Outline of four day workshop on Reproducible research/science

Required software to be installed beforehand

1. Git (<https://git-scm.com/downloads>)
2. Create a Github account on ()
3. Github desktop (<https://desktop.github.com/>)
4. R from (https://cran.r-project.org/)
5. RStudio (https://www.rstudio.com/products/rstudio/download/)
6. Reading (<https://guides.github.com/activities/hello-world/>)
7. Book about Github [https://orchid00.github.io/actions\_sandbox/]
8. Ggplot2 second edition [https://r-graphics.org/]
9. R Markdown book [https://bookdown.org/yihui/rmarkdown-cookbook/]
10. Rblogdown book [<https://bookdown.org/yihui/blogdown/structure-of-the-book.html>]
11. Doing mMeta analysis in R [https://bookdown.org/MathiasHarrer/Doing\_Meta\_Analysis\_in\_R/]
12. Ggraph[] https://cran.r-project.org/web/packages/ggraph/ggraph.pdf
13. <https://rpubs.com/thealk/academic-writing>
14. <https://happygitwithr.com/ssh-keys.html>

Day 01

Student group discussion based on distributed articles (60 minutes)

Student group presentation (28 minutes: 4 minutes per group x 7 group of 3)

Video for introducing the need of reproducible science or research (10 minutes)

Coffee break ( minutes)

**Morning session**

1. Reproducible research / science (45 minutes)

1.1 Definitions

* + 1. Replication
    2. Reproducibility
    3. Reproducible research

1.2. Background: Underlying trends

1.3. Outstanding problems: Complicated results

1.4. Data science pipeline: Authors vs.readers

1.5. Out of reproducibility realm

1.6. Reproducibility assumptions

1.7. Reproducibility as preventive measure

1.8. Who reproduces research?

1.9. Reproducers map

1.10. Reproducibility story so far

1.11. Evidence based data analysis

1.12. Evidence base data analysis 2

1.13. Desired data analysis map

1.14. Summary

**Afternoon session**

1. Reproducible research: Bridging the gap among actors

2.1. What is the existing gap?

2.2. How can we bridge the gap?

2.3. Research pipeline

2.4. Recent developments in reproducible research

2.5. What do we need to be reproducible?

2.6. Who are the players involved?

2.7. Reproducible challenges

2.8. In reality: What is common?

2.9. Literate (statistical) programming

2.10. Literate (statistical) programming 2

2.11. Sweave limitations

2.12. Alternative to sveawe: knitr

2.13. Summary

3. Dealing with data: Tidyverse [Olivier]

3.1. What is tidyverse and advantages

3.2.Tidyverse: well arranged collection of r packages

3.3. Flowchart of data science

3.4. Flowchart of data science in tidyverse

3.5. Practice in tidyverse “Use twitter to predict citation rate”

3.5.1. Import data

3.5.2. Tidy, transform

3.5.2.1. Rename columns

3.5.2.1. Clean up column names

3.5.2.3. Create and modify columns

3.5.3. Piping

3.5.3.1. Base r approach

3.5.3.2. Piping approach

3.5.4. Other functions in Tidyverse

3.5.4.1. Select columns

3.5.4.2.Drop columns

3.5.4.3. Split columns in multiple columns

3.5.4.4. Transform column in date format

3.5.4.5. Joint tables together

3.5.5. Character manipulation

3.5.5.1. Filter

3.5.5.2. Convert character to lowercase

3.5.5.3. Remove space in string of characters

3.6. Basic data exploratory analysis

3.6.1. Count()

3.6.2. Group variables to compute stats [summary()]

3.6.3. Order stuff [arrange()]

3.6.4. Work on several columns [dplyr:::across()]

3.6.5.Tidying tibbles [wide(), long()]

3.7. Cheat sheets

3.8. Summary

3.9. References

4. Data visualization in Tidyverse: the power of ggplot2

4.1. ggplot2: Introduction

4.2. ggplot2: Grammar

4.3. Examples of plots

4.4. Some manipulations [Look at slides by Resulumit 131-139]

5. Literate statistical programming: knitr

5.1. Problem

5.2. Literate statistical programming: Origin

5.3. Literate statistical programming needs

5.4. When to decide to work reproducibly?

5.5. Literate programming: Pros

5.6. Literate programming: Cons

5.7. knitr: Definition and usages

5.8. knitr: Requirements

5.9. What is Markdown?

5.10. What is knitr good for?

5.11. What is knitr not good for?

5.12. First knitr document as example

5.13. First knitr document as example 2

5.14. Processing a knitr document

5.15. Processing knitr document: Complex way

5.16 knitr to HTML output

5.17. What knitr produces: Markdown

5.18. A few notes

5.19. Processing knitr documents

6. R Markdown

6.1. Introduction to R Markdown [Olivier, Resulumit P39]

6.2. R Markdown workflow [P5]

6.3. Dissecting R Markdown structure [P6]

6.3.1. Details about YAML (Yet Another Make up Language)

6.3.1.1. YAML arguments / variables / Parameters

6.3.1.1.1 String

6.3.1.1.2. Options

6.3.1.1.3. Sub-options

6.3.1.1.4. Code

6.3.1.1.5. Output: “HTML”, “LATEX”, “pdf”, “Word doc”

6.3.1.1.6. R code

6.3.1.1.7. Further settings for pdf outputs

6.3.2. Text content

6.3.2.1. Language and syntax

6.3.2.1.1. Lines

6.3.2.1.2. Hard breaks [Resu P67]

6.3.2.1.3. Line blocks “|” [P68]

6.3.2.1.4. Block quotes [P69]

6.3.2.1.5. Paragraphs [P70]

6.3.2.1.6. Comments [P71]

6.3.2.1.7. Exercise

6.3.2.1.8. Headers [P73]

6.3.2.1.9. Emphases [P74]

6.3.2.1.10. Strikethrough [P75]

6.3.2.1.11. Exercise

6.3.2.1.12. Links-Internal [P77]

6.3.2.1.13. Links-External [P78]

6.3.2.1.14. Exercise

6.3.2.1.15. Equations [P80]

6.3.2.1.16. Footnotes – Inline notes [P81]

6.3.2.1.17. Footnotes – Notes with identifiers [P82]

6.3.2.1.18. Exercises

6.3.2.1.19. List [P84]

6.3.2.1.20. Lists Nesting [P85]

6.3.2.1.21. Lists Numbering [P81]

6.3.2.1.22. Dashes [P87]

6.3.2.1.23. Subscript and upperscript [P88]

6.3.2.1.24. Exercise

6.3.3. Code in and out chunks [P102]

6.3.3.1. Code overview [P103]

6.3.3.1.1. Code in chunks labels [P104-105]

6.3.3.1.2. In chunk -labels [P105]

6.3.3.1.3. Code chunks options [P106]

6.3.3.1.4. Code chunks options default

6.3.3.1.5. Code chunks-others useful options [ResulumP110]

6.3.3.1.6. Code chunk- other useful option [Cache, eval, error, message, warning, P111]

6.3.3.1.7. Code chunk- other useful options – Figure dimension [fig.height; fig.width; fig.caption; resolution; out.extra; P112-113]

6.3.3.1.8. Code chunks- The setup chunk [P114]

6.3.3.1.9. Code chunk-The data chunk [P115]

6.3.3.2. Inline code- Overview [P118]

6.3.3.2.1. Inline code- Example

6.3.3.2.2. Exercises

6.3.4. Special figure session

6.3.4.1. Figures – Image – Markdown syntax [P122]

6.3.4.2. Figures – dimensions options [P125]

6.3.4.3. Figure—Image-kntir [P126]

6.3.4.4. Figure-Image knitr dimension option [out.width; fig.align; P128]

6.3.4.5. Exercises

6.3.5. Speciation session on tables

6.3.5.1. Makrdown syntax [P141]

6.3.5.1.1.By hand [P142-145]

6.3.5.1.2. By hand but by grid [P146-147]

6.3.5.1.3. Exercises

6.3.5.1.4. By stargazer- Overview [P149-150]

6.3.5.1.4.1.Stargazer- Basics [151]

6.3.5.1.4.2. Stargazer-Data tables [P152]

6.3.5.1.4.2.1. Defaults [P153]

6.3.5.1.4.2.2. Modifying default [P154-159]

6.3.5.1.4..2.3. Exercises [P160]

6.3.5.1.4.2.4. Analyses tables [P161-166]

6.3.5.1.4.2.5. Exercises

6.3.6. Special session on citations| References [Good graphic: Olivier P54, get it together.people P59]

7 Version control

7.1. Version control overview [R P184]

7.2. Version control- Manual / traditional way [P186]

7.3. Version control- The power of Git

7.3.1. Version control- definitions [Git; Github; Repository or repo; to commit; to push, to pull; P188]

7.3.2. Command line interface

7.3.2.1. Definitions

7.3.2.2. Importance

7.3.2.3. Language

7.3.2.3.1. Directories

7.3.2.3.2. Directory structure

7.3.2.3.3. Special directories

7.3.2.3.3.1. Root /

7.3.2.3.3.2. Home ~

7.3.2.3.3.3. Path or the use of pwd

7.3.2.3.3.4. Interface commands

7.3.2.3.3.4.1. Definition of command

7.3.2.3.3.4.2. Clear

7.3.2.3.3.4.3. cd (change directory)

7.3.2.3.3.4.4. mkdir (Make directory)

7.3.2.3.3.4.5. touch (create an empty file)

7.3.2.3.3.4.6. cp (copy a file)

7.3.2.3.3.4.7. rm (remove a file)

7.3.2.3.3.4.8. mv (move a file)

7.3.2.3.3.4.9. echo (return whatever is after it)

7.3.2.3.3.4.10. date (For getting date)

7.3.2.3.3.4.11. summary of basic commands

7.4. Version control Git -Setup

git config –global user.name “name”

git config –global user.email [name@email.com](mailto:name@email.com)

7.5. GitHub – Setup

7.6. Basic git commands

7.6.1. Snapshot or overview [Lecture 5 P2, lecture 5 P3]

7.6.2. add (Adding files)

7.6.3. commit (Recording changes made)

7.6.4. push (pushing commit and changes from git to github)

7.6.5. Branches

7.6.6. Main

7.6.8. Pull

7.7. Core concepts in git (PPT Lecture 10)

7.7.1. commit (snapshots)

7.7.2. Graph (Relatiosnship between commits)

7.7.3. Changeset

7.7.4. Git record lines changes

7.8. States of git repository

7.8. States of git repository

7.8.1. What a repository?

7.8.2. Three local states

7.8.3. Three local states plus remotes

7.9. Git basic command

7.9.1. git init To initialize an empty repo

7.9.2. git add -Add document to staging area

7.9.3. Viewing repository

7.9.3.1 git status – What’s happening?

7.9.3.2. git log – To view history of the repository

7.9.3.3. git log – With options

7.9.4. git commit – Record changes in the repository

7.9.4.1. git commit what occurred

7.9.4.2. Good commit message

7.9.4.3. Git commit 7 great rules

7.9.4.4. Git rm

7.9.4.5. Git mv

7.10. Workflow visualized

7.11. Life cycle of status

7.12. Commit graph visualized

7.13. .gitignore [Resulmit 198]

7.14. .gitignore clear cache [P205]

7.15. Exercises [P206]

7.16. Collaboration

7.16.1. Pull: Definition [P209]

7.16.2. Merge: Definition

7.16.3. Merge conflict

7.16.4. Branch:Definition

7.16.5. Pull request [P210]

7.16.6. GitHub project setup

7.16.6.1. Project setup owner [P211-212]

7.16.6.2. Project setup collaborator

7.16.6.3. Exercises

7.16.7. Workflow

7.16.7.1. Pull/commit/push [P215]

7.16.7.2. Exercises [P216-218]

7.16.8. Workflow alternative

7.16.9. Alternative example [P220]

7.16.10. Exercises

7.17. Workflow notes Edit .Rmd [P222]

7.18. exercises

8. All in one: Git, GitHub, & R Studio

8.1. Setup R Studio [190-191,192-197]

8.2. Clone repository: From remote to locale

8.2.1. Clone public repository

8.2.1.1. Download zip: Traditional approach

8.2.1.2. Clone with https approach

8.2.2. Clone private repository

8.2.2.1. What is SSH?

8.2.2.2.Setup SSH key

8.2.2.2.1.Setup SSH key on Windows computer

8.2.2.2.2. Setup SSH key on Mac computer

8.2.2.3. Link SSH on GitHub

8.2.2.4. Clone with SSH

8.3. Commit in R Studio

8.4. Push in R Studio

8.5. Pull in R Studio

8.6. Merge conflict

git commit – Records changes in the repository

9 Workflow for reproducible work: Project

9.1. What is a project?

9.2. Good enough flowchart of a project

9.3. What goes into each part of a project?

9.3.1. Data folder

9.3.2. src floder

9.3.3. results folder

9.3.4. doc folder

9.3.5. README file

9.3.6. CITATION file

9.3.7. LICENSE file

9.4. Example: Good enough repository to follow

1. What is version control(30 minutes Morning + 230 minutes with 15 minutes Coffee break)
   1. Definition
   2. Examples of version control
   3. Introduction to Git
   4. Command line interface
   5. Basic Git command
   6. Introduction to Github

**Coffee break (**15 minutes**)**

* 1. R Studio
  2. Introduction to R-Markdown
  3. Knitr (Literate Statistical Programming)

**Day 02**

**Morning session**

1. Reproducible research / science workflow
   1. Project
   2. Data
   3. Doc
   4. Results
   5. SRC

**Coffee break (**15 minutes**)**

1. Create repository of a project- Practice in Git / Github / R

**Afternoon session**

1. Dynamic of repository through time
2. Connecting a local repository and remote: The power of Github

**Coffee break (**15 minutes**)**

1. Outstanding questions