STAT 1010 Lecture Notes

Yi Wang

2023-08-11

Table of contents

Preface 3

1 Introduction 4

2 Setting-up Python Computing Environment 5

2.1 Use Google Colab 5

2.2 On your own computer 5

3 Setting-up R Studio Computing Environment 6

3.0.1 Setting up your own computing environment on a personal computer . . 6

3.1 Use R-Studio Cloud (No setting-up needed) 6

4 Use Git and GitHub 7

4.1 Download Git 7

4.2 Establish a connection between a local repo and a remote GitHub repo 7

4.2.1 Create your own 7

4.2.2 Clone an existing GitHub account 8

4.3 Some other common commands 8

4.4 When the upstream repo changes 9

4.5 Create branch 9

4.6 Merge branch to main branch 10

4.7 Collaborate directly by cloning the author’s github repo 10

4.8 Collaborate by fork a GitHub repo and commit the fork repo and create a pull request 10

5 Summary 11

References 12

Preface

This is a book for STAT 1010: Introduction to Data Science at Auburn University at Montgomery. The book is written using Quarto.

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

1 Introduction

This is a book for STAT 1010: Introduction to Data Science offered at Auburn University at Montgomery.

This an ongoing project and updates are perpetually added.

2 Setting-up Python Computing Environment

2.1 Use Google Colab

All you need is a Google account. Sign in your Google account in a browser, and navigate to Google Colab. Google Colab supports both `Python` and `R`. `Python` is the default engine. Change the engine to `R` in `Connect->change runtime type`. Then you are all set. Your file will be saved to your Google Drive or you can choose to send it to your `GitHub` account (recommended).

2.2 On your own computer

1. **Anaconda**: Download anaconda and install using default installation options
2. **VSC**: Download VSC and install
3. start VSC and install VSC extensions in VSC: Python, Jupyter, intellicode
4. (optional) **Quarto** for authoring: Download Quarto and install
5. Start an anaconda terminal. Navigate to the file directory.
6. Setup a conda **virtual environment**: stat1010 and install python and ipykernel engines

   ```
   conda create -n stat1010 python ipykernel
   ```
7. Activate the venv: `conda activate stat1010`
8. start VSC by typing `code .` in the anaconda terminal
9. open/create a `.ipynb` or `.py` file.
10. Select the kernel `stat1010`
11. Run a code cell by pressing `Shift+Enter` or click the triangular play button.
12. Continue to run other cells.
13. After finishing using VSC, close the VSC, and deactivate the virtual environment in a conda terminal: `conda deactivate`

3 Setting-up R Studio Computing Environment

3.0.1 Setting up your own computing environment on a personal computer

This is the recommended way and the advantage is that it's easy to handle files.

- Go to the website <https://posit.co/download/rstudio-desktop/>.
- Follow the two steps:

  1. download and install R: Choose the appropriate operating system, and then choose "base" to "install R for the first time". You can simply accept all default options.
  2. download Rstudio Desktop and Install it.

After installation, start R-Studio, and you are ready to use it.

3.1 Use R-Studio Cloud (No setting-up needed)

Alternatively, one can save the hassle of setting up on a personal computer and use the R-Studio Cloud for **free**. Here are the steps:

- Go to the website https://login.rstudio.cloud.
- Either create a new account using an email address such as your AUM email or simply "Log in using Google" or click on other log-in alternative.

After log-in to your account, you are ready to use R Studio.

4 Use Git and GitHub

I assume you already have an account on https://github.com. If not, you need to create an account there.

4.1 Download Git

1. Go to the website https://git-scm.com/downloads, select an appropriate operating system, select "Click here to download"
2. Run the downloaded setup file with a name such as `Git-2.42.0.2-64-bit.exe`, and accept all default options.

4.2 Establish a connection between a local repo and a remote GitHub repo

4.2.1 Create your own

3. Sign in to your github account.
4. Create a GitHub **empty** repo (such as named `homework0`) on `GitHub` (https://github.com) but make sure it is empty (do not add Readme.md file)
5. Start a Git Bash Terminal window on your local computer (You could also use the Terminal Window in RStudio or VSC). Navigate to the project directory; if you haven't yet created a project directory such as `homework0`, do

   `mkdir project_dir` Example: `mkdir homework0`

   Use `cd project_directory_name` to enter your local project directory;

   `cd ..` # back to the dir of the parent level of the current dir

   use `ls` to list all files and directories or use `ls -al` to include all hidden files and directories. In your local Git Terminal, (note at this moment your local project directory is empty)

```
echo "# homework0" >> README.md  #create a file README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main #rename the branch name to main
git remote add origin https://github.com/ywanglab/homework0.git #(change the remote to y
git push -u origin main
```

if your local project directory already 1) contains files and 2) had performed `init git` before, then

```
git remote add origin https://github.com/ywanglab/homework0.git` #(change the remote to
git branch -M main
git push -u origin main
```

6. in the pop-out `GitHub Sign in` window, click on `Sign in with your browser`.

4.2.2 Clone an existing GitHub account

This is an easier way to establish a connection between a local repo and a remote repo if the remote repo is created ahead.

`git clone https://github.com/ywanglab/tflite-pi.git` (change the remote repo to your remote repo)

4.3 Some other common commands

6. check git status: `git status`
7. `git add filename` or `git add .`# to add everything
8. use `git log` to check all commits. Use `git log --pretty=oneline` for shorter display.
9. use `git checkout .` to revert back to previous commit. Any changes after the previous commit will be abandoned.
10. to get to a previous commit, use `git checkout six_character_commit_ID`. To get back to main, use `git checkout main`.
11. To permanently go back to a previous commit, use

    ```
    git reset -hard six_char_commit_ID
    ```

12. `git remote -v` Get the reminder of the remote repo

13. if you want to remove the file only from the remote GitHub repository and not remove it from your local filesystem, use:

~~~~
```
git rm -rf --cached file1.txt # otherwise, remove --cached
git commit -m "remove file1.txt"

```

And to push changes to remote repo

```
git push origin branch_name
```
~~~~

14. you might need to tell GitHub who you are. To to this type the following two commands in your terminal window:

```
git config --global user.name "Your Name"
git config --global user.mail "your@email.com"
```

This will change the Git configuration in a way that anytime you use Git, it will know this information. Note that **you need to use the email account that you used to open your GitHub account**.

4.4 When the upstream repo changes

When Git tells you the upstream repo is ahead,

15. Do `git pull`. Then you can commit and push a new version to the remote repo.

4.5 Create branch

16. To add a branch to the main branch `git branch branchname`

Switch the branch `git checkout branchname`

Adding a file in branch `echo "#content" >> filename.txt`

Then `add` the file and commit the file. To create the branch remotely we **have to use**

```
git push --set-upstream origin branchname
```

4.6 Merge branch to main branch

Switch the branch again to the main using
`git checkout main`

on the `main` branch, Merge command to merge the branches
`git merge branchname`

4.7 Collaborate directly by cloning the author's github repo

16. `git clone remote-repo` to a local directory
17. create a new branch: `git branch [your_branch_name]`
18. `git checkout [your_branch_name]`
19. Submitting your changes for review

    1. **Commit your changes locally.** Once you are ready to submit your changes, run these commands in your terminal:

       ```
       git add -A                           # Stages all changes
       git status                           # Lists all staged changes
       git commit -m '[your commit message]' # Makes a git commit
       ```

       Make a pull request. A GitHub pull request allows a collaborator to review and make comments on your changes. Once approved, the collaborator can merge the changes. Run:

       ```
       git push origin HEAD # Push current branch to the same branch on GitHub
       ```

       Now, open the remote GitHub repo such as: https://github.com/ywanglab/textbook in your browser. You should see a **green** button titled "Compare & pull request". Click that button. Fill out the form on the resulting page with a title and description for your changes. Finally, click the "Create pull request" button.

4.8 Collaborate by fork a GitHub repo and commit the fork repo and create a pull request

20. after forking a GitHub repo to your own GitHub account, `git clone` that account to your local repo.
21. make changes to a file, and `git add, commit and push` the changes to the remote repo in your account.
22. Then go to your remote repo on the GitHub site and **Create pull request**.

5 Summary

In summary, this book has no content whatsoever.

References