STAT 1010 Lecture Notes

Yi Wang

2023-08-11

Table of contents

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

3.1 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.

3.2 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 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.

1. Sign in to your GitHub account, and create a GitHub repo (such as named `homework0`) on `GitHub` (https://github.com), you can add a README.md file or just choose not to add a README.md file.

2. On your local computer, open a `Git Bash` terminal. In the terminal, type `mkdir homework0` (create a folder named `homework0` within the current directory and then `cd homework0` (enter the directory `homework0`). The directory name `homework0` can be any name you want.

3. `git clone https://github.com/YourGitHubUserName/homework0.git` (change the remote repo to your remote repo)

4. Now you have established a connection between your local directory `homework0` and the remote repo `homework0` on GitHub.

5. Create a new file in the current local directory `homework0` on your local computer, such as using your favorite editor to create a file named `myfirstlocalfile.txt` with any content in it. Or for the sake of demonstration, you can use the following Linux command to create this file containing the line `#My first local file`.

   ```
   echo "#My first local file" >> myfirstlocalfile.txt
   ```

6. In the terminal, `git add .` This will add all changes to the **staging area**.

7. Now you are ready to **commit** the changes, which versions the current files in the directory.

   ```
   git commit -m "my first commit from local"
   ```

8. Now you are ready to sync with the remote repo.

   ```
   git push
   ```

   The GitHub might ask you to sign in for the first time. Choose `Sign in with your brower` to sign in to complete it.

4.2.2 Initializing a Git Directory Locally First

The previous approach initializes a local Git repo by cloning a remote repo. You can also initializing a local Git repo by using `git init`. Follow the following steps:

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;

   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 branch -M main #rename the branch name to main
   git add .  # maybe be changed to git add --all
   git commit -m "first commit"
   ```

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

Note:

1) the general command format: `git push [remote-name] [branch-name]`

2) difference between `git add .` and `git add --all`: `git add .`: stages changes in the current directory and its subdirectories but does not include file deletions `git add --all`: stages changes in the entire working tree, including deletions and untracked files. It is a more aggressive option and can be useful when you want to ensure that every change, including file deletions, is included in the next commit.

   `git add -all` is equivalent to `git add -A`

6. 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
```

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

8. Note an empty folder would not be pushed to the remote repo until it has a file (even empty) in it.

4.3 Some other common commands

6. check git status: `git status` and `git status --short` for a compact way.

7. `git commit -a -m "message"` will stage and commit every changed, already tracked file without using `git add changed_file`

8. `git add filename` or `git add .`

   # add changes to the staging environment, i.e., git repo to start track those changes.

9. use `git log` to check all commits. Use `git log --pretty=oneline` for shorter display.

   `git log origin/main` #check the remote repo `origin/main` commits

10. use `git diff origin/main` to show the differences between our local `main` and `origin/main`.

11. use `git checkout .` to revert back to the previous commit. Any changes after the previous commit will be abandoned.

12. to get to a previous commit, use `git checkout six_character_commit_ID`. To get back to main, use `git checkout main`.

13. To permanently go back to a previous commit, use

    `git reset -hard six_char_commit_ID`

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

15. 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 #This will only remote files; If intending to remove local file
git commit -m "remove file1.txt"
```

And to push changes to remote repo

```
git push origin main
```

14. For some operating system, such as Mac or Linux, you might be asked to tell GitHub who you are. When you are prompted, 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**. `global` sets the username and em-mail for **every repo** on your computer. If you want to set the username/e-mail just for the current repo, remove `global`.

4.4 Use Git help

1. `git command -help` See all the available options for the specific command. Use '`--help` instead of `-help` to open the relevant Git manual page.
2. `git help --all` See all possible commands

4.5 When the upstream repo changes

When Git tells you the upstream repo is ahead,

15. Do `git pull` or `git pull origin`

    This is equivalent to `git fetch origin`, and then `git merge origin/main.` Then you can commit and push a new version to the remote repo.

16. `git pull` will not pull a new branches on the remote repo to local, but it will inform you if there is a new branch on the remote repo. In this case, just `git checkout the_remote_new_branch_name` will be pulled to local. Note there is **no need** to create the branch by `git branch the_remote_new_branch_name`

4.6 Create branch

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

    Switch the branch `git checkout branchname`

    To combine the above two actions, `git checkout -b branchname`, create a new branch named `branchname` if it does not exist and move to it.

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` The option `--set-upstream` can be replaced by `-u`

to see all branches in both local and remote: `git branch -a` Or `git branch -r` for remote only.

4.7 Merge branch to main branch

Switch from a branch (with name such as `branchname` to the `main` using
`git checkout main`

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

To delete a branch:

`git branch -d branchname`

4.8 Contribute 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, short for --all
    git status                        # Lists all staged changes
    git commit -m '[your commit message]' # Makes a git commit
    ```

    Make a pull request. (A pull request is a proposal to change) A GitHub pull request allows a collaborator to review and make comments on your changes you proposed. 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 that you cloned such as: https://github.com/ywanglab/stat1010 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.

4.9 Contribute 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 git push origin HEAD` the changes to the remote repo in your account.
22. Then go to your remote repo on the GitHub site and **Create pull request**.

4.10 Project

1. Please git clone the course directory

```
git clone https://github.com/ywanglab/stat1010.git
```

to your local computer

2. create a new branch called: `YourFirstName_branch` by

`git branch YourFirstName_branch`

3. enter this branch:

`git checkout YourFirstName_branch`

4. add your resume file in the folder `./resume`

`git add`, `commit` and `push` your changes to the upstream repo using

`git add .`

`git commit -m "added YourFirstName's resume"`

`git push origin HEAD`

5. Now, open the remote GitHub repo that you cloned such as:https://github.com/ywanglab/stat1010in 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.

5 My Jupyter Notebook

Yi Wang (boldfaced using ** **)

Educator AUM

The following line is italicized using * *

I am interest in data science because it is a discipline that I feel love with.

5.0.1 Perform addtion

```
# code block
1+1
```

2

5.0.2 Horizontal Rule

Three or more

first rule using ***

---

using dashes —

Using (underscores) ____

5.0.3 Bulet list

using *

- Bird
- Frog
- Cat
- Dog

5.0.4 Numbered list

using 1. item (there is a space between 1. and item)

1. Apple
2. Pear
3. Peach

5.0.5 Tables

left-aligned	centered	right-aligned
1/2/2020	Mary	Apple
1/3	Johnason	Tomato

5.0.6 Hyperlinks

Click here to access my github account.

5.0.7 Images

Figure 5.1: A computer monitor

5.0.8 Code/Syntax highlighting

```
s = "Python syntax highlighting"
print s
```

5.0.9 Blocked quotes

using >

> Blockquotes are very handy in email to emulate reply text.
>
> This line is part of the same quote.

5.0.10 Strikethrough

using ~~ before and after a phrase

~~strikethrough this~~

6 Summary

In summary, this book has no content whatsoever.

References