Lab Exercise 2: Introduction to R

Hannah Roach

1/20/2019

Purpose:	This lab introduces you to the use of the R statistical package within the Data Science and Big Data Analytics environment. After completing the tasks in this lab you should able to: • Read data sets into R, save them, and examine the contents

Tasks:	Tasks you will complete in this lab include: • Invoke the R environment and examine the R workspace • Read tables created in Lab 1 into the R statistical package • Examine, manipulate and save data sets • Exit the R environment

References:	References used in this lab are located in your ***Student Resource Guide Appendix***. See the Appendix for: • R Commands – Quick Reference • Surviving LINUX – Quick Reference

Step	Action

1	**Invoke the R Environment:**
	
	Logon to RStudio environment.
	
	1. The RStudio is accessed through the “safari” browser available as a desktop icon
	2. Refer to the access details provided by your instructor RStudio is accessed with URL http:<Your assigned IP-address for the backend system>:8787/
	3. RStudio access Userid and Password are also provided by your instructor
	
	This will start your web browser and connects to the “be” server. You should see the RStudio four- panel display.
	
	1. Verify that you see the following text in the lower left-hand pane:
	
	`R version 2.15.0 (2012-03-30)`
	`Copyright (C) 2012 The R Foundation for Statistical Computing`
	`ISBN 3-900051-07-0`
	`Platform: x86_64-redhat-linux-gnu (64-bit)`
	
	`--- <snip> ---`
	
	`Type 'demo()' for some demos, 'help()' for on-line help, or`
	`'help.start()' for an HTML browser interface to help.`
	`Type 'q()' to quit R`
	`>`

2	**Examine the Workspace:**
	
	Type the following command into the R command panel, and hit [ENTER]
	
	`ls()`
	
	You should see the following:
	
	`character(0)`
	
	Note: R is telling you that you have nothing in your workspace.

Step	Action

3	**Getting Familiar with R**
	
	1. Click each tab in each panel. What happens?
	2. Type the following commands into the R command panel
	
	`help()`
	`help.start()`
	`demo()`
	`demo(graphics)`
	
	`Hit esc to exit out of the demo`

4	**Read-in the Lab1 Script** 1. Now, in the script window, open the script called “Module3Lab1.R”. (Click on “File”, “Open File” and Navigate to directory LAB02 and click on file “Module3Lab1.R”.) 2. All the commands we will be executing in this lab are contained in this script. In order to execute a command, do the following: o Position your cursor inside the line that represents the command you wish to execute. o Either click on the “Run” button, or hit [CTRL-Enter]. You can execute many commands at once by selecting a sequence of commands and then issuing the “Run” command. 3. The command will be executed in the command pane. If the command produces graphical output, it will appear in the graphic frame. Note that you can expand this panel by clicking on the “expand window” box. In some instances, this will show more information that has been hidden because of the size of the panel. • The (*Module3Lab1.R*) file is divided into sections. Each section corresponds to a step in this lab. By selecting an individual line or lines, you can click “**Run**…” and the command(s) will be executed in the R panel. 1. On the 1st line in Section 1, put your cursor on the line containing the word ls(). 2. Click **Run.** The ls() command will execute in the command window and show you the contents of your workspace.

Step	Action

5	<u>**Working with R:**</u>
	
	Load the .txt files you created in the first lab. Load the first file, **lab1_01.txt**
	
	1. Set the working directory to LAB01 where we have stored the data. On the console window type:
	`setwd("~/LAB01")`
	
	2. Select the line and press <ctl>**Enter**:
	
	`lab1 <- read.table(“lab1_01.txt”, sep=”|”, header=TRUE)`
	
	• If correct, R will simply return you to the command prompt (“> “).
	
	3. Now load the second .txt file, **lab1_02.txt**, by modifying the command (using the line of code in the RStudio command panel) you just entered.
	(Use the up/down, left/right arrow buttons to move from and within lines; change each occurrence of “lab1” to “lab2”.)
	The command should read:
	
	`lab2 <- read.table(“lab1_02.txt”, sep=”|”, header=TRUE)`
	
	4. When you have completed the edits, make sure that your cursor is within the line, press **Enter**.
	
	Note: R supports copy and paste, as well as up and down arrows for moving to previous commands, left and right arrows to move within/between lines and home/end to move to the beginning or end of a line.

6	<u>**Verify the Contents of the Tables:**</u> It is always a good idea to look at the data to make sure that everything works. You can use the **`head()`** command to print out the first 6 lines of a table or the, **`tail()`** command to print out the last 6 lines of the table. 1. Select and run the command: **`head(lab1, n=10)`** Record the value of the 10th line here: 1403, 30600, 4 2. Now do the same for the lab2 table, but use the **`tail(lab2, n=10)`** command instead. 3. Record the value of the 1st line here: 7, 564980, 147656, 18332, 14614, 2088, 61154

Step	Action

7	<u>**Manipulating Data Tables (data frames) in R:**</u>
	
	Examine the contents of the table in more detail.
	
	1.Execute the following command:
	
	`summary(lab1)`
	
	Ignore the values for the *hinc* and *rooms* columns for now. The *serialnoid* field represents a unique identifier (it's the household identifier) from the Postgres database. You no longer need it and it will interfere with some of the procedures you want to run against this data set, so create a copy of the lab1 table without that column.
	
	2. Select and run:
	
	`nlab1 <- lab1[,2:3]`
	
	This uses a feature of R that allows us to refer to rows and columns in a dataframe as if they were entries in a matrix. A blank entry in a row or column position means "use all available." This statement says: use all the rows in the table, but only use columns 2 and 3
	
	You could have used the following for the same effect (Note that the following code is not part of the script you can see in the source file *Module3lab1.R)*:
	
	`hinc <- lab1$hinc`
	`rooms <- lab1$rooms`
	`nlab1 <- data.frame(hinc, rooms)`
	
	You're taking advantage of R behavior that names the columns after the name of the variable. You could have used the following for the same effect:
	
	`nlab1 <- data.frame(lab1$hinc, lab1$rooms)`
	`names(nlab1) = c("hinc", "rooms")`

Step	Action

7
Cont.
	3. The dim(<table>) has the nice property of telling us how many rows exist in the table. Execute the following commands:
	
	`dim(nlab1)`
	`typeof(nlab1)`
	`class(nlab1)`
	
	Each of these commands tells us something about this particular object. You may not use these often, but they can be useful when R complains that it doesn't like something about the object that you just used.

8	**<u>Continue to Investigate Your Data:</u>**
	
	1. Select and execute the following commands:
	
	`summary(nlab1)`
	`cor(nlab1)`
	
	The summary function for data frames prints out summary statistics.
	
	2. Compare the median and the mean. What does it mean if the mean is less than the median? The average of a value is less than the middle value.
	3. How about the mean greater than the median? The average value is greater than the middle value.
	4. Does the min and max value for the quartiles make sense to you? Yes
	
	Here again you have a chance to do further cleaning of your data sets, but postpone this until you've finished the next few lessons.
	
	5. How do the values returned by the cor() function differ from the results obtained in lab 1? In Lab1 the correlation was 0.374485423827578. In Lab2 the correlation was 0.3820145

Step	Action

9	**<u>Save the Data Sets:</u>**
	
	1. Execute the following commands:
	
	`rm(lab1)`
	`lab1 <- nlab1`
	`save(lab1, lab2, file=”Labs.Rdata”)`
	`rm(lab1, lab2)`
	`ls()      # make sure they’re not in the workspace`

10

Examine Your Data:

1. Experiment with some of the examples used in the lecture portion of this lesson. Using the same selection techniques that you used earlier, run each line in the file.

    - Some commands don't print their results. If this is the case, type in the value of the variable you created in the command window. If the variable was named "x", you can type "x". You can also type "print(x)" which will do the same thing.

2. Experiment with R functions that identify the class and data type of a particular variable, type:

```
typeof(x), class(x), attributes(x), names(x),  dim(x)
```

3. Which ones work on which kind of data types?

typeof(x) – x can be any R object
class(x) – x can be an R object
attributes(x) – x is an object; returns a named list of attributes or NULL
names(x) – x an R object; returns a character vector of up to the same length as x or null.
dim(x) – x an R object; a matrix or data frame; returns NULL or a numeric vector
https://www.rdocumentation.org

4. Type these values into the RStudio command panel. Returns "double", "numeric", NULL, NULL, and NULL for typeof(x), class(x), attributes(x), names(x), and dim(x).
5. Typing all these commands for each variable is tedious. Alternatively, we will write a function `tellme` that takes a variable as an argument and performs `typeof, class, names and str` on that variable.
Select and run the lines beginning with "**`tellme <- function(x) {extending`** through the right curly brace.
6. Now execute the following command
`tellme`
You should see the definition of the function that you just entered! This is because R doesn't interpret a plain **tellme** as a function, but rather as an object to be printed out. The default print function for a function is to print its definition. You can try this with any other R function. Type **mean** and inspect the results.
7. Try tellme**()** with a series of variables.
8. Which commands actually list something?
Tellme(1) returns:
Function(x)
.Internal(typeof(x))
<bytecode: 0xcd45990>
<environment: namespace: base>
9. How might you get the other commands to list their return value?
[Hint: try `print()`]

Step	Action
11	<u>**Exit R:**</u> 1. Execute the following command: **`q()`** 2. R will ask you if you want to save your workspace. Answer “**`no`**”.

End of Lab Exercis