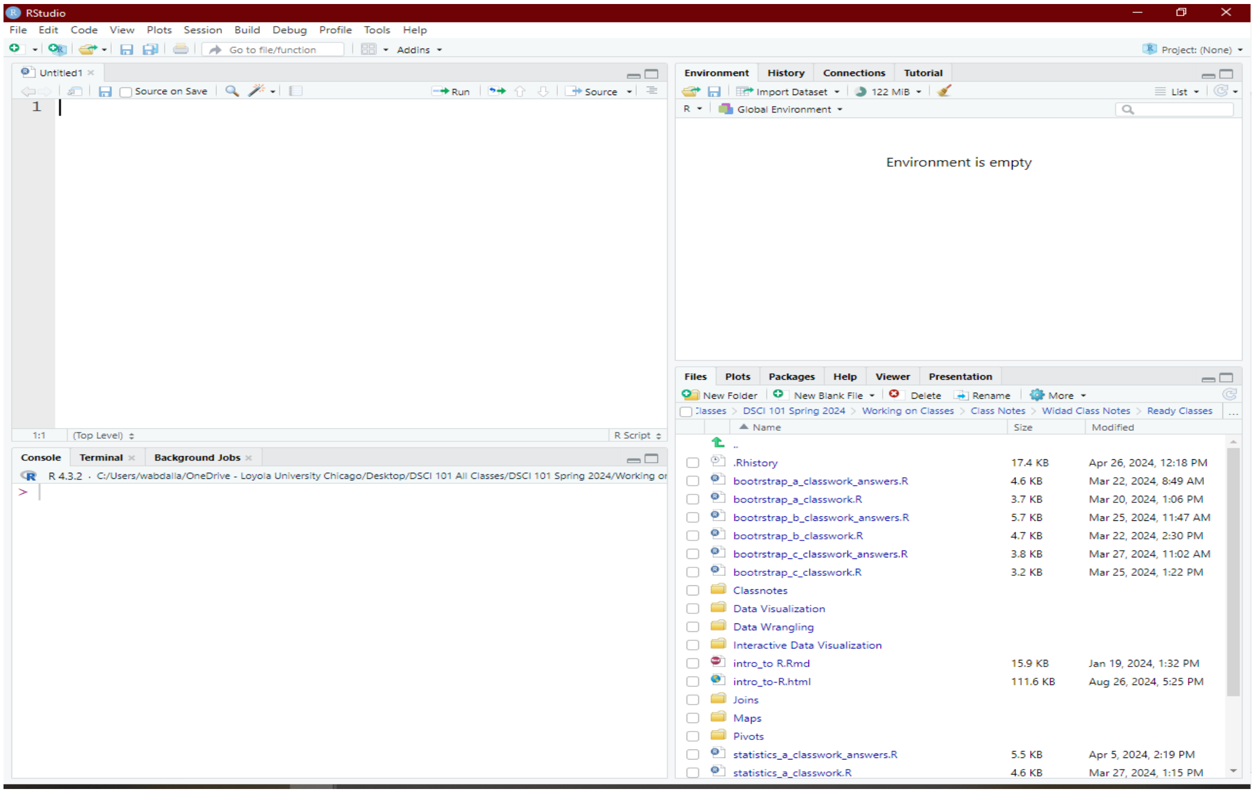
Coding Basics

Part 1 – Introduction to R & RStudio

R & RStudio - Basic Components



Environment is empty

```
R 4.3.2 · C:/Users/wabdalla/OneDrive - Loyola University Chicago/Desktop/DSCI 101 All Classes/DSCI 101 Spring 2024/Working or
>
```

Name	Size	Modified
..		
.Rhistory	17.4 KB	Apr 26, 2024, 12:18 PM
bootrstrap_a_classwork_answers.R	4.6 KB	Mar 22, 2024, 8:49 AM
bootrstrap_a_classwork.R	3.7 KB	Mar 20, 2024, 1:06 PM
bootrstrap_b_classwork_answers.R	5.7 KB	Mar 25, 2024, 11:47 AM
bootrstrap_b_classwork.R	4.7 KB	Mar 22, 2024, 2:30 PM
bootrstrap_c_classwork_answers.R	3.8 KB	Mar 27, 2024, 11:02 AM
bootrstrap_c_classwork.R	3.2 KB	Mar 25, 2024, 1:22 PM
Classnotes		
Data Visualization		
Data Wrangling		
Interactive Data Visualization		
intro_to R.Rmd	15.9 KB	Jan 19, 2024, 1:32 PM
intro_to-R.html	111.6 KB	Aug 26, 2024, 5:25 PM
Joins		
Maps		
Pivots		
statistics_a_classwork_answers.R	5.5 KB	Apr 5, 2024, 2:19 PM
statistics_a_classwork.R	4.6 KB	Mar 27, 2024, 1:15 PM

Console Window

- located in the bottom left.
- It’s where you often will find the output of your coding and computations.
- It is also possible to write code directly into the console.

Source Window

- Located in the top left.
- “Source” can be understood as any type of file.
- Function of the source window include:
  - ✓ Inspect data in an Excel-like format.
  - ✓ Open programming code and code editing.
  - ✓ Run the analysis you have written.

Environment/History/Connections/Tutorial Window

- Located in the top-right.
- Shows multiples panes, but the most important one is “Environment”.
- Environment: shows you objects which are available for computation. Some examples of objects include:
  - ✓ Dataset
  - ✓ Vectors
  - ✓ Matrices
  - ✓ Lists
  - ✓ Functions you create yourself.

Files/Plots/Packages/Help/Viewer Window

- Located in the bottom-right.
- It consists of five essential panes:

✓ ***Files pane:*** lists all the files and folders in your root directory.

✓ ***Plots pane:*** This pane is exclusively designed to show you any plots you have created using R.

  - o To delete a plot, you can click on the red circle with a white x symbol.
  - o To remove all plots, you can use the broom.
  - o There are options to export your plot and move back and forth between different plots.

✓ ***Packages:*** they are additional tools you can import and use when performing your analysis.

  - o A frequent analogy people use to explain packages is your phone and the apps you install. Each package you download is equivalent to an app on your phone.
  - o It can enhance different aspects of working in R, such as creating animated plots, or simply making your life easier by doing multiple computations with just one single line of code.

✓ ***Help pane:*** you can search for specific topics, for example how certain computations work. The Help pane also has documentation on different datasets that are included in R, RStudio or R packages you have installed.

Customize your Interface

Go to Tools > Global Options > Appearance

Packages

- Many of R's most useful functions do not come preloaded when you start R, but reside in packages that can be installed on top of R.
- An R package bundles together useful functions, help files, and data sets.
- You can use these functions within your own R code once you load the package they live in.
- You only need to install a package once in your computer.

code:

- You need to load the package every time you open R.

code:

File Types

- R Script
  - ✓ Extension: .R
  - ✓ This is a file you can write your codes in it.
- RMarkdown Script
  - ✓ Extension: .Rmd
  - ✓ Incorporates code and texts together to *knit* a document together.
  - ✓ Knitting an rmd file compiles the text and code together to create another file such as an .html or .pdf
- R Project
  - ✓ Extension: .Rproj
  - ✓ Helps you organize your different coding projects.
  - ✓ The working directory will automatically be set to the directory that the .RProj file is located in allowing easier organization and access to your data and files.

Data Structure

Data Structure is a specific way to organize, store, process, and retrieve your data.

- **Vectors**

Examples of vectors:

[4, 1, 3, 8, 6, 7, 5, 3, 0, 9]

Code:

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

Code:

[x.1, x.2, x.3, x.4, x.5]

Code:

- **Matrices**

Examples:

$$\begin{bmatrix} 4 & 6 \\ 7 & 8 \\ 9 & 10 \end{bmatrix}$$

code:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

code:

Part 2 – Data Structure, Classes, and Functions

More on Data Structures

Subsetting Vectors

Command: `name_of_vector[position/s you want to extract]`

Example 1: For the following vector:

```
vector_1 <- c(2,3,4,5,5,6)
```

- What code do you need to write to extract the third element?

Code: **Output:**

- What code do you need to write to extract the fourth through sixth element?

Code: **Output:**

Subsetting Matrices

Since matrices are two-dimensional organization of data, to subset a matrix you need a number for the row and a number for the column.

Command: `name_of_matrix[row number, column number]`

Example 2:

- The following code defines a matrix, write out what the matrix will "look like" if we ran that code.

```
matrix_1 <- matrix(c(5,6,7,2,3,6), ncol = 2, byrow = FALSE)
```

- Extract the element in row 3, column 1.

Code:

Output:

- Extract the element in column 2, row 1.

Code:

Output:

- Extract elements in column 2

Code:

Output:

- Extract all elements in column 1

Code:

Output:

- Extract all elements in row 2

Code:

Output:

Classes (Data Type)

1. numeric: any real number(s)
2. character: strings or individual characters, quoted
3. integer: any integer(s)/whole numbers
4. factor: categorical/qualitative variables
5. logical: variables composed of TRUE or FALSE
6. Date/POSIXct: represents calendar dates and times

How to check the class of a vector?

Command: `class(name of the vector)`

Example 3: In the following table, I define 5 different vectors. Type the code you would use to check the class of each vector and in the third column, type what the output would be if you ran that code.

Vector	Code to Check each vector's class	Output
`x <- c(1,2,3,4,5)`		
`y <- c("mary", "joe", "john")`		
`k <- c("1", "2", "3")`		
`z <- factor(c("A", "B", "C", "A", "A", "B"))`		
`w <- c(FALSE, TRUE, TRUE)`		

How to change a vector from one class to another class?

Use commands the following commands:

- To change to a numeric vector: as.numeric(name of the vector)
- To change to a character vector: as.character(name of the vector)
- To change to a factor vector: as.factor(name of the vector)
- To change to a logical vector: as.logical(name of vector)

Example 4: Suppose you have the following vector

```
vector_2 <- c("3", "5", "7", "9")
```

- What is the class of this vector as it is?

Code:

Output:

- How would you change it to numerical?

Code:

Dataframes

Dataframe: is a rectangular collection of variables (in the columns) and observations (in the rows). You can think of every column in a dataframe as a vector.

How to define dataframe?

Example 5: Define the following dataset as a dataframe.

age	month	year
18	1	2006
18	2	2006
19	5	2005
20	12	2004
18	1	2006

Code:

Example 6: Define the following dataset as a dataframe.

V1	V2	V3
1	1	1
2	1	3
3	1	5
4	1	7
5	1	9

Code:

Subsetting dataframes

Command: `name_of_dataframe$name_of_column`

Example 7: What code would you write to pull/subset the column year from the dataframe in Example 5.

Code:

Output:

Example 8: What code would you write to check the class of the second column from the dataframe in Example 6.

Code:

Output:

Functions in R

For today we will study the following functions:

mean, median, min, max, sd, summary

Command: `name_of_the_function(name of the vector)`

Example 9: Suppose you have the following vector:

```
x <- c(1,1,3,5,5,6,7)
```

- How would you compute the mean?

Code:

Output:

- How would you compute the minimum value?

Code:

Output:

- How would you compute the standard deviation?

Code:

Output: