2.1 Data Visualization and Summary Statistics

After clearly defining the scientific question we try to answer, selecting a set of representative members from the population of interest and collecting data (either through observational studies or randomized experiments), we usually begin our analysis with data exploration. This chapter focuses on data exploration for one variable at a time. (Data exploration techniques aimed at identifying possible relationship between two or more variables are discussed in the next chapter.) Our objective is to develop a high-level understanding of the data, learn about the possible values for each characteristic, and find out how a characteristic varies among individuals in our sample. In short, we want to learn about the distribution of variables. Recall that for a variable, the distribution shows the possible values, the chance of observing those values, and how often we expect to see them in a random sample from the population.

The data exploration methods allow us to reduce the amount of information so that we can focus on the key aspects of the data. We do this by using data visualization techniques and summary statistics. The visualization techniques and summary statistics we use for a variable depend on its type. Therefore, before we continue with data exploration methods, we briefly discuss different variable types. (More discussion is provided in Chap. 4.)

2.2 Variable Types

Let us revisit the Pima.tr data discussed in the previous chapter (Fig. 2.1). For each individual, there are eight measurements for eight different variables. In this book, variables will be represented by capital letters, such as X, Y, Z. Each observation in our sample has an index $i$, where $i = 1, 2, \ldots, n$, and $n$ is the total sample size. Here, the term observation refers to an observed value of a variable, and the term sample refers to the collection of these observations. We denote by $x_i$ the $i$th
observed value of variable $X$. For example, if the variable age is denoted by $X$, then $x_5 = 23$ means that the 5th individual in our sample is 23 years old. (Try checking this by viewing the Pima.tr data set.)

Based on the values a variable can take, we can classify it into one of two groups: **numerical** variables or **categorical** variables. In Pima.tr, variables npreg, age, and bmi in the Pima.tr data set are numerical variables since they take numerical values, and the numbers they take have their usual meaning. For example, we say that the second individual in our sample is older than the first individual since $x_2 = 55$ is bigger than $x_1 = 24$. We can also subtract their ages to find their age difference: $55 - 24 = 31$. For numerical variables, we can talk about the distance between two values.

If the values of a numerical variable are *counts* (e.g., number of pregnancies, number of physician visits), we refer to the variable as a **count variable** to distinguish it from other types of numerical variables. Often, the statistical methods we choose for count variables are different from the method we choose for other numerical variables.

The **type** variable in Pima.tr is categorical since the set of values it can take consists of a finite number of categories; here, *Yes* (for diseased) and *No* (for nondiseased). In other words, a categorical variable assigns one of the possible categories to each individual in our sample.

It is common to use numerical codings for categorical variables. Let us denote the **type** variable $Y$. We can use $Y = 1$ for nondiabetic individuals (i.e., $\text{type}=\text{No}$), and $Y = 2$ for diabetic women (i.e., $\text{type}=\text{Yes}$). Note, however, that these numbers merely represent different categories (disease status) and do not have their usual meaning. For example, we cannot talk about the distance between two values of the **type** variable or say that the value of this variable for diabetic women is two times more than that of nondiabetic women. Indeed, the assignment of numbers to different categories in this case is quite arbitrary. For the **type** variable, we could have decided to represent diabetics by $Y = 1$ and nondiabetics by $Y = 2$. 
2.2 Variable Types

Categorical variables are either nominal or ordinal, depending on the extent of information the numerical coding provides. For nominal variables, the numbers are simply labels, which are chosen arbitrarily. Therefore, they do not provide any information. The type variable in Pima.tr is nominal. For ordinal variables, although the numbers do not have their usual meaning, they preserve a rank ordering. Therefore, they provide information about the ordering of categories. For example, we would use an ordinal variable to denote the severity of a disease as \( Y = 1 \) for low, \( Y = 2 \) for medium, and \( Y = 3 \) for high. Although these numerical values do not suggest that medium is two times more severe than low, we can say that medium is more severe than low.

Now let us consider another data set called birthwt, which is also available from the MASS package. This data set includes the birth weight (in grams) of 189 newborn babies along with some characteristics (e.g., age, smoking status) of their mothers. The data were collected at Baystate Medical Center, Springfield, MA, during 1986. To load this data set, click Data → Data in packages → Read data set from an attached package. Select MASS under Package and birthwt under Data set.

View the data set by clicking the View data set button (Fig. 2.2). The data set includes the following variables:

- **low**: indicator of birth weight less than 2.5 kg (0 = normal birth weight, 1 = low birth weight).
- **age**: mother’s age in years.
- **lwt**: mother’s weight in pounds at last menstrual period.
- **race**: mother’s race (1 = white, 2 = African-American, 3 = other).
- **smoke**: smoking status during pregnancy (0 = not smoking, 1 = smoking).
- **ptl**: number of previous premature labors.
- **ht**: history of hypertension (0 = no, 1 = yes).
- **ui**: presence of uterine irritability (0 = no, 1 = yes).
- **ftv**: number of physician visits during the first trimester.
- **bwt**: birth weight in grams.
Variables age, lwt, ptl, ftv, and bwt are numerical variables. Among these variables, ptl and ftv are count variables. The variables low, race, smoke, ht, and ui are all categorical. Note that all categorical variables are coded with numerical values. In these situations, R and R-Commander cannot automatically recognize them as categorical variables. In fact, they are considered as numerical variables by default. Therefore, we need to convert them to categorical variables. To do this, make sure birthwt is the active data set, then click on Data → Manage variables in active data set → Convert numeric variables to factors. (In R, categorical variables are usually stored as factors.) Under Variables, select low, race, smoke, ht, and ui. Under Factor Levels, check the Use numbers option (unless you would like to provide specific names for each category). Click OK and accept the overwrite option when prompted. The data set is now ready for exploration and analysis.

2.3 Exploring Categorical Variables

In this section, we discuss visualizing and summarizing categorical data. Consider the type variable in Pima.tr data set. A simple way for summarizing the data is to create a table that shows the number of times each category has been observed. The number of times a specific category is observed is called frequency. We denote the frequency for category \( c \) by \( n_c \).

Table 2.1 shows that in this sample, the number of women not affected by diabetes (type=No) is \( n_1 = 132 \), and the number of diabetic (type=Yes) women is \( n_2 = 68 \). Here, 1 represents “No”, and 2 represents “Yes” for the type variable. To obtain the frequencies for this variable, click Statistics → Summaries → Frequency distributions and select type as the Variable. The results are displayed in the Output window (Fig. 2.3).

The sum of the frequencies for all categories is equal to the total sample size,

\[
\sum_c n_c = n,
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>132</td>
</tr>
<tr>
<td>Yes</td>
<td>68</td>
</tr>
<tr>
<td>Total</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 2.1 Frequency table for the type variable in the Pima.tr data set
2.3 Exploring Categorical Variables

Fig. 2.3  Using R-Commander to obtain and view the frequency table for type from the Pima.tr data set

where \( \sum c \) means the sum over all categories. For the type variable, we have

\[
\sum c n_c = n_1 + n_2 = 132 + 68 = 200.
\]

2.3.1 Relative Frequency and Percentage

Follow the above steps to create the frequency table for the race variable in the birthwt data set. For this variable, the frequencies are \( n_1 = 96 \), \( n_2 = 26 \), and \( n_3 = 67 \) for “White”, “African-American”, and “Other” categories, respectively. The sum of these frequencies is equal to the sample size \( n = 189 \).

Now suppose that we want to ensure that the racial make up of our sample is similar to that of the whole US population. To do this, we use relative frequencies or percentages as summary statistics.

The relative frequency is the sample proportion for each possible category. It is obtained by dividing the frequencies \( n_c \) by the total number of observations \( n \):

\[
p_c = \frac{n_c}{n}.
\]

Relative frequencies are sometimes presented as percentages after multiplying proportions \( p_c \) by 100.

The relative frequencies and percentages for the race variable in birthwt are

\[
p_1 = \frac{96}{189} = 0.508 = 50.8%,
\]

\[
p_2 = \frac{26}{189} = 0.138 = 13.8%,
\]

\[
p_3 = \frac{67}{189} = 0.354 = 35.4%.
\]

Therefore, 50.8% (almost half) of the women in the sample were white, 13.8% were African-American, and the remaining 35.4% were from other races. We can now compare these relative frequencies with their corresponding proportions in the US population.
In R-Commander, make sure *birthwt* is the active data set, then click Statistics → Summaries → Frequency distributions, and select *race* as the Variable. The frequencies and percentages are given in the *Output* window, as shown in Fig. 2.4. Note that R-Commander automatically multiplies the proportions by 100 to obtain the percentages.

For *race*, the category “1” (i.e., white women) has the highest frequency. In this case, we say that the **mode** of the variable *race* is “1”.

For a categorical variable, the mode of is the most common value, i.e., the value with the highest frequency.

For the *type* variable, if we use 1 for “No” (i.e., nondiabetic) and 2 for “Yes” (i.e., diabetic), the mode of the variable is 1.

Since the relative frequencies are proportions of the sample size, their sum is 1,

$$ \sum_c p_c = 1, $$

where $p_c$ is the relative frequency of category $c$. For the *race* variable, we have

$$ \sum_c p_c = 0.508 + 0.138 + 0.354 = 1. $$

Similarly, the sum of the percentages for different categories is 100%. Table 2.2 shows the frequencies and relative frequencies of the three categories for *race*.

### 2.3.2 Bar Graph

For categorical variables, **bar graphs** are one of the simplest ways for visualizing the data. Using a bar graph, we can visualize the possible values (categories) a categorical variable can take, as well as the number of times each category has been
Table 2.2  Frequency table for the race variable in the birthwt data set

<table>
<thead>
<tr>
<th>Race</th>
<th>Frequency</th>
<th>Relative frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>96</td>
<td>0.508</td>
</tr>
<tr>
<td>African-American</td>
<td>26</td>
<td>0.138</td>
</tr>
<tr>
<td>Other</td>
<td>67</td>
<td>0.354</td>
</tr>
<tr>
<td>Total</td>
<td>189</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 2.5  Using R-Commander to create and view a frequency bar graph for type in the Pima.tr data set. The heights of the bars sum to the sample size $n$. Overall, bar graphs show us how the observed values of a categorical variable in our sample are distributed.

observed in our sample. The bar graph for variable type (Fig. 2.5) shows that the possible values are “No” (nondiseased) and “Yes” (diseased). The height of each bar in this graph shows the frequency of the corresponding category. Therefore, the bar heights (frequencies) add up to the total sample size (in this case, $n = 200$).

In R-Commander, make sure Pima.tr is the active data set. (If you have loaded Pima.tr, but it is not currently the active data set, click on the name of the active data set and select Pima.tr from the list of available data sets.) Then, create a bar graph for type by clicking Graphs → Bar graph and then selecting type as the Variable. (Notice how bar graphs can only be created for categorical variables.) On the resulting plot shown in Fig. 2.5, the horizontal axis represents the possible values of the variable, and the height of each bar represents the number of observations in that category. Indeed, a quick glance at the graph reveals that the number of nondiabetic women in our sample is almost two times more than the number of diabetic women. You can save this graph by clicking Graphs → Save graph to file and choosing either as bitmap or as PDF/Postscript/EPS for the file format.
Fig. 2.6  Bar graph for mother's race in the birthwt data set, where 1, 2, and 3 represent the categories “white”, “African-American”, and “other”, respectively.

Fig. 2.7  Pie charts for the type variable from Pima.tr and the race variable from birthwt, where 1, 2, and 3 represent the categories “white”, “African-American”, and “other”, respectively.

Follow the above steps to create the bar graph for the variable race in birthwt. The resulting graph is shown in Fig. 2.6.

2.3.3 Pie Chart

We can use a pie chart to visualize the relative frequencies of different categories for a categorical variable. In a pie chart, the area of a circle is divided into sectors, each representing one of the possible categories of the variable. The area of each sector $c$ is proportional to its frequency. To create pie charts in R-Commander, click Graphs → Pie chart. Figure 2.7 shows the pie charts for the type variable from Pima.tr and the race variable from birthwt.
In this section, we discuss visualization and summarization of numerical data. As a running example, we consider a numerical variable, \( X \), for which we have collected three sets (samples) of observations denoted as Sample 1, Sample 2, and Sample 3. (You can assume that each set of observations are collected from a distinct group in the population.) Figure 2.8 shows the dot plots for these three sets of observations. Here, each point represents one observation in the corresponding sample.

As before, we use data visualization techniques and summary statistics to learn about the distribution of variables. For numerical variables, we are especially interested in two key aspects of the distribution: its location and its spread. The location of a distribution refers to the central tendency of values, that is, the point around which most values are gathered. The spread of a distribution refers to the dispersion of possible values, that is, how scattered the values are around the location. In Fig. 2.8, we can see that the observed values in Sample 1 are gathered around \( X = 2 \); whereas, the observations in Sample 2 and Sample 3 are gathered around \( X = 4 \). Therefore, Sample 2 and Sample 4 have roughly the same location. On the other hand, Sample 1 and Sample 2 have roughly the same spread, which is smaller than the spread in Sample 3. The individual observations in Sample 3 tend to be further away from the location compared to those in Sample 1 and Sample 2. This might not be very clear from dot plots, where we show all the observed values. In what follows, we present more effective visualization techniques and summary statistics that reduce the amount of information in order to make it easier to learn about the distribution of numerical variables.
2.4.1 Histograms

Histograms are commonly used to visualize numerical variables. A histogram is similar to a bar graph after the values of the variable are grouped (binned) into a finite number of intervals (bins). For each interval, the bar height corresponds to the frequency (count) of observation in that interval. That is, we treat each interval as a category. Similar to bar graphs, the heights sum to sample size $n$. Figure 2.9 shows the histograms for Sample 1, Sample 2, and Sample 3. For Sample 1, observations are grouped into six intervals. Most observed values are around 2. Sample 2 and Sample 3 have roughly the same locations. However, the histogram for Sample 3 is more spread out compared to that of Sample 2.

As an example, we use the variable $bmi$ in the Pima.tr data set and create its histogram. In R-Commander, click Graphs $\rightarrow$ Histogram and select $bmi$ for the Variable. (Now we can only select from the numerical variables in our data set.) The resulting histogram is shown in Fig. 2.10. The x-axis represents $bmi$, where its observed values are divided into seven equal bins of width $w = 5$. The height of each bar shows the frequency (count) in the corresponding interval. Indeed, a quick glance of the plot suggests that the age interval $(30, 35]$ has the highest frequency. The notation $(30, 35]$ is the interval greater than 30 and less than or equal to 35. By default, each interval includes the right-hand point (here, 35) but not the left-hand point (here, 30). For the $bmi$ variable, Fig. 2.11 shows that most observations are gathered around 32.5, and the observed values spread roughly from 15
Fig. 2.10 The frequency histogram for the numerical variable \texttt{bmi} in the \texttt{Pima.tr} data set. The height of the rectangles represent the frequency of the interval and sum to the total sample size \( n \). Here, the values of the variable are divided into seven bins.

Fig. 2.11 The density histogram for \texttt{bmi} from the \texttt{Pima.tr} data set. Here, the scale on the y-axis is density (not frequency). Once again, the values of \texttt{bmi} are divided into seven bins of width \( w = 5 \) to 50. (Later, we use summary statistics to describe these features of data more precisely.) As before, you can save this graph by clicking Graphs \( \rightarrow \) Save graph to file and choosing either as bitmap or as PDF/Postscript/EPS for the file format.

In the above example, the bar height for each interval, \( c \), is equal to its frequency, \( n_c \). Alternatively, the bar height for each interval could be set to its relative frequency \( p_c = n_c/n \), or the percentage \( p_c \times 100 \), of observations that fall into that
interval. For histograms, however, it is more common to use the **density** instead of the relative frequency or percentage.

The density is the relative frequency for a unit interval. It is obtained by dividing the relative frequency by the interval width:

\[
f_c = \frac{p_c}{w_c},
\]

(2.2)

Here, \(p_c = n_c/n\) is the relative frequency with \(n_c\) as the frequency of interval \(c\) and \(n\) as the total sample size. The width of interval \(c\) is denoted \(w_c\).

Let us try calculating the density of the interval \((30, 35]\), which is the fourth interval. There are \(n_4 = 67\) observations in this interval. Therefore, the relative frequency is \(p_4 = 67/200 = 0.335\). The interval width is \(w_4 = 5\). The density for this interval is therefore

\[
f_4 = \frac{0.335}{5} = 0.067.
\]

To create the density histogram for \(bmi\) in R-Commander, click **Graphs → Histogram**, select \(bmi\) as the Variable, and choose **Densities** for the Axis Scaling. The resulting histogram (Fig. 2.11) is similar to that of Fig. 2.10. However, the height of each bar in this histogram shows the density of the corresponding interval (as opposed to its frequency).

For each interval \(c\), the area of the corresponding bar in the density histogram is calculated as follows (height \(\times\) width):

\[
a_c = f_c \times w_c \\
= \frac{p_c}{w_c} \times w_c \\
= p_c.
\]

Therefore, the area of each bar (rectangle) is the relative frequency for the corresponding interval. Since the sum of relative frequencies is 1, the total area of bars in a density histogram is 1.

**Number of Bins**  We typically use the same width, denoted as \(w\), for all bins. When creating a histogram, it is important to choose an appropriate value for \(w\). This is equivalent to choosing an appropriate number of bins. In R-Commander, by default, the number of bins is selected automatically using Sturges’ formula [32].

You can set the number of bins manually. In R-Commander, click **Graphs → Histogram**, select \(bmi\) for the Variable, and set **Number of bins** to 3. Compare the resulting histogram to Fig. 2.10.
Shapes of Histograms  

Besides the location and spread of a distribution, the shape of a histogram also shows us how the observed values spread around the location. Consider the histograms shown in Fig. 2.12. We say that this histogram is symmetric around its location (here, zero) since the densities are the same for any two intervals that are equally distant from the center. In reality, we rarely see perfectly symmetric histograms such as the one shown in Fig. 2.12. However, we usually consider a histogram as symmetric if the densities are almost the same for intervals that are equally distant from the location. For example, we can consider the histogram of bmi in Fig. 2.11 as symmetric.

In many situations, we find that a histogram is stretched to the left or right. We call such histograms skewed. More specifically, we call them left-skewed if they are stretched to the left, or right-skewed if they are stretched to the right. For instance, the histogram of Y in Fig. 2.13 is left-skewed. The majority of observations are around 102, but the decrease in densities is slower on the left of the location than on the right. This gives the histogram a long left (lower) tail. On the other hand, the histogram of variable Z in Fig 2.13 is right-skewed. The histogram is stretched to the right and has a long right (upper) tail. In the birthwt data set, the histogram of lwt (mother’s weight in pounds at last menstrual period) is right-skewed (Fig. 2.14).

The above histograms, whether symmetric or skewed, have one thing in common: they all have one peak (or mode). The overall pattern (disregarding minor details) for these histograms can be described as rising to a single peak and then declining. We call such histograms (and their corresponding distributions) unimodal. Sometimes histograms have multiple modes. For example, the histogram of variable W in Fig. 2.15 is said to be bimodal, since it has two peaks. (Here, a smooth curve has been superimposed to show the overall pattern.)
Fig. 2.13  *Left panel:* Histogram of variable $Y$ whose histogram is left-skewed. *Right panel:* Histogram of variable $Z$ whose histogram is right-skewed

Fig. 2.14 Histogram of variable `lwt` in the `birthwt` data set. The histogram is right-skewed

The bimodal histogram appears to be a combination of two unimodal histograms. Indeed, in many situations bimodal histograms (and multimodal histograms in general) indicate that the underlying population is not *homogeneous* and may include two (or more in case of multimodal histograms) subpopulations. For example, the variable $W$ in Fig. 2.15 represents blood pressure, and the sample might have been obtained from a population comprised of two groups: a healthy group, whose blood pressure is normal (around 120), and a hypertensive group, who suffer from high blood pressure (around 150).

As another example, suppose that we want to study the protein consumption of European countries [9]. Download the Protein data set from [http://lib.stat.cmu.edu/DASL/Datafiles/Protein.html](http://lib.stat.cmu.edu/DASL/Datafiles/Protein.html). In R-Commander, import the Protein data set
2.4 Exploring Numerical Variables

**Fig. 2.15** Histogram of a bimodal distribution. A smooth curve is superimposed so that the two peaks are more evident.

**Fig. 2.16** Histogram of protein consumption in 25 European countries for white meat. The histogram is bimodal, which indicates that the sample might be comprised of two subgroups.

and view it. This data set was collected in 1973 and includes the consumption measurements of nine food groups: **RedMeat**, **WhiteMeat**, **eggs**, **Milk**, **Fish**, **Cereals**, **Starch** (starchy foods), **nuts** (pulses, nuts, and oil-seeds), and **Fr.Veg** (fruits and vegetables). Use the steps described above to plot the density histogram of **WhiteMeat**. Figure 2.16 shows that the resulting histogram is bimodal. It seems that European countries are divided into two subgroups with respect to the amount of protein consumption from white meat.
2.4.2 Mean and Median

Histograms are useful for visualizing numerical data and identifying their location and spread. However, we typically use summary statistics for more precise specification of the central tendency and dispersion of observed values. A common summary statistic for location is the **sample mean**.

The **sample mean** is simply the average of the observed values. For observed values $x_1, \ldots, x_n$, we denote the sample mean as $\bar{x}$ and calculate it by

$$\bar{x} = \frac{\sum_i x_i}{n},$$

where $x_i$ is the $i$th observed value of $X$, and $n$ is the sample size.

For Sample 1, Sample 2, and Sample 3, the means are 2.1, 3.9, and 4.1, respectively. The means are shown as short vertical lines in Fig. 2.17.

The sample mean for bmi in Pima.tr is 32.3. In Fig. 2.18, the mean is shown by a solid line. In this case, the mean 32.3 appropriately represents the location (center) of the distribution and the central tendency of the observed values.

While sample mean is a very useful summary statistic for location, it is sensitive to very large or very small values, which might be outliers (unusual values). For instance, suppose that we have measured the resting heart rate (in beats per minute) for five people. The five measurements are {74, 80, 79, 85, 81}. We can calculate
Fig. 2.18  Histogram of bmi with the mean (solid line) and the median (dashed line) are shown as vertical lines. The mean and median are nearly equal since the histogram is symmetric.

the sample mean as

\[ x = \{74, 80, 79, 85, 81\}, \quad \bar{x} = \frac{74 + 80 + 79 + 85 + 81}{5} = 79.8. \]

In this case, the sample mean is 79.8, which seems to be a good representative of the data.

Now suppose that the heart rate for the first individual is recorded as 47 instead of 74. Compared to other four people, this is a much smaller number, which is either due to a data recording mistake, or the first person is in fact a well-trained athlete with low resting heart rate. In this case, the sample mean is heavily affected by this observation, which is regarded as an outlier, and it is drastically reduced to 74.4:

\[ x = \{47, 80, 79, 85, 81\}, \quad \bar{x} = \frac{47 + 80 + 79 + 85 + 81}{5} = 74.4. \]

Now, the sample mean does not capture the central tendency of the observed data since four out of five measurements are much larger than \( \bar{x} = 74.4 \).

The sample median is an alternative measure of location, which is less sensitive to outliers. For observed values \( x_1, \ldots, x_n \), the median is denoted \( \tilde{x} \) and is calculated by first sorting the observed values (i.e., ordering them from the lowest to the highest value) and selecting the middle one. If the sample size \( n \) is odd, the median is the number at the middle of the sorted observations. If the sample size is even, the median is the average of the two middle numbers.
The sample medians for the above two scenarios are

\[ x = \{74, 79, 80, 81, 85\}, \quad \bar{x} = 80; \]
\[ x = \{47, 79, 80, 81, 85\}, \quad \tilde{x} = 80. \]

In this example, the median remains equal to 80, which properly captures the central tendency of the observed values. In general, the median is not heavily influenced by outliers. We say that the median is more robust against outliers.

When there are no outliers and the histogram is almost symmetric, such as the histogram of bmi in Fig. 2.18, both the mean (solid line) and the median (dashed line) are close to each other, and both reasonably represent the location of data. However, when there are outliers, or when the histogram is skewed, such as the histogram of lwt in Fig. 2.19, the mean (solid line) moves toward the outliers or the direction of skewness in the histogram more than the median.

Occasionally, we might find situations in which neither the mean nor the median is a good representative of the central tendency. For example, Fig. 2.20 shows that the mean (solid line) and the median (dashed line) for the WhiteMeat variable do not capture the central tendency of the data. Most observed values in this case are clustered away from the mean and median. This is usually true for bimodal distributions.

### 2.4.3 Variance and Standard Deviation

While summary statistics such as mean and median provide insights into the central tendency of values for a variable, they are rarely enough to fully describe a distribution. We need other summary statistics that capture the dispersion of the distribution.
For example, consider Sample 2 and Sample 3 in Fig. 2.17. The two samples have similar locations, but Sample 3 is more dispersed than Sample 2. The deviations (differences) of observations from the center (e.g., mean) tend to be larger in Sample 3 compared to Sample 2.

As a further example, consider the following measurements of blood pressure (in mmHg) for two patients:

Patient A: \( x = \{95, 98, 96, 95, 96\} \), \( \bar{x} = 96, \ \tilde{x} = 96 \).

Patient B: \( y = \{85, 106, 88, 105, 96\} \), \( \bar{y} = 96, \ \tilde{y} = 96 \).

While the mean and median for both patients are 96, the readings are more dispersed for Patient B. Suppose that we choose 96 as the representative value of systolic blood pressure for both patients. For Patient A, there is a good chance that the next reading of blood pressure would be close to 96, for example, in the \([95, 97]\) range. For Patient B, the chance of seeing a blood pressure value close to 96 (e.g., in the \([95, 97]\) range) would be relatively smaller. For a better description of a variable, we need summary statistics that measure the dispersion (i.e., variability) of its observed values.

Two common summary statistics for measuring dispersion are the sample variance and sample standard deviation. These two summary statistics are based on the deviation of observed values from the mean as the center of the distribution. For each observation, the deviation from the mean is calculated as \( x_i - \bar{x} \). It is easy to show that the sum of these deviations over all observed values is always zero. (Note that \( \bar{x} = \frac{1}{n} \sum x_i \).) Therefore, we cannot simply use the sum of the deviations as a measure of dispersion. However, the sum would not be zero in general if we ignore the signs of these deviations (i.e., focus on the distances from the mean). For this, we can either take the absolute value of deviations, \( |x_i - \bar{x}| \), or square them,
Either way, the sign of deviations becomes irrelevant. Taking the squares of the deviations is a more popular choice. We can then use the average of these squared deviations over all observations as a measure of dispersion:

\[
\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}.
\]  

(2.4)

Instead of dividing by \( n \), it is more common to divide by \( n - 1 \). (This increases the above dispersion measurement by a small amount.) The result is called the sample variance.

The sample variance is a common measure of dispersion based on the squared deviations. The variance, denoted \( s^2 \), is calculated as

\[
s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}.
\]

(2.5)

If we take the square root of the variance,

\[
s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}},
\]

(2.6)

the result is called the sample standard deviation:

Table 2.3 shows the steps for calculating the sample variance and sample standard deviation of blood pressure readings for Patient A and Patient B in the above example. In comparison, the standard deviation for Patient A is much smaller than the standard deviation for Patient B. Thus, we can conclude that the observed blood pressure values are less dispersed for Patient A compared to Patient B.

<table>
<thead>
<tr>
<th>Patient A</th>
<th>Patient B</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i )</td>
<td>( y_i )</td>
</tr>
<tr>
<td>( x_i - \bar{x} )</td>
<td>( y_i - \bar{y} )</td>
</tr>
<tr>
<td>( (x_i - \bar{x})^2 )</td>
<td>( (y_i - \bar{y})^2 )</td>
</tr>
<tr>
<td>95</td>
<td>85</td>
</tr>
<tr>
<td>98</td>
<td>106</td>
</tr>
<tr>
<td>96</td>
<td>88</td>
</tr>
<tr>
<td>95</td>
<td>105</td>
</tr>
<tr>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>( \Sigma )</td>
<td>( \Sigma )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>366</td>
</tr>
</tbody>
</table>

\[
s^2 = \frac{6}{4} = 1.5 \quad s^2 = \frac{366}{4} = 91.5
\]

\[
s = \sqrt{1.5} = 1.22 \quad s = \sqrt{91.5} = 9.56
\]
2.4 Exploring Numerical Variables

2.4.4 Quantiles

Informally, the sample median could be interpreted as the point that divides the ordered values of the variable into two equal parts. More precisely, the median is the point that is greater than or equal to at least half of the values and smaller than or equal to at least half of the values. Therefore the median is called the 0.5 quantile, which, as we discussed above, provides a measure of location. Similarly, the 0.25 quantile is the point that is greater than or equal to at least 25% of the values and smaller than or equal to at least 75% of the values. In general, the $q$ quantile is the point that is greater than or equal to at least $100q\%$ of the values and smaller than or equal to at least $100(1 - q)\%$ of the values. Sometimes, we refer to the $q$ quantile as the $100q$th percentile. For example, the 0.25 quantile is the 25th percentile, and the median is the 50th percentile.

We can divide the ordered values of a variable into four equal parts using 0.25, 0.5, and 0.75 quantiles. The corresponding points are denoted $Q_1$, $Q_2$, and $Q_3$, respectively. Note that $Q_2$ is the 0.5 quantile and is therefore the same as the median. $Q_1$ is the point that divides the lower half of the data (i.e., below the median) into two equal parts. $Q_3$ is the point that divides the upper half of the data into two equal parts. We refer to these three points as quartiles, of which $Q_1$ is called the first quartile or the lower quartile, $Q_2$ (i.e., median) is called the second quartile, and $Q_3$ is called the third quartile or upper quartile. The interval from $Q_1$ (0.25 quantile) to $Q_3$ (0.75 quantile) covers the middle 50% of the ordered data.

The minimum (min), which is the smallest value of the variable in our sample, is in fact the 0 quantile. On the other hand, the maximum (max), which is the largest value of the variable in our sample, is the 1 quantile. The minimum and maximum along with quartiles ($Q_1$, $Q_2$, and $Q_3$) are known as five-number summary. These are usually presented in the increasing order: min, first quartile, median, third quartile, max. This way, the five-number summary provides 0, 0.25, 0.50, 0.75, and 1 quantiles.

We can use R-Commander to obtain the five-number summary along with mean and standard deviation. Make sure birthwt is the active data set. Click Statistics → Summaries → Numerical summaries (Fig. 2.21). Now select

![Image](image-url)
The five-number summary can be used to derive two measures of dispersion: the range and the interquartile range. The range is the difference between the maximum observed value and the minimum observed value. The interquartile range (IQR) is the difference between the third quartile ($Q_3$) and the first quartile ($Q_1$). Compared to the range, the IQR is less sensitive to outliers, which usually fall below $Q_1$ or above $Q_3$.

Using the results in Fig. 2.22, the range for $bwt$ is $4990 - 709 = 4281$ grams, while the IQR is $3487 - 2414 = 1073$ grams. For this variable, 50% of the birth weight values fall within the $[2414, 3487]$ interval. The birth weight for 25% of babies is above 3487 grams, and for 25% of babies is below 2414 grams.

### 2.4.5 Boxplots

To visualize the five-number summary, the range and the IQR, we often use a boxplot (a.k.a. box and whisker plot). Figure 2.23 shows the boxplot for $bwt$ along with the plot of actual observed values. The thick line at the middle of the “box” shows the median $\tilde{x} = 2977$. The left side of the box shows the lower quartile $Q_1 = 2414$. Likewise, the right side of the box is the upper quartile $Q_3 = 3487$. Therefore, the box stretches from the lower quartile to the upper quartile and represents the middle 50% of the values of the ordered data. The length of the box is therefore the IQR, which in this case is equal to 1073. 25% of the observations are to the left of this box, and 25% are to the right of it.

The dashed lines extending from the box are known as the whiskers. The whisker on the right of the box extends to the largest observed value or $Q_3 + 1.5 \times \text{IQR}$, whichever it reaches first. The whisker on the left extends to the lowest value or $Q_1 - 1.5 \times \text{IQR}$, whichever it reaches first. Data points beyond the whiskers (i.e.,
2.4 Exploring Numerical Variables

**Fig. 2.23** Horizontal boxplot along with the actual observed values of birth weight from the \texttt{birthwt} data set. The gray box shows the middle 50\% of ordered observed values. The thick line in the middle of the box is the median ($Q_2$) of 2977 observations.

![Horizontal boxplot](image)

**Fig. 2.24** Vertical boxplot for \texttt{bwt} using R-Commander

![Vertical boxplot](image)

either less than $Q_1 - 1.5 \times \text{IQR}$ or greater than $Q_3 + 1.5 \times \text{IQR}$) are shown as circles and considered as possible outliers. For \texttt{bwt}, the right whisker extends to the maximum value 4990 since it reaches to this value before $3487 + 1.5 \times 1073 = 5096.5$. The left whisker extends to $2414 - 1.5 \times 1073 = 804.5$ since it reaches this point before it reaches the minimum value 709. There is one observation to the left of this whisker, which is shown as a circle. This is, in fact, the minimum observed value, 709, which in this case is considered as a potential outlier.

Very often, boxplots are drawn vertically. This is the default option in R-Commander. To create a boxplot for \texttt{bwt} in R-Commander, make sure \texttt{birthwt} is the active dataset, click Graphs $\rightarrow$ Boxplot, and select \texttt{bwt}. The resulting boxplot is shown in Fig. 2.24. This is the same as the boxplot shown in Fig. 2.23 after 90° rotation.
Now, consider the boxplot of $lwt$ (Fig. 2.25), whose is distribution is right-skewed. The sample median ($\bar{x} = 121$) is closer to the bottom ($Q_1 = 110$) than to the top ($Q_3 = 140$) of the box. This is an indication of skewed distribution. Moreover, the upper whisker extends substantially further than the lower whisker. There are several possible outliers, whose observed values fall beyond the whisker on the top of the box.

### 2.5 Data Preprocessing

Many of the data sets we have been using as examples have been collected in scientific studies. Typically, such data are not ready for immediate analysis. The most common issues are missing values and outliers. For example, the original data on women of Pima Indian Heritage (collected by US National Institute of Diabetes and Digestive and Kidney Diseases) included many observations with missing values. The data set we have been using so far (Pima.tr) was obtained after removing these observations. We refer to data in their original form (i.e., collected by researchers) as the raw data. Before using the original data for analysis, we should thoroughly check them for missing values and possible outliers. Data exploration techniques we discussed in this chapter can help us to identify data issues that need to be addressed before further analysis. Collectively, we refer to the process of preparing the raw data for analysis as data preprocessing. Here, we discuss some simple preprocessing steps.

#### 2.5.1 Missing Data

For our first example, we look at the Pima.tr2 data set, which includes the Pima.tr data set plus many other observations with missing values. The
Pima.tr2 is available in the MASS package. Follow the steps described in the previous chapter to load the MASS package and select Pima.tr2 (which is located right after Pima.tr in the list) as the active data set. Figure 2.26 shows a part of this data set. Here, missing values are denoted NA (Not Available).

In general, it is up to the researcher to decide whether to remove the observations with missing values or impute (guess) the missing values in order to keep the observations. If we choose to remove all observations with missing values (this is how the Pima.tr data set was created based on Pima.tr2), we can do so by clicking Data → Active data set → Remove cases with missing data. Under Name for new data set enter Pima.complete. This creates a data set, which does not include any observation with missing values. (Notice that Pima.complete becomes the active data set.) Indeed, this data set is exactly the same as Pima.tr, which we have been using so far.

While simply removing observations with missing values is an easy approach for handling missing data, it is quite wasteful and inefficient. On the other hand, missing data imputation techniques, i.e., using statistical methods to fill-in missing values, tend to be complex. However, if done properly, they can improve our analysis. For an overview of statistical methods for analyzing data with missing values, refer to [18].

Sometimes we can temporarily ignore missing values if the variable whose values are missing is not the focus of our analysis at the moment. In the above example, if we are focusing on the bp (blood pressure) variable, we do not need to remove observations 201, 202, 203, 205, .. . Of course, we still need to either remove or impute the observation 204 and any other observation whose blood pressure reading is missing. To remove individual observations, click Data → Active data set → Remove row(s) from active data and enter the row numbers (the leftmost number in the data set) for observations you want to remove.

2.5.2 Outliers

Dealing with missing values is not the only challenge of working with raw data. Sometimes, an observed value of a variable is suspicious since it does not follow the
Fig. 2.27  Frequency table for gender from the AsthmaLOS data set. The value of gender for two observations are entered as “4”, while gender can only take 0 or 1.

overall patterns presented by the rest of the data. We refer to such observations as outliers. Suppose, for example, that almost all BMI values in our sample are between 20 and 40. Observing a BMI value of 50 would be suspicious. Further investigation might reveal that in fact this is the correct value of BMI for an individual in our sample. In this case, this outlier is a legitimate value. However, a BMI value of 500 or –50 is clearly an erroneous observation, which is possibly due to a data entry mistake.

We could identify outliers using data exploration techniques. As an example, we use the AsthmaLOS data collected by [12] to study the length of stay in hospital for asthmatic children in the USA. Download the data set from the book website (http://extras.springer.com) and import it to R-Commander. The variables in this data sets are:

- los: length of stay in hospital (in days).
- hospital.id: hospital ID.
- insurer: the insurer, which is either 0 or 1.
- age: the age of the patient.
- gender: the gender of the patient; 1 for female, and 0 for male.
- race: the race of the patient; 1 for white, 2 for Hispanic, 3 for African-American, 4 for Asian/Pacific Islander, 5 for others.
- bed.size: the number of beds in the hospital; 1 means 1 to 99, 2 means 100 to 249, 3 means 250 to 400, 4 means 401 to 650.
- owner.type: the hospital owner; 1 for public, 2 for private.
- complication: if there were any treatment complication; 0 means there were no complications, 1 means there were some complications.

Before working with this data set, follow the steps discussed in Sect. 2.2 to convert the variables hospital.id, insurer, gender, race, owner type, and complication to factors (categorical). Next, obtain the frequency tables for gender. The resulting tables are shown in Fig. 2.27. Notice that while the gender variable can take only two values, 1 for female and 0 for male, the data include two observations whose values for gender is “4”. These values are entered by mistake and should be either removed (as described above) or corrected if possible. If we know the correct values for these observations (e.g., by examining the medical records), we can edit the data and keep the observations. To edit a data set, click Edit data set button in front of its name on the menu bar. This opens the R Data Editor window, where you can find the erroneous values and correct them.
Now consider the variable \textit{los} (length of stay) in the AsthmaLOS data set. Figure 2.28 shows the boxplot for this variable. As we can see, there are two children whose length of stay is extremely large (50 and 58). These values are not consistent with the rest of data. (All other values are less than 10.) However, if we find that they are legitimate and correctly recorded values, we should keep them in our data since they provide important information on the distribution of the variable (e.g., how extreme could be). Of course, such observations can drastically affect our results. For analyzing such data, we could use statistical methods that are more robust against outliers (e.g., median, IQR).

2.5.3 Data Transformation

Occasionally, we rely on data transformation techniques (i.e., applying a function to the variable) to reduce the influence of extreme values in our analysis. Two of the most commonly used transformation functions for this purpose are logarithm and square root. The logarithm function, \( \log(x) \), is usually used to transform right-skewed variables with positive values. The square root function is usually used for right-skewed count variables. We use these transformations to reduce the skewness, i.e., to make it more symmetric, and reduce the influence of extreme values.

Consider the \textit{lwt} variable in the birthwt data set. As shown in the left panel of Fig. 2.29, the variable is right-skewed. To use log-transformation, click Data \rightarrow Manage variables in active data set \rightarrow Compute new variable. Under New variable name, enter log.lwt, and under Expression to compute, enter \( \log(\text{lwt}) \). (If we want to use the square root transformation, we use \( \text{sqrt} \) instead of \( \log \).) This creates a new variable \textit{log.lwt} whose values are the natural logarithm of \textit{lwt}. Next, create the density histogram for this newly created variable. As shown in the right panel of Fig. 2.29, the resulting variable is less skewed compared to the original variable.
The transformation techniques discussed so far are used commonly in statistical analysis. You can of course use the above approach to transform a variable in many other ways. For example, suppose that you want to apply the square transformation to a variable $X$. (This is also a common transformation in regression analysis.) To do this, you can follow the above steps and simply enter $X^2$ under Expression to compute. (Here, symbol “^” is used for exponentiation.)

### 2.5.4 Creating New Variable Based on Two or More Existing Variables

In the previous chapter, we discussed creating new variables based on existing ones as a common data preprocessing step. Here, we show how we can create a new variable based on two or more existing variables. Consider the bodyfat data set, which includes weight and height. Using these two variables, we can calculate BMI for each person in the sample using the equation

$$BMI = \frac{\text{weight} \times 703}{(\text{height})^2},$$

where weight is in pounds, and height is in inches.

To create BMI, click Data → Manage variables in active data set → Compute new variable. Under New variable name, enter BMI, and under Expression to compute, we enter (Fig. 2.30)

$$(\text{weight} \times 703)/(\text{height}^2)$$

This will create a new variable called BMI. You can now investigate the linear relationship between this variable and percent body fat by calculating their sample correlation coefficient. Pearson’s correlation coefficient between $siri$ and BMI is 0.72, which indicates a strong positive linear relationship as expected.
2.5 Data Preprocessing

**Fig. 2.30** Creating a new variable BMI based on weight and height for each person in the bodyfat data set

<table>
<thead>
<tr>
<th>Table 2.4</th>
<th>Standard weight status based on BMI according to CDC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BMI</strong></td>
<td><strong>Weight Status</strong></td>
</tr>
<tr>
<td>Below 18.5</td>
<td>Underweight</td>
</tr>
<tr>
<td>18.5–24.9</td>
<td>Normal</td>
</tr>
<tr>
<td>25.0–29.9</td>
<td>Overweight</td>
</tr>
<tr>
<td>30.0 and Above</td>
<td>Obese</td>
</tr>
</tbody>
</table>

2.5.5 Creating Categories for Numerical Variables

Another common preprocessing technique is to create categorical variables based on numerical variables. This could help us to see the patterns more clearly and identify relationships more easily. Recall that histograms are created by dividing the range of a numerical variable into intervals. Instead of using arbitrary intervals, we might prefer to group the values in a meaningful way. This way, we can create a categorical variable based on a numerical variable. For example, according to the Centers for Disease Control and Prevention (CDC), the standard weight status categories associated with BMI ranges for adults are as in Table 2.4.

In R-Commander, let us divide subjects based on their **bmi** (from the **Pima.tr**) into four groups: Underweight, Normal, Overweight, and Obese. Click **Data → Manage variables in active data set → Recode variables**. Select **bmi** as the Variable to recode and enter “**weight.status**” as the New variable name (Fig. 2.31). Then in the Enter recode directives box, type

```
0:18.5 = "Underweight"
18.5:24.9 = "Normal"
25.0:29.9 = "Overweight"
30.0:100 = "Obese"
```

Now view the **Pima.tr** data set. The newly created variable **weight.status** is added to the data set. This variable is categorical. More specifically, it is an ordinal variable. To specify the order of categories in R-Commander, click **Data → Manage variables in active data set → Reorder factor**
Recoding the numerical variable bmi to be categorical (weight.status)

Reordering the categories for the variable weight.status such that “Underweight” is the first category, “Normal” is the second category, “Overweight” is the third category, and “Obese” is the fourth category.

levels. Then select weight.status. R-Commander will open a window to reorder levels of the categorical variable. Change the order according to Fig. 2.32. (Note that the default order is alphabetical.) Now you can create the barplot for weight.status (Fig. 2.33). The graph of the weight.status variable clearly indicates that the “Obese” category has the highest frequency.

### 2.6 Advanced

In this section, we discuss some data exploration and data transformation techniques that are slightly more advanced. We also discuss some commonly used R functions for data exploration.

#### 2.6.1 Coefficient of Variation

Suppose that we want to compare the dispersion of bwt to that of lwt using their standard deviations. Use R-Commander to obtain the means and standard deviations for bwt and lwt in the birthwt data set. Based on the results shown in Fig. 2.34,
it seems that \( bwt \) is more dispersed than \( lwt \) since it has higher standard deviation compared to \( lwt \). However, the two variables are not comparable; they have different units. Let us change the unit of \( bwt \) from grams to pounds. For this, we need to divide its values by 453.6. In R-Commander, click Data → Manage variables in active data set → Compute new variable. This opens a window (Fig. 2.35), where we create new variable for birth weight in pounds. Under new variable name, enter \( bwt.lb \). Under Expression to compute, enter \( bwt/453.6 \). The newly created variable \( bwt.lb \), whose values are birth weight in pound, will be added to the birthwt data set. (View the data set to make sure that this is done correctly.)
Fig. 2.36 Creating a new variable \(bwt\cdot lb\) (birth weight in pounds) and obtaining its summary statistics.

Now, use R-Commander to find the mean and standard deviation of \(bwt\) and \(bwt\cdot lb\). The results are shown in Fig. 2.36. After changing the measurement unit from grams to pounds, the standard deviation changes from 729.2 to 1.6. Now, this is much smaller than the standard deviation of \(lwt\), which is 30.6 (see Fig. 2.34). This is of course expected since the values of \(lwt\) are much larger than the values of \(bwt\cdot lb\). As a result, \(lwt\) has much larger sample mean and larger deviations around the mean compared to \(bwt\cdot lb\).

The above results illustrate how difference in measurement units and large differences in sample means make it difficult to compare variables based on their standard deviations. In many situations, we can avoid these issues by using another measure of variation called the **coefficient of variation** instead of standard deviation.

To quantify dispersion independently from units, we use the coefficient of variation, which is the standard deviation divided by the sample mean (assuming that the mean is a positive number):

\[
CV = \frac{s}{\bar{x}}.
\]

The coefficient of variation for \(bwt\) (birth weight in grams) is 729.2/2944.6 = 0.25 and for \(bwt\cdot lb\) (birth weight in pounds) is 1.6/6.5 = 0.25. Therefore, the coefficient of variation is the same, even though \(bwt\) has a larger standard deviation compared to \(bwt\cdot lb\). Comparing this coefficient of variation to 30.6/129.8 = 0.24, which is the coefficient of variation for \(lwt\), suggests that the two variables have roughly the same dispersion in terms of CV. In general, the coefficient of variation is used to compare variables in terms of their dispersion when the means are substantially different (possibly as the result of having different measurement units).

### 2.6.2 Scaling and Shifting Variables

To see why the coefficient of variation \((CV = s/\bar{x})\) is independent of measurement units in the above example, we need to learn about how the mean and standard deviation change when we change the scale of a variable. For example, we changed the scaled of \(bwt\) by multiplying it by the constant \(1/453.6\) (i.e., dividing it by 453.6).
In general, when we multiply the observed values of a variable by a constant $a$, its mean, standard deviation, and variance are multiplied by $a$, $|a|$, and $a^2$, respectively. That is, if $y = ax$, then

$$\bar{y} = a\bar{x},$$
$$sy = |a|sx,$$
$$\sigma^2_y = a^2\sigma^2_x,$$

where $\bar{x}$, $sx$, and $\sigma^2_x$ are the sample mean, standard deviation, and variance of the original observations $x$, and $\bar{y}$, $sy$, and $\sigma^2_y$ are the sample mean, standard deviation, and variance of scaled observations $y$.

In the above example, the mean and standard deviation of $bwt$ (denoted $x$) were $\bar{x} = 2944.6$ and $sx = 729.2$, respectively (Fig. 2.22). To convert the measurement unit to pounds, we multiplied $bwt$ by $a = 1/453.6$ to create a new variable $bwt\.lb$ (denoted $y$). The mean and standard deviation of $bwt\.lb$ are therefore as follows:

$$\bar{y} = \frac{1}{453.6} \times 2944.6 = 6.5,$$
$$sy = |a|sx = \frac{1}{453.6} \times 729.2 = 1.6,$$

which are the same values as what we obtained by using R-Commander (Fig. 2.36).

When the measurement units are changed by multiplying the observed values by a positive constant (e.g., multiplying by $1/453.6$ in the above example to convert grams to pounds), the coefficient of variation is not affected since both mean and standard deviation will be multiplied by that number. If $y = ax$ (where $a$ is a positive constant), then

$$CV_y = \frac{sy}{\bar{y}} = \frac{asx}{a\bar{x}} = \frac{sx}{\bar{x}} = CV_x.$$

What happens if instead of scaling the observed value, we shift them by a constant $b$ (which can be negative): $y = x + b$? For example, suppose after researchers collected the birthwt data set, they realized that the weighting scale they used to measure birth weight was not calibrated properly, and they need to add 20 grams to the weight of each child, i.e., $y = x + 20$. Therefore, all the observed values for $bwt$ will be shifted upwards by 20 points. Intuitively, this shifts the sample mean by 20 points. However, since the difference between observed values and the mean do not change, the standard deviation and variance remain unchanged. In general, if we shift the observed values by $b$, i.e., $y = x + b$, then

$$\bar{y} = \bar{x} + b,$$
$$sy = sx,$$
$$\sigma^2_y = \sigma^2_x.$$
If we multiply the observed values by the constant \( a \) and then add the constant \( b \) to the result, i.e., \( y = ax + b \), then
\[
\bar{y} = a\bar{x} + b,
\]
\[
s_y = |a|s_x,
\]
\[
s_y^2 = a^2s_x^2.
\]

Therefore, when changing measurement units involved adding a constant (e.g., adding 273.15 to convert Celsius to Kelvin), the coefficient of variation will change. If \( y = ax + b \) (assuming \( a > 0 \) and \( b \neq 0 \)), then
\[
CV_y = \frac{s_y}{\bar{y}} = \frac{as_x}{a\bar{x} + b} \neq \frac{s_x}{\bar{x}}.
\]

### 2.6.3 Variable Standardization

**Variable standardization** is a common *linear* transformation, where we subtract the sample mean \( \bar{x} \) from the observed values and divide the result by the sample standard deviation \( s \), in order to shift the mean to zero and make the standard deviation 1:
\[
y_i = \frac{x_i - \bar{x}}{s}.
\]

Using such transformation is especially common in regression analysis (Chap. 11) and clustering (Sect. 12.1). Following the rules we discussed above, subtracting \( \bar{x} \) from the observations shifts the sample mean to zero. This, however, does not change the standard deviation. Dividing by \( s \), on the other hand, changes the sample standard deviation to 1. The mean is also divided by \( s \). However, since the sample mean has become zero after subtracting \( \bar{x} \), it remains zero. Therefore, variable standardization creates a new variable with mean 0 and standard deviation 1.

Suppose that we want to standardize \( lwt \) using R-Commander. For this, we can follow the steps for computing a new variable (Sect. 2.6.1), enter `std.lwt` under New variable name, and `(lwt - 129.8)/30.6` under Expression to compute. This creates the standardized version of \( lwt \) called `std.lwt`. Now, find the mean and standard deviation of `std.lwt`. Alternatively, we can standardize a variable by clicking Data \( \rightarrow \) Manage variables in active data set \( \rightarrow \) Standardize variables. Select `lwt` under Variables. This will create a new variable called `Z.lwt`, which will be added to the data set. View the `birthwt` data set and find the mean and standard deviation of the newly created variable `Z.lwt`. 
2.6.4 Data Exploration with R Programming

Writing your own R commands (as opposed to using R-Commander) gives you more control over the output and a deeper understanding of the material. In Appendix B, we provide a brief introduction to R programming. Here, we review the functions that are commonly used for data exploration. We start by loading the `Pima.tr` data set, which is available from the MASS package.

```r
> library(MASS)
> data(Pima.tr)
```

The `library()` command loads the MASS package, and the `data()` command loads the `Pima.tr` data set. Note that the package should be loaded first before we can access its data sets.

Type `Pima.tr` to view the entire data set. If the data set is large, it is better to use the `head()` function, which shows only the first part (few rows) of the data set.

```r
> head(Pima.tr)
```

```
npreg glu bp skin bmi ped age type
1  5  86  68  28  30.2 0.364  24  No
2  7 195  70  33  25.1 0.163  55  Yes
3  5  77  82  41  35.8 0.156  35  No
4  0 165  76  43  47.9 0.259  26  No
5  0 107  60  25  26.4 0.133  23  No
6  5  97  76  27  35.6 0.378  52  Yes
```

When you obtain a data set from a package, you can use the `help()` function to view the description on the data available in the package.

```r
> help(Pima.tr)
```

*Bar Graphs and Frequencies* A common summary statistic for categorical variables is its frequencies, $n_c$. Use the `table()` function to obtain the frequencies for the categorical variable `type` from the `Pima.tr` data set.

```r
> type.freq <- table(Pima.tr$type)
> type.freq

No Yes
132  68
```
Note that the $ symbol is being used to access the type variable in the Pima.tr data set.

Now, use the type.freq table to create the bar graph. Bar graphs show us how observations categorical variables are distributed in the sample.

```r
> barplot(type.freq, xlab = "Type", ylab = "Frequency", + main = "Frequency Bar Graph of Type")
```

The first parameter to the `barplot()` function is the frequency table. The options `xlab` and `ylab` label the x and y axes, respectively. Likewise, the `main` option puts a title on the plot.

Often it is more informative to report the relative frequencies. The relative frequency is the percentage or proportion in each category and is calculated by $p_c = n_c/n$ as in Eq. 2.1. Therefore, we need the frequencies $n_c$ (stored in the `type.freq` table) and the total sample size $n$. Since the sum of the frequencies is the total sample size, $\sum n_c = n$, we can use the `sum()` function to add the entries in the frequency table:

```r
> n <- sum(type.freq)
> n
[1] 200
```

Now create a table of relative frequencies by dividing the frequency table by the sample size:

```r
> type.rel.freq <- type.freq/n
```

Use the `round()` function to limit the output to 2 decimal places:

```r
> round(type.rel.freq, 2)
```

No   Yes
0.66 0.34

We can also multiply the relative frequencies by 100 to get the percentages:

```r
> round(type.rel.freq, 2) * 100
```

No   Yes
66   34

Finally, you can create a relative frequency barplot with

```r
> barplot(type.rel.freq, xlab = "Type", + ylab = "Relative Frequency", + main = "Relative Frequency Bar Graph of Type")
```
If the levels of a categorical variable in the data set is coded as numbers, we need to convert the type of variable to factor using the `factor()` function, so that R recognizes it as categorical. You can use the function `is.factor()` to examine whether a variable is a factor. For example, the `smoke` variable (smoking status) in `birthwt` is coded as 0 for mothers who did not smoke during their pregnancy and 1 for mothers who smoked during their pregnancy. R automatically considers this variable as numerical. To convert the variable to categorical, use the following code:

```r
> data(birthwt)
> is.factor(birthwt$smoke)

[1] FALSE

> birthwt$smoke <- factor(birthwt$smoke)
> is.factor(birthwt$smoke)

[1] TRUE

> table(birthwt$smoke)

   0  1
115  74
```

**Histograms**   Histograms are commonly used to visualize numerical variables. To create a frequency histogram for `age`, use the `hist()` function with the `freq` option set to “TRUE” (which is the default):

```r
> hist(Pima.tr$age, freq = TRUE, 
+       xlab = "Age", ylab = "Frequency", 
+       col = "grey", main = "Frequency Histogram of Age")
```

Then create a density histogram of `age` by setting the `freq` option to “FALSE”:

```r
> hist(Pima.tr$age, freq = FALSE, 
+       xlab = "Age", ylab = "Density", 
+       col = "grey", main = "Density Histogram of Age")
```

**Summary Statistics**   We can obtain the mean and median of numerical data with the `mean()` and `median()` functions. Find these statistics for numerical variables in `Pima.tr`:

```r
> mean(Pima.tr$npreg)
```
The `quantile()` function with the `probs` option returns the specified quantiles:

```r
> quantile(Pima.tr$bmi, probs = c(0.1, 0.25, 0.5, 0.9))
10% 25% 50% 90%
24.200 27.575 32.800 39.400
```

Here, the desired quantiles are specified as a vector using the combine `c()` function. The five-number summary along with the mean can simply be obtained with the `summary()` function:

```r
> summary(Pima.tr$bmi)
```

```
     Min.  1st Qu.   Median     Mean   3rd Qu.     Max.     
 18.2000   27.580   32.8000   32.3100   36.5000   47.9000
```

We can present the five-number summary visually with a boxplot:

```r
> boxplot(Pima.tr$bmi, ylab = "BMI")
```

While the default is to create vertical boxplots, we can also create horizontal boxplots by specifying the `horizontal` option to `true`:

```r
> boxplot(Pima.tr$bmi, ylab = "BMI", horizontal = TRUE)
```

Find the interquartile range (IQR) with the `IQR()` function:

```r
> IQR(Pima.tr$bmi)
[1] 8.925
```

The smallest and largest observations can be obtained with the `range()` function (the functions `min()` and `max()` could also be applied):

```r
> minMax <- range(Pima.tr$bmi)
> minMax
[1] 18.2 47.9
```
Here, we created a vector object `minMax` with the minimum as the first element and the maximum as the second element. Obtain the range by subtracting the first element from the second:

```r
> minMax[2] - minMax[1]
[1] 29.7
```

The variance and standard deviation are also easily calculated with `var()` and `sd()`:

```r
> var(Pima.tr$bmi)
[1] 37.5795

> sd(Pima.tr$bmi)
[1] 6.130212
```

**Creating Categories for Numerical Variables** The `hist()` function automatically divides the range of possible values into several intervals. Instead, as discussed above, we can create more meaningful intervals, which will be treated as categories. To create a categorical variable `weight.status` based on the `bmi` variable in `Pima.tr`, we can go through each observation one by one and assign each observation to one of the four categories: “Underweight”, “Normal”, “Overweight”, and “Obese”. To do this, we can use loops and conditional statements, which are discussed in Appendix B.

First, we start by creating an empty vector of size 200 within the `Pima.tr` data frame:

```r
> Pima.tr$weight.status <- rep(NA, 200)
```

Next, we set the values of `weight.status` for all observations by using if-else() statements within a for() loop:

```r
> for (i in 1:200) {
+   if (Pima.tr$bmi[i] < 18.5) {
+     Pima.tr$weight.status[i] <- "Underweight"
+   }
+   else if (Pima.tr$bmi[i] >= 18.5 & Pima.tr$bmi[i] < 24.9) {
+     Pima.tr$weight.status[i] <- "Normal"
+   }
+   else if (Pima.tr$bmi[i] >= 24.9 & Pima.tr$bmi[i] < 29.9) {
+     Pima.tr$weight.status[i] <- "Overweight"
+   }
+   else if (Pima.tr$bmi[i] >= 29.9) {
+     Pima.tr$weight.status[i] <- "Obese"
+   }
+ }
```
+ Pima.tr$weight.status[i] <- "Overweight"
+ } else {
+ Pima.tr$weight.status[i] <- "Obese"
+ }
+ }

Here, the loop counter goes from 1 to 200. Use the `head()` function to view the result:

```r
> head(Pima.tr)

   npreg glu bp skin bmi ped age type weight.status
1      5   86  68  28 30.2  0.364  24   No   Obese
2      7  195  70  33 25.1  0.163  55   Yes  Overweight
3      5   77  82  41 35.8  0.156  35   No   Obese
4      0  165  76  43 47.9  0.259  26   No  Obese
5      0  107  60  25 26.4  0.133  23   No  Overweight
6      5   97  76  27 35.6  0.378  52   Yes  Obese
```

Before we use the newly created variable `weight.status` in statistical analysis, we should convert its type to factor.

```r
> Pima.tr$weight.status <- factor(Pima.tr$weight.status)
```

While the above code makes `weight.status` a factor variable, it does not take into account the ordering of levels. The levels are ordered alphabetically and can be examined using the `levels()` function:

```r
> levels(Pima.tr$weight.status)
```

```
[1] "Normal"  "Obese"  "Overweight"  "Underweight"
```

We can provide the right ordering when we use the `factor()` function to convert the variable:

```r
> Pima.tr$weight.status <- factor(Pima.tr$weight.status, 
+ levels = c("Underweight", "Normal", 
+ "Overweight", "Obese"))
> levels(Pima.tr$weight.status)
```

```
[1] "Underweight"  "Normal"  "Overweight"  "Obese"
```
Handling Missing Data in R  
To find missing values of a variable, we can use the `is.na()` function, which returns “TRUE” when the value is missing and “FALSE” otherwise. Consider the Pima.tr2 data set from the MASS library (the Pima.tr data set is obtained from Pima.tr2 by removing observations with missing values):

```r
> data(Pima.tr2)
> is.na(Pima.tr2$bp)
```

To obtain the indices of observations whose values are missing, we can use the `which()` function along with the `is.na()` function. In general, `which()` can be used to find the indices of “TRUE” values for a logical vector:

```r
> which(is.na(Pima.tr2$bp))
```

The `complete.cases()` function returns a logical vector indicating which cases (observations) in the data set are complete, i.e., have no missing values:

```r
> complete.cases(Pima.tr2)
```

To remove cases with missing values, we can use the `na.omit()` function:

```r
> Pima.complete <- na.omit(Pima.tr2)
```

Here, the newly created Pima.complete data set includes only the complete cases from Pima.tr2.

### 2.7 Exercises

1. Download the calcium data set from the Data and Story Library: [http://lib.stat.cmu.edu/DASL/Datafiles/Calcium.html](http://lib.stat.cmu.edu/DASL/Datafiles/Calcium.html). The data were collected to investigate whether increasing calcium intake reduces blood pressure. 21 people participated in this experiment, where ten of them took a calcium supplement for 12 weeks, while the remaining 11 received a placebo. The blood pressure of each subject was measured before and after the 12-week period. Plot the histogram of the variables Begin and End. Compare the two histograms in terms of their central tendency and the form of their histogram.

2. Download the “Survival.txt” data set from the book website ([http://extras.springer.com](http://extras.springer.com)). This data set appeared in Haberman (1976) and was obtained from the UCI Machine Learning Repository. The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago’s Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The variables are:
Table 2.5 Height (in inches) and weight (in pounds) for five newborn babies

<table>
<thead>
<tr>
<th>Observation</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>7.8</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>9.1</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>8.2</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>6.4</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>8.8</td>
</tr>
</tbody>
</table>

Fig. 2.37 Boxplot of variable X

- Age: Age of patient at time of operation.
- Nodes: Number of positive axillary nodes detected.
- Status: Survival status.

Plot the boxplot for Age and the bar graph for Status. Plot the histograms for Nodes and $\sqrt{\text{Nodes}}$. Which one is more skewed?

3. Show that the total area of rectangles in a density histogram is 1.

4. We have measured the height (in inches) and weight (in pounds) for five newborn babies. Manually calculate the mean and standard deviation of height and weight; show all the steps (Table 2.5).

5. Based on the boxplot in Fig. 2.37, write down the five-number data summary, range and IQR of variable X.

6. Download the “BodyTemperature.txt” from the book website (http://extras.springer.com), and find the five-number data summary for all numerical variables. For numerical variables, provide the histograms and boxplots. Comment on the central tendency and the form of the histograms. Are there any outliers in the data?

7. For the previous question, find the coefficient of variation for Age and Temperature variable. Show that the coefficient of variation remains the same if we change the units of Age to months (i.e., multiplying by 12). Change the body temperature scale to Celsius and recalculate the coefficient of variation. Comment on your findings.

8. The coefficient of variation for variable X is 2. If the sample mean of this variable is 3, what is the sample variance?
9. Download the “AsthmaLOS.txt” data from the book website (http://extras.springer.com). Read the description of variables provided in Sect. 2.5. Using R-Commander, identify data entry errors for race and owner.type. Remove the corresponding observations (i.e., rows) from the data set. Plot the histogram age and comment on its shape. For this variable, find the mean, variance, range, and IQR.

10. Upload the Animals data from the MASS package. This data set includes average brain and body weights for 28 species of land animals. Plot the histograms of the two numerical variables. Next, use the log transformation for both variables and plot the histograms again. Comment on the shapes of these new histograms.
Biostatistics with R
An Introduction to Statistics Through Biological Data
Shahbaba, B.
2012, XVI, 352 p. 162 illus., 73 illus. in color., Softcover
ISBN: 978-1-4614-1301-1