, but in many cases the data will deviate in a systematic way from the normal distribution. One way in which the data can deviate is when they are asymmetric, such that one tail of the distribution is more dense than the other. We refer to this as "skewness". Skewness commonly occurs when the measurement is constrained to be non-negative, such as when we are counting things or measuring elapsed times (and thus the variable can't take on negative values).

An example of skewness can be seen in the average waiting times at the airport security lines at San Francisco International Airport, plotted in the left panel of Figure 4.7. You can see that while most wait times are less than 20 minutes, there are a number of cases where they are much longer, over 60 minutes! This is an example of a "right-skewed" distribution, where the right tail is longer than the left; these are common when looking at counts or measured times, which can't be less than zero. It's less common to see "left-skewed" distributions, but they can occur, for example when looking at fractional values that can't take a value greater than one.

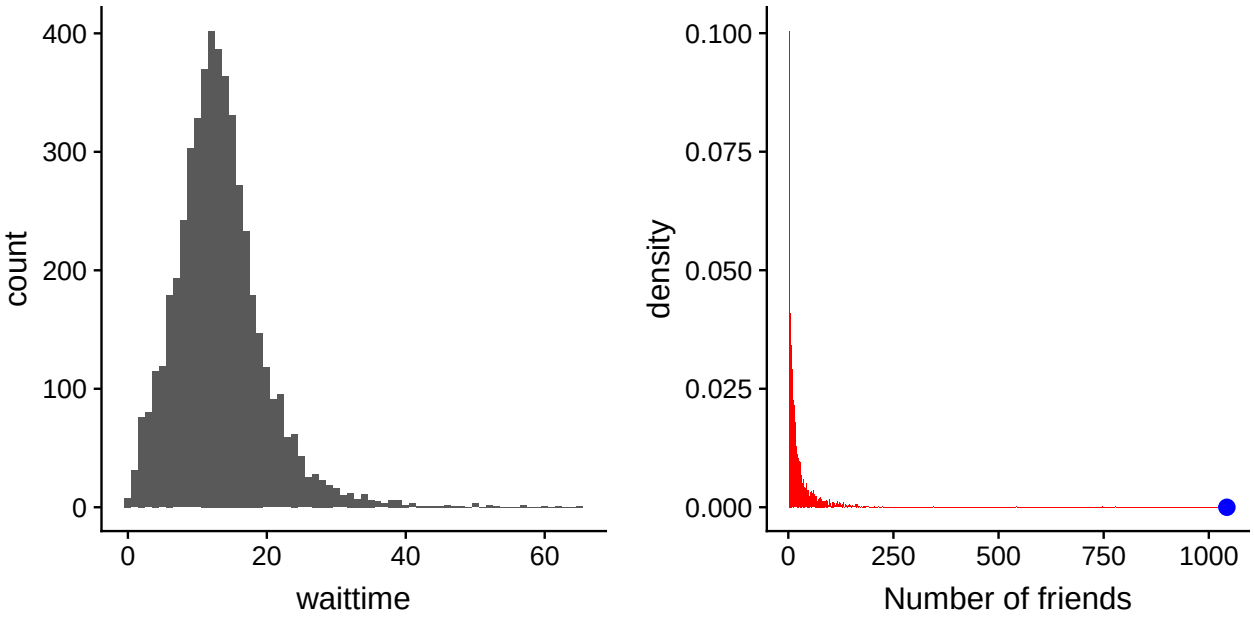

Figure 4.7: Examples of right-skewed and long-tailed distributions. Left: Average wait times for security at SFO Terminal A (Jan-Oct 2017), obtained from https://awt.cbp.gov/ . Right: A histogram of the number of Facebook friends amongst 3,663 individuals, obtained from the Stanford Large Network Database. The person with the maximum number of friends is indicated by the blue dot.

4.3.2 Long-tailed distributions

Historically, statistics has focused heavily on data that are normally distributed, but there are many data types that look nothing like the normal distribution. In particular, many real-world distributions are "long-tailed", meaning that the right tail extends far beyond the most typical members of the distribution. One of the most interesting types of data where long-tailed distributions occur arise from the analysis of social networks. For an example, let's look at the Facebook friend data from the Stanford Large Network Database and plot the histogram of number of friends across the 3,663 people in the database (see right panel of Figure 4.7). As we can see, this distribution has a very long right tail – the average person has 24.09 friends, while the person with the most friends (denoted by the blue dot) has 1043!

Long-tailed distributions are increasingly being recognized in the real world. In particular, many features of complex systems are characterized by these distributions, from the frequency of words in text, to the number of flights in and out of airports, to the connectivity of brain networks. There are a number of different ways that long-tailed distributions can come about, but a common one occurs in cases of the so-called "Matthew effect" from the Christian Bible:

> For to every one who has will more be given, and he will have abundance; but from him who has not, even what he has will be taken away. — Matthew 25:29, Revised Standard Version

often paraphrased as "the rich get richer". In these situations, advantages compound, such that those with more friends have access to even more new friends, and those with more money have the ability do things that increase their riches even more.

As the course progresses we will see several examples of long-tailed distributions, and we should keep in mind that many of the tools in statistics can fail when faced with long-tailed data. As Nassim Nicholas Taleb pointed out in his book "The Black Swan", such long-tailed distributions played a critical role in the 2008 financial crisis, because many of the financial models used by traders assumed that financial systems would follow the normal distribution, which they clearly did not.

4.4 Suggested readings

- *The Black Swan: The Impact of the Highly Improbable*, by Nassim Nicholas Taleb