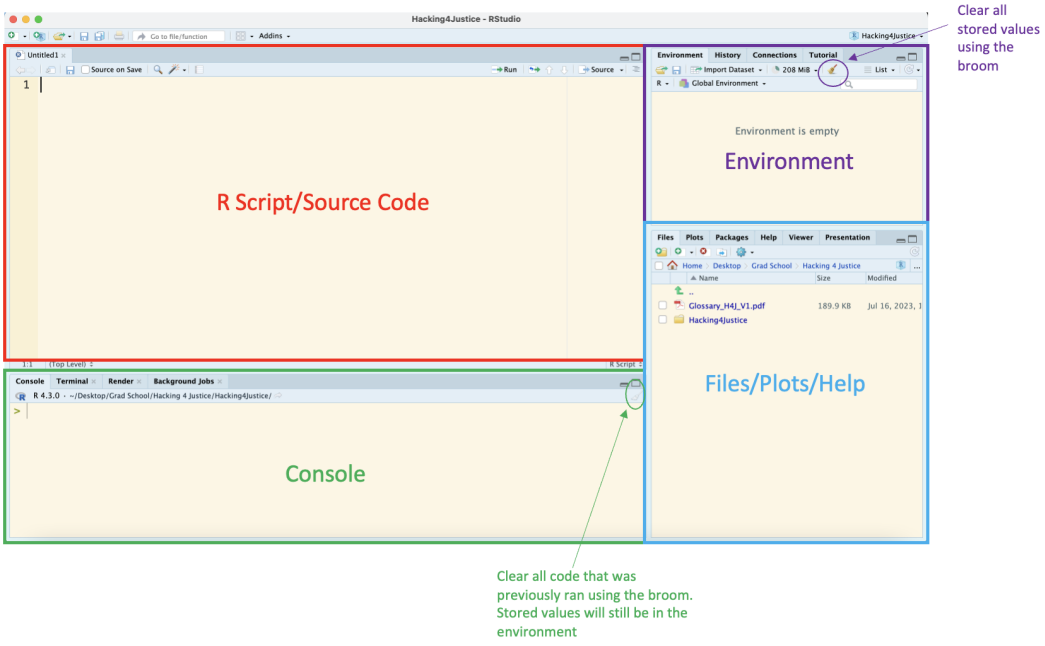
Glossary

Hacking4Justice

R Studio Environment

1. **R Script/Source Code**: By default, R Studio will populate this area with an untitled r script. This is where your code will go and the script can be saved to a folder on your computer.
2. **Environment**: All stored objects (Ex. constants, data frames, vectors, etc.) will be stored here. Data frames can be viewed in another window by clicking on the name of it here.
3. **Console**: The output of the code that you run in the script will show up in this pane. Code can also be run here, but it will not be saved for future use.
4. **Files/Plots/Help**: Some useful tabs in this pane is the "Files", "Plots", and "Help" tabs. The files tab will show the files in the working directory that has been set. When plots are created they will show in "Plots" tab.



Common Terms Associated With R

1. **Object**: A stored value in R. This could be a constant, vector, data frame, etc.
2. **Vector**: Multiple values of the same data type stored to one object. (The c() function is used to create a vector)

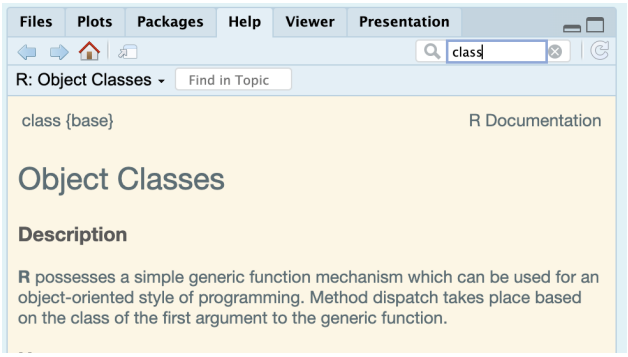
3. **Data Frame**: A table with rows and columns. Each column has a header or name.
4. **Working Directory**: This is where R Studio will look for files that you want to load into the environment and it is where R Studio will save

Data Types and Definitions

1. **Integer/Numeric**: This data type is going to be for categories that are numbers. (Ex. age, height, and weight)
2. **Characters**: This data type will be for "string" variables, or data that is in words, letters, and phrases. We tell R that the data type is a character or string by putting the value in "quotation marks". Both double or single quotation marks work, a general rule of thumb is to pick either single or double, and stick with the same type throughout the length of your R script. (Ex. name, address, charge, etc.)
3. **Logical**: Also known as, Boolean, this data type is for when we are working with TRUE and FALSE statements. (Ex. "Did the case go to trial?" or "Was a plea deal taken?")
4. **Factors**: Sometimes a variable might have categories or groupings. In this case, it is best practice to store the variable as a factor. When a variable is stored as a factor, R is then able to discern the different categories or buckets that is within the variable. Factors can also be used to create "labels" or "unique identifiers" for each row in a data frame. (Ex. Race, Gender, ID column, and zip codes)

Functions

Note: If you are ever unsure on what a specific function does, you can look up the function in the help tab on the lower right hand box in the R studio environment and it will provide a description.



1. **class()**: Returns the data type of the variable, constant, or vector.
2. **is.numeric()**: Tests whether a value/object/vector is numeric and returns a logical statement(TRUE or FALSE) for each element in the object. **is.character()**, **is.factor()**, and **is.logical()** can also be used to test whether an object is a character, factor, or logical data type, respectively.
3. **factor()**: Coerces, or "makes", a numeric or character vector into a factor.
4. **levels()**: This function is used with objects that are factors and can be used to see how many groups or buckets are within the vector.
5. **is.na()**: Tests whether objects within a data frame or vector has any NA values, or missing data. This will return a vector or matrix of logical statements.
6. **c()**: Creates a vector by combining multiple values of the same data type together.
7. **data.frame()**: Creates a data frame by combining vectors. The vectors will become the columns of the data frame.

8. **names()**: Returns the names of the columns in a data frame. This function can also be used to rename the columns of a data frame. (Ex. names(dataframe_name) <- c("newname_1", "newname_2", "newname_3"))
9. **str()**: Returns a summary of a data frame. This will return each variable, or column name, in the data frame and give us information about each of their data types.
10. **head()**: Returns the first number of a specified amount of rows of a data frame.(Ex. head(dataframe_name,5))
11. **getwd()**: This function will return the current working directory of R Studio.
12. **setwd()**: This function can be used to set your working directory to a different location on your computer.
13. **read.csv("file_name.csv", header = TRUE)**: Read a csv file into R Studio.
14. **install.packages()**: Install a package to R. Inputs to the function should be in "quotation marks".
15. **library()**: Once the desired package has been installed, then this function is used to load the desired package into the current R Studio session.
16. **nrow()**: Returns the number of rows in a data frame.
17. **ncol()**: Returns the number of columns in a data frame.
18. **tail()**: This is similar to the head function, except it will return the last number of specified rows in the data frame.
19. **summary()**: This is another function that will summarize the columns and data types in a data frame.
20. **table()**: Returns a frequency table from a data frame. (Ex. table(dataframe$column), table(dataframe$column1, dataframe$column2))
21. **duplicated()**: Determines which elements of a vector or data frame are duplicates and returns a logical vector indicating which elements(rows) are duplicates of others.
22. **sum()**: Adds all values in a numeric vector.
23. **subset()**: Returns a subset of a vector or data frame that meets given condition.
24. **barplot()**: Creates a bar plot with vertical or horizontal bars.
25. **hist()**: Creates a histogram (distribution of frequency) of a specified vector.

Logical Operators

1. > (greater than)
2. >= (greater than or equal to)
3. < (less than)
4. <= (less than or equal to)
5. == (equal to)
6. != (not equal to)
7. & (and)
8. | (or)
9. ! (not)