Doing reproducible research
Best practices and practical tools for the social sciences

Practical

Alexandru Cernat

To do before course

1. Install the latest version of R: cran.r-project.org
2. Install the latest version of Rstudio Desktop: rstudio.com/products/rstudio/download/#download
3. Make an account on GitHub: github.com
4. Install Git: git-scm.com/downloads
5. Make sue you can install packages. Open Rstudio and in the console write: `install.packages("rmarkdown", dep = T)` and press Enter. *Alternatively you can go to the menu: Tools -> Install Packages, Write "rmarkdown" and press Install.*

Some institutional machines have issues installing R packages. If you get an error talk to IT.

Optional:

If you are a student or work at an educational institution you can get a Pro GitHub account for free which enables you to have private repositories: education.github.com/benefits/offers

Activities at the course

R projects

We will start by making a Rproject. This is a useful way to organize your work. It sets up the working directory for you and facilitates using packages like `renv` (which makes sure you will get the same outputs in the future) and git (for version control).

1. Open Rstudio and make a new Rproject. In the menu at top go to: File -> New Project.
2. Select "New Directory" and then "New Project".
3. Give a name for the directory (like "test_reproduce").
4. Press "Create project".

Reproducible documents

Next you will create your first reproducible document!

1. Got to File -> New File -> R markdown.
2. Give it a name and an author and make sure "HTML" output is selected.
3. Press "Knit" *(you will be prompted to save the file and give it a name)*.

Optional

- have a look at the description of the file here: bookdown.org/yihui/rmarkdown/basics.html
- you can also have a look at this useful cheatsheet:

raw.githubusercontent.com/rstudio/cheatsheets/master/rmarkdown-2.0.pdf

- you can play around with the document. Modify it and then press "Knit" to see how the output changes.

Linking Rstudio and GitHub

Next we will try to link Rstudio and GitHub. This can be tricky as it depends a lot on the operating system and the settings you have. If any of the steps bellow do not work, I would recommend you either go through the steps and troubleshooting presented in this online book: happygitwithr.com/hello-git.html or using a Git interface (like desktop.github.com) which can help considerably with the set-up.

1. Install the package `usethis` with this code:

```
install.packages("usethis")
```

2. Load the package using:

```
library(usethis)
```

3. Next tell R your GitHub username and email (replace what is in " " with your info) *(replace names accordingly)*:

```
use_git_config(user.name = "Jane Doe", user.email = "jane@example.org")
```

Trying out link between Rstudio and GitHub

There are a couple of different workflows for using Rstudio and GitHub together. Probably the simplest way to start is by making a project on GitHub that you link to Rstudio. Then you can "Push" (send things from local computer to GitHub) and "Pull" (get changes from GitHub on your computer) easily. Let's see if we can make this work. *Like I mentioned before there, are multiple reasons why this could fail so do check out this online book: happygitwithr.com/hello-git.html or using a Git interface (like desktop.github.com).*

1. Got to GitHub, log in, and press "new" (github.com/new).

2. Copy the link of the repository. It should appear under "SSH" and end with ".git". For example: "https://github.com/alex-cernat/test.git"

3. Let's make a new project where we "clone" the one we created in GitHub. This way we can link our computer to GitHub. Got to File -> New Project. Then select "Version control", next select "Git" and paste the URL you copied in the previous step. Select a local folder where you want to work. Click "Create project".

4. Open a new Rscript (File -> New File -> Rscript).

5. Write in it "# Testing".

6. Save it as "README.md".

7. Find the "Git" tab in Rstudio and press "Commit" (or go to Tools -> Version Control -> Commit). Select all the files (click the squares next to them). Press "Commit" *(remember that we need to write a message for each commit, so try to describe what you are submitting under "Commit message").* Then press "Push". When we "Commit" we are telling the local Git version control to remember the changes we have made. When we "Push" we are sending these changes to the server, on GitHub.

8. Go back to the GitHub web page for this project and refresh the page (F5). You should now see the message you have sent. Congratulations! You have done your first commit and push to GitHub. Now you and collaborators can use GitHub for version control and sharing files. Using Git and GitHub is very flexible but also quite complicated so this should just be your first steps in getting proficient with these tools!

Optional

- If you want, you can edit the readme file on GitHub and save it. Then go to Rstudio and pull the latest version of the file (Tools -> Version Control -> Pull Branches).
- You can add more information to the readme file and also format it see: garrettgman.github.io/rmarkdown/authoring_basics.html.
- You can also make a new R script, try writing some code and submit that as well.

Refrences

- Bryan, J. & Hester, J. (2021). *Happy Git and GitHub for the useR.* https://happygitwithr.com/index.html
- Rmarkdown cheetsheet: raw.githubusercontent.com/rstudio/cheatsheets/master/rmarkdown-2.0.pdf
- Xie, Y., Allaire, J. J., Grolemund, G. (2021). *R Markdown: The Definitive Guide.* https://bookdown.org/yihui/rmarkdown/