

Charlee Cobb - Transcriptomics, Exercise 1

netID: cac9995

Charlee Cobb

2023-01-16

2.1) How do you start a new R script in RStudio? - A new R script in RStudio is made by clicking “new script” in the file menu

2.2) Why is it not a good idea to use MS Word to save your commands? - because a lot of special characters in other documents written in pages or MS word don’t correctly format to the R console.

2.3) Where does R save the history of your commands? - The history of your commands is saved in the history tab

3.1) How do you know when R is ready for your command? - R is ready for your command when the greater than sign “>” pops up on the left of the screen.

3.2) How do you comment your code? - comments of code can be added by including a hashtag “#” in front of the words you write as a comment. #this sentence would be a comment.

3.3) What does it mean then R prompts you with a “+” in the beginning of the line? - the plus sign “+” at the beginning of the line indicates a continuation of the previous line of code.

4.1) What is an R workspace? How do you save it and how do you load it? - an R workspace is where your functions and variables are stored for a specific file.

4.2) What is a working directory? How do you set it and how do you determine which directory is the working directory? - the working directory is the directory in which the current R script / workspace reads/loads in and saves files. This directory also has the workspace and history. To set the working directory to a folder other than default, use the command “setwd(“newFolder”)“. To determine the working directory, use command “getwd()“.

4.3) How is an RStudio project related to the R workspace and Working directory? - In RStudio, you can read in various workspaces and directories for different projects. You can also use the Projects tab to set and save specific workspaces and directories that are the default for the specific project.

5.1) What is the command to install a package from CRAN? - To install a packages from CRAN, use the following command: `install.packages()`

5.2) What is the command to retrieve documentation related to a function and its usage? - To get documentation on the function and usage, use command “`help()`” or put a question mark “?” in front of the function name.

7.1) Create two integer vectors, one called GeneA with the values 3,4,5,7 and the other GeneB with the values 4,6,8,5

```
GeneA <- c(3,4,5,7)
GeneB <- c(4,6,8,5)
```

```
print(GeneA)
```

```
## [1] 3 4 5 7
```

```
print(GeneB)
```

```
## [1] 4 6 8 5
```

7.2) Create a character vector with the words, “control”, “nitrogen”, “potassium”, and “phosphate” and call it Experiments.

```
Experiments <- c("control", "nitrogen", "potassium", "phosphate")
print(Experiments)
```

```
## [1] "control" "nitrogen" "potassium" "phosphate"
```

8.1) Add vectors GeneA and GeneB and save it as GeneAB

```
GeneAB <- c(GeneA, GeneB)
print(GeneAB)
```

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

8.2) Add vectors GeneAB with Experiments. Do you get an error? Why or Why not? - No error arises, you can combine vectors GeneAB with Experiments. However, this changes the data type in GeneAB from numbers(numericals) to characters because it's an easier conversion.

```
GeneABEx <- c(GeneAB, Experiments)
print(GeneABEx)
```

```
## [1] "3" "4" "5" "7" "4" "6"
## [7] "8" "5" "control" "nitrogen" "potassium" "phosphate"
```

8.3) Add 1 to vector GeneAB and call it GeneABplusOne. Which value has changed and by how much? - when you add 1 to vector GeneAB, all the numbers in the vector increase by 1

```
print("GeneAB:")
```

```
## [1] "GeneAB:"
```

```
print(GeneAB)
```

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

```
GeneABplusOne <- GeneAB + 1
print("GeneABplusOne:")
```

```
## [1] "GeneABplusOne:"
```

```
print(GeneABplusOne)
```

```
## [1] 4 5 6 8 5 7 9 6
```

9.1) Create a matrix called expvalues where GeneA is the first row and GeneB is the second row.

```
expvalues <- rbind(GeneA, GeneB)
expvalues
```

```
##      [,1] [,2] [,3] [,4]
## GeneA    3    4    5    7
## GeneB    4    6    8    5
```

9.2) Assign the Experiments as the column names of the matrix expvalues.

```
colnames(expvalues) <- Experiments
expvalues
```

```
##      control nitrogen potassium phosphate
## GeneA      3         4         5         7
## GeneB      4         6         8         5
```

9.3) Subset the matrix to retrieve the control and nitrogen values for GeneA.

```
subset.expval <- expvalues[1, 1:2] #row 1 is GeneA
subset.expval
```

```
## control nitrogen
##      3         4
```

10.1) In what situation would you prefer a dataframe over a matrix? - A dataframe is preferred over a matrix when you want to store values that are different data types (ie. numericals and characters together)

10.2) In what situation would you prefer a list over a dataframe? - Lists are preferred in situations where you simply want a collection of objects, including dataframes, vectors, and matrices. They can also be different lengths

11.1) Convert the expvalues matrix into a dataframe and call it expvaluesdf

```
expvaluesdf <- as.data.frame(expvalues)
expvaluesdf
```

```
##      control nitrogen potassium phosphate
## GeneA      3         4         5         7
## GeneB      4         6         8         5
```

11.2) Give three different ways to retrieve the “potassium” column from expvaluesdf

```
get.potassium1 <- expvaluesdf["potassium"]
get.potassium1
```

```
##      potassium
## GeneA      5
## GeneB      8
```

```
#get.potassium2
print(expvaluesdf["potassium"])
```

```
##      potassium
## GeneA      5
## GeneB      8
```

```
#get.potassium3
print(expvaluesdf$potassium)
```

```
## [1] 5 8
```

12.1) What class of an object is created when using the read.table() function to import files?

- The class of an object made by the read.table function is a data frame because it assumes that not all the values are of the same type.

12.2) Use the help function to look at the details of how read.table() function works. Compare read.table() and read.delim(). Which default options are different for the two commands?

- In read.table() information, it's stated that this function reads a file in table format and makes a data frame from it. Each row of the table is one line from the file that is read in. For the read.delim(), it also creates a data frame object and represents one line of the file in each row. The difference in defaults is that the header is set to TRUE, and the sep value is a tab "\t" and not any whitespace " ".

```
help("read.table")
```

```
## starting httpd help server ... done
```

```
help("read.delim")
```

12.3) What does the stringsAsFactors option do? - According to the read.table() help documentation, stringsAsFactors is a logical argument that checks if character vectors should be converted to factors