This module will cover the basics of programming in R, with particular attention to getting started in R Studio and a few different kinds of data objects we will use throughout the course of these modules.

Lines 3 to 18 provide a quick overview of the R studio interface. Your script is displayed in the upper left. The console is in the lower right. The upper right window displays the global environment. The lower right window can display a variety of different things including your working directory, data visualizations, software packages, and other help queries.

Lines 20 to 40 provide an overview of how to identify and change your working directory. This is the folder that R will check for files when reading in data and also export your output files to.

It’s worth noting that the directories printed in the example script correspond to my computer, and these will need to be changed depending on the output of the command getwd(). If R returns an error saying “Cannot change working directory” the path name may have a typo.

You can recover the last command you entered by typing the UP arrow key.

Lines 42 to 67 demonstrate how to read a data file into R. The read.table() command will read in the file adult19.csv, provided it is saved in your working directory, as an object in R named df. There are also a few commands here to check the dimensions and column names of the file to check that it was read in correctly.

Lines 68 to 101 demonstrate how to install, load, and troubleshoot software packages in R. Software packages include additional functionality that build on the basic commands that are available in R. We’ll use a number of packages throughout these tutorials, but here just install and load the psych package.

Lines 103 to 166 provide an introduction to a few different object classes in R. R is an object-oriented programming language, meaning that essentially anything that can be done in R revolves around manipulating different data objects. Here, we focus on dataframes for loading datasets in R and a few kinds of column object classes: numeric, character, and factor.

Column class determines what sort of operators R will perform on the column; for example, line 132 presents an example of a basic math operation used on a numeric column. Math operations can be used to apply data transformations; for example log transformation can sometimes be used to make data more normally distributed.

For text data, we can use character class objects or factors. Character class objects store literal text strings, while factors include additional elements such that they can store data as labeled numbers. Factors interact with many functions in R differently than character class objects and their use is a matter of preference. In these tutorials, we will not use factors if character class objects are a viable alternative.

Lines 167 to 201 discusses the brackets operator, which can be used to access the rows and columns of a dataframe. In brackets, rows are indexed on the left of the comma and columns are indexed on the left of the comma. The example script also goes over a few different ways to provide arguments to the brackets operator, indexing either by number or text labels.

Lines 202 to 219 describe the List object class. Lists are a way to combine ANY series of objects in R. Since they can be used to combine any series of objects, lists are very flexible for storing output files. This can be useful for streamlining some tasks in R and will be discussed in a later tutorial.

Line 223 provides a command to exit your R session. Your script can be saved by clicking the floppy disc icon the upper right-hand corner. COMMAND + S on Mac or CTRL + S on Windows will also save your script.