R Class Homework 5: Working with your own data

The assignment

I'd like everyone to start making the leap from pre-fabricated exercises and never-fail regurgitation of code, to actually doing useful work with your own data. This is a self-directed exercise, flexible to your own needs and interests, but within the following constraints:

- I would like a neatly formatted and commented R script that:
  - 1) Calls your data into R.
  - 2) Does some modifications that will be useful to easier coding or analysis.
  - 3) Asks some questions about your data that are interesting or useful to you.
- Below are some suggestions. You do not need to be confined to them. But:
  - DO follow the instructions under "Preparing your data for R".
  - DO see the section at bottom "Sharing your dataset with me".
- If you already have scripted analyses, please make a clean new script for this assignment with exceptionally nice formatting and commenting. If you want an exception to that, email me.

Suggestions

<u>How to do things not covered in the tutorials</u>

Use Google. Word your search queries carefully. Find tutorials, or answers on the site "stackoverflow".

<u>Preparing your data for R</u>

- Do only a MINIMUM of modification of your data outside of R.
  - Make sure there is only 1 header row.
  - Make a single, contiguous dataset. (Excel files often have multiple chunks of datasets and summary tables and graphs on the same sheet.)
  - From Excel: grab your rawest data (least modified, most original), put it into a clean sheet with one header row, go to "Save as…", and choose "Comma separated values (.csv)" as the file type. Call that .csv file into R.
- For datasets *not* in Excel (e.g., '.txt', '.dat'), you might just pull them directly into R. You might skip lines of meta-data at the top of the file with the 'skip' argument in read.table(). Make sure to set the right separator ' sep= ' argument.
- You should feel that any time you directly manipulate your data in something like Excel, you are (in a probabilistic manner) irreparably destroying your dataset. So save original files, make minor modifications, and get them into R where your data is safe!

<u>Common problems with pulling data into R</u>

- Your computer probably automatically hides extensions for known file types. If you paste your file name into read.csv(), but it doesn't have the ".csv" at the end, R won't be able to find it. I recommend looking up how to make your operating system always show file extensions.
- If you make a directory object like dat.dir <- "my directory path/" make sure it ends with "/"! Also make sure all "\" are converted to "/" (for Windows users).

Useful dataset modifications in R

- **Column names**: Make your column names sensible, but also short, and easy to type. I always make my column names lowercase, and make all separators ".", that way I never have to hit 'shift' while typing, and I never have to remember whether I used capital letters or a different separator.
- **Classes**: Check the class() of your different columns. You can use str() on the whole dataset. Are they the class() you expected? Change them with one of the as.*() functions if you need.
- **NAs**: R will recognize numerical columns, and turn blanks into NAs for you. If you have a different value of missing data, such as "-", R will probably turn the column into class character. So you'll have to index all of the "-" cells, and make them equal to NA (no quotes, NA is a particular type of element in R), then convert the column to numeric with as.numeric().
- **Make new columns**: These could be useful sums, means, log transformed data, character columns pasted together.
- **Edit data in the columns**: Correcting spellings, changing case (lower, upper, proper).
- **Reorganize your columns**: You can do this by indexing, e.g., df <- df [c(3,2,1)].
- **Merging multiple datasets into one**: Look up the merge() function.
- **Times and dates**: Convert your character string times and dates into R formatted (POSIX) times and dates. Google this. You'll want to use as.POSIXct() and strptime(). For dates only, as.Date() is simpler.

Useful questions about your data in R

- **Basic information:** Dimensions of the dataset, number of unique entries in some column (plots, species, …), list of unique entries, look at the head and tail of the dataset.
- **Summary statistics:** There are shortcuts for getting basic summary stats. Google them.
- **Summary tables:** You could make a new data frame and populate it with summary stats. For example, it could have all of your unique plots, or unique species, and then columns of summary data about those plots or species (mean values, total count of data, whatever is useful).
- **Statistical analyses:** You know some basics like aov(), TukeyHSD(), and lm(). Go ahead!
- **Plotting:** Use examples from previous homework assignments. Histograms and data-count bar charts are useful for getting to know the structure of your data. Google any other kind of graph you might want to make. Try making some nice titles and axis labels.

Sharing your dataset with me

- **DO NOT share your original, functional dataset with me.**
- I have included a custom function with this assignment that will take your dataset into R, randomly reorganize the data in each column, and return it to your directory as a *different* file with the word "SCRAMBLED" in the title. This *will not* alter your original data file in any way. But as a good practice, you should have your original data locked away somewhere else, and have copies of '.csv' files that you are using for R analyses.
- You can share the new, altered data file(s) with me. This will allow me to run your code, even though all graphs and analyses will come out different due to the reorganized data.
- I'll provide simple instructions for how to use the function, once I've made it.
- **This is not a requirement. If you don't want to share your data in any way, shape or form, that is fine. It will just limit the amount of useful feedback I can provide.**