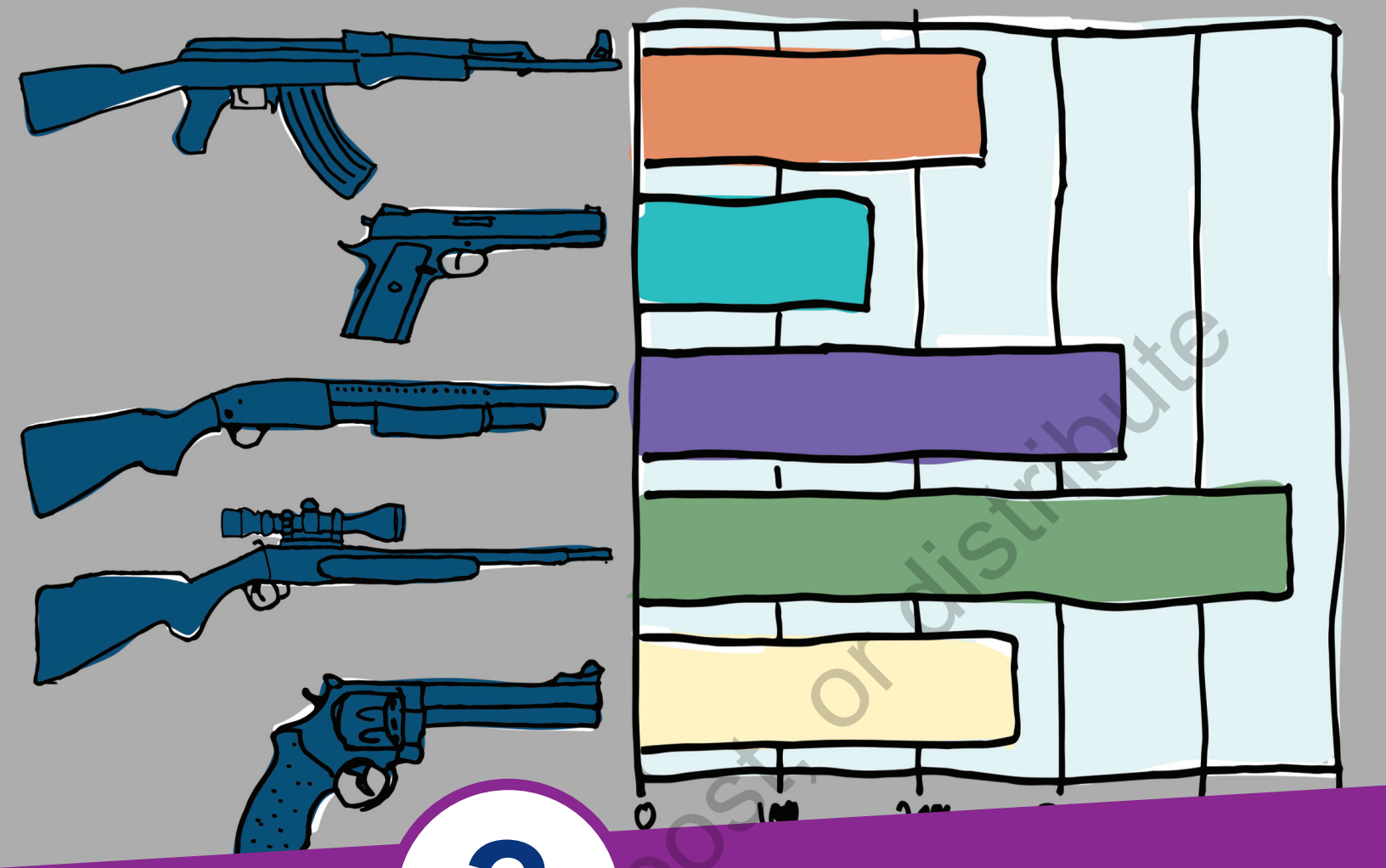


3

DATA VISUALIZATION

The R-Team and the Tricky Trigger Problem

It was just their third R-Team meeting, but Nancy and Kiara were impressed at how quickly Leslie had been learning R coding. They were meeting today at the business school of Leslie's university. Leslie had described a big open area with a lot of tables and a little coffee shop in the corner, which sounded perfect to Kiara and Nancy. Leslie was already on campus that morning, so she arrived first, but it was only a minute or two before Nancy and Kiara walked up. After a quick delay for coffee, they were all seated and ready to go.

"You're killing it on the coding front," said Kiara.

"Genius!" Nancy exclaimed. "Which means it's time to teach you all about one of the biggest strengths of R: making graphs! Graphs are so important for adding context to the numbers."

"For sure," Kiara said. "Remember last meeting when the `PHYSHLTH` histograms showed so clearly that it was not normally distributed?" A quick histogram review was enough evidence to suggest the appropriate descriptive statistics to use for the situation.

"I remember," said Leslie.

Nancy continued, "In addition to helping choose the right statistics to use, graphs are one of the best ways to communicate about data with various audiences."

"What are we waiting for?" Leslie asked.

"I like your enthusiasm," said Kiara.

Nancy explained that they would primarily use the **`ggplot2`** package from the **`tidyverse`** to create and format common graphs used to display data. She promised to cover which graphs are appropriate for different data types, the important features of a well-formatted graph, ways to avoid creating a misleading graph, and how to interpret graphs.

Nancy and Kiara created a list of achievements for Leslie's third R-Team meeting.

3.1 Achievements to unlock

- Achievement 1: Choosing and creating graphs for a single categorical variable
- Achievement 2: Choosing and creating graphs for a single continuous variable
- Achievement 3: Choosing and creating graphs for two variables at once
- Achievement 4: Ensuring graphs are well-formatted with appropriate and clear titles, labels, colors, and other features

3.2 The tricky trigger problem

Leslie's friend Leanne was very involved in an activist group called Moms Demand Action (https://momsdemandaction.org/) and had been sending Leslie information about guns and gun violence in the United States. Leslie had emailed Kiara and Nancy some of what she understood about this problem and the lack of research related to gun violence.

Note: In shaded sections throughout this text, the rows starting "##" show the output that will appear after running the R code just above it.

3.2.1 COMPARING GUN DEATHS TO OTHER CAUSES OF DEATH IN THE UNITED STATES

The United States has a high rate of gun ownership and a similarly high rate of gun injury and death (Giffords Law Center, n.d.). However, there has been relatively little research into gun injury and gun violence in recent decades after government funding was limited by the Dickey Amendment, a 1996 appropriations bill that cut $2.6 million from the Centers for Disease Control and Prevention (CDC) budget (Kellermann & Rivara, 2013). The Dickey Amendment, named after the Arkansas representative who introduced it, removed the funds in an effort by some members of the U.S. Congress to eliminate the National Center for Injury Prevention and Control (Kellermann & Rivara, 2013): "None of the funds made available for injury prevention and control at the Centers for Disease Control and Prevention may be used to advocate or promote gun control" (Omnibus Consolidated Appropriations Act of 1997, p. 244; Rubin, 2016). While the bill failed to eliminate the center, it was successful in eliminating funding for research on gun injury and violence.

An article published in the *Journal of the American Medical Association* (*JAMA*) used publicly available data to compare the amount of research money spent on each of the top 30 causes of death in the United States, including gun violence, between 2004 and 2015 (Stark & Shah, 2017). The authors also examined the number of publications describing research findings for the top 30 causes of death in the United States over the same time period. With the exception of falls, research on gun violence as a cause of death had the lowest research funding level of any of the top 30 causes of death. With the exception of drowning and asphyxia, gun violence was the topic of the fewest publications of any of the top 30 causes of death in the United States. Before their meeting, Nancy and Kiara requested the data for the gun violence article from the authors and reproduced some of the figures.

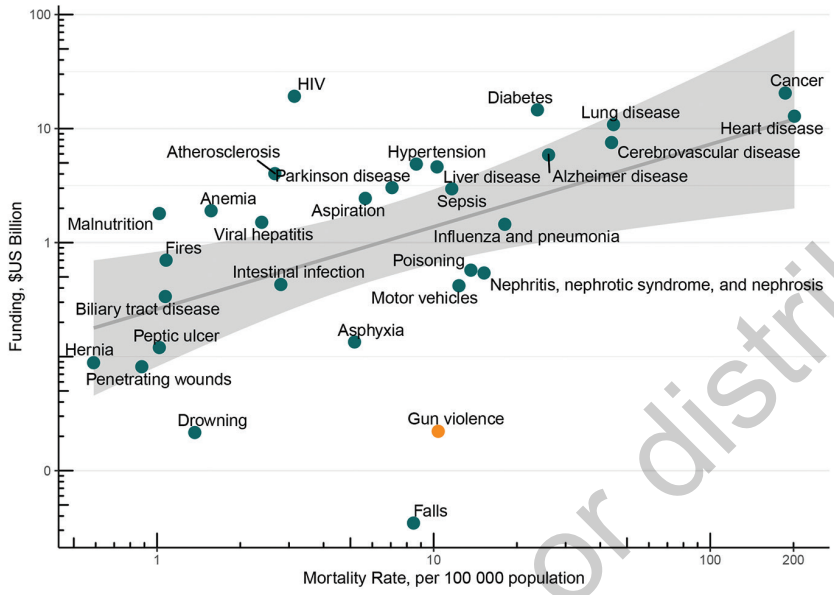
The figures in the article are ***scatterplots***, which show one dot for each observation in the data set. In this case, each dot represents one of the top 30 causes of death in the United States. There is a label on each dot showing which cause of death it represents. The *x*-axis (horizontal) is the mortality rate, or the number of deaths per 100,000 people per year in the United States. The *y*-axis (vertical) shows the amount of funding spent on research. The relative position of each dot on the graph shows how many people the cause of death kills and how many dollars of funding are available for research. Causes with dots in the lower left have lower levels of mortality and lower levels of funding. Causes with dots in the upper right have higher mortality and higher research funding. Overall, as the mortality rate rises, the amount of research funding also rises. There are two exceptions, falls and gun violence, which are toward the middle of the group for rate of mortality but at the bottom for research funding levels. Overall, gun violence has the second lowest funding for research of the top 30 mortality causes (see Figure 3.1).

The second figure reproduced (Figure 3.2) from the paper (Stark & Shah, 2017) shows a similar pattern of number of publications on each topic on the *y*-axis and mortality rate on the *x*-axis. This time, there are four mortality causes that do not fit the pattern of more publications for higher mortality rates: drowning, asphyxia, aspiration, and gun violence. Of these, gun violence has the highest mortality rate. Overall, gun violence has the third lowest publication rate of the top 30 mortality causes.

3.2.2 WEAPONS USED IN HOMICIDE

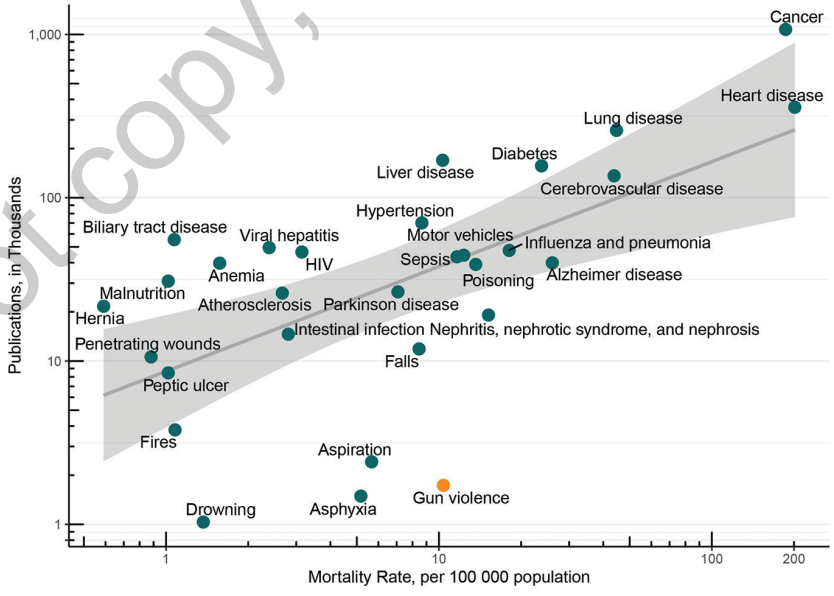
To get a sense of the extent of gun violence in the United States, Nancy and Kiara looked up the annual Federal Bureau of Investigation (FBI) crime data reported in the Uniform Crime Reporting program (https://www.fbi.gov/services/cjis/ucr). They found that one of the data sources included the types of weapons used in homicides, such as firearms, knives or cutting instruments, blunt objects, and several other categories. Kiara and Nancy decided to make a bar chart of gun and non-gun homicides from the FBI data for the most recent 5 years reported. They flipped the axes so that the bars are horizontal. The *x*-axis shows the number of homicides, while the *y*-axis shows the year. In each year, the number of homicides by gun (green bars) was more than 2.5 times higher than all non-gun weapons combined (purple bars). For more than one of the years, there were three times as many homicides by gun as by all non-gun weapons combined (Figure 3.3).

FIGURE 3.1 Mortality rate versus funding from 2004 to 2015 for 30 leading causes of death in the United States



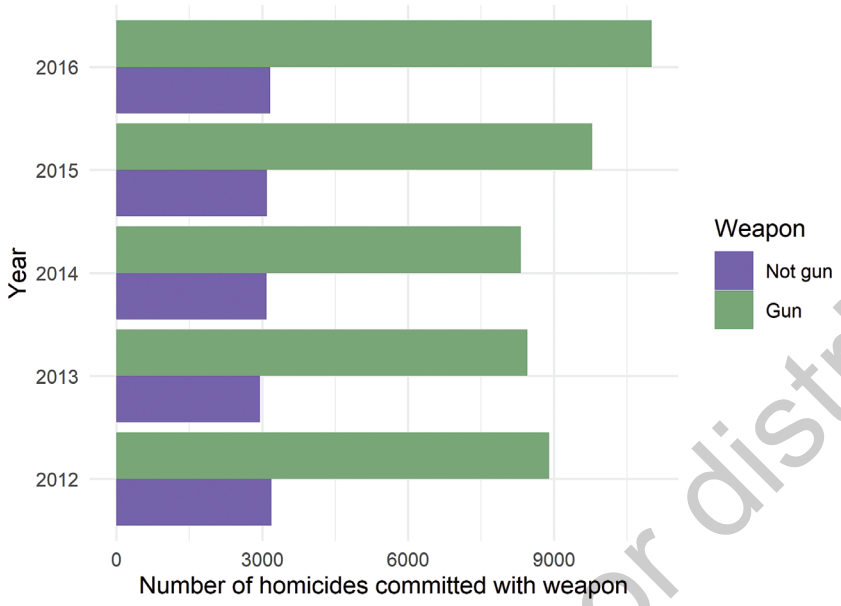
Source: Reproduced with permission from *JAMA*. 2017. 317(1): 84–85. Copyright©2017, American Medical Association. All rights reserved.

FIGURE 3.2 Mortality rate versus publication volume from 2004 to 2015 for 30 leading causes of death in the United States



Source: Reproduced with permission from *JAMA*. 2017. 317(1): 84–85. Copyright©2017, American Medical Association. All rights reserved.

FIGURE 3.3 Homicides by guns and known non-gun weapons per year in the United States, 2012–2016



Source: FBI Uniform Crime Reports data.

Nancy and Kiara were interested in finding other patterns that might be useful in understanding gun violence.

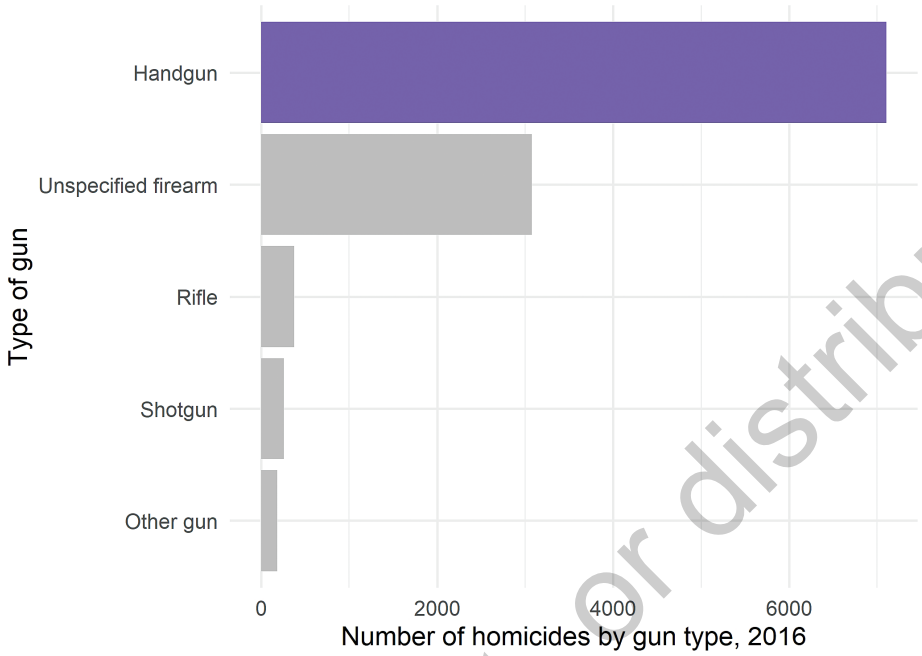
3.2.3 TYPES OF GUNS USED IN HOMICIDE

Looking a little deeper into the FBI data, Nancy and Kiara found that within the firearms category are handguns, rifles, shotguns, other guns, and unknown gun type. They made another bar chart that suggested handguns were the most widely used type of gun for homicide in 2016. The graph includes all the homicides by gun for 2016 and shows the number of homicides by each type of gun. The *x*-axis has the number of homicides, while the *y*-axis has the type of gun (Figure 3.4).

3.2.4 THE ROLE OF GUN MANUFACTURERS IN REDUCING GUN DEATHS

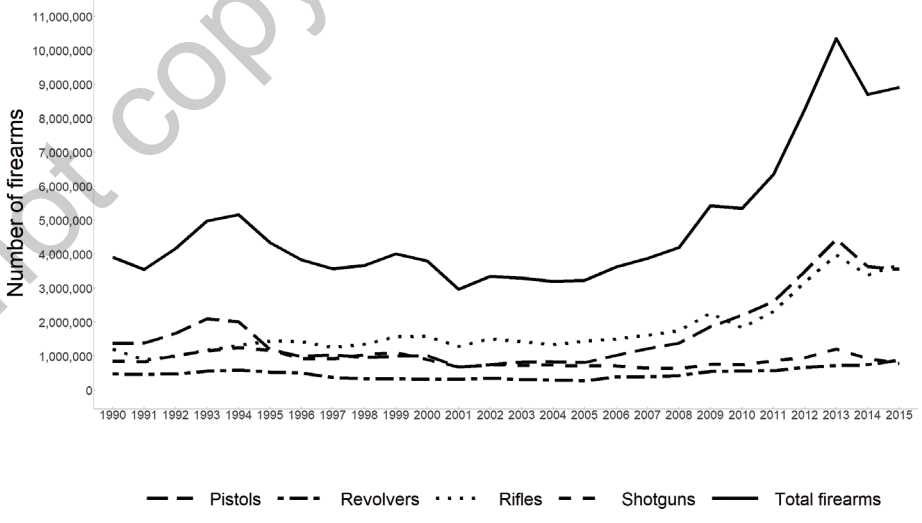
Leslie remembered another article she had read recently from her friend Leanne. It was about the role that gun manufacturers could potentially play in reducing gun violence (Smith et al., 2017). The authors of this article argued that there is little information about how gun manufacturing is related to gun ownership or gun violence. They suggested that a better understanding of manufacturing could identify changes in manufacturing practices to increase safety and reduce injury and death. The authors used publicly available data from the Bureau of Alcohol, Tobacco, Firearms, and Explosives to examine how many guns were manufactured in the United States over a 25-year period from 1990 to 2015. The authors also examined the types of guns manufactured during this period and the types of guns confiscated after use in crime. Kiara and Nancy worked together to reproduce the graph from the article (Figure 3.5). This time it was a line graph, which is the type of graph often used to show change over time. In this case, time is on the *x*-axis and the number of firearms manufactured is on the *y*-axis.

FIGURE 3.4 Types of firearms used in homicides in the United States, 2016



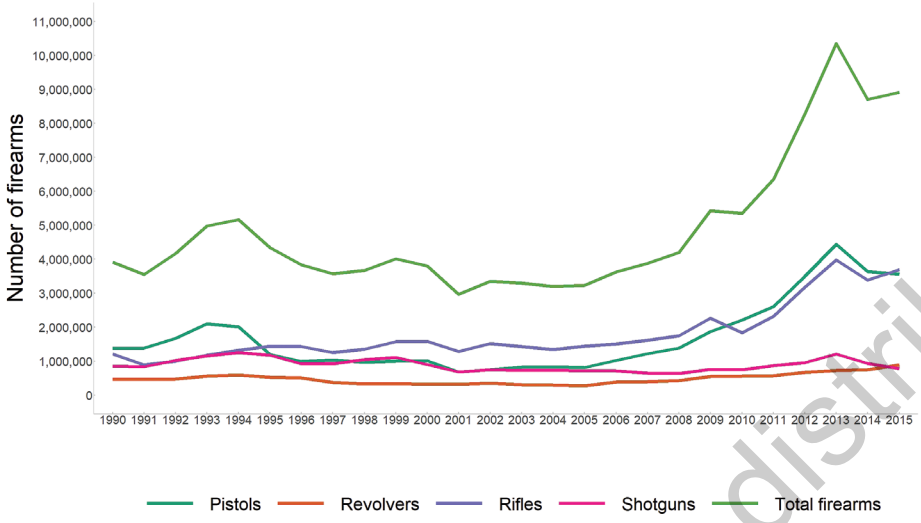
Source: FBI Uniform Crime Reports data.

FIGURE 3.5 Firearm types manufactured in the United States from 1990 to 2015



Source: U.S. Bureau of Alcohol, Tobacco, Firearms, and Explosives data.

FIGURE 3.6 Firearm types manufactured in the United States from 1990 to 2015



Source: U.S. Bureau of Alcohol, Tobacco, Firearms, and Explosives data.

Different types of lines show different types of guns being produced over time. Although the article used the line types to differentiate, Nancy and Kiara found it difficult to tell some of the line types apart, so they used solid lines and added color to the graph for easier reading. The resulting graph, Figure 3.6, showed a sharp increase in gun manufacturing after 2010, with the increase mostly being in the production of pistols and rifles.

Leslie, Nancy, and Kiara discussed how these four graphs tell a story about guns in the United States and may suggest policy solutions, such as funding research that examines the relationship, if any, between the types and quantities of guns manufactured and the number of gun homicides by weapon type. Nancy and Kiara explained to Leslie that the graphs shown here about gun research and manufacturing demonstrate just a few of the many ways to visualize data. ***Data visualization***, or graphing, is one of the most powerful tools an analyst has for communicating information. Three graph types were demonstrated by Nancy and Kiara: the scatterplot, the bar chart, and the line graph. Kiara explained to Leslie that these are not the only types of graphs, but they are common types. She explained that, like descriptive statistics, there are different ways to visualize data that are appropriate for each data type.

3.3 Data, codebook, and R packages for graphs

Before they examined the data, Kiara made a list of all the data, documentation, and packages needed for learning about graphs.

- Two options for accessing the data
  - Download the three data files from **edge.sagepub.com/harris1e**
    - **nhanes_2011_2012_ch3.csv**

    - **fbi_deaths_2016_ch3.csv**
    - **gun_publications_funds_2004_2015_ch3.csv**
  - Download the raw data directly from the Internet for the FBI deaths data and the NHANES data by following the instructions in Box 3.1 and download **gun_publications_funds_2004_2015_ch3.csv** from **edge.sagepub.com/harris1e**
- Two options for accessing the codebooks
  - Download from **edge.sagepub.com/harris1e**
    - **nhanes_demographics_2012_codebook.html**
    - **nhanes_auditory_2012_codebook.html**
  - Access the codebooks online on the National Health and Nutrition Examination Survey (NHANES) website (https://wwwn.cdc.gov/nchs/nhanes/Search/DataPage.aspx?Component=Questionnaire&CycleBeginYear=2011)
- Install the following R packages if not already installed:
  - `tidyverse`, by Hadley Wickham (https://www.rdocumentation.org/packages/tidyverse/)
  - `ggmosaic`, by Haley Jeppson (https://github.com/haleyjeppson/ggmosaic)
  - `waffle`, by Bob Rudis (https://github.com/hrbrmstr/waffle)
  - `gridExtra`, by Baptiste Auguie (https://www.rdocumentation.org/packages/gridExtra/)
  - `readxl`, by Jennifer Bryan (https://www.rdocumentation.org/packages/readxl/)
  - `ggrepel`, by Kamil Slowikowski (https://www.rdocumentation.org/packages/ggrepel/)
  - `scales`, by Hadley Wickham (https://www.rdocumentation.org/packages/scales/)
  - `httr`, by Hadley Wickham (https://www.rdocumentation.org/packages/httr/)
  - `data.table` (Dowle & Srinivasan, 2019)
  - `RNHANES` (Susmann, 2016)

3.4 Achievement 1: Choosing and creating graphs for a single categorical variable

The first thing the team wanted to work on were graphs appropriate for displaying single variables. Before selecting a graph type, it is useful to think about the goal of the graph. Kiara suggested that making graphs to check whether something is normally distributed before calculating a mean is very different from making graphs to communicate information to an audience. The team decided to start by creating graphs that convey information from a single categorical variable. Kiara reminded Leslie that a categorical variable has categories that are either ordinal, with a logical order, or nominal, with no logical order. Categorical variables are the factor data type in R. Nancy explained that single categorical variables have several options for graphing. Some of the more commonly used graphs for a single categorical variable are the following:

- Pie chart
- Waffle chart
- Bar chart
- Point chart

Pie charts and waffle charts are similar; they are both used for showing parts of a whole. Bar charts and point charts tend to be used to compare groups. Leslie chose to start with the pie chart since it sounded delicious and she had always preferred dessert to breakfast!

3.4.1 PIE CHARTS

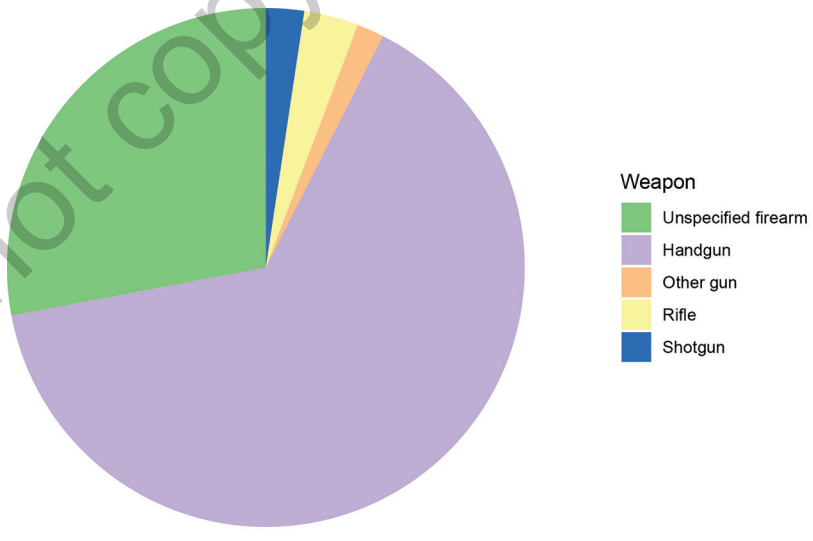
To create a pie chart for a single categorical variable, the team needed a variable to start with. Leslie suggested gun type from Figure 3.4. There were five categories of gun type represented in the graph: Other gun, Shotgun, Rifle, Unspecified firearm, and Handgun. Kiara thought focusing on one gun type might be one way to start understanding gun manufacturing. Maybe examining the type of gun with the highest (or lowest) quantity manufactured would be a good strategy? One way to do this would be to create a pie chart, like Figure 3.7.

Pie charts are meant to show parts of a whole. The pie, or circle, represents the whole. The slices of pie shown in different colors represent the parts. While pie charts are often seen in newspapers and other popular media, they are considered by most analysts as an unclear way to display data. A few of the reasons for this were summarized in an R-bloggers post (C, 2010):

- Pie charts are difficult to read since the relative size of pie pieces is often hard to determine.
- Pie charts take up a lot of space to convey little information.
- People often use fancy formatting like 3-D, which takes up more space and makes understanding the relative size of pie pieces even more difficult.

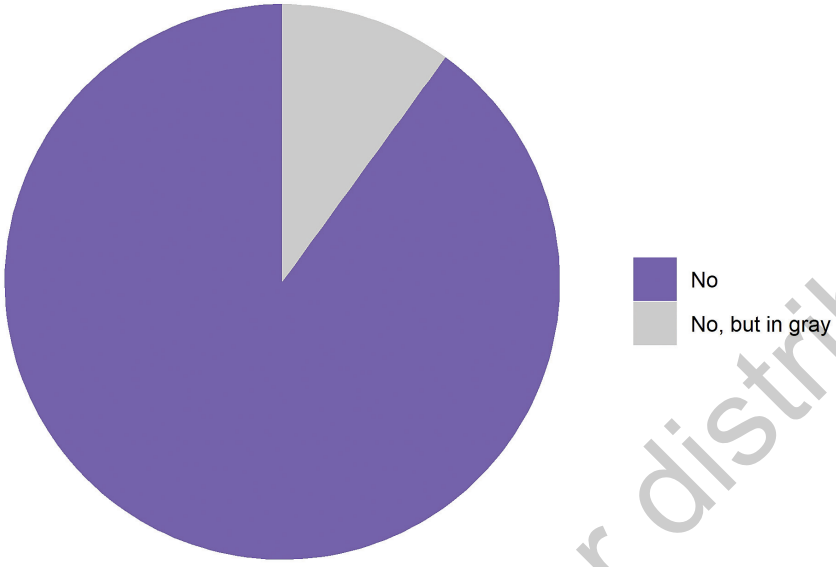
In fact, if you asked 100 data scientists, "Should I make a pie chart?" the answers might resemble Figure 3.8.

FIGURE 3.7 Firearm types manufactured in 2016 in the United States



Source: U.S. Bureau of Alcohol, Tobacco, Firearms, and Explosives data.

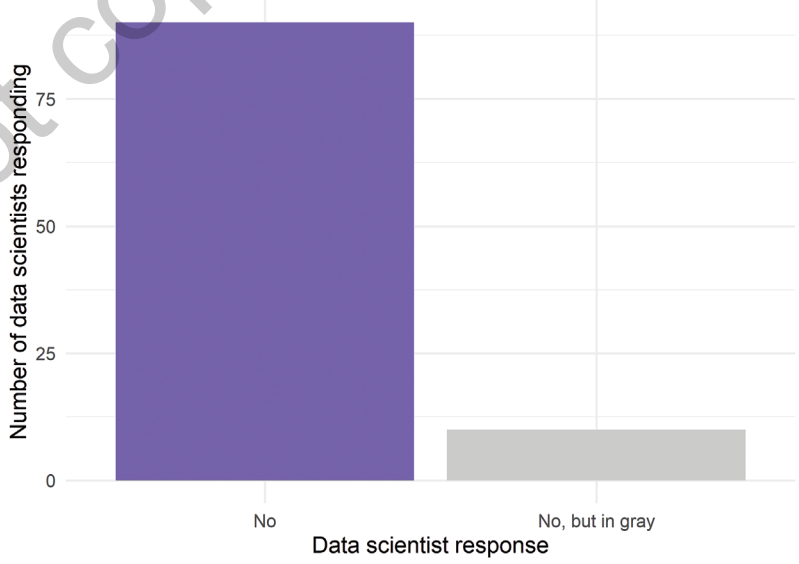
FIGURE 3.8 Should I make a pie chart?



A ***bar chart*** and a ***point chart*** are more effective ways to present and compare the sizes of groups for a single variable. Instead of a pie chart, try a bar chart for showing responses to "Should I make a pie chart?" (Figure 3.9).

Nancy suggested that if a pie chart is truly the most appropriate way to communicate data (it isn't), or if you have been directed to make a pie chart, there is guidance on creating pie charts on many websites. As an alternative, Nancy told Leslie they would review how to make a waffle chart as a better way to show parts of a whole. Kiara suggested that they stick with the bar charts now that they have one example. Leslie hoped they were lemon bars because all this talk of pies and waffles has really made her hungry for something sweet.

FIGURE 3.9 Should I make a pie chart?



3.4.2 BAR CHARTS

One thing that might be useful to know in better understanding guns in the United States is the rate of gun use. Although publicly available data on gun use are rare because of the Dickey Amendment, Nancy knew of a few persistent researchers who had found ways to collect gun-related data.

3.1 Kiara's reproducibility resource: Bringing data in directly from the Internet

3.1.1 Excel data

Kiara explained that including the code to download data directly from the Internet improves reproducibility because it ensures that anyone reproducing the work is working with the same exact data set. The FBI data used in this meeting (saved as **fbi_deaths_2016_ch3.csv** at **edge.sagepub.com/harris1e**) were downloaded directly from the FBI's Uniform Crime Reporting database. To download this data set directly to create the **fbi_deaths_2016_ch3.csv** data set, follow these instructions.

The first thing to notice before importing a data set directly from the Internet is the format of the file. This file is saved with the file extension ".xls," which indicates that it is an Excel file. Excel files with the .xls extension cannot be imported using `read.csv()` because they are not csv files. Instead, these files can be imported using the **`readxl`** package. The **`readxl`** package does not read things directly from the Internet, so another package will have to be used to import the data first before they can be read by **`readxl`**. The **`httr`** package has the `GET()` function, which is useful for getting data directly from an online location (URL) and temporarily storing it. Once the data are temporarily stored in a local location, often the `tempfile`, `read_excel()` from the **`readxl`** package can be used to read in the data from the Excel file.

Kiara wrote out the exact instructions for downloading the data directly from their online source. She included lots of comments for each step.

```
# install and then load the readxl
# and httr packages
library(package = "readxl")
library(package = "httr")

# create a variable that contains the web
# URL for the data set
url1 <- "https://ucr.fbi.gov/crime-in-the-u.s/2016/crime-in-the-
u.s.-2016/tables/expanded-homicide-data-table-4.xls/output.xls"

# use the GET function in httr to put the URL
# in a temporary file and specify the .xls file extension
```

```
# temporary file is named tf in this example
GET(url = url1, write_disk(tf <- tempfile(fileext = ".xls")))

# use the read_excel function in readxl
# to get the temporary file tf
# skip the first 3 rows of the file because they
# are header rows and do not contain data
# specify the number of observations using n_max
fbi.deaths <- data.frame(read_excel(path = tf, sheet = 1, skip = 3,
n_max = 18))
# option to save the file to a folder called "data"
write.csv(x = fbi.deaths, file = "[data folder location]/data/fbi_
deaths_2016_ch3.csv", row.names = FALSE)
```

3.1.2 NHANES data

The data files for NHANES are available on the CDC website. Kiara noticed that each part of the survey is saved in a separate file in the SAS Transport or xpt format. Luckily, she recently learned about an R package called **RNHANES** that can be used to download demographic data and other NHANES data into a data frame for use in R. This package includes all the steps needed to download the data directly given the `file_name` and the `year` of the data of interest. These two pieces of information can be found on the NHANES website.

To download the NHANES data for the examples in this chapter, install the **RNHANES** package and run the following code:

```
# open package
library(package = "RNHANES")

# download audiology data (AUQ_G)
# with demographics
nhanes.2012 <- nhanes_load_data(file_name = "AUQ_G", year = "2011-2012",
                                demographics = TRUE)

# option to save the data to a "data" folder
write.csv(x = nhanes.2012, file = "[data folder location]/data/
nhanes_2011_2012_ch3.csv ",
row.names = FALSE)

# examine gun use variable (AUQ300)
summary(object = nhanes.2012$AUQ300)
```

(Continued)

(Continued)

To use the **RNHANES** package to open a different NHANES data set, find the data files available for download on the NHANES website and note the name and year of the data file of interest. Use the code provided, but change the `file_name = "AUQ_G"` to include the name of the file of interest and change the `year = "2011-2012"` to the year(s) for the data of interest. Kiara noted that the `nhanes_load_data()` process takes a few minutes on her laptop.

One example Nancy was aware of is a set of questions in the nationally representative ***National Health and Nutrition Examination Survey***, or NHANES. Several administrations of the NHANES survey asked about gun use in the Audiology section concerned with how loud noise may influence hearing loss. The most recent year of NHANES data available with a gun use question was 2011–2012 (https://wwwn.cdc.gov/nchs/nhanes/Search/DataPage.aspx?Component=Questionnaire&CycleBeginYear=2011). One of the cool things about NHANES data is that an R package called **RNHANES** allows direct access to NHANES data from R, which is great for reproducibility. Kiara used **RNHANES** to download the 2011–2012 data (Box 3.1) and saved it for the R-Team as a csv file. Leslie used `read.csv()` to import the data and noticed that it had 82 variables, which is a lot for the `summary()` function. She checked that the import worked by using `head()` instead.

```
# import the data
nhanes.2012 <- read.csv(file = "[data folder location]/data/nhanes_
2011_2012_ch3.csv")

# check the import
head(x = nhanes.2012)
##    SEQN      cycle SDDSRVYR RIDSTATR RIAGENDR RIDAGEYR RIDAGEMN RIDRETH1
## 1 62161 2011-2012        7        2        1       22       NA        3
## 2 62162 2011-2012        7        2        2        3       NA        1
## 3 62163 2011-2012        7        2        1       14       NA        5
## 4 62164 2011-2012        7        2        2       44       NA        3
## 5 62165 2011-2012        7        2        2       14       NA        4
## 6 62166 2011-2012        7        2        1        9       NA        3
##   RIDRETH3 RIDEXMON RIDEXAGY RIDEXAGM DMQMILIZ DMQADFC DMDBORN4 DMDCITZN
## 1        3        2       NA       NA        2      NA        1        1
## 2        1        1        3       41       NA      NA        1        1
## 3        6        2       14      177       NA      NA        1        1
## 4        3        1       NA       NA        1       2        1        1
## 5        4        2       14      179       NA      NA        1        1
## 6        3        2       10      120       NA      NA        1        1
##   DMDYRSUS DMDEDUC3 DMDEDUC2 DMDMARTL RIDEXPRG SIALANG SIAPROXY SIAINTRP
## 1       NA       NA        3        5       NA       1        1        2
## 2       NA       NA       NA       NA       NA       1        1        2
## 3       NA        8       NA       NA       NA       1        1        2
```

```
## 4       NA      NA         4        1        2       1       2         2
## 5       NA       7        NA       NA       NA       1       1         2
## 6       NA       3        NA       NA       NA       1       1         2
##   FIALANG FIAPROXY FIAINTRP MIALANG MIAPROXY MIAINTRP AIALANGA  WTINT2YR
## 1       1        2        2       1        2        2         1 102641.406
## 2       1        2        2      NA       NA       NA        NA  15457.737
## 3       1        2        2       1        2        2         1   7397.685
## 4       1        2        2      NA       NA       NA        NA 127351.373
## 5       1        2        2       1        2        2         1  12209.745
## 6       1        2        2       1        2        2        NA  60593.637
##    WTMEC2YR SDMVPSU SDMVSTRA INDHHIN2 INDFMIN2 INDFMPIR DMDHHSIZ DMDFMSIZ
## 1 104236.583      1        91       14       14     3.15         5         5
## 2  16116.354      3        92        4        4     0.60         6         6
## 3   7869.485      3        90       15       15     4.07         5         5
## 4 127965.226      1        94        8        8     1.67         5         5
## 5  13384.042      2        90        4        4     0.57         5         5
## 6  64068.123      1        91       77       77       NA         6         6
##   DMDHHSZA DMDHHSZB DMDHHSZE DMDHRGND DMDHRAGE DMDHRBR4 DMDHREDU DMDHRMAR
## 1        0        1        0        2       50        1        5        1
## 2        2        2        0        2       24        1        3        6
## 3        0        2        1        1       42        1        5        1
## 4        1        2        0        1       52        1        4        1
## 5        1        2        0        2       33        2        2       77
## 6        0        4        0        1       44        1        5        1
##   DMDHSEDU AUQ054 AUQ060 AUQ070 AUQ080 AUQ090 AUQ100 AUQ110 AUQ136 AUQ138
## 1        5      2      1     NA     NA     NA      5      5      1      1
## 2       NA      1     NA     NA     NA     NA     NA     NA     NA     NA
## 3        4      2     NA     NA     NA     NA     NA     NA     NA     NA
## 4        4      1     NA     NA     NA     NA      4      5      2      2
## 5       NA      2     NA     NA     NA     NA     NA     NA     NA     NA
## 6        5      1     NA     NA     NA     NA     NA     NA     NA     NA
##   AUQ144 AUQ146 AUD148 AUQ152 AUQ154 AUQ191 AUQ250 AUQ255 AUQ260 AUQ270
## 1      4      2     NA     NA      2      2     NA     NA     NA     NA
## 2     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 3     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 4      4      2     NA     NA      2      1      5      1      2      1
## 5     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 6     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
##   AUQ280 AUQ300 AUQ310 AUQ320 AUQ330 AUQ340 AUQ350 AUQ360 AUQ370 AUQ380
## 1     NA      2     NA     NA      2     NA     NA     NA      2      1
```

```
## 2     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 3     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 4      1      1      2      1      2     NA     NA     NA      2      6
## 5     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 6     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
##   file_name begin_year end_year
## 1     AUQ_G       2011     2012
## 2     AUQ_G       2011     2012
## 3     AUQ_G       2011     2012
## 4     AUQ_G       2011     2012
## 5     AUQ_G       2011     2012
## 6     AUQ_G       2011     2012
```

Leslie looked through the audiology codebook and found the gun use question `AUQ300`, which asked participants, "Have you ever used firearms for any reason?" Before they started analyses, Leslie used `summary()` to check the `AUQ300` variable.

```
# check the data
summary(object = nhanes.2012$AUQ300)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   2.000   1.656   2.000   7.000    4689
```

The gun use data imported as a numeric variable type. The audiology data codebook shows five possible values for `AUQ300`:

- 1 = Yes
- 2 = No
- 7 = Refused
- 9 = Don't know
- . = Missing

Using her skills from the earlier chapters, Leslie recoded the variable to a factor with these five levels and a more logical variable name.

```
# open tidyverse
library(package = "tidyverse")

# recode gun use variable
nhanes.2012.clean <- nhanes.2012 %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No',
```

```
                                  `7` = 'Refused',
                                  `9` = 'Don\'t know'))

# check the recoding
summary(object = nhanes.2012.clean$AUQ300)

##      Yes       No Refused NA's
##     1613     3061        1 4689
```

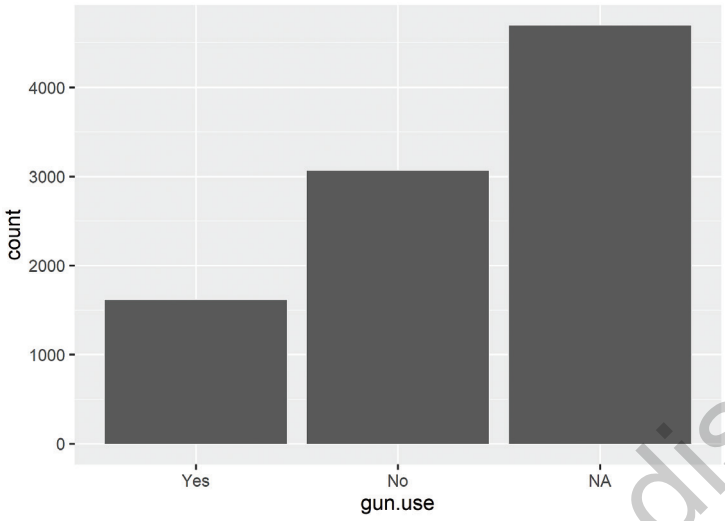
Kiara noted there was a single `Refused` response to the gun use question and no `Don't know` responses. These categories are not likely to be useful for visualizing or analyzing this variable. Leslie recoded them as `NA` for missing. She also thought it would be easier to work with the gun use variable if the variable name were something more intuitive so she could remember it while coding. She almost asked Nancy if there was a function for renaming something, but she wanted to keep coding, so she looked up how to change variable names in R and found `rename()`. The `rename()` function works with the pipe structure. Filling the new name as the first argument of `rename()` and the old name as the second argument, Leslie renamed `AUQ300` to `gun.use`.

```
# recode gun use variable
nhanes.2012.clean <- nhanes.2012 %>%
  mutate(AUQ300 = na_if(x = AUQ300, y = 7)) %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No')) %>%
rename(gun.use = AUQ300)

# check recoding
summary(object = nhanes.2012.clean$gun.use)
## Yes  No   NA's
## 1613 3061 4690
```

Kiara started by reminding Leslie about the `ggplot()` function from the **`ggplot2`** package to create a bar chart of gun use. The **`ggplot2`** package uses the *grammar of graphics*, which is what the *gg* stands for. Kiara reminded Leslie that graphs built with `ggplot()` are built in layers. The first layer starts with `ggplot()` and `aes()` or *aesthetics*, which contains the basic information about which variables are included in the graph and whether each variable should be represented on the *x*-axis, the *y*-axis, as a color, as a line type, or something else. The next layer typically gives the graph type—or graph geometry, in the grammar of graphics language—and starts with `geom_` followed by one of the available types. In this case, Leslie was looking for a bar chart, so `geom_bar()` was the geometry for this graph. Leslie remembered that `geom_bar()` is a layer of the plot, so it is added with a `+` instead of a `%>%`. She wrote the code for Figure 3.10, highlighted it, and held her breath while she clicked Run.

```
# plot gun use in US 2011-2012 (Figure 3.10)
nhanes.2012.clean %>%
  ggplot(aes(x = gun.use)) +
  geom_bar()
```

FIGURE 3.10 Gun use by NHANES 2011–2012 participants



Leslie was again surprised at how quickly she could make a graph in R. While it needed a lot of work on the format, this was a good start. Leslie made a list of the things she wanted to change about this initial graph:

- Remove the `NA` bar from the graph.
- Change the labels on the axes to provide more information.
- Use a theme that does not use so much ink.

Leslie also thought she could

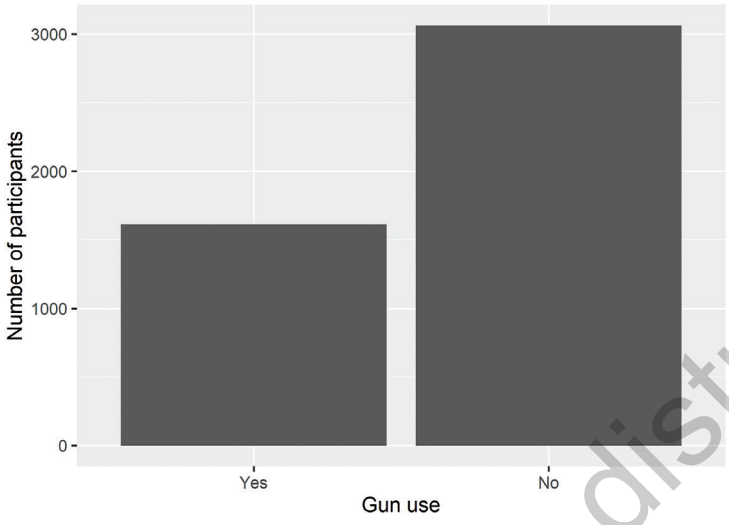
- Make each bar a different color.
- Show percentages instead of counts on the *y*-axis.

Kiara knew all of those things were possible, although adding the percentage on the *y*-axis could be tricky. She started with removing the `NA` bar using `drop_na()` for the `gun.use` variable before building the plot with `ggplot()`. Then she added a custom labels layer with `labs(x = , y = )` to add better labels on the two axes (Figure 3.11).

```
# omit NA category from gun.use plot and add axis labels (Figure 3.11)
nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use)) +
  geom_bar() +
  labs(x = "Gun use", y = "Number of participants")
```

Before Kiara helped her add percentages to the *y*-axis, Leslie worked on the color. She remembered to add `fill = gun.use` to the aesthetics in `aes()`, and Kiara explained more about the `aes()` parentheses. She said that changing the way the graph looks based on the data should happen within `aes()`. For example, Leslie wanted the colors of the bars to be different depending on the gun use category, which comes from the

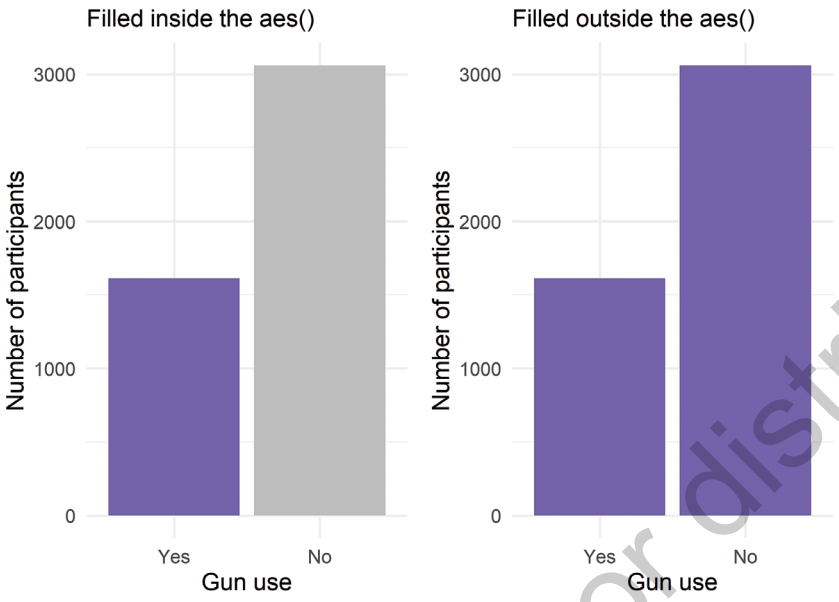
FIGURE 3.11 Gun use among 2011–2012 NHANES participants



`gun.use` variable in the data set. Therefore, `fill =` should be within `aes()` like this: `geom_bar(aes(fill = gun.use))`. If Leslie wanted to make the bars a color without linking it to the categories of `gun.use`, Kiara explained, since this is *not* based on the data, Leslie would put the `fill =` *outside of* the `aes()` parentheses like this: `geom_bar(fill = "purple")`. Kiara wrote the code both ways to show Leslie and used the `grid.arrange()` function from the **`gridExtra`** package to show the plots side by side in Figure 3.12.

```
# fill bars inside aes
fill.aes <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use)) +
  geom_bar(aes(fill = gun.use)) +
  labs(x = "Gun use", y = "Number of participants",
       subtitle = "Filled inside the aes()") +
  scale_fill_manual(values = c("#7463AC", "gray"), guide = FALSE) +
  theme_minimal()

# fill bars outside aes
fill.outside <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use)) +
  geom_bar(fill = "#7463AC") +
  labs(x = "Gun use", y = "Number of participants",
       subtitle = "Filled outside the aes()") +
  theme_minimal()

# arrange the two plots side by side (Figure 3.12)
gridExtra::grid.arrange(fill.aes, fill.outside, ncol = 2)
```

FIGURE 3.12 Gun use among 2011–2012 NHANES participants



Leslie noticed that there were `aes()` options for both the `ggplot()` layer and the `geom_bar()` layer and asked what the difference was. Kiara thought that was a great question and explained that some aesthetics can be set in either place, like the color of the bars, for example. She made a quick change to the code and showed Leslie Figure 3.13 with the color set in the `ggplot()` layer `aes()` and in the `geom_bar()` layer `aes()`.

```
# fill inside aes for ggplot layer
fill.aes <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use, fill = gun.use)) +
  geom_bar() +
  labs(x = "Gun use", y = "Number of participants", subtitle = "Filled in
ggplot layer") +
  scale_fill_manual(values = c("#7463AC", "gray"), guide = FALSE) +
  theme_minimal()

# fill inside aes for geom_bar layer
fill.outside <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use)) +
  geom_bar(aes(fill = gun.use)) +
  labs(x = "Gun use", y = "Number of participants",
       subtitle = "Filled in geom_bar layer") +
```

```
  scale_fill_manual(values = c("#7463AC", "gray"), guide = FALSE) +
  theme_minimal()

# arrange the two plots side by side (Figure 3.13)
gridExtra::grid.arrange(fill.aes, fill.outside, ncol = 2)
```

Some aesthetics are specific to the type of graph geometry. For example, there is an aesthetic called `linetype =` that can make lines appear in different patterns, such as dotted. This is not an available aesthetic for graphs that have no lines in them. Kiara told Leslie that the aesthetics being relevant by geom is one reason why she prefers to put aesthetics in the `geom_` instead of in the `ggplot()` layer.

Kiara advised Leslie to save the URL for the Data Visualization Cheat Sheet from RStudio (https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf), which had the aesthetics available for the different geometries. Leslie opened the cheat sheet and saw many different types of graphs. Kiara pointed out that below the name of each type of `geom_` was a list, and these were the aesthetics available for that `geom_`. She pointed out the `geom_bar()` entry, which listed the following available aesthetics: x, alpha, color, fill, linetype, size, and weight (Figure 3.14).

FIGURE 3.13 Gun use among 2011–2012 NHANES participants

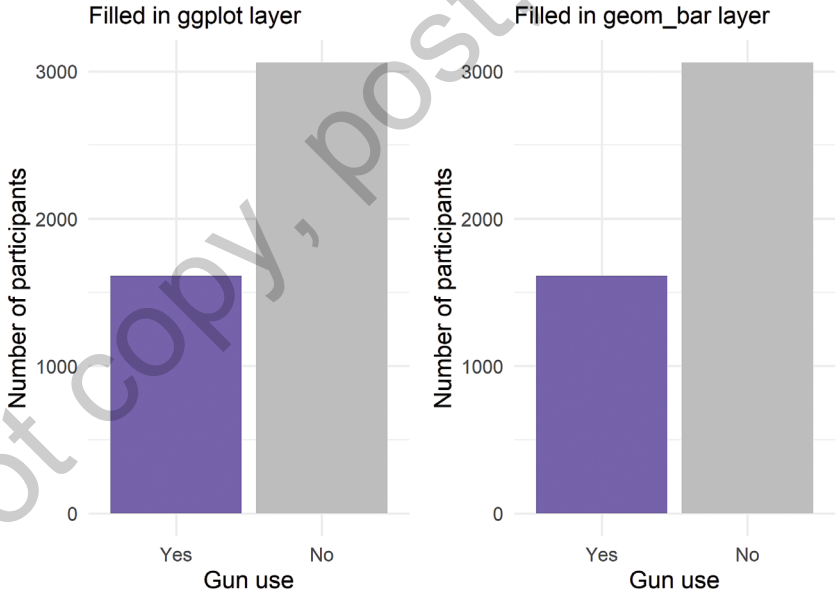


FIGURE 3.14 Entry in RStudio Data Visualization Cheet Sheet for the aesthetics available for geom_bar()

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

Leslie saved the URL. She knew this would be useful to have in the future.

They looked back at the graph and remembered that they had one more change to make. The change from the number of people to the percentage of people along the *y*-axis is tricky, although Leslie had seen it done before when Kiara wrote the code for her to create Figure 1.17. To get the *y*-axis to show percentage rather than count, the *y*-axis uses *special variables* with double periods around them. Special variables are statistics computed from a data set; the special variable *count* counts the number of observations. After reviewing Section 1.10.3, Kiara added the special variables to the aesthetics using `..count..` to represent the frequency of a category and `sum(..count..)` to represent the sum of all the frequencies. She multiplied by 100 to get a percent in Figure 3.15.

```
# formatted bar chart of gun use (Figure 3.15)
nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use,
             y = 100*(..count..)/sum(..count..))) +
  geom_bar(aes(fill = gun.use)) +
  labs(x = "Gun use", y = "Percent of participants") +
  scale_fill_manual(values = c("#7463AC", "gray"), guide = FALSE) +
  theme_minimal()
```

Leslie wondered about why the *y*-axis of the graph only went to 60%. She had heard that people sometimes limit the range of the *y*-axis in order to make a difference between groups or a change over time look bigger (or smaller) than it actually is. Kiara showed her how to make the *y*-axis go to 100% by creating a `ylim()` layer in the `ggplot()`. The `ylim()` layer takes the lowest value for the *y*-axis and the highest value for the *y*-axis, separated by a comma. For a *y*-axis that goes from 0 to 100, it looks like this: `ylim(0, 100)` (Figure 3.16).

FIGURE 3.15 Gun use among 2011–2012 NHANES participants

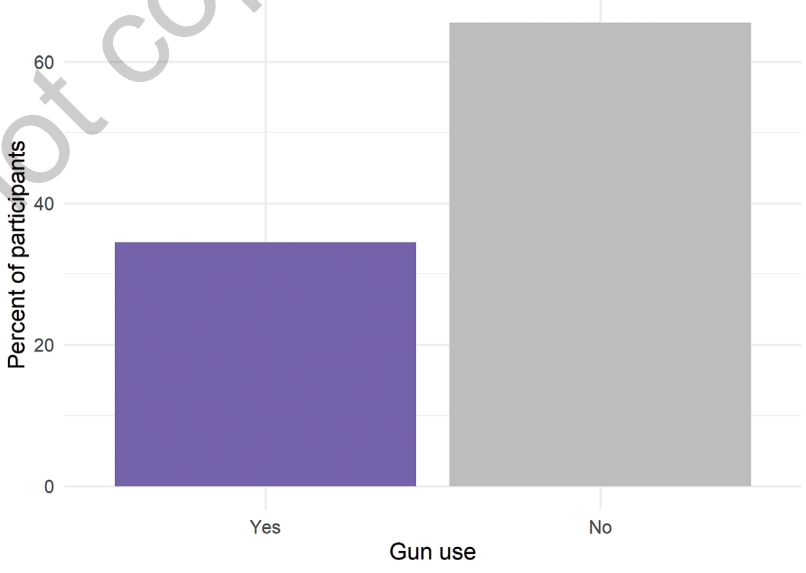
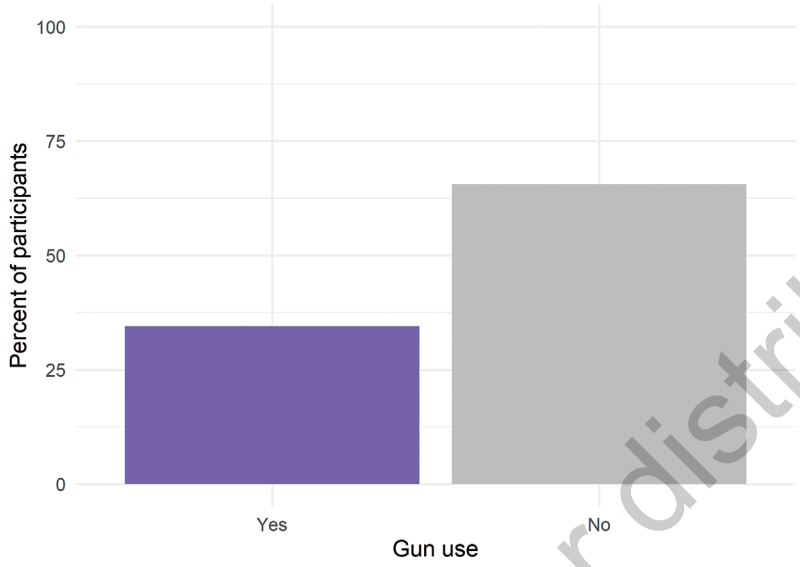


FIGURE 3.16 Gun use among 2011–2012 NHANES participants




```
# formatted bar chart of gun use (Figure 3.16)
nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use,
             y = 100*(..count..)/sum(..count..))) +
  geom_bar(aes(fill = gun.use)) +
  labs(x = "Gun use", y = "Percent of participants") +
  scale_fill_manual(values = c("#7463AC", "gray"), guide = FALSE) +
  theme_minimal() +
  ylim(0, 100)
```

The expanded *y*-axis did change the look of the graph, but the difference still seems large between the groups. The bar for the `No` group is about twice as large as the bar for the `Yes` group.

3.4.3 WAFFLE CHARTS

Waffle charts are similar to pie charts in showing the parts of a whole. However, the structure of a waffle chart visually shows the relative contributions of categories to the whole waffle more clearly. Nancy explained that, while pie may arguably be more delicious than waffles in real life, for reporting parts of a whole, waffles > pie.

Kiara suggested making a graph of the `AUQ310` variable from the NHANES data set since it has more than two categories and so may be more interesting to view. The `AUQ310` variable is the response to the question "How many total rounds have you ever fired?" for survey participants who reported that they had used a gun. The audiology data codebook shows eight categories for `AUQ310`:

- 1 = 1 to less than 100 rounds
- 2 = 100 to less than 1,000 rounds

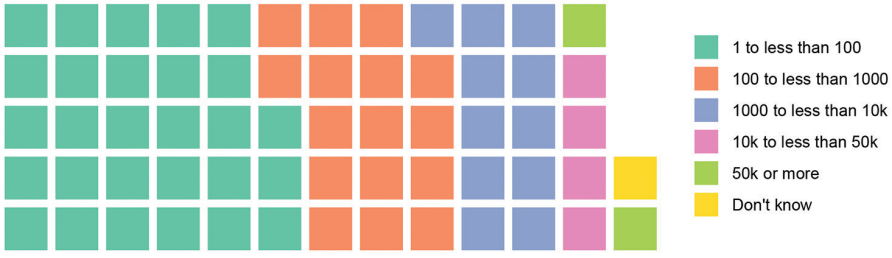
- 3 = 1,000 to less than 10,000 rounds
- 4 = 10,000 to less than 50,000 rounds
- 5 = 50,000 rounds or more
- 7 = Refused
- 9 = Don't know
- . = Missing

Leslie added on to the existing data management code to add labels, change `AUQ310` to a factor, and rename `AUQ310` to `fired`, which was easier to remember and type.

```
# recode gun use variable
nhanes.2012.clean <- nhanes.2012 %>%
  mutate(AUQ300 = na_if(x = AUQ300, y = 7)) %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No')) %>%
  rename(gun.use = AUQ300) %>%
  mutate(AUQ310 = recode_factor(.x = AUQ310,
                                `1` = "1 to less than 100",
                                `2` = "100 to less than 1000",
                                `3` = "1000 to less than 10k",
                                `4` = "10k to less than 50k",
                                `5` = "50k or more",
                                `7` = "Refused",
                                `9` = "Don't know")) %>%
  rename(fired = AUQ310)

#check recoding
summary(object = nhanes.2012.clean$fired)
##    1 to less than 100 100 to less than 1000 1000 to less than 10k
##                   701                   423                   291
##  10k to less than 50k           50k or more            Don't know
##                   106                    66                    26
##                  NA's
##                  7751
```

Now it was time to make the graph. Unfortunately, Nancy explained, there is no built-in `geom_waffle()` option for `ggplot()`, so they would use the **`waffle`** package instead. Before they started graphing, Leslie reviewed the package documentation (https://github.com/hrbrmstr/waffle).

FIGURE 3.17 Rounds shot by 1,613 gun users, NHANES 2011–2012



Note: One square represents 25 people.

The first argument for the `waffle()` is a table or vector of ***summary statistics*** used to make the waffle squares, sort of like the table used for the B index from their last meeting. That is, the data used by `waffle()` are not the ***individual-level*** data with one observation per row. Instead, the first argument is a frequency table or a vector of frequencies that shows how many observations are in each category. The `table()` code works well for use with `waffle()`.

By default, `waffle()` makes one square per observation. There are more than 1,000 observations in the `nhanes.2012` data frame, which seems like a lot of squares! Nancy suggested making each square represent 25 observations. Finally, the last argument for `waffle()` is the number of rows of squares. Leslie suggested they start with five rows and see what happens. Leslie made a table of the `fired` variable and named the table `rounds`. She then entered the `rounds` table into the `waffle()` function, divided by 25, and added `rows = 5` (Figure 3.17).

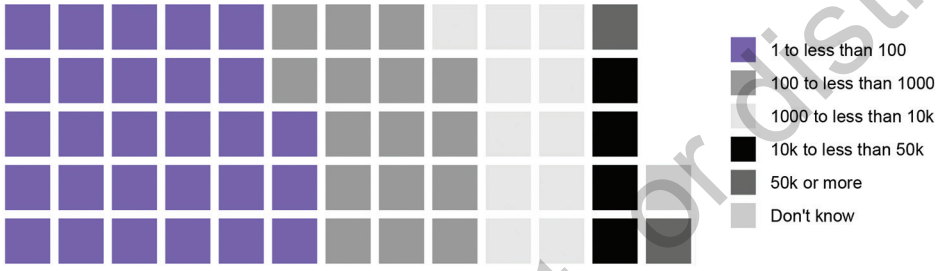
```
# open the waffle library
library(package = "waffle")

# make a table of rounds fired data
rounds <- table(nhanes.2012.clean$fired)

# each square is 25 people (Figure 3.17)
# 5 rows of squares
waffle(parts = rounds / 25, rows = 5)
```

It was clear from this waffle that the `1 to less than 100` category is the biggest.

Nancy suggested that color could be used to make a point about the size of a certain category. For example, if the goal was to examine people who own firearms but are less experienced in using firearms, they could use a bright color to highlight the group that had fired fewer rounds. Color is added by using a `colors =` option and listing the colors in a vector. Leslie added the RGB code for the purple color she had been using and then found some different shades of gray to include. Nancy showed her how to make sure the colors were assigned to the right parts of the waffle by entering the category labels for each color. If she did not enter the category labels, the list of colors would be assigned to the categories in alphabetical order, which could be tricky. After entering the colors, they reviewed Figure 3.18.

```
# change the colors (Figure 3.18)
waffle(parts = rounds / 25, rows = 5,
     colors = c("1 to less than 100" = "#7463AC",
                "100 to less than 1000" = "gray60",
                "1000 to less than 10k" = "gray90",
                "10k to less than 50k" = "black",
                "50k or more" = "gray40",
                "Don't know" = "gray80"))
```

FIGURE 3.18 Rounds shot by 1,613 gun users, NHANES 2011–2012



Note: One square represents 25 people.

The bright color for the `1 to less than 100` category made this category stand out. The two recommended graphs for displaying a single categorical or factor-type variable are bar charts and waffle charts. The bar chart is useful for showing relative group sizes. The waffle chart is an alternative to a pie chart and is best when demonstrating parts of a whole. Pie charts are available in R but are not recommended because they tend to be less clear for comparing group sizes.

3.4.4 ACHIEVEMENT 1: CHECK YOUR UNDERSTANDING

Create a bar chart for the gender variable (`RIAGENDR`) from the NHANES 2012 data set. Examine the codebook for coding hints and clean up the data first.

3.5 Achievement 2: Choosing and creating graphs for a single continuous variable

After making it through the options for graphing a single categorical variable, Leslie wanted to learn which graphs were appropriate for graphing a single continuous variable. Three commonly used options are histograms, ***density plots***, and ***boxplots***. Histograms and density plots are very similar to each other and show the overall shape of the data. These two types of graphs are especially useful in determining whether or not a variable has a ***normal distribution*** (see Figure 2.10). Boxplots show the central tendency and spread of the data, which is another way to determine whether a variable is normally distributed or skewed. Kiara added that ***violin plots*** are also useful when looking at a continuous variable and are like a combination of boxplots and density plots. Violin plots are commonly used to examine the distribution of a continuous variable for different levels (or groups) of a factor (or categorical) variable.

Kiara suggested focusing on histograms, density plots, and boxplots for now, and returning to violin plots when they are looking at graphs with two variables.

Kiara noted that the gun research data included a measure of the amount of research funding devoted to examining the different causes of death. Funding falls along a continuum and would be best examined as a continuous variable using a histogram, density plot, or boxplot. She helped Leslie with the coding needed to create each type of graph.

3.5.1 HISTOGRAMS

Kiara explained that histograms can be developed with **ggplot2**. She showed Leslie the data set they had received from the authors of the *JAMA* article (Stark & Shah, 2017) and Leslie imported it.

```
# bring in the data
research.funding <- read.csv(file = "[data folder location]/data/gun_
publications_funds_2004_2015_ch3.csv")

# check out the data
summary(object = research.funding)
##                Cause.of.Death Mortality.Rate.per.100.000.Population
##  Alzheimer disease    : 1     Min.   :  0.590
##  Anemia               : 1     1st Qu.:  1.775
##  Asphyxia             : 1     Median :  7.765
##  Aspiration           : 1     Mean   : 22.419
##  Atherosclerosis      : 1     3rd Qu.: 14.812
##  Biliary tract disease: 1     Max.   :201.540
##  (Other)              :24
##   Publications        Funding          Predicted.Publications
##  Min.   :   1034 Min.   :3.475e+06   8,759  : 2
##  1st Qu.:  12550 1st Qu.:3.580e+08   10,586 : 1
##  Median :  39498 Median :1.660e+09   11,554 : 1
##  Mean   :  93914 Mean   :4.137e+09   15,132 : 1
##  3rd Qu.:  54064 3rd Qu.:4.830e+09   16,247 : 1
##  Max.   :1078144 Max.   :2.060e+10   16,751 : 1
##                                      (Other):23
##  Publications..Studentized.Residuals.       Predicted.Funding
##  Min.   :-2.630                          $264,685,579   : 2
##  1st Qu.:-0.355                          $1,073,615,675 : 1
##  Median : 0.125                          $1,220,029,999 : 1
##  Mean   :-0.010                          $1,242,904,513 : 1
##  3rd Qu.: 0.745                          $1,407,700,121 : 1
##  Max.   : 1.460                          $1,417,564,256 : 1
##                                          (Other)        :23
##  Funding..Studentized.Residuals.
##  Min.   :-3.71000
```

```
##  1st Qu.:-0.53250
##  Median : 0.33500
##  Mean   :-0.02467
##  3rd Qu.: 0.57750
##  Max.   : 1.92000
##
```

Kiara showed Leslie the geometry for a histogram, `geom_histogram()`, and Leslie used it to start with a very basic histogram (Figure 3.19).

```
# make a histogram of funding (Figure 3.19)
histo.funding <- research.funding %>%
 ggplot(aes(x = Funding)) +
 geom_histogram()
histo.funding
```

Figure 3.19 shows frequency on the *y*-axis and mortality rate on the *x*-axis. Leslie noticed that the *x*-axis is shown using scientific notation. While she was familiar with scientific notation, which is useful for printing large numbers in small spaces, she knew it is not well understood by most audiences (Box 3.2). Kiara suggested changing the axis to show numbers that can be more easily interpreted; this can be done in several ways. One strategy is to convert the numbers from dollars to billions of dollars by dividing the `Funding` variable by 1,000,000,000 within the `aes()` for the `ggplot()` (Figure 3.20).

FIGURE 3.19 Research funding in billions for the top 30 mortality causes in the United States

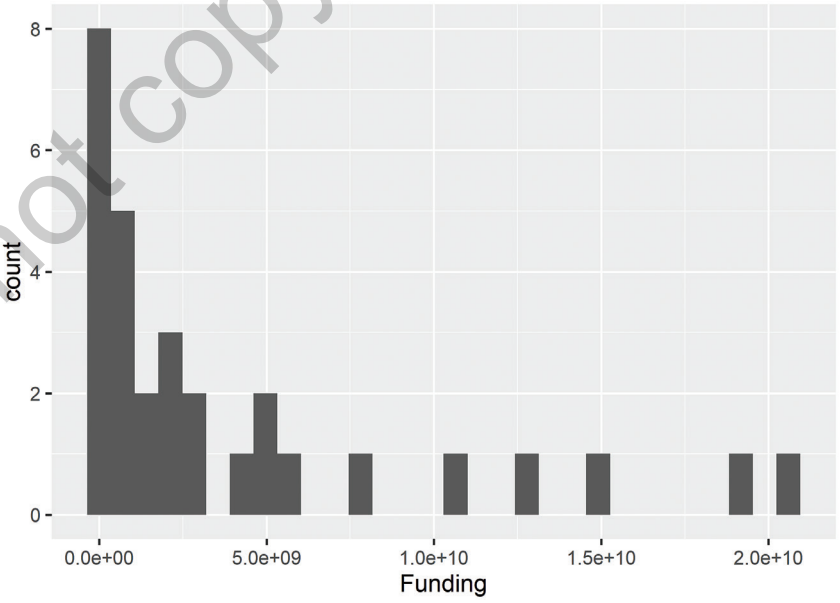
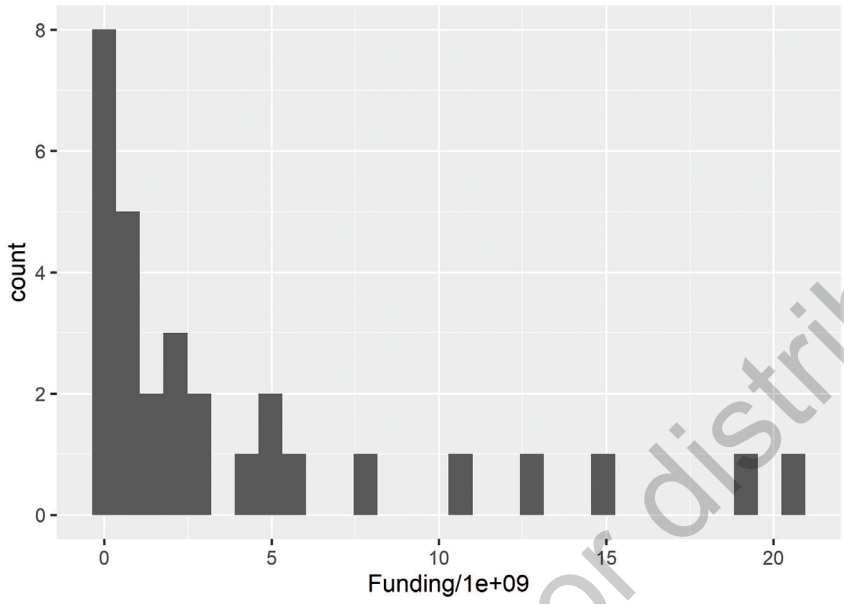


FIGURE 3.20 Research funding in billions for the top 30 mortality causes in the United States




```
# make a histogram of funding (Figure 3.20)
histo.funding <- research.funding %>%
 ggplot(aes(x = Funding/1000000000)) +
 geom_histogram()
histo.funding
```

Now the values on the *x*-axis are easier to understand. From the histogram, it appears that most mortality causes are funded at 0 to $5 billion annually. However, several causes receive more than $5 billion, up to more than $25 billion. The very large values on the right of the graph suggested to Leslie that the ***distribution*** of the funding data is right-skewed.

Kiara reminded Leslie that each of the bars shown in the histogram is called a bin and contains a certain proportion of the observations. To show more bins, which may help to clarify the shape of the distribution, specify how many bins to see by adding `bins =` to the `geom_histogram()` layer. Leslie tried 10 bins in Figure 3.21.

```
# make a histogram of funding (Figure 3.21)
# adjust the number of bins to 10
histo.funding <- research.funding %>%
 ggplot(aes(x = Funding/1000000000)) +
 geom_histogram(bins = 10)
histo.funding
```

3.2 Leslie's stats stuff: Scientific notation

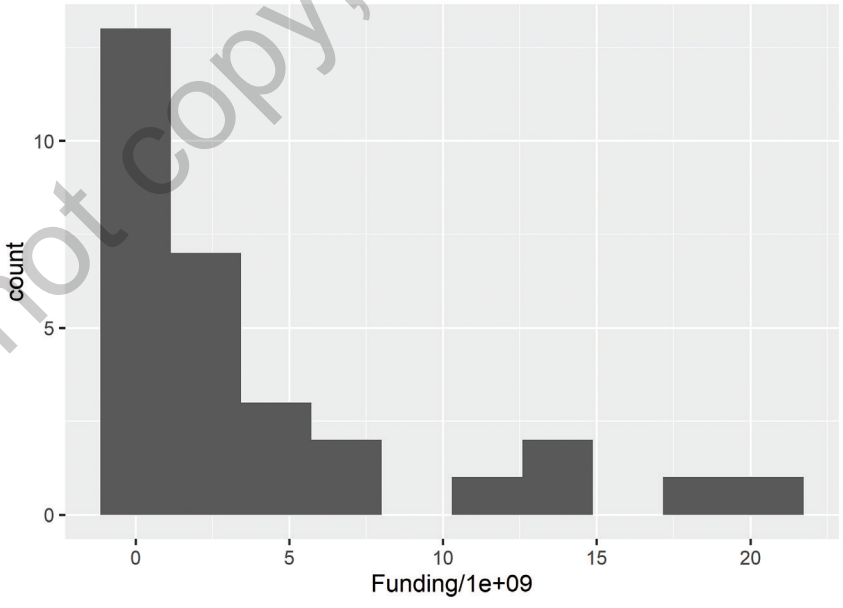
Scientific notation is used to display extremely large or extremely small numbers efficiently. For example, 250,000,000,000,000,000,000 is both difficult to read and difficult to use on a graph or in a table. Instead, the decimal point that is implied at the end of the number is moved to the left 20 places and the number becomes 2.5×10^{20} or 2.5 times 10 to the 20th power.

Likewise, .00000000000000000025 is difficult to read and display in a graph or a table. In this case, reporting the value in scientific notation would require moving the decimal point to the right 20 places. The result would be 2.5×10^{-20} or 2.5 times 10 to the –20th power.

While scientific notation is an efficient way of displaying extremely large or small values, it is not well understood. For this reason, it should be used only when there is not another option for displaying the information. For example, if the reported numbers could be divided by a million or a billion and then reported in millions or billions, this is a much better option.

When numbers are very large or very small, R will convert them to scientific notation. To turn off this option in R, type and run `options(scipen = 999)`. To turn it back on, type and run `options(scipen = 000)`.

FIGURE 3.21 Research funding for the top 30 mortality causes in the United States in 10-bin histogram



Leslie noticed that the range of the *y*-axis had changed when she changed the bins. Instead of the top value being 8, it was now 10. Nancy assured her that this was logical because the number of observations in each bar often changes when the bar is representing a different range of values. Nancy pointed to the first tall bar in Figure 3.20 and showed Leslie that the bar contains mortality causes with between 0 and maybe $0.5 billion in funding. Once `bins = 10` was added in Figure 3.21, the first tall bar represented mortality cases with between 0 and $1.25 billion in funding. With a wider range of funding values, there are more causes of mortality that fit into this first bar and therefore the bar is taller. Leslie thought she understood and noticed that there appear to be 8 causes of mortality with funding between 0 and $0.5 billion (Figure 3.20) but 13 causes of mortality with funding between 0 and $1.25 billion (Figure 3.21). Nancy confirmed that this is the case. Leslie was satisfied with this explanation and tried 50 bins next to see if it changed the shape (Figure 3.22).

```
# make a histogram of funding (Figure 3.22)
# adjust the number of bins to 50
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 50)
histo.funding
```

The 10-bin version looked best to Leslie, so she turned her attention to formatting, adding better titles for the axes in a `labs()` layer and making the graph printer-friendly to use less ink by adding a `theme_minimal()` layer (Figure 3.23).

FIGURE 3.22 Research funding for the top 30 mortality causes in the United States in 50-bin histogram

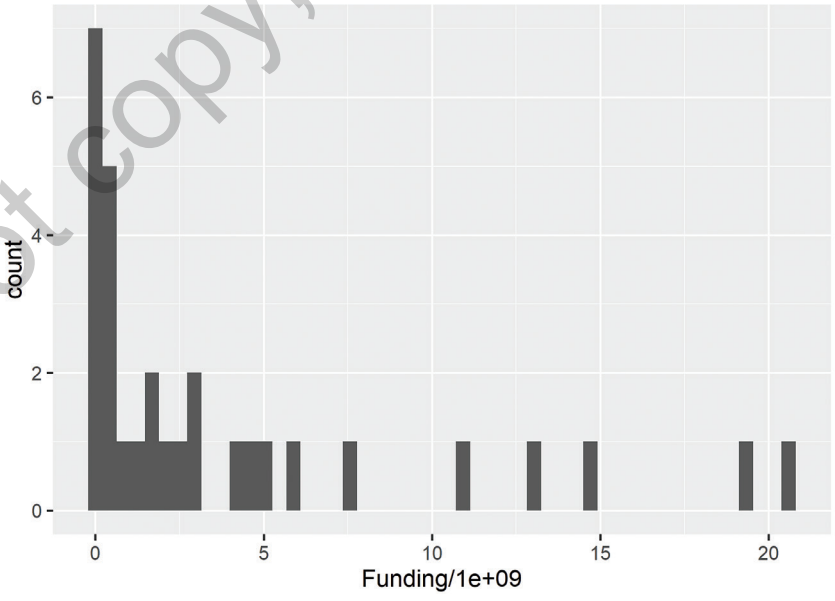
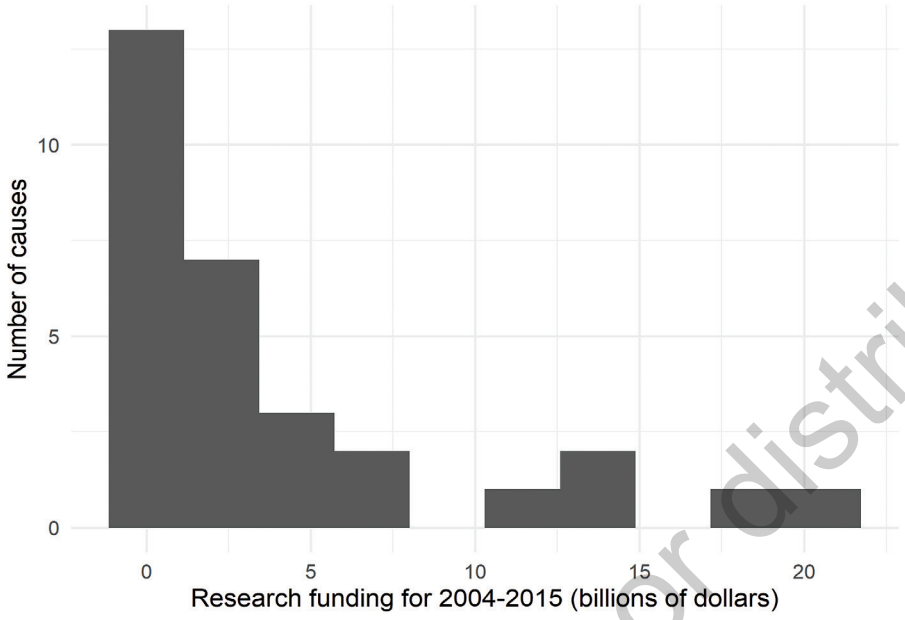


FIGURE 3.23 Research funding for the top 30 mortality causes in the United States




```
# make a histogram of funding (Figure 3.23)
# adjust the number of bins to 10
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 10) +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Number of causes") +
  theme_minimal()
histo.funding
```

This looked even better to Leslie, so she asked Kiara and Nancy if they had any suggestions for other formatting that might make the graph easier to interpret. Kiara said she liked to add thin borders around the bins and fill the bins with white, similar to the plots in the previous chapter (e.g., Figure 2.15). Leslie asked her how to do this, and Kiara said that `geom_histogram()` can take arguments for `fill =`, which takes a color to fill each bin, and `color =`, which takes a color for the border of each bin. Leslie decided that since she is adding a `color` and `fill` based on what she wants, and not based on the data set, she should add these arguments to `geom_histogram()` *without* putting them in `aes()` (Figure 3.24).

```
# make a histogram of funding (Figure 3.24)
# formatting options
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 10, fill = "white", color = "gray") +
```

```
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Number of causes") +
  theme_minimal()
histo.funding
```

The R-Team was happy with this final plot. Before moving on to density plots, they paused for a minute to discuss the shape of the distribution. Leslie noted that it is right-skewed and would therefore be best described using the median rather than the mean as they had discussed at the previous meeting (Section 2.6.2). She also thought that the IQR would probably be better than the range for reporting spread given how wide the range is (Section 2.6.4). Nancy and Kiara agreed with this assessment.

3.5.2 DENSITY PLOTS

A density plot is similar to a histogram but more fluid in appearance because it does not have the separate bins. Density plots can be created using `ggplot()` with a `geom_density()` layer. Leslie took the code from the histogram and replaced the `geom_histogram()` layer to see if that would work. Before she ran the code, Nancy stopped her to let her know that the *y*-axis is a different measure for this type of plot. Instead of frequency, it is the probability density, which is the probability of each value on the *x*-axis. The probability density is not very useful for interpreting what is happening at any given value of the variable on the *x*-axis, but it is useful in computing the percentage of values that are within a range along the *x*-axis. Leslie remembered seeing this on the *y*-axis of many of the histograms in their previous meeting and asked if it is used in histograms too. Nancy confirmed that probability density is the value of the *y*-axis for many different types of plots.

When she saw some confusion on Leslie's face, Nancy clarified that the area under the curve adds up to 100% of the data and the height of the curve is determined by the distribution of the data, which is scaled so that the area will be 100% (or 1). Leslie changed the `y =` option in the `labs()` layer to label the *y*-axis "Probability density" (Figure 3.25).

FIGURE 3.24 Research funding for the top 30 mortality causes in the United States

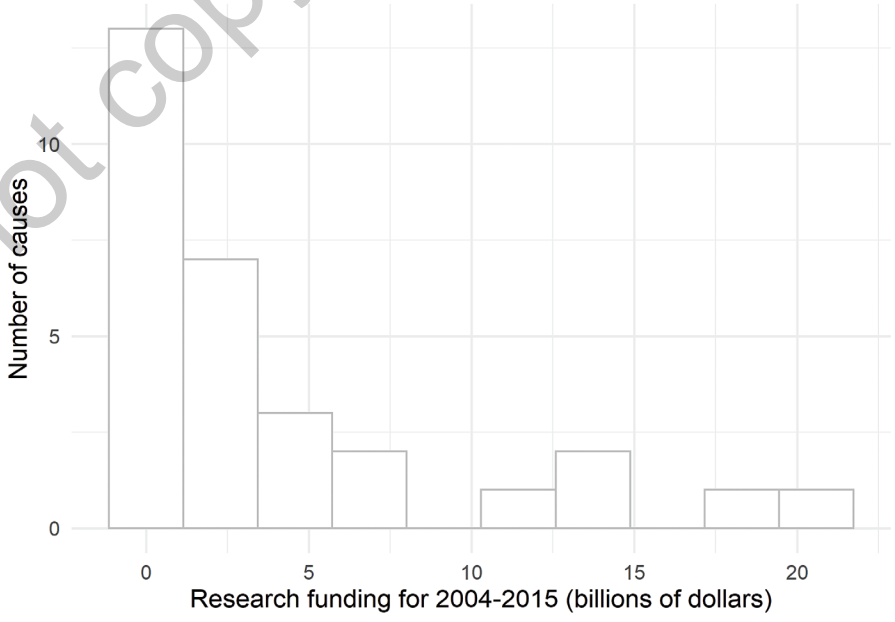
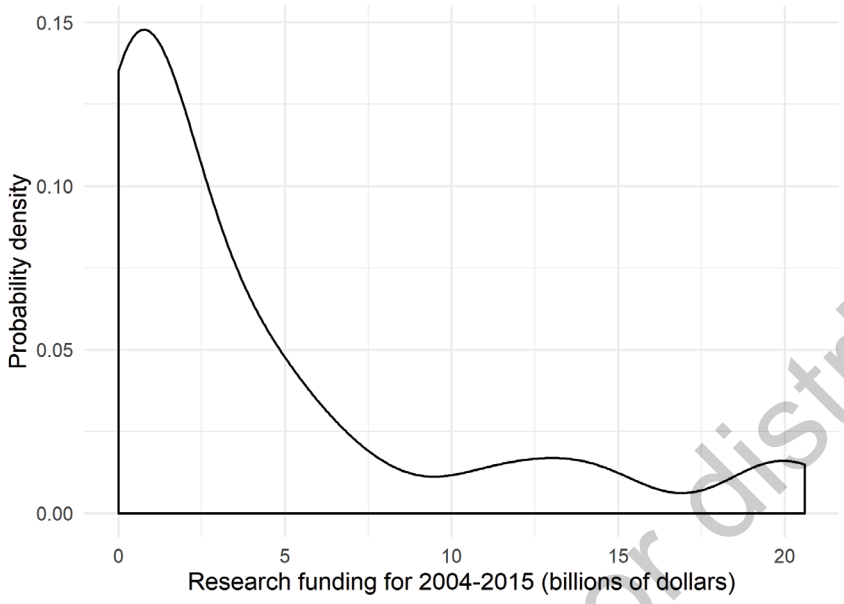


FIGURE 3.25 Research funding for the top 30 mortality causes in the United States




```
# density plot of research funding (Figure 3.25)
dens.funding <- research.funding %>%
   ggplot(aes(x = Funding/1000000000)) +
   geom_density() +
   labs(x = "Research funding for 2004-2015 (billions of dollars)",
        y = "Probability density") +
   theme_minimal()
dens.funding
```

Kiara tried to clarify some more. She told Leslie that the ***area under the curve*** in a density plot could be interpreted as the probability of a single observation or a range of observations. Probabilities are also useful for learning more about a population from a ***sample***, or a subgroup selected from the ***population***. She explained that they would discuss the use of density plots to demonstrate more about probability in their next meeting when they worked on probability concepts.

Leslie was not happy with the way the density plot looked. She added some color in order to be able to see the shape a little more. Nancy suggested trying a few values of `bw =` within the `geom_density()`, noting that `bw` usually takes much smaller values than `bins`. The `bw` stands for *bandwidth* in a density plot, which is similar to the bin width in a histogram. Leslie played with the bandwidth and some color in Figures 3.26 and 3.27.

```
# density plot of research funding (Figure 3.26)
# bw = .5
dens.funding <- research.funding %>%
```

```
  ggplot(aes(x = Funding/1000000000)) +
  geom_density(bw = .5, fill = "#7463AC") +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Probability density") +
  theme_minimal()
dens.funding
```

```
# density plot of research funding (Figure 3.27)
# bw = 1.5
dens.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_density(bw = 1.5, fill = "#7463AC") +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Probability density") +
  theme_minimal()
dens.funding
```

It appeared that the higher the value used as a bandwidth in `bw =`, the smoother the graph looks. Leslie thought the final version with the bandwidth of 1.5 looked good. Nancy agreed but wanted to add one word of caution on density plots before they moved on. While density plots are generally similar to histograms, they do have one feature that some data scientists suggest is misleading. Compare Figure 3.24 with Figure 3.27. These are both from the same data; Figure 3.24 shows gaps where there are no observations, while Figure 3.27 has the appearance of data continuing without gaps across the full range of values. For this reason, data scientists sometimes recommend histograms over density plots, especially for small data sets where gaps in the data are more likely (Wilke, 2019).

FIGURE 3.26 Research funding for the top 30 mortality causes in the United States

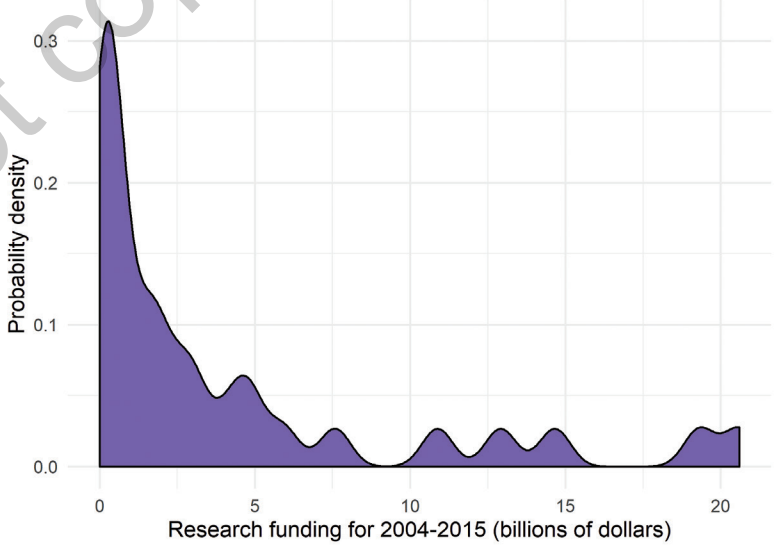
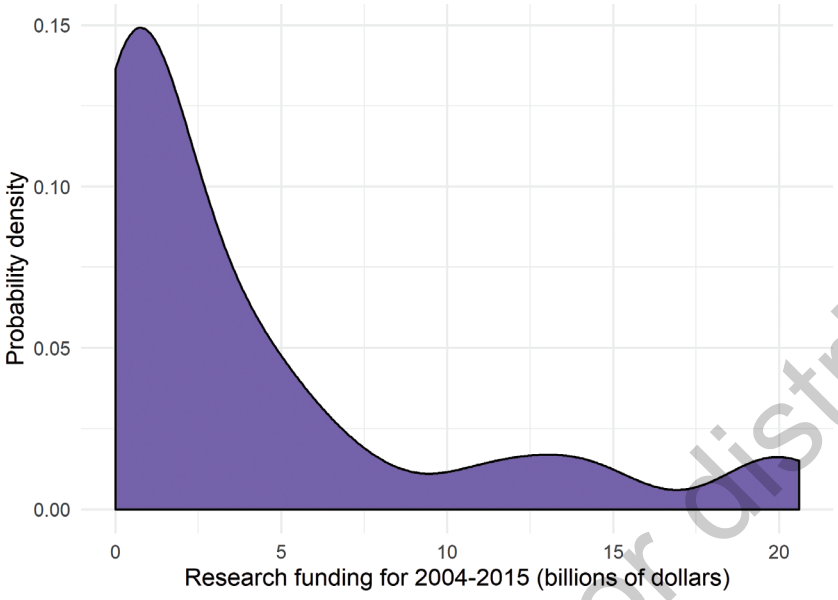


FIGURE 3.27 Research funding for the top 30 mortality causes in the United States



3.5.3 BOXPLOTS

Nancy explained that histograms and density plots were great for examining the overall shape of the data for a continuous variable, but that the boxplot was useful for identifying the middle value and the boundaries around the middle half of the data. Typically, boxplots consist of several parts:

1) A line representing the median value
2) A box containing the middle 50% of values
3) Whiskers extending to the value of the largest observation past the edge of the box, but not further than 1.5 times the IQR past the edge of the box
4) ***Outliers*** more than 1.5 times the IQR past the edge of the box

In `ggplot()`, the boxplot uses the `geom_boxplot()` function. Leslie copied her density plot code and changed the `geom_` type. Nancy explained that the boxplot would show the values of the variable along the *y*-axis by default, so instead of `x = Funding/1000000000`, Leslie needed to use `y = Funding/1000000000` in the plot aesthetics, `aes()`.

```
# boxplot of research funding (Figure 3.28)
box.funding <- research.funding %>%
  ggplot(aes(y = Funding/1000000000)) +
  geom_boxplot() +
  theme_minimal() +
  labs(y = "Research funding for 2004-2015 (billions of dollars)")
box.funding
```

FIGURE 3.28 Research funding for the top 30 mortality causes in the United States

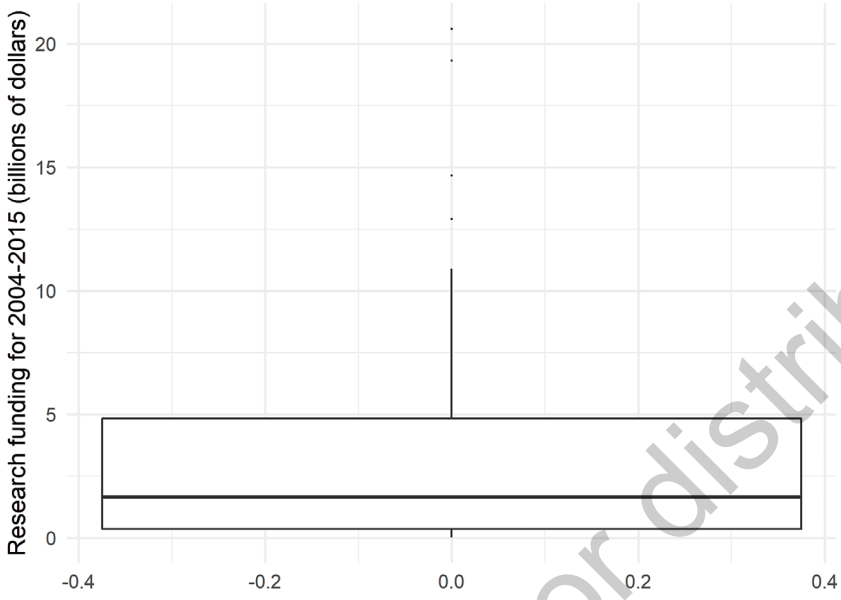


FIGURE 3.29 Research funding for the top 30 mortality causes in the United States

Figure 3.28 was a little difficult to interpret, so Kiara suggested that Leslie add a new layer of `coord_flip()` to flip the coordinates so that what used to be on the *y*-axis is now on the *x*-axis and vice versa in Figure 3.29.

```
# boxplot of research funding (Figure 3.29)
box.funding <- research.funding %>%
```

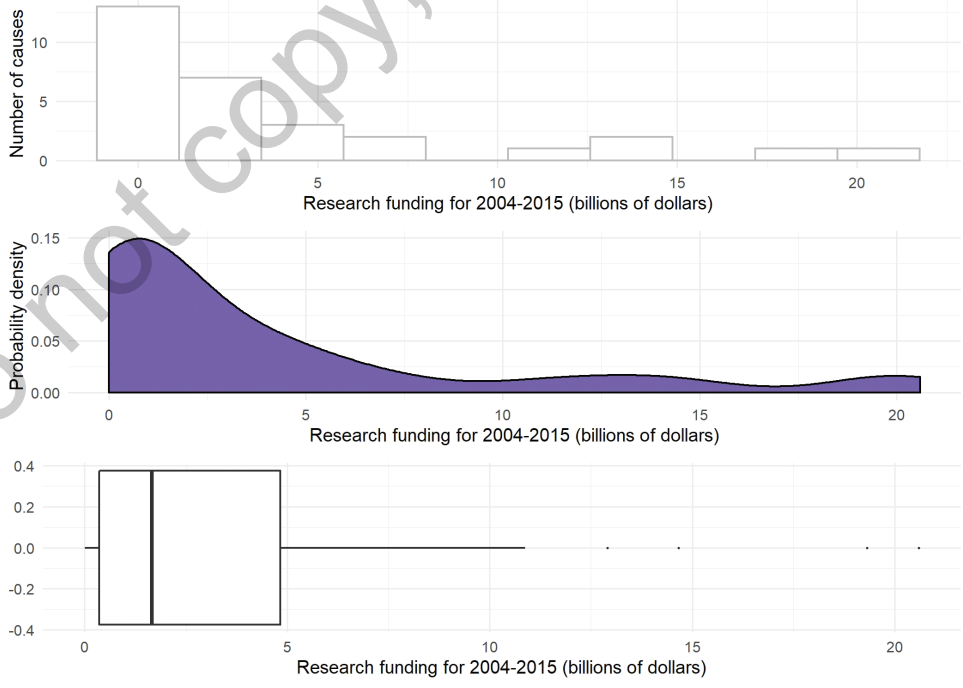
```
  ggplot(aes(y = Funding/1000000000)) +
  geom_boxplot() +
  theme_minimal() +
  labs(y = "Research funding for 2004-2015 (billions of dollars)") +
  coord_flip()
box.funding
```

She could then see the median funding level was about $2 billion based on the location of the thick black line in the middle of the box. Based on the boundaries of the box, she also determined that the middle half of the data appeared to be between about $1 billion and $5 billion.

Nancy pointed out that the right skew shown in the histogram and density plot can also be seen in this graph, with the long whisker to the right of the box and the outliers on the far right. The left whisker coming from the box and the right whisker coming from the box both extend to 1.5 times the value of the IQR away from the edge of the box (the box extends from the 25th percentile to the 75th percentile and contains the middle 50% of the data). Leslie noticed that the left whisker stopped at zero, because zero was the furthest value from the box even though it was not 1.5 × IQR below the value at the end of the box. The team agreed that each of the three graphs had strengths and weaknesses in revealing how the values of a numeric variable are distributed.

Nancy suggested they arrange the histogram, density plot, and boxplot together in order to see the similarities and differences between the three. Kiara had just the thing for that; the `grid.arrange()` function in the **`gridExtra`** package that she had used earlier to show the bar charts side by side (Figure 3.13) allows multiple graphs to be printed together. Leslie gave it a try, using the option `nrow = 3` to display one graph per row rather than side by side in columns.

FIGURE 3.30 Three graphs for examining one continuous variable at a time



```
# plot all three options together (Figure 3.30)
gridExtra::grid.arrange(histo.funding,
                        dens.funding,
                        box.funding,
                        nrow = 3)
```

Looking at the three graphs together, it was clear that they tell a consistent story, but there are some different pieces of information to be learned from the different types of graphs. All three graphs show the right skew clearly, while the histogram and boxplot show gaps in the data toward the end of the tail. The boxplot is the only one of the three that clearly identifies the central tendency and spread of the variable. The R-Team was satisfied that they had good options for displaying a single continuous variable.

3.5.4 ACHIEVEMENT 2: CHECK YOUR UNDERSTANDING

Create a histogram, a boxplot, and a density plot to show the distribution of the age variable (`RIDAGEYR`) from the NHANES 2012 data set. Explain the distribution, including an approximate value of the median, what the boundaries are around the middle 50% of the data, and a description of the skew (or lack of skew).

3.6 Achievement 3: Choosing and creating graphs for two variables at once

Kiara and Nancy explained that graphs are also used to examine relationships among variables. As with single-variable graphs and descriptive statistics, choosing an appropriate plot type depends on the types of variables to be displayed. In the case of two variables, there are several different combinations of variable types:

- Two categorical/factor
- One categorical/factor and one continuous/numeric
- Two continuous/numeric

3.6.1 MOSAIC PLOTS FOR TWO CATEGORICAL VARIABLES

There are few options for visually examining the relationship between two categorical variables. One option is a ***mosaic plot***, which shows the relative sizes of groups across two categorical variables. The NHANES data set used to demonstrate the waffle chart has many categorical variables that might be useful in better understanding gun ownership.

One of the first questions Leslie had was whether males were more likely than others to have used a gun. She had noticed that most, but not all, of the mass shootings in the United States had a male shooter and wondered if more males use guns overall. Nancy and Kiara thought that examining whether males were more likely than others to have used a gun was a good question to answer using mosaic and bar charts.

Leslie already had the `gun.use` variable ready but needed to know more about the sex or gender variables available in NHANES. She looked in the codebook to find how sex and gender were measured. She found a single sex- or gender-related variable called `RIAGENDR` that had the text "Gender

of the participant." Leslie assumed that this was the biological sex variable and looked at the way it was categorized:

- 1 = Male
- 2 = Female
- . = Missing

Leslie checked the variable before recoding.

```
# check coding of RIAGENDR
table(nhanes.2012$RIAGENDR)
##
##    1    2
## 4663 4701
```

There were no missing values, so she added the labels to the two categories in her growing data management list and renamed the variable `sex`.

```
# recode sex variable
nhanes.2012.clean <- nhanes.2012 %>%
  mutate(AUQ300 = na_if(x = AUQ300, y = 7)) %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No')) %>%
  rename(gun.use = AUQ300) %>%
  mutate(AUQ310 = recode_factor(.x = AUQ310,
                                `1` = "1 to less than 100",
                                `2` = "100 to less than 1000",
                                `3` = "1000 to less than 10k",
                                `4` = "10k to less than 50k",
                                `5` = "50k or more",
                                `7` = "Refused",
                                `9` = "Don't know")) %>%
  rename(fired = AUQ310) %>%
  mutate(RIAGENDR = recode_factor(.x = RIAGENDR,
                                  `1` = 'Male',
                                  `2` = 'Female')) %>%
  rename(sex = RIAGENDR)
```

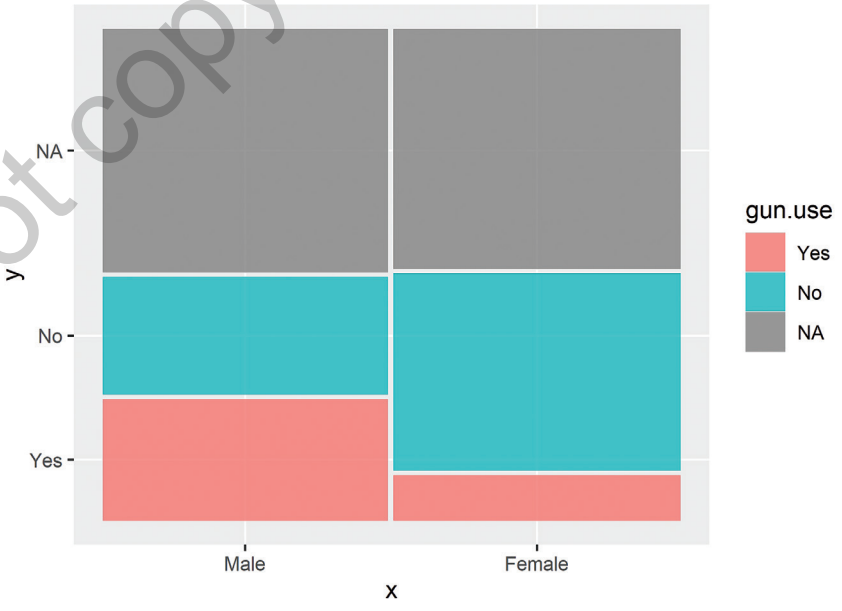
Leslie checked her recoding before working on the graph.

```
#check recoding
summary(object = nhanes.2012.clean$sex)
##   Male Female
##   4663   4701
```

Leslie showed Kiara and Nancy that the variables were now ready for graphing. The `geom_mosaic()` function is not one of those included in `ggplot()`, so it requires use of the **`ggmosaic`** package. Leslie checked the documentation to see how it was used (https://github.com/haleyjeppson/ggmosaic). It looked like the `geom_mosaic()` layer was similar to the other `geom_` options, but the variables were added to the aesthetics in the `geom_mosaic()` layer rather than the `ggplot()` layer. She wrote the basic code to see how it would look in Figure 3.31.

```
# open library
library(package = "ggmosaic")

# mosaic plot of gun use by sex (Figure 3.31)
mosaic.gun.use.sex <- nhanes.2012.clean %>%
  mutate(gun.use, gun.use = na_if(x = gun.use, y = 7)) %>%
  ggplot() +
  geom_mosaic(aes(x = product(gun.use, sex), fill = gun.use))
mosaic.gun.use.sex
```

FIGURE 3.31 Firearm use by sex in the United States among 2011–2012 NHANES participants



The resulting graph shows boxes representing the proportion of males and females who have used a gun and those who have not. There were a few things Leslie wanted to fix to make the graph more clearly convey the difference in gun use between males and females in this sample:

- Remove the NA category
- Add useful labels to the axes
- Remove the legend
- Change the colors to highlight the difference more clearly
- Change the theme so the graph is less cluttered

Nancy helped Leslie add a few of these options (Figure 3.32).

```
# formatted mosaic plot of sex and gun use (Figure 3.32)
# mosaic gun use by sex
mosaic.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot() +
  geom_mosaic(aes(x = product(gun.use, sex), fill = gun.use)) +
  labs(x = "Participant sex", y = "Ever used firearm") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    guide = FALSE) +
  theme_minimal()
mosaic.gun.use.sex
```

FIGURE 3.32 Firearm use by sex in the United States among 2011–2012 NHANES participants

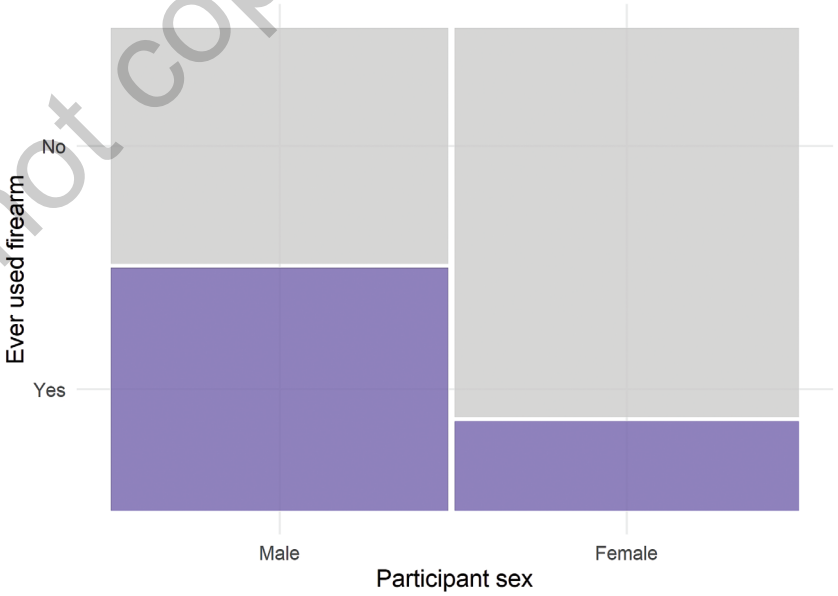


Figure 3.32 shows that the proportion of males who have used a firearm (purple bottom left) is higher than the proportion of females who have used a firearm (purple bottom right).

3.6.2 BAR CHARTS FOR TWO CATEGORICAL VARIABLES

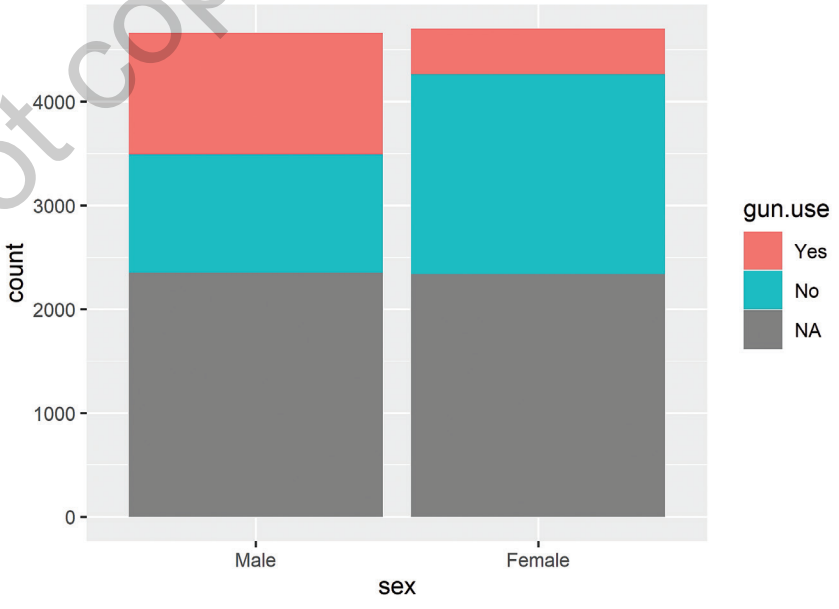
Kiara was not a big fan of the mosaic plot. She complained that it might be OK for variables with a small number of categories like `gun.use`, but using a mosaic plot for variables with many categories is not useful. She said mosaic plots have some similarity to pie charts because it is hard to tell the relative sizes of some boxes apart, especially when there are more than a few.

Kiara preferred bar charts for demonstrating the relationship between two categorical variables. Bar charts showing frequencies across groups can take two formats: stacked or grouped. Like pie charts, ***stacked bar charts*** show parts of a whole. Also like pie charts, if there are many groups or parts that are similar in size, the stacked bar chart is difficult to interpret and *not* recommended.

Grouped bar charts are usually the best option. Kiara noted that stacked and grouped bar charts could be created with `ggplot()`, but that there are two types of `geom_` that work: `geom_bar()` and `geom_col()`. After reviewing the help page, Leslie learned that `geom_bar()` is used to display the number of cases in each group (parts of a whole), whereas `geom_col()` is used to display actual values like means and percentages rather than parts of a whole. This was a little confusing to Leslie, but she expected it would become more clear if she tried a few graphs. Since the R-Team was examining the proportion of gun use by sex, Leslie decided to start with `geom_bar()` and wrote some code to get Figure 3.33.

```
# stacked bar chart (Figure 3.33)
stack.gun.use.sex <- nhanes.2012.clean %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar()
stack.gun.use.sex
```

FIGURE 3.33 Firearm use by sex in the United States among 2011–2012 NHANES participants



The resulting graph showed boxes representing the proportion of males and females who have ever used a gun or not used a gun. Like with the mosaic plot, there were a few things Leslie wanted to fix to make the graph more clearly convey the difference in gun use between males and females. Specifically, she wanted to remove the NA values, fix the titles, use the minimal theme, and add some better color. Leslie added some new layers to improve the graph (Figure 3.34).

```
# formatted stacked bar chart (Figure 3.34)
stack.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "Participant sex", y = "Number of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
stack.gun.use.sex
```

Leslie was curious about how to change this graph to a grouped bar chart since that was the recommended option. Nancy explained that the `position =` option for the `geom_bar()` layer is the place to specify whether the bars should be stacked or grouped. The default is stacked, so to get grouped, she suggested that Leslie add `position = "dodge"` to the `geom_bar()` layer. Leslie asked why "dodge" rather than "grouped" or something else. Nancy was not sure but thought it might have something to do with the use of `group =` for other purposes within the grammar of graphics. Leslie made this change to the code and tested it out in Figure 3.35.

FIGURE 3.34 Firearm use by sex in the United States among 2011–2012 NHANES participants

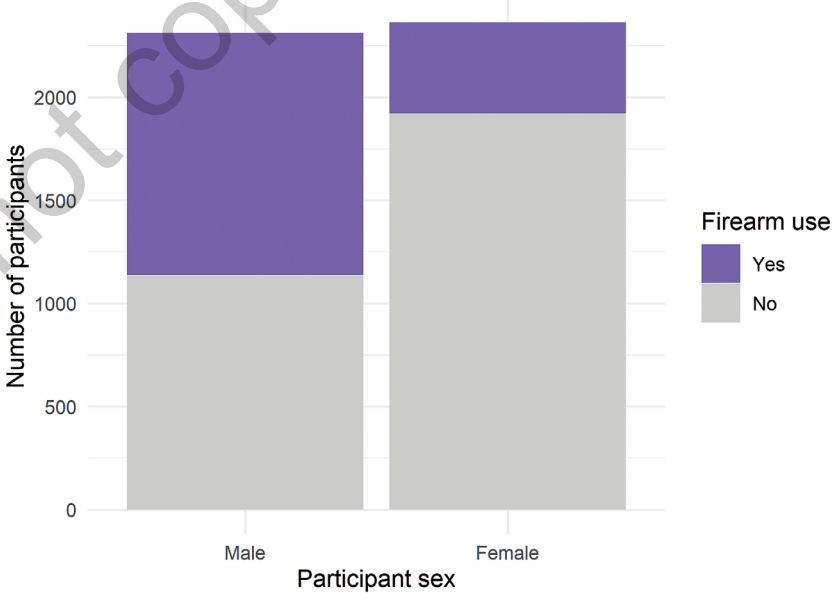
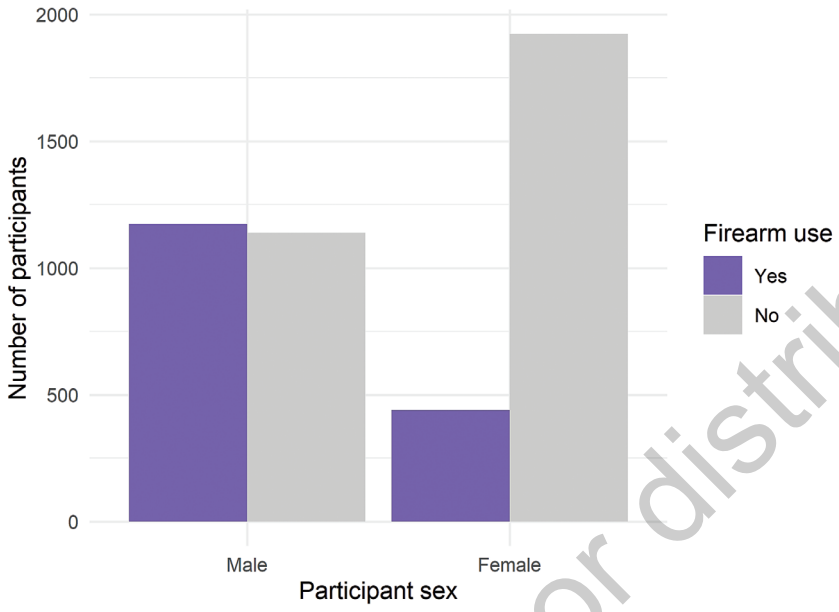


FIGURE 3.35 Firearm use by sex in the United States among 2011–2012 NHANES participants



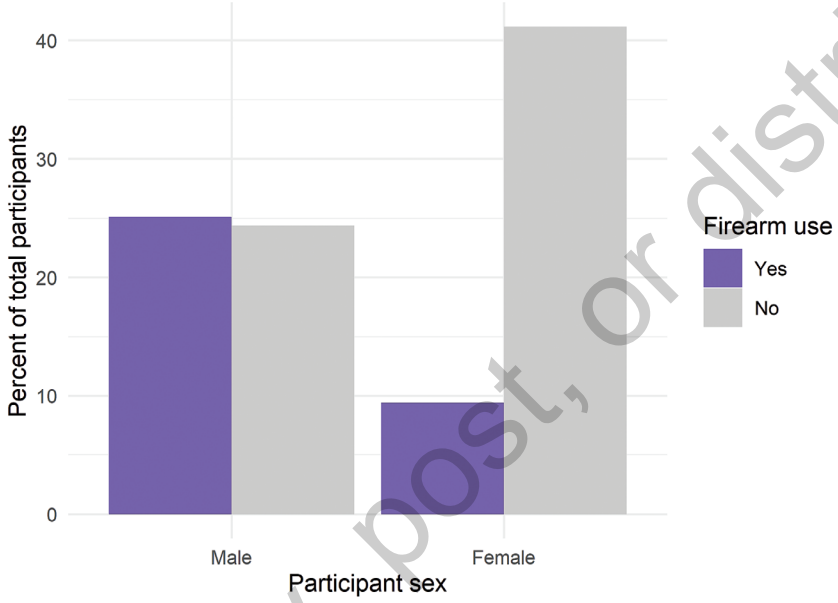

```
# formatted grouped bar chart (Figure 3.35)
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar(position = "dodge") +
  theme_minimal() +
  labs(x = "Participant sex", y = "Number of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

Nancy noted that sometimes percentages are more useful than frequencies for a bar chart. Leslie reviewed the "Using special variables in graphs" section (Section 1.10.3) from their first meeting (Figure 1.17) to remind herself how this was done. To change to percentages, Leslie added a percent calculation to the *y*-axis in the `ggplot()` to create Figure 3.36.

```
# formatted grouped bar chart with percents (Figure 3.36)
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use,
             y = 100*(..count..)/sum(..count..))) +
  geom_bar(position = "dodge") +
```

```
  theme_minimal() +
  labs(x = "Participant sex", y = "Percent of total participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

FIGURE 3.36 Firearm use by sex in the United States among 2011–2012 NHANES participants

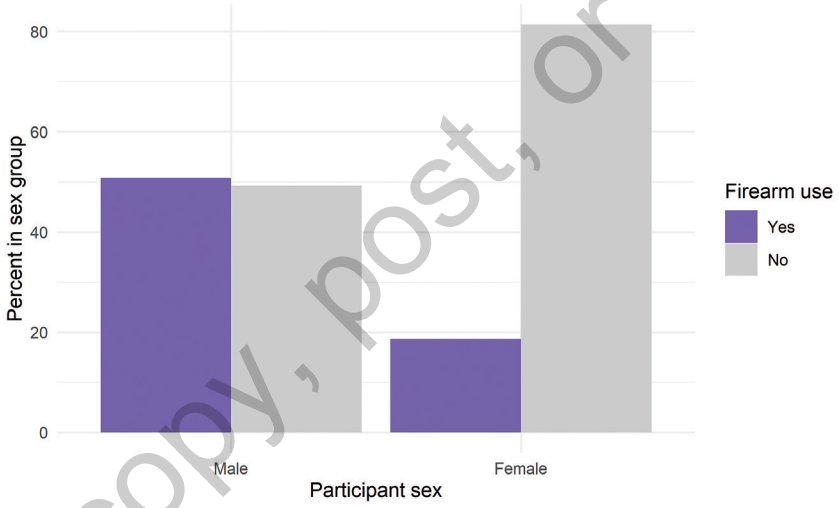


Leslie thought there was something odd about the percentages in this graph. She started estimating what they were and adding them together in her head. She figured out that all the bars together added up to 100%. This didn't seem quite right for comparing males to females since there could be more males than females overall or vice versa. She wondered if Nancy knew any code to change the percentages so that they added up to 100% *within each group*. Nancy said yes, as long as Leslie didn't mind learning some additional **tidyverse**. Leslie said she had time for one more graph, so Nancy jumped right in. Kiara worried that this code was too complicated to rely on a single comment at the top for reproducibility, and she asked Nancy if she could add in some extra comments as they went. Nancy was fine with this, and she edited the code with extra comments to get Figure 3.37.

```
# formatted grouped bar chart with percents (Figure 3.37)
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  group_by(gun.use, sex) %>%              # make groups of gun.use by sex
  count() %>%                             # count how many are in each group
  group_by(sex) %>%                       # pick the variable that will add
                                          # to 100%
```

```
  mutate(percent = 100*(n/sum(n))) %>%        # compute percents within
                                              # chosen variable

  ggplot(aes(x = sex, fill = gun.use,
             y = percent)) +                  # use new values from mutate
  geom_col(position = "dodge") +
  theme_minimal() +
  labs(x = "Participant sex",
       y = "Percent in sex group") +
  scale_fill_manual(values = c("#7463AC",
                               "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

FIGURE 3.37 Firearm use by sex among 2011–2012 NHANES participants



Nancy was pretty pleased with herself when this ran. Kiara was not happy with the documentation, but it was a start. While the code was long and Leslie was a little overwhelmed, Nancy reassured her that learning `ggplot()` code can be really complicated. Nancy shared a tweet from Hadley Wickham, the developer of **`ggplot2`** (Figure 3.38).

Leslie found that putting all of the options together in a grid to compare how well they do at conveying information was really useful for the single continuous variables, so she wrote one last section of code to compare the graph types for the two categorical variables (Figure 3.39).

```
# plot all three options together (Figure 3.39)
gridExtra::grid.arrange(mosaic.gun.use.sex,
                        stack.gun.use.sex,
                        group.gun.use.sex,
                        nrow = 2)
```

FIGURE 3.38 Screenshot of tweet from ggplot2 developer Hadley Wickham's Twitter account. It turns out even experts need to use the help documentation!

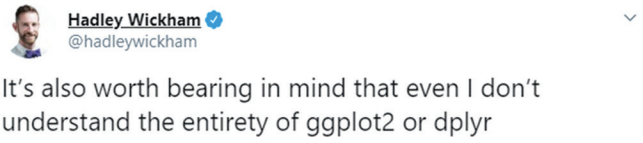
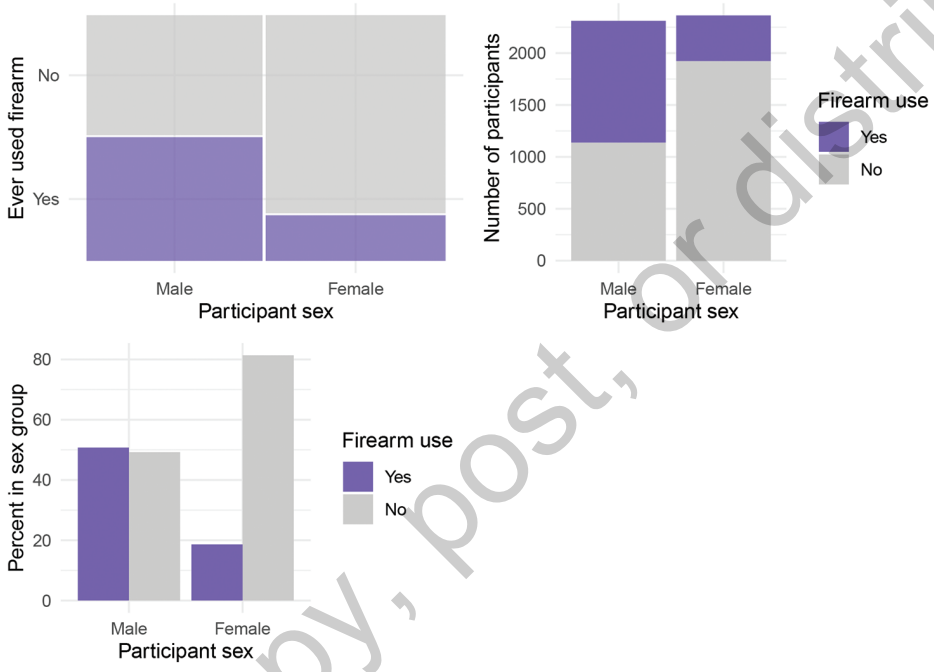


FIGURE 3.39 Three options for graphing two categorical variables together



For Leslie, these graphs were pretty similar in the information they conveyed. The mosaic plot and stacked bar chart were almost the same, with the exception of the *y*-axis, which showed the number of participants for the stacked bar chart. The grouped bar chart did seem to convey the difference between the groups most clearly, making it easy to compare firearm use both within the male and female group and between the two groups. In terms of communicating statistical results, Leslie thought the grouped bar chart might become one of her favorites along with the boxplot.

3.6.3 BAR CHARTS, POINT CHARTS, BOXPLOTS, AND VIOLIN PLOTS FOR ONE CATEGORICAL AND ONE CONTINUOUS VARIABLE

Kiara suggested that bar charts can also be useful for examining how continuous measures differ across groups. For example, the NHANES data include a measure of age in years. The R-Team already knew that a higher percentage of males than females use firearms. They decided to also examine whether firearm users tend to be younger or older than those who do not use firearms. Age is measured in years, which is not *truly* continuous since partial years are not included, but the underlying concept is a continuous one, with age spanning across a continuum of years rather than being broken up into categories.

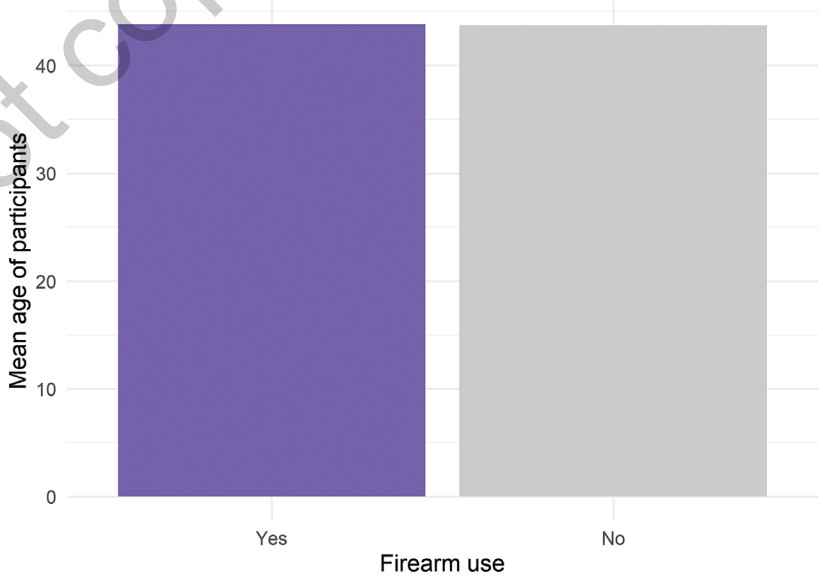
3.6.3.1 DATA MANAGEMENT

Age in years is measured along a continuum while firearm use is categorical, with two categories. A bar chart could show two bars for gun use (Yes and No) with the height of each bar based on the mean or median age of gun users or gun nonusers. Nancy eagerly started to type the code, but Leslie slid the laptop away from her and tried it herself by copying and editing the code from making Figure 3.15. After a couple of minutes, Nancy pulled the laptop back when she saw Leslie had gotten stuck on how to get the mean age on the *y*-axis. She showed Leslie how to add summary statistics in a bar chart by adding `stat = "summary"` to the `geom_bar()` layer. Once summary is specified, the layer also needs to know which summary statistic to use. Adding `fun.y = mean` will result in the mean of the `y =` variable from the aesthetics, which, in this case, is `y = RIDAGEYR` for age. Leslie nodded and pulled the laptop back to herself to edit some of the axis labels and run the code (Figure 3.40).

```
# bar chart with means for bar height (Figure 3.40)
bar.gun.use.age <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use, y = RIDAGEYR)) +
  geom_bar(aes(fill = gun.use), stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm use", y = "Mean age of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    guide = FALSE)
bar.gun.use.age
```

There was not much of a difference in the mean age of those who have used a firearm and those who have not used a firearm. Both groups were just under 45 years old as a mean. This graph was not a very

FIGURE 3.40 Mean age by firearm use for 2011–2012 NHANES participants



good use of space when just reporting that both means were 45 would suffice. Kiara was too busy looking over the code format to notice. Leslie thought back to the descriptive statistics meeting and remembered that the mean is only useful when the data are normally distributed. She suggested to Nancy that they check the distribution of age for people who do and do not use firearms. Nancy started typing immediately. Leslie looked over her shoulder as she made a density plot (Figure 3.41).

```
# density plots of age by firearm use category (Figure 3.41)
dens.gun.use.age <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = RIDAGEYR)) +
  geom_density(aes(fill = gun.use), alpha = .8) +
  theme_minimal() +
  labs(x = "Age of participants", y = "Probability density") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Used firearm")
dens.gun.use.age
```

Leslie was impressed! It looked like two density plots on top of each other. Nancy showed her the code and pointed out the `fill = gun.use` option in the `geom_density()` layer, which resulted in two density plots with two colors. Nancy also pointed out the `alpha = .8` in the `geom_density()` layer. The alpha sets the level of transparency for color, where 1 is not transparent and 0 is completely transparent. The .8 level allows for some transparency while the colors are mostly visible. She reminded Leslie that since the transparency level is not based on anything in the data set, the `alpha = .8` option should *not* be wrapped within the `aes()`. The rest of the graph was familiar to Leslie since it had the same options they had been using all day.

FIGURE 3.41 Distribution of age by firearm use for 2011–2012 NHANES participants

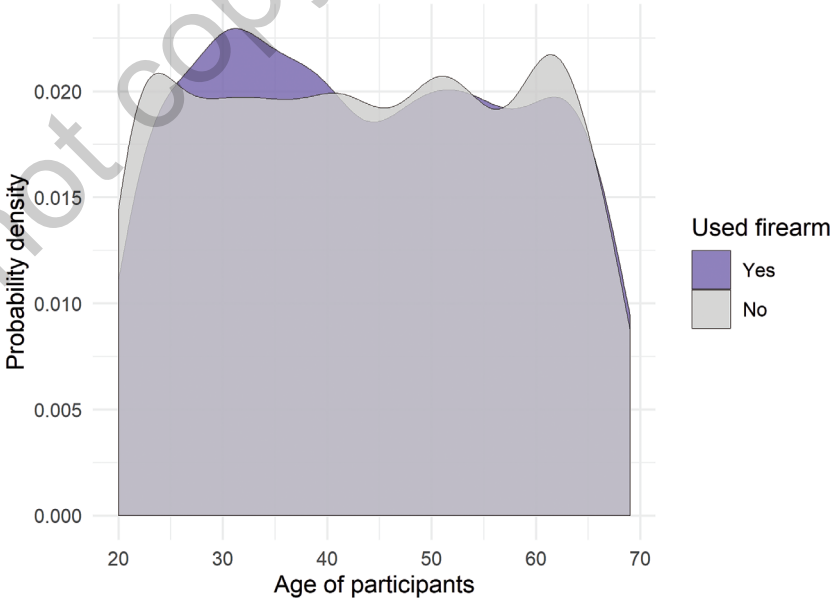
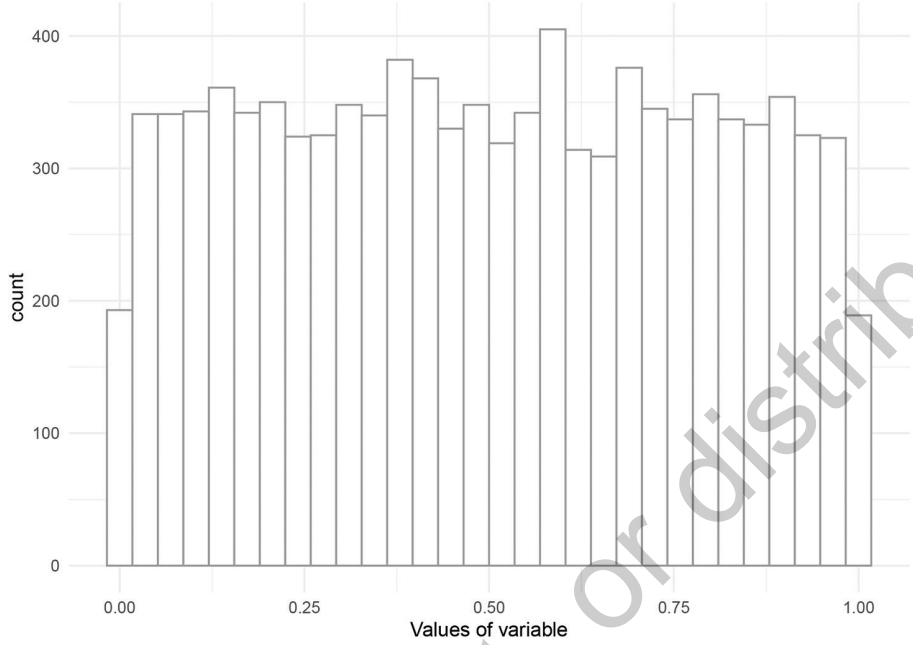


FIGURE 3.42 Example of a uniform distribution

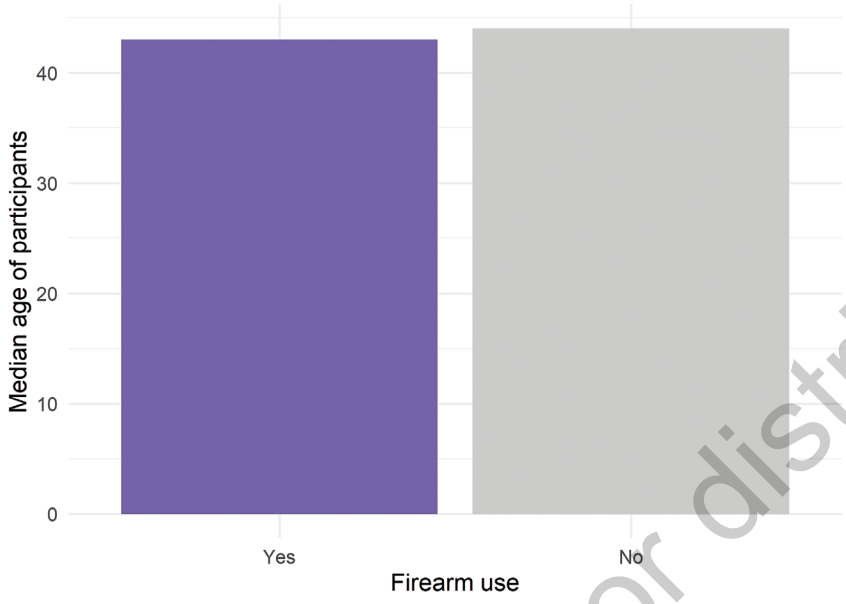


After talking through the code, Leslie looked at the graph. The distributions definitely did not look normal, she thought. But they did not look skewed either. Leslie remembered her stats class and looked through her old notes for distribution shapes. She found that this graph looked more like a ***uniform distribution*** than any of the other options. She read that a perfect uniform distribution had the same frequency for each value of the variable. Essentially, it looked like a rectangle. Nancy went ahead and plotted an example of a uniform distribution in Figure 3.42.

Leslie was not sure they needed an example, but this confirmed that the distribution of age for firearm users and nonusers had a uniform distribution. Since the distribution was not normally distributed, Leslie suggested they use the median instead of the mean. She started copying the code from the previous bar chart, replacing the `mean` with `median` in the `fun.y =` option of the `geom_bar()` layer (Figure 3.43).

```
# bar chart with median for bar height (Figure 3.43)
bar.gun.use.age.md <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = gun.use, y = RIDAGEYR)) +
  geom_bar(aes(fill = gun.use), stat = "summary", fun.y = median) +
  theme_minimal() +
  labs(x = "Firearm use", y = "Median age of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    guide = FALSE)
bar.gun.use.age.md
```

FIGURE 3.43 Median age by firearm use for 2011–2012 NHANES participants



Leslie sighed. It still didn't look like a necessary plot. The median age of those who have used a firearm is maybe 1 or 2 years younger than the median age of those who have not used a firearm. Leslie thought this graph might be more useful if it included data spread, too. She remembered that measures of central tendency tend to be reported with measures of spread, and she asked Nancy if there was a way to add some indication of spread to the graph. Since they had decided on the median, was there a way to show its corresponding measure of spread, the IQR?

Nancy thought for a minute and remembered using a `geom_errorbar()` layer to add standard deviations to bar charts in the past, and thought this might also work to add IQR. She asked Leslie if she was up for more **`tidyverse`**. Leslie gave her two thumbs up, so Nancy started coding. Kiara noticed they were on to something new and wanted to make sure it was documented well for reproducibility. She squeezed in between Nancy and Leslie so she could suggest comments to write as they worked (Figure 3.44).

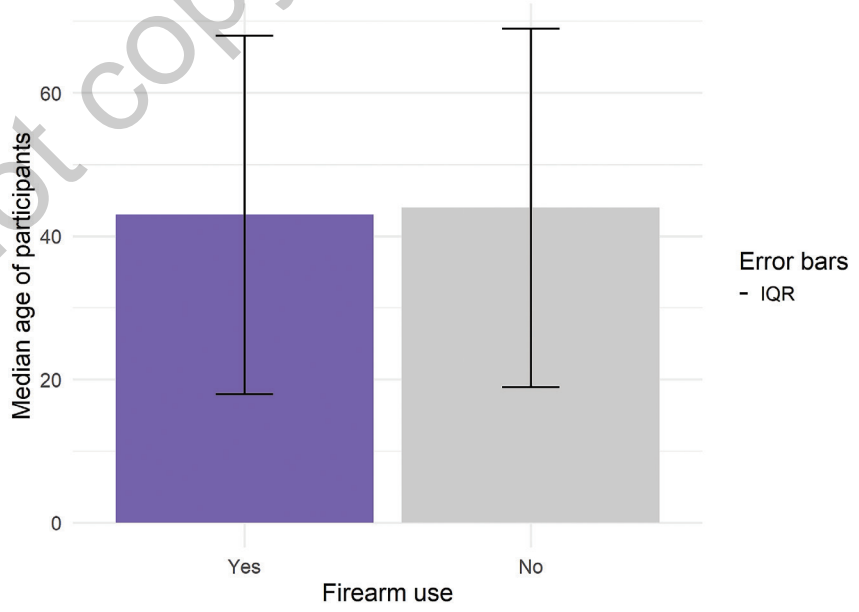
```
# bar chart with median for bar height and error bars (Figure 3.44)
gun.use.age.md.err <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  group_by(gun.use) %>%  # specify grouping variable
  summarize(central = median(RIDAGEYR),       # median, iqr by group
            iqr.low = quantile(x = RIDAGEYR, probs = .25),
            iqr.high = quantile(x = RIDAGEYR, probs = .75) ) %>%
  ggplot(aes(x = gun.use, y = central)) +     # use central tend for y-axis
  geom_col(aes(fill = gun.use)) +
  geom_errorbar(aes(ymin = iqr.low,           # lower bound of error bar
                    ymax = iqr.high,          # upper bound of error bar
```

```
                    linetype = "IQR"),
                width = .2) +                    # width of error bar
  theme_minimal() +
  labs(x = "Firearm use", y = "Median age of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    guide = FALSE) +
  scale_linetype_manual(values = 1, name = "Error bars")
gun.use.age.md.err
```

While Leslie was happy that they had gotten error bars to show up, Figure 3.44 still wasn't all that interesting. Both groups had median ages in the early 40s, and about 50% of the observations in each group were between ages 30 and 60 based on the IQR error bars. The information in the graph could be easily reported in a single sentence rather than use so much space for so little information. She did notice that Nancy included `linetype =` in the `aes()` for the error bars. Nancy explained that she wanted it to be clear that the error bars were the IQR and not some other measure of spread. To specify IQR, she added a legend. Options included in the `aes()` (other than `x` and `y`) are added to a legend. By using `linetype = "IQR"`, she added a legend that would label the linetype as "IQR." Leslie noticed there was a line at the bottom of the code that included `scale_linetype_manual(values = 1, name = "")`. Nancy explained that, like the `scale_color_manual()` they had been using to specify colors, this allowed her to specify what type of line she wanted (she picked type 1) and several other options, such as whether or not to include the legend at all (`guide =`) and what the title of the legend should be (`name = "Error bars"`).

Kiara thought the same graph types with the FBI deaths data might show more variation and could be useful to add to their understanding of the role of gun types in homicides. Specifically, they could determine if there was a difference in the mean number of gun homicides per year by gun type. Leslie thought this was a good idea and would give her the chance to work with some of the great code that Nancy had created.

FIGURE 3.44 Median age with IQR for groups of firearm use for 2011–2012 NHANES participants



Nancy was interested to learn how she had downloaded the file, so Kiara copied the code and put it in Box 3.1 for Nancy to review. Leslie imported the **fbi_deaths_2016_ch3.csv** data set into R. Since it was a small data frame with just six variables, she used `summary()` to examine the data.

```
# import FBI data
fbi.deaths <- read.csv(file = "[data folder location]/data/fbi_deaths_2016_
ch3.csv")

# review the data
summary(object = fbi.deaths)
##                              Weapons X2012
##  Asphyxiation                          : 1 Min.   :    8.00
##  Blunt objects (clubs, hammers, etc.): 1 1st Qu.:   87.75
##  Drowning                              : 1 Median :  304.00
##  Explosives                            : 1 Mean   : 1926.28
##  Fire                                  : 1 3rd Qu.: 1403.50
##  Firearms, type not stated             : 1 Max.   :12888.00
##  (Other)                               :12
##      X2013               X2014               X2015               X2016
##  Min.   :    2.00   Min.   :    7.0   Min.   :    1.00   Min.   :    1.0
##  1st Qu.:   87.25   1st Qu.:   75.5   1st Qu.:   87.75   1st Qu.:  100.2
##  Median :  296.50   Median :  261.0   Median :  265.00   Median :  318.0
##  Mean   : 1831.11   Mean   : 1825.1   Mean   : 2071.00   Mean   : 2285.8
##  3rd Qu.: 1330.00   3rd Qu.: 1414.2   3rd Qu.: 1410.00   3rd Qu.: 1428.8
##  Max.   :12253.00   Max.   :12270.0   Max.   :13750.00   Max.   :15070.0
##
```

It looks like each year was a variable in this data frame, and each observation was a type of weapon. Kiara thought a few things needed to happen before the data could be graphed. The most important thing to do would be to change the data set from wide, with one variable per year, to long. A long data set would have a variable called `year` specifying the year. Nancy knew what she wanted to do to make this happen and pulled the laptop over to write some code.

```
# make a long data frame
fbi.deaths.cleaned <- fbi.deaths %>%
  slice(3:7) %>%
  gather(key = "year", value = "number", X2012,
         X2013, X2014, X2015, X2016) %>%
  mutate(year, year = substr(x = year, start = 2, stop = 5)) %>%
  rename(weapons = Weapons)
```

Leslie asked Nancy to walk her through this code. Kiara suggested Leslie add comments as they went so that the code would be easier to understand later. Leslie agreed and Nancy started explaining. The

first step was to isolate the different types of firearms in the data. One way to do this was to select the rows that had firearms in them. She suggested Leslie open the original data and identify the rows with firearm information. Leslie opened the original data set and saw that rows 3 to 7 had the five firearm types in them. Nancy then introduced `slice()`, which allows the selection of observations (or rows) by their position; in this case she used `slice(3:7)` to select rows 3 through 7. Kiara nudged Leslie to write a comment next to the `slice(3:7)` line. Leslie asked if `slice(3:7)` in this code would be like taking a subset. Would it be the same as `fbi.deaths[3:7,]`? Nancy confirmed that this was true.

Next, Nancy explained that `gather()` is a little tricky, but essentially it takes variables (i.e., columns) and turns them into observations (i.e., rows). The first two arguments were the new variable names (in quotes), and the last five arguments were the old variable names. Leslie was confused, but she wrote a comment and Nancy kept going.

The third task was `mutate()`, which was just used to remove the `X` from the beginning of the year values. The years were showing up as `X2012` instead of just `2012`. Using `substr()`, or substring, allows part of the word to be removed by specifying which letters to keep. By entering 2 and 5, `substr()` kept the values of the year variable starting at the second letter through the fifth letter.

Finally, `rename()` changed the variable named `Weapons` to a variable named `weapons`. Nancy preferred lowercase variable names for easier typing. Leslie understood, and Kiara was OK with the new comments in the code.

```
# make a long data frame
fbi.deaths.cleaned <- fbi.deaths %>%
  slice(3:7)            %>% # selects rows 3 to 7
  gather(key = year, value = number, X2012,
         X2013, X2014, X2015, X2016) %>% # turn columns into rows
  mutate(year,
         year = substr(x = year,
                       start = 2,
                       stop = 5)) %>% # remove X from front of year entries
  rename(weapons = Weapons)
```

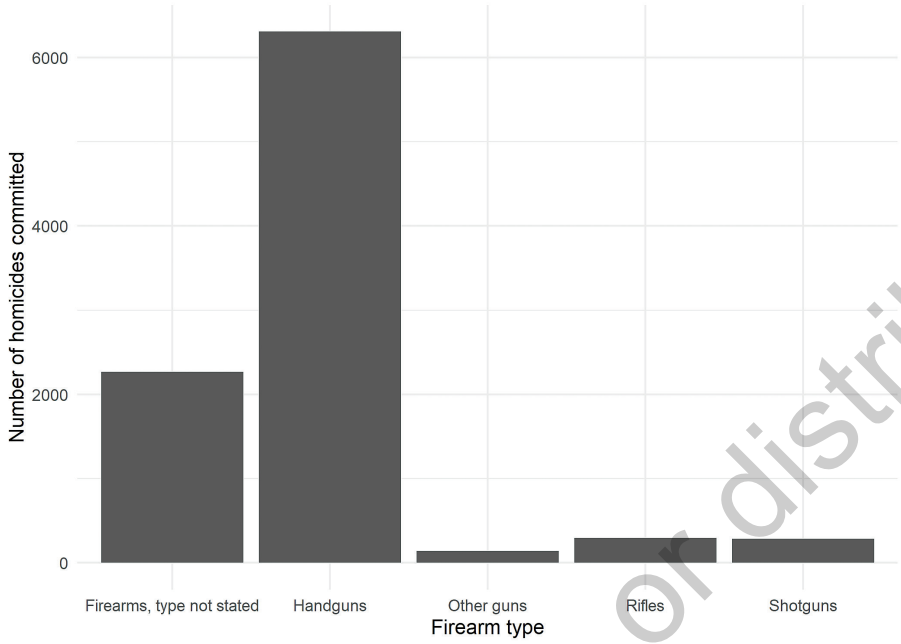
They decided it was time to make a bar chart.

3.6.3.2 BAR CHART

Kiara ran the same code she used for graphing mean age and gun use, but with the new data frame and variable names. She also changed the axis labels so they fit the data being graphed (Figure 3.45).

```
# plot number of homicides by gun type (Figure 3.45)
bar.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = weapons, y = number)) +
  geom_bar(stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides committed")
bar.homicide.gun
```

FIGURE 3.45 Mean annual homicides committed by gun type in the United States, 2012–2016

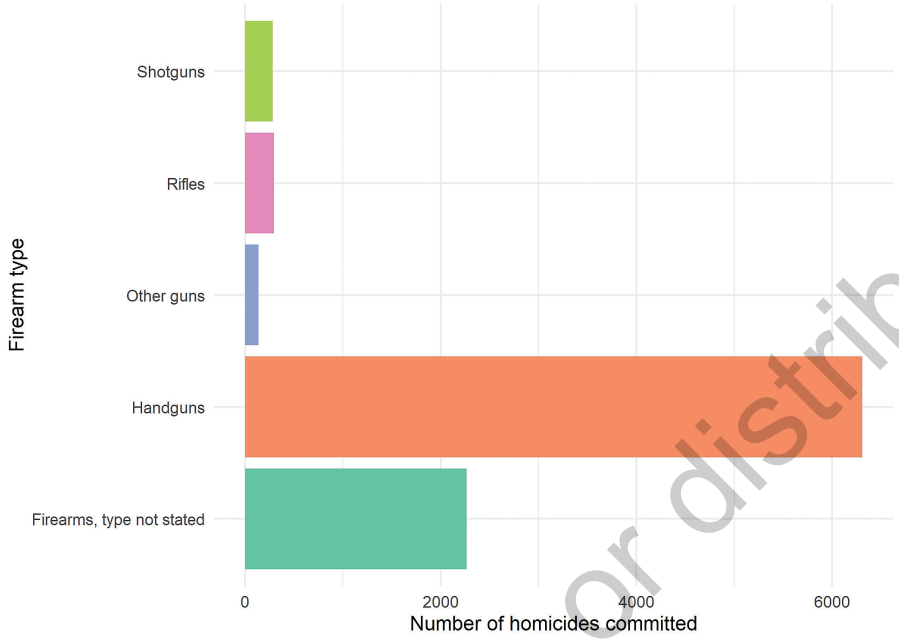


Leslie thought it might be easier to read this bar chart if it were flipped since some of the bar labels were complicated. Nancy flipped the coordinates by adding a `coord_flip()` layer. While she was working on it, she added some color to the bars using `scale_fill_brewer()` (Wickham et al., n.d.), which has a number of built-in color schemes (including many that are color-blind friendly) that are directly from the Color Brewer 2.0 website (http://colorbrewer2.org/). She tried a few of the palette options before choosing to use the Set2 palette by adding `palette = "Set2"` (Figure 3.46).

```
# flip the coordinates for better reading (Figure 3.46)
# add color and remove unnecessary legend
bar.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = weapons, y = number)) +
  geom_bar(aes(fill = weapons), stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides committed") +
  coord_flip() +
  scale_fill_brewer(palette = "Set2", guide = FALSE)
bar.homicide.gun
```

The team discussed their strategies for using different options in the code to produce graphs that demonstrate a certain point or idea. For example, if the primary reason for creating the graph had been to highlight the role of handguns in homicide, using color to call attention to the length of the handgun bar would have been one way to highlight this fact. Nancy also suggested changing the order of the bars so that the bars would be in order by length. Leslie asked her to explain the code so she could practice it. Nancy said that `reorder()` can be used to order the bars from largest to smallest by the value of the number variable. She instructed Leslie to type `reorder()` in the `aes()` as part of the `x =` argument

FIGURE 3.46 Mean annual homicides by firearm type in the United States, 2012–2016

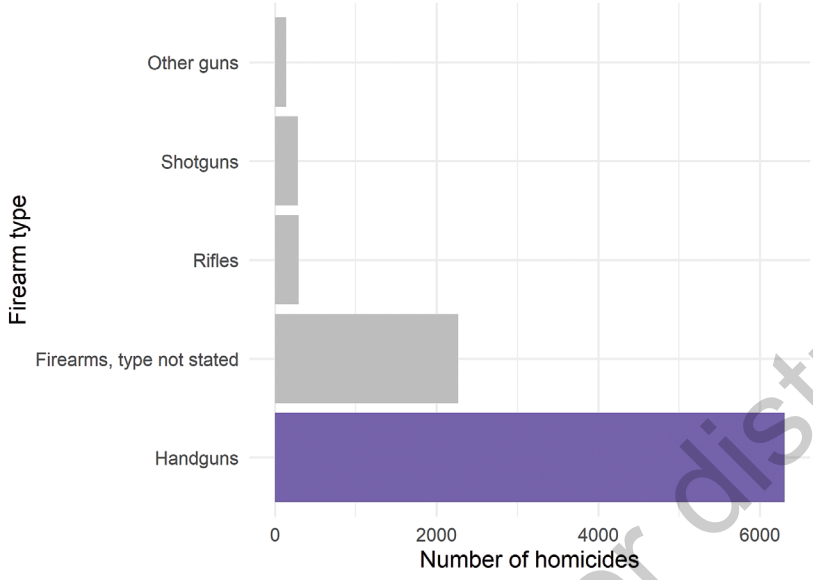


and then, within the parentheses, add the variable to be put in order and the variable that should be used to decide the order, like this: `reorder(weapons, -number)`. This means the factor `weapons` will be placed in order based on the numeric `number` variable. Leslie asked why the minus sign was in there. Nancy said this was to specify that the order should go from the smallest value to the largest value.

Kiara said they could use a strategy for assigning color to the bars similar to the one they used to assign color to categories in the waffle chart (Figure 3.18). Specifically, they could set each category equal to the name of the color to represent it. Leslie and Nancy both looked confused. Nancy quoted statistician Linus Torvalds when she said to Kiara, "Talk is cheap. Show me the code." Kiara added `reorder()` and the colors and showed her the code for Figure 3.47.

```
# highlight handguns using color (Figure 3.47)
bar.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number), y = number)) +
  geom_bar(aes(fill = weapons), stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_fill_manual(values = c("Handguns" = "#7463AC",
                               "Firearms, type not stated" = "gray",
                               "Rifles" = "gray",
                               "Shotguns" = "gray",
                               "Other guns" = "gray"), guide=FALSE)
bar.homicide.gun
```

FIGURE 3.47 Mean annual homicides by firearm type in the United States, 2012–2016



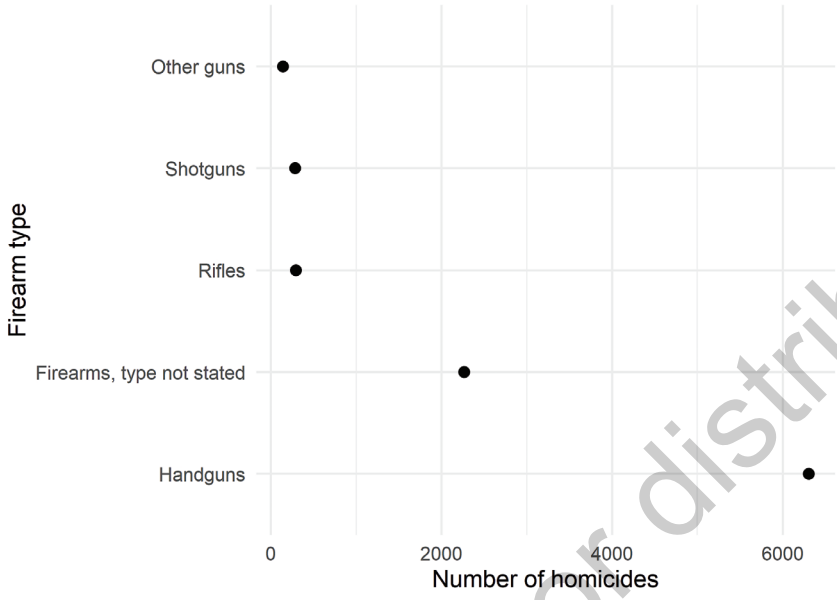
The R-Team agreed this use of color added emphasis to understanding the pattern of weapons used in homicides. Leslie asked if it was dishonest to emphasize a bar like this. Kiara thought the added emphasis was fine; dishonesty occurred when people changed the *x*-axis or *y*-axis or used other strategies to make differences look artificially bigger or smaller. In this case, handguns were the most used weapon, with a mean of more than 6,000 homicides per year by handgun.

3.6.3.3 POINT CHART

Nancy showed Leslie that the same data could be displayed with a single point rather than a bar. She used the code for the bar chart above and changed the `geom_bar()` layer to a `geom_point()` layer (Figure 3.48).

```
# gun deaths by gun type (Figure 3.48)
# highlight handguns using color
point.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number), y = number)) +
  geom_point(aes(fill = weapons), stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_fill_manual(values = c("Handguns" = "#7463AC",
                               "Firearms, type not stated" = "gray",
                               "Rifles" = "gray",
                               "Shotguns" = "gray",
                               "Other guns" = "gray"), guide=FALSE)
point.homicide.gun
```

FIGURE 3.48 Mean annual homicides by firearm type in the United States, 2012–2016

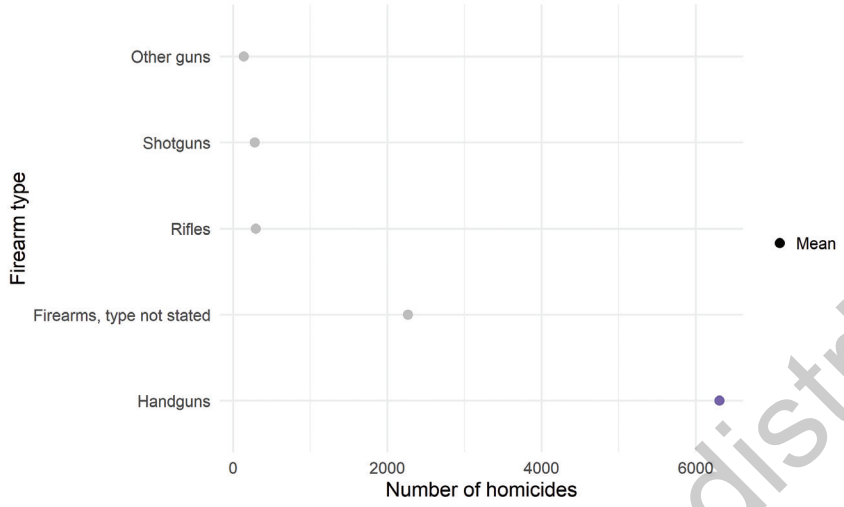


Leslie noticed that the colors did not show up in this new graph. Nancy fixed this by changing the term `fill =` to `color =`. She explained that `fill` is used to fill bars, while `color` works to color dots. Nancy thought this was also a good time to show off one more code trick, and she made the points larger using the `size =` option in the `geom_point()` layer (Figure 3.49).

```
# change fill to color add size to geom_point (Figure 3.49)
point.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number), y = number)) +
  geom_point(aes(color = weapons, size = "Mean"),
             stat = "summary", fun.y = mean) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_color_manual(values = c("Handguns" = "#7463AC",
                                "Firearms, type not stated" = "gray",
                                "Rifles" = "gray",
                                "Shotguns" = "gray",
                                "Other guns" = "gray"), guide = FALSE) +
  scale_size_manual(values = 4, name = "")
point.homicide.gun
```

Leslie thought the bar chart was a little better at emphasizing, although this graph was not bad and would require less ink to print. Leslie asked Nancy if she could add the error bars to this graph, like she did with Figure 3.44. Nancy was up to the challenge and coded Figure 3.50. She took this opportunity to

FIGURE 3.49 Mean annual homicides by firearm type in the United States, 2012–2016



show Leslie one additional feature, which was to move the legend to another part of the graph, like the top or bottom. Leslie remembered seeing legends on the bottom of histograms in their previous meeting (Figure 2.7). Nancy confirmed that this was how they formatted those figures.

```
# add error bars (Figure 3.50)
point.homicide.gun <- fbi.deaths.cleaned %>%
  group_by(weapons) %>%
  summarize(central = mean(x = number),
            spread = sd(x = number)) %>%
  ggplot(aes(x = reorder(x = weapons, X = -central),
             y = central)) +
  geom_errorbar(aes(ymin = central - spread,
                    ymax = central + spread,
                    linetype = "Mean\n+/- sd"),
                width = .2) +
  geom_point(aes(color = weapons, size = "Mean"), stat = "identity") +
  theme_minimal() +
  labs(x = "Firearm type",
       y = "Number of homicides") +
  coord_flip() +
  scale_color_manual(values = c("Handguns" = "#7463AC",
                                "Firearms, type not stated" = "gray",
                                "Rifles" = "gray",
                                "Shotguns" = "gray",
```

```
                                  "Other guns" = "gray"), guide=FALSE) +
  scale_linetype_manual(values = 1, name = "") +
  scale_size_manual(values = 4, name = "") +
  theme(legend.position = "top")
point.homicide.gun
```

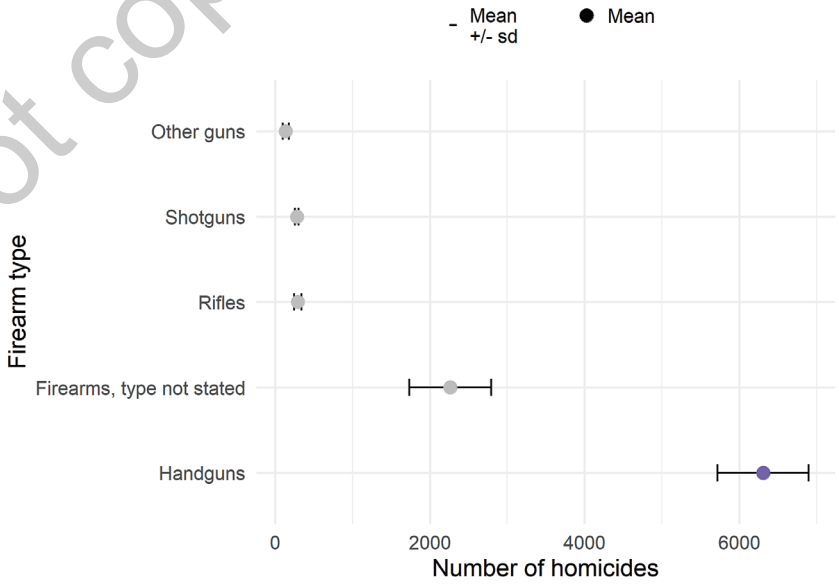
Leslie's first thought was that the means with their standard deviation error bars looked like TIE Fighters from *Star Wars*! She noticed that the standard deviations were very small for the `Other guns`, `Shotguns`, and `Rifles` groups. For these groups, the error bars did not even extend outside the dots. There was not much spread or variation in the number of homicide deaths by these three types of firearms. For `Handguns`, the error bar showed that the observations are spread to a few hundred homicides above and below the mean of 6,000. Leslie remembered that the data set was pretty small, based on just 5 years of data, which might be one of the reasons there was not a lot of spread or variation in the number of homicides per type of firearm. If the data were for more years, there might (or might not) be more variation due to mass homicide events, policy changes, or other factors.

Leslie was getting nervous that they might be using inappropriate measures of central tendency and spread since they did not know if the data were normally distributed. She thought the boxplots might be better at showing the distribution in each group so they could be sure they were choosing the most appropriate plots to interpret and report.

3.6.3.4 BOXPLOTS

Nancy slid the laptop over in front of her to make the boxplots. She used the code from the point chart and changed the `geom_` layer to make a boxplot (Figure 3.51).

FIGURE 3.50 Mean annual homicides by firearm type in the United States, 2012–2016



```
# change to boxplot (Figure 3.51)
box.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number),
             y = number)) +
  geom_boxplot(aes(color = weapons)) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_color_manual(values = c("Handguns" = "#7463AC",
                                "Firearms, type not stated" = "gray",
                                "Rifles" = "gray",
                                "Shotguns" = "gray",
                                "Other guns" = "gray"), guide=FALSE)
box.homicide.gun
```

Nancy noted that boxplot color is specified with `fill =` in order to fill the boxplots instead of outlining them (Figure 3.52).

```
# fix color for boxplots (Figure 3.52)
box.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number),
             y = number)) +
  geom_boxplot(aes(fill = weapons)) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_fill_manual(values = c("Handguns" = "#7463AC",
                               "Firearms, type not stated" = "gray",
                               "Rifles" = "gray",
                               "Shotguns" = "gray",
                               "Other guns" = "gray"), guide=FALSE)
box.homicide.gun
```

Nancy pushed the laptop back to Leslie so that she could practice. She was starting to feel bad that Leslie wasn't coding all that much. Leslie found that while the bar chart and point chart were great for comparing the means of the groups, the boxplot provided more information about the distribution in each group. For example, over the 2012–2016 time period, the number of handguns and unspecified firearms used in homicides varied a lot more than the use of the other three firearm types. She could tell this was the case because the boxes encompassing the middle 50% of the data were wider, so the IQR was larger. This might suggest that a closer examination of the trends in the production and use of handguns could be useful for understanding what was going on.

FIGURE 3.51 Annual homicides by firearm type in the United States, 2012–2016

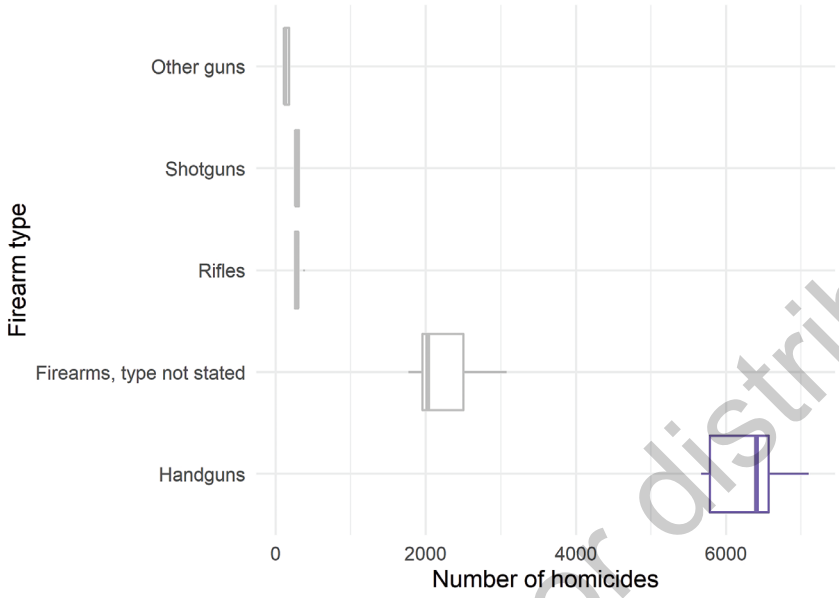
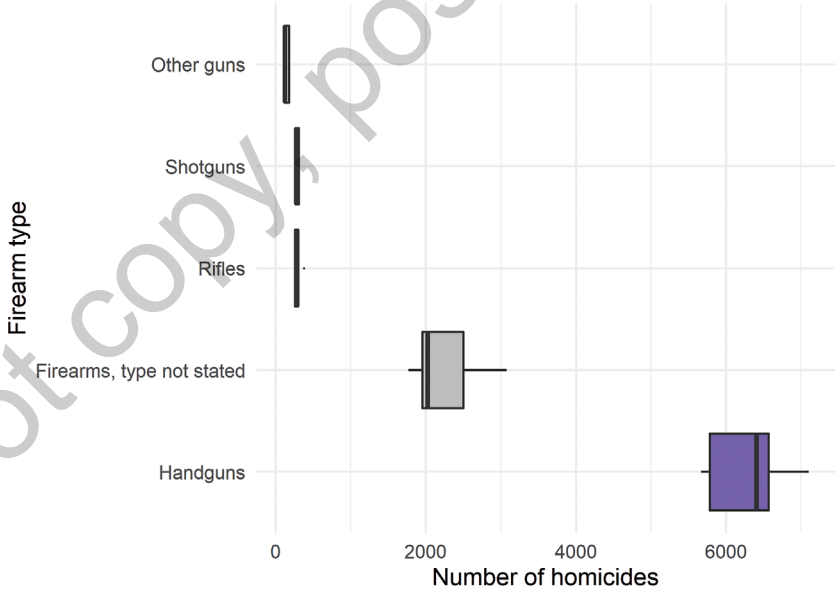


FIGURE 3.52 Annual homicides by firearm type in the United States, 2012–2016



Leslie noticed that the boxplot also suggested that the distributions for the `Firearms, type not stated` and `Handguns` categories were skewed. She could tell because for the `Firearms, type not stated` category, the median is on the far left of the box, indicating that there are some larger values on the right of this distribution. She remembered what she had learned about calculating means and medians and thought that the mean values they had been reviewing might have been misleading for

this group since the large values would make the mean seem larger (just like Bill Gates's salary would make the mean salary of your friends seem larger). Likewise, Leslie noticed that the median was toward the right-hand side of the `Handguns` box. This indicated there might be small values in this group that would have resulted in a smaller mean value. Given the skew, Leslie thought they would be better off using the boxplot or changing the bar chart or point chart to medians rather than means.

Nancy was not all that interested in the statistical concepts but wanted to show Leslie one more code trick. She knew a way to show the data points and the boxplots at the same time. Leslie liked this idea since it would help her to understand why the boxplots seem to show some skew. Nancy took over the keyboard and added a new `geom_jitter()` layer to the `ggplot()`. She also used the `alpha = .8` option with the boxplots to make the color a little less bright so that it was easier to see the data in Figure 3.53.

```
# Add points to boxplots (Figure 3.53)
box.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number),
             y = number)) +
  geom_boxplot(aes(fill = weapons), alpha = .8) +
  geom_jitter() +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_fill_manual(values = c("Handguns" = "#7463AC",
                               "Firearms, type not stated" = "gray",
                               "Rifles" = "gray",
                               "Shotguns" = "gray",
                               "Other guns" = "gray"), guide=FALSE)
box.homicide.gun
```

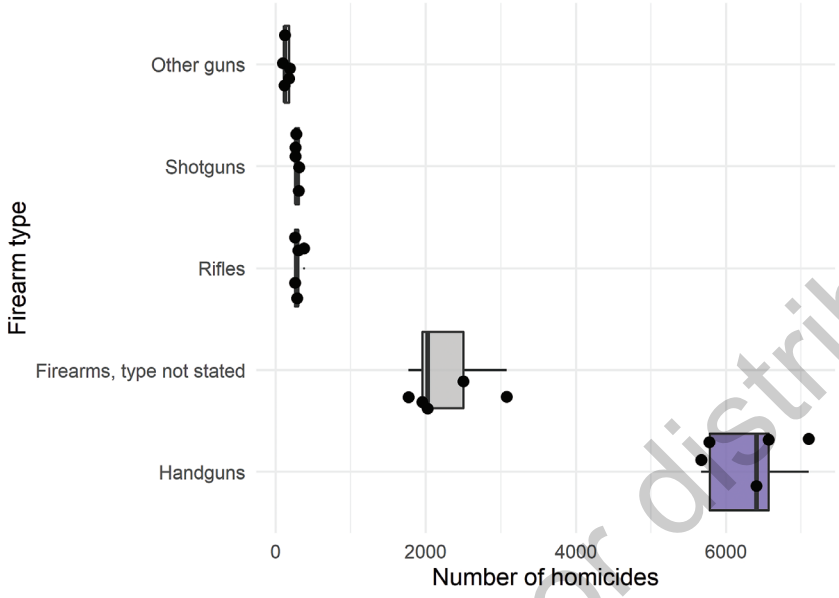
Leslie loved the addition of the data to the plot! Although this data set was very small, she could imagine how putting the points with the boxes for a larger data set would be very useful for seeing how well the boxes were capturing the distribution of the underlying data.

Kiara looked over the code and asked about the `geom_jitter()` layer. She asked why Nancy hadn't just used `geom_point()` like they had before when they wanted to see points on a graph. Nancy explained that `geom_jitter()` is a shortcut code for `geom_point(position = "jitter")`. She added that both of these would do the same thing, which is place the points on the graph, but add some "jitter" so that they are not all along a straight line. Having points all along a line makes it difficult to see patterns, especially in large data sets where many of the data points may be overlapping. Kiara was satisfied and did not think they needed to add more documentation.

3.6.3.5 VIOLIN PLOTS

Nancy described violin plots as somewhere between boxplots and density plots, typically used to look at the distribution of continuous data within categories. Leslie copied the code from above, removing the `geom_jitter` and changing `geom_boxplot` to `geom_violin` (Figure 3.54).

FIGURE 3.53 Annual homicides by firearm type in the United States, 2012–2016




```
# violin plot (Figure 3.54)
violin.homicide.gun <- fbi.deaths.cleaned %>%
  ggplot(aes(x = reorder(x = weapons, X = -number),
             y = number)) +
  geom_violin(aes(fill = weapons)) +
  theme_minimal() +
  labs(x = "Firearm type", y = "Number of homicides") +
  coord_flip() +
  scale_fill_manual(values = c('gray', "#7463AC", 'gray',
                               'gray', 'gray'), guide=FALSE)
violin.homicide.gun
```

Leslie, Nancy, and Kiara agreed that this graph type did not work for these data. Kiara suggested that this was because, as they learned from the other plots above, there were too few cases per group for some graphs to be appropriate. Nancy still wanted to illustrate to Leslie the utility of violin plots because in many scenarios, they are useful. She wrote some quick code using the `nhanes.2012.clean` data from above to look at whether the distributions of age were the same for males and females (Figure 3.55).

```
# violin plot of age by sex for NHANES (Figure 3.55)
nhanes.2012.clean %>%
  ggplot(aes(x = sex, y = RIDAGEYR)) +
  geom_violin(aes(fill = sex)) +
```

```
    scale_fill_manual(values = c("gray", "#7463AC"), guide = FALSE) +
    labs(y = "Age in years", x = "Sex") +
    theme_minimal()
```

FIGURE 3.54 Annual homicides by firearm type in the United States, 2012–2016

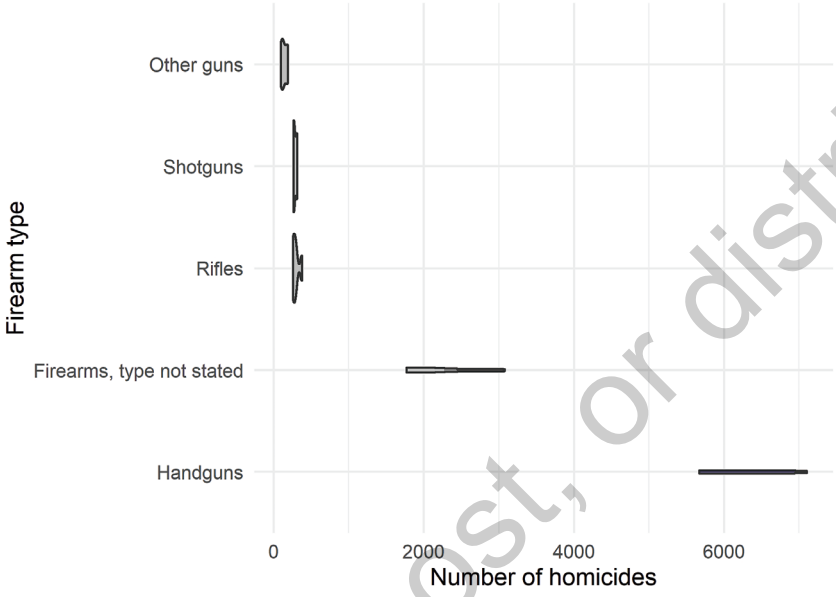


FIGURE 3.55 Distribution of age by sex for participants in the 2011–2012 NHANES survey

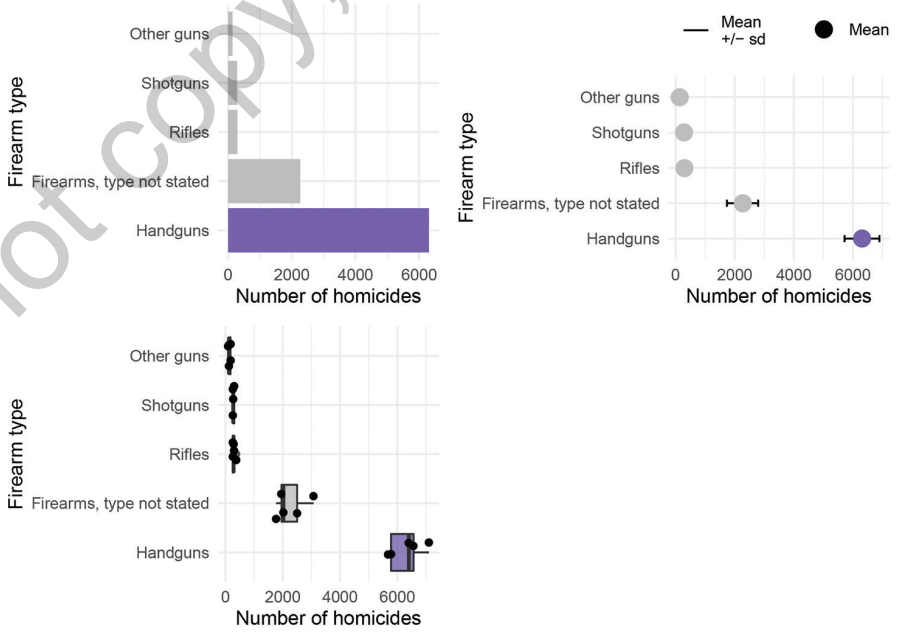
Although there didn't seem to be much of a difference in the distribution of ages between males and females, Leslie saw how she could tell the distribution by group from this plot, which might be useful for future projects. She was also amused that the shapes looked like beakers from a chemistry lab.

Before they moved to the last section, Leslie wanted to look at the different graph types of the weapons used in homicide one last time. The violin plots didn't work well, but the other three were good, so she copied her code from the previous section and added the new graph names (Figure 3.56).

```
# plot all three options together (Figure 3.56)
gridExtra::grid.arrange(bar.homicide.gun,
                        point.homicide.gun,
                        box.homicide.gun,
                        ncol = 2)
```

Leslie thought the purple bar in the bar chart stood out the most, probably because it was a lot of color in one place. This might be a good choice if the goal was to clearly and quickly communicate how big the mean is for the `Handgun` group compared to all the other means. Both the point chart and the boxplot were better at showing spread in addition to the central tendency. The boxplot gave the most information about the actual data underlying the plot. Leslie reminded them that whichever graph they chose, the median and IQR were better to show than the mean and standard deviation, given the skew they could see in the boxplot. Of all these graphs, she would choose the boxplot. Nancy and Kiara agreed.

FIGURE 3.56 Graph types for one factor and one numeric variable



3.6.4 LINE GRAPHS AND SCATTERPLOTS FOR TWO CONTINUOUS VARIABLES

Now it was time to see what was useful for examining the relationship between two numeric variables. Nancy looked through the data they had discussed so far and found the `Number of handguns produced` and the `Year` variables. She explained that the production of handguns over time could be examined using a scatterplot or a line graph. These two types of graphs are useful for examining the relationship between two numeric variables that have values that are along a continuum, whether they are truly continuous or just close to continuous. The number of handguns produced is most like a continuous variable because it spans a continuum from zero to some upper limit. The year of production might be considered continuous if the underlying idea is to examine how things changed over a continuous measure of time. In other cases, the year might be considered a categorical idea rather than continuous, with each year treated as a category.

```
# bring in the data
guns.manu <- read.csv(file = "[data folder location]/data/total_firearms_
manufactured_US_1990to2015.csv")
summary(object = guns.manu)
##       Year          Pistols           Revolvers        Rifles
##  Min.   :1990   Min.   : 677434   Min.   :274399   Min.   : 883482
##  1st Qu.:1996   1st Qu.: 989508   1st Qu.:338616   1st Qu.:1321474
##  Median :2002   Median :1297072   Median :464440   Median :1470890
##  Mean   :2002   Mean   :1693216   Mean   :476020   Mean   :1796195
##  3rd Qu.:2009   3rd Qu.:2071096   3rd Qu.:561637   3rd Qu.:1810749
##  Max.   :2015   Max.   :4441726   Max.   :885259   Max.   :3979568
##     Shotguns       Total.firearms
##  Min.   : 630663   Min.   : 2962002
##  1st Qu.: 735563   1st Qu.: 3585090
##  Median : 859186   Median : 3958740
##  Mean   : 883511   Mean   : 4848942
##  3rd Qu.:1000906   3rd Qu.: 5300686
##  Max.   :1254924   Max.   :10349648
```

Nancy looked at the data and noticed that each firearm type was included as a different variable. Instead of this, she thought that gun type should be one factor variable with each type of gun as a category of the factor. This is another case of wide data that should be long. Nancy looked back at her code for making wide data long and applied the same code here along with a line of code to ensure that the new `gun.type` variable was the factor data type.

```
# make wide data long
guns.manu.cleaned <- guns.manu %>%
  gather(key = gun.type, value = num.guns, Pistols,
         Revolvers, Rifles, Shotguns, Total.firearms) %>%
  mutate(gun.type = as.factor(gun.type))
```

```
# check the data
summary(object = guns.manu.cleaned)
##       Year                gun.type     num.guns
##  Min.   :1990   Pistols      :26   Min.   :  274399
##  1st Qu.:1996   Revolvers    :26   1st Qu.:  741792
##  Median :2002   Rifles       :26   Median : 1199178
##  Mean   :2002   Shotguns     :26   Mean   : 1939577
##  3rd Qu.:2009   Total.firearms:26   3rd Qu.: 3119839
##  Max.   :2015                      Max.   :10349648
```

3.6.4.1 LINE GRAPHS

Once the data were formatted, Nancy hurried on to the graphing. She started by piping the new data frame into the `ggplot()` function with `geom_line()` to create a ***line graph***. To reproduce the line graph in Figure 3.5, Nancy used a different line for each gun type by adding `linetype = gun.type` to the `aes()` (Figure 3.57).

```
# plot it (Figure 3.57)
line.gun.manu <- guns.manu.cleaned %>%
  ggplot(aes(x = Year, y = num.guns)) +
  geom_line(aes(linetype = gun.type))
line.gun.manu
```

FIGURE 3.57 Firearms manufactured in the United States over time (1990–2015)

The graph was a good start, but Nancy was not satisfied with it. She made herself a list of the things to change:

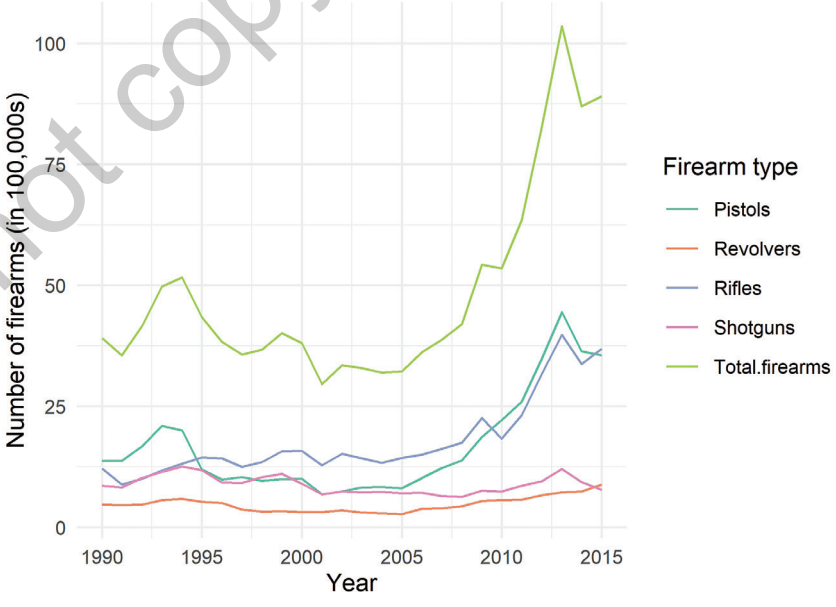
- Convert the scientific notation on the *y*-axis to regular numbers
- Add a theme to get rid of the gray background
- Make better labels for the axes and legend
- Add color to the lines to help differentiate between gun types

```
# update the y-axis, theme, line color, labels (Figure 3.58)
line.gun.manu <- guns.manu.cleaned %>%
  ggplot(aes(x = Year, y = num.guns/100000)) +
  geom_line(aes(color = gun.type)) +
  theme_minimal() +
  labs(y = "Number of firearms (in 100,000s)") +
  scale_color_brewer(palette = "Set2", name = "Firearm type")
line.gun.manu
```

Kiara suggested that more formatting options could be changed to reproduce Figure 3.5 exactly, but Figure 3.58 was actually easier to read. She wondered if Nancy knew a way to make the lines thicker so they were easier to tell apart.

Leslie was still interested in handguns after learning how many more were used in homicides. Pistols and revolvers are both types of handguns, so to see more clearly whether the number of handguns has increased, she asked if Nancy knew an easy way to sum these two values for each year. Nancy said this could be done in the original data set before creating the long data set used to graph. Nancy was delighted to show off yet another of her code skills and wrote some code for Figure 3.59.

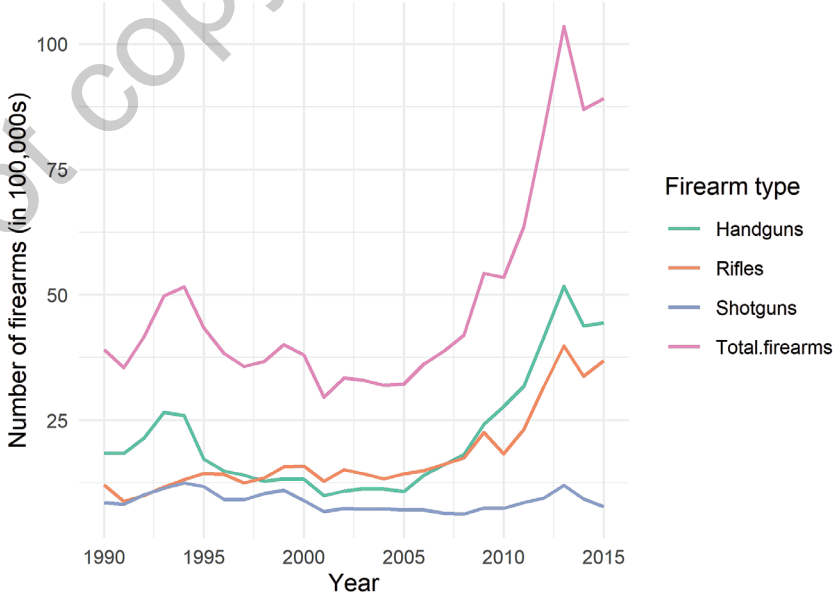
FIGURE 3.58 Firearms manufactured in the United States over time (1990–2015)



```
# make a handguns category that is pistols + revolvers
# remove pistols and revolvers from graph
guns.manu.cleaned <- guns.manu %>%
  mutate(Handguns = Pistols + Revolvers) %>%
  gather(key = gun.type, value = num.guns, Pistols, Revolvers,
         Rifles, Shotguns, Total.firearms, Handguns) %>%
  mutate(gun.type, gun.type = as.factor(gun.type)) %>%
  filter(gun.type != "Pistols" & gun.type != "Revolvers")

# update the line graph with new data and thicker lines (Figure 3.59)
line.gun.manu <- guns.manu.cleaned %>%
  ggplot(aes(x = Year, y = num.guns/100000)) +
  geom_line(aes(color = gun.type), size = 1) +
  theme_minimal() +
  labs(y = "Number of firearms (in 100,000s)") +
  scale_color_brewer(palette = "Set2", name = "Firearm type")
line.gun.manu
```

The graph suggested the number of handguns manufactured increased steadily from 2005 to 2013, and handguns were the most manufactured type of gun from 2009 to 2015. The team was happy with this graph and found it easier to read than Figure 3.5, so they moved on to the next graph type for two numeric variables.

FIGURE 3.59 Firearms manufactured in the United States over time (1990–2015)



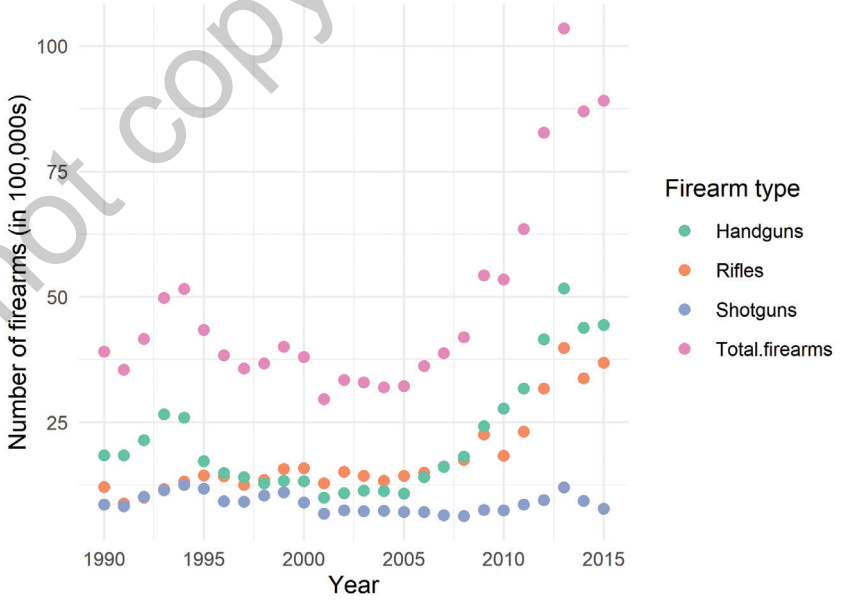
3.6.4.2 SCATTERPLOTS

Nancy explained that a scatterplot is also useful to show the relationship between two continuous variables. In a scatterplot, instead of connecting data points to form a line, one dot is used to represent each data point. Leslie had recently learned about the situations where a line graph was more useful than a scatterplot: (a) when the graph is showing change over time, and (b) when there is not a lot of variation in the data. Relationships where there is no measure of time and data that include a lot of variation are better shown with a scatterplot. Leslie slid the laptop away from Nancy while she still could and started to work on the code. Nancy suggested that they try their usual strategy of changing the `geom_line()` layer to `geom_point()` to see how a scatterplot would work for the graph they just built (Figure 3.60).

```
# use scatterplot instead of line (Figure 3.60)
scatter.gun.manu <- guns.manu.cleaned %>%
  ggplot(aes(x = Year, y = num.guns/100000)) +
  geom_point(aes(color = gun.type)) +
  theme_minimal() +
  labs(y = "Number of firearms (in 100,000s)") +
  scale_color_brewer(palette = "Set2", name = "Firearm type")
scatter.gun.manu
```

The three of them looked at the graph and rolled their eyes. It appeared that the guidance Leslie had received was correct; data over time are better shown with a line graph than a scatterplot. Leslie thought about the graphs they had been examining and remembered Figure 3.1 and Figure 3.2.

FIGURE 3.60 Firearms manufactured in the United States over time (1990–2015)



These graphs showed the amount of funding for research for the top 30 causes of death. Both included a lot of variation, and the information they conveyed was clear with the scatterplot. Kiara checked the `research.funding` data frame and wrote some code using `ggplot()` with a `geom_point()` layer to show the variation in funding by cause of death (Figure 3.61).

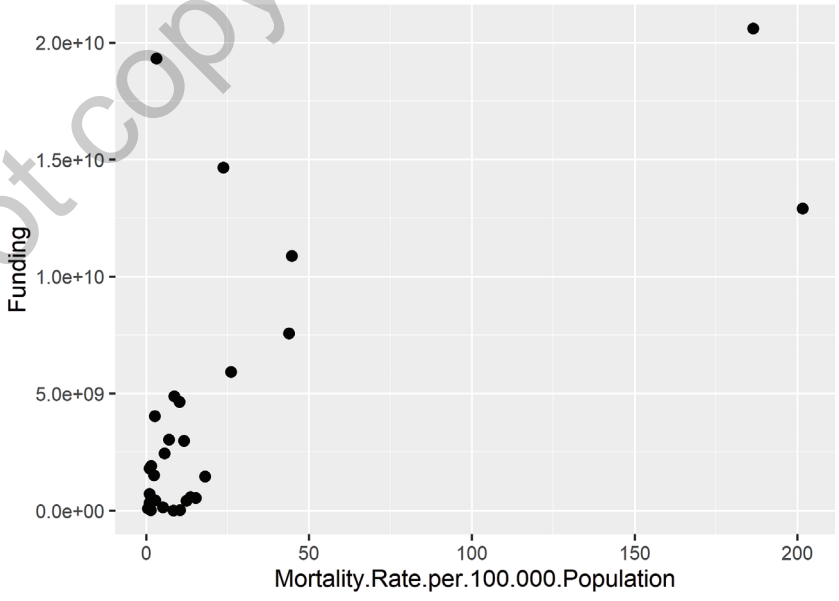
```
# scatterplot of gun research by funding (Figure 3.61)
scatter.gun.funding <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population, y = Funding))+
    geom_point()
scatter.gun.funding
```

Leslie was surprised that Figure 3.61 did not look at all like Figure 3.1. She was curious about changing this graph to a line graph where the dots would be connected instead of separate dots.

```
# Line graph of gun research by funding (Figure 3.62)
line.gun.funding <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population, y = Funding))+
    geom_line()
line.gun.funding
```

Figure 3.62 looked even worse. Clearly a scatterplot was a better idea, but Leslie wondered why the first graph looked so different from the original Figure 3.1. In the interest of reproducibility, Kiara took a

FIGURE 3.61 Research funding for the top 30 causes of mortality in the United States



closer look at the Figure 3.1 graph (copied to Figure 3.63). She noticed that the *x*- and *y*-axes in the original figure did not have even spacing between numbers.

There was a large distance between 1 and 10, but the distance between 10 and 100 was about the same even though this should be nine times as far. Leslie thought it looked like a variable transformation. That is, the values of the variable had been transformed by adding, multiplying, or performing some

FIGURE 3.62 Research funding for the top 30 causes of mortality in the United States

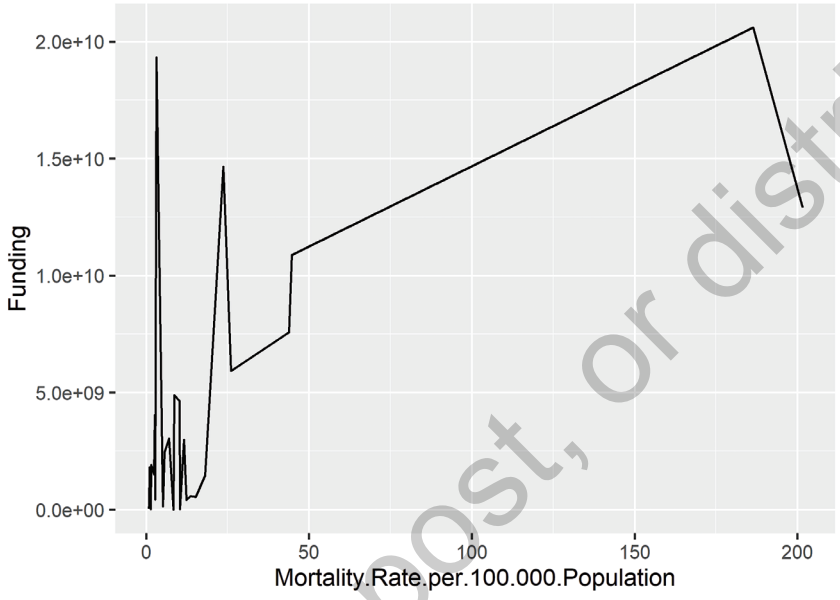
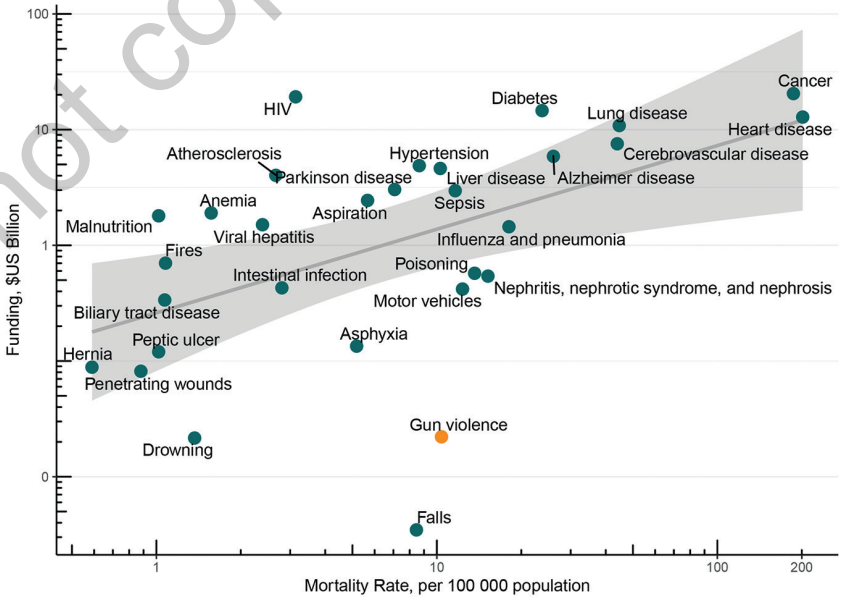


FIGURE 3.63 Reproduced figure showing mortality rate versus funding from 2004 to 2015 for the 30 leading causes of death in the United States

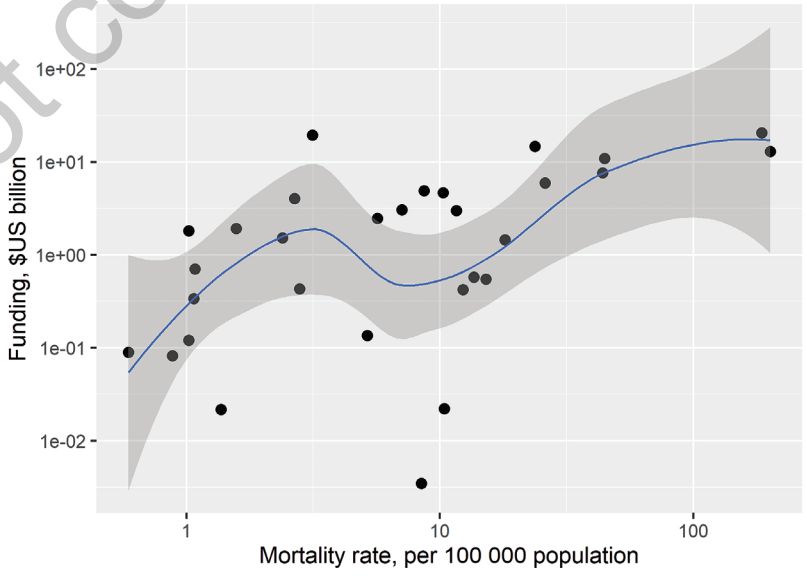


other mathematical operation. Kiara looked in the methods section of the paper that reported the graph and found this sentence: "The predictor and outcomes were log-transformed" (Stark & Shah, 2017, p. 84). As soon as they found this information, Nancy was ready to code it! Leslie wanted to decide what would be best, though. Nancy had an idea: Since they were not doing any analyses with the transformed variable, she could just use a log scale for the axes of the figure. Kiara and Leslie agreed and slid the laptop to Nancy to code it.

Leslie reminded Nancy that they might also want to add a ***trend line*** to provide an additional visual cue about the relationship between the variables. For example, while connecting all the dots with a line was not useful, a line showing the general relationship between cause of mortality and research funding could help clarify the relationship between the two variables. Nancy had just the trick. She would add layers for scaling with `scale_x_log10()` and `scale_y_log10()` for the axes and a layer with `stat_smooth()` for a smooth line through the dots. Nancy decided to reproduce the labels for the *x*- and *y*-axes from the original as well while she was working on the code. The *y*-axis layer appears to be in billions, so the funding variable should be divided by billions to make this true (Figure 3.64).

```
# scatterplot of gun research by funding (Figure 3.64)
# with axes showing a natural log scale
scatter.gun.funding <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population,
             y = Funding/1000000000))+
  geom_point() +
  stat_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  labs(y = "Funding, $US billion",
       x = "Mortality rate, per 100 000 population")
scatter.gun.funding
```

FIGURE 3.64 Research funding for the top 30 causes of mortality in the United States



That line does not look right, thought Nancy. She had forgotten to use the `method = lm` or linear model option to add a straight line with the `stat_smooth()` function. She added this and used `theme_minimal()` to get rid of the gray background (Figure 3.65).

```
#scatterplot of gun research by funding (Figure 3.65)
#with axes showing a natural log scale
scatter.gun.funding <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population,
             y = Funding/1000000000))+
  geom_point() +
  stat_smooth(method = "lm") +
  scale_x_log10() +
  scale_y_log10() +
  labs(y = "Funding, $US billion",
       x = "Mortality rate, per 100 000 population") +
  theme_minimal()
scatter.gun.funding
```

Nancy showed off one last `ggplot()` skill with some additional options to label the points, highlight the point representing gun violence, and make the formatting better match the original (Figure 3.66).

```
# fancy graph (Figure 3.66)
scatter.gun.funding.lab <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population,
             y = Funding/1000000000))+
  geom_point() +
  stat_smooth(method = "lm") +
  scale_x_log10() +
  scale_y_log10() +
  labs(y = "Funding, $US billion",
       x = "Mortality rate, per 100 000 population") +
  theme_minimal() +
  geom_text(aes(label = Cause.of.Death))
scatter.gun.funding.lab
```

This was pretty close to done, thought Leslie. But Nancy noticed that the *y*-axis was still in scientific notation, and some of the labels were overlapping and cut off. She did a little research to see if she could fix these things and came up with a new package to use to prevent label overlapping, **`ggrepel`**, and an idea for fixing the axes to show nonscientific notation by adding in the exact numbers for each axis (Figure 3.67). To fix the labels for Figure 3.66, Nancy also used `library(package = "scales")` to open the `scales` package before running the Figure code.

FIGURE 3.65 Research funding for the top 30 causes of mortality in the United States

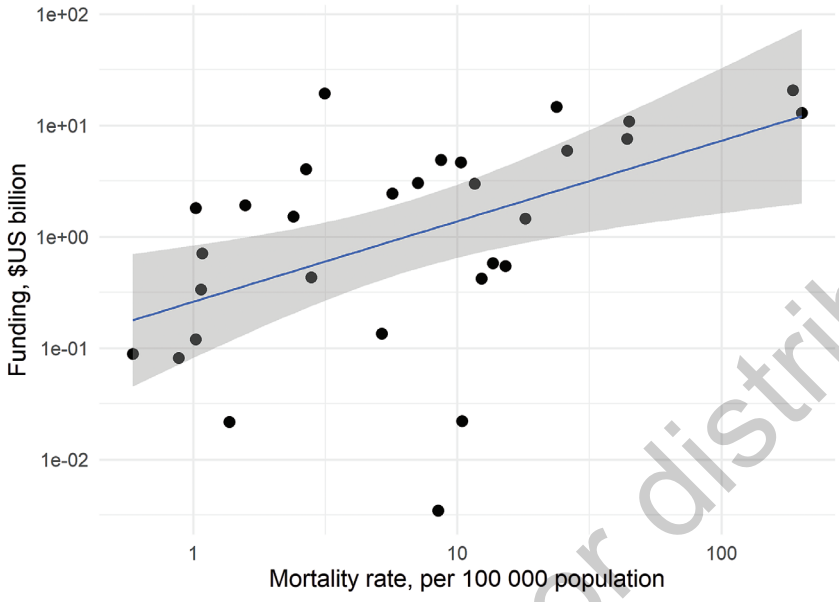
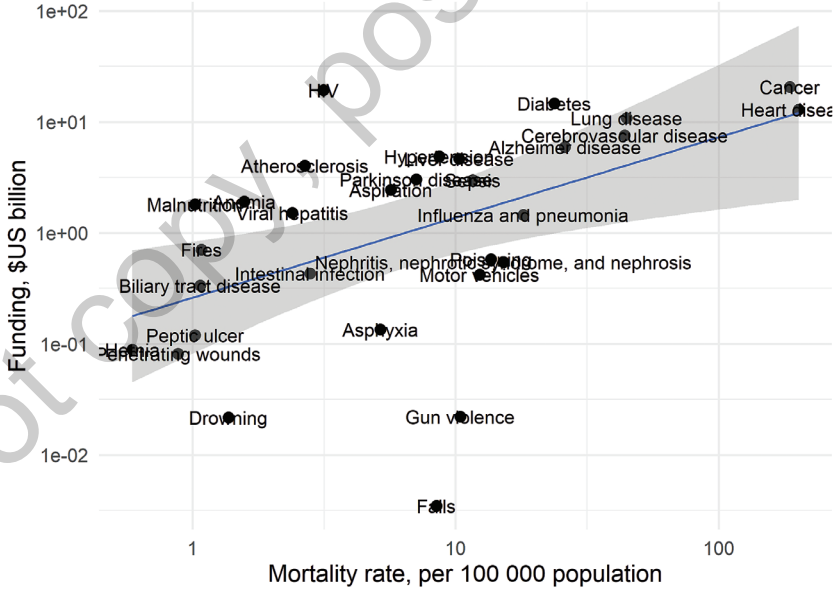


FIGURE 3.66 Research funding for the top 30 causes of mortality in the United States




```
# fancy graph with better labels (Figure 3.67)
scatter.gun.funding.lab <- research.funding %>%
  ggplot(aes(x = Mortality.Rate.per.100.000.Population,
             y = Funding/1000000000)) +
```

```
  geom_point() +
  stat_smooth(method = "lm") +
  scale_x_log10(breaks = c(1,10,100), labels = comma) +
  scale_y_log10(breaks = c(1,10,100), labels = comma) +
  labs(y = "Funding, $US billion",
       x = "Mortality rate, per 100 000 population") +
  theme_minimal() +
  ggrepel::geom_text_repel(aes(label = Cause.of.Death), size = 3.5)
scatter.gun.funding.lab
```

It might not be perfect, but the team thought Figure 3.67 was good enough. The final scatterplot pretty clearly showed the relationship between funding and mortality rate, with some outliers like falls, gun violence, and HIV. Kiara put together the graph options for two numeric variables in Figure 3.68.

```
# show graph types (Figure 3.68)
gridExtra::grid.arrange(line.gun.manu,
                        scatter.gun.funding,
                        nrow = 2)
```

The type of graph clearly had to match the data when working with two numeric variables. Line graphs were useful to show change over time or to graph data with little variation. Scatterplots were better

FIGURE 3.67 Research funding for the top 30 causes of mortality in the United States

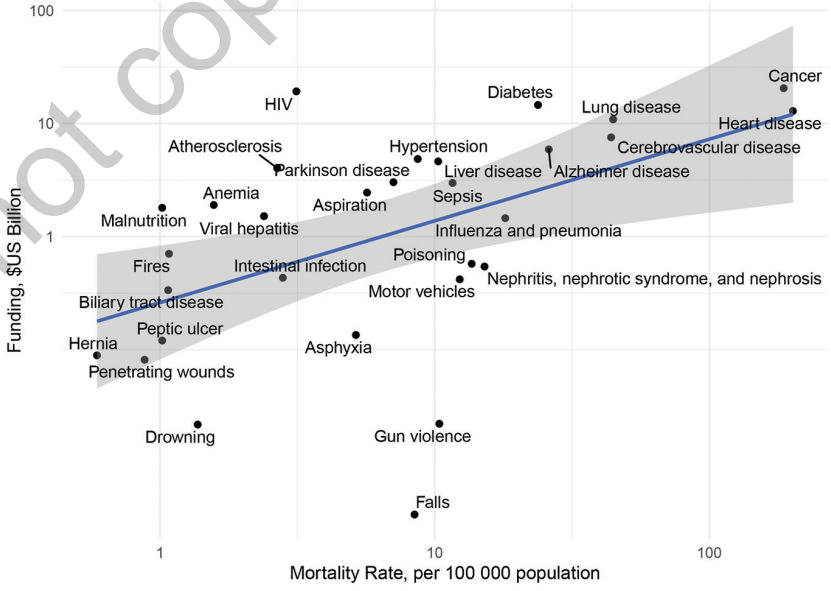


FIGURE 3.68 Graph types for two continuous or numeric variables

for when there was some variation in the relationship between the two variables. Nancy thought they were done with graphs, but Kiara had one more thing she wanted to discuss. Because graphs are such an important part of communicating data, it is extremely important that they are well-formatted. Formatting graphs well is also key for making sure your work is reproducible.

3.6.5 ACHIEVEMENT 3: CHECK YOUR UNDERSTANDING

Plot an appropriate graph to show the relationship between marital status (`DMDMARTL`) and sex (`RIAGENDR`) in the NHANES data. Explain why you chose this graph and not the others available.

3.7 Achievement 4: Ensuring graphs are well-formatted with appropriate and clear titles, labels, colors, and other features

In addition to choosing an appropriate graph, Kiara reminded Leslie that the primary goal was to be clear and for the graph to stand alone without relying on additional text to explain it. For a graph to stand alone, it should have as many of these features as possible:

- Clear labels and titles on both axes
- An overall title describing what is in the graph along with
  - Date of data collection
  - Units of analysis (e.g., people, organizations)
  - Sample size

In addition, researchers often use the following to improve a graph:

- Scale variables with very large or very small values (e.g., using millions or billions)
- Color to draw attention to important or relevant features of a graph

Leslie reviewed the graphs they had created so far and saw that Nancy and Kiara had been demonstrating these concepts and that most of the graphs they worked on had most of these features. The R-Team was once again exhausted from so much R and was ready to call it a day.

Leslie thought about everything they had learned through graphing. They had discovered that a higher percentage of males use guns than females, and that median age was nearly the same for gun users compared to nonusers. They had also learned that handguns were responsible for homicides at a far greater rate than were other types of guns, and that more handguns were manufactured than other types of guns. They confirmed through reproducing a published graph that funding for gun violence research was the third lowest of the top 30 mortality causes.

Leslie was so eager to text Leanne and share the graphs that she almost forgot to say goodbye to Nancy and Kiara. She thought the information they gained by exploring the data with graphs might be useful for Leanne's work with Moms Demand Action.

When she turned around to wave, she saw that Nancy was lost in her phone checking emails and had nearly collided with a large group of students who were just walking into the business school.

She waved to Nancy and looked around for Kiara. She saw Kiara across far ahead, no phone in sight and almost to the parking garage. "See you later!" Leslie yelled.

Kiara turned around and waved. "Looking forward to it!" she yelled back.

/// 3.8 CHAPTER SUMMARY

3.8.1 Achievements unlocked in this chapter: Recap

Congratulations! Like Leslie, you've learned and practiced the following in this chapter.

3.8.1.1 Achievement 1 recap: Choosing and creating graphs for a single categorical variable

Bar charts and waffle charts are the best options to plot a single categorical variable. Even if it makes the R-Team hungry just thinking about it, waffle charts are a better option than pie charts for showing parts of a whole.

3.8.1.2 Achievement 2 recap: Choosing and creating graphs for a single continuous variable

Histograms, density plots, and boxplots demonstrate the distribution of a single continuous variable. It is easier to see skew in histograms and density plots, but central tendency is easier to identify in boxplots.

3.8.1.3 Achievement 3 recap: Choosing and creating graphs for two variables at once

For two categorical variables, a mosaic plot or a bar chart with grouped or stacked bars are recommended. For one categorical and one continuous variable, boxplots are a good choice and the two types of bar charts work well. To examine distribution across groups, grouped histograms and density plots (and violin plots) can also be used. Line graphs and scatterplots are useful for two continuous variables. Line graphs are good for graphing change over time and for when there is little variability in the data; scatterplots are better for data with a lot of variability.

3.8.1.4 Achievement 4 recap: Ensuring graphs are well-formatted with appropriate and clear titles, labels, colors, and other features

Graphs should be able to stand alone. They should include clear labels and titles on both axes and an overall title that includes date of data collection, units of analysis, and sample size. In addition, researchers could scale variables with very large or very small values (e.g., using millions or billions) and use color to draw attention to important or relevant features of a graph.

3.8.2 Chapter exercises

The coder and hacker exercises are an opportunity to apply the skills from this chapter to a new scenario or a new data set. The coder edition evaluates the application of the concepts and functions learned in this R-Team meeting to scenarios similar to those in the meeting. The hacker edition evaluates the use of the concepts and functions from this R-Team meeting in new scenarios, often going a step beyond what was explicitly explained.

The coder edition might be best for those who found some or all of the Check Your Understanding activities to be challenging or if they needed review before picking the correct responses to the multiple-choice questions. The hacker edition might be best if the Check Your Understanding activities were not too challenging and the multiple-choice questions were a breeze.

The multiple-choice questions and materials for the exercises are online at **edge.sagepub.com/harris1e**.

Q1: Which of the following is appropriate to graph a single categorical variable?

a. Histogram
b. Bar chart
c. Boxplot
d. Scatterplot

Q2: Which of the following is appropriate to graph a single continuous variable?

a. Waffle chart
b. Histogram
c. Bar chart
d. Pie chart

Q3: A mosaic plot is used when graphing

a. the relationship between two continuous variables.
b. the relationship between one continuous and one categorical variable.
c. the relationship between two categorical variables.
d. data that are not normally distributed by group.

Q4: Which of the following is not a recommended type of graph:

a. Pie chart
b. Bar chart
c. Waffle chart
d. Density plot

Q5: Density plots, histograms, and boxplots can all be used to

a. examine frequencies in categories of a factor.
b. examine the relationship between two categorical variables.
c. determine whether two continuous variables are related.
d. examine the distribution of a continuous variable.

3.8.2.1 Chapter exercises: Coder edition

Use the NHANES data to examine gun use in the United States. Spend a few minutes looking through the NHANES website before you begin. Create well-formatted, appropriate graphs using the NHANES 2011–2012 data (available at **edge.sagepub.com/harris1e** or by following the instructions in Box 3.1) examining each of the variables listed below. Be sure to code missing values appropriately.

1) Income (Achievements 2 and 4)
2) Marital status (Achievements 1 and 4)
3) Race (Achievements 1 and 4)
4) Income and gun use (Achievements 3 and 4)
5) Race and gun use (Achievements 3 and 4)
6) Marital status and gun use (Achievements 3 and 4)

Use good code-formatting practices, and include a sentence after each graph in the comments that explains what the graph shows.

3.8.2.2 Chapter exercises: Hacker edition

Complete the coder edition with the following additional features:

- Format Graphs 3 and 5 to use color to highlight the group with the highest gun usage.

- In the percentage graph for Graph 1, highlight the highest percentage group with a different color.

3.8.2.3 Chapter exercises: Ultra hacker edition

Use the FBI data to re-create Figure 3.3.

3.8.2.4 Instructor note

Solutions to exercises can be found on the website for this book, along with ideas for gamification for those who want to take it further.

SAGE edge™

Visit **edge.sagepub.com/harris1e** to download the datasets, complete the chapter exercises, and watch R tutorial videos.