Scenario Selection Activity

Michael Surdek

Southern New Hampshire University

|  |  |
| --- | --- |
| **Step** | **Action** |
| 1 | **Invoke the R Environment:**  Log in to SNHU’s virtual desktop environment. If you have not yet done so, refer to the [instructions](https://learn.snhu.edu/d2l/lor/viewer/view.d2l?ou=6606&loIdentId=15847).  Once logged in to the SNHU VDI, you can access the RStudio environment by following these steps:   1. The RStudio is accessed through a desktop icon 2. Double-clicking this icon will invoke the R environment. 3. Verify that you see the following once you have invoked the R environment.   *\*The following images are examples and can vary depending on version.*  Screenshot of an R environment window |
| 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. |
| 3 | **Getting Familiar With R:**   1. Click each tab in each panel. What happens? 2. Type the following commands into the R command panel. The Help and Demo are both good commands to know in R so that you can get information you need on different commands and packages.   **help()  help.start()**  **demo() demo(graphics)**  Hit **ESC** to exit the demo. |
| 4 | **Read-in the Lab1 Script**  To complete this lab, you will need to download the Module4Lab1.R from the DAT 510 Brightspace Classroom into the virtual desktop environment. To do that, follow these steps:   1. Open a browser window inside the SNHU VDI. 2. Navigate to <http://learn.snhu.edu/> and enter your credentials as you normally would, then navigate to your DAT 510 course.   The file you will need to download, DAT 510 Data Files.zip, is located in the Assignment Guidelines and Rubrics folder located in the Start Here module. The zip file has all the files you will need for all the R assignments in this course.  Screenshot of the Assignment Guidelines and Rubrics page showing a link for "Graduate Discussion Rubric."Screenshot of the Assignment Guidelines and Rubrics page showing the Data Files link   1. To download the file DAT-510\_Data Files, left-click on it. You should see something like this:   Screenshot of a window showing the DAT 510 Data Files dot zip saving option   1. Select **Save File** and save the file (with the exact name) to the Virtual Lab image. When you click **Save as…**, it will default to “This PC/Downloads” which is a target area for downloading files. You are strongly encouraged to create a folder on the hard drive to save code and data files in so you can find them more readily throughout this course. The examples in the lab that follows have created a DAT-510 directory on the Desktop. 2. Now that you have downloaded the files you need, close the browser window. 3. Navigate to the folder that you downloaded DAT 510 Data Files.zip file to. Right-click on this file and select **Extract All**. This is where you will put the files in a convenient place for you to use. Follow the wizard, and extract the files to This PC, U:/DFS-MyDocuments (U:). You will now have a new folder called DAT 510 Data Files. In this folder, there will be a subfolder called MOD4. In this subfolder, you will have the files needed for the rest of this lab assignment. These files are:    * Module4Lab1.R    * lab1\_01.txt    * lab1\_02.txt    * Module\_Four\_Homework\_Template.doc 4. Navigate back to the RStudio environment. 5. Now, in the script window, open the script called “Module4Lab1.R”. (Click on **File** then **Open File**, navigate to directory from Step 6 where you just extracted the zip file, and click on file “Module4Lab1.R.” )   Screenshot of an RStudio environment script window. The File menu is selected and an arrow points to the Open file option in the drop-down menu.  Screenshot of RStudio Open File dialog box where the path "This PC to Documents to DAT 510 to MOD4" is highlighted in the address bar. Within the Open file option window, the file “Module4Lab1” is highlighted and has an arrow pointing toward it.   1. All the commands you will execute in this lab are contained in this script. To execute a command, do the following. (If this does not make sense when you read it, you may want to watch [this quick video](http://screencast.com/t/jmXdyxoAVgY9) that walks you through what it will look like in the R environment.)    * Position your cursor inside the line that represents the command you wish to execute.    * 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.   *NOTE: Be sure to resize both the R Console and the command pane so you can see both at the same time. If you do not do this, you will not be able to see if the commands you execute are working properly. An example screenshot is below.*     1. 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.   **Let’s Practice:**  *The (Module4Lab1.R) file is divided into sections. Each section corresponds to one of the remaining steps 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 first 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. |
| 5 | **Working With R:**  Load the first file, lab1\_01.txt. This file was included in the Module4\_Lab02.zip file you downloaded from Brightspace.  Navigate to the R Console and ensure your cursor is at the > prompt. This is where you will enter the lab1 and lab2 file load commands as noted in the steps below.   1. You will need to know the EXACT path where the file is stored. For example, your directory path format will look something like this: U:/DAT 510 Data Files/MOD4/ 2. Once you locate your directory, you will need to read in the lab1\_01.txt table. The command below will need to be modified with YOUR exact folder path. Below is an example showing how the format will look. (Notice the slashes are going in a different direction than those in Step 1. The reason is a “\” tells R that you have a command following.)     lab1 <- read.table(“U:/DAT 510 Data Files/MOD4 /lab1\_01.txt”, sep=”|”, header=TRUE)   * If correct, R will simply return you to the command prompt (“> “).  1. 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\_01” to “lab1\_02.”)  The command should read:  lab2 <- read.table(“U:/DAT 510 Data Files/MOD4 /lab1\_02.txt”, sep=”|”, header=TRUE)   1. When you have completed the edits, make sure that your cursor is within the line, and press **Enter**.   *NOTE: When copying and pasting from a Word document into the R environment, the format of “” changes. That said, it is recommend that you hand type commands with quotes in them so the compiler will recognize the appropriate letters and symbols.*  *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.*  **Explanation of Steps**   * Our call to the read.table function has used three arguments:  1. The location of the file — typically this would be a file on your file system (though, as here, it can also be a web address) 2. The sep argument says what character is used to separate items (in our case, the “|”) 3. The header argument says whether the first line in the file contains labels  * The result is a data frame. This is a rectangular object that can have different types of items in each column. * We have used the <- operator to make an assignment to the name “lab1.” We could have used the = operator also; there is no difference in this case (and most cases). You will not go wrong with object names if they start with a letter and include only letters, digits, dots (.), and underscores (\_). Names are case-sensitive: “LAB1” is different than “lab1.”   **In your homework template, respond to the prompt below for Step 5:**  **Write down the directory on your virtual desktop where your files are stored.**  "U:/DAT 510 Data Files/MOD4/" |
| 6 | **Verify the Contents of the Tables:**  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 six lines of a table or the **tail()** command to print out the last six lines of the table.  You will now run the lines of script from the Section 0 of the *Module4Lab1.R* file loaded in RStudio.   1. Select and run the command:   **head(lab1, n=10)**   1. Now, do the same using the lab2 table and use the tail command:   **tail(lab2, n=10)**  **In your homework template, respond to the prompts below for Step 6:**   * For the **HEAD(lab1,n=10)** command, record the value of the 10th line * For the **TAIL(lab2,n=10)** command, record the value of the 1st line   10 1791 271800 8  7 7 564980 147656 18332 14614 2088 61154 |
| 7 | **Manipulating Data Tables (Data Frames) in R:**  Let’s examine the contents of the table in more detail.   1. Execute the following command from *Module4Lab1.R* file loaded in RStudio:   **summary(lab1)**   1. Next, ignore the values for the **hinc** and **rooms** columns for now. The **serialnoid** field represents a unique identifier (it is the household identifier) from the Postgres database. You will no longer need it as it will interfere with some of the procedures you want to run against this data set, so you need to create a copy of the lab1 table without those columns. To do this, select the code in the *Module4Lab1.R* file loaded in RStudio and click **Run**:   **nlab1 <- lab1[,2:3]**  *This uses a feature of R that allows us to refer to rows and columns in a data frame 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.*  ***Extra Learning Opportunity***  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 *Module4lab1.R*.)  **hinc <- lab1$hinc rooms <- lab1$rooms nlab1 <- data.frame(hinc, rooms)**  In this code, you are taking advantage of R behavior that names the columns after the name of the variable.  Further, you could have additionally used the following for yet another way to achieve the same effect:  **nlab1 <- data.frame(lab1$hinc, lab1$rooms) names(nlab1) = c(“hinc”, “rooms”)**   1. The dim(<table>) has the nice property of telling how many rows exist in the table. Execute the following commands from *Module4Lab1.R* file loaded in RStudio:   **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 does not like something about the object that you just used.* |

|  |  |
| --- | --- |
| 8 | **Continue to Investigate Your Data:**   1. Select and execute the following commands from *Module4Lab1.R* file loaded in RStudio:   **summary(nlab1)**  **cor(nlab1)**  *The summary function for data frames prints out summary statistics.*  *Here again you have a chance to do further cleaning of your data sets, but postpone this until you have finished the next few lessons.*  **In your homework template, respond to the prompt below for Step 8:**   * Compare the median and the mean.   1. What does it mean if the mean is less than the median?   If the mean is less than the median, as in rooms, it can indicate that the spread of the data extends wider for values below the median than for those above the median. Outliers below the median tend to pull the mean in that direction.   * 1. How about if the mean is greater than the median?   If the mean is greater than the median, as in hinc, it usually means that the data spreads out further in values above the median than those below the median.   * Do the minimum and maximum values for the quartiles make sense to you?   1. Provide a brief description of your understanding.   The mininmum, maximum, and quartiles of the variables make sense to me because they are similarly spread around the centers, and for whichever direction has a larger difference from the centers, the relationship between the median and the mean matches what I would expect. |
| 9 | **Save the Data Sets:**   1. Execute the following commands from *Module4Lab1.R* file loaded in RStudio:   **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:**   * 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 Steps 1 through 4 found in the *Module4Lab1.R* file loaded in RStudio. Begin with the commands in Step 1: Scalars and Strings, and end with the commands in Step 4: Matrices, Tables, and Data Frames.   *NOTE: Some commands do not 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.*   * Experiment with R functions that identify the class and data type of a particular variable. In the R Console at the > prompt, type each of the commands below. After you type the command, hit **Enter** to see the result before typing the next command. Do this for all five commands:   **typeof(t)**  **class(t)**  **attributes(t)**  **names(t)**  **dim(t)**  **In your homework template, respond to the prompt below for Step 10:**   * Capture the results of each of the five commands above.   [1] "integer"  [1] "table"  $dim  [1] 3 4  $dimnames  $dimnames$ratings  [1] "Bad" "Good" "Worst"  $dimnames$reviewers  [1] "Ebert" "Martin" "Rowan" "Siskel"  $class  [1] "table"  NULL  [1] 3 4 |
| 11 | **Writing a Function:**  As you can tell, typing all the commands in Step 10 for each variable is tedious. R offers an alternate option to write a function.  To run all the commands using a single call, write a function called *tellme* that takes a variable as an argument and performs typeof, class, names and str on that variable.  To create the function, use the code provided in Step 5: Defining a Function in the *Module4Lab1.R* file loaded in RStudio.   * Select and run the lines in the block beginning with **“tellme <- function(x) {…** and ending with  **…}** * Next, execute the next line in the file. The command to run is:   **tellme**   * You should see the definition of the function you just entered. This is because R does not 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. For example, type **mean and Enter** and inspect the results. * Try tellme**()** with a series of variables. You can use variables like movies, critics, and ratings just to name a few.   **In your homework template, answer the question below for Step 11:**   * Which commands actually list something?   The first 2 commands typeof and class actually list results. in the case of movies it listed character for them all. The second 2 commands str and names do not list anything.   * How might you get the other commands to list their return value? *[Hint: try print()]*   To get the second 2 commands to list their return value, I tried to add print() in the function around the str() and names() where assigning p3 and p4. When I tried to execute tellme(movies) after making the change it still did not return any values. |
| 12 | **Exit R:**   1. Execute the following command:   **q()**   1. R will ask you if you want to save your workspace.    1. Answer “**no.**” |

*End of Lab Exercise*