GITHUB 101 FEAT. TERMINAL

SCHOOL OF
CULTURE AND SOCIETY
AARHUS UNIVERSITY

23 JANUARY 2022

SIGURD FYHN SØRENSEN
STUDENT TEACHER

TODAYS PLAN:

1. **Introducing** the **plan** for today and myself. ;)

2. **Summarise** the **plan** for **classes** throughout the semester.

3. Introducing **GitHub**, **Terminal** commands and **Git.**

4. Short **resume** of what we **learned last semester.**

5. **Exercises:**
    1. Creating a GitHub and uploading first file.
    2. First look at your selected datasets
    3. Repeating what we learned last semester.

WHAT TO EXPECT THIS SEMESTER.

- **Roughly** the same as last semester.

What will be the same:

- ME!!!
- Weekly exercises.
- Portfolio.
- Coding in R.
- Solution files (the following week).

What will be different

- Chris!! (wuhuu!!)
- More math & a deeper understanding.
- You have to be more independent (no pre-made Rmd so use the book).
- Using Github to push your weekly exercises and portfolios.

LINEAR:

- **Why is something linear?**
  - **Chris's** analytical method

  - **Sigurd version** (dumb but easy).
    - In order for a function to be linear f'(x) = Constant and f''(x) = 0.
      - $f(x) = B_0 + B_1 x_1 + B_2 x_2^2 + error$
      - $f'(x) = B_1 + B_2 * 2x_2 \ + error$; $f''(x) = B_2 * 2$

      - $f(B) = B_0 + B_1 x_1 + B_2 + x_2^2 + error$
      - $f'(B) = 1 + x_1 + x_2^2 + error$; $f''(B) = 0$

GITHUB, TERMINAL AND GIT?!?!

Terminal:

WHAT IS IT!?

- An old school hacker way to access your computer.

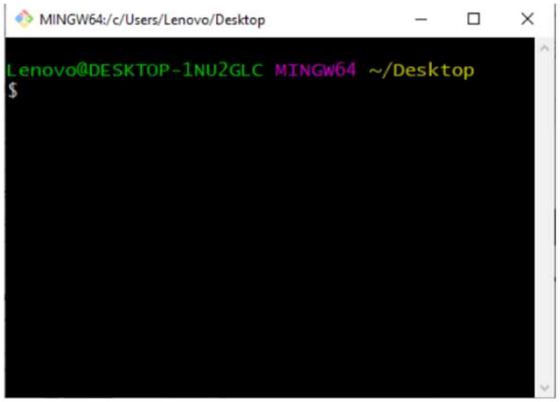
WHY?

- Really useful when using GitHub and Git.
- Much faster than drop and dragging once you get used to it.
- Almost every workplace with programmers and software developers uses GitHub.

Types:

- Shells:
  - Bash.
  - Cmd.
  - PowerShell.

=

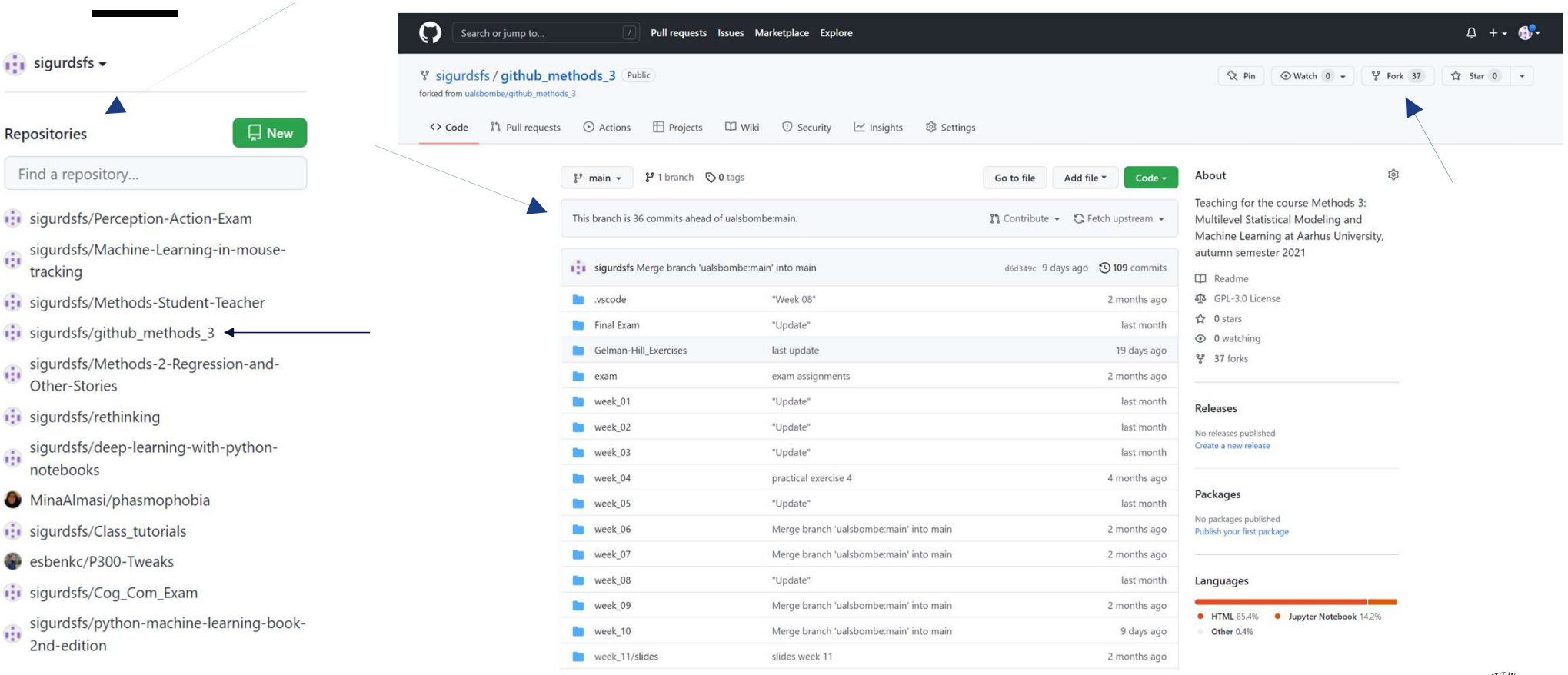


GITHUB

GITHUB

- Online host for Git
- Enables:
  - Collaboration across computers
  - Script/project sharing
- Used in increasingly many workplaces

OVERVIEW OF GITHUB:



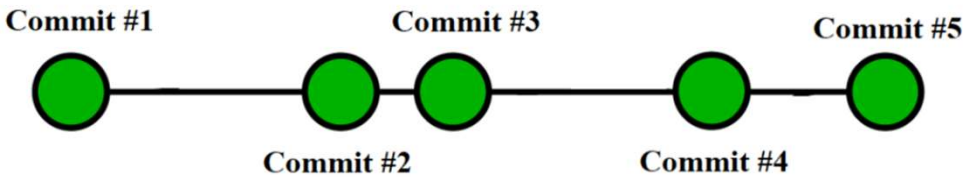
GITHUB & GIT

GIT

- Version control system
- Allows for going back in time

GITHUB AND GIT

GIT



- Version control system
- Allows for going back in time
- Commit -> creating checkpoint

WHEN WE COMBINE THEM ALL!

- Tool that keep track of previous work.
- Allows for cooperation on scripts at the same.
- And also, a nice “online backup”.

Our plan

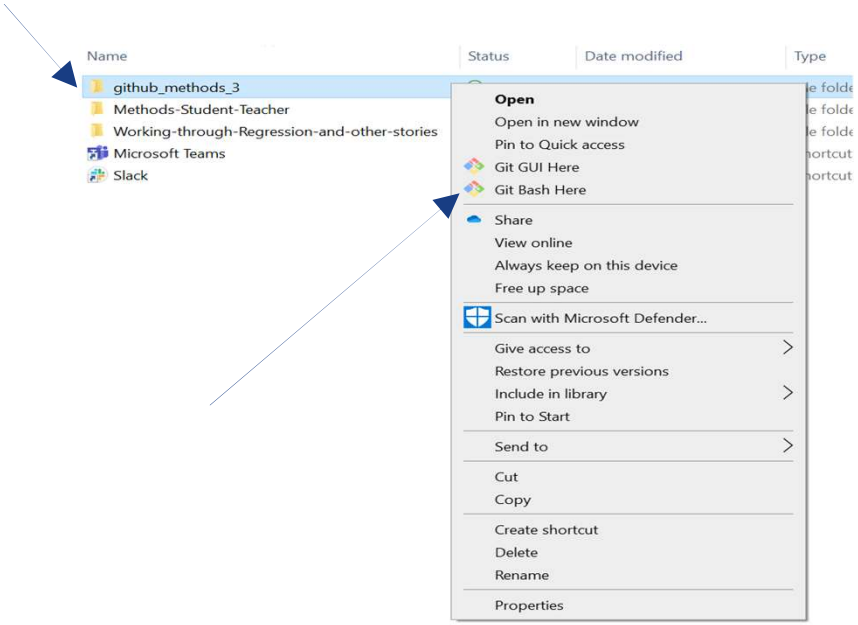
- 1) Methods 2 repository in GitHub, 2) We use Git to version control and upload to GitHub, 3) We use the terminal to command Git.

- It will take some time to get comfortable and we will revisit this again.
- Let’s try and do it together.

STEP 1 (NAVIGATE)

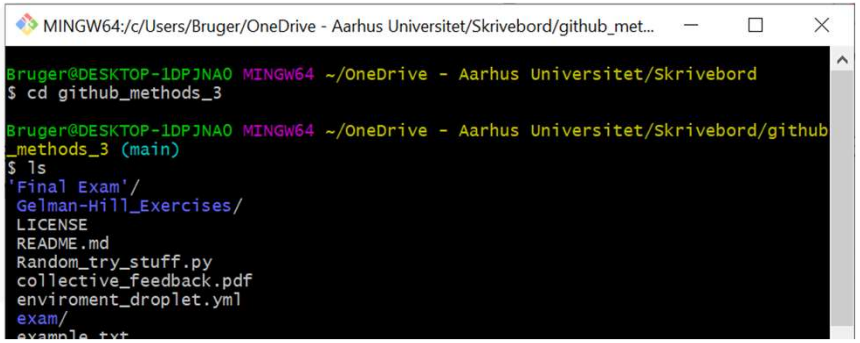
- **Create/Fork a GitHub repository.**

- We must open our terminal and navigate to the right folder. Can be done 2 ways.
    1. Using the right click method.
    2. Using the terminal.



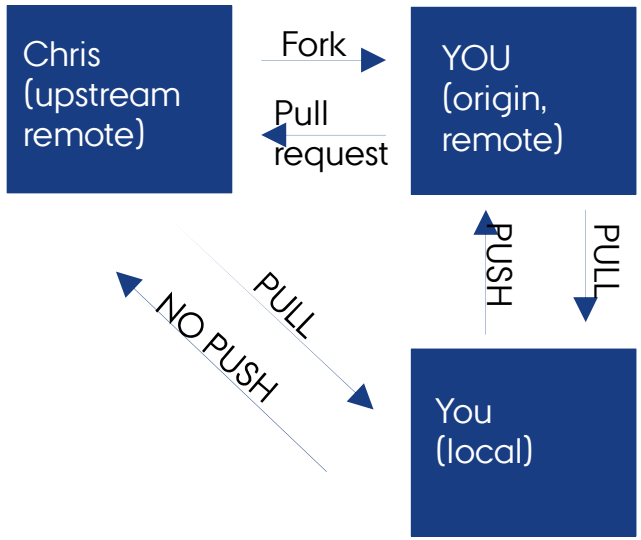
How to use the terminal: (commands)

```
cd <a_folder_name> # change directory to a given folder
mkdir <a_folder_name> # make a directory
pwd # print working directory
ls # list files in directory
cd # go to your home directory
cd .. # go one directory up
cd -- # go back to the directory you were in before
```



STEP 2 (CLONING TIME)

1. Terminal is in the desired location, and we want a local version of our GitHub.
2.
```
## go to the path where you want to clone YOUR fork
git clone <url_to_your_repository>
```



GITHUB COMMANDS

- Forking
- Cloning repo
- Navigate remote(s)
- Push to GitHub
- Pull from GitHub

STEP 3 (VERSION CONTROL)

Now that we have a both a forked version of Chris repository and a local on our own computer it is time to manipulate. We either create a Rmd new folder or anything and want to stash it

Commands:

Pushing your solution to your GitHub repository

Then do the following to push changes to the origin (your fork of my repository)

```
cd <your_local_directory> # the folder you cloned your repository into
git status ## check which files differ
git add <name_of_your_solution_file.Rmd> # add files to what you will later commit
git commit -m "<your message>" # commit all the files you have added above
git push origin main ## push the commited changes to your origin (your remote location)
```

Adding upstream (my repository) to pull changes from there

You can do this to more easily keep up to date - we'll also go through on Sep 22
See also Figure 1

```
## add another remote spot (my repository)
git remote add upstream https://github.com/ualsbombe/github_methods_3.git
# see which remote channels you have (you should have origin (your repository) and ...
## upstream (my repository)
git remote -v
git pull upstream main ## call this from your local copy to pull my latest changes ...
git push origin main ## ... you can then push these changes to your origin,
## so they match with the upstream
```

A schematic of the workflow

Other useful commands

```
git pull origin main ## pull what is on origin (your remote) into your local folder
git stash ## temporarily shelves away changes you made (useful when you don't want to commit
```

PHEW.....

- A lot of information, you will have A LOT of bugs the first weeks...
- Don't despair it will work at some point. <3

- Now let's do some exercises to get it under our skin.
  - Find the "**pushing_your_solutions.pdf"** for an overview of commands
  - 1) Fork https://github.com/chmathys/methods-2-course.
  - 2) Clone your forked version.
  - 3) Create a new folder with a .Rmd file in it.
  - 4) Check the status of your local folder.
  - 4) Add, Commit and Push the changes in your local folder to the online repository.

BREAK (MAYBE)

RECAP OF LAST SEMESTER:

- Did you learn anything last semester?
- What did you learn last semester?

My Perspective:

1. Introduction to the general and generalized linear model.

   Lm() & Glm()
2. Ways to evaluate our models; Residuals, RSS, Information Criteria & Anova.
3. Multiple regression & interaction effects.
4. We learned about mixed effect models. Lmer(), glmer()
5. Overfitting and ways to counteract it.
6. Train/test split of our data. Prediction on unseen data is the best evaluation of a model.
7. Inter annotator agreement score.

FEEDBACK


what did you...
like most?
want most?
dislike most?
fear most?
...?
and why...
54
Wednesday, March 25, 2009

COUPLE OF NEW FUNCTION().

When we want to simulate gaussian data or distribtuions:

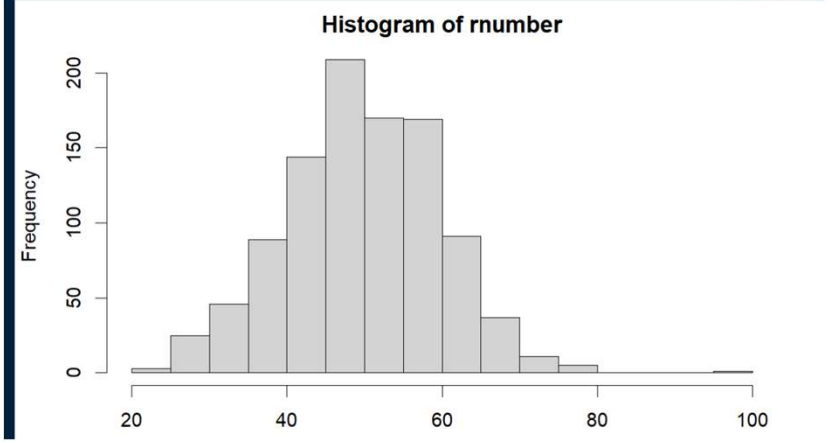
```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

We need rnorm() for today.

When would we want to simulate something that is Gaussain?

```
rnumber =rnorm(1000, mean= 50, sd = 10)

hist(rnumber)
```



EXERCISES FOR TODAY.

1. Do exercise 1.2 in the book
2. Applying learned skills on your selected dataset: (IDEA)
    1. Use a descriptive statistics and visual investigation to give a preliminary look into the data. (Use some of the things from chapter 2)
    2. Create several models and discuss what each model reliably can infer anything about.
        - Mixed effect, which random effects? Logistic regression? Interaction effects? Should we dummy code some variables?
    3. Do model comparison.
    4. Test the model's prediction capability.
    5. Write up a short summary of your model.
3. Do Exercise 2.3 in the book **(Hard)** hints. (Summarise_at() , str_sub(), pivot_longer())
4. Upload the .Rmd to your GitHub.

THANKS FOR TODAY.

- Next week we will follow the books exercises…

- Check out Chris' GitHub for a link to a cryptpad.
  - It is a text editor like google docs just encrypted. You should write the links to your GitHub.
  - You can also ask question in the cryptpad if there is something you don't understand.


AARHUS
UNIVERSITY