Data Analysis Assignment 1

COMM 7370 | Spring 2023 | Due: see syllabus

In this assignment, you will learn how to:

- Prepare your R script for your assignment.
- Create foldable/collapsible headers in your R script.
- Install and load packages in R.
- Open a help page in R.
- Manually enter data into R and read/load data into R from a .csv file.
- Identify variables and their types in a tibble.
- Use the pipe operator,`|>` or `%>%`.
- Select variables from a data frame using the `select()` function.
- Filter cases using the `filter()` function.
- Obtain descriptive statistics for variables using the `descr()` function.

Instructions

1) Read Ch. 3 of COMM 3710: Getting Started with R. Then, create a new R script. Prepare your R script. In the first four (4) lines of your script, include the following, each on a new line, as a comment:

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

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

2) Make a section of foldable 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 be logical.

> Warning
>
> While you are welcome to work with another student on your assignments, your pseudocode **should be in your own words**. Identical submissions will be flagged as plagiarism, which will be dealt with in accordance with the course policy.

Then, write commands to **install** the above packages and **load** them into your current session of R.

> Tip
>
> Sections 2.3.5 and 2.3.6 in COMM 3710: Getting Started with R should help.

3) Save your R script as `LA-3_FirstName-LastName.R`.

4) 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. Create a new foldable section in your code for this–give it an appropriate heading. Write pseudocode to pull up the help documentation for the `c()` function (e.g., "exploring the `c()` function before using it"). Then, write and run the R command (`?c`) that opens the documentation in your R script. Use the results to describe the `c()` function from the documentation as a comment under your `?c` command. You can copy and paste the text under "Description" from the documentation, but I encourage you to add your own description so that you understand what the function is doing.

> Tip
>
> Typing `?` before any function in R will open the documentation for that function. I recommend that you use this any time you are unsure what a R function does. For example, if I wanted to learn more about the `library()` function, I would type and run the command, `?library`.

5) Time to enter data! Write pseudocode to enter the data in the table below (Table 1) as four (4) objects called `Aname`, `Ateam`, `ppg`, and `pts`, respectively. Writing pseudocode as comments before actual code should become a habit as it will help you understand how to write appropriate code for data analysis. After writing code in your R script, remember to **run** it.

Table 1: Data from 2021 of top 5 women's college basketball players by points per game.

Aname	Ateam	ppg	pts
C. Clark	Iowa	26.6	799
A. Hayes	Middle Tenn	26.4	663
C. hooks	Ohio	25.1	628
K. Bell	FGCU	24.3	632
A.Joens	Iowa State	24.2	678

There are two types of data in Table 1, numeric and character. To enter these data into objects in R as the appropriate type of data, review Section 2.4 in the book.

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. Write pseudocode for this, followed by the R command to call each object (`Aname`, `Ateam`, `ppg`, `pts`).

Then, combine the objects into a single data frame that has a name of your choosing and look at the dataframe. Use the `tibble()` function to do this. Again, pseudocode and R code should be included in your script. The data in the tibble should match the information in Table @tab-data.

> **Tip**
>
> 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.

The pseudocode and R code below is generic. This means that you will have to replace the placeholders (e.g., `obj1`, `obj2`, `df`) with names of the objects in your R environment.

```
# Calling obj1 to check that I have created it correctly
obj1

# Put obj1, obj2, etc. into a tibble (tidyverse-style data frame)
df <- tibble(obj1, obj2, obj3, ...)
```

6) 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. First, write pseudocode and R code to examine the `class` function. Copy and paste the description of this function as a comment in your R script. Then, write pseudocode and R code to check the class of each column in your data frame. To reference a column in a data frame, we use one type of extractor operator, `$`. Let's say I have a data frame called `cats` and I wanted to examine the data type of a column within the `cats` data frame called `cuteness`. To do so, I will use the class function and an extractor operator: `class(cats$cuteness)`.

   Run the code you just wrote and note the class of each column of your data frame in a comment. If necessary, review the brief section in the book about data types.

Optional

If you feel you need more practice, there is a tutorial in an existing R package. Install and load the `learnr` package in your current session of R. In the **Tutorial** panel (usually top right with the **Environment** panel), complete the **Data Basics** tutorial.

7) Next, we will learn to read data into R. We seldom enter data manually–instead, we often have a data file, typically saved as a `.csv` file, that we load into R when we are ready to conduct data analysis.

   Download the HELP dataset and save it to your working directory. Read the dataset into an object in R–give it a name of your choice; feel free to be creative but note that you will likely be typing this name repeatedly during data analysis so do not make it too long or cumbersome.

```
# To read data into R, use the read_csv() function

object <- read_csv("HELP.csv")
```

   The codebook for the HELP dataset can be found in Table 2. A codebook provides information on the structure, contents, and layout of a data file.

Table 2: : Codebook for the HELP dataset.

Variable	Description
`age`	R age at baseline
`anysub`	Use of any substance post-detox (no, yes)
`cesd`	Center for Epidemiologic Studies Depression measure at baseline (higher scores indicate more depressive symptoms)
`d1`	Lifetime number of hospitalizations for medical problems (measured at baseline)
`daysanysub`	Time (in days) to first use of any substance post-detox
`dayslink`	Time (in days) to linkage to primary care
`drugrisk`	Risk Assessment Battery drug risk scale at baseline
`e2b`	Number of times in the past 6 months R entered a detox program (measured at baseline)
`female`	Biological sex (female coded high)
`sex`	Biological sex (male, female)
`g1b`	Experienced serious thoughts of suicide in last 30 days (measured at baseline)
`homeless`	Housing status (housed, homeless)
`i1`	Average number of drinks consumed per day in past 30 days (measured at baseline)
`i2`	Maximum number of drinks consumed per day in past 30 days (measured at baseline)
`id`	R identifier
`indtot`	Inventory of Drug Use Consequences (InDUC) total score (measured at baseline)
`linkstatus`	Post-detox linkage to primary care (yes coded high)
`link`	Post-detox linkage to primary care (no, yes)

Variable	Description
mcs	SF-36 Mental Component Score (measured at baseline; lower scores indicate worse status)
pcs	SF-36 Physical Component Score (measured at baseline; lower scores indicate worse status)
pss_fr	Perceived social support by friends (measured at baseline; higher scores indicate more support)
racegrp	Race/Ethnicity (black, hispanic, white, other)
satreat	Any BSAS substance abuse treatment at baseline (no, yes)
sexrisk	Risk Assessment battery
substance	Primary substance of abuse (alcohol, cocaine, heroin)
treat	Randomized to HELP clinic (no, yes)

8) Now, we will learn to use the pipe operator, `|>` or `%>%`. The former is native to RStudio while the latter is part of the `tidyverse` package. they are a little different but, for the most part, you should be able to use them interchangeably. For now, we will use the `magrittr` version, `%>%`.

   You can think of `%>%` as "then." The pipe operator pipes the contents from the left-hand side of the pipe to the right-hand side. If we used the pipe operator to describe a typical daily routine, it might look something like this:

```
Woke up %>%
    Took a shower %>%
    Got dressed %>%
    Made breakfast %>%
    Ate breakfast %>%
    Went to work
```

   Let's work through an example using the HELP data. Determine the average number of drinks consumed per day in the past 30 days by males and females (use the variable, `sex`; refer to Table Table 2 as needed). Answer the question below in a comment in your R script (be sure to label this clearly in the script); step-by-step instructions to help you answer this question are shown below.

   a) What is the name of the variable that measures the average number of drinks consumed per day in the past 30 days?

   Write pseudocode to determine the average number of drinks among females (remember that lines that begin with # are comments in R):

```
# Start with the HELP data frame, which I called hdata
# Select female-only cases from the HELP data using the sex variable
# Use descr() to get descriptive statistics
```

   Next, translate the pseudocode to R functions using the pipe operator:

```
hdata %>% # Start with the HELP data frame
        filter(sex == "female") %>% # Select female-only cases
        descr(i1) # Use the descr() to get descriptive statistics
```

b) What is the minimum, maximum, and mean number of average drinks consumed per day in the past 30 days among females in the sample? ““

9) Using the same process from the previous step, determine the minimum, mean, and maximum number of average drinks consumed per day in the past 30 days among males in the sample.

10) Using the pipe operator and the functions you learned, determine the minimum, maximum, and mean SF-36 mental component score among males and females in the sample whose primary substance of abuse is alcohol.

Submission

Submit your R script (which should have a `.R` extension) to Canvas. Your R script should:

1) Include code to install and load the packages.
2) Contain comments and/or pseudocode.
3) Run in its entirety without errors.

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

- Save your script.
- Completely shut down RStudio or restart your R session.
- Reopen RStudio and your `.R` script.
- Run the entire script by clicking the “Run” button in the top right of the R script.

❗ 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 data analysis assignments.