

GITHUB 101 FEAT. TERMINAL

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 $f'(x) = \text{Constant}$ and $f''(x) = 0$.
 - $f(x) = B_0 + B_1x_1 + B_2x_2^2 + \text{error}$
 - $f'(x) = B_1 + B_2 * 2x_2 + \text{error} ; f''(x) = B_2 * 2$
 - $f(B) = B_0 + B_1x_1 + B_2 + x_2^2 + \text{error}$
 - $f'(B) = 1 + x_1 + x_2^2 + \text{error} ; f''(B) = 0$

GITHUB, TERMINAL AND GIT?!?!

Terminal:

WHAT IS IT!?

- An old school hacker way to access your computer.

WHY?

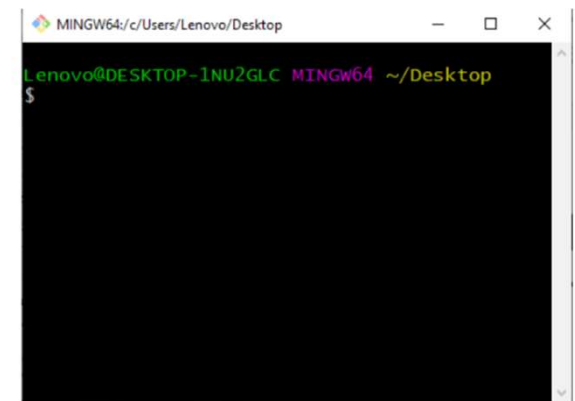
- Really useful when using GitHub and Git.
- Much faster than drop and dragging once you get used to it.
- Almost every workplace with programmers and software developers uses GitHub.

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:

Repositories

Find a repository...

- sigurdsgithub_methods_3
- sigurdsgithub_methods_2-Regression-and-Other-Stories
- sigurdsgithub_methods_1
- sigurdsgithub_methods_0
- sigurdsgithub_methods_4
- sigurdsgithub_methods_5
- sigurdsgithub_methods_6
- sigurdsgithub_methods_7
- sigurdsgithub_methods_8
- sigurdsgithub_methods_9
- sigurdsgithub_methods_10
- sigurdsgithub_methods_11
- sigurdsgithub_methods_12
- sigurdsgithub_methods_13
- sigurdsgithub_methods_14
- sigurdsgithub_methods_15
- sigurdsgithub_methods_16
- sigurdsgithub_methods_17
- sigurdsgithub_methods_18
- sigurdsgithub_methods_19
- sigurdsgithub_methods_20
- sigurdsgithub_methods_21
- sigurdsgithub_methods_22
- sigurdsgithub_methods_23
- sigurdsgithub_methods_24
- sigurdsgithub_methods_25
- sigurdsgithub_methods_26
- sigurdsgithub_methods_27
- sigurdsgithub_methods_28
- sigurdsgithub_methods_29
- sigurdsgithub_methods_30
- sigurdsgithub_methods_31
- sigurdsgithub_methods_32
- sigurdsgithub_methods_33
- sigurdsgithub_methods_34
- sigurdsgithub_methods_35
- sigurdsgithub_methods_36
- sigurdsgithub_methods_37
- sigurdsgithub_methods_38
- sigurdsgithub_methods_39
- sigurdsgithub_methods_40
- sigurdsgithub_methods_41
- sigurdsgithub_methods_42
- sigurdsgithub_methods_43
- sigurdsgithub_methods_44
- sigurdsgithub_methods_45
- sigurdsgithub_methods_46
- sigurdsgithub_methods_47
- sigurdsgithub_methods_48
- sigurdsgithub_methods_49
- sigurdsgithub_methods_50
- sigurdsgithub_methods_51
- sigurdsgithub_methods_52
- sigurdsgithub_methods_53
- sigurdsgithub_methods_54
- sigurdsgithub_methods_55
- sigurdsgithub_methods_56
- sigurdsgithub_methods_57
- sigurdsgithub_methods_58
- sigurdsgithub_methods_59
- sigurdsgithub_methods_60
- sigurdsgithub_methods_61
- sigurdsgithub_methods_62
- sigurdsgithub_methods_63
- sigurdsgithub_methods_64
- sigurdsgithub_methods_65
- sigurdsgithub_methods_66
- sigurdsgithub_methods_67
- sigurdsgithub_methods_68
- sigurdsgithub_methods_69
- sigurdsgithub_methods_70
- sigurdsgithub_methods_71
- sigurdsgithub_methods_72
- sigurdsgithub_methods_73
- sigurdsgithub_methods_74
- sigurdsgithub_methods_75
- sigurdsgithub_methods_76
- sigurdsgithub_methods_77
- sigurdsgithub_methods_78
- sigurdsgithub_methods_79
- sigurdsgithub_methods_80
- sigurdsgithub_methods_81
- sigurdsgithub_methods_82
- sigurdsgithub_methods_83
- sigurdsgithub_methods_84
- sigurdsgithub_methods_85
- sigurdsgithub_methods_86
- sigurdsgithub_methods_87
- sigurdsgithub_methods_88
- sigurdsgithub_methods_89
- sigurdsgithub_methods_90
- sigurdsgithub_methods_91
- sigurdsgithub_methods_92
- sigurdsgithub_methods_93
- sigurdsgithub_methods_94
- sigurdsgithub_methods_95
- sigurdsgithub_methods_96
- sigurdsgithub_methods_97
- sigurdsgithub_methods_98
- sigurdsgithub_methods_99

sigurdsgithub_methods_3

This branch is 36 commits ahead of ualsbombe:main.

File	Commit Message	Commit Date
.vscode	"Week 08"	2 months ago
Final Exam	"Update"	last month
Gelman-Hill_Exercises	last update	19 days ago
exam	exam assignments	2 months ago
week_01	"Update"	last month
week_02	"Update"	last month
week_03	"Update"	last month
week_04	practical exercise 4	4 months ago
week_05	"Update"	last month
week_06	Merge branch 'ualsbombe:main' into main	2 months ago
week_07	Merge branch 'ualsbombe:main' into main	2 months ago
week_08	"Update"	last month
week_09	Merge branch 'ualsbombe:main' into main	2 months ago
week_10	Merge branch 'ualsbombe:main' into main	9 days ago
week_11/slides	slides week 11	2 months ago

About

Teaching for the course Methods 3: Multilevel Statistical Modeling and Machine Learning at Aarhus University, autumn semester 2021

Readme

GPL-3.0 License

0 stars

0 watching

37 forks

Releases

No releases published

Create a new release

Packages

No packages published

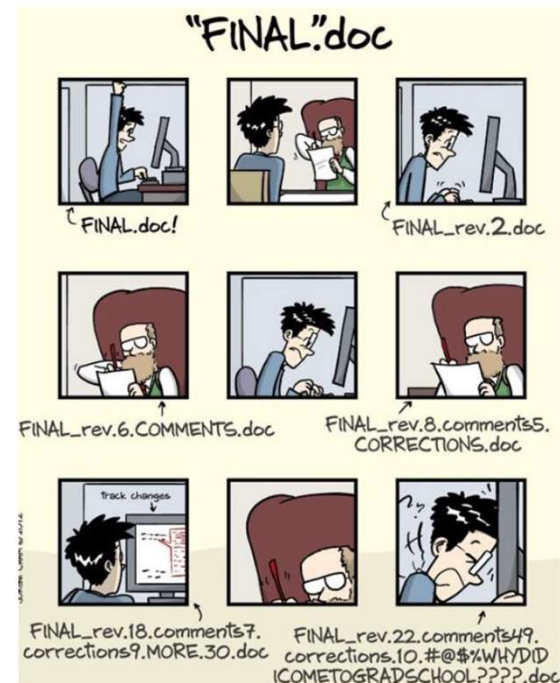
Publish your first package

Languages

HTML 85.4% Jupyter Notebook 14.2% Other 0.4%

████████████████████

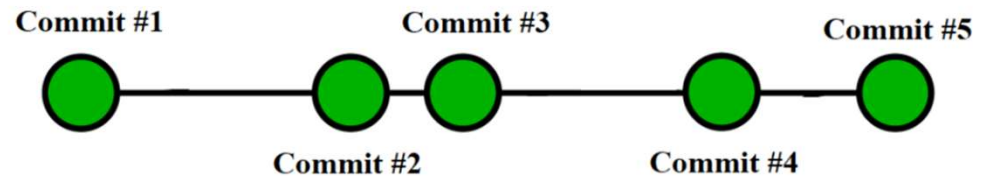
- Version control system
- Allows for going back in time



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.

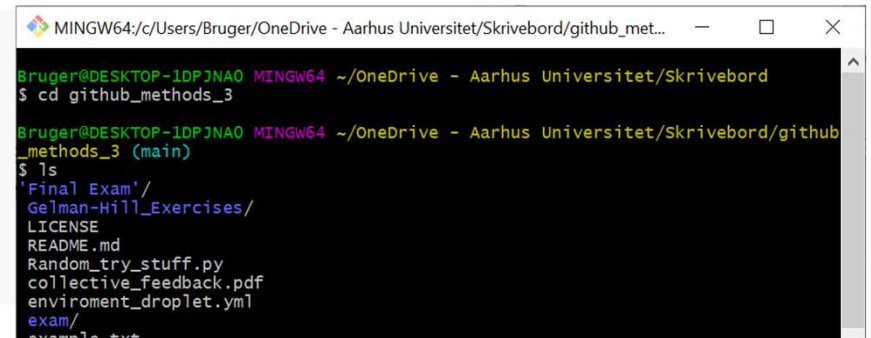
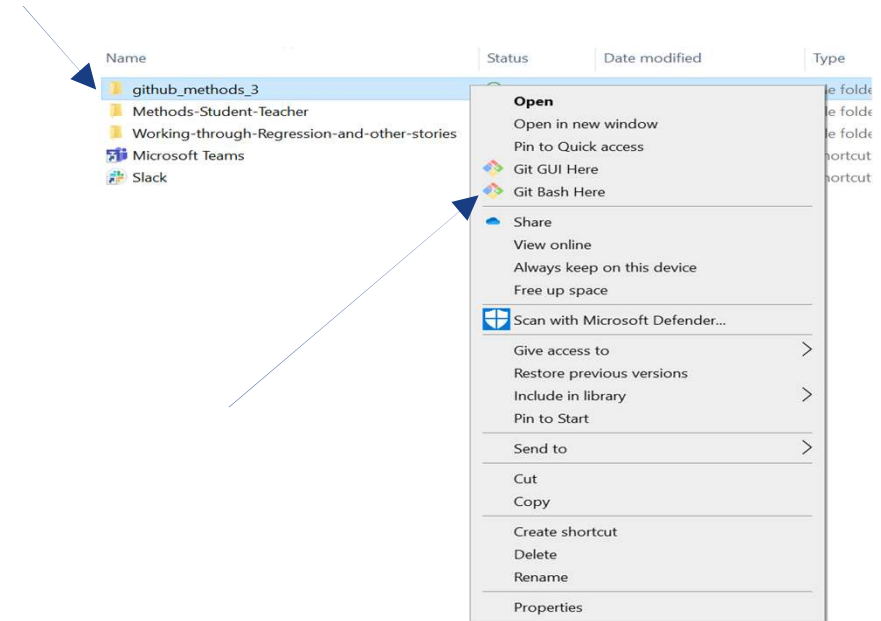


STEP 1 (NAVIGATE)

- 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.

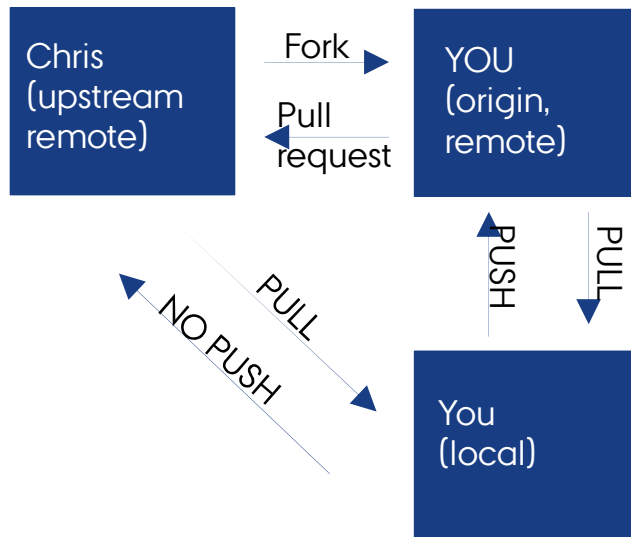
How to use the terminal: (commands)

```
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 committed 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?
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.

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()

FEEDBACK

what did you...

- like most?
- want most?
- dislike most?
- fear most?
- ...?

and why...

54

Wednesday, March 25, 2009

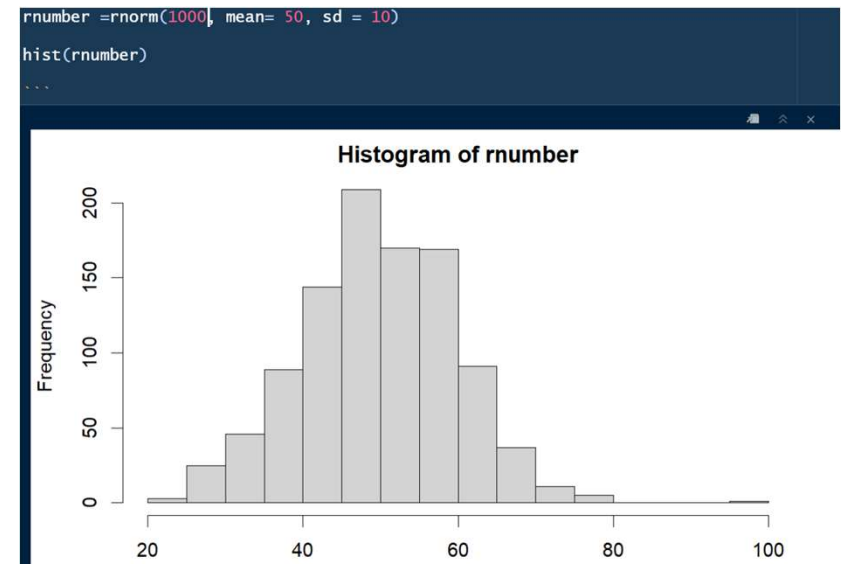
COUPLE OF NEW FUNCTION().

When we want to simulate gaussian data or distributions:

```
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 Gaussian?



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