GitHub practical

##############################
PART 1: Check if git is installed and if not, install it
Git is installed by default on Mac and Linux systems.

In Windows
Go to a Command Prompt window and type:
git --version

If not, then go to: https://git-scm.com/download/win and run the download.
Accept all default options.

##############################
PART 2: Create a GitHub account
1. Create a GitHub account at:
https://github.com/join
Use your LSHTM address because then you can get a free academic account.
2. When logged in, find the green “New” button to create a new repository
3. Give it a name: “test_repo” and a description.
4. Make it public
5. Press create.

##############################
PART 3: Using Git inside RStudio
Create a new project in R Studio from this repository and add some files
1. In Rstudio, click:
File —> New Project —> Version Control —> Git
You may need to restart RStudio after installing Git in Part 1.
2. Enter the details of the repository you just created
3. Click create project
4. Create an R script file, and make some edits to it, save it.
5. Click on the Git tab of the RStudio window.
6. Select the newly created file, and press commit. A new window comes up, why?
7. Add a commit message (make it useful!), and then commit.
8. Push the commit. Check on GitHub.com to see if the commit has appeared.
9. Make some more changes to the script, commit them, and push them.

Emphasis: you don't have to change file names, or label with dates, the repo tracks every change.
It records everything, FOREVER.

There are lots of public repositories where you can see a graph of activity and contributors.

##############################
PART 4: Using GitHub Desktop
Git can track a range of file types, and can be used separately from RStudio projects.
There is also a helpful GUI made by Github that makes things easier.

This is an exercise in using that app.

1. Download and Open GitHub Desktop
2. There should be 4 options underneath “Let’s get started!”
3. Click “Create a New Repository on your Hard Drive”
4. Fill in the details about your new repository
5. Create Repository
6. Once you’ve created it, it asks if you would like to publish to GitHub. Say yes!
7. What happened now? Hint: check on GitHub.com in your account
8. The app asks if you want to view the Files in the repository
9. Say yes. Is there anything there?
10. Go back to RStudio, and create a new R script. Write something, then save the script into the repository.
11. What has happened in the GitHub Desktop app?
12. Now we are going to commit this change. In the app make sure the file is selected in the check box. Write a commit message (make it useful, but not too long). Press commit.
13. What has happened? Has the file appeared on GitHub.com?
14. The blue button asks if you want to push to the origin. Say yes.
15. What has happened now? Has the file appeared on GitHub.com? Can you see the commit message?
16. Go back to RStudio. Make some changes to the script. Save it.
17. Check in the GitHub app. What has happened?
18. Add a commit message, commit, and push.
19. Go back to RStudio, make another script and save it in the same folder.
20. Commit and push.

Now what?
1. Find the repository folder.
2. DELETE IT!
3. Go back to the GitHub app. What does it say?
4. Press “Clone again”.
5. What has happened?

What if the repository exists on Github but you want to work on a different computer?
1. Delete the repository folder again.
2. In the app, this time click “remove”.
—> OH NO! (or is it…)
3. Back at the main menu, click “Clone a Repository from the Internet”
4. Select the URL tab, and put in the URL of the repository on GitHub, click Clone.
5. Check for the folder on your computer.
It’s back! (or it should be)

If you were working on this analysis on 2 (or more) computers, git can help you stay up to date.
But, the difference is you need the “Pull” button as well.
What does this do?

Key points:

- Whenever you save a file in the repository (the special folder) the App will track it.
- If you want to keep those changes, commit and push them to GitHub.com.
- Write a useful commit message for each one, to help you remember what they do.
- Make each commit quite small (~2-5 per day, if you're doing analysis all day)
- More than 1 person can do this at the same time and it will help you keep track of all the changes.

Extension:
What does Git Ignore do?

##############################
PART 5: Pair up to collaborate on a repository

1. Invite your partner to your repository (on GitHub.com)
2. Open this repository in RStudio. Figure out how to do this.
3. Modify a file in this repository and then save it.
4. Commit the file & write a message, and push it. Check on github where it has appeared.
5. The partner should now pull the file. What happened here?
6. Now repeat, the other way around!

Extension exercise: try to generate (and then solve) a merge conflict.

Well done!