Day 1 - R Basics

Part 1 - Practice!

\* If you get stuck, feel free to ask your peers in your breakout group or refer to the corresponding sections in the lesson template code.

1. Introduction
   1. Open up the file named “Wave 5 RShiny Day 1 Workshop.R” in RStudio.
   2. Introduce yourself to the other students in your breakout group (First name, Hometown, favorite number).
   3. Write this information in the script as a comment.
   4. Print the following message to the console using one print command:

“[Name] is from [Hometown] and their favorite number is [Number].”

1. Data Types
   1. Create a new variable called “name” and assign to it a string that contains one of the students’ names in your breakout group.
   2. Create a new variable called “hometown” and assign to it a string that contains the student’s hometown.
   3. Create a new variable called “number” and assign to it the value of the student’s favorite number.
   4. Convert the value inside the variable called “number” from a number to a string. For example, if “number” contains the value 10, make it now have the value “10”.
   5. Print the following message to the console by pasting together the information stored inside the variables “name”, “hometown”, and “number”.

“[Name] is from [Hometown] and their favorite number is [Number].”

1. Data Structures
   1. Create three vectors, each containing information about the students in your breakout group: one consisting of all the students’ names, one consisting of all the students’ hometowns, and one consisting of all the students’ favorite numbers.
   2. Combine these vectors into a matrix, where each row represents different students’ information.
      1. Rename the row and column headers to reflect the information stored inside this matrix.
      2. Print everyone’s favorite number by accessing the corresponding information in the matrix in two different ways.
      3. Print your favorite number by accessing it in the matrix.
   3. Combine these vectors into a list.
      1. Rename the list elements to reflect the information contained in the list.
      2. Print everyone’s favorite number by accessing the corresponding information in the list in two different ways.
      3. Print your favorite number by accessing it in the list.
2. Conditionals and Functions (Challenge!!!)
   1. Create your own greeting function that takes as inputs a name, a hometown, and a number and outputs the message:

“[Name] is from [Hometown] and their favorite number is [Number].”

Example. INPUT: (Lucas, Basel, 8)

OUTPUT: “Lucas is from Basel and their favorite number is 8.”

* 1. Create your own greeting function that takes as inputs vectors of names, hometowns, and numbers and outputs the message:

“[Name 1] is from [Hometown 1] and their favorite number is [Number 1].”

“[Name 2] is from [Hometown 2] and their favorite number is [Number 2].”

“[Name 3] is from [Hometown 3] and their favorite number is [Number 3].”

… and so on …

Example. INPUT: (c(Lucas, Karina), c(Basel, Los Altos), c(8, 23)),

OUTPUT: “Lucas is from Basel and their favorite number is 8.

Karina is from Los Altos and their favorite number is 23.”

* 1. Create your own greeting function that inputs vectors of names, hometowns, and numbers and outputs the message:

“[Name] is from [Hometown] and their favorite number is [Number].”

for each set of inputs whose corresponding favorite number is odd.

Example. INPUT: (c(Lucas, Karina), c(Basel, Los Altos), c(8, 23)),

OUTPUT: "Karina is from Los Altos and their favorite number is 23.”

Part 2 - Apps! (Bonus)

In tomorrow’s lesson, we will start creating an app related to prime factorization. Please write the following functions that will be implemented in the app.

1. Create a function that takes as an input an integer and outputs a list of its factors

Example. INPUT: 24, OUTPUT: "The factors of 24 are: 1,2,3,4,6,8,12,24."

1. Create a function that takes as an input an integer and outputs its prime factorization

Example. INPUT: 360, OUTPUT: "The prime factorization of 360 is: 2 x 2 x 2 x 3 x 3 x 5."

1. Create error handling messages for these functions so that these functions only work for integers greater than 2.

Example. INPUT: -1, OUTPUT: "Error. Input must be an integer greater than 2."

Example. INPUT: 2.5, OUTPUT: "Error. Input must be an integer greater than 2."

Example. INPUT: "Hello world", OUTPUT: "Error. Input must be an integer greater than 2."