

Worksheet 4: Input/Output

Lab worksheet: Input/Output (I/O)

File input and output

Before doing any file input/output, we need to know our working directory. The working directory is the file system location (or folder) where R knows to look for files. To find out the current working directory, use the function `getwd()`, for instance

```
> getwd()
[1] "/Users/rrohlfis"
```

The result there is the path to the working directory. In the path, each slash (/) indicates a folder. That's like clicking into a folder in the finder. In this case the working directory is "rrohlfis," which is inside the folder "Users."

In order to have easy access to data files, you may want to change your working directory using the `setwd()` function.

1. Take the following steps to prepare to read in the file `hundredFibs.dat`
 - a. Download `hundredFibs.dat` from iLearn and save it to your homework folder (wherever that may be)
 - b. Identify the path to your working directory. (one way to do this is to right click (control click) on the folder, then press option to get "Copy 'FOLDERNAME' as pathway", click that, and then paste out the path)
 - c. Find out your current working directory in R using `getwd()`.
 - d. Use `setwd()` to set your working directory to be your homework folder.

Now then, especially when dealing with large datasets (think DNA databases with millions of entries), it's more practical to load or read in data, rather than inputting it manually. There are a number of functions for this kind of file input, the simplest being `scan()`.

2. Look up the manual page for `scan()`
 - a. Use `scan()` to read in the data from `hundredFibs.dat`, saving it to `fibSeq100`.
 - b. Confirm that `fibSeq100` contains the first 100 Fibonacci numbers.
3. Research a different function that can read in data from a file and try it out.
 - a. What function did you use?
 - b. How did it differ from `scan()`?
 - c. Trade experiences with a lab colleague.

Similarly, data can be written to file using a number of functions, the simplest being `write()`.

4. Look up the manual page for `write()`
 - a. Use `write()` to write the first 50 Fibonacci numbers to a new file named `fiftyFibs.dat`.
 - b. Check out the file you wrote to using a simple text editor (like notepad on windows or `TextEdit` on mac). What do you see?
 - c. What do you notice about the format/appearance of this file?

5. Research a different function that can write output to file and try it out.
 - a. What function did you use?
 - b. How did it differ from `write()`?
 - c. Trade experiences with a lab colleague.
6. Look up the function `save.image()`.
 - a. Try it out with your current session.
 - b. Confirm that you've created an `.Rdata` file in your working directory.
7. Look up the function `load()`.
 - a. Clear out your environment so you no longer have any actively defined variables.
 - b. Confirm that your environment is empty by typing `fibSeq100` into the console.
 - c. Use the `load()` function to retrieve your variables (including `fibSeq100`)