# Homework 4

For this homework you will create an R Markdown file and output (HTML file) and upload both to wolfware. Be sure to include text explaining your thought process/what you are doing with your questions.

The purpose of this homework is to get practice reading in raw data from different sources. Some files are available in the homework link, others via a URL, and others may be connected to using a package of some type.

## Part 1 - Some concept questions

1. If your working directory is `myfolder/homework/`, what path would you specify to get the file located at `myfolder/MyData.csv`?
2. What are the major benefits of using R projects? Should you be using an R project for each homework assignment (or at least for the course)?? (The last question is rhetorical!)
3. What is git and what is github?

## Part 2 - Reading in Delimited data

The data sets we'll use for this part comes from the UCI machine learning repository.

### Glass data

The first data set is called glass.data. You'll need to open the raw data set to determine the type of delimiter. The data is available at: https://www4.stat.ncsu.edu/~online/datasets/glass.data.

The description of the data (not super useful!):

"Vina conducted a comparison test of her rule-based system, BEAGLE, the nearest-neighbor algorithm, and discriminant analysis. BEAGLE isa product available through VRS Consulting, Inc.; 4676 Admiralty Way, Suite 206; Marina Del Ray, CA 90292 (213) 827-7890 and FAX: -3189. In determining whether the glass was a type of 'float' glass or not, the following results were obtained (# incorrect answers): Type of Sample Beagle NN DA Windows that were float processed (87) 10 12 21 Windows that were not: (76) 19 16 22 The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence. . . if it is correctly identified!"

The variables and their descriptions:

| Variable | Description |
|---|---|
| Id | Number 1-214 |
| RI | Refractive index |
| Na | Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10) |
| Mg | Magnesium |
| Al | Aluminum |
| Si | Silicon |
| K | Potassium |
| Ca | Calcium |
| Ba | Barium |
| Fe | Iron |

With the last variable being `Type of Glass` with values of – 1 building_windows_float_processed, – 2 building_windows_non_float_processed, – 3 vehicle_windows_float_processed, – 4 vehicle_windows_non_float_processed (none in this database), – 5 containers, – 6 tableware, – 7 headlamps.

1. Read this data into R using functions from the tidyverse. Notice that the data doesn't include column names - add those (in a manner of your choosing). Print out the tibble (just call the object name).
2. Overwrite the `Type_of_glass` variable by creating a `factor` there instead. (See `help(factor)`.) Use the variable descriptions above to give meaningful factor levels.
3. Print the data frame with only observations where the `Fe` variable is less than 0.2 and the Type of Glass is either tableware or headlamp.

**Yeast data**

The second data set is called yeast.data. You'll need to open the raw data set to determine the type of delimiter. The data is available at: https://www4.stat.ncsu.edu/~online/datasets/yeast.data.

The description of the data (not super useful!):

"The references below describe a predecessor to this dataset and its development. They also give results (not cross-validated) for classification by a rule-based expert system with that version of the dataset. Reference: 'Expert Sytem for Predicting Protein Localization Sites in Gram-Negative Bacteria', Kenta Nakai & Minoru Kanehisa, PROTEINS: Structure, Function, and Genetics 11:95-110, 1991."

The variables and their descriptions:

| Variable | Description |
|---|---|
| seq_name | Accession number for the SWISS-PROT database |
| mcg | McGeoch's method for signal sequence recognition. |
| gvh | von Heijne's method for signal sequence recognition. |
| alm | Score of the ALOM membrane spanning region prediction program. |
| mit | Score of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial andnon-mitochondrial proteins. |
| erl | Presence of 'HDEL' substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute. |
| pox | Peroxisomal targeting signal in the C-terminus. |
| vac | Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins. |
| nuc | Score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins. |
| class | Localization site |

1. Read this data into R using functions from the tidyverse. Notice that the data doesn't include column names - add those (in a manner of your choosing). Print out the tibble (just call the object name).
2. Select only the `class` and `mcg` columns. Report the mean and standard deviation of the `mcg` value for each setting of the `class` variable (recall the use of `summarize()` instead of `mutate()` to do this type of thing easily!).

## Part 3 - Database

We will be working with an example SQLite database called `chinook`. It has tables of songs, playlists, customers, invoices, and more. See the 'chinook-database-diagram.pdf' to get a better understanding of how these tables are related to one another.

1. Download the `chinook.db` database. (If needed install and) load the `DBI` and `RSQLite` packages, and load the `tidyverse` package. Use `dbConnect()` to connect to the this local database.

You'll only need two arguments to `dbConnect()`, the type of database and then the path to the database you've downloaded.

2. Now print out the tables in the database using `dbListTables()`.
3. Use `dbGetQuery()` or `tbl()` to grab and print out the `invoices` table and the `customers` table.
4. Use an `inner_join()` to combine the two tables above by the `CustomerID` variable.

## Part 4 - Querying an API

For this section, you'll connect to the news API we connected to in the notes. You'll need to go to newsapi.org to register for a key.

1. Use `GET` from the `httr` package to return information about a topic that you are interested in that has been in the news lately (store the result as an R object). Note: We can only look 30 days into the past with a free account.
2. As done in the notes find your way to the data frame that has the actual article information in it. This time, save that as an R object. Select only the source, author, and title columns and print the tibble out.