

Literate programming

Modèles Statistiques (MSPL)

Danilo.Carastan-dos-Santos@univ-grenoble-alpes.fr

Laboratoire LIG
Équipe-Projet Inria DATAMOVE

Grenoble 2024

UE MODÈLES STATISTIQUES

1 **REPORTING Thanks to GitHub SMPE**

- An IMRAD Report
- Good Practice for Setting up a Laboratory Notebook

2 **R/KNITR CRASH COURSE Thanks to GitHub SMPE**

- General Introduction
- Reproducible Documents : knitr

STRUCTURE

Research articles are often structured in this basic order :

Introduction Why was the study undertaken ? What was the research question, the tested hypothesis or the purpose of the research ?

Methods When, where, and how was the study done ? What materials/hardware were used ? How was it configured ?

Results What answer was found to the research question ; what did the study find ? Was the tested hypothesis true ? **Present useful results in a synthetic way with a logical order.**

Discussion What might the answer imply and why does it matter ? How does it fit in with what other researchers have found ? What are the possible bias and points to improve ? What are the perspectives for future research ?

Such structure **facilitates literature review** and is a very effective way to convey information.

If the report is a few pages long then **an abstract is required.**

STEP 0 : TAKING NOTES

Document your :

- ▶ **Hypotheses** : keep track of your ideas/line of thoughts
- ▶ **Experiments** : details on how and why an experiment was run, including failed or ambiguous attempts.
- ▶ **Initial analysis or interpretation** of these experiments : was the outcome conform to the expectation or not ? does it (in)validate the hypothesis ?
- ▶ **Organization** : keep track of things to do/fix/test/improve

Structure :

- 1 General information about the document and organization **conventions** (e.g., directory structure, notebook structure, experimental result storing mechanism, ...)
- 2 Documentation of **commonly used commands** and of how to set up experiments (e.g., git cloning, environment deployment, connection to machines, compiling scripts)
- 3 Experiment results can be either structured **by dates** (↪ add tags) or **by experiment campaigns** (↪ add date/time)

WHICH FORMAT SHOULD I USE ?

- ▶ Wikis are encouraged to favor collaboration but I do not find them really effective
- ▶ Blogging systems are also a way of managing such notebook but they should rather be considered as an effective way to share information with others
- ▶ I recommend to use basic plain-text format and to structure it hierarchically

Examples :

- Logging activities
- Logging experiments
- Structuring the repository

Last but not least :

Provide links to Raw Data!!!

WHEN/HOW OFTEN SHOULD I USE IT ?

If you don't document what you do, in the long term, you are just wasting your time.

Few advices

- ▶ Spending **more than an hour without** at least **writing** what you're working on **is not right**. . .
 - **Take a 5 minutes** break and ask yourself what you're doing, what is keeping you busy and where all this is leading you
- ▶ While working on something, you will often notice/think about something you should fix/improve but you just don't want to do it now. Take 20 seconds to write a **TODO** entry.
- ▶ There are moments where you have to **wait for something** (compiling, deployment, . . .). It is generally the perfect time for improving your notes (e.g., detail the steps to accomplish a TODO entry).
- ▶ **By the end of the day** : daily (and weekly) **review** !
 - Update your lists, write what the next steps are
 - **Summarize in a 2-4 lines** (for your advisor) what you did, what was difficult, what you learnt.

STEP 1 : SHARING CODE AND DATA

What kinds of systems are available ?

- ▶ "Good" - The cloud (Dropbox, Google Drive, **Figshare**)
- ▶ **Better** - Version control systems (SVN, **Git** and Mercurial)
- ▶ "Best" - Version control systems on the cloud (GitLab, GitHub, Bitbucket)

Depends on the level of privacy you expect but you probably already know these tools.
Few handle GB files...

Is this enough ?

- 1 Use a workflow that **documents both data and process**
- 2 Use the machine readable **CSV format**
- 3 Provide **raw** data and **meta** data, not just statistical outputs
- 4 **Never** do data manipulation and statistical tests **by hand**
- 5 **Use R**, Python or another free software to read and process raw data (**ideally** to **produce complete reports** with code, results and prose)

STEP 2 : LITERATE PROGRAMMING

Donald Knuth : explanation of the program logic in a **natural language interspersed with snippets of macros and traditional source code**.

This maybe is too complicated for traditional programming but that's **exactly what we need for writing a reproducible article/analysis !**

Org-mode (requires emacs)

- ▶ plain text, very smooth, works both for html, pdf, . . .
- ▶ allows to combine all my favorite languages even with sessions

KnitR (a.k.a. Sweave)

For non-emacs users and as a first step toward **reproducible papers** :

- ▶ Click and play with a modern IDE (e.g., Rstudio)

Ipython notebook

Jupyter Notebooks also possible. . .

UE MODÈLES STATISTIQUES

1 REPORTING Thanks to GitHub SMPE

- An IMRAD Report
- Good Practice for Setting up a Laboratory Notebook

2 R/KNITR CRASH COURSE Thanks to GitHub SMPE

- General Introduction
- Reproducible Documents : knitr

WHY R ?

R is a great language for data analysis and statistics

- ▶ Open-source and multi-platform
- ▶ Very expressive with high-level constructs
- ▶ Excellent graphics
- ▶ Widely used in academia and business
- ▶ Very active community
 - Documentation, FAQ on <http://stackoverflow.com/questions/tagged/r>
- ▶ Great integration with other tools

WHY IS SUCH R A PAIN FOR COMPUTER SCIENTISTS ?

- ▶ R is **not** really a **programming** language
- ▶ Documentation is for statisticians
- ▶ Default plots are *cumbersome* (meaningful)
- ▶ Summaries are *cryptic* (precise)
- ▶ **Steep learning curve** even for us, computer scientists whereas we generally switch seamlessly from a language to another ! That's frustrating ! ;)

DO'S AND DONT'S

R is high level, I'll do everything myself

- ▶ CTAN comprises 6,163 T_EX, L^AT_EX, and related packages and tools. Most of you do not use plain T_EX.
- ▶ Currently, the CRAN package repository features 18,674 available packages.
- ▶ How do you know which one to use ??? Many of them are highly exotic (not to say useless to you).

I learnt with <http://www.r-bloggers.com/>

- ▶ Lots of introductions but not necessarily what you're looking for so I'll give you a short tour. You should quickly realize though that you need proper training in statistics and data analysis if you do not want tell nonsense.
- ▶ Again, you should read Jain's book on The Art of Computer Systems Performance Analysis
- ▶ You may want to follow online courses :
 - <https://www.coursera.org/course/compdata>
 - <https://www.coursera.org/course/repdata>

SETTING UP THE ENVIRONMENT

The complete installation process is [here](#).

Git :

- ▶ <https://git-scm.com/downloads>
- ▶ Understand git and Tutorial : <https://try.github.io/>

R :

- ▶ <https://cran.r-project.org/>
- ▶ R for Data Science : <https://r4ds.had.co.nz/>

RStudio :

- ▶ <https://www.rstudio.com/products/rstudio/download/>
- ▶ [How to use FAQ](#)

VERIFYING THE INSTALLATION

Using the terminal :

```
git --version  
git version 2.25.1
```

If it shows up at RStudio

```
R --version  
  
R version 4.1.2 (2021-11-01) -- "Bird Hippie"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under the terms of the  
GNU General Public License versions 2 or 3.  
For more information about these matters see  
https://www.gnu.org/licenses/.
```

Any R version ≥ 4.0 should work fine. However, if you had previously installed R and it is version 3.x.x you probably want to update it!

SETTING UP YOUR GIT REPOSITORY

1 Login to your Gitlab account

- Go to the login page : <https://gricad-gitlab.univ-grenoble-alpes.fr>
- Using UGA Agalan login/password
- Documentation : Have a look at the Gitlab documentation
<https://docs.gitlab.com/ee/gitlab-basics/README.html>

2 Add your public SSH key

- Add it here : <https://gricad-gitlab.univ-grenoble-alpes.fr/-/profile/keys>
- Tutorial : <https://docs.gitlab.com/ee/user/ssh.html>

3 Fork the reference repository

- Fork this repository :
<https://gricad-gitlab.univ-grenoble-alpes.fr/MSPL/MSPL-2023-24/>
- Please, use the Fork button in the web interface
- Tutorial :
https://docs.gitlab.com/ee/user/project/repository/forking_workflow.html

4 Clone your forked repository locally

- `git clone git@gricad-gitlab.univ-grenoble-alpes.fr:YOURLOGIN/MSPL-2023-24.git`
- Tutorial : <https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html> –
<https://about.gitlab.com/images/press/git-cheat-sheet.pdf>

SETTING UP YOUR GIT REPOSITORY

5 Configure git :

- `git config --global user.name "MY NAME"`
- `git config --global user.email "MY EMAIL"`

6 Create your first commit, then push it to gitlab

- Create a folder `reports`
- Create a file `reports/README.md` with your name, binôme, which git you and your binôme will upload the reports, the team name with the 4 students names and emails :
 - Student Name: Danilo Carastan dos Santos
 - Student Git `https://gricad-gitlab.univ-grenoble-alpes.fr/carastda/mspl-2023-24`
 - Binôme: Jean Marc Vincent
 - Binôme Git: `https://gricad-gitlab.univ-grenoble-alpes.fr/jmv/MSPL-2023-24`
 - Team Name: The Professors
 - Jean Marc Vincent - `Jean-Marc.Vincent@univ-grenoble-alpes.fr`
 - Danilo Carastan dos Santos - `Danilo.Carastan-dos-Santos@univ-grenoble-alpes.fr`
 - Member 3 - E-mail 3
 - Member 4 - E-mail 4
- Then, run `git add reports/README.md`
- Commit it locally using `git commit -m "my first commit"`
- Push to github with `git push origin main`

7 Add MSPL-2023-24 remote

- Do in your local repository, the one that you've cloned.
- `git remote add MSPL-2023-24 git@gricad-gitlab.univ-grenoble-alpes.fr:MSPL/MSPL-2023-24.git`
- Check for updates: `git pull MSPL-2023-24 main`
- For other choices have a look at `fetch`, `merge` and `rebase` commands
- Push then to your gitlab repository: `git push origin main`
- Repeat the last two commands to keep your gitlab repository updated.
- This is necessary to keep yourself up to date with latest TD updates.

INSTALL A FEW COOL PACKAGES

- ❷ Install R packages. R has it's own package management mechanism so just run R and type the following commands :

- `dplyr`, `tidyr` and `ggplot2` by Hadley Wickham (<http://had.co.nz/>)

```
install.packages("dplyr")  
install.packages("tidyr")  
install.packages("ggplot2")
```
- Or install a bundle of many usefull packages (that includes those three) :

```
install.packages("tidyverse")
```
- `knitr` by Yihui Xie <https://yihui.org/knitr/>

```
install.packages("knitr")
```

RUN SOME MARKDOWN

- 9 Open one of the Markdowns inside `exercices/2-Installation-first-steps` :
 - `2_snow.Rmd` : If you want to check some snow research stations in France and successfully installed `tidyverse`
 - `2_ping-pong.Rmd` : If you want to check some computer communication latency experiments
- 10 Read the document and execute the chunks
- 11 Put your name on it !
- 12 Knit it to pdf (or word and save it to pdf), and copy the pdf to the reports folder of the binôme git.
- 13 `commit` and `push` it !

IDE

Using R interactively is nice but quickly becomes painful so at some point, you'll want an IDE.

Emacs is great but you'll need **Emacs Speaks Statistics**

```
sudo apt-get install ess
```

In this tutorial, I will briefly show you **rstudio** (<https://www.rstudio.com/>) and later how to use `org-mode`

RStudio

File Edit Code View Project Workspace Plots Tools Help

Go to file/function

Project: (None)

Workspace History

Load Save Import Dataset Clear All

Data

df 10 obs. of 2 variables

Values

x integer[10]

y numeric[10]

```
28 '''(r basicconsole)
29 x <- 1:10
30 y <- round(rnorm(10, x, 1), 2)
31 df <- data.frame(x, y)
32 df
33 '''
34
35
36 ## Plots
37 Images generated by 'knitr' are saved in a figures folder. However,
38 | they also appear to be represented in the HTML output using a [data
39 | URI scheme]( http://en.wikipedia.org/wiki/Data_URI_scheme). This
40 | means that you can paste the HTML into a blog post or discussion
41 | forum and you don't have to worry about finding a place to store the
42 | images; they're embedded in the HTML.
43
44
45 ### Simple plot
46 Here is a basic plot using base graphics:
47
48 '''(r simpleplot)
49 plot(x)
50 '''
51
52 '''(r simpleplot)
53 plot(x)
54 '''
55
56 plot(x)
57 '''
```

47:1 Chunk 3: simpleplot

R Markdown

Console

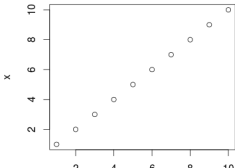
~/research/statistics/rmarkdown-meetup-2012/

'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

```
> set.seed(1234)
> library(ggplot2)
> library(lattice)
> x <- 1:10
> y <- round(rnorm(10, x, 1), 2)
> df <- data.frame(x, y)
> df
  x y
1 1 1.31
2 2 2.31
3 3 3.36
4 4 3.27
5 5 5.04
6 6 6.11
7 7 8.43
8 8 8.98
9 9 8.38
```

Files Plots Packages Help

Zoom Export Clear All



REPRODUCIBLE ANALYSIS IN MARKDOWN + R

- ▶ Create a new **R Markdown** document (Rmd) in rstudio
- ▶ R chunks are interspersed with ```{r}` and ````
- ▶ Open R block/chunk : `Control+Alt+i`
- ▶ Execute block/chunk : `Control+Alt+C`
- ▶ Execute current line : `Control+Enter`
- ▶ Inline R code : ``r sin(2+2)``
- ▶ You can **knit** the document and share it via **rpubs**
- ▶ I usually work mostly with the current environment and only knit in the end
- ▶ Other engines can be used (use rstudio **completion**)

```
```{r engine='sh'}``  
ls /tmp/
```
```

- ▶ Makes **reproducible analysis as simple as one click**
- ▶ Great tool for quick analysis for self and colleagues, homeworks, ...

REPRODUCIBLE ARTICLES WITH L^AT_EX + R

- ▶ Create a new **R Sweave** document (Rnw) in rstudio
- ▶ R chunks are interspersed with `<<>>=` and `@`
- ▶ You can **knit** the document to produce a pdf
- ▶ You'll probably quickly want to **change default behavior** (activate the cache, hide code, ...). In the preamble :

```
<<echo=FALSE>>=
opts_chunk$set (cache=TRUE, dpi=300, echo=FALSE, fig.width=7,
                warning=FALSE, message=FALSE)
@
```

- ▶ Great for journal articles, theses, books, ...

ACTIVITIES

- ▶ Incorporate the spirit of literate programming
- ▶ Basic statistical concepts (using R)
- ▶ Mean, Median, Min, Max
- ▶ Histograms, boxplots, summary
- ▶ How to interact with RStudio (and get a nice looking PDF/HTML)