

In-class assignment

Michelle Donzallaz

22 March 2023

Git and Rmarkdown

Get to know git

1. Make a new repository.
2. Clone it using RStudio to make a local repository.
3. Write a (short) README file for your test repository using Markdown formatting.
4. Add your unstaged files, commit the changes
5. Before you can push the changes to your repository, you need to set up PAT (personal access token) or SSH based authentication. This is part of the general set-up and does not need to be repeated often. There are several ways of doing this and you are of course free to choose your preferred method. I recommend (aka this is how I did it ;-)), you use PAT based authentication and follow steps 1-7 outlined *here*. **After step 7**, copy the PAT and run the following command in R and follow the steps:

```
credentials::set_github_pat("Your_token")
```

6. Now push your changes to your remote repository!



Branch out

Now it is time to make a development branch. You can also do this within RStudio but let's try out the terminal!

5. Open the terminal in RStudio. Make sure you are in the right directory. Type:

```
git checkout -b dev
```

This moves you to the new branch called dev. You can also see it in the git interface in RStudio (typically top right panel).

Now you are ready to go to work!

Interact with git repositories and use rmarkdown

Now it is time to put your new knowledge on Rmarkdown and git to good use.

6. Open a new Rmd file and save it in the git repository.

We will now use a data set from another repo for a simple analysis. This *repository* contains data from several cognitive tasks. The *R file* in the repo loads and cleans the data.

7. Source the R-file from github directly. *Hint: Google is your best friend*

You should now have the data set `stroop` loaded in R.

```
##      ID congruency RT accuracy cond trial    rt
## 6     1 incongruent 701          1     2     6 0.701
## 7     1 incongruent 838          1     2     7 0.838
## 8     1 incongruent 659          1     2     8 0.659
## 9     1 congruent 739           1     1     9 0.739
## 12    1 congruent 670           1     1    12 0.670
## 13    1 congruent 626           1     1    13 0.626
```

We want to analyze the response time (`rt`) as a function of condition (`cond`).

8. Conduct a paired t-test for the stroop task data.

Hint 1: You first need to aggregate the data.

Hint 2: This should be the results

```
##
## Paired t-test
##
## data: mean_rt by cond
## t = -15.089, df = 120, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## -0.07350869 -0.05645509
## sample estimates:
## mean difference
## -0.06498189
```

9. Make sure to knit your Rmarkdown document either to html or pdf.

Back to git

10. Now add, commit, and push both the rmd file and the result file.

You can either use the RStudio git interface or the terminal. In the terminal, type:

```
git add myfile.Rmd myfile.html
git commit -am "my commit message"
git push --set-upstream origin dev
```

11. And then merge the `dev` branch into `master`. You can do it in your remote repository on the github webpage. Or you can use the terminal for this:

```
git checkout master
git merge dev --no-ff -m "merge dev"
git push
```