

Handed out: 01/31/2014

Due by 17:25PM on Friday, 02/07/2014

Download and install the latest version of R and R Studio.

Problem 1. When you open R Studio, go to the text editor in the top left corner first. Type all commands in the text editor. Execute one or several commands by highlighting those commands, holding Ctrl key and hitting Return button. In that way you are preserving your work as a script. You are advised to read help pages for all R functions you will use.

- a) Ask the system for the current working directory. On your operating system, create a special directory for work with R, something like `C:\Code\R` or `/code/R`. Make that directory both readable and writable.
- b) Change your working directory to the new directory using appropriate R function. Copy the attached file `Smokers.txt` to your new directory.
- c) Create a vector `vv` with elements which repeat numbers 1, 2, 3 five times, using functions `rep()` and `c()`. What is the length of your vector. Do not just tell us, please ask R to tell you the length. Save that vector into a file in your working directory using function `save()`. Initially, name your file `vv.RData`. Try to open the file with an editor like `Vim` or `notepad`. Please do not change anything in the file.
- d) Save your vector `vv` into another file, called `vv.txt`. This time turn the `ascii` parameter of `save()` function to `TRUE`. Try reading your file. Could you do it this time?
- e) Use R function `list.files()` to list the content of your working directory.
- f) Point function `list.files()` to any other directory and assign its result to a variable. What is the class of that variable? (Hint, use function `class()`). What is the structure of variable `vv`. (Hint, use function `str()`). What is the mode of variable `vv`. (Hint, use function `mode()`).
- g) Remove variable `vv` from your workspace. Next, load that variable from file `vv.RData`. Verify that you got whatever you stored.
- h) Remove variable `vv` from your workspace, again. This time, load that variable from file `vv.txt`. Again, verify that you have recovered your variable.
- i) Save the content of your text editor as a file with extension R. You can rerun all commands in that file as a script, if you want, one day. You might want to edit the file first, but that is another matter. Provide that file as a part of your submission.
- j) Close R Studio. Do not forget to save your workspace. Restart R Studio. Convince yourself that all of your variables are there.

Problem 2. Create a vector `num` of numbers with 8 randomly ordered values between 1 and 20. Use `c()` function.

- a) Turn that vector into another vector of characters `cnum` using function `as.character()`.

- b) Calculate minimum, maximum and mean value of two vectors.
- c) Transform vector `cnum` back into numbers using function `as.numeric()`. Verify that you got numbers back.
- d) Create another vector `snum` by selecting only those numbers from vector `num` which are greater than 10.
- e) Create a logical vector `lnum` which will tell you whether an entry in vector `cnum` is greater than 10.
- f) Set 3rd element of vector `cnum` to NA. Recreate logical vector `lnum` and report its elements. Recreate vector `snum` by selecting those elements of vector `cnum` which are greater than 10. What was the effect of NA.
- g) Create a character vector `cdate` with three dates between January 01, 2014 and March 01, 2014. As the format for character representation of dates use "January-01-2014". Transform that vector into vector `ddate` of Dates class using function `as.Date()` and format "%B-%d-%Y". Please read help for `as.Date()`. What are the class and mode of vector `ddate`. Display values of vector `ddate`. Sort values in vector `ddate`. Provide the information on the order that sorting imposed, in other words ask R to tell once sorted which elements came first, which second, and so on.

Problem 3. Create matrix A with 3 rows and 4 columns. Start with small integer values between -3 and 3 for elements of matrix A.

- a) Create new matrix B obtained by subtracting 1 from all elements of matrix A.
- b) Create new matrix C obtained by multiplying by 2 all elements of matrix A.
- c) Create matrix TA which is a transpose of A.
- d) Calculate matrix ATA which is a matrix product of matrix A multiplied by matrix TA from the right. Ask R to tell you dimensions of matrix ATA.
- e) Calculate matrix TAA which is a matrix product of matrix TA multiplied by matrix A from the right. Ask R to tell you dimensions of matrix TAA.

Problem 4. Create a square, almost symmetric 4 X 4 matrix B with values 1, 2, 3 and 4 on the diagonal. Let values off diagonal be between 0.01 and 0.2. Almost symmetric matrix is not a scientific term. It just means that most of the off diagonal terms on transpose positions are close in value.

- Determine matrix BINV which is an inverse of matrix B.
- Demonstrate that the matrix multiplied by its inverse produces a unit matrix. Unit matrix has all elements on the diagonal equal to 1 and all other equal to 0.
- Find eigen values of Matrix B and matrix BINV.
- Find eigen values of matrix `t(B)` where `t()` is transpose function.

Problem 5. Load data in attached file `Smokers.txt` into variable `smokers` using function `read.delim()`. Use parameters `header=TRUE`, instructing R to read column header names from the first row of the file, and `sep="\t"`, telling R that file is tab delimited. As a side note, parameter `header=TRUE` works only if the first row of the data file has one element fewer than the rest of the file.

- a) What are the `class`, `mode` and `str(ucture)` of variable `smokers`.

- b) What are the dimensions of variable `smokers`.
- c) What are the labels (names of horizontal rows) of variable `smokers`.
- d) List values in individual columns of variable `smokers`.
- e) You have noticed that R interpreted column `GDPPerCapita` as a factor, i.e. a low cardinality column that could be represented by a small number of levels. R does that because we made a mistake and forgot to include parameter `stringAsFactors=FALSE` when reading the content of the file. We want that column to be a set of integers just like column `GDPRank`. To achieve this let us use function `gsub()` (see note below), which can replace a comma used as a thousand separator with null space (that is, remove comma). We should also use function `as.numeric` to transform character representation of numbers in column `GDPPerCapita` with numeric values. You can remove unused factors from column `GDPPerCapita` by using function `droplevels()`. Verify that offending factor is gone by redisplaying the structure of variable `smokers`.
- f) Display columns `PercentSmokers` and `GDPPerCapita` using bracket notation or `[, index]` selection.
- g) Use function `plot` to try to understand whether there is any correlation between variables `PercentSmokers` (that is the number of people of a country who smoke) and its Gross Domestic product Per Capita. We are just practicing syntax. There may not be any correlation. Please label horizontal and vertical axis and place a title on top of the graph. Represent points on the graph with circles.
- h) Create a histogram displaying the number of countries in `GDPPerCapita` brackets 0 to 2,000, 2,000 to 3,000, 3,000 to 5,000, 5,000 to 10,000 and 10,000 to 50,000. Label your histogram. Paint the title in purple color. Please be free to display the histogram with more meaningful brackets.
- i) Present the same histogram as a pie chart.

Note: Offending factor appears in imported data frame because R did not understand the separator used for thousands, so it did not treat column `GDPPerCapita` as numbers but rather as strings (characters). We could have told `read.delim` function not to transform characters into factors, what R does by default. One does that by specifying parameter `stringAsFactor=FALSE`. We forgot to do that and R converted GDP data into a factor. We want that column to be plain numeric. So, we can get rid of offending commas ("," i.e. acting separators for thousands), using function `gsub()` (global substitute). The first parameter of that function is the pattern we want to remove and the second is its replacement. The third parameter is the variable where replacement should be performed. You can run `droplevels()` function on column `smokers$GDPPerCapita`. However, you first want to transform the character values into numbers. You do that using a statement like the following.

```
smokers$GDPPerCapita <- as.numeric(gsub(",", "", smokers$GDPPerCapita))
```

SUBMISSION INSTRUCTIONS:

Your main submission should be an MS Word document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you just copy your code and results from the R console and paste them into the Word document. Start with this text of homework assignment as the template. Please add any other files that you might have used or generated.

- Package all materials you are submitting into an archive called E63_LastNameFirstNameHW01.zip.
- Naming your file properly is important. We download many files and if they are all named Assignment01.zip it becomes hard not to overwrite or lose them.
- Please do not use archiving tools which do not produce ZIP files. Please do not submit *.rar or *7zip files.
- If you are using a Mac, please make sure that your files are READABLE to users of Windows.
- You are welcome to save and submit your work as a PDF file, however please, always submit a Word document, as well.
- Upload your ZIP archive to the course web site. Every assignment has its drop box.
- If you miss the deadline, please submit your solution into the 00_AnyHW_WayLate Drop Box. Late assignments will be graded as well. We will chop 5% of your grade for every day you are late.
- Your grade for every assignment will be entered as a comment next to your submission.

If you have issues with the formulation of the assignment or the software you are using, please FIRST go to the Discussion Forum on the class web site:

<http://isites.harvard.edu/icb/icb.do?keyword=k102025> and check whether someone else raised the same issue and whether the answer is already there. If not, raise the issues yourself. A person from the class or a member of the teaching staff will respond.

If the issue is not addressed for a while, please send an inquiry to cscie63@fas.harvard.edu. The discussion forum is a very important tool. We all learn from the discussions on the forum.

If we respond to your inquiry to class email address or any email address of the teaching staff, PLEASE DO NOT RESPOND WITH A THANK YOU NOTE. This is not a joke. We will take 2% of your grade for that week's assignment for every "thank you note".

We will apply the same penalty to any trivial email. Please do not complain when you lose a few points on your assignment.

If you have issues with the class web site, please let us know right away. In the past, we experienced issues with the visibility of various folders, upload permissions and so on.

We will try to resolve such issues as soon as we hear about them. For some issues we depend on the university support services and delays are possible.