Part IV - Tips and Tricks

Practical tips for organizing scripts, files and folders

- working directory concept
  - setwd("path/to/folder/")

- a w.d. to each research project/experiment/analysis ... I suggest to avoid use the default R wd. After a bit you will have an incomprensible multitude of input files, outputs, from different projects, analysis, errors and reconsiderations ... trust me I already lived it XD.

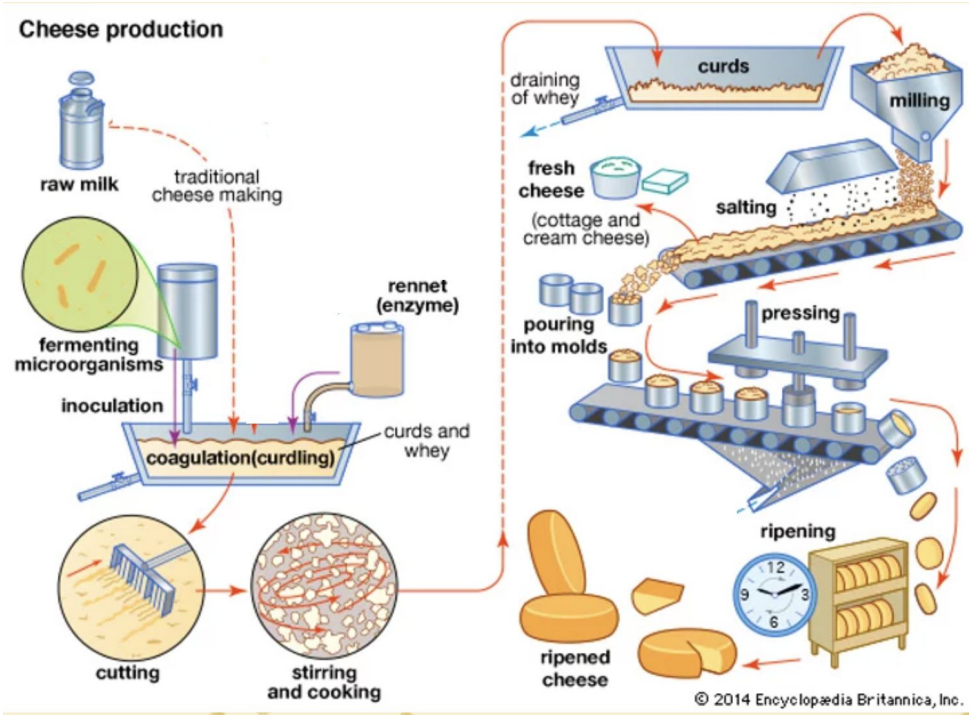
- long analysis need order (teutonic strengthness!) it is easy to create bugs in mis-organized scripts.

  - script "chapterization" using

```
# 1. Chapter 1 ------------------------------
## 1.1 Sub-chapter —-------------------------
```

    this will create tracking points in your script

practical tips for organizing scripts, files and folders



- long analysis needs order (tip2). Long scripts (I mean > 500 rows) are good for braggarts: several scripts generating intermediate results are better.

- only you know where to stop! my suggestion (actually is an example)

  script 1: data loading + basic transformations + preliminary data surveys

  script 2: main analysis with transformed data

  script 3: cool figures

Your project folder

README file
explains the content of the project and the folders/files contained in the project's main directory

RAW data folder
contains the untouched raw data. Ideally should have read-only permissions.
They must be stored in a very safe place, and possibly have a recovery backup.
Have you made them available after publication? This prevents to have your materials lost and is increasingly required by journals

Scripts folder
The tidy collection of all your script used in the several steps of data analysis

Analysis (or Results) folder
Contains the outputs of your analyses (elaborated tables, figures, etc..)

practical tips for organizing scripts, files and folders

- R scripts should be published to address reproducible research requirements. Don’t do the assholes, sharing is good for scientific community!

- You can add a nice header to your scripts including:
  - Paper title
  - Script author(s)
  - A brief explanation on its usage, if it is needed
  - R and Rstudio version you used
  - List of packages and their version

practical tips for organizing scripts, files and folders

```
# Herbonaute
# Adamo M., Marmeisse R.
# R script by M. Adamo

# R version 4.2.2 Patched (2022-11-10 r83330) -- "Innocent and Trusting"
# Copyright (C) 2022 The R Foundation for Statistical Computing
# Platform: x86_64-pc-linux-gnu (64-bit)

# used libraries -------------------------------------------------------
library("raster")         # Geographic data analysis and modeling v3.5-29
library("ggbiplot")       # A ggplot2 based biplot v0.55
library("factoextra")     # Multivariate data analysis v1.0.7
…
```

repositories

Some good places where to deposit your data (raw, elaborated or scripts)

panic! - warnings *vs* errors-

 # Warnings: Your function was able to run but there are some conditions that needs to be checked to avoid possible issues

```
> matrix(data = sample(seq(1,30)), nrow = 10, ncol = 5, byrow = F)
      [,1] [,2] [,3] [,4] [,5]
 [1,]    3   19    9    3   19
 [2,]    7   27   25    7   27
 [3,]   13    1   22   13    1
 [4,]   17   16    6   17   16

Warning message:
In matrix(data = sample(seq(1, 30)), nrow = 10, ncol = 5, byrow = F) :
  data length differs from size of matrix: [30 != 10 x 5]
```

panic! - warnings *vs* errors-

 # Errors: Your command cannot be executed

```
> library(unicorns)
Error in library(unicorns) : there is no package called ‘unicorns’
```

Several reasons behind:

- there is a typo in your command
- the function you typed does not exist or the library containing it has not be loaded
- the object name you typed has some typos or does not still exist
- your command does not comply with the R language syntax
  - e.g. there are unclosed brackets/quotation marks

panic! - warnings *vs* errors

Some useful tips:

- RStudio most of the times tells you that something is wrong with the syntaxis

- sometimes function names overlap between different packages.
  You need to get aware of that an force the function from the right package:

```
dplyr::filter
```

- functions can be updated over the time and their usage can have changed from the last time you used it. No way back: the best solution is to integrate the new function in the script.

panic! - Other strategies

- bug fixing
  - it is usually possible when problems are simple and/or you're experienced
  - check the presence of NAs or other lacks in the data
  - check the format of your input data
- manuals
  - almost all packages have the help page
  - some package have dedicated web-pages and online tutorials. *e.g.*: http://www.ggtern.com/docs/
- forums
  - "Google knows!"
  - you will probably ground in one of the several R-specialized forums

I get an error message that I don’t understand:

- Googling the error message
- Check Stack Overflow forum pages
- Ask your skilled colleagues (do not abuse of this solution)
- Ask on R-help mailing list

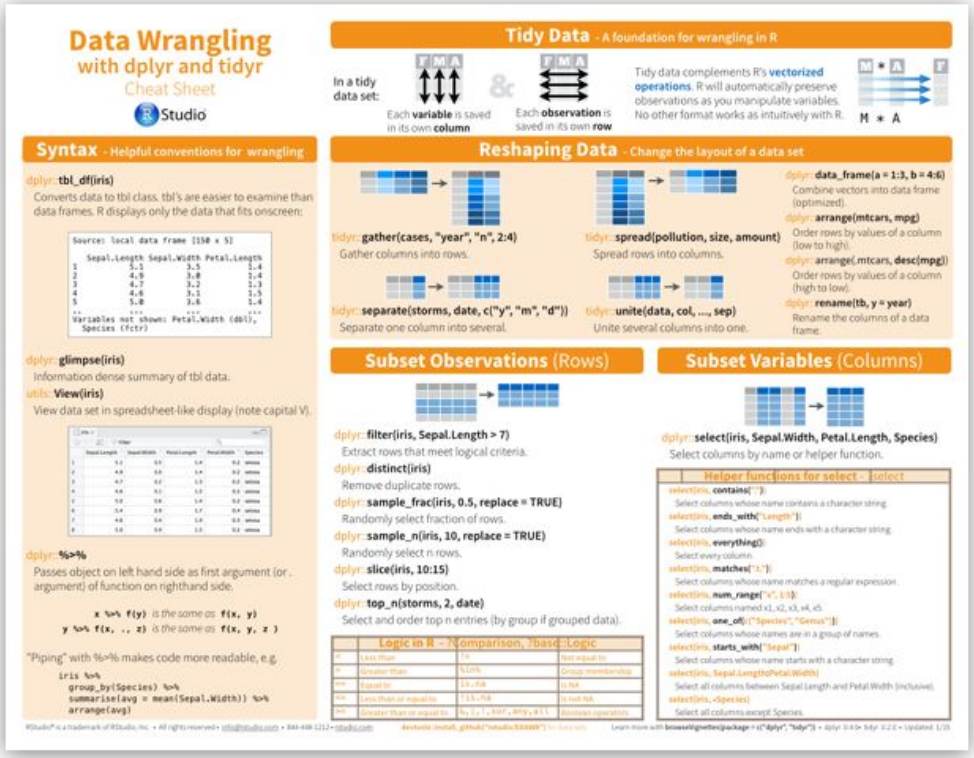
Note: Most of the time some other else already encountered your same issue, or a bug is already known (and not fixed, sure!)

How to make easier to help you

- Use the correct terms to describe your problem
- Try to generalize what you are doing
- Include the output of sessionInfo() in your requests
- Try to share a reproducible example, including your input data!

see here for further ideas: http://adv-r.had.co.nz/Reproducibility.html

Cheatsheets



You cannot known in detail all the function of a given package:

Cheatshets (“bigliettino”) help you in summarizing the main functions and their usage!