Reproducible Research (1)

Zhuanghua Shi

14 April 2016

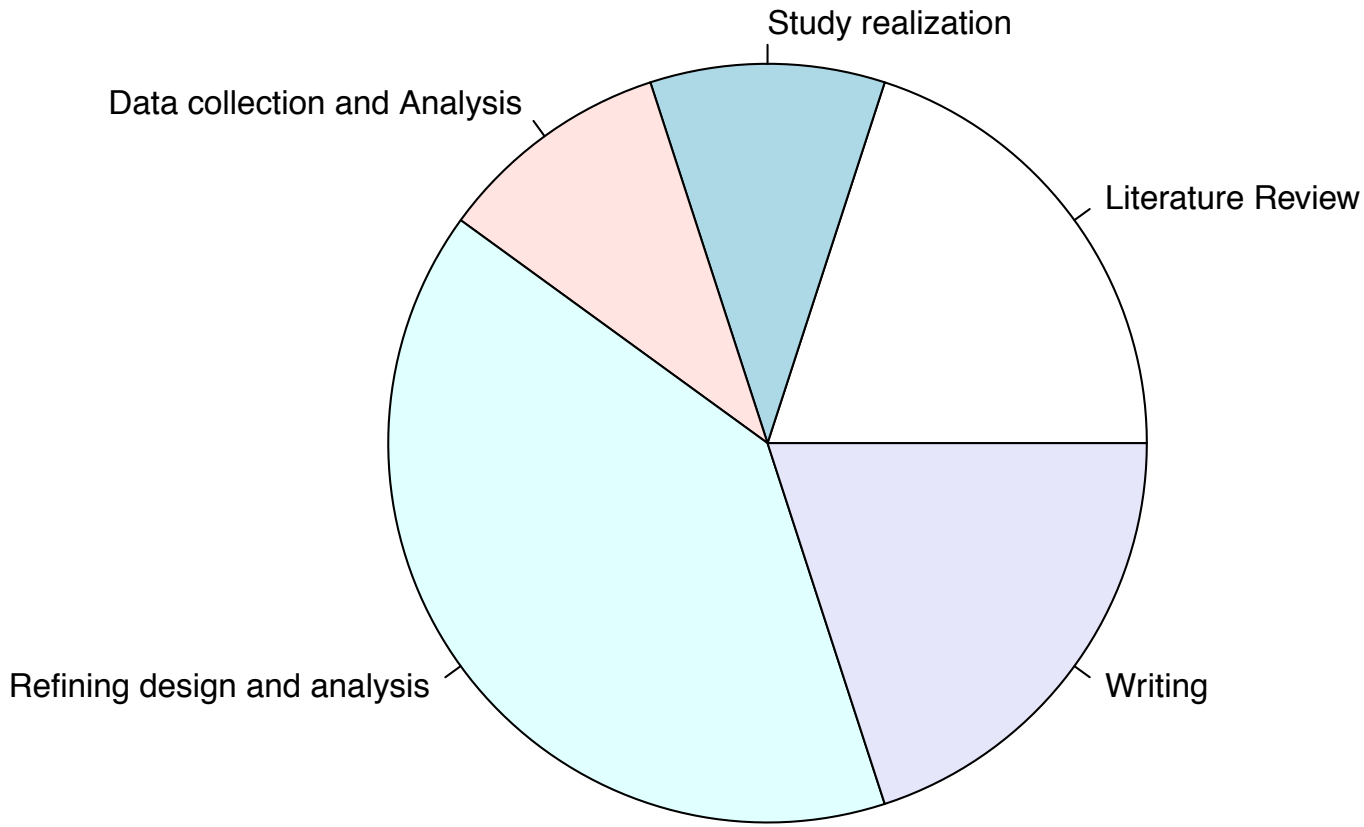
shared folder

https://goo.gl/1KIMnQ

A survey

- How many times do you usually repeat for analyzing the same dataset?
- Have you ever experienced big mistakes simply due to small coding errors or dumb copy-and-paste?
- Do you feel a need for filling a gap between data analysis and writing?

Estimation of my time spent on Research



Traditional approach

Traditional tools: **SPSS, Matlab, R** and **Word**

- problems occur for multiple experiments and data mining
- copy-and-past is error-prone
- data and files are separated, independent
- codes are not reusuable
- multiple copies

Reproducible Research (RR)

An ideal RR process should be able to adapt to flexible research process:

- easy to expand experiments and data analyses
- easy to maintain codes and text
- easy to replot figures and to output statistics

Git, R, Markdown, and Knitr /Pandoc

- git for version controls (e.g., multiple experiments, minor variations)
- R for data analysis
- RMarkdown for writing
- Knitr / Pandoc for universal document converters

Examples of RR

- An except from my own study - audiovisual temporal integration in AM
- The book 'Reproducible Research with R and RStudio'
- The book Pro Git
- A demo from Lakens/perfect-t-test [1] Lakens/perfect-t-test.git

[1] *Ondrej* mentioned this, thanks.

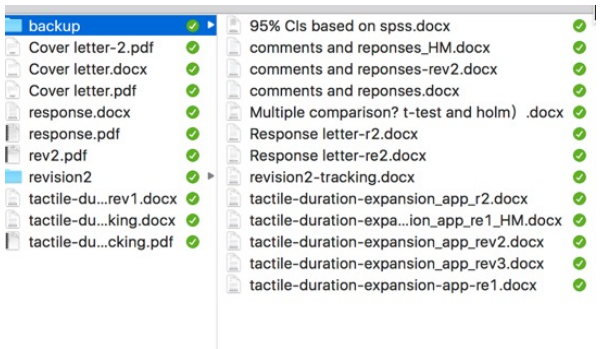
First step: organizing your data and files

- literature
- Mendeley
- experimental codes (matlab, R) and data
- dropbox / owncloud with git
- git for managing different versions and changes

multiple version of documents

Traditional manual version control:

- error prone
- difficult to find a right version



git can help you solve this problem

what is git

- A widely used source code version control system
- Distributed system
- Developed by Linus Torvalds (father of Linux system)

what is git

- Managing codes, files for collaboration
- Adopted by many applications
- Psychtoolbox
- Matlab
- R
- Open source codes
- Github.com
- Bitbucket.org
- ...

Benefits of using git in research

- Managing experimental codes
- Reusing codes for multiple experiments
- Tracking various changes
- Playground for your new projects
- Tracking your data analyses and writing
- Data minging requires multiple tries
- Multiple revisions of a manuscript
- Collaborating
- Parallel data analyses / writing

git basics

- Initializing a Repository in an Existing Directory

```
git init
```

- Tracking New Files

```
git add FILENAME
```

- Committing your changes

```
git commit -m "Version 9: new methods"
```

- Viewing your changes

```
git diff
```

- Viewing history

```
git log
```

Working with remove clouds

- Cloning an Existing Repository

`git clone https://github.com/christophergandrud/Rep-Res-Book.git`

- Adding remote repositories

`git remote add [shortname] [url]`

- Fetching and Pulling from Remotes

`git fetch [remote-name]`
`git pull [remote-name]` (difference from fetch: fetch and merge with local)

- Pushing to your remotes

`git push origin master`

- Tagging

`git tag`

Git GUI clients

You can find many git GUI clients in its official website, e.g.:

- SourceTree
- GitHub Desktip
- TortoiseGit

Git in RStudio and Matlab

Git is integrated in latest RStudio and Matlab. You can direct do git tasks inside R or Matlab.

Course schedule

dates	contents
2016-04-14	Version control system - git
2016-04-28	Data import and manipulation
2016-05-12	Publication-ready figures
2016-05-26	Statstics and Modeling
2016-06-09	Reusable data analysis
2016-06-30	RMarkdown and writing