Exploring and Transforming Data 3

Statistical graphs play three important roles in data analysis. Graphs provide an initial look at the data, a step that is skipped at the peril of the data analyst. At this stage, we learn about the data, its oddities, outliers, and other interesting features. John Tukey (Tukey, 1977) coined the term *exploratory data analysis* for this phase of an analysis. Graphs are also employed during model building and model criticism, particularly in diagnostic methods used to understand the fit of a model. Finally, *presentation graphics* can summarize a fitted model for the benefit of others.

In the first two applications, we need to be able to draw many graphs quickly and easily, while in the presentation phase we should be willing to spend more time on a graph to get it just right for publication. In this chapter, we present some basic tools for exploratory graphs, such as histograms, boxplots, and scatterplots. Some of these tools are standard to R, while others are in the **car** package associated with this book. We will return to regression graphics in Chapter 6, with equally easy to use functions for various diagnostic methods, which differ from the basic graphs of this chapter mostly in the quantities that are graphed, not in the graphing paradigm. Finally, in Chapter 7 we show how to produce customized, potentially elaborate, graphs suitable for almost any purpose.

3.1 Examining Distributions

3.1.1 HISTOGRAMS

The most common graph of the distribution of a quantitative variable is the *histogram*. A histogram dissects the range of the variable into class intervals, called *bins*, and counts the number of observations falling in each bin. The counts—or percentages, proportions, or densities calculated from the

This document is authorized for use by **Kevin Hartman**, from 4/23/2019 to 8/10/2019, in the course:
MIDS DATASCI W203: Statistics and Data Science - Laskowski, Kappedal, Penner (Summer 2019), University of California, Berkeley.
Any unauthorized use or reproduction of this document is strictly prohibited*.

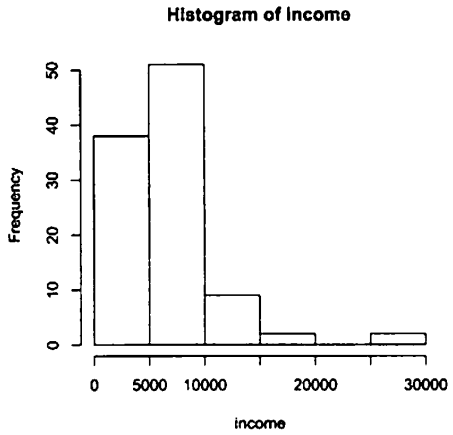

Figure 3.1 Default histogram of `income` in the Canadian occupational-prestige data.

counts—are plotted in a bar graph. An example, constructed by the following R commands, appears in Figure 3.1:[1]

```
> library(car)
> head(Prestige) # first 6 rows

                    education income women prestige census type
gov.administrators      13.11  12351 11.16     68.8   1113 prof
general.managers        12.26  25879  4.02     69.1   1130 prof
accountants             12.77   9271 15.70     63.4   1171 prof
purchasing.officers     11.42   8865  9.11     56.8   1175 prof
chemists                14.62   8403 11.68     73.5   2111 prof
physicists              15.64  11030  5.13     77.6   2113 prof

> with(Prestige, hist(income))
```

The first of these commands loads the **car** package, giving us access to the `Prestige` data. The second command displays the initial six lines of the data set. The histogram is drawn by the `hist` function, in this case with no arguments other than the variable to be plotted, `income`. The `with` command allows `hist` to access `income` from the `Prestige` data frame (as explained in Section 2.2.2).

The default histogram, produced by `hist` with no extra arguments, has bins of equal width, and the height of each bar is equal to the *frequency*—the number of observations—in the corresponding bin. In an alternative definition of the histogram, the height of each bar is selected so that its *area* is equal to the fraction of the data in the corresponding bin. To distinguish it from the more common frequency histogram, we call this latter graph a *density histogram*. The `hist` function draws density histograms if the argument `freq`

[1] The Canadian occupational-prestige data set, on which this example is based, was introduced in Section 2.1.2.

This document is authorized for use by **Kevin Hartman**, from 4/23/2019 to 8/10/2019, in the course:
MIDS DATASCI W203: Statistics and Data Science - Laskowski, Kappedal, Penner (Summer 2019), University of California, Berkeley.
Any unauthorized use or reproduction of this document is strictly prohibited*.

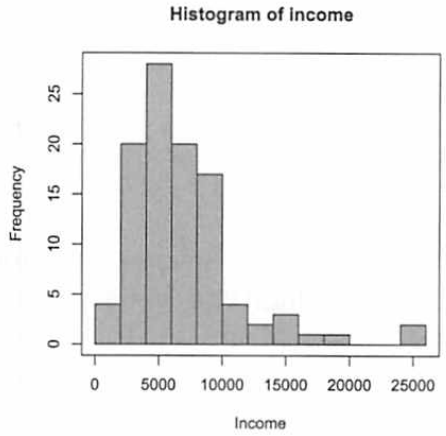

Figure 3.2 Revised histogram of `income`.

is set to `FALSE` or if the `breaks` argument is used to define bins of unequal width.

The shape of the histogram is determined in part by the number of bins—too few and the plot hides interesting features of the data, too many and the histogram is too rough, displaying spurious features of the data. The default method for selecting the number of bins, together with the effort to locate nice cut points between the bins, can produce too few bins. An alternative rule, proposed by Freedman and Diaconis (1981), sets the target number of bins to

$$\left\lceil \frac{n^{1/3}(\max - \min)}{2(Q_3 - Q_1)} \right\rceil$$

where n is the number of observations, $\max - \min$ is the range of the data, $Q_3 - Q_1$ is the interquartile range, and the ceiling brackets indicate rounding up to the next integer. Applying this rule to `income` in the Canadian occupational-prestige data produces the histogram in Figure 3.2:

```
> with(Prestige, hist(income, breaks="FD", col="gray"))
> box()
```

Setting `col="gray"` specifies the color of the histogram bars.[2] The `box` function draws a box around the histogram and could have been omitted. In this example, both histograms suggest that the distribution of income has a single mode near $5,000 and is skewed to the right, with several occupations that have relatively large incomes.

As with most of the graphics functions in R, `hist` has a dizzying array of arguments that can change the appearance of the graph:

[2]The general use of color in R is discussed in Section 7.1.4.

This document is authorized for use by **Kevin Hartman**, from 4/23/2019 to 8/10/2019, in the course:
MIDS DATASCI W203: Statistics and Data Science - Laskowski, Kappedal, Penner (Summer 2019), University of California, Berkeley.
Any unauthorized use or reproduction of this document is strictly prohibited*.

```
> args(hist.default)

function (x, breaks = "Sturges", freq = NULL, probability = !freq,
 include.lowest = TRUE, right = TRUE, density = NULL, angle = 45,
 col = NULL, border = NULL, main = paste("Histogram of", xname),
 xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE,
 plot = TRUE, labels = FALSE, nclass = NULL, ...)
```

The `args` command (Section 1.1.2) displays all the arguments of a function. We asked for the arguments for `hist.default` rather than just `hist` because `hist` is a generic function and it is the default method that actually draws the graph.[3] While a more complete description is available from `?hist`, here are some of the key arguments. The `breaks` argument is used to specify the edges of the bins. We can choose these values ourselves [e.g., `breaks=c(0, 5000, 10000, 15000, 20000, 25000)`], give the number of bins we want (e.g., `breaks=10`), or set `breaks` equal to the name of a rule that will determine the number of equal-size bins (e.g., `breaks ="FD"`). The possible settings are given on the help page for the function. The `xlab`, `ylab`, and `main` arguments are used, as in most graphical functions in R, to label the horizontal axis, vertical axis, and plot title, respectively. If we don't set these arguments, then `hist` will construct labels that are often reasonable. The remaining arguments generally change the appearance of the graph.

You may be familiar with *stem-and-leaf displays*, which are histograms that encode the numeric data directly in their bars. We believe that stem-leaf-displays, as opposed to more traditional histograms, are primarily useful for what Tukey (1977) called *scratching down* numbers—that is, paper-and-pencil methods for visualizing small data sets. That said, stem-and-leaf displays may be constructed by the standard R function `stem`; a more sophisticated version, corresponding more closely to Tukey's original stem-and-leaf display, is provided by the `stem.leaf` function in the **aplpack** package.

If you are looking for fancy three-dimensional effects and other chart junk (an apt term coined by Tufte, 1983) that are often added by graphics programs to clutter up histograms and other standard graphs, you will have to look elsewhere: The basic R graphics functions intentionally avoid chart junk.

3.1.2 DENSITY ESTIMATION

Nonparametric density estimation often produces a more satisfactory representation of a distribution by smoothing the histogram. The *kernel-density estimate* at the value x of a variable X is defined as

$$\widehat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

[3] See Sections 1.4 and 8.7 on object-oriented programming in R for a detailed explanation of generic functions and their methods.

This document is authorized for use by **Kevin Hartman**, from 4/23/2019 to 8/10/2019, in the course:
MIDS DATASCI W203: Statistics and Data Science - Laskowski, Kappedal, Penner (Summer 2019), University of California, Berkeley.
Any unauthorized use or reproduction of this document is strictly prohibited*.