CENTRAL TENDENCY AND VARIABILITY

MEASURES OF CENTRAL TENDENCY

Measures of central tendency indicate the central, average, or typical values of a distribution. Choosing the correct measure of central tendency for your data requires some knowledge about the variable(s). In particular, the level of measurement must be known, and in some cases, it would be useful to know if the distribution is skewed. These issues are discussed in Chapter 3.

The mean is a measure of central tendency suitable to calculate for variables measured at the interval or ratio level. However, when the distribution of an interval or ratio level variable is skewed, either positively or negatively, the median is likely a better reflection of the center of the distribution. For example, personal income in the United States is a variable that is positively skewed; a relatively small number of individuals make an astronomical amount of money.

In calculating the mean for personal income, the relatively few extremely large income amounts will cause the mean to be artificially high as a representation of the middle of the distribution. The median, however, locates the true middle, where 50% of the distribution lies above it and 50% of the distribution lies below it. As such, it cannot be artificially inflated or deflated by a relatively small number of outliers.

Using the *General Social Survey* 2016 data (Smith, Davern, Freese, & Morgan, 2019), we calculate the mean. The first logical step to building this command would be to write `mean(GSS2016$age)` to calculate `age`. This will work but only if there are no missing (NA) cases. If any cases are missing, then R will return the response NA. Essentially, if any of the cases have an unknown value for age (e.g., if someone refused to tell the interviewer their age), then R informs you that the mean is truly unknown, as well, since, technically, the mean truly is unknown without having the age of each and every person from the sample included in the calculation. That is very proper of R, but it does not help when

you need to compute the mean for the available data! So, the solution is simple and can be programmed as follows:

```
mean(GSS2016$age, na.rm=TRUE)
```

```
GSS2016$age[GSS2016$age==99]=NA
```

Tip: The addition of the code **na.rm=TRUE** tells R to remove the unknown entries from the list that will be used to make calculations (in this case for the mean).

```
49.15576
```

As you can see the average age of the *General Social Survey* (Smith et al., 2019) respondent in 2016 was just over 49 years old.

The median is a measure of central tendency that is appropriate to calculate for variables measured at the ordinal level. There are also variables for which the median would be appropriate to calculate for some interval or ratio level variables, but typically only when those variables are either positively or negatively skewed. Also, you might be interested in calculating both the median and the mean to see if a distribution is skewed. For example, if the mean and the median are approximately equal, then the distribution is said to be symmetric. If the mean is much larger than the median, this is an indication that the distribution is positively skewed. Similarly, if the mean is much smaller than the median this would present evidence of a negatively skewed distribution.

In developing the appropriate command code to calculate the median, you must keep in mind the same issue we encountered above with the mean surrounding missing or unknown data. Failure to remind R that we are not interested in considering the unknown values will result in a swift return of NA by R. To avoid that issue, be sure to include the details in your command to remove unknown data from the list:

```
median(GSS2016$age, na.rm=TRUE)
```

```
49
```

The mode is a measure of central tendency that is appropriate for variables measured at the nominal level. There are some instances where the mode could be used for some variables measured at the ordinal level, too.

Using R to calculate the mode can create considerable frustration for users, particularly new users. Simply put, there is no function in R to calculate the mode. It seems like a simple thing, yet it is just not available. As with any programming language, if there is not a direct method to complete a task, you can create a program to take advantage of other tools to complete your work, but this involves the creation or usage of a whole new program routine just to request a very simple statistic. We recommend that you save this program so that you can call on it each time you need to compute the mode.

If we are interested in knowing the mode of age in the 2016 *General Social Survey* (Smith et al., 2019) data, we can use the following command code to produce the mode. Note that R will return the value of the mode first; then in the next row R will provide the frequency of that category, as well:

```
table _ age <- table(GSS2016$age)

subset(table _ age, table _ age==max(table _ age))
```

```
57
70
```

In this case, the mode for age within the GSS data file is 57. That is, the most frequent age that respondents reported was 57. The second row indicates that there were 70 respondents who were 57 years old.

MEASURES OF VARIABILITY

Measures of variability are appropriate tools to complement measures of central tendency. While measures of central tendency illustrate where the middle of a distribution lies, measures of variability explain how far from that middle the distribution tends to fall. Most often, these two types of measures are calculated and used together to describe a distribution. Determining which measure of variability is appropriate for your data requires understanding how the data were collected and at what level they were measured: nominally, ordinally, or on an interval or ratio basis. The following measures require at least an ordinal level of measurement.

Range

The range is a value that expresses the full variation across a variable, from the lowest value to the highest value. If we were to compute this by hand, we would take the maximum value (highest value) and subtract the minimum (lowest value) from that. This would yield the range. Another way of thinking of the range is to include both the maximum and minimum values, stating the interval that covers the variable. (One can always perform the subtraction if, for instance, the value of the range is being used in comparison with other variables' ranges.) Below, you will find the command code requesting R to produce the range. In this case, the range for respondent's age in the *General Social Survey* (Smith et al., 2016) has been requested. The result is a range from 18 to 89. In other words, the youngest respondent was 18 years of age and the oldest respondent was 89 years old. The code is shown as follows:

```
range(GSS2016$age, na.rm=TRUE)
```

```
18 89
```

While the range is useful, it is highly susceptible to the effect of outliers. If there were just one person in a sample, for instance, who is 99 years old, but the next youngest person was 65, the range would be inflated to 99 because of only one person—even if the sample contained thousands of people! As you can see, the range can very often overstate the variation in a variable. The solution to this problem brings us to the next measure to be discussed, the IQR (inter-quartile range).

IQR (Inter-Quartile Range)

The inter-quartile range (IQR) is a measure of variation for use with interval ratio variables. It could also be used with some ordinal variables. The IQR reveals the width of the middle 50% of a distribution. Since the IQR represents the middle 50% of a distribution, any outliers on either end of the distribution will not have an effect on the computed value for the IQR, making this a more valid assessment of variation for distributions with—or even without—outliers. We can define the IQR as the difference between the upper and lower quartiles (Q3 and Q1, respectively). The lower quartile (Q1) represents the 25th percentile, while the upper quartile (Q3) represents the 75th percentile.

$$IQR = Q3 - Q1$$

$$Q3 = 75\text{th percentile}$$

$$Q1 = 25\text{th percentile}$$

Now we will compute the IQR for the variable, `age`, in the *General Social Survey* (Smith et al., 2019) using R. You will need to utilize the following command code:

```
IQR(GSS2016$age, na.rm=TRUE)
```

```
28
```

Not surprisingly, the command to compute the IQR is **IQR**. Once again, it is important to make sure that missing data are omitted from the calculation, so `na.rm=TRUE` tells R to remove any instances of NA or unknown data from the calculation. The output from R after running the command yields 28. This means that there are 28 years between the age of respondents at the 75th percentile (Q3) and the age of respondents at the 25th percentile (Q1).

Variance

Variance is a measure of variation suitable for interval and ratio variables. The variance can be computed by taking the average of the squared deviations from the mean of a distribution—the result is always a positive number ranging from zero to infinity. Zero would be indicative of absolutely no variation whatsoever; all elements of the distribution would be the same on the variable being measured (e.g., all respondents having the exact same annual income would yield a variance of zero for the annual income distribution). As the computed variance begins to increase, so too does the variation within a distribution.

In order to have R compute the variance of age in the *General Social Survey* 2016 data (Smith et al., 2019), use the **var** command.

Tip: Again, remember to include the code to remind R to exclude missing or unknown cases from the list (otherwise R will return an NA).

```
var(GSS2016$age, na.rm=TRUE)
```

```
313.0349
```

Notice that the computed variance is *much* larger than any person's age in the sample. You might be wondering how the value of the variance corresponds to the scale of the distribution of the variable, age. Part of the reason that the variance is not used—and instead the standard deviation has emerged as the pervasive comparable variability measure—is that the scale does not match and cannot be directly compared with the values, including the mean. It can, however, be compared with variance values from other distributions.

Standard Deviation

The standard deviation is a measure of variation for interval/ratio variables. The standard deviation is calculated simply by taking the square root of the variance. Like the variance, the standard deviation cannot be negative and can range from zero to infinity. At zero, the standard deviation would indicate that the distribution has absolutely no variation and therefore all elements of the distribution are the same (e.g., all members of the sample are exactly the same age in years). As the calculated value of the standard deviation becomes larger, the greater the indication of variation in the sample and, by extension (when using a representative sample), in the population. In order to have R compute the standard deviation of the variable age from the *General Social Survey* (Smith et al. 2019), you will need to use the **sd** command:

```
sd(GSS2016$age, na.rm=TRUE)
```

```
17.69279
```

Now you can see that the standard deviation seems to be in an appropriate scale with the distribution of the variable, age. In fact, the standard deviation provides critical information that will ultimately lead to making inferences about a population from a sample. The next step in that process is to discuss standard scores, also called *z*-scores.

THE *Z*-SCORE

A *z*-score or standard score is a value that denotes how many standard deviations away from the mean a particular raw score lies. This could be an indication of a raw score

being either above or below the mean. A positive (+) *z*-score is an indicator of a raw score that is greater than the mean. A negative (–) *z*-score is an indicator of a raw score that is lower than the mean. As explained below, a *z*-score that is neither positive nor negative, but equal to zero, is an indicator that the particular raw score happens to be equal to the value of the mean.

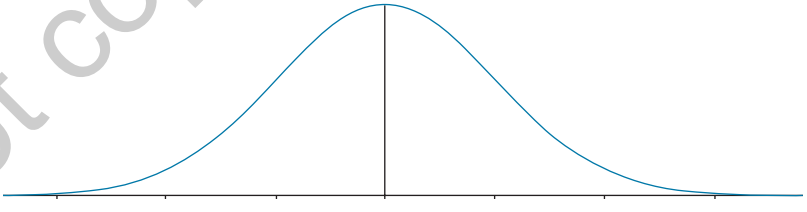
A raw score is the value of a particular case on a variable in your dataset. So, 65 years might be the raw score for the variable, `age`, in your dataset. To compute the *z*-score for that, we need some additional information first.

It is important to remember that this work surrounding *z*-scores ultimately involves using a sample to make predictions about a population using a theoretical family of distributions (the standard normal distribution). This is called inferential statistics. As such, it is assumed that the sample is representative of the population. Later in this book (starting with Chapter 6), we will show how to use R to compute inferential statistics. It might help to consult a statistics book for assistance understanding representative sampling, such as *Using and Interpreting Statistics in the Social, Behavioral, and Health Sciences* (Wagner & Gillespie, 2018).

The normal distribution is a bell-shaped, symmetrical, theoretical distribution (see Figure 4.1). In fact, it is a family of theoretical distributions that adheres to certain principles. The mode, median, and mean are all equal, coinciding at the peak in the middle of this theoretical distribution. Frequencies decrease as you move in either direction toward the ends of the curve.

The normal distribution is an ideal distribution. Real-life empirical distributions will not perfectly mirror this ideal type. However, a great many things in life do approximate the normal distribution, and we can say that they are "normally distributed."

FIGURE 4.1 ⬢ AN EXAMPLE OF A NORMAL DISTRIBUTION WHERE THE MEAN, MEDIAN, AND MODE ALL COINCIDE AT THE SAME POINT



How to Calculate a *z*-Score

Let's start with a real-world scenario and some data. In Dr. Handelsman's biostatistics course, the average final score was 75/100 (75%) and the standard deviation of those scores was 10 percentage points.

We also know that the scores were approximately normally distributed. This information reveals a great deal—the middle of the distribution is at 75 and the standard deviation of 10 illustrates how steep the curve is on both sides.

To convert a raw score (your score earned in the class, for example), use the following formula:

$$z = \frac{Y - \overline{Y}}{S_y}$$

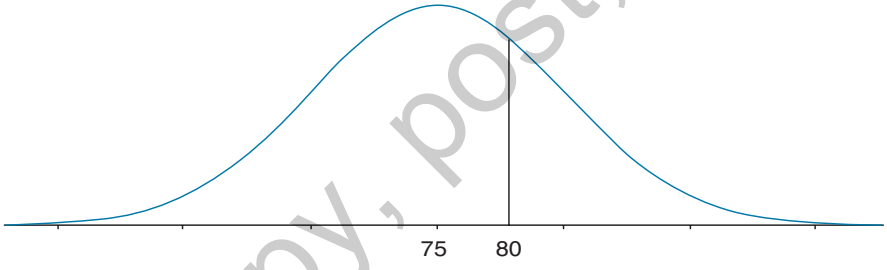
The *z*-score, or standard score, represents how many standard deviations away from the mean that particular raw score lies. As mentioned, when the *z*-score is positive, the raw (original) score lies above the mean. If the *z*-score is negative, the raw score is lower than the mean. If the *z*-score is equal to 0, that means the raw score must have been exactly equal to the mean (if your raw score is equal to the mean, there is no need to even do any calculation—the *z*-score is 0).

So, if your score was 80, the *z*-score should be positive, since it is greater than the mean (75) and therefore on the right side of the normal curve. To confirm,

$$z = (80\text{-}75)/10 = .5$$

Therefore, the *z*-score associated with a raw score of 80 in this distribution is .5 (one half SD above the mean). See the graph, below, for a visual representation of the location of the raw score and note that the location corresponds to a positive *z*-score:

FIGURE 4.2

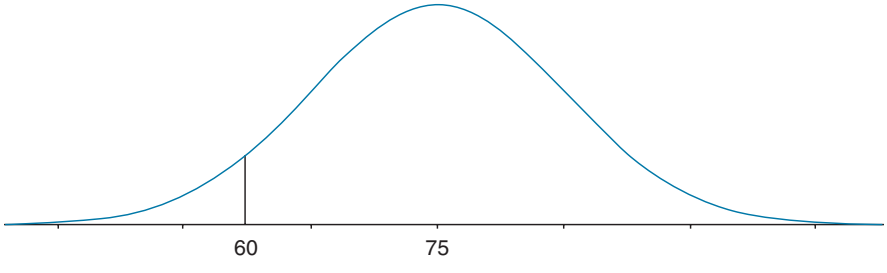


What about a raw score of 60, someone who barely passed the course with a D-? This *z*-score will be negative since the raw score is less than the mean of 75:

$$z = (60\text{-}75)/10 = -1.5$$

So, the *z*-score associated with a raw score of 60 in this distribution is -1.5, as shown in the following:

FIGURE 4.3



Suppose you want to convert all of the ages of the respondents in your dataset to standard scores (*z*-scores). One way to do that is to use the **scale()** command.

Since the dataset with which we are working (GSS 2016) is fairly large at over 2,800 cases or respondents, we will need to address the line limit in R. The default setting is only 1000. You can see what the setting is currently on your version of R by typing the following command:

```
getOption("max.print")
```

```
1000
```

Then, you can change this limit to a different value. To be on the safe side, you can select a number much higher than the GSS 2016 sample size. We will try 5000.

```
options(max.print=5000)
```

Now we can use the **scale** command to instruct R to return standardized scores for all of the values of respondents' ages. Below, you will find the command code, but this time we will not include the output:

```
scale(GSS2016$age, center=TRUE, scale=TRUE)
```

What you will be given is a very long table showing all 2,867 standard scores (*z*-scores).

This procedure is particularly useful for much smaller data files. When working with the GSS, it becomes a bit cumbersome.

We have already learned how to calculate the mean and the standard deviation earlier in this chapter, so in order to calculate a *z*-score in R, we simply need a raw score, the Y value. We can convert this formula to terms that R will understand in command language. Below is an example, using 72 as the case for which we would like to calculate a *z*-score. Notice that we have identified the mean of the variable `age` in the GSS data file and the standard deviation of the variable `age` in the GSS data file. We have also taken the precaution to tell R to ignore any cases that are missing or unknown while calculating both the mean and standard deviation.

```
(72 - (mean(GSS2016$age, na.rm=TRUE)))/(sd(GSS2016$age,
na.rm=TRUE))
```

```
1.291161
```

The *z*-score given is 1.29. Since this is a positive number, we know that a positive *z*-score is associated with a number larger than the mean. An age of 72 years is indeed larger than the mean that we calculated earlier in this chapter: 49.16 years of age.

SELECTING CASES FOR ANALYSIS

For certain analyses, researchers may not be interested in including all of the cases from a particular data file. There are many reasons why this might be the case—and it often is. For example, if the researcher is interested in studying only the characteristics of persons 21 years and over, then it will be necessary to remove all of the respondents under 21 years of age from the analysis.

This condition can commonly emerge when working with secondary datasets—a data file created by a third party, such as the *General Social Survey* (Smith et al., 2019) data that we examine in this book. Because the dataset was not originally custom tailored to meet your particular needs, you will likely need to select the appropriate cases or respondents (as well as possibly recode variables into new variables).

There are several available approaches for selecting cases, and the best decision is usually based on your purposes for doing so. It is possible that you may want to work with a smaller subsample of a data file and therefore might want to select a random number or fraction of cases or rows from the original data file. Alternatively, you may want to filter out respondents with certain characteristics, such as being below or over a certain age, or select men or women only. Each of these purposes requires a different sequence of command code.

Before we begin, however, we need to revisit our discussion of R packages. The package that allows us to sample and select cases is called **dplyr**. If you have installed the **tidyverse** package, then you will not need to download the package; all you need to do is enter the following command code:

```
library(dplyr)
```

Once the **dplyr** package is active, R can respond to commands related to selecting cases randomly or based on criteria, including scores on one or more variables.

First, we will begin by discussing how to create a random subsample of data. Suppose you are in need of just a very small data file to work with, one containing only five cases or respondents. We can do this by creating a new data file (data frame) in R. The command code to select five random cases from the GSS 2016 data file is as follows:

```
GSS2016random <- sample_n(GSS2016, 5, replace = FALSE)
```

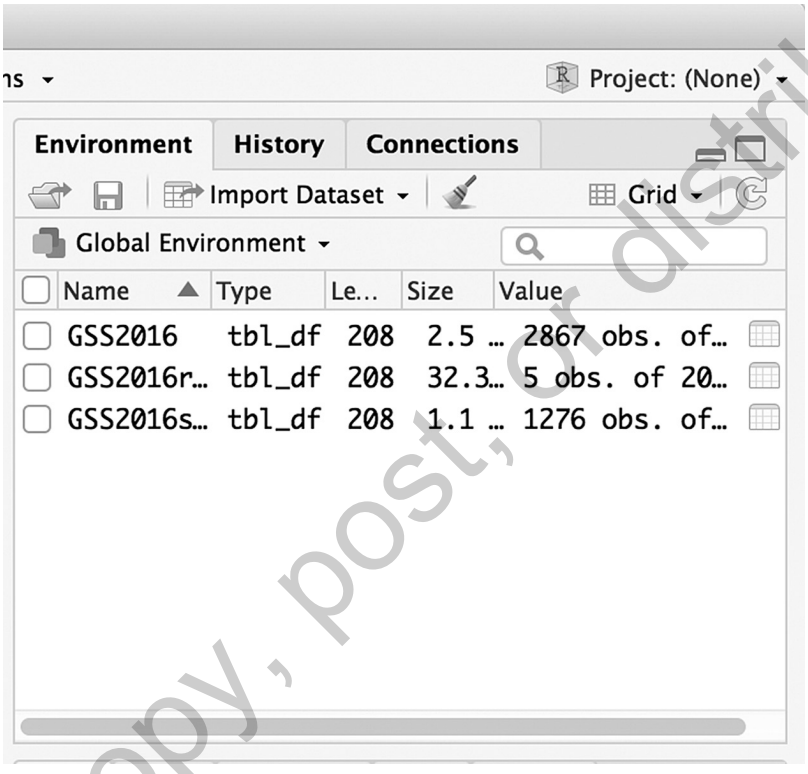
The first term, `GSS2016random` is the name of the new data frame (designated by "<-"). Then, as one might imagine, the **`sample_n`** command selects a certain number of cases; in this case, of course, we have selected five cases.

Now, suppose instead of random case selection, you have an interest in using a variable to select cases from the main data file. If your research involves men only, then you can use the `sex` variable to choose only men for a subsample to analyze. The following command code will accomplish that and place the resulting cases in a data file (data frame) called GSS2016sub:

```
GSS2016sub <- subset(GSS2016, sex==1)
```

Notice that the upper right frame of the R window, we can now see the original GSS2016 dataset listed first, but now we also have two new data sets: GSS2016random and GSS-2016sub. These are the two datasets that we just created. You can save them or even export them to different files for use in other programs if need be.

FIGURE 4.4



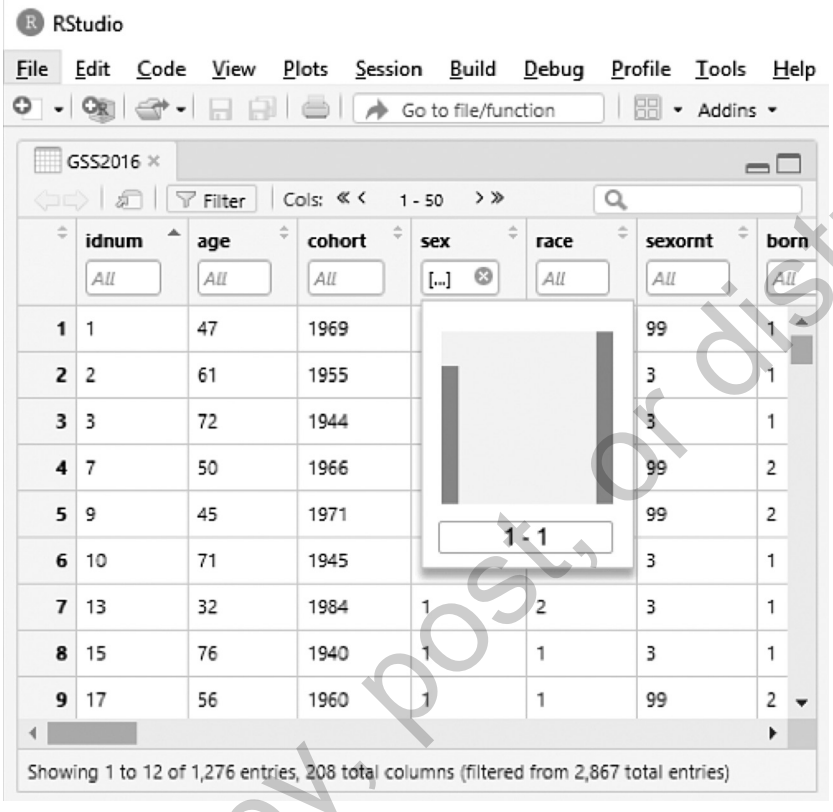
For those who prefer the point-and-click method, there is another way to accomplish this. We will review the case of selecting only the men for analysis.

In the upper left pane of the R window, you will see the data matrix. Rows represent cases and columns represent variables. If you click the word "Filter", you will see fill-in boxes appear beneath each of the variable names. There will be gray italicized printing that says "All" under each variable name.

If you click there, you will have the option to change what values are filtered. In this case, after changing the variable `sex` from a range from 1 to 2, to a range of 1 to 1 (just 1—just males), you can look toward the bottom of that pane to see that the number of cases in the data frame has decreased from 2867 to 1276, as a result of the filtering operation. We can also confirm that this is the same number of observations (i.e., cases) as in the newly created subsample above, GSS2016sub.

Measures of central tendency and variability showcase the basic features of a distribution. Standard scores (*z*-scores) allow for the raw data to be standardized (using *z*-scores)

FIGURE 4.5 ⬢ SELECTING A SUBGROUP FOR ANALYSIS USING THE POINT-AND-CLICK METHOD



and are also important for inferential statistics, which are covered later in this book. Sampling—selecting cases for analysis—of course, plays a key role in working with subsets of populations as well as understanding sampling distributions in inferential statistics.

CONCLUSION

R allows you to compute measures of central tendency, measures of variability, and to work with standard (*z*-) scores. From here, it is possible to work with percentiles and probabilities within a distribution. These operations help form the foundation of univariate data analysis, to begin understanding the nature of the distribution of a single variable. (Later in this book, we will discuss bivariate and multivariate relationships.) In this chapter, we also learned how to work with the cases in a dataset, selecting cases tailored to the particular needs of the analysis. Among other things, this can be helpful for drawing samples and conducting an analysis of different segments of a population/dataset.

References

Smith, T. W., Davern, M., Freese, J., & Morgan, S. L. (2019). *General Social Surveys, 1972–2018* [Machine-Readable Data File]. Chicago, IL: NORC at the University of Chicago. Retrieved from http://www.gss.norc.org/getthedata/Pages/Home.aspx

Wagner, W. E., III, & Gillespie, B. J. (2018). *Using and interpreting statistics in the social, behavioral, and health sciences.* Thousand Oaks, CA: Sage.

Exercises

1. Using R, how would you hypothetically compute the average for a variable: number of dog walks per week (walks) in a dataset, named: `socialdogs`? In other words, what command would you type to return the mean as the result? Assume that the dataset is already open in R and remember to exclude missing cases from the calculation.
2. Using R, how would you create a subset of a data file, named `CAjobs2019`, that only includes people who were employed full time, at the time the data were collected. Assume that there is a variable called `employed` and the value for full-time employment is *1*.
3. What command would you use to compute the inter-quartile range using R?
4. Which measure of central tendency cannot be directly computed with a single command using R, and instead requires a short work-around?

Supplementary Digital Content

Download datasets and R code at the companion website at https://study.sagepub.com/researchmethods/statistics/gillespie-r-for-statistics