Lab 1

Daniel Chen

2024-08-08

Lab 1

Working with Vectors

There are various ways to generate data. Here is an example of how to create a vector of all integers from 1 to 10.

```
example_nums = 1:10
example_nums
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

In discussion 1 we saw how vectors can be indexed to extract the individual values from a vector. We can also use indexing to get multiple values from a vector.

The first method of indexing is to specify each index in the form of a vector. Below are some examples.

```
# Obtain the 2nd and 5th item in a vector
example_nums[c(2, 5)]
```

```
## [1] 2 5
```

```
# Obtain the first 5 items in a vector
example_nums[1:5]
```

```
## [1] 1 2 3 4 5
```

You can also index a vector based on logical comparisions

```
# Obtain all numbers greater than 5
example_nums[example_nums > 5]
```

```
## [1]  6  7  8  9 10
```

```
# Obtain all numbers not equal to 9
example_nums[example_nums != 9]
```

```
## [1]  1  2  3  4  5  6  7  8 10
```

Problem 1: Create a variable named `nums` with a value equal to a vector of all integers from 1 to 100. Print out the 10th number in the vector.

```
nums =
nums[10]
```

Problem 2: Create a variable named `nums2` with a value equal to a vector of all integers from -10 to 10. Print out the third, fourth, and fifth items in `nums2` using indexing.

Problem 3: Print out only the negative items in `nums2` using indexing

Working Directories and Reading data into RStudio

A working directory is the default folder that R works with. You can check which folder is the current working directory using the `getwd()` function.

```
getwd()
```

It is good practice to create a folder where you will be saving all your data and code and set that folder to your working directory. If you use a file name in your code, R will automatically look in your working directory for that file.

Problem 4: Create a folder called "Stats 13" and save this file to the "Stats 13" folder.

Problem 5: On the top of your screen click: "Session > Set Working Directory > To Source File Location" Run the `getwd()` function to show that your current working directory is now the "Stats 13" folder.

```
getwd()
```

Problem 6: Download the file AgeBMI.txt from the disscussion section website and save it to the "Stats 13" folder. To load the data into RStudio, run the following code chunk.

```
AgeBMI = read.table("AgeBMI.txt", header = TRUE)
```

Problem 7: You should see that there is a variable named `AgeBMI` in the environment panel now. You can also inspect the data by using the `head()` function to view the first few rows of the data. Run the following code block. Describe what kind of data is contained in `AgeBMI`.

```
head(AgeBMI)
```

Problem 8: You can access the data in the `Age` column by typing `age = AgeBMI$Age`. Create a variable called `age` containing a vector of all the data in the `Age` column of `AgeBMI`. Create a variable called `bmi` containing all the data in the `bmi` column of `AgeBMI`.

Summarizing data

Problem 9: Using the `mean()` function the same way we did in disscussion 1, find the average age in this dataset.

Problem 10: Create a variable called `bmi_under_29` containing information about whether each BMI is less than 29 or not. (Hint: Use indexing)

Problem 11: Using the `table()` function we can tabulate the results of our comparison to see how many people in this dataset have a BMI under 29 and how many people have a BMI over 29. Run the code `table(bmi_under_29)`. Interpret the result. What does each number mean?