R Bootcamp
Homework 4: More Merging, Indexing, and Function Practice

Marguerite A. Butler

February 11, 2018

DUE: Friday, February 16rd by 5pm. Please try to submit via GitHub. If you must email, use subject "710 Homework 4"

Instructions

Do the exercises and **turn in your homework by creating a script of R commands** (`HW4_myinitials.R`). Make one script for questions 1 and 2, and edit `SpecFunctionScript_stub.R` for question 3, replacing stub with your initials. Your script should generate all necessary files, so no need to attach any other files. As before, please make sure to print to screen when requested, to include comments in the code so that you remember what the code does, and **please answer all questions indicated in bold in the comments (please indicate question number)**.

1 Using logical comparisons to extract objects

It's probably a good idea to do a little more practice with indexing and manipulating data.

Create the matrix using the code below and do the following exercises. **In each case print the result to console in your script using the `print()` function.** Because we are generating the matrix with a random sample, each time you run the code it will look a little different. For all of the exercises, write generic code that will work for any x matrix that is generated.

```
> x <- matrix(sample(-50:50, 25), nrow=5)
> rownames(x) <- LETTERS[1:5]
> colnames(x) <- LETTERS[6:10]
> x
```

```
   F   G   H   I   J
A -17 -30  20  40 -31
B -11 -45 -37 -20  -4
C -14 -49  27  10 -40
D -34  47 -50 -42  41
E -46   9 -25  19  12
```

a. Print to console the matrix formed by extracting rows A, E, and columns G and I.

```
   G  I
A -30 40
E   9 19
```

b. Extract each of the elements `x[1,3]`, `x[2,2]`, and `x[3,1]`, and replace these entries in the matrix `x` by zeros. There should be a diagonal line of 0's in the matrix now, and it should look like this:

```
   F   G   H   I   J
A -17 -30   0  40 -31
B -11   0 -37 -20  -4
C   0 -49  27  10 -40
D -34  47 -50 -42  41
E -46   9 -25  19  12
```

c. Extract all negative elements of `x`, replace these entries of `x` with missing values `NA`.

```
   F  G  H  I  J
A NA NA  0 40 NA
B NA  0 NA NA NA
C  0 NA 27 10 NA
D NA 47 NA NA 41
E NA  9 NA 19 12
```

d. Extract all elements of `x` that are not `NA`, replace these with one hundreds.

```
   F   G   H   I   J
A  NA  NA 100 100  NA
B  NA 100  NA  NA  NA
C 100  NA 100 100  NA
```

```
D  NA 100  NA  NA 100
E  NA 100  NA 100 100
```

e. Extract all elements of `x` that are `NA`, replace these with zeros.

```
    F   G   H   I   J
A   0   0 100 100   0
B   0 100   0   0   0
C 100   0 100 100   0
D   0 100   0   0 100
E   0 100   0 100 100
```

2 Exploring Data using R skills

Living in Hawaii, you are curious as to whether there is a link between climate and crime across the USA, and decide to look into with R. You notice that the built-in dataset `USArrests` has data on crime, and `state.x77` has data on climate. Look up the help pages for these to read the data descriptions and explore the data in the question below. **Answer: Write 1-2 sentences to describe the dataset, especially Frost and the rates reported in USArrests. Make sure to include relevant sample population and units.**

a. Merge the two data sets by `row.names` and save it as `dat`. *hint: Look up the help page for* ***merge*** *if you forgot how to do this, make sure to read the Details section.* **Answer: What happened to the `rownames`? Where is it stored in `dat`?**

b. **Housekeeping (practice indexing by name and replacing values)**. Look at the names of `dat` and replace any problematic ones (e.g., uninformative or contain spaces). `"Row.names"` is not informative, so change it to `"state"`.

Hint: save ***names(dat)*** *to a temporary object* ***nn****. Test for the element of* ***nn*** *that matches the pesky name, and use that index vector to select the member of* ***nn*** *that you want to replace and replace it with a nicer name. Then copy the nice names back to* ***names(dat)****.* This way if you mess up you can easily start over with no harm done. If you can?t figure it out then look at the hint at the bottom of this question.

c. Now you can begin looking at your data. Subset the dataframe using a logical condition to show us which rows have frost values less than 20. **Answer: Which states are warmest? Which states are added if we increase the frost limit to 30?** Print both to console.

d. **Answer: Do warmer climates seem to have higher crime rates in terms of murder, assault, and rape? What about income, illiteracy, and high**

school graduation rates? Plot murder, assault, and rape crime statistic as a function of `Frost` (make sure `Frost` is the x-variable). Make a six panel figure by setting the graphical parameter `par(mfrow=c(3,2))` before you start your six plots. (Hint at bottom if you need it).

e. You notice a funny pattern with the frost vs. rape graph. There are a few outliers in the upper right corner (high for both rape and frost). Which states are they and how can you isolate those values? Frost greater than what value and Rape greater than what value? Make a compound logical condition and subset the dataframe to get only those rows. Use `print()` to print to console. Make a plot that highlights those points with large red points. Where does Hawaii fall? Identify the point for Hawaii with a green dot. **Write comments on each of these lines to explain what they do and what code changes were required.**

f. Test for a significant relationship between rape and frost using the `lm()` function and code below. Save the linear model output as `rp.lm`. Run `anova()` and `summary()` on the linear model object using the code below, and print the anova and summary results to screen in your script. **Answer: Is there a significant relationship between rape and frost? (look for a $P-value < 0.05$ in the anova table for the Frost effect). Is it a positive or a negative relationship? Look at the sign of the slope between Frost and Rape in the summary output under "Estimate" (this is the slope for the continuous effect). Does the crime rate go up or down as it gets warmer?**

```
> rp.lm <- lm(Rape ~ Frost, data=dat)
> anova(rp.lm)
> summary(rp.lm)
```

g. Look up the help page for `lm`. What type of object is returned from this function (look under the VALUE heading)? What kinds of components are included in this object (you don't have to list them all, just give us an idea). It mentions some generic accessor functions. Try a couple of them on the `lm` object and print it to screen. What are they (just give us the names)? The slope of the regression line is the second coefficient. How would you grab that value? Grab it and print it to screen.

h. To your dismay, there seems to be a slightly significant relationship between rape and frost, and it's going in the wrong way for the warmer climates! To your even greater dismay, you suspect that if we exclude the three cold-weather outliers, it might be an even stronger relationship. Just because we can, let's see what happens when we replace these values with values more inline with the trend. (We are just exploring our curiosity here and learning the "what ifs" regarding our data, you would never try to publish data muckery like this.) Using the subsetting condition above, replace the frost values for these rows with "15" which is about where it should be to fit the trend.

i. Replot the data and redo the lm, anova, and summary on the lm. Print them all to screen. What happens? You could also exclude the outliers and see what happens. It will reduce the sample size, but given that there are 50 data points it would probably not reduce the power too much. In any case, excluding outliers is a very controversial topic, so we wouldn't do it. But we did learn that the results are very strongly influenced by the outliers. It may be a good reason to think about what, if anything, might be different about these three states.

j. Code hints if you get stuck:

```
> nn <- names(dat)
> nn[ nn == "Row.names"] <- "state"
```

First plot of six:

```
> par(mfrow=c(3,2))
> with( dat, plot(Frost, Murder.x))
```

3 Function Practice

Complete the script that we have been working on in class to automate what we?ve done so far and process all of the files. Design it so that it works from your `Rclass` directory and your input files are in `Rclass\Data`. Recall in class we: (1) read in a spectroradiometer file, (2) subset it to a relevant wavelength range, (3) plot intensity as a function of wavelength, (4) find the max intensity and wavelength at which it occurs, and (5) output a dataframe of the final max intensities and wavelengths. Start from `SpecFunctionScript_stub.R` and modify. Use comments to document what your code is doing. Steps are provided to guide you to the final result, but they do not require answers. You will turn in the final, commented script with your initials that performs all of these tasks and produces the output as specified below.

a. Starting from what we did in class, the first step is to make sure that the two functions `read.spec()` and `plot.spec()` work as you expect. If not, then debug it to get it to work.

b. `plot.spec()` should take in a dataframe for one file, create a lineplot of intensity as a function of wavelength, find the point on the plot where maximum intensity occurs, and plot a red point there. It should also return the coordinates where maximum intensity occurs (intensity and wavelength).

c. Use the `list.files()` function to get all the files in your `Data` directory and save as object `myfiles`. What is the class of `myfiles`? Use grep() to get all the files that contain "irr.txt" and no other files (you should have other files in your `Data`

directory. This also is a way to "error-proof" your code to make sure you have valid input.

d. Because your files are in the `Data` directory, add the path to each element of `myfiles` using the `paste()` function. Don't forget to include the `\`. Consult help `?paste` if needed. We will be running this script from your `Rclass` folder with the `Data` directory within it.

e. Figure out how to use "myfiles" in your script to read in data without typing in filenames!

f. Write a wrapper function that will read in a file, create the plot, and return the max intensity and lambda for that file. It will be a very short function. Execute once for each file, and save the output as a dataframe with one row per file. Instead of plotting to console, generate pdfs. To do this, you can just call `pdf()` it outside of the call to the wrapper function, and close it afterwards. It will create a pdf with multiple pages, one for each time a plot is created.

g. Write the final dataframe to an output file. It should have columns `lambda` and `max_intensity`, and one row per input file.

h. Clean up your code and delete any unnecessary code and comments. Organize your script to put all function definitions at the top of the script, with all the code that does the actual work below it. Use comments to label these sections. Make sure you have good comments for each major step, and that the code is readable. Test it by sourcing it from the R console before turning in.

i. Congratulations! You wrote a beautiful script!