LA-3: Working with Data (15 points)

Learning Outcomes

In this assignment, you will learn how to:

- Create a new R script.
- Prepare your R script for your assignment.
- Create headers in your R script.
- Install and load packages in R.
- Open a help page in R.
- Manually enter data into R.
- Look at the contents of a data frame or tibble.
- Identify variables and their types in a tibble.

Setup

1) Create a new R script and set it up for your assignment. In the first four (4) lines of your script, include the following, each on a new line, as a comment:

- Full name
- Student ID (u########)
- Date (MM/DD/YY)
- Assignment name

> 💡 Tip
>
> Comments (text preceded by **#** in your R script) are not run or executed. R will only run/execute commands or functions (text not preceded by **#**).

2) Make a section of code titled "Installing and loading packages." Under this heading, write pseudocode in preparation to install and load the following packages:

- `tidyverse`
- `summarytools`
- `rstatix`

> 💡 Tip
>
> Pseudocode is the plain English version of your R code that described the steps you are taking in R. It should be written as comments in your R script and should make sense to you. Your pseudocode is a tool that helps you decide which R function or command you need to accomplish a given task.

3) Install the packages above using the function, `install.packages()`. For example, to install the `tidyverse` package, the command is `install.packages("tidyverse")`. Remember that typing the command in your R script does not run or execute the function. You must select the text of the function that you want to run and click "Run" in the top right corner of the script pane (or use the shortcut keys).

4) Load the packages. The function to load the packages is `library()`. To load the `tidyverse` package, type `library(tidyverse)` in your script and run it. Note that for each new project you create in Your Workspace on Posit Cloud, you will have to install and load any necessary packages again.

Entering Data Manually

In the next step, we will enter data into R manually using the `c()` function, which is short for `combine()`. Before we do so, let's learn about the function we will be using.

1) Create a new section of code where you can explore and learn about the `c()` function.

2) To read the documentation for any function, you can add `?` in front of the command and run it. This will pull up the documentation (including examples at the very end of the documentation) in the Help tab in the bottom right pane. For example, if I wanted to read the documentation for the `library()` function, I would run the command: `?library()`. Copy and paste the description from the Help pane for the `c()` function in your R script as a comment.

> 💡 Tip
>
> The documentation may not seem helpful at first glance, but once you become more familiar with R and its commands, the Help documentation becomes a useful resource.

3) Enter the data in the table below as four (4) objects called `Aname`, `Ateam`, `PPG` (points per game), and `REB` (rebounds per game). There are two data types in the data table, numeric and character. Your TA will show you how to assign the data to the objects.

Table 1: 2023-24 data of top 5 women's college basketball players by points per game (PPG).

Aname	Ateam	PPG	REB
C. Clark	Iowa	31.6	7.4
J. Watkins	USC	27.1	7.3
L. Olsen	Vill	23.2	4.8
C. Watson	Buffalo	23.0	5.1
E. Kitley	VT	22.8	11.4

4) Check that you have created the objects correctly by calling each object. Calling an object in R allows you to see it in the Console. To call an object, type the name of the object in the Console and press Enter. Or type the name of the object in your R script, then highlight and run it. Copy and paste the results of calling the objects into your R script as comments. You can find the output in the Console pane.

5) Combine the objects into a single data frame that has a name of your choosing and look at the data frame. Use the `tibble()` function to do this. The data in the data frame should match that in Table 1. Call the data frame in the Console. Copy and paste the output as a comment in your R script.

Examining Types of Data

1) Now, we will examine the type of data stored in each column of our data frame. To do this, we will use the `class()` function. Let's start by reading the documentation for the `class()` function. Copy and paste the description of this function from the Help pane as a comment in your R script.

2) Determine the class of each column in your data frame. For example, if I wanted to check the class of `Aname`, I would use the function `class()` with the arguments: `df$Aname`. The whole command would look like this: `class(df$Aname)`. Copy and paste the class of each column in your R script.

The `learnr` Tutorial

In this last section, we will use a Tutorial that is built into an existing R package.

Install and load the `learnr` package in your current session of R. In the Tutorial tab (usually top right with the Environment pane), complete the **Data Basics** tutorial. If you are asked to install packages or dependencies, do so.

Include the answers to the questions below in your R script as comments. Clearly label your answers with the question number (e.g., a, b). You will answer these questions as part of the Data Basics Tutorial. List the answers from your tutorial as comments in your R script.

a) What does the `drv` variable of `mpg` describe? Read the help for `?mpg` to find out.
b) How many rows are in the data frame named `cars`?
c) How many columns are in the data frame named `cars`?
d) Which types of variables does `flights` contain? List all the types that you checked in the tutorial.

Submission

Submit your R script (named `LA-#_FirstName-LastName.R`) to Canvas.

Your R script should:

1) Include commands and functions that are necessary to address all the questions in the assignment.
2) Contain comments that answer the questions in the assignment.
3) Run in its entirety without errors.

To ensure that your R script runs without errors, you should:

- Save your script.
- Navigate back to Your Workspace on Posit Cloud.
- Reopen your project.
- Run the entire script line-by-line without editing it to ensure there are no errors.

> **!** Important
>
> These standards apply to all submissions in this course that require R scripts. You should follow these instructions for preparation, naming, and saving of your R script for all of your individual lab assignments.