Introduction to Git and GitHub

Christoph Kern

University of Mannheim

CONNECT@IPSDS 2018

10.06.2018

Outline

1. Introduction
2. Git
3. GitHub
4. Practical session
5. Resources

Introduction

Organizing coding projects

- Data/ coding projects often involve many iterations
- Organizing versions particularly difficult in collaborative projects
- Manual versioning (final_paper_version_x.doc) can become cumbersome

→ Version control management systems!

- Keep track of changes (**who** changed **what** code **when**?)
- Most recent version visible, previous versions can be restored
- Work best with text files (e.g. .txt, .R, .md, .tex)

Git and GitHub

(Why) Git and GitHub?

- Git is a version control management system
  - `https://git-scm.com/`
  - (Most) popular software for organizing coding projects
  - Other VCSs available, e.g. Subversion (SVN)
- GitHub is a **remote host** for **local** Git repositories
  - Widely used provider for sharing and storing Git projects online
  - Can also be used as a file hoster without Git
  - GitHub alternative; GitLab

Git

Working with Git

- Command line
- Graphical User Interface (GUI)
  - Git GUI
  - `https://git-scm.com/downloads/guis`
  - `https://desktop.github.com/`
  - RStudio

Git configuration

- First-Time Git Setup
  - `git config --global user.name "Your name"`
  - `git config --global user.email your@email.com`
  - `git config --global core.editor editor_name`

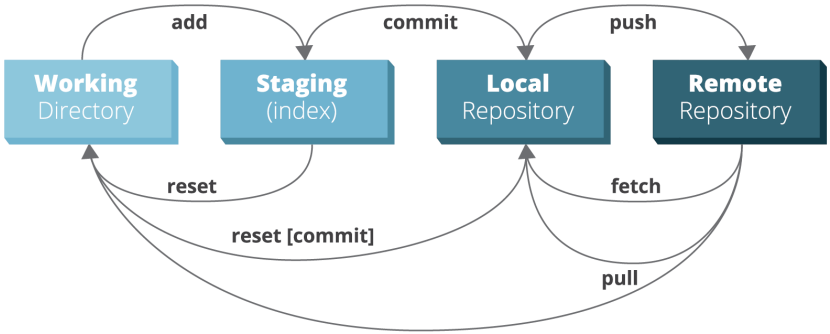
Git init

Initialize a Git repository

- Create a new (local) project
  - `git init`
- Create a new (local) R project
  - Initialize Git within an RStudio project
- Clone an existing remote project
  - `git clone`

Git workflow

Figure: Git workflow[1]



[1] https://zeroturnaround.com/rebellabs/git-commands-and-best-practices-cheat-sheet/

Example

```
> cd my_working_directory
> git init
# create some_file.R
# hack, do some work, hack
# hack
> git status # check current state of repository
> git add some_file.R
> git commit -m "Initial commit"
# hack
# more hacking
> git diff # show differences between file version
> git commit -am "Add important code"
> git log # list of previous commits
```

Git concepts

Branches

- A branch is a set of code changes that are kept separate
- Default branch created by `git init`: `master`
- Pointer to current branch: `HEAD`

`.gitignore`

- A file that tells Git which types of (e.g. temporary) files to ignore

Remotes

- A remote is an external repository to sync with
- Set up by `git clone` or `git remote add`
- Default remote name: `origin`

GitHub

Add a Git project to GitHub

1. Create a new clean repository via the GitHub web interface
2. Copy the resulting remote repository URL
   - `https://github.com/user_name/repo_name.git`
3. Add remote repo to Git project; `git remote add`
4. Interact with remote repo with `git push` and `git pull`

Notes!

- Free accounts only allow public projects (open read access)
- Paid plans can be used to set up private repositories
  - Never push sensitive information (e.g. passwords) to a remote repository!

Example continued

```
> git remote add origin url
> git remote -v # list remote repositories
> git push origin master # push changes to GitHub
# sleep
> git pull origin master # update local repository
# hack, hack, hack, make some changes
> git commit -am "Some changes again"
> git push # push to GitHub
```

GitHub flow

Workflow in a collaborative Git project

1. Clone or fork project and pull current version
2. Create a new branch and switch to it (`git checkout -b`)
3. Make changes, commit and push to remote branch
4. Create a pull request (via GitHub)
5. Proposed changes are reviewed and eventually merged into master branch (`git merge`)
6. Pull changes and tidy up branches (`git branch -d`)

Practical session

Setup: Install Git, get a GitHub account

1. Work with Git and GitHub via the command line
2. Collaborative coding with Git, GitHub and RStudio

Resources

- Pro Git book
  - https://git-scm.com/book/en/v2
- Cheatsheet
  - https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
- Resource collection
  - https://try.github.io/
- Git and GitHub
  - https://help.github.com/articles/git-and-github-learning-resources/
- Git and RStudio
  - https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

https://github.com/chkern/git-intro