Learn, Visualize, & Analyze

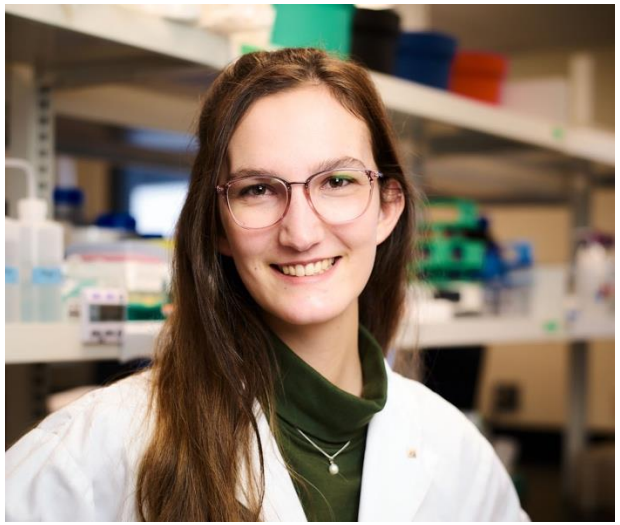
LAFOREST LAB R WORKSHOP

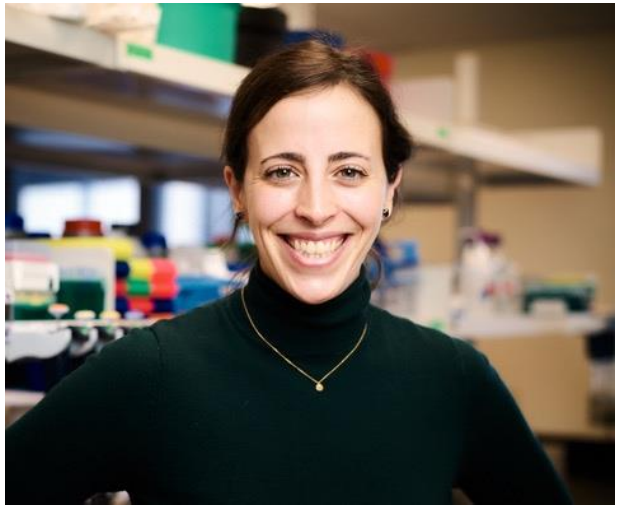
May 12-14th, 2025

WELCOME
BIENVENUE
BIENVENIDOS

THE TEACHERS

- Jonathan Rondeau-Leclaire (étudiant MSc)
- Sophie Boutin (étudiante PhD)
- Amy Heim (postdoc)
- Isabelle Laforest-Lapointe (prof)





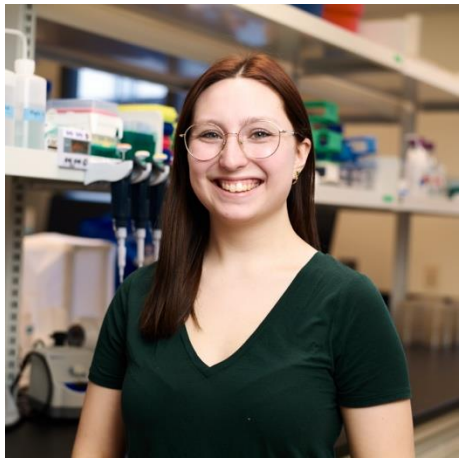
THE SUPPORT TEAM

Ema Lussier
PDR

Maria Lucia Gonzalez Torres
MSc

Jacob Beauregard
MSc

Morgane Enea
PhD



Gaële Lajeunesse
PhD

THE LEARNING CREW

THE “SCHEDULE”

NOTE

This schedule is a draft and might evolve depending on how things go.

Day 1 (D7-3017) AM Session 1

08:30	Launching the workshop
09:00	An intro to the RStudio environment
09:50	Challenge 1
10:10	20min Break
10:30	The basics: functions, data exploration, and summaries
11:20	Challenge 2
11:40	Recap of session 1

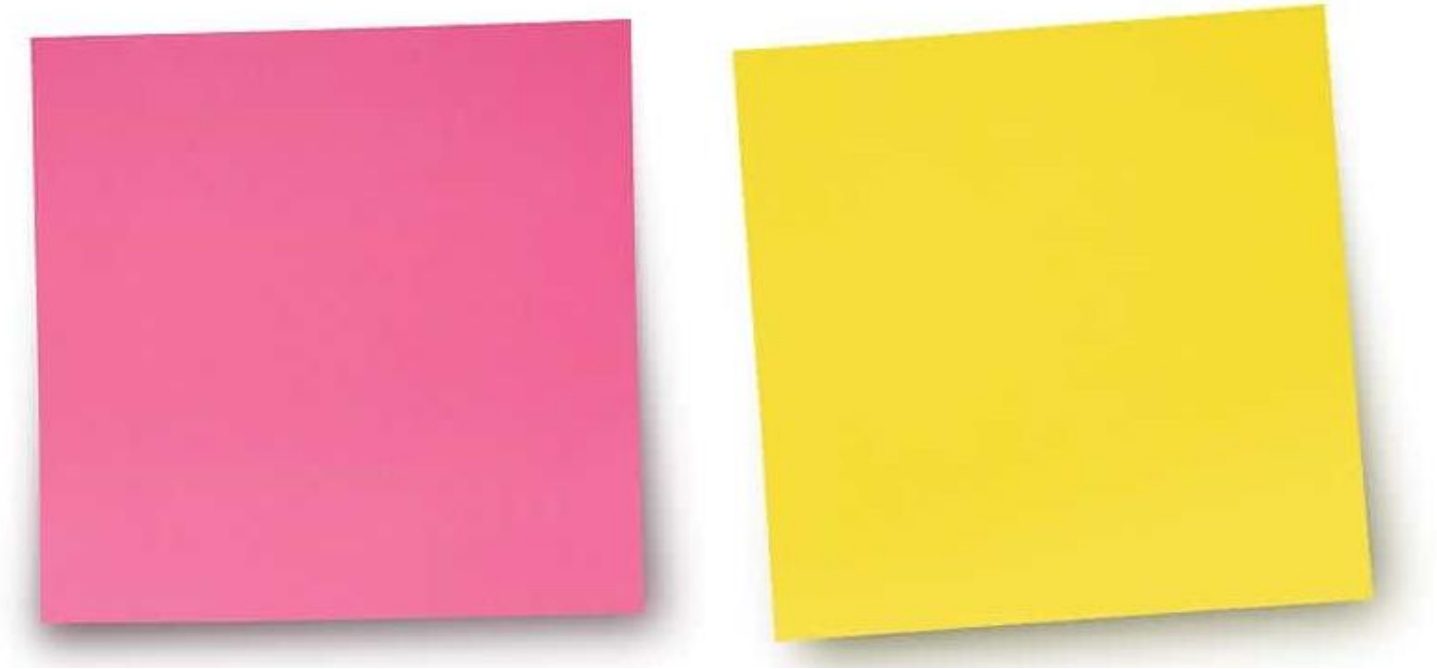
Day 1 (D7-3017) PM Session 2

13:00	Subsets and plots
13:50	Challenge 3
14:10	20min Break
14:30	Do's and don'ts of data entry
15:20	Challenge 4
15:40	Recap of session 2
16:00	Hands-on with you own data or practice with ours

CODE OF CONDUCT

- **Respect & Inclusivity**
  Treat all participants with kindness and respect, regardless of background or skill level.
- **Collaborative Learning**
  Encourage questions and discussions; avoid interrupting others.
- **Professionalism**
  Use appropriate language and avoid disruptive behavior.
- **Zero Tolerance for Harassment**
  Discrimination, harassment, or offensive remarks will not be tolerated.
- **Ask for Help**
  Reach out to organizers if you experience or witness any issues.

THE POST-ITS



RESOURCES

Website

Schedule & code of conduct
https://isabelle-laforest.github.io/RworkshopUdS_2025/

Github

Scripts and data.
https://github.com/isabelle-laforest/RworkshopUdS_2025/

Teams Group

Data and active notes by Gaële.

R Cheatsheet

Functions and examples.

DAY1: LEARNING GOALS

1. **Build Confidence with R/RStudio**
   - ✓ Overcome initial hesitation and start enjoying working in R!
2. **Navigate RStudio**
   - ✓ Understand the four panes (Script, Console, Environment, Files/Plots) and their functions.
3. **Work with Objects**
   - ✓ Create, inspect, and delete objects (e.g., vectors, variables).
4. **Perform Basic Operations**
   - ✓ Execute arithmetic calculations and use simple built-in functions (e.g., sum(), mean()).
5. **Write and Organize Code**
   - ✓ Create, annotate (with # comments), and save an R script for reproducibility.
6. **Leverage Packages**
   - ✓ Install and load packages to extend R's functionality.
7. **Avoid Common Pitfalls**
   - ✓ Learn key *do's and don'ts* (e.g., naming conventions, avoiding attach()).
8. **Visualize Data**
   - ✓ Generate basic plots (e.g., histograms, scatterplots) using base R.
9. **Subset Data**
   - ✓ Extract specific data using logical conditions (e.g., which(), []).

LET'S GO!

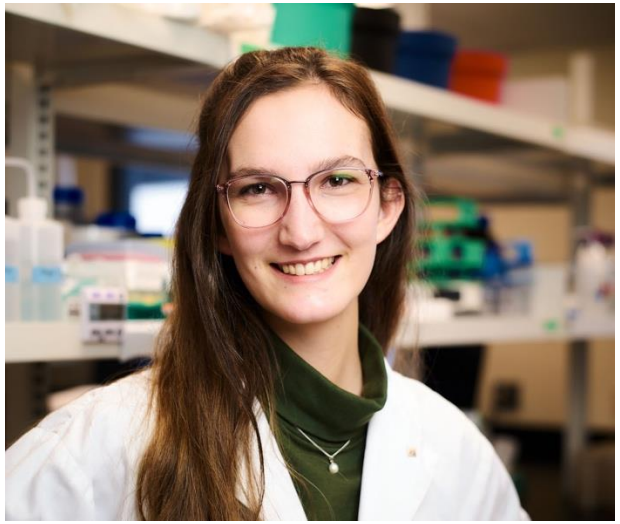
I have RStudio up and running

I have a folder on my Desktop named Rworkshop2025

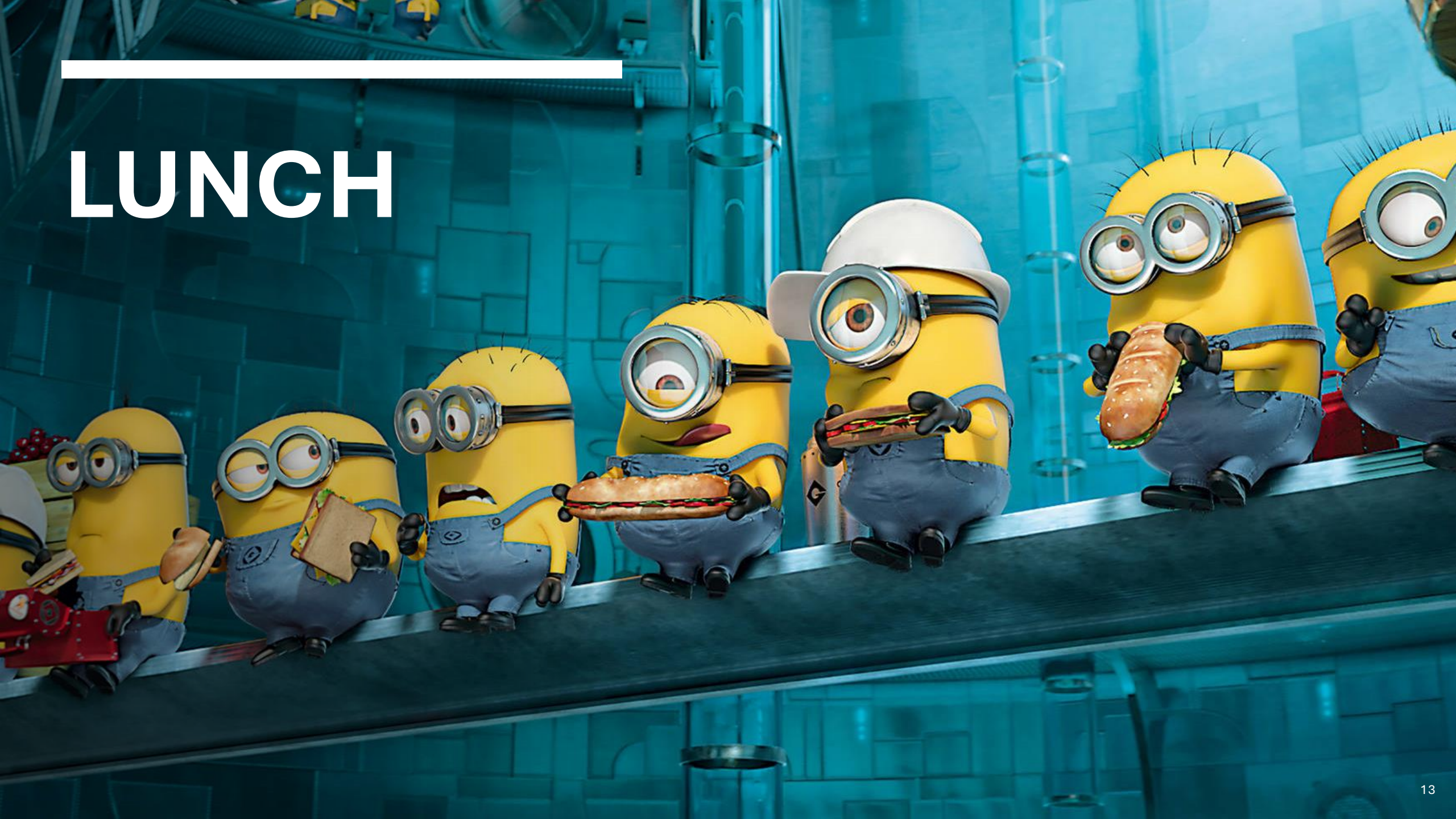
I have downloaded the data folder and put it in Rworkshop2025

I can put my yellow post-it up! I am ready to go!

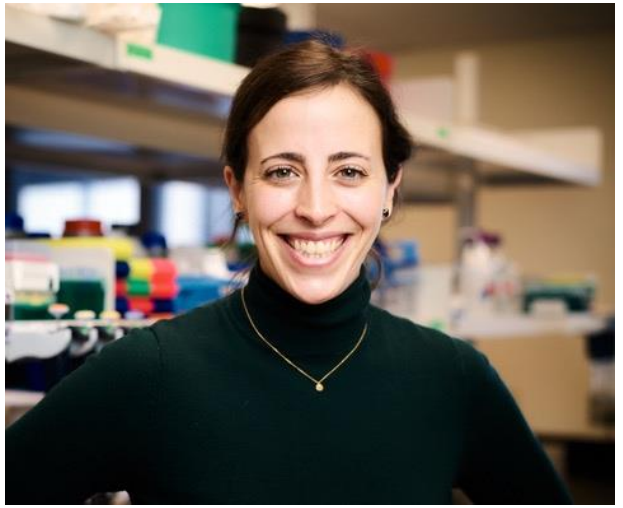
SESSION 1



LUNCH



SESSION 2



DAY1 RECAP (1)

1.R is an object-oriented language.

This means that objects are created and stored in your R environment and are then available for later use. What is an object? An object can be something as simple as a number that you assign to a variable (e.g., x <- 5) or an object can be a dataset or complex model output.

The other important thing about objects is that you will often create objects using the assignment operator, which is simply <-. **Do not use the equals sign (=)** to set something as an object (thought it will work in some cases), but stick with the assignment operator, and interpret it as the left side is the object name that is storing the object information specified on the right side.

2. R is case sensitive.

MyData and mydata are completely different terms in R and not at all the same thing. I suppose the tips here would be to avoid similar names for things and consider developing some capitalization or spacing routine that you can get used to and comfortable with.

https://bookdown.org/steve_midway/DAR/learning-r.html

DAY1 RECAP (2) Symbols

3. Some singular letters in R are actually functions

For this reason it is advisable to avoid using these letters for naming objects. Although you may get away with using these letters in some cases, it's best to just avoid them other than their use as a function. See the table below for a list of these letters and their associated function calls.

Letter to avoid	Function
c	concatenate
q	quit
t	transpose
C	contrasts
D	derivatives
I	keep object type
F	False
T	True

https://bookdown.org/steve_midway/DAR/learning-r.html

DAY1 RECAP (3) The Syntax

```r
# 10 samples from a normal distribution with mean = 0 and sd = 1
rnorm(n = 10, mean = 0, sd = 1)

# The same exact function as above, but split into separate lines
# This can be useful when there are a lot of arguments or they are complex
rnorm(n = 10,
      mean = 0,
      sd = 1)

# This is the same exact function as above, but no left side of arguments
# Use this when you get comfortable with arguments and their order
rnorm(10, 0, 1)

# Mean = 0 and sd = 1 are actually the default values, so this function
# is the exact same as the ones above
rnorm(10)

# This function is different from those above
rnorm(1, 10, 0)
```

https://bookdown.org/steve_midway/DAR/learning-r.html

DAY1 RECAP (4) Saving objects

```r
# Load CSV data
mydata <- read.csv('mydata.csv', header=T)

# Load TXT data
mydata <- read.table('mydata.txt', header=T)
```

https://bookdown.org/steve_midway/DAR/learning-r.html

DAY1 RECAP (5) What to save?

R Script This is one document you will want to save often. The script is where you are writing and editing your code, and it is worth saving your script (with the small save icon above the script) frequently.

Workspace Your workspace is the term for what you have compiled in that particular R session and saved in your R environment. So if you read a datafile into your environment or save a model as an object, that is now part of the workspace. You can save an R workspace and may be prompted to do so when you quit R. However, most often you will not want to save your R workspace because what you created in that workspace can be recreated within seconds by running your script again. So a general rule might be that if you are running models that take a long time to fit (yes, some complex models may take minutes or hours or days), they you will likely want to save that model object by saving that R workspace.

Data / Numerical Output You may have read in some data, but perhaps you manipulated it or otherwise generated something new that you would like to save. Maybe you need to send a collaborator a data file of something that was generated in R. One easy way to accomplish this is with the function write.table(), which simply writes a data file to your working directory.

```
# Write a (numeric) object to an external file (CSV)
write.table(x = trees, file = "treesdata.csv", sep = ",")
```

Plots Plots and figures are an object you will want to save.

Your Original Data You may have read into R a datafile that you are now working with. This file never needs to be saved. Once you read that file into R and it is an object, R has essentially made a copy of it and it not accessing or manipulating the original file. This will quickly become obvious as you work in R, but it is different from working in a software like Excel, where you do need to save the actual file (because the work is done in the file).

https://bookdown.org/steve_midway/DAR/learning-r.html