2.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
average outcome for the treatment group ($\bar{Y}_{\text{treatment group}}$)	average observed outcome for the individuals who received the treatment (after the treatment)	in the STAR dataset, the average reading score of the students who attended a small class was about 632.7 points
average outcome for the control group ($\bar{Y}_{\text{control group}}$)	average observed outcome for the individuals who did not receive the treatment (after no treatment)	in the STAR dataset, the average reading score of the students who attended a regular-size class was about 625.49 points
experimental data	data from a randomized experiment	since Project STAR was a randomized experiment, the data we analyze in this chapter are experimental data
observational data	data collected about naturally occurring events, in which treatment was received or not received without the intervention of researchers	data on class sizes and student performance from districts where the size of the classes varies as a result of factors such as school budgets, student enrollment, or the physical limitations of the school buildings
observational study	type of study that analyzes observational data	(see previous entry)

2.7.2 R SYMBOLS AND OPERATORS

code	description	example(s)
==	relational operator used to test whether the observations of a variable are equal to a particular value; values should be in quotes if text but without quotes if numbers (see)	`data$variable==1` `data$variable=="yes"`
$	character used to identify an element inside an object, such as a variable inside a dataframe, either to access it or to create it; to its left, we specify the name of the object where the dataframe is stored (without quotes); to its right, we specify the name of the element or variable (without quotes)	`data$variable` # identifies the variable named *variable* inside the dataframe stored in the object named *data*
[]	operator used to extract a selection of observations from a variable; to its left, we specify the variable we want to subset; inside the square brackets, we specify the criteria of selection; for example, we can specify a logical test using the relational operator ==; only the observations for which the logical test is true will be extracted	`data$var1[data$var2==1]` # extracts the observations of the variable *var1* for which the variable *var2* equals 1

2.7.3 R FUNCTIONS

function	description	required argument(s)	example(s)
ifelse()	creates the contents of a new variable based on the values of an existing one	three, separated by commas, in the following order: (1) logical test (see ==) (2) return value if test is true (3) return value if test is false values should be in quotes if text but without quotes if numbers (see)	`ifelse(data$variable=="yes", 1, 0)` # returns a 1 whenever the observation of *variable* equals "yes" and a 0 otherwise, creating the contents of a binary variable using the existing character variable *variable*

3. INFERRING POPULATION CHARACTERISTICS VIA SURVEY RESEARCH

Another common goal for data analysis in the social sciences is to estimate population characteristics using surveys. Surveys enable us to infer the characteristics of an entire population by measuring them in a representative sample. In this chapter, we explain how survey research works and discuss some methodological challenges that may arise in the process. We also learn how to visualize and summarize both the distribution of a single variable and the relationship between two variables. To illustrate these concepts, we analyze data from and about the 2016 British referendum on European Union (EU) membership.

R symbols, operators, and functions introduced in this chapter: table(), prop.table(), na.omit(), hist(), median(), sd(), var(), ", plot(), abline(), and cor().

3.1 THE EU REFERENDUM IN THE UK

Faced with growing discontent among the British people with the relationship between the United Kingdom (UK) and the EU, in 2016 the UK government held a referendum. British voters were asked to weigh in on whether the UK should stay in or leave the EU. The second choice became known as Brexit, an abbreviation for "British exit."

This was a high-stakes referendum, with global political, legal, and socioeconomic ramifications. Leading up to the vote, a group of researchers from the British Election Study (BES) conducted a large survey to measure public opinion and predict the outcome. In the first few sections of this chapter, we analyze data from this survey to measure support for Brexit and determine the demographic makeup of Brexit supporters. Subsequently, we analyze the actual referendum results to determine whether patterns observed in the BES sample can also be observed in the population of interest as a whole.

Based on Sara B. Hobolt, "The Brexit Vote: A Divided Nation, a Divided Continent," *Journal of European Public Policy* 23, no. 9 (2016): 1259–77. The data come from Wave 7 of the British Election Study.

3.2 SURVEY RESEARCH

In the social sciences, we often want to know the characteristics of a population of interest. Yet collecting data from every individual in the target population may be prohibitively expensive or simply not feasible.

A **sample** is a subset of observations from a target population. A **representative sample** accurately reflects the characteristics of the population from which it is drawn.

In survey research, we collect data from a subset of observations in order to understand the target population as a whole. The subset of individuals chosen for study is called a **sample**. The number of observations in the sample is represented by *n*, and the number of observations in the target population is represented by *N*. For example, in the aforementioned BES survey, researchers collected data from just under 31,000 people to infer the attitudes of more than 46 million eligible UK voters (n=31,000; N=46 million). Even more remarkably, in the United States, researchers typically survey only about 1,000 people to infer the characteristics of more than 200 million adult citizens (n=1,000; N=200 million).

TIP: Recall, the notation *n* stands for the number of observations in a dataframe or in a variable. As we see here, it also stands for the number of observations in a sample.

In survey research, it is vital for the sample to be representative of the population of interest. A **representative sample** accurately reflects the characteristics of the population from which it is drawn. Characteristics appear in the sample at similar rates as in the population as a whole.

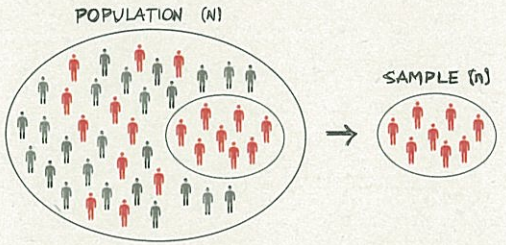


FIGURE 3.1. A sample is a subset of observations from a target population. In this case, the sample is clearly not representative of the population. The proportion of red individuals in the sample is substantially different than the proportion of red individuals in the population.

If the sample is not representative, our inferences regarding the population characteristics based on the sample will be invalid. For example, in figure 3.1 above, the sample is clearly not representative of the population; the proportion of red individuals in the sample is 100% (8/8=1), while the proportion of red individuals in the population is only about 43% (19/44=0.43). As a result, the sample would lead us to infer the wrong population characteristics.

RECALL: The proportion of observations that meet a criterion is calculated as:

$$\frac{\text{number of observations that meet criterion}}{\text{total number of observations}}$$

To interpret this fraction as a percentage, we multiply the resulting decimal by 100.

3.2.1 RANDOM SAMPLING

Random sampling consists of randomly selecting individuals from the population.

The best way to draw a representative sample is to select individuals at random from the population. This procedure is called **random sampling**. For example, to select individuals from a population randomly, we could number the individuals from 1 to *N*, write the numbers on slips of paper, put the slips of paper in a hat, shake the hat, and choose *n* slips of paper from the hat. (In practice, researchers do not use a hat but instead use a computer program like R to draw *n* random numbers from 1 to *N*.)

See figure 3.2 for an example of a randomly selected sample. In this case, the proportion of red individuals in the sample is 38% (3/8=0.38), which is not far from the proportion of red individuals in the population (43%). It is not exactly the same because *n* is relatively small. As we will see later in the book, as the sample size (*n*) increases, the characteristics of the sample will more closely approximate those of the population.

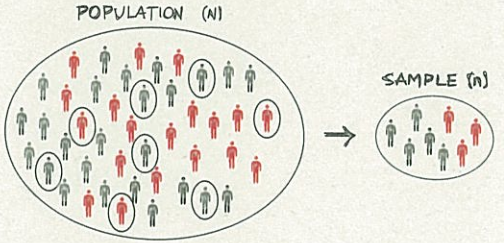


FIGURE 3.2. By randomly selecting individuals from the population, the proportion of red individuals in the sample more closely approximates the proportion of red individuals in the population than the sample shown in figure 3.1.

In the previous chapter, we saw how random assignment of individuals into treatment and control groups makes the two groups identical to each other, on average, before the treatment, in both observed and unobserved traits. Here, the random selection of individuals from the population makes the sample and the target population identical to each other, on average, in both observed and unobserved traits.

TIP: Do not confuse random treatment assignment with random sampling. Random treatment assignment means assigning treatment (deciding who receives it and who doesn't) at random; random sampling means selecting individuals from the population at random to be part of the sample.

INFERRING POPULATION CHARACTERISTICS VIA RANDOM SAMPLING: By randomly selecting a sample of observations from the target population, we ensure that the target population and the sample are, on average, identical to each other in all observed and unobserved characteristics. In other words, we ensure that the sample is representative of the target population, which enables us to make valid inferences about the population.

3.2.2 POTENTIAL CHALLENGES

While random sampling is straightforward in theory, in practice it often faces complications that might invalidate the results.

The **sampling frame** is the complete list of individuals in a population. **Unit nonresponse** occurs when someone who has been selected to be part of the survey sample refuses to participate. **Item nonresponse** occurs when a survey respondent refuses to answer a certain question. **Misreporting** occurs when respondents provide inaccurate or false information.

First, to implement random sampling, we need the complete list of observations in the target population. This list is known as the **sampling frame**. In practice, the sampling frame of a population can be difficult to obtain. Lists of residential addresses, emails, or phone numbers often do not include the entire population of interest. More problematically, the individuals missing tend to be systematically different from those included. For example, a list of residential addresses may miss people who are either homeless or have recently moved, two segments of the population that are notably different from the rest. These omissions may render the lists not only incomplete but also unrepresentative of the population.

Second, even if we have access to a comprehensive list of the individuals in the population, some of those randomly selected might refuse to participate in the survey. This phenomenon is called **unit nonresponse**. If the individuals who refuse to participate differ systematically from those who agree, the resulting sample will be unrepresentative.

Third, participants might agree to answer some but not all of the questions in the survey. Respondents might feel uncomfortable sharing with strangers certain information about themselves. Whenever we have unanswered questions, we encounter what is called **item nonresponse**. If the missing answers differ systematically from the recorded answers, the data collected for the question at hand will not accurately reflect the characteristics of the population.

Fourth, participants might provide inaccurate or false information. This phenomenon, known as **misreporting**, is particularly likely when one answer is more socially acceptable or desirable than the others. For example, in the United States, official turnout rates in presidential elections have recently been around 60%, yet more than 70% of respondents in the American National Election Studies (ANES) report voting. Voting is often perceived to be a civic duty, so respondents might feel social pressure to lie about their voting behavior. As a rule, whenever we rely on self-reporting, we should be aware that misreporting might contaminate the data collected.

The statistical adjustments necessary to address these problems are beyond the scope of this book. For the purpose of our analysis, we assume that the sample from the BES survey is representative of the target population of interest, all eligible UK voters. Consequently, we use it to infer the population's support for Brexit.

3.3 MEASURING SUPPORT FOR BREXIT

Let's analyze the BES survey data to see how much support there was for Brexit a few weeks before the referendum occurred. (The survey was conducted between April 14 and May 4, 2016, and the referendum took place on June 23.)

The code for this chapter's analysis can be found in the "Population.R" file. Alternatively, you may choose to create a new blank R script and practice typing the code yourself. The file "BES.csv" contains the survey data, and table 3.1 provides the names and descriptions of the variables.

TABLE 3.1. Description of the variables in the BES survey data, where the unit of observation is respondents.

variable	description
vote	respondent's vote intention in the EU referendum: "leave", "stay", "don't know", or "won't vote"
leave	identifies leave voters: 1=intends to vote "leave" or 0=intends to vote "stay"; (NA=either "don't know" or "won't vote")
education	respondent's highest educational qualification: 1=no qualifications, 2=general certificate of secondary education (GCSE), 3=general certificate of education advanced level (GCE A level), 4=undergraduate degree, or 5=postgraduate degree; (NA=no answer)
age	respondent's age (in years)

Before starting our analysis of the BES survey dataset, we need to load and make sense of it, just as we did in chapter 1 with the STAR dataset. (See section 1.7 for details.)

First, we change the working directory so that R knows where to look for the data. Go ahead and run the code you used in chapter 1 to direct R to the DSS folder. Now we can read and store the dataset in an object named *bes* by running:

```
bes <- read.csv("BES.csv") # reads and stores data
```

RECALL: If the DSS folder is saved directly on your Desktop, to set the working directory, you must run `setwd("~/Desktop/DSS")` if you have a Mac and `setwd("C:/user/Desktop/DSS")` if you have a Windows computer (where *user* is your own username). If the DSS folder is saved elsewhere, please see subsection 1.7.1 for instructions on how to set the working directory.

To get a sense of the dataset, we can look at the first six observations using the function `head()`:

```
head(bes)  # shows first observations
##              vote leave education age
## 1           leave     1         3  60
## 2           leave     1        NA  56
## 3            stay     0         5  73
## 4           leave     1         4  64
## 5 don't know      NA        2  68
## 6            stay     0         4  85
```

ADVANCED TIP: Recall that ifelse() creates the contents of a new variable based on the values of an existing one. It requires three arguments in the following order, separated by commas: (1) the logical test, (2) return value if test is true, and (3) return value if test is false. If the variable *leave* had not been part of the dataframe, we could have created it using one ifelse() function nested in another:

```
bes$leave <-
  ifelse(bes$vote=="leave", 1,
    ifelse(bes$vote=="stay", 0, NA))
```

The observations of the variable *leave* will be a 1 when *vote* equals "leave", a 0 when *vote* equals "stay", and an NA in all other cases. The structure of this piece of code is as follows: ifelse(test1, value if test1 is true, ifelse(test2, value if test1 is false and test2 is true, value if both test1 and test2 are false)).

Based on this output and table 3.1 (including the title of the table), we learn that each observation represents a survey respondent, and that the dataset contains four variables:

- *vote* captures how each respondent intended to vote in the referendum on Britain's EU membership at the time of the survey. It is a character variable that can take the following four values: "leave", "stay", "don't know", or "won't vote".
- *leave* is a binary variable that identifies leave voters, that is, Brexit supporters. It equals 1 if respondent intended to vote "leave" and 0 if respondent intended to vote "stay". For respondents who either didn't know how they would vote or did not intend to vote, we have NAs, which is how R represents missing values. (More on missing data soon.) Note that if this variable had not been part of the dataframe, we could have created it using the contents of *vote* by using multiple ifelse() functions. (See ADVANCED TIP in the margin.)
- *education* represents respondents' highest educational qualification. It is a non-binary numeric variable that can take five values: 1, 2, 3, 4, or 5. Each of these represents a different level of educational attainment, where 1 is the lowest level and 5 is the highest. Nonresponses are coded as NAs.
- *age* captures respondents' age in years, which means that it is a non-binary numeric variable that can take many values.

Putting it all together, for example, we interpret the first observation as representing a survey respondent who intended to vote "leave" in the EU referendum and was, therefore, a Brexit supporter, whose highest educational qualification was the general certificate of education advanced level (the British equivalent of a high school diploma), and who was 60 years old at the time of the survey.

Finally, to find out how many respondents were part of the survey, we run:

```
dim(bes) # provides dimensions of dataframe: rows, columns
## [1] 30895     4
```

Based on this output, we determine that the dataset contains information about 30,895 respondents. In other words, the sample size (n) is 30,895. (This is an impressively large survey!)

3.3.1 PREDICTING THE REFERENDUM OUTCOME

To predict the outcome of the referendum, we need to estimate the proportions of eligible UK voters who were (i) in favor of Brexit and (ii) opposed to Brexit, at the time the survey was conducted.

If the sample of respondents in the BES survey is representative of all eligible UK voters, then we can use the proportion of individuals' characteristics in the sample as good approximations of the proportion of individuals' characteristics in the entire target population.

To compute the proportions of individuals who were in favor of and opposed to Brexit in the BES sample, we create the table of proportions of the variable *vote*, but first we need to create a table of its frequencies.

3.3.2 FREQUENCY TABLES

The **frequency table** of a variable shows the values the variable takes and the number of times each value appears in the variable.

The **frequency table** of a variable shows the values the variable takes and the number of times each value appears in the variable.

For example, if X={1, 0, 0, 1, 0}, the frequency table of X is:

values	0	1
frequencies	3	2

The table shows that X contains three observations that take the value of 0 and two observations that take the value of 1.

To create a frequency table in R, we use the function table(). The only required argument is the code identifying the variable to be summarized. In this case, to calculate the frequency table of *vote*, we run:

```
table(bes$vote) # creates frequency table
##    don't know         leave          stay      won't vote
##          2314         13692         14352           537
```

table() creates the frequency table of a variable. The only required argument is the code identifying the variable. Example: table(*data*$*variable*).

RECALL: We use the $ character to access a variable inside a dataframe. To its left, we specify the name of the object where the dataframe is stored (without quotes). To its right, we specify the name of the variable (without quotes). Example: *data*$*variable*.

This frequency table shows that out of the 30,895 respondents in the BES survey, 2,314 were undecided, 13,692 intended to vote "leave", 14,352 intended to vote "stay", and 537 had no intention of voting. Note that the sum of all the frequencies equals the total number of observations in the sample, n (2314+13692+14352+537=30895).

3.3.3 TABLES OF PROPORTIONS

The **table of proportions** of a variable shows the proportion of observations that take each value in the variable. By definition, the proportions in the table should add up to 1 (or 100%).

A **table of proportions** shows the proportion of observations that take each value in a variable.

For example, if X={1, 0, 0, 1, 0}, the table of proportions of X is:

values	0	1
proportions	0.6	0.4

The table shows that 60% of the observations in X take the value of 0 and 40% take the value of 1. (Recall, to interpret a proportion as a percentage, we multiply the decimal value by 100.)

prop.table() converts a frequency table into a table of proportions. The only required argument is the output of the function table() with the code identifying the variable inside the parentheses. Example: prop.table(table(*data*$*variable*)).

To create a table of proportions in R, we use the function prop.table(), which converts a frequency table into a table of proportions. This function takes as its main argument either (a) the name of the object containing the output of the function table() or (b) the function table() directly; in both cases, the variable of interest is specified inside the parentheses of table(). In our current example, then, to calculate the table of proportions of *vote*, we could run:

```
## option a: create frequency table first
freq_table <- table(bes$vote)  # object with frequency table
prop.table(freq_table)  # creates table of proportions
##  don't know       leave        stay  won't vote
##     0.07490     0.44318     0.46454     0.01738
```

Alternatively, we could skip the step of creating an object with the frequency table and run instead:

```
## option b: do it all at once
prop.table(table(bes$vote))  # creates table of proportions
##  don't know       leave        stay  won't vote
##     0.07490     0.44318     0.46454     0.01738
```

Based on the proportions in the sample shown in the outputs above, we can estimate that when the survey was administered, 44% of eligible UK voters intended to vote "leave" and 46% to vote "stay"; more than 7% of the population was still undecided.

At this time, then, a slightly higher proportion of respondents intended to vote "stay" rather than "leave". The proportion of undecided, however, was larger than this difference (7% > 46%−44%), and thus, the survey results did not provide a clear prediction of the outcome of the referendum.

In reality, the referendum turned out to be quite close. The leave camp received 51.9% of the vote, and the stay camp received 48.1% of the vote. Thus, the leave camp won with a margin of only 3.8 percentage points (51.9%−48.1%=3.8 p.p.).

3.4 WHO SUPPORTED BREXIT?

We can also analyze the BES survey data to examine the characteristics of Brexit supporters and non-supporters. Specifically, we can determine how these two groups compare in terms of education level and age.

We begin this section by learning how different functions deal with missing data, and then we learn how to conduct our analysis on the observations that do not have missing information. Next, to compare the level of education of Brexit supporters to that of non-supporters, we explore the relationship between *leave* and *education* by creating a two-way table of frequencies and a two-way table of proportions. These tables are similar to the ones we created when exploring the contents of the variable *vote*, except that now we examine the contents of two variables at a time.

Then, to compare the age distribution of Brexit supporters to that of non-supporters, we explore the relationship between *leave* and *age*. In this case, we do not create a two-way table of frequencies or a two-way table of proportions. Because *age* (in years) can take a large number of distinct values, these tables would be too large to be informative. Instead, to visualize both age distributions and compare them to each other, we create histograms of *age* for supporters and non-supporters. Finally, to summarize and compare the characteristics of the two age distributions, we compute descriptive statistics such as the mean, median, standard deviation, and variance of *age* for each group.

3.4.1 HANDLING MISSING DATA

As we saw earlier, missing values are common in survey data. In the BES dataset, two variables contain NAs, which is how R represents missing values. The variable *leave* has NAs when respondents were undecided or didn't intend to vote. The variable *education* has NAs when respondents refused to provide an answer. (See the second and fifth observations of the dataframe shown in the output of head() at the beginning of section 3.3.)

Some functions in R automatically remove missing values before performing operations; others do not. For example, the function table() ignores missing values by default. If you want the function to include them, you need to specify the optional argument named exclude and set it to equal NULL. This asks R not to exclude any values from the table of frequencies. (See the RECALL in the margin for a brief overview of how optional arguments work.) In the current example, to create the table of frequencies of *education*, including missing values, we run:

RECALL: Inside the parentheses of a function, we can specify optional arguments by including the name of the optional argument (without quotes) and setting it to equal a particular value. TRUE, FALSE, NA, and NULL are special values in R and should not be written in quotes. Finally, optional arguments are specified after the required arguments, separated by commas.

```
table(bes$education, exclude=NULL) # table() including NAs
##    1    2    3    4    5 <NA>
## 2045 5781 6272 10676 2696 3425
```

Based on the output, a little more than 3,400 respondents refused to provide their level of education. The item nonresponse rate here, or the proportion of respondents who refused to provide an answer to this question, was about 11% (3425/30895=0.11).

The function mean() does not automatically exclude missing values. If a variable contains any NAs, R will not be able to compute

the average of the variable unless we change the default settings. For example, run the following:

RECALL: In R, the function mean() calculates the mean of a variable. Example: mean(data$variable).

```
mean(bes$leave) # mean() without removing NAs
## [1] NA
```

Here, R returns an NA, indicating the presence of missing values.

We can instruct R to remove the NAs before computing the average by specifying the optional argument named na.rm (which stands for "NA remove") and setting it to equal TRUE.

```
mean(bes$leave, na.rm=TRUE) # mean() removing NAs
## [1] 0.4882328
```

RECALL: The mean of a binary variable can be interpreted as the proportion of the observations that have the characteristic identified by the variable (that have a value of 1).

Now, R provides the result of the operation. We interpret the output as indicating that, in the BES survey, out of the respondents who had already made up their minds to vote for one camp or the other, about 49% were Brexit supporters (0.49×100=49%).

To see how other functions deal with missing values, we can use the help tab of RStudio (in the lower-right window). This tab provides descriptions of all the R functions, including the actions they perform, the arguments they require, and the settings they use by default as well as how to change them. To read about a particular function, all we need to do is manually select the help tab, type the name of the function next to the magnifying glass icon, and hit enter. (See figure 3.3 as an example.)

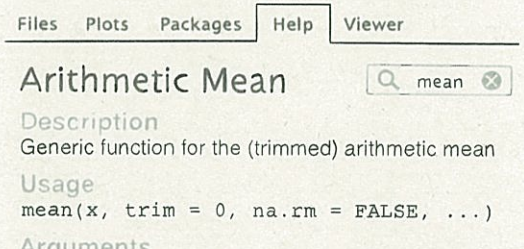


FIGURE 3.3. Example of the type of information displayed in RStudio's help tab.

To remove from the dataframe all observations with missing values, we can use the function na.omit(). For our current purposes, to get rid of all observations with at least one NA from *bes*, we run:


```
bes1 <- na.omit(bes) # removes observations with NAs
```

na.omit() deletes all observations with missing data from a dataframe. The only required argument is the name of the object where the dataframe is stored. Example: na.omit(*data*).

The code na.omit(bes) returns the original dataframe without the observations that have any missing values. With the assignment operator <-, we store this new dataframe in an object named *bes1*. The environment (the storage room of the current R session shown in the upper-right window) should now contain two objects: *bes* (the original dataframe) and *bes1* (the new dataframe).

A word of caution: The function na.omit() instructs R to delete all observations with any missing data. To avoid removing observations needlessly, before applying this function to a dataframe, we should make sure that all the variables in the dataframe that contain any missing values are needed for the analysis. (Instructions for extracting the variables we want to use in the analysis from a dataframe are in the ADVANCED TIP in the margin.)

ADVANCED TIP: Recall that $ is the operator used to extract a selection of observations from a variable. It is also the operator used to extract a selection of observations from a dataframe. In both cases, to its left, we specify what we want to subset (whether it is a variable or a dataframe), and inside the square brackets, we specify the criterion of selection.

To extract a subset of variables from a dataframe, we can use the $ operator in conjunction with the function c(), which combines values into a vector (as we will see in detail in chapter 6). Example:

```
reduced_data <-
        original_data[c("var1", "var2")]
```

This piece of code will create a new object, named *reduced_data*, containing a dataframe with the variables named *var1* and *var2* from the dataframe stored in *original_data*.

In the case of the BES survey, only two variables contain NAs: *leave* and *education*. We are not interested in the respondents for whom we have a missing value in *leave*. They either did not intend to vote or had not yet made up their minds about Brexit. And, since we will use *education* in our analysis, we will need to exclude respondents who refuse to provide their educational background. Consequently, applying na.omit() to *bes* does not result in unnecessarily removing any observations.

After using the function na.omit(), it is a good idea to (i) look at a few observations from both dataframes to ensure the function worked as expected, and (ii) compute how many observations were deleted.

To accomplish the first task, we can use the function head():

```
head(bes) # shows first observations of original dataframe
##          vote leave education age
## 1       leave     1         3  60
## 2       leave     1        NA  56
## 3        stay     0         5  73
## 4       leave     1         4  64
## 5 don't know    NA         2  68
## 6        stay     0         4  85

head(bes1) # shows first observations of new dataframe
##          vote leave education age
## 1       leave     1         3  60
## 3        stay     0         5  73
## 4       leave     1         4  64
## 6        stay     0         4  85
## 7       leave     1         3  78
## 8       leave     1         2  51
```

Comparing the two outputs above, we observe that, as expected, na.omit() deleted from the original dataframe the second and fifth observations because they both contain at least one NA. (Note that, by default, R keeps the original row numbers; as a result, *bes1* does not have any rows numbered 2 or 5.)

To accomplish the second task, we can use the function dim():

```
dim(bes)  # provides dimensions of original dataframe
## [1] 30895     4

dim(bes1)  # provides dimensions of new dataframe
## [1] 25097     4
```

By deleting observations with missing data, we reduced the dataset from 30,895 to 25,097 observations. A total of 5,798 observations, or close to 19% of the original observations, were removed because they contained at least one NA (30895−25097=5798 and 5798/30895=0.19).

Before continuing with the analysis, it is worth noting that removing observations with missing values from a dataset might make the remaining sample of observations unrepresentative of the target population, thereby rendering our inferences of population characteristics invalid. Here, for example, if respondents who refused to provide their level of education were all in favor of Brexit, our analysis of the new dataframe, *bes1*, would undermine the level of support for Brexit. The statistical methods used to address this problem are beyond the scope of this book. For our purposes, we assume that the sample from the BES survey is representative of all eligible UK voters, with or without the observations with missing values.

Going forward, we will analyze the data in the new dataframe, *bes1*, which does not contain any NAs. (The code identifying the variables will follow the structure *bes1$variable_name* instead of *bes$variable_name*.)

3.4.2 TWO-WAY FREQUENCY TABLES

A **two-way frequency table**, also known as a cross-tabulation, shows the number of observations that take each combination of values of two specified variables.

To see the level of education of Brexit supporters and non-supporters within the sample, we can create the two-way frequency table of *leave* and *education*. A **two-way frequency table**, also known as a cross-tabulation, shows the number of observations that take each combination of values of two specified variables.

For example, if X and Y are as defined in the first table below (the dataframe), the two-way frequency table of X and Y is the second table below:

i	X	Y
1	1	1
2	0	1
3	0	1
4	1	0
5	0	0

The two-way frequency table of X and Y is:

		values of Y	
		0	1
values of X	0	1	2
	1	1	1

The two-way frequency table shows that in the dataframe:

- there is one observation for which both X and Y equal 0 (the fifth observation)
- there are two observations for which X equals 0 and Y equals 1 (the second and third observations)
- there is one observation for which X equals 1 and Y equals 0 (the fourth observation)
- there is one observation for which both X and Y equal 1 (the first observation).

To produce a two-way frequency table, we use the function table(), just as we did to produce a one-way frequency table. For the two-way version, however, we need to specify two variables as required arguments (separated by a comma). In the study at hand, to create the two-way frequency table of *leave* and *education*, we run:

```
table(bes1$leave, bes1$education) # two-way frequency table
##        1    2    3    4    5
##  0   498 1763 3014 6081 1898
##  1  1356 3388 2685 3783  631
```

table() creates a two-way frequency table when two variables are specified as required arguments (separated by a comma). In the output, the values of the first specified variable are shown in the rows; the values of the second specified variable are shown in the columns. Example: table(*data$variable1, data$variable2*).

In the output above, we can see that *leave* takes two values (0 or 1) and that *education* takes five (1, 2, 3, 4, or 5). (Note that the values of the variable specified as the first argument in the function are shown in the rows; the values of the second variable are shown in the columns.) The numbers in each cell indicate the frequency, or count, of each combination of values in the dataset. For example, we see from the first cell that in the BES sample, there were 498 respondents who were not Brexit supporters (*leave*=0) and had no educational qualification (*education*=1).

Two-way frequency tables can help us discover the relationship between two variables. For example, in the table above we observe that among respondents with no educational qualification (*education*=1), there were fewer Brexit non-supporters than supporters (498 non-supporters vs. 1,356 supporters). In contrast, among respondents with the highest educational qualification (*education*=5), there were more non-supporters than supporters (1,898 non-supporters vs. 631 supporters).

3.4.3 TWO-WAY TABLES OF PROPORTIONS

To infer the level of education of Brexit supporters and non-supporters among all eligible UK voters, we need to compute the proportion of individuals in the sample with each combination of relevant characteristics. Recall, if the sample is representative, characteristics should appear in similar proportions in the sample as in the population as a whole.

A **two-way table of proportions** shows the proportion of observations that take each combination of values of two specified variables.

To calculate the relevant proportions within the sample, we create a two-way table of proportions of *leave* and *education*. A **two-way table of proportions** shows the proportion of observations that take each combination of values of two specified variables.

Let's return to the simple example from the previous subsection. If X and Y are as defined in the first table below, the two-way table of proportions of X and Y is the second table below:

i	X	Y
1	1	1
2	0	1
3	0	1
4	1	0
5	0	0

The two-way table of proportions of X and Y is:

		values of Y	
		0	1
values of X	0	0.2	0.4
	1	0.2	0.2

The two-way table of proportions shows that in the dataframe:

- both X and Y equal 0 in 20% of the observations
- X equals 0 and Y equals 1 in 40% of the observations
- X equals 1 and Y equals 0 in 20% of the observations
- both X and Y equal 1 in 20% of the observations.

`prop.table()` converts a two-way frequency table into a two-way table of proportions. The only required argument is the output of the function `table()` with the code identifying the two variables inside the parentheses (separated by a comma). Example: `prop.table(table(data$variable1, data$variable2))`.

To create a two-way table of proportions in R, we use the same function as with the one-variable version: `prop.table()`. Here, though, we need to specify two variables inside the function `table()`, which is the required argument. By default, R produces the two-way table of proportions where the whole sample is the reference group (the denominator). Run:

```
## two-way table of proportions
prop.table(table(bes1$leave, bes1$education))
##              1       2       3       4       5
##   0  0.01984 0.07025 0.12009 0.24230 0.07563
##   1  0.05403 0.13500 0.10698 0.15074 0.02514
```

Because the whole sample is the reference group, the sum of all the proportions within the table equals 1. We interpret the first cell of the table as indicating that 2% of the respondents in the BES survey (0.02×100=2%) were against Brexit (*leave*=0) and had no educational qualification (*education*=1).

If we wanted to know proportions within subsets of the sample, we would need to change the reference group of the calculations. To do so, we specify the optional argument `margin` and set it to equal either 1 or 2. If it equals 1, R will use the first specified variable to set the reference groups. For example, to compute the proportion of different levels of education within Brexit supporters and within Brexit non-supporters, we run:

```
## two-way table of proportions with margin=1
prop.table(table(bes1$leave, bes1$education), margin=1)
##              1       2       3       4       5
##   0  0.03757 0.13302 0.22740 0.45880 0.14320
##   1  0.11450 0.28608 0.22672 0.31943 0.05328
```

Because we included the optional argument `margin=1` and the first specified variable is *leave*, the proportions are calculated within two groups: Brexit non-supporters (*leave*=0) and Brexit supporters (*leave*=1). The proportions in each row now add up to 1. We interpret the first cell of the table as indicating that among all Brexit non-supporters in the sample, close to 4% had no educational qualification (*education*=1).

Alternatively, if we include the optional argument `margin=2`, R will use the second specified variable to define the reference groups. For example, to calculate the proportion of support for Brexit within each educational level, we run:

```
## two-way table of proportions with margin=2
prop.table(table(bes1$leave, bes1$education), margin=2)
##              1       2       3       4       5
##   0  0.26861 0.34226 0.52886 0.61648 0.75049
##   1  0.73139 0.65774 0.47114 0.38352 0.24951
```

The new proportions are calculated within five groups, one for each level of educational attainment. The proportions in each column now add up to 1. We interpret the first cell of the table as indicating that among respondents with no educational qualification (*education*=1), about 27% did not support Brexit (*leave*=0).

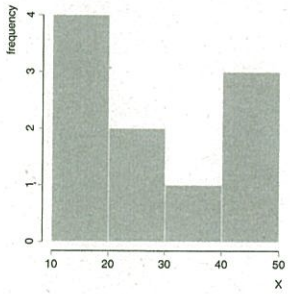
Two-way tables of proportions can also help us discover the relationship between two variables. For example, in the previous table, we find that among respondents with no educational qualification (*education*=1), the majority are Brexit supporters (27% non-supporters vs. 73% supporters). This phenomenon reverses with higher levels of education. Among respondents with the British equivalent of a high school diploma (*education*=3), Brexit supporters are in the minority by a slight margin (53% non-supporters vs. 47% supporters). Among respondents with the highest educational qualification (*education*=5), Brexit supporters are in the clear minority (75% non-supporters vs. 25% supporters).

If the BES sample is representative of all eligible UK voters, we can infer that voters with low levels of education were likely to support Brexit, and voters with high levels of education were likely to oppose Brexit.

3.4.4 HISTOGRAMS

The **histogram** of a variable is the visual representation of its distribution through bins of different heights. The position of the bins along the x-axis indicates the interval of values. The height of the bins indicates the frequency (or count) of the interval of values within the variable.

To compare Brexit supporters to non-supporters in terms of age, we can visualize the two age distributions by creating histograms. A **histogram** is a graphical representation of the variable's distribution, made up of bins (rectangles) of different heights. The position of the bins along the x-axis (the horizontal axis) indicates the interval of values. The height of the bins represents how often the variable takes the values in the corresponding interval.



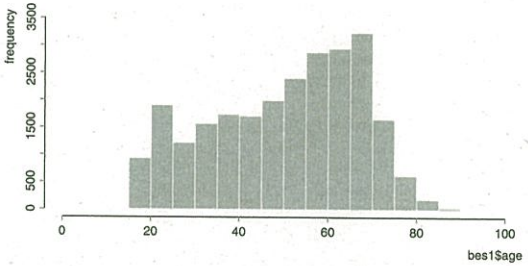
For example, if X={11, 11, 12, 13, 22, 26, 33, 43, 43, 48}, the histogram of X is the graph in the margin. It shows that the variable X contains:

- four observations in the interval from 10 to 20
- two observations in the interval from 20 to 30
- one observation in the interval from 30 to 40
- three observations in the interval from 40 to 50.

The R function to create the histogram of a variable is `hist()`. In the case at hand, to produce the histogram of *age*, we run:

```
hist(bes1$age) # creates histogram
```

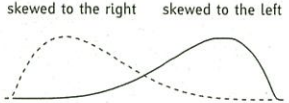
`hist()` creates the histogram of a variable. The only required argument is the code identifying the variable. Example: `hist(data$variable)`.



After running this piece of code, R will display the graph shown above in the plots tab of RStudio (the lower-right window). If R gives you the error message "Error in plot.new(): figure margins too large" instead, try making the lower-right window larger and then re-run the code that creates the plot. (Note that the graphs in the book might look a little different from those you see on your computer. To make the book easier to read, we often modify the default color schemes and styles of graphs. The overall patterns should be the same, however.)

Based on the histogram above, we see that the survey does not have any respondents below the age of 15. (The minimum value this variable takes is actually 18.) This makes sense since researchers purposely reached out only to eligible voters. We can see that the distribution roughly follows a bell curve, although it is skewed to the left. (See TIP in the margin for an explanation of what we mean by skewed.) The largest segment (the tallest bin) is made up of respondents between 65 and 70 years old.

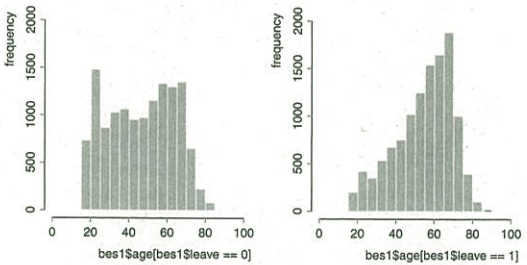
TIP: A bell curve is skewed to the left if the tail on the left side of the distribution is longer than the tail on the right side (as in the solid-line distribution below) and is skewed to the right if the opposite is true (as in the dashed-line distribution below).



The histogram above includes the age of both supporters and non-supporters. To compare the age distribution of Brexit supporters to that of non-supporters, we need to create two histograms, one for each group. For each of these histograms, we need to select only the observations of *age* that meet the criteria (the respondent must be a supporter or a non-supporter, respectively). For this purpose, we can use the `[]` operator in conjunction with the `==` operator, just as we did in chapter 2. (See subsection 2.5.3.) Then we can apply the `hist()` function to each subset. All together, the code to produce the two histograms is:

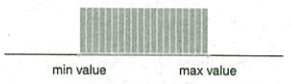
```
## create histograms
hist(bes1$age[bes1$leave == 0]) # for non-supporters
hist(bes1$age[bes1$leave == 1]) # for supporters
```

RECALL: To extract a selection of observations from a variable, we use the `[]` operator. To its left, we specify the variable we want to subset. Inside the square brackets we specify the criterion of selection, using for example the relational operator `==`. Only the observations for which the criterion is true are extracted. Example: `data$var1[data$var2==1]` extracts only the observations of the variable *var1* for which the variable *var2* equals 1.



Looking at the histogram for non-supporters (the one on the left), we see that the age distribution is relatively uniform and that the largest segment is between 20 and 25 years old. In contrast, the histogram for supporters (the one on the right) shows that the age distribution approximates a bell curve, although clearly skewed to the left, and that the largest segment is between 65 and 70 years old. Based on the visual comparison of the age distributions of the two groups, we conclude that Brexit supporters tended to be older than non-supporters.

TIP: In the uniform distribution, all values between the minimum and the maximum are equally likely.



3.4.5 DENSITY HISTOGRAMS

A **density histogram** uses densities instead of frequencies as the height of the bins, where densities are defined as the proportion of the observations in the bin divided by the width of the bin.

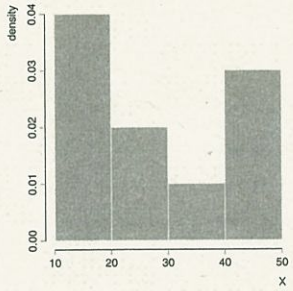
Arguably a better option for visualizing the differences between the age distributions of the two groups is to use density histograms. Density histograms are especially useful for comparing groups with substantially different numbers of observations. In a **density histogram**, the height of each bin indicates the density of the bin, defined as the proportion of the observations in the bin divided by the width of the bin. This is true because the area of each bin (rectangle) is equivalent to the proportion of observations that fall in the bin, that is, that take any of the values within the interval identified by the position of the bin on the x-axis.

Here is the mathematical reasoning. The area of a rectangle or bin is computed as follows:

$$\text{area of the bin} = \text{height of the bin} \times \text{width of the bin}$$

To determine the height of each bin, we (i) rearrange the formula above and (ii) substitute the area of the rectangle with the proportion of observations because, as mentioned, in density histograms these two terms are equivalent:

$$\begin{aligned}\text{height of the bin} &= \frac{\text{area of the bin}}{\text{width of the bin}} \\ &= \frac{\text{proportion of observations in the bin}}{\text{width of the bin}} \\ &= \text{density of the bin}\end{aligned}$$



Let's return to the simple example from the previous subsection. If X={11, 11, 12, 13, 22, 26, 33, 43, 43, 48}, the density histogram of X is the graph in the margin. As we can see, the height of the first bin is 0.04. Here is why:

- out of the 10 observations in the variable, 4 are in this bin; the proportion of observations in the bin is, therefore, 0.4 or 40% (4/10=0.4)
- the width of the bin is 10 because the bin is positioned from 10 to 20 on the x-axis (20−10=10)
- this results in a density of 0.04 (proportion/width=0.4/10=0.04).

Density histograms have two useful properties. First, if the width of the bins is constant, the relative height of the bins implies the relative proportion of observations that fall in the bins. In other words, if one bin is twice as high as another, it means that it contains twice as many observations.

For example, the density histogram above shows that in the variable X, there are:

- twice as many values in the interval from 10 to 20 as in the interval from 20 to 30
- twice as many values in the interval from 20 to 30 as in the interval from 30 to 40
- three times as many values in the interval from 40 to 50 as in the interval from 30 to 40.

Second, because the area of each bin equals the proportion of observations in the bin, the areas of all the bins in the density histogram add up to 1.

For example, the sum of the areas of all the bins in the density histogram above is:

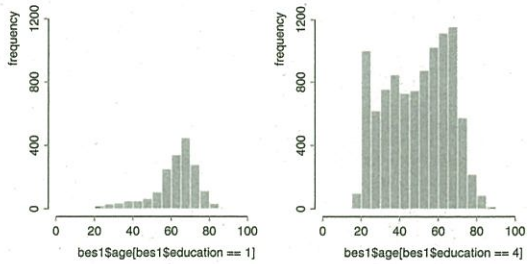
$$\begin{aligned}\sum_{\text{all bins}} \text{height}_{\text{bin}} \times \text{width}_{\text{bin}} &= (0.04\times10) + (0.02\times10) \\ &\quad + (0.01\times10) + (0.03\times10) = 1\end{aligned}$$

Why are density histograms a better option than histograms for visualizing the differences between two distributions? Unlike frequencies, the unit of measurement of densities is comparable across distributions with different numbers of observations. Densities are related to proportions (percentages), which are not affected by changes in the total number of observations. In contrast, frequencies are related to counts, which are affected by changes in the total number of observations. As a result, whenever comparing two distributions with substantially different numbers of observations, it is better to use density histograms than histograms.

To illustrate this, let's compare the age distribution of respondents who have no educational qualifications with the age distribution of respondents who have an undergraduate degree but no postgraduate degree. Because the first group of respondents is much smaller than the second, this comparison highlights the advantages of using density histograms. As we saw earlier, in the BES survey only about 2,000 respondents have no educational qualification (*education*=1), but more than 10,000 have an undergraduate degree as their highest educational qualification (*education*=4).

To compare these two distributions, let's start by creating histograms where the height of the bins reflect frequencies:

```
## create histograms
hist(bes1$age[bes1$education==1]) # w/ no qualifications
hist(bes1$age[bes1$education==4]) # w/ undergraduate degree
```



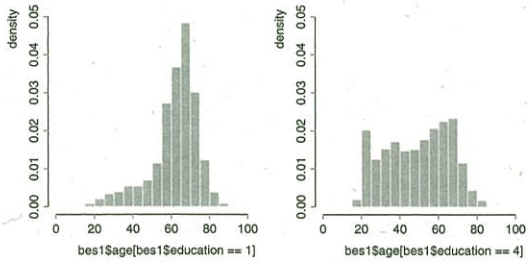
Is the proportion of respondents between 65 and 70 years old among those with no qualifications equivalent to the proportion of respondents in that age group among those with an undergraduate degree? Looking at the two histograms above, it is hard to say. The large difference in the size of the two groups makes comparisons difficult. To more easily compare these two distributions, we can create density histograms.

`hist()` creates the density histogram of a variable when the optional argument `freq` is set to equal `FALSE`. The only required argument is the code identifying the variable. Example: `hist(data$variable, freq=FALSE)`.

To create a density histogram in R, we also use the `hist()` function, but we need to set the optional argument `freq` (which stands for "frequencies") to `FALSE`. In the current example, to produce the density histograms of *age* for respondents with no qualifications and for respondents with undergraduate degrees, we run:

```
## create density histograms
hist(bes1$age[bes1$education==1],
     freq=FALSE) # w/ no qualifications
hist(bes1$age[bes1$education==4],
     freq=FALSE) # w/ undergraduate degree
```

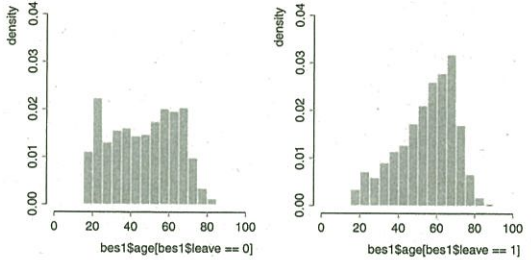
TIP: Here, to facilitate the comparison of the heights (or densities) of the bins across the two histograms, we purposely made both y-axes display the same range of values (from 0 to 0.05).



Looking at the density histograms, we can clearly see that the proportion of respondents between 65 and 70 years old among those with no qualifications (in the graph on the left) is about twice as large as the proportion of respondents in that age group among those with an undergraduate degree (in the graph on the right). We can draw this conclusion by just comparing the heights (or densities) of the bins across the two histograms because in both histograms the bins have all the same widths (5 years).

Now that we have learned the advantages of density histograms, let's return to exploring the distributions of *age* for Brexit supporters and non-supporters. To produce the two relevant density histograms, we run:

```
## create density histograms
hist(bes1$age[bes1$leave == 0]),
     freq=FALSE) # for non-supporters
hist(bes1$age[bes1$leave == 1]),
     freq=FALSE) # for supporters
```



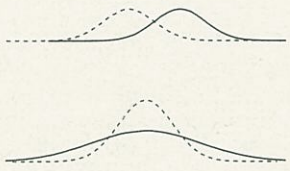
Here we can see, for example, that the proportion of respondents between 20 and 25 years old among Brexit non-supporters (in the graph on the left) is close to three times the proportion of respondents in the same age group among supporters (in the graph on the right). In addition, the proportion of respondents between 65 and 70 years old among Brexit supporters (in the graph on the right) is about one and a half times the proportion of respondents in that age group among non-supporters (in the graph on the left).

In practice, we rarely care about the exact value of each density. We usually just care about the shape of the histogram as demarcated by the height of the bins. We use this shape to describe or illustrate the different distributions. (See figure in the margin.)

3.4.6 DESCRIPTIVE STATISTICS

Another option for measuring the differences between Brexit supporters and non-supporters in terms of age distribution is to compute and compare **descriptive statistics**. Descriptive statistics numerically summarize the main traits of the distribution of a variable.

The **descriptive statistics** of a variable numerically summarize the main characteristics of its distribution.



We can use two different types of descriptive statistics:

- Measures of centrality, such as the mean and the median, summarize the center of the distribution. (See the top figure in the margin, which shows two distributions that are identical except for their centrality.)
- Measures of spread, such as the standard deviation and the variance, summarize the amount of variation of the distribution relative to its center. (See the bottom figure in the margin, which shows two distributions that are identical except for their spread.)

RECALL: The average, or mean, of a variable equals the sum of the values across all observations divided by the number of observations. If the variable is non-binary, the mean should be interpreted as an average, in the same unit of measurement as the variable. If the variable is binary, the mean should be interpreted as a proportion, in percentages after multiplying the result by 100. In R, `mean()` calculates the mean of a variable. Example: `mean(data$variable)`.

In chapter 1, we saw how to compute and interpret the **mean** of a variable. (See section 1.8.) In the running example, the code to compute the average age of each group is:

```
## compute mean
mean(bes1$age[bes1$leave == 0]) # for non-supporters
## [1] 46.89

mean(bes1$age[bes1$leave == 1]) # for supporters
## [1] 55.06823
```

TIP: If we were interested in calculating the average age among *all* respondents of the BES survey, we would run `mean(bes1$age)`, without subsetting *age*.

Based on the results above, the average Brexit non-supporter was 47 years old, while the average supporter was 55 years old. This means that Brexit supporters were eight years older than non-supporters, on average (55−47=8).

The **median** of a variable is the value in the middle of the distribution that divides the data into two equal-size groups.

We can also describe the center of a distribution by using the **median**. The median is the value at the midpoint of the distribution that divides the data into two equal-size groups (or as close to it as possible). When the variable contains an odd number of observations, the median is the middle value of the distribution. When the variable contains an even number of observations, the median is the average of the two middle values.

For example, if X={10, 4, 6, 8, 22}, the median of X is 8. To see this more clearly, we need to sort the values of X in ascending order (as they would be in the distribution). We end up with {4, 6, 8, 10, 22}. Now we clearly see that the value in the middle of the distribution is 8.

Unlike the mean, the median should always be interpreted in the same unit of measurement as the values in the variable, regardless of whether the variable is binary or non-binary.

`median()` calculates the median of a variable. The only required argument is the code identifying the variable. Example: `median(data$variable)`.

The R function to calculate the median of a variable is `median()`. The only required argument is the code identifying the variable. In the running example, to calculate the medians of the two age distributions, we run:


```
## compute median
median(bes1$age[bes1$leave == 0]) # for non-supporters
## [1] 48

median(bes1$age[bes1$leave == 1]) # for supporters
## [1] 58
```

TIP: If we were interested in computing the median age among *all* respondents of the BES survey, we would run `median(bes1$age)`.

The median Brexit non-supporter was 48 years old, while the median supporter was 58 years old. In other words, about half of Brexit non-supporters were 48 years old or younger, and about half of supporters were 58 years old or younger.

In the case of the age distributions here, the mean values (47 and 55) are very similar to the median values (48 and 58), but this is not always true. One important distinction between the two statistics is that while the mean is sensitive to outliers (extreme values in the variable), the median is not. If, for example, we replaced the oldest Brexit supporter aged 97 with a Brexit supporter aged 107, the median value would remain the same because the value of the observation in the middle of the distribution would not have changed. In contrast, the new mean would be higher than the original, since the sum of all the observations (the numerator of the formula) would be 10 units larger.

The **mean** of a variable is more sensitive to outliers than the **median**.

To describe the amount of variation relative to the center of a distribution, we can use the **standard deviation**. Mathematically, it is the result of the following calculation:

$$sd(X) = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\overline{X})^2}{n}}$$
$$= \sqrt{\frac{(X_1-\overline{X})^2+(X_2-\overline{X})^2+\cdots+(X_n-\overline{X})^2}{n}}$$

The **standard deviation** of a variable measures the average distance of the observations to the mean. The larger the standard deviation, the flatter the distribution.

where:

- $sd(X)$ stands for the standard deviation of X
- X_i stands for a particular observation of X, where i denotes the position of the observation
- $\overline{X}$ stands for the mean of X
- n is the number of observations in the variable
- $\sum_{i=1}^{n}(X_i-\overline{X})^2$ means the sum of all $(X_i-\overline{X})^2$ from i=1 to i=n.

Roughly speaking, the standard deviation of a variable provides the average distance between the observations and the mean (in the same unit of measurement as the variable). To better understand this, let's look at a simple example step by step.

RECALL: The unit of measurement of the mean of a variable is the same as the unit of measurement of the variable, when the variable is non-binary.

FORMULA IN DETAIL

If $X=\{2, 4, 6\}$ and the unit of measurement of X is miles:

- The average of X (including its unit of measurement) is:

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{2+4+6}{3} = \frac{12}{3} = 4 \text{ miles}$$

- For each i, we can calculate the term $X_i - \overline{X}$, which gives us a sense of the distance between each observation and the mean of X:
  - for i=1: $X_1 - \overline{X} = 2-4 = -2$ miles
  - for i=2: $X_2 - \overline{X} = 4-4 = 0$ miles
  - for i=3: $X_3 - \overline{X} = 6-4 = 2$ miles
- Note that the term $X_i - \overline{X}$ above can result in both negative and positive numbers. If we calculated the average of this term, positive distances would cancel out negative distances. We do not want such cancellation, since we are trying to measure the average deviation from the center of the distribution. To avoid the cancellation, we need to get rid of the signs. To do so, we square the term $X_i - \overline{X}$. The resulting term, $(X_i - \overline{X})^2$, provides the squared distance from the mean for each observation:
  - for i=1: $(X_1 - \overline{X})^2 = (2-4)^2 = (-2)^2 = 4$ miles^2
  - for i=2: $(X_2 - \overline{X})^2 = (4-4)^2 = (0)^2 = 0$ miles^2
  - for i=3: $(X_3 - \overline{X})^2 = (6-4)^2 = (2)^2 = 4$ miles^2
- To compute the average of the squared distances across all observations, we add them up and divide them by the number of observations:

$$\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n} = \frac{(X_1 - \overline{X})^2 + (X_2 - \overline{X})^2 + (X_3 - \overline{X})^2}{3}$$
$$= \frac{4+0+4}{3} = 2.67 \text{ miles}^2$$

- To return to the same unit of measurement as the original variable, we need to get rid of the square. To do so, we calculate the square root of the average of the squared distances across all observations:

$$sd(X) = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n}} = \sqrt{2.67 \text{ miles}^2} = 1.63 \text{ miles}$$

- We can now interpret this number as the average distance between the observations and the mean in the same unit of measurement as the original variable (miles here).

In short, a smaller standard deviation indicates the observations are closer to the mean, on average. The distribution is concentrated around the mean, and consequently, the density is higher at the center. Analogously, a larger standard deviation indicates that the observations are farther from the mean, on average. The distribution is dispersed, and consequently, the density is lower at the center. For example, in the top figure in the margin, the standard deviation of the dashed distribution is smaller than the standard deviation of the solid distribution.

The R function to calculate the standard deviation of a variable is sd(). The only required argument is the code identifying the variable. Therefore, to compute the standard deviations of the age distributions of Brexit supporters and non-supporters, we run:

```
## compute standard deviation
sd(bes1$age[bes1$leave == 0]) # for non-supporters
## [1] 17.3464

sd(bes1$age[bes1$leave == 1]) # for supporters
## [1] 14.96106
```

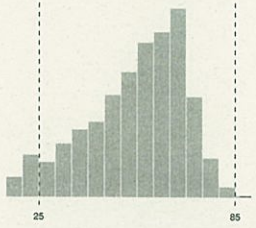
sd() calculates the standard deviation of a variable. The only required argument is the code identifying the variable. Example: sd(*data*$*variable*).

TIP: If we were interested in computing the standard deviation of the distribution of age among *all* respondents of the BES survey, we would run sd(bes1$age).

Among Brexit non-supporters, the average difference between respondents' age and the mean age is 17 years. Among supporters, the average difference is 15 years. If we look back at the two density histograms (at the end of subsection 3.4.5), we can see that the distribution of supporters is more concentrated around the mean than the distribution of non-supporters. It makes sense, then, that the standard deviation of the age distribution of supporters is smaller than that of non-supporters.

One final note about standard deviations: Knowing the standard deviation of a variable helps us understand the range of the data, especially when dealing with bell-shaped distributions known as normal distributions.

As we will see in detail later in the book, one of the distinct characteristics of normal distributions is that about 95% of the observations fall within two standard deviations from the mean (that is, are between the mean minus two standard deviations and the mean plus two standard deviations). For example, we know that the average *age* of Brexit supporters is 55, and the standard deviation of their age distribution is 15 years. If the age distribution of Brexit supporters were a perfect normal distribution, then 95% of Brexit supporters would be between 25 and 85 years old (55−2×15=25 and 55+2×15=85). Looking at the histogram shown in the bottom figure in the margin, this seems about right, although the histogram is skewed to the left, and thus, the formula does not apply exactly.



The **variance** of a variable is the square of the standard deviation.

We sometimes use another measure of the spread of a distribution called **variance**. The variance of a variable is simply the square of the standard deviation:

$$var(X) = sd(X)^2$$

where:

- $var(X)$ stands for the variance of X
- $sd(X)$ stands for the standard deviation of X.

`var()` calculates the variance of a variable. The only required argument is the code identifying the variable. Example: `var(data$variable)`.

To calculate the variance of a variable in R, we can use the function `var()` or simply square the standard deviation of that variable using the `^` operator. For example, to calculate the variance of the age distribution of Brexit supporters, we can run either one of the following lines of code:

```
var(bes1$age[bes1$leave==1]) # calculates variance
## [1] 223.8334

sd(bes1$age[bes1$leave==1])^2 # calculates square of sd
## [1] 223.8334
```

`^` is the operator that raises a number to a power. The number that follows it is the power, that is, the number of times we want to multiply the preceding number by itself. Example: `3^2` raises 3 to the 2nd power (3^2=9).

We are usually better off using standard deviations as our measure of spread. They are easier to interpret because, as we just saw, they are in the same unit of measurement as the variable. (The variance of a variable is in the unit of measurement of the variable squared.)

RECALL: `sqrt()` calculates the square root of the argument specified inside the parentheses. Example: `sqrt(4)`.

If we know the variance of a variable, we take its square root to compute the standard deviation, using the `sqrt()` function:

```
sqrt(var(bes1$age[bes1$leave==1])) # square root of variance
## [1] 14.96106
```

Not surprisingly, running this code produces the same output as `sd(bes1$age[bes1$leave==1])` on the previous page.

3.5 RELATIONSHIP BETWEEN EDUCATION AND THE LEAVE VOTE IN THE ENTIRE UK

Based on Sascha O. Becker, Thiemo Fetzer, and Dennis Novy, "Who Voted for Brexit? A Comprehensive District-Level Analysis," *Economic Policy* 32, no. 92 (2017): 601–50.

In the previous section, in our analysis of the data from the BES survey, we noted that respondents who had higher levels of education were less likely to support Brexit. In this section, we examine the actual referendum results to see whether a similar relationship can be identified in the whole population of UK voters. In particular, we use district-level data to explore how the proportion of residents with high levels of education (who earned at least an undergraduate degree or equivalent) relates to the vote share received by the leave camp. For this purpose, we learn how to create scatter plots to visualize the relationship between two variables and how to compute the correlation coefficient to summarize their linear relationship numerically.

For this analysis, we use a dataset that contains the referendum results on Brexit aggregated at the district level. The dataset is provided in the file "UK_districts.csv". Table 3.2 shows the names and descriptions of the variables included. (Note again that the dataset we use in this section is not from a sample of the population but rather from the entire population of interest.)

TIP: In an individual-level analysis, the unit of observation is individuals. By contrast, in an aggregate-level analysis, the unit of observation is collections of individuals. Here, our unit of observation is districts; each observation represents the residents of a particular district.

TABLE 3.2. Description of the variables in the UK district-level data, where the unit of observation is districts.

variable	description
name	name of the district
leave	vote share received by the leave camp in the district (in percentages)
high_education	proportion of district's residents with an undergraduate degree, professional qualification, or equivalent (in percentages)

In preparation for this section's analysis (assuming we have already set the working directory), we read and store the dataset by running:

```
dis <- read.csv("UK_districts.csv") # reads and stores data
```

RECALL: If the DSS folder is saved directly on your Desktop, to set the working directory, you must run `setwd("~/Desktop/DSS")` if you have a Mac and `setwd("C:/user/Desktop/DSS")` if you have a Windows computer (where *user* is your own username). If the DSS folder is saved elsewhere, please see subsection 1.7.1 for instructions on how to set the working directory.

To get a sense of the dataset, we look at the first few observations by using the function `head()`:

```
head(dis)  # shows first observations
##              name leave high_education
## 1      Birmingham 50.42          22.98
## 2         Cardiff 39.98          32.33
## 3 Edinburgh City 25.56          21.92
## 4 Glasgow City 33.41          25.91
## 5       Liverpool 41.81          22.44
## 6         Swansea 51.51          25.85
```

Based on table 3.2 and the output above, we learn that each observation in the dataset represents a district in the UK, and that the dataset contains three variables:

- *name* is a character variable that identifies the district
- *leave* is a numeric non-binary variable that captures the vote share received by the leave camp in each district, measured in percentages
- *high_education* is a numeric non-binary variable that captures the proportion of residents in the district, measured in percentages, that had undergraduate degrees, professional qualifications, or the equivalent.

We interpret the first observation as representing the district called Birmingham, where leave received a little more than 50%

of the vote share, and about 23% of residents had a high level of education (at least an undergraduate degree or equivalent).

To determine the number of observations in the dataset, we use the function dim():

```
dim(dis) # provides dimensions of dataframe: rows, columns
## [1] 382   3
```

We find that the original dataframe contains information about 382 districts.

Although we did not see any NAs in the first six observations shown by head() above, there might be some missing values in the rest of the data. (Note that the description of variables does not always explicitly report on NAs.) In case there are any NAs in the dataset, we apply the function na.omit() to the dataframe. Because we will use all the variables in our analysis, this will not eliminate observations unnecessarily.

```
dis1 <- na.omit(dis) # removes observations with NAs
```

As is common practice, we use dim() to find out how many observations were deleted:

```
dim(dis1) # provides dimensions: rows, columns
## [1] 380   3
```

Deleting observations with missing values reduces the dataframe to 380 districts. This means that there were only two districts with at least one NA.

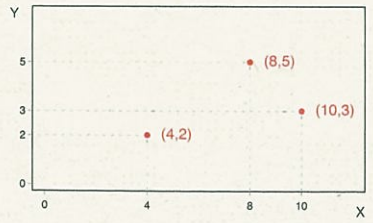
3.5.1 SCATTER PLOTS

A **scatter plot** is the graphical representation of the relationship between two variables, where one variable is plotted along the x-axis, and the other is plotted along the y-axis.

A **scatter plot** enables us to visualize the relationship between two variables by plotting one variable against the other in a two-dimensional space.

Imagine we have the dataframe shown below with two variables of interest, X and Y. The scatter plot of X and Y is the graph shown to its right:

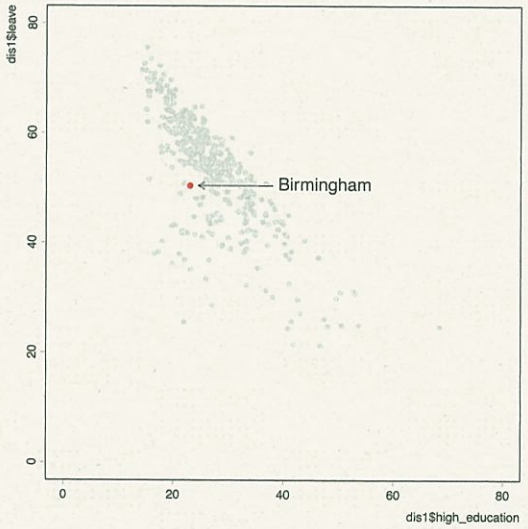
i	X	Y
1	4	2
2	8	5
3	10	3



This dataframe contains only three observations. We can think of each observation i consisting of two coordinates in the two-dimensional space. The first coordinate indicates the position of the point on the x-axis (the horizontal axis), and the second coordinate indicates the position of the point on the y-axis (the vertical axis). Let's look at the first observation (the observation for which i=1). The value of X_1 is 4, which means that the dot for this observation should be lined up with the number 4 on the x-axis. The value of Y_1 is 2, which means that the dot for this observation should be lined up with the number 2 on the y-axis. Together, these two coordinates create the dot (4,2).

To create a scatter plot in R, we use the plot() function. It requires that we specify two arguments in a particular order: (1) the variable we want on the x-axis and (2) the variable we want on the y-axis. Alternatively, we can specify which variables we want to plot on the x- and y-axes by including the names of the arguments in the specification, which are x and y, respectively. Then, the order of the arguments no longer matters. To create the scatter plot of *high_education* and *leave* in the UK district-level dataset, we can run any of the following pieces of code:

```
plot(dis1$high_education, dis1$leave) # scatter plot X, Y
plot(x=dis1$high_education, y=dis1$leave) # scatter plot
plot(y=dis1$leave, x=dis1$high_education) # scatter plot
```



plot() creates the scatter plot of two variables. It requires two arguments, separated by a comma, in this order: (1) the variable to be plotted on the x-axis and (2) the variable to be plotted on the y-axis. Example: plot(*data*$*x_var*, *data*$*y_var*). As an alternative, we can specify which variables we want to plot on the x- and y-axes by including the names of the arguments in the specification, which are x and y, respectively. For example, both of these pieces of code will create the same scatter plot: plot(x=*data*$*x_var*, y=*data*$*y_var*) plot(y=*data*$*y_var*, x=*data*$*x_var*).

TIP: In R functions, the order of the arguments only matters when we do not specify the name of the arguments.

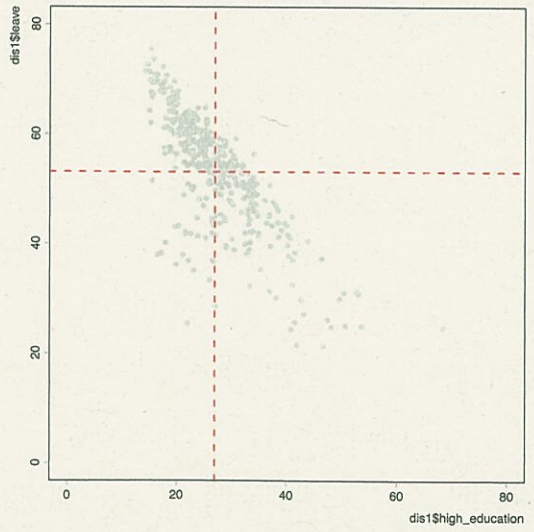
Just as in the simple example, every dot in the scatter plot above represents an observation, a district in this case. For example, the red dot is the observation that represents the district of Birmingham, where about 23% of residents had a high level of education, and close to 50% of the votes were cast in support of Brexit.

What can we learn from this scatter plot about the relationship between these two variables? Are districts with low proportions of highly educated residents likely to support Brexit? What about districts with high proportions of highly educated residents? An intuitive way to answer these questions is by finding the averages of both variables on the graph and using them to divide the graph into four parts (in our imagination or otherwise).

To add straight lines to a graph in R, we can use the `abline()` function. To add a vertical line, we set the argument `v` to equal the value on the x-axis where we want the line drawn. To add a horizontal line, we set the argument `h` to equal the value on the y-axis where we want the line drawn. By default, R draws solid lines. To draw dashed lines, we set the `lty` argument (which stands for "line type") to equal `"dashed"`. For example, go ahead and run:

```
## add straight dashed lines to the most recent graph
abline(v=mean(dis1$high_education), lty="dashed") # vertical
abline(h=mean(dis1$leave), lty="dashed") # horizontal
```

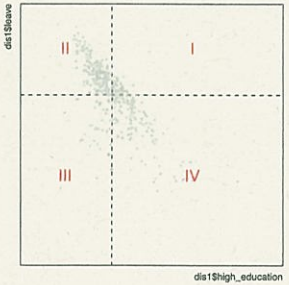
`abline()` adds a straight line to the most recently created graph. To add a vertical line, we set the argument `v` to equal the value on the x-axis where we want the line. To add a horizontal line, we set the argument `h` to equal the value on the y-axis where we want the line. To change the default solid line to a dashed line, we set the optional argument `lty` to equal `"dashed"`. Examples: `abline(v=2)` and `abline(h=3, lty="dashed")`.



If you run the code in the sequence provided here, you should see the graph above. This is the scatter plot of *high_education* and *leave* we created earlier with the function `plot()`, with two added dashed lines: a vertical line marking the mean of *high_education* and a horizontal line marking the mean of *leave*. (Note that the function `abline()` will add lines to the most recently created graph, but R will give you an error message if you have yet to create a graph.)

As shown in the figure in the margin, the dashed lines divide the graph into four quadrants (from top right and counterclockwise):

- Quadrant I: values of the observations are above both means
- Quadrant II: observations have a value of *high_education* below the mean but a value of *leave* above the mean
- Quadrant III: values of the observations are below both means
- Quadrant IV: observations have a value of *high_education* above the mean but a value of *leave* below the mean



Now we can more easily answer our initial questions:

- Are districts with low proportions of highly educated residents likely to support Brexit? In other words, are districts with values of *high_education* below the mean likely to have values of *leave* above the mean?

  Looking at the bulk of the data in the scatter plot above, we determine that the answer is yes. Here is the logic: the districts with values of *high_education* below the mean are in quadrants II and III. Between these two quadrants, quadrant II contains a higher proportion of the data (more dots). This means that districts with values of *high_education* below the mean tend to have values of *leave* above the mean.

- Are districts with high proportions of highly educated residents likely to support Brexit? In other words, are districts with values of *high_education* above the mean likely to have values of *leave* also above the mean?

  Looking at the bulk of the data again, we see that the answer is no. The districts with values of *high_education* above the mean are in quadrants I and IV. Between these two quadrants, quadrant IV contains a higher proportion of the data. This means that districts with values of *high_education* above the mean tend to have values of *leave* below the mean.

We conclude that, at the district level, a higher proportion of highly educated residents is associated with a lower proportion of Brexit supporters. This is consistent with the individual-level relationship we observed using the BES survey data from a sample of the population.

The **correlation coefficient** summarizes the direction and strength of the linear association between two variables. It ranges from -1 to 1. The sign reflects the direction of the linear association: It is positive whenever the slope of the line of best fit is positive and negative whenever the slope of the line of best fit is negative. Its absolute value reflects the strength of the linear association, ranging from 0 (no linear association) to 1 (perfect linear association). The absolute value of the correlation coefficient increases as the observations move closer to the line of best fit and the linear association becomes stronger.

3.5.2 CORRELATION

While the scatter plot provides us with a visual representation of the relationship between two variables, sometimes it is helpful to summarize the relationship with a number. For that purpose, we use the **correlation coefficient**, or correlation for short. Before looking into how to compute this statistic, let's get a sense of how to interpret it.

The correlation coefficient ranges from -1 to 1, and it captures the following two characteristics of the relationship between two variables:

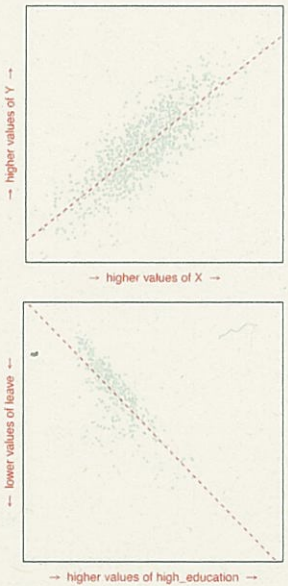
- the direction of their linear association, that is, the sign of the slope of the line of best fit (which is the line that best summarizes the data)
- the strength of their linear association, that is, the degree to which the two variables are linearly associated with each other.

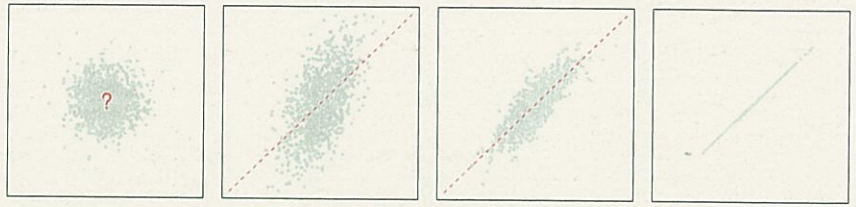
While the direction of the linear association determines the sign of the correlation, the strength of the linear association determines the magnitude of the correlation. Let's look at this in detail.

Depending on the direction of the linear association, that is, whether the line that best fits the data slopes upward or downward, the correlation will be positive or negative:

- The correlation is positive whenever the two variables move in the same direction relative to their respective means, that is, when high values in one variable are likely to be associated with high values in the other, and low values in one variable are likely to be associated with low values in the other. In other words, the correlation is positive whenever the slope of the line of best fit is positive. For example, see the top scatter plot in the margin and the line of best fit that we added. Is the slope positive or negative? Positive. On average, higher values of X are associated with higher values of Y. This means that the correlation between X and Y is positive.

- The correlation is negative whenever the two variables move in opposite directions relative to their respective means, that is, when high values in one variable are likely to be associated with low values in the other, and vice versa. For example, as we saw in the previous subsection, the variables *high_education* and *leave* in the UK district-level dataset move in opposite directions relative to their respective means. As shown in the bottom scatter plot in the margin, the slope of the line of best fit is negative. On average, higher values of *high_education* are associated with lower values of *leave*. This means that the correlation between *high_education* and *leave* is negative.





Depending on the strength of the linear association, that is, how close the observations are to the line of best fit, the absolute value of the correlation coefficient will be closer to 0 or to 1:

- At one extreme, the absolute value of the correlation coefficient is approximately 0 when the linear relationship between the two variables is non-existent. This is the case in the first scatter plot of figure 3.4 above. Here, we would have a hard time fitting a line that would adequately summarize the data.
- At the opposite extreme, the absolute value of the correlation coefficient is exactly 1 if the association between the two variables is perfectly linear. This is the case in the last scatter plot of figure 3.4, where the points are all on a single line.
- All other linear relationships result in a correlation coefficient with an absolute value between 0 and 1. As the observations move closer to the line of best fit, the linear association between the two variables becomes stronger, and the absolute value of the correlation coefficient increases. See, for example, the progression from left to right in figure 3.4.

FIGURE 3.4. Scatter plots of variables with weaker to stronger linear associations. As the observations move closer to the line of best fit, the absolute value of the correlation coefficient increases. From left to right, the correlations are approximately 0, 0.5, 0.8, and 1.

Putting it all together, the correlation between two variables ranges from -1 to 1. The sign of the correlation indicates the direction of the linear association between the variables. And the absolute value of the correlation depicts the strength of the linear association between the variables. (See figure 3.5 above, which illustrates how the value of the correlation coefficient depends on the direction and strength of the linear association between the two variables.)

FIGURE 3.5. Scatter plots of variables with correlations ranging from -1 to 1. From left to right, the correlations are -1, -0.8, -0.5, approximately 0, 0.5, 0.8, and 1.

The **z-score** of an observation is the number of standard deviations the observation is above or below the mean.

FORMULA IN DETAIL

How is the correlation coefficient computed? In order to understand the formula for the correlation coefficient, we first need to learn about **z-scores**. The z-score of an observation is the number of standard deviations the observation is above or below the mean. Specifically, the z-score of each observation of X is defined as:

$$Z_i^X = \frac{X_i - \overline{X}}{sd(X)}$$

where:

- Z_i^X stands for the z-score of observation X_i
- X_i stands for a particular observation of X, where i denotes the position of the observation
- $\overline{X}$ stands for the mean of X
- $sd(X)$ stands for the standard deviation of X.

Returning to the example we saw when learning about standard deviations, if $X=\{2, 4, 6\}$, then $\overline{X}=4$ and $sd(X)=1.63$ (as we computed earlier), and the z-score of each observation of X is:

- for $i=1$: $Z_1^X = \frac{X_1-\overline{X}}{sd(X)} = \frac{2-4}{1.63} = -1.23$
- for $i=2$: $Z_2^X = \frac{X_2-\overline{X}}{sd(X)} = \frac{4-4}{1.63} = 0$
- for $i=3$: $Z_3^X = \frac{X_3-\overline{X}}{sd(X)} = \frac{6-4}{1.63} = 1.23$

The unit of measurement of z-scores is always in standard deviations, regardless of the unit of measurement of the original variable. In addition, the sign of the z-score indicates whether the observation is above or below the mean. For example, we interpret the three z-scores above as follows:

- for $i=1$: $Z_1^X=-1.23$ standard deviations; indicates that X_1 is a little more than one standard deviation below the mean of X
- for $i=2$: $Z_2^X=0$ standard deviations; indicates that X_2 is zero standard deviations away from the mean of X because X_2 and the mean coincide in value
- for $i=3$: $Z_3^X=1.23$ standard deviations; indicates that X_3 is a little more than one standard deviation above the mean of X.

FORMULA IN DETAIL

To compute the correlation between two variables, X and Y, we first convert the observations of both variables to z-scores. Then, the correlation coefficient is calculated as the average of the products of the z-scores of X and Y. Mathematically, the correlation between X and Y is:

$$cor(X,Y) = \frac{\sum_{i=1}^{n} Z_i^X \times Z_i^Y}{n} = \frac{Z_1^X \times Z_1^Y + Z_2^X \times Z_2^Y + \cdots + Z_n^X \times Z_n^Y}{n}$$

where:

- $cor(X, Y)$ stands for correlation between X and Y
- Z_i^X and Z_i^Y denote the z-scores of observation i for X and Y, respectively
- $\sum_{i=1}^{n} Z_i^X \times Z_i^Y$ stands for the sum of the product of the z-scores of X and Y from $i=1$ to $i=n$, meaning from the first observation to the last one
- n is the number of observations.

For example, if X and Y are as defined in the first two columns of the table below, the z-scores of X and Y are as shown in the adjacent two columns:

i	X	Y	Z^X	Z^Y
1	2	6	-1.23	1.23
2	4	4	0	0
3	6	2	1.23	-1.23

And the correlation coefficient between X and Y is:

$$cor(X,Y) = \frac{\sum_{i=1}^{n} Z_i^X \times Z_i^Y}{n} = \frac{-1.23 \times 1.23 + 0 \times 0 + 1.23 \times -1.23}{3} = -1$$

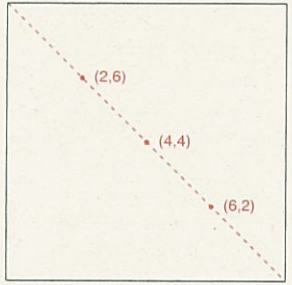
The product of the two z-scores for each observation is:

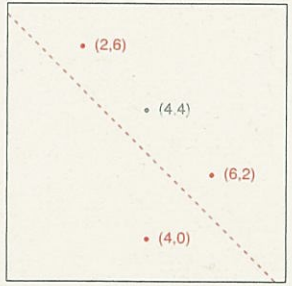
- positive when both z-scores are positive (the observation is above the mean in both variables)
- positive when both z-scores are negative (the observation is below the mean in both variables)

- negative when one z-score is negative, but the other is positive (the observation is below the mean in one variable but above the mean in the other).

As a result, the sign of the correlation coefficient will be:

- positive when the two variables tend to move in the same direction relative to their respective means, that is, when above-average values in one variable are usually associated with above-average values in the other (both z-scores are positive), and when below-average values in one variable are usually associated with below-average values in the other (both z-scores are negative)
- negative when the two variables tend to move in the opposite direction relative to their respective means, that is, when above-average values in one variable are usually associated with below-average values in the other (the two z-scores are of opposite signs).





In the formula in detail above, we manually computed that, if $X=\{2, 4, 6\}$ and $Y=\{6, 4, 2\}$, the correlation between X and Y is -1. What does this tell us?

- The negative sign indicates that the two variables tend to move in opposite directions relative to their respective means. (As we can see in the scatter plot of these two variables shown in the margin, the slope of the line of best fit is indeed negative.)
- The absolute value of 1 indicates that the two variables have a perfect linear association with each other. (As we can see in the same scatter plot, all the points are on the line of best fit.)

Note that this is an extreme example. Most correlations are between -1 and 1, not including the endpoints. If we change the second observation in the example above to (4,0) instead of the original (4,4), then the new correlation between X and Y is about -0.65. As we can see in the new scatter plot shown in the margin, while the slope of the line of best fit continues to be negative, now the points are no longer on the line of best fit. This means that the negative linear association is no longer perfect, which explains why the correlation is no longer exactly -1.

To calculate the correlation coefficient between two variables in R, we use the function `cor()`. Inside the parentheses, we must identify the two variables (separated by a comma and in no particular order). For example, to calculate the correlation between *high_education* and *leave*, we run:

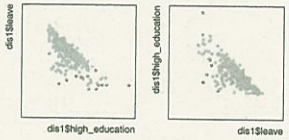

```
cor(dis1$high_education, dis1$leave) # computes correlation
## [1] -0.7633185
```

`cor()` calculates the correlation coefficient between two variables. It requires the code identifying the two variables (separated by a comma and in no particular order). Example: *cor(data$variable1, data$variable2)*.

The correlation between *high_education* and *leave* is -0.76, a strong negative correlation. It is negative because the slope of the line of best fit is negative. Its absolute value is closer to 1 than to 0 because the observations are scattered tightly around the line of best fit. (See the scatter plot of *high_education* and *leave* on the left side of the figure in the margin.)

A few final remarks about the correlation coefficient. First, the correlation between Y and X is the same as the correlation between X and Y. Mathematically: $cor(Y,X)=cor(X,Y)$. For example, by running the following code we see that the correlation between *leave* and *high_education* is the same as the correlation between *high_education* and *leave* (computed above):

```
cor(dis1$leave, dis1$high_education) # computes correlation
## [1] -0.7633185
```



By switching the order of the variables, we are flipping the axes of the scatter plot—the variable that was on the x-axis is now on the y-axis, and vice versa—but the relationship between the variables does not change. Both the direction and strength of their linear association remain the same. Compare the scatter plot of *leave* and *high_education* on the right side of the figure in the margin to the scatter plot of *high_education* and *leave* on the left side. The slope of both lines of best fit are negative, and the points are equally clustered around both lines.

Second, a steeper line of best fit does not necessarily mean a higher correlation in absolute terms, or vice versa. What determines the absolute value of the correlation coefficient is how close the observations are to the line of best fit. For example, in figure 3.6, the absolute value of the correlation is lower in the second scatter plot than in the first (despite the steeper line) because the observations are farther away from the line of best fit.

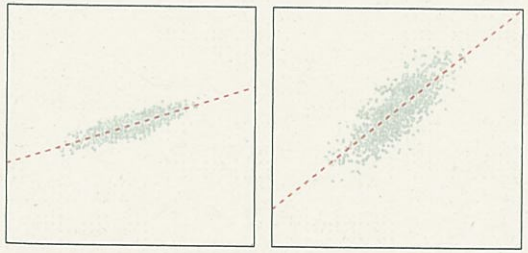


FIGURE 3.6. A steeper line of best fit does not necessarily mean a higher correlation in absolute terms.

Third, if two variables have a correlation coefficient of zero, it does not necessarily mean that there is no relationship between them. It just means that there is no *linear* relationship between them. For example, the two variables depicted in the figure in the margin have a strong parabolic relationship. Their correlation is approximately zero, however, because there is no line that would summarize the relationship well.

Finally, correlation does not necessarily imply causation. Just because two variables have a strong linear association does not mean that changes in one variable cause changes in the other. As we will see in detail in chapter 5, correlation does not necessarily imply causation when the treatment and control groups are not comparable with respect to all the variables that affect the outcome (not including the treatment itself).

CORRELATION DOES NOT NECESSARILY IMPLY CAUSATION: Just because two variables are highly correlated with each other does not necessarily mean that changes in one variable cause changes in the other.

Despite the strong negative correlation between *high_education* and *leave*, without further evidence we cannot conclude that if UK voters became more highly educated, they would also become less likely to support Brexit. In other words, we do not know whether voters' level of education and support for Brexit are causally related in any way. Perhaps the observed relationship is spurious, that is, the product of some third variable that affects both the education level of voters and their support for Brexit, such as the local economy. (We will discuss spurious relationships in more detail in chapter 5.)

3.6 SUMMARY

This chapter introduced us to survey research. We saw how random sampling can help us obtain a representative sample from a population, enabling us to infer population characteristics from a subset of observations.

In addition, we learned some tools that we can use to visualize and summarize the distribution of one variable or the relationship between two. Most data analyses in the social sciences, whether for the purpose of measurement, prediction, or explanation, involve exploring one variable at a time and/or trying to understand the relationship between two variables. In this chapter, we have seen various methods we can use for these purposes in different contexts. Below is a quick review.

To explore one numeric variable at a time, we can:

- create a frequency table
- create a table of proportions
- create a histogram with frequencies or densities to visualize the distribution of the variable
- numerically summarize the center of the distribution by computing the mean and/or the median
- numerically summarize the spread of the distribution by computing the standard deviation and/or the variance.

When exploring the relationship between two numeric variables, we can:

- create a two-way frequency table
- create a two-way table of proportions
- create a scatter plot to visualize their relationship
- numerically summarize the direction and strength of their linear association by computing the correlation coefficient.

These are major building blocks of data analysis, and we will use them in many of the analyses in the remainder of the book.

3.7 CHEATSHEETS

3.7.1 CONCEPTS AND NOTATION

concept/notation	description	example(s)
sample	subset of observations from a target population	the subset of students in a particular class constitutes a sample from the population of students who attend the school
representative sample	sample that accurately reflects the characteristics of the population from which it is drawn; characteristics appear in the sample at similar rates as in the population as a whole	if we randomly select students from those who attend a particular school, we will end up with a representative sample of the population of students from that school; the characteristics of the sample should resemble those of the population; they should have the same proportion of political science majors, females, foreign-born students, and so on
random sampling	procedure that consists of randomly selecting a sample of individuals from the target population	to draw observations from a population randomly, we could number the individuals in the population from 1 to N (where N stands for the number of observations in the population), write the numbers on slips of paper, put the slips in a hat, shake the hat, and choose n slips of paper from the hat (where n stands for the number of observations in the sample)
sampling frame	complete list of individuals in a population	the directory of students attending a particular school is the sampling frame of the population of students in that school
unit nonresponse	phenomenon that occurs when someone who has been selected to be part of a survey sample refuses to participate	when you refuse to participate in a survey via phone or in person, your lack of participation is referred to as a unit nonresponse
item nonresponse	phenomenon that occurs when a survey respondent refuses to answer a certain question	survey respondents might feel uncomfortable answering questions about income and leave those questions blank
misreporting	phenomenon that occurs when respondents provide inaccurate or false information	respondents might claim to have voted in the last election, even if they did not, to conform with social norms
frequency table of a variable	table that shows the values the variable takes and the number of times each value appears in the variable	if $X=\{1, 0, 0, 1, 0\}$, the frequency table of X is: values: 0, 1 frequencies: 3, 2 the table shows that X contains three observations that take the value of 0 and two observations that take the value of 1

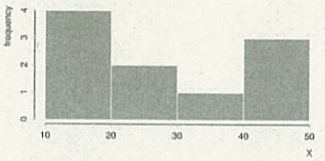
continues on next page...

3.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
table of proportions of a variable	table that shows the proportion of observations that take each value in a variable; by definition, the proportions in the table should add up to 1 (or 100%)	if $X=\{1, 0, 0, 1, 0\}$, then the table of proportions of X is: values: 0, 1 proportions: 0.6, 0.4 the table shows that 60% of the observations in X take the value of 0 and 40% take the value of 1
two-way frequency table of two variables	also known as a cross-tabulation, shows the number of observations that take each combination of values of two specified variables	if X and Y are as defined in the dataframe below: i: 1, 2, 3, 4, 5 X: 1, 0, 0, 1, 0 Y: 1, 1, 1, 0, 0 then the two-way frequency table of X and Y is: values of X: 0, 1 values of Y: 0: 1, 2; 1: 1, 1 the two-way frequency table shows that in the dataframe: - there is one observation for which both X and Y equal 0 (the fifth observation) - there are two observations for which X equals 0 and Y equals 1 (the second and third observations) - there is one observation for which X equals 1 and Y equals 0 (the fourth observation) - there is one observation for which both X and Y equal 1 (the first observation)

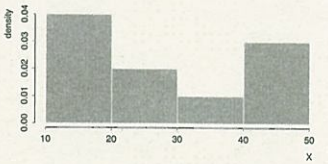
continues on next page...

3.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
two-way table of proportions of two variables	shows the proportion of observations that take each combination of values of two specified variables; by definition, the proportions in the table should add up to 1 (or 100%)	if X and Y are as defined in the dataframe below: i | X | Y 1 | 1 | 1 2 | 0 | 1 3 | 0 | 1 4 | 1 | 0 5 | 0 | 0 then the two-way table of proportions of X and Y is: values of Y: 0, 1 values of X: 0 | 0.2, 0.4 1 | 0.2, 0.2 the two-way table of proportions shows that in the dataframe: - both X and Y equal 0 in 20% of the observations - X equals 0 and Y equals 1 in 40% of the observations - X equals 1 and Y equals 0 in 20% of the observations - both X and Y equal 1 in 20% of the observations
histogram of a variable	visual representation of a variable's distribution through bins of different heights; the position of the bins along the x-axis indicates the interval of values; the height of the bins indicates the frequency (or count) of the interval of values within the variable	if X={11, 11, 12, 13, 22, 26, 33, 43, 43, 48}, the histogram of X is:  the histogram shows that the variable X contains: - four observations in the interval from 10 to 20 - two observations in the interval from 20 to 30 - one observation in the interval from 30 to 40 - three observations in the interval from 40 to 50

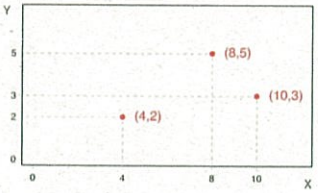
continues on next page...

3.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
density histogram of a variable	histogram that uses densities instead of frequencies as the height of the bins, where densities are defined as the proportion of the observations in the bin divided by the width of the bin; because the width of the bins is constant, the relative height of the bins in a density histogram implies the relative proportion of the observations in the bins; the sum of the areas of all the bins in a density histogram always equals 1	if X={11, 11, 12, 13, 22, 26, 33, 43, 43, 48}, the density histogram of X is:  the density histogram shows that in the variable X, there are: - twice as many values in the interval from 10 to 20 as in the interval from 20 to 30 - twice as many values in the interval from 20 to 30 as in the interval from 30 to 40 - three times as many values in the interval from 40 to 50 as in the interval from 30 to 40
descriptive statistics of a variable	numerically summarize the main characteristics of a variable's distribution: (i) measures of centrality such as mean and median, and (ii) measures of spread such as standard deviation and variance	see mean (in chapter 2), median, standard deviation, and variance
median of a variable; *median*(X)	characterizes the central tendency of the variable; value in the middle of the distribution that divides the data into two equal-size groups; it equals the middle value of the distribution when the variable contains an odd number of observations; it equals the average of the two middle values when the variable contains an even number of observations	if X={10, 4, 6, 8, 22}, the median of X is 8 because the middle value of the distribution of X is 8: {4, 6, 8, 10, 22} (recall that the values in the distribution are always sorted in ascending order) if X={10, 4, 6, 8, 22, 5}, the median of X is 7 because the average of the two middle values of the distribution (6 and 8) is 7: {4, 5, 6, 8, 10, 22}
standard deviation of a variable; $sd(X)$	characterizes the spread of the variable's distribution; it measures the average distance of the observations to the mean; the larger the standard deviation, the flatter the distribution $sd(X) = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n}}$ it is the square root of the variable's variance $sd(X) = \sqrt{var(X)}$	the standard deviation of the dashed distribution is smaller than that of the solid one:  if $var(X) = 4$, then $sd(X) = \sqrt{4} = 2$

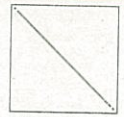
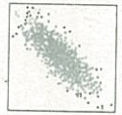
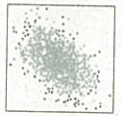
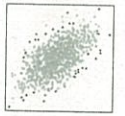
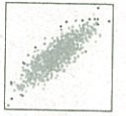
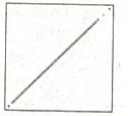
continues on next page...

3.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
variance of a variable; $var(X)$	characterizes the spread of the variable's distribution; it is the square of the variable's standard deviation $var(X) = sd(X)^2$	if $sd(X) = 2$, then $var(X) = 2^2 = 4$
scatter plot of X and Y	graphical representation of the relationship between two variables, X and Y; the X variable is plotted along the horizontal axis, and the Y variable is plotted along the vertical axis	if X and Y are as defined in the dataframe below: i | X | Y 1 | 4 | 2 2 | 8 | 5 3 | 10 | 3 then the scatter plot of X and Y is: 
z-score of an observation of X; Z_i^X	number of standard deviations the observation is above or below the mean of the variable; to transform the observations of a variable into z-scores, we subtract the mean, and then divide the result by the standard deviation: $Z_i^X = \frac{X_i - \overline{X}}{sd(X)}$	if $X=\{2, 4, 6\}$, then $\overline{X}=4$, $sd(X)=1.63$, and the z-score of each observation of X is: - for i=1: $Z_1^X = \frac{X_1-\overline{X}}{sd(X)} = \frac{2-4}{1.63} = -1.23$ - for i=2: $Z_2^X = \frac{X_2-\overline{X}}{sd(X)} = \frac{4-4}{1.63} = 0$ - for i=3: $Z_3^X = \frac{X_3-\overline{X}}{sd(X)} = \frac{6-4}{1.63} = 1.23$

continues on next page...

3.7.1 CONCEPTS AND NOTATION (CONTINUED)

concept/notation	description	example(s)
correlation or correlation coefficient between two variables; $cor(X, Y)$	statistic that summarizes the direction and strength of the linear association between two variables it ranges from -1 to 1 the sign reflects the direction of the linear association: it is positive whenever the slope of the line of best fit is positive, and negative whenever the slope of the line of best fit is negative its absolute value reflects the strength of the linear association, ranging from 0 (no linear association) to 1 (perfect linear association); the absolute value of the correlation coefficient increases as the observations move closer to the line of best fit and the linear association becomes stronger a strong correlation between X and Y does not imply that either X causes Y or that Y causes X; correlation does not necessarily imply causation; more on this in chapter 5 to compute the correlation between two variables, X and Y, we first convert the observations of both variables to z-scores; then, the correlation coefficient is calculated as the average of the products of the z-scores of X and Y: $cor(X, Y) = \frac{\sum_{i=1}^{n} Z_i^X \times Z_i^Y}{n}$	$cor(X,Y) = -1$ perfect negative correlation  $cor(X,Y) = -0.8$  $cor(X,Y) = -0.5$  $cor(X,Y) = 0$ no linear relationship  $cor(X,Y) = 0.5$  $cor(X,Y) = 0.8$  $cor(X,Y) = 1$ perfect positive correlation 

3.7.2 R SYMBOLS AND OPERATORS

code	description	example(s)
^	operator that raises a number to a power; the number that follows this symbol is the power, that is, the number of times we want to multiply the preceding number by itself	3^2 # raises 3 to the 2nd power (3^2=9)

3.7.3 R FUNCTIONS

function	description	required argument(s)	example(s)
table()	creates the frequency table of one variable or the two-way frequency table of two variables	code identifying the variable(s) (separated by a comma, if two) optional argument exclude: if set to equal NULL, the table includes NAs	table(*data$variable*) # frequency table table(*data$variable1, data$variable2*) # two-way frequency table table(*data$variable*, exclude=NULL) # includes NAs

prop.table()	converts a frequency table into a table of proportions and a two-way frequency table into a two-way table of proportions	either (a) the name of the object containing the output of the function table() or (b) the function table() directly; in both cases the code identifying the variable(s) should be specified inside the parentheses of table()
		
		optional argument margin for two-way table of proportions: if set to equal 1, the first specified variable defines the groups of reference; if set to equal 2, the second specified variable defines the groups of reference; if unspecified, the whole sample is the reference group
			freqtable <- table(*data$variable*)
			prop.table(freqtable) # or
			prop.table(table(*data$variable*))
			# table of proportions
			
			prop.table(table(*data$variable1, data$variable2*))
			# two-way table of proportions; the whole sample is the reference group
			
			prop.table(table(*data$variable1, data$variable2*, margin=1))
			# two-way table of proportions; variable1 defines the reference groups

na.omit()	deletes all observations with missing data from a dataframe	name of object where the dataframe is stored	na.omit(*data*)
hist()	creates the histogram of a variable; by default, it creates the histogram where the heights of the bins indicate frequencies	code identifying the variable optional argument freq: if set to equal FALSE, the function creates the density histogram	hist(*data$variable*) # frequency histogram hist(*data$variable*, freq=FALSE) # density histogram
mean()	calculates the mean of a variable; by default, it does not exclude missing values	code identifying the variable optional argument na.rm: if set to equal TRUE, R ignores the NAs when computing the average of the variable	mean(*data$variable*) # without removing NAs mean(*data$variable*, na.rm=TRUE) # removing NAs

continues on next page...

3.7.3 R FUNCTIONS (CONTINUED)

function	description	required argument(s)	example(s)
median()	calculates the median of a variable	code identifying the variable	median(*data$variable*)
sd()	calculates the standard deviation of a variable	code identifying the variable	sd(*data$variable*)
var()	calculates the variance of a variable	code identifying the variable	var(*data$variable*)
plot()	creates the scatter plot of two variables	two, separated by a comma and in this order: (1) variable on the x-axis (2) variable on the y-axis alternatively, we can specify the arguments x and y to indicate which variables we want to plot on the x and y axes, respectively	## all of these pieces of code produce the same scatter plot: plot(*data$x_var, data$y_var*) plot(x=*data$x_var*, y=*data$y_var*) plot(y=*data$y_var*, x=*data$x_var*)
abline()	adds a straight line to the most recently created graph; by default, it draws a solid line	to add a vertical line, we set the argument v to equal the value on the x-axis where we want the line; to add a horizontal line, we set the argument h to equal the value on the y-axis where we want the line optional argument lty: if set to equal "dashed", R draws a dashed line instead of a solid one	abline(v=2) # draws solid vertical line at 2 abline(h=3) # draws solid horizontal line at 3 abline(v=3, lty="dashed") # draws dashed vertical line at 3
cor()	calculates the correlation coefficient between two variables	code identifying the two variables, separated by a comma and in no particular order	cor(*data$variable1, data$variable2*)