

HuGen2080 book

Daniel E. Weeks

2024-01-04

Table of contents

Preface	3
1 Introduction	4
2 Logistics	5
2.1 GitHub: Set up an account	5
2.2 GitHub Classroom	5
3 GitHub	6
3.1 GitHub Introduction lecture	6
3.2 GitHub Introduction slides	6
4 Git Commands	7
4.1 git - best practices	7
4.2 Outline of essential Git commands	7
4.2.1 Initialization and Configuration	7
4.2.2 Basic Workflow	7
4.2.3 Remote Repositories	7
4.2.4 Status and Changes	8
4.2.5 History and Logs	8
4.2.6 Ignoring Files	8
4.2.7 Branching	8
4.2.8 Undoing Changes	8
4.2.9 Tagging	8
4.2.10 Stashing	8
5 Summary	9
References	10

Preface

This is a Quarto book created from markdown and executable code using Quarto within RStudio.

Book web site: <https://danieleweeks.github.io/HuGen2080/>

Book source code: <https://github.com/DanielEWeeks/HuGen2080>

Created by Daniel E. Weeks

Websites:

<https://www.sph.pitt.edu/directory/daniel-weeks>

To learn more about Quarto books visit <https://quarto.org/docs/books>.

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Logistics

2.1 GitHub: Set up an account

Please go to <https://github.com> and set up a GitHub account.

Choose your GitHub user name carefully, as you may end up using it later in a professional context.

2.2 GitHub Classroom

As GitHub Classroom will be used to distribute course materials and to submit assignments, it would be best if you get git working on your own computer. The easiest way to do this is to install RStudio, R, and git on your computer.

Please follow the detailed instructions in <https://github.com/jfikel/github-classroom-for-students>

In particular, see Step 5 re generating an ssh key so you don't need to login every time.

3 GitHub

3.1 GitHub Introduction lecture

Here's a recording of this lecture (32 minutes 8 seconds):

[Recording](#)

3.2 GitHub Introduction slides

[PDF slide set](#)

4 Git Commands

4.1 git - best practices

pull - work - commit - pull - push

- `git pull`
- Make changes
- `git commit` your changes to your local repository
- `git pull` the latest remote changes to your local repository
- `git push` your changes.

Pay attention to any error messages.

4.2 Outline of essential Git commands

Here's an outline of essential Git commands, initially created by ChatGPT:

4.2.1 Initialization and Configuration

- `git init`: Initializes a new Git repository in the current directory.
- `git config`: Configure Git settings.

4.2.2 Basic Workflow

- `git add`: Stage changes.
- `git commit -m "message"`: Commits staged changes with a descriptive message.

4.2.3 Remote Repositories

- `git clone`: Clones a remote repository to your local machine.
- `git push`: Send local changes to remote repository.
- `git pull`: Retrieve changes from remote.
- `git remote`: Manage remote repositories.

4.2.4 Status and Changes

- `git status`: Shows the current state of your working directory.
- `git diff`: Displays changes between working directory and the last commit.

4.2.5 History and Logs

- `git log`: View commit history.
- `git log --oneline`: Compact commit history.

4.2.6 Ignoring Files

- Create `.gitignore` file.

4.2.7 Branching

- `git branch`: List/create branches.
- `git checkout`: Switch branches.
- `git merge`: Merge branches.

4.2.8 Undoing Changes

- `git reset`: Unstage or reset changes.
- `git revert`: Create undoing commits.

4.2.9 Tagging

- `git tag`: Create and manage tags.

4.2.10 Stashing

- `git stash`: Temporarily store changes.

5 Summary

In summary, this book has no content whatsoever.

$1 + 1$

[1] 2

References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.