

## 17.3: Introduction to Programming

---

In many respects, R is a programming language similar to other languages such as Java, Python, and others. As such, it comes with a terminology that may be unfamiliar to most readers. In this section we introduce some of this terminology in order to give readers the working knowledge necessary to utilize the rest of the book to the best of its ability. One particular thing to note is that R is an **object oriented** programming language. This means the program is organized around the data we are feeding it, rather than the logical procedures used to manipulate it. This introduces the important concept of **data types and structures**. For R, and programming languages generally, there is no agreed upon or common usage of the terms **data type** versus **data structure**. For the purposes of this book, we will attempt to use the term **data structure** to refer to the ways in which data are organized and **data type** to the characteristics of the particular data within the structure. Data types make up the building blocks of data structures. There are many data types; we will cover only the most common ones that are relevant to our book. The first is the **character** type. This is simply a single Unicode character. The second is a **string**. Strings are simply a set of characters. This data type can contain, among other things, respondents' names and other common text data. The next data type is the **logical** type. This type indicates whether or not a statement or condition is True or False. It is often represented as a 0/1 in many cases. Finally, there are **numerical** data types. One is the **integer** which is, as you may recall, a number with nothing after the decimal point. On the other hand, the **float** data type allows for numbers before and after the decimal point.

In R, there are a plethora of data structures to structure our data types. We will again focus on a few common ones. Probably the simplest data structure is a **vector**. A vector is an object where all elements are of the same data type. A **scalar** is simply a vector with only one value. For the purposes of this book, a variable is often represented as a vector or the column of a dataset. **Factors** are vectors with a fixed set of values called levels. A common example of this in the social sciences is sex with only two levels—male or female. A **matrix** is a two dimensional collection of values, all of the same type. Thus, a matrix is simply a collection of vectors. An **array** is a matrix with more than 2-dimensions. The data structure we will use most is a **dataframe**. A dataframe is simply a matrix where the values do not all have to be the same type. Therefore, a dataframe can have a vector that is text data type, a vector that is numerical data type, and a vector that is a logical data type or any possible combination. Finally, **lists** are collections of these data structures. They are essentially a method of gathering together a set of dataframes, matrices, etc. These will not commonly be used in our book but are important in many applications. Now that we have covered the basic types and structures of data, we are going to explain how to load data into R.

---

This page titled [17.3: Introduction to Programming](#) is shared under a [CC BY 4.0](#) license and was authored, remixed, and/or curated by [Jenkins-Smith et al. \(University of Oklahoma Libraries\)](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.