First Year Project

IT University of Copenhagen – Spring 2022

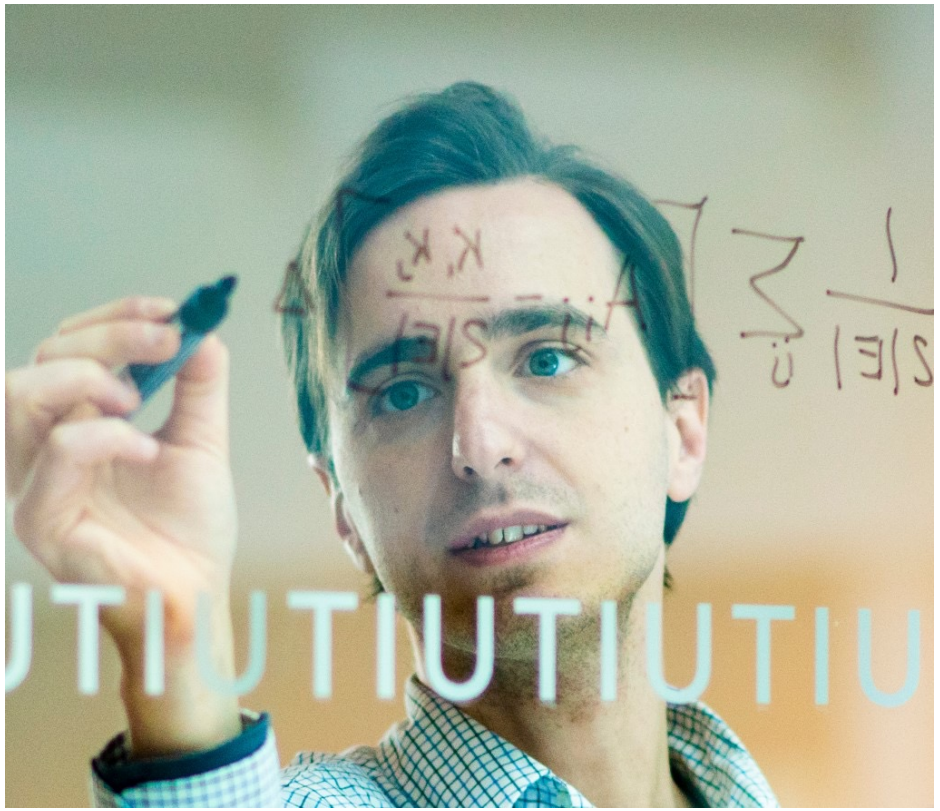
IT UNIVERSITY OF CPH

Who are we

Veronika Cheplygina

- Associate professor at ITU since 2021
- Computer science -> Machine learning -> medical imaging (The Netherlands)

Who are we

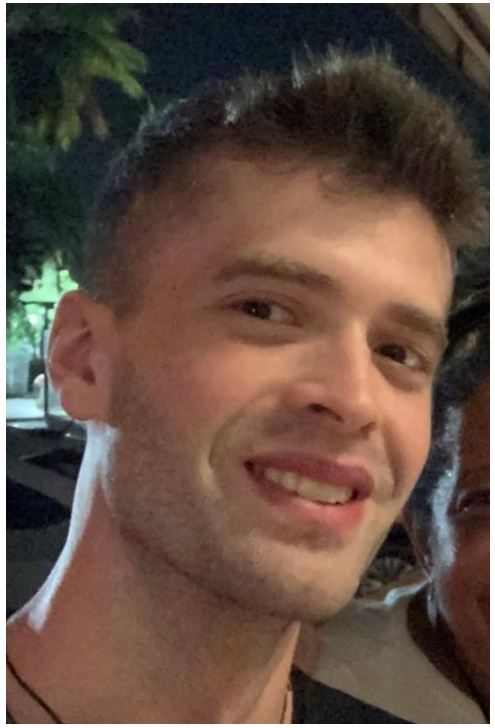


Michele Coscia

Christian Hardmeier

Teaching assistants

Noah



Johan

Gino

Danielle

You!

https://www.menti.com/q5z1i2ecqv

Program for today

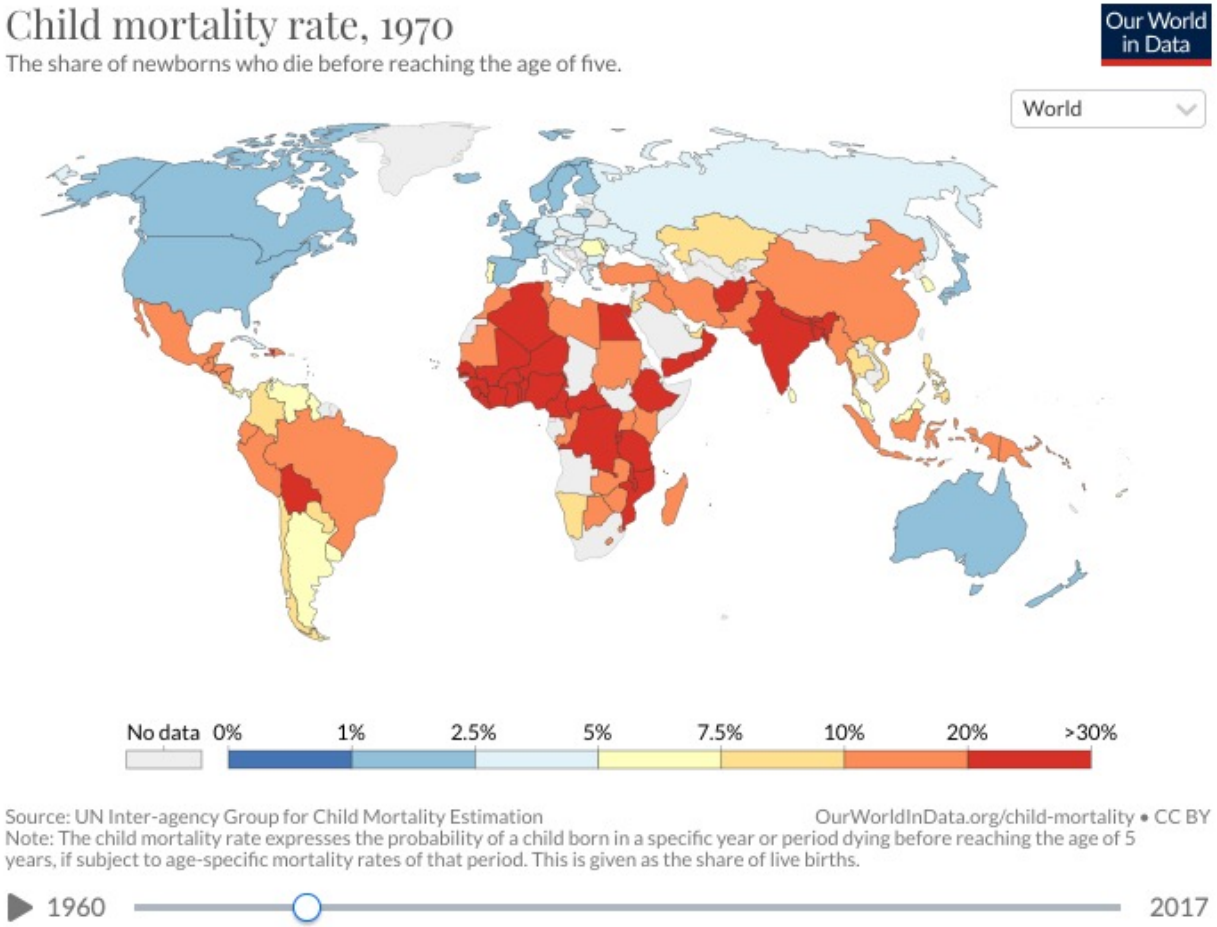
- Why this course
- Course overview & logistics
- Project groups
  - ~~Group exercise from SAP~~ → postponed due to Covid
- Git(hub)
- Thursday: Start of project 1

Why this course

What can we do with data

- OurWorldInData.org - “Research and data to make progress against the world’s largest problems”

https://ourworldindata.org/child-mortality



What can we do with data

- Detect tuberculosis in low-resource areas [Image NY Times]

What can we do with data

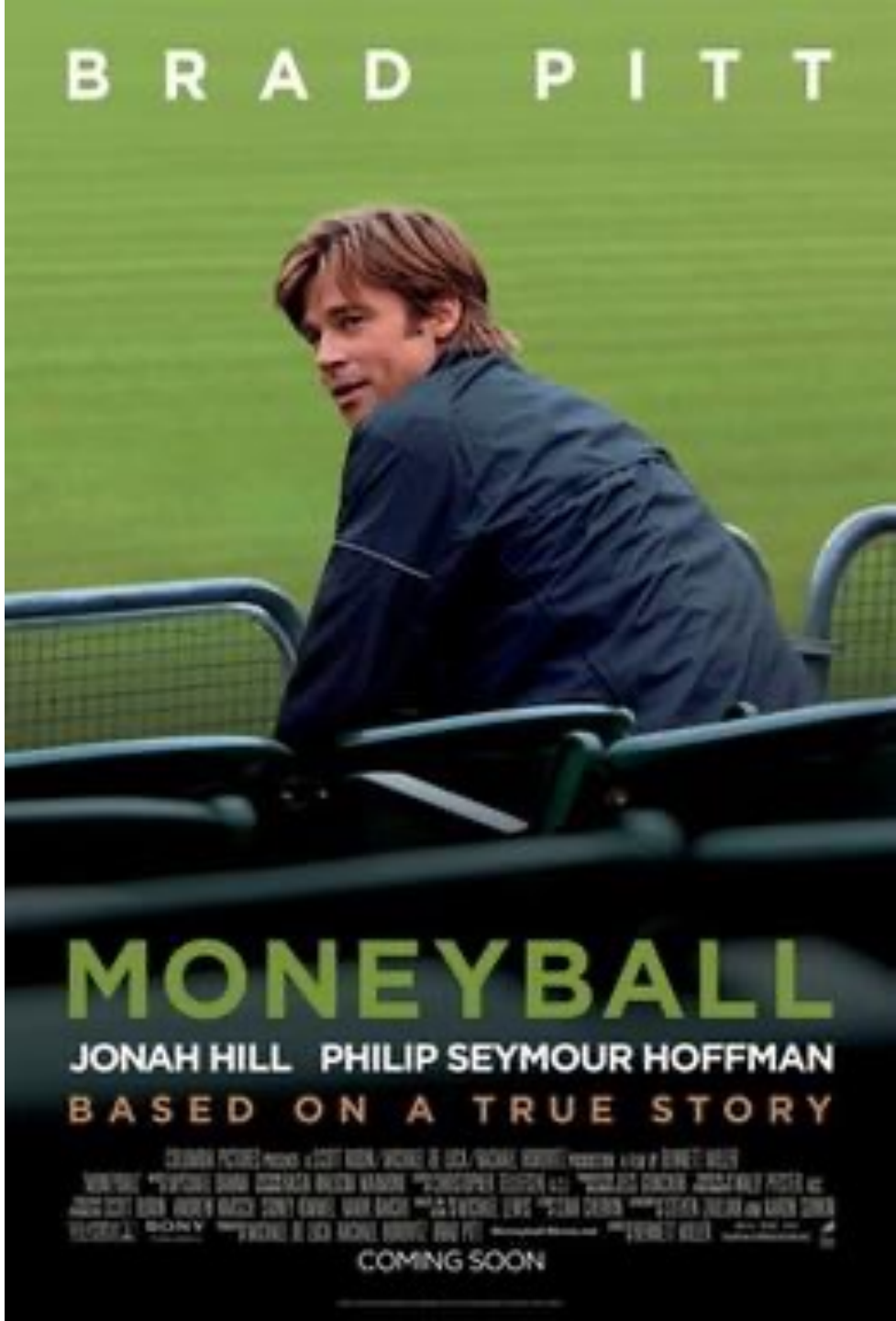
- Correlations between things you buy now & later
- Predict customer needs, send offers/coupons
- *“My daughter got this in the mail! She’s still in high school, and you’re sending her coupons for baby clothes and cribs? Are you trying to encourage her to get pregnant?”*

Article

What can we do with data

- Hiring baseball players

https://en.wikipedia.org/wiki/Moneyball_(film)



“Data is the new oil”

- Quote from The Economist, May 6th 2017
- Most valuable companies: Alphabet, Amazon, Apple Facebook, Microsoft

Different data, different approaches

- Common element: extract information learn from data, to explain / predict
- But data comes in different shapes and sizes…
- This course:
  - tabular data
  - images
  - text

Why this course

- You need hands-on experience (lectures and tutorials are not enough)
- You need to work on *real-world* problems with *real-world* data
- You need to experience the whole data science pipeline:
  - Problem formulation
  - Data analysis
  - Communication of results
- While working with others

After this course you will be able to

- Identify and delimit a problem in Data Science within a given domain-specific context
- Discuss the relevant options for an appropriate scientific methodology to address the problem
- Carry out the full analysis according to the selected methodology
- Communicate their work to both experts and non-experts

- Using Git(hub), in a group

Overview

Three projects

- Project 1 - COVID19 & weather data (tabular data) - Michele
- Project 2 - Skin lesion diagnosis (images) - Veronika
- Project 3 - Sentiment classification (text) - Christian

Schedule

Calendar week	Dates	Topic	Who
5	1 February	Introduction	Veronika
5, 6, 7, 8, 9	3 February - 3 March (deadline 4 March)	Project 1	Michele
10	8 March, 10 March	Project 1 exams	Michele & Veronika
11, 12, 13, 14	15 March - 7 April (deadline 8 April)	Project 2	Veronika
15	12 April, 14 April	Easter, no lectures	
16, 17, 18, 19 22	19 April - 12 May (deadline 3 June)	Project 3	Christian
24	16 June, 17 June	Project 2 & 3 exams	Christian & Veronika

Lectures

- First lecture (today) at ITU, other lectures for weeks 5 to 9 will be online via Zoom
- You can use the reserved classroom to watch the lecture
- We will decide on the lectures for project 2 & 3 in a few weeks

Exercises

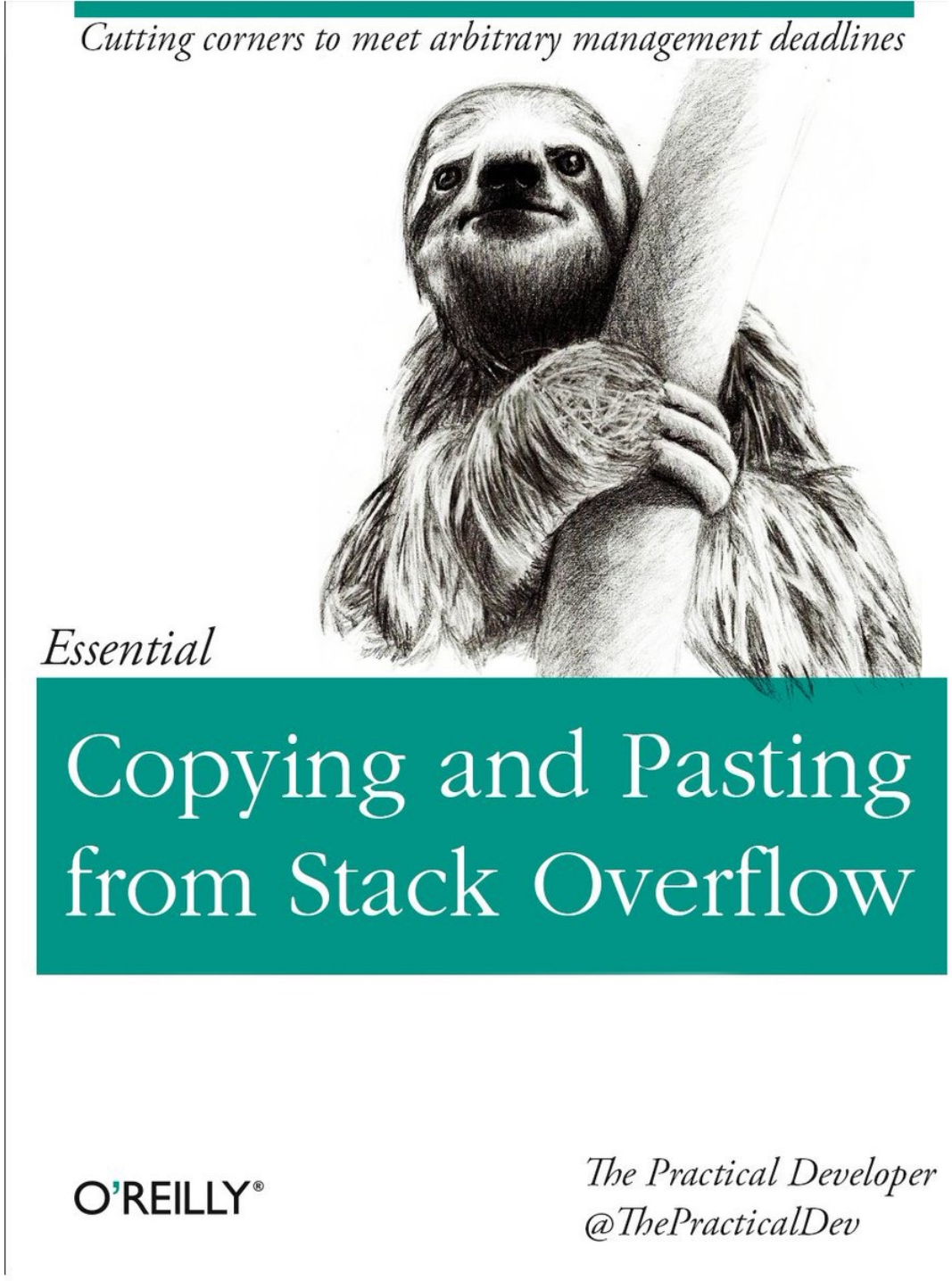
- Exercise sessions are at ITU, with online access via Discord
- In some cases, the TAs will show you additional material, for example on using Github
- But in most cases, the TAs are there to help you with your questions about the exercises and project - **you** need to ask for help

Extra help

- Help us find out what is unclear and needs more explanation
- We have some flexibility to add material, for example math concepts from first semester, Overleaf, etc.
- StudyLab - Schedule 3A50
- 2nd semester https://learnit.itu.dk/course/view.php?id=3021539 (TBA)

Course materials

- We will share relevant papers, tutorials etc on LearnIT
- You are welcome to use other resources/code but you must reference these in your code and report



Project exams

- Project in groups of 4 or 5
- Deadline = submit your report on LearnIT
- Exam = short presentation + questions (B1I form)

- Project 1 exam: we will give you feedback and suggestions on how to improve for the next exam.

- Project 2 & 3 exam: your group will present on either project 2 OR 3, but questions will be about both projects.

Contact

Any course-related questions that other students should also see: **LearnIT forum**

Connecting with TAs and other group members: **Discord**

Other issues: course manager Veronika (vech@itu.dk) or SAP (sap@itu.dk)

Project groups

Project groups

- We created random groups for you (4 or 5 people) - real-life situation

- For project 2 & 3, groups will be different (more information TBA)

Project groups

- Find your group members today, or try to contact them today or tomorrow

- If some students are not joining the course, the groups might change a bit, so that there are at least 4 people in each

Project groups

- If by Thursday 10:00 you cannot reach some of your group members, please send an email:
  - To: vech@itu.dk
  - CC: all your group members (including the missing ones)
  - Subject: FYP group X
  - Content: who you were not able to contact

- I will adjust the groups after the lecture on Thursday

Course expectations

https://www.menti.com/iph5zijtjb

Advice from past years

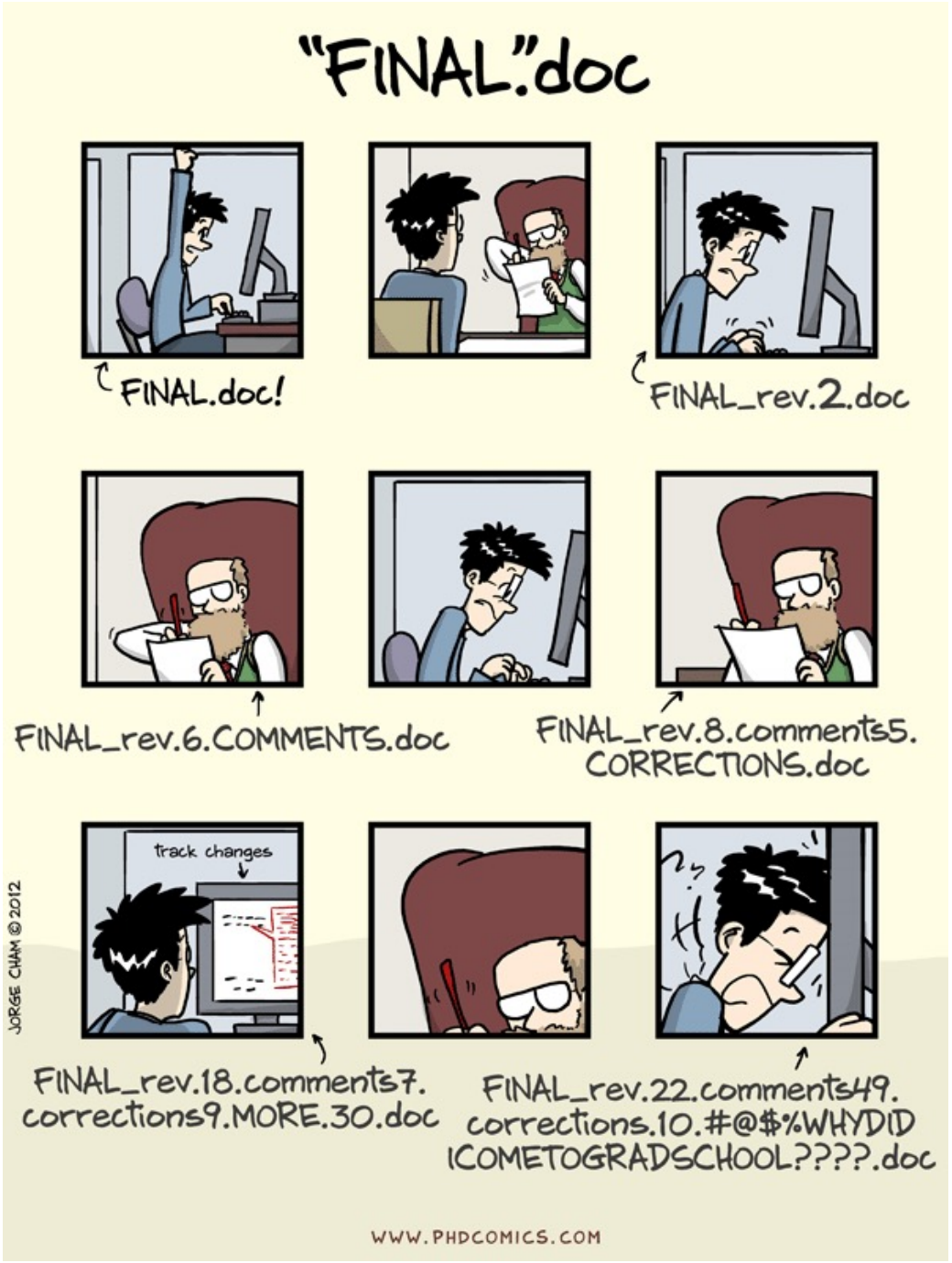
- “Don’t spend too long on literature in the beginning”
- “Make a plan, especially for time away from campus”
- “Start coding early”
- “Don’t be afraid to try things out”
- “You can do it even if nobody has programmed before

Git/Github

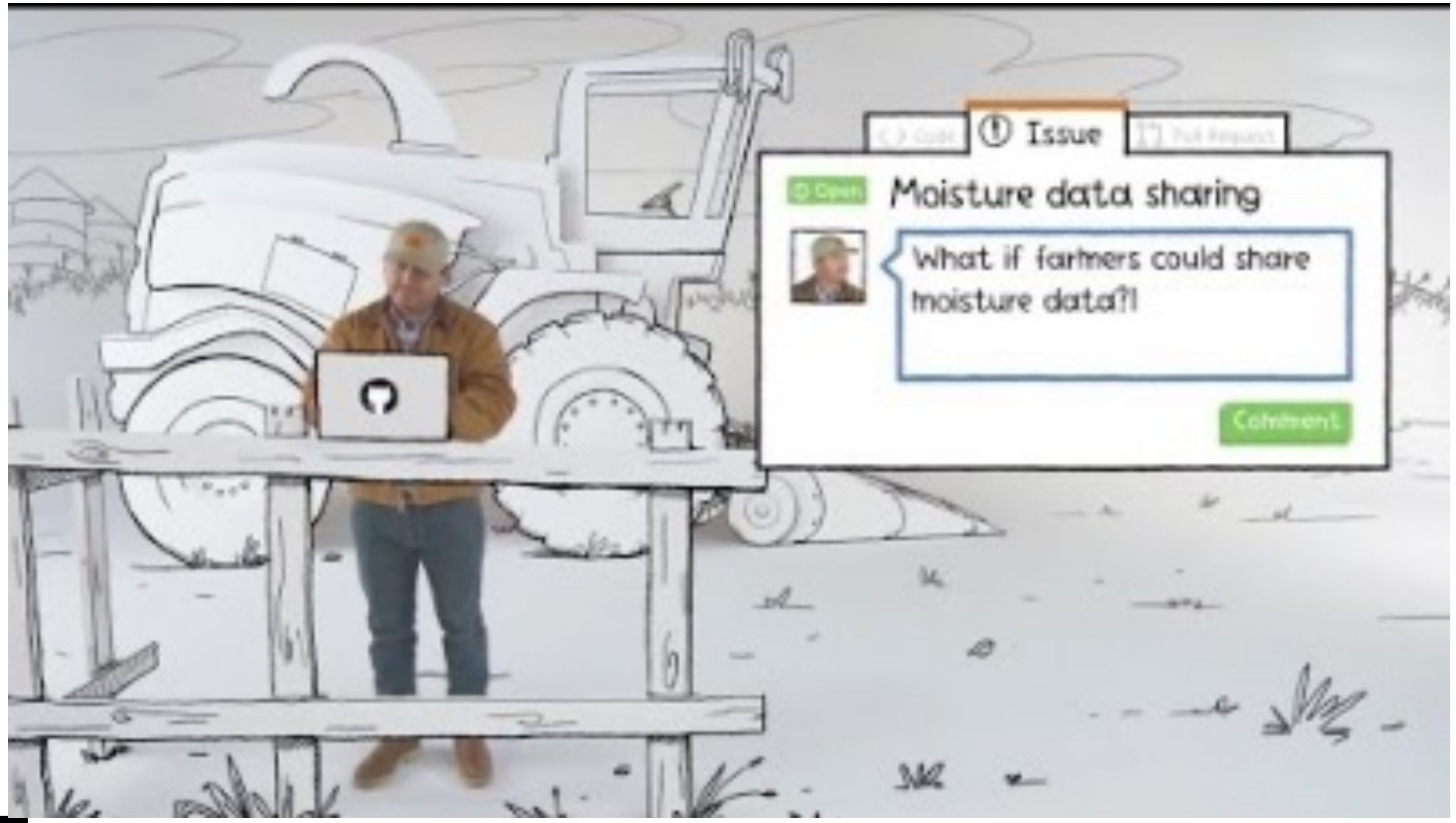
Git & Github

- Git is a distributed version-control system for file changes, originally designed to coordinate group work.

- Github is a service for software development and version control, that uses Git



Git & Github


Issue
Open
Moisture data sharing
What if farmers could share moisture data?!
Comment

Github examples

- Software

https://github.com/scikit-learn/scikit-learn

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

- Online book

https://github.com/alan-turing-institute/the-turing-way

README.md

The Turing Way

> *This README.md file in also available in Dutch (README-Dutch), French (README-French.md), German (README-German.md), Indonesian (README-Indonesian), Italian (README-Italian), Korean (README-Korean), Portuguese (README-Portuguese), and Spanish (README-Spanish) (listed alphabetically).*

Total Contributors:

all contributors 311

Github examples

- DASYA website (private, URL will not work for others: https://github.com/ITU-DASYALab/website) - 13 contributors

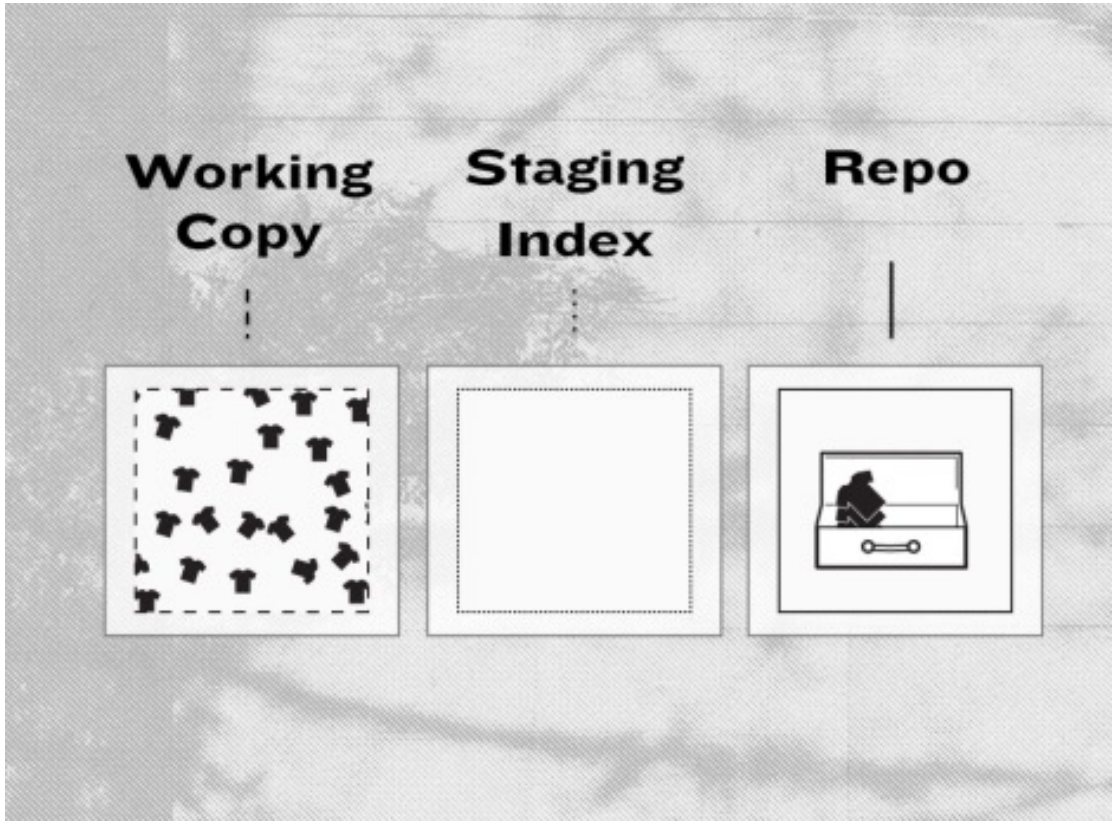
- Student project of 1 person:
- https://github.com/tueimage/transfer-medical-msc-2019

Github

- Lots of functionality
- Most important to get started:
  - pull
  - add/commit/push
- Branching, conflicts, … → later

Github

- Suitcase analogy:
  - Put clothes on the bed (=working copy)
  - Select some clothes you want to bring and fold them (=staging index)
    - git add
  - Put them into the suitcase (=repository)
    - git commit
  - Bring suitcase with you
    - git push
  - https://www.slideshare.net/AnnetteLiskey/git-in-the-van-highedweb-2013



Git/Github

Most important things to get started:

- https://lgatto.github.io/github-intro/

Github

More in-depth, why does it work this way?

- https://missing.csail.mit.edu/2020/version-control/

Assignment for today's exercises

Create a practice repository for your group, invite group members, and create one file (for example, pancakes.txt)

Make sure that every group member can access the repository outside the browser

- Command line interface + text editor
- Editors like PyCharm
- …

Assignment

Outside of the browser, each group member then should, in sequence:

get the latest version of the repository
modify an existing file
create a new file
push the changes

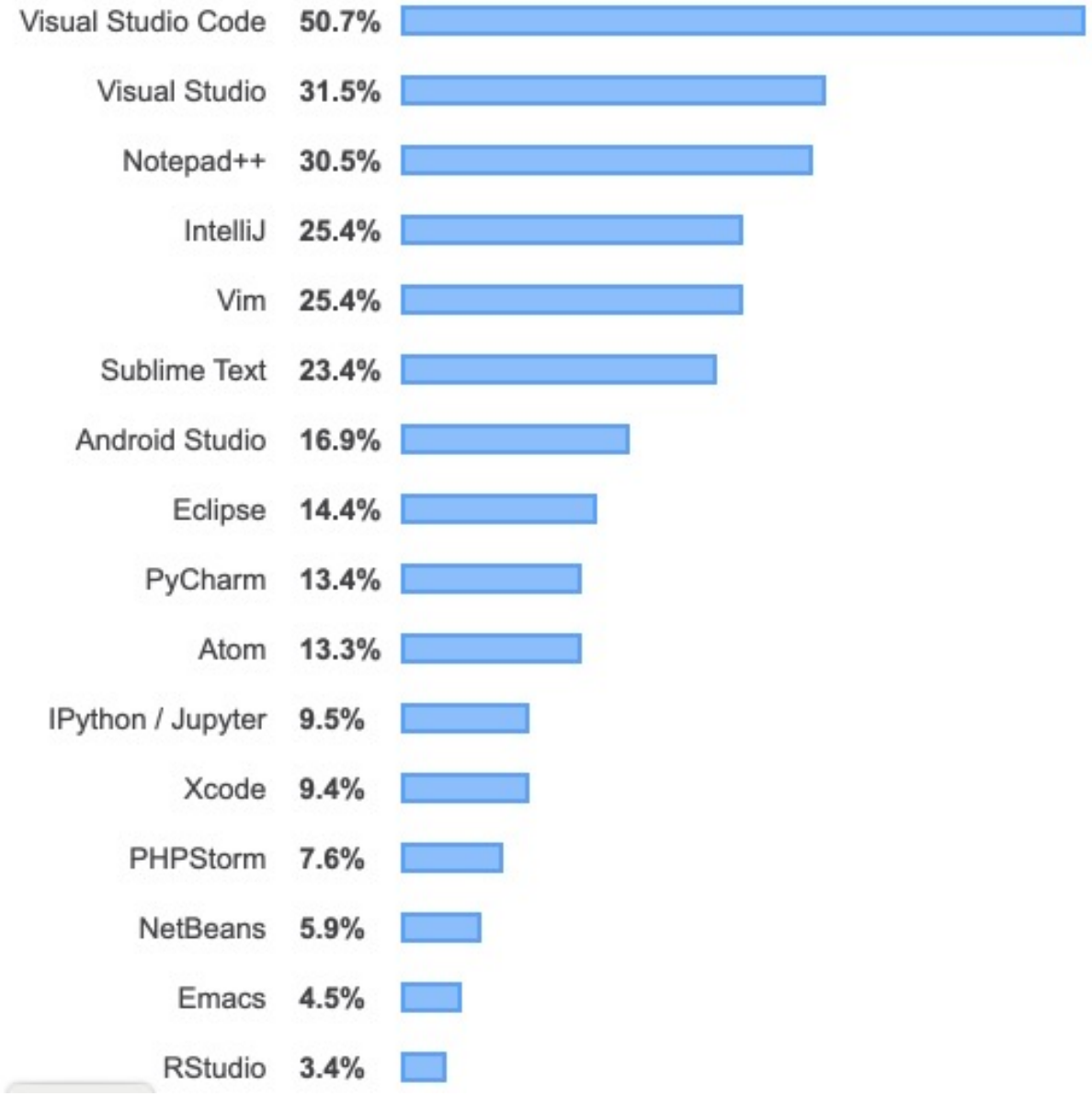
Once done submit gitlog.txt: your repo’s git log, e.g. by running: git log > gitlog.txt to LearnIT

Github

- Command line interface
- Graphical user interface (desktop.github.com)

Text editor

Source
https://insights.stackoverflow.com/survey/2019/#development-environments-and-tools



Working together - organized repository

- https://github.com/drivendata/cookiecutter-data-science

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
```

```
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.readthedocs.io
```

Working together - organized repository

- For your projects, particularly important:

- README.MD
- data (*)
- reports
- reports/figures
- requirements.txt
- src

Larger data storage

- Github is not usually suitable for large data
- There are hard limitations: 100 MB files, 2 GB push, ~5 GB repo
- If you have hundreds of MB of data, add the data folder to .gitignore

- https://stackoverflow.com/questions/38768454/repository-size-limits-for-github-com

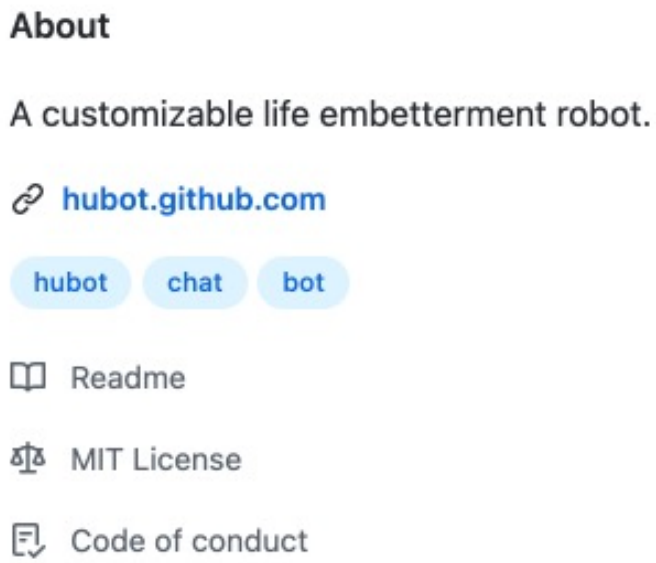
Larger data storage

- Various platforms with persistent storage have agreements with e.g. publishers, to guarantee data preservation
  - Open Science Framework https://osf.io/
  - FigShare https://figshare.com/
  - Zenodo

Github is slow for large files, solve with pointers to files hosted elsewhere https://git-lfs.github.com/

Licensing

- Various type of licences for data and software
  - https://the-turing-way.netlify.app/reproducible-research/licensing.html for an overview
  - https://choosealicense.com/ for a “flowchart”
  - https://tldrlegal.com/
- For data, CreativeCommons (CC) is often used:
  - BY - creator must be credited
  - SA - ShareAlike, derivatives/redistributions must have identical license
  - NC - only non-commercial
  - ND - no derivatives
- On Github you can specify this in LICENSE.md or LICENSE.txt



Working together - files

report.docx
Vs
20211108-DataScience-group01-report.pdf

Aim: readable for humans and machines, intuitive ordering

More examples in:
https://speakerdeck.com/jennybc/how-to-name-files

NO

myabstract.docx
Joe's Filenames Use Spaces and Punctuation.xlsx
figure 1.png
fig 2.png
JW7d^(2sl@deletethisandyourcareerisoverWx2*.txt

YES

2014-06-08_abstract-for-sla.docx
joes-filenames-are-getting-better.xlsx
fig01_scatterplot-talk-length-vs-interest.png
fig02_histogram-talk-attendance.png
1986-01-28_raw-data-from-challenger-o-rings.txt

Working together - files

Data organization in spreadsheets - also tips on consistent names, etc.

Working together - code

What NOT to do (funny): https://github.com/Droogans/unmaintainable-code

Be Abstract

In naming functions and variables, make heavy use of abstract words like *it*, *everything*, *data*, *handle*, *stuff*, *do*, *routine*, *perform* and the digits e.g. `routineX48`, `PerformDataFunction`, `DoIt`, `HandleStuff` and `do_args_method`.

Misleading names

Make sure that every method does a little bit more (or less) than its name suggests. As a simple example, a method named `isValid(x)` should as a side effect convert `x` to binary and store the result in a database.

Working together - code

Don’t put too much into one file

clean_data.py
train_model.py
plot_figures.py

Consistent & descriptive names

Useful comments

Look at repositories that you use!

90% of all code comments:

Good practices for code

Keep parameters & logic separated (e.g. thresholds)

define parameter file which can be linked to experiments

Keep track of the random seed

Reproducibility

Your results need to be **reproducible**

Tables and figures in your report can be generated again with a few clicks

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions

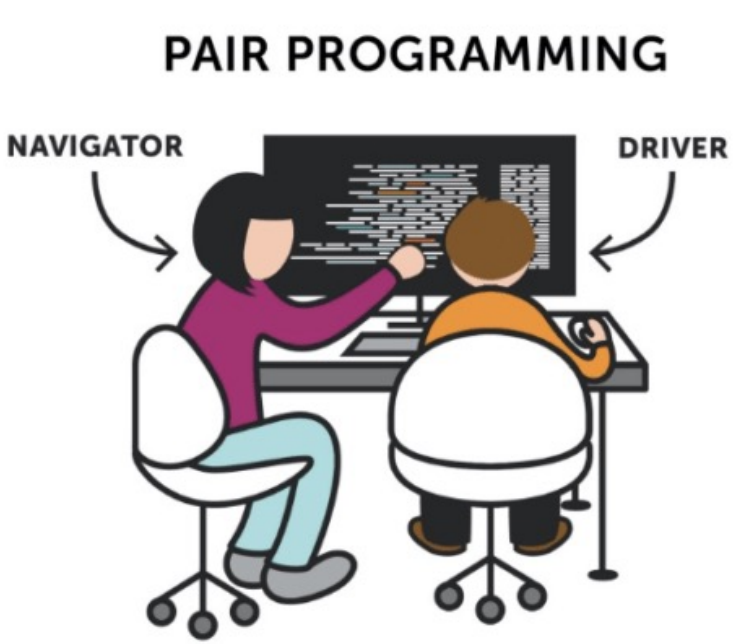
Pair programming

2 people, 1 computer

Consider both “main direction”
& specifics of programming language

Write comments as you go

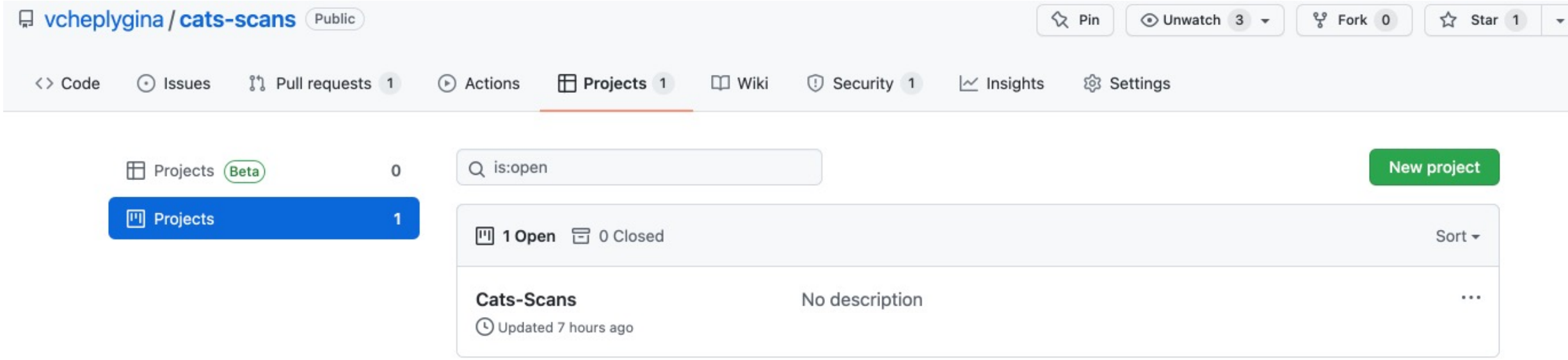
Reduces mistakes



https://unruly.co/blog/article/2019/08/27/what-is-pair-programming

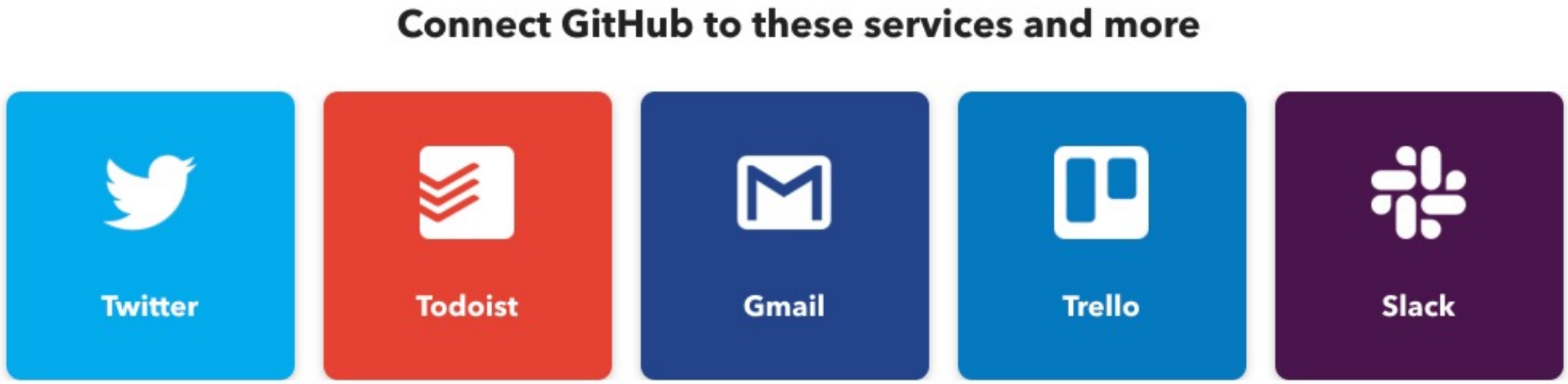
Github project management

- Github project - Trello-like system with cards
- Example: https://github.com/vcheplygina/cats-scans/projects/1



Github project management

- Integrates with your favorite todo-list software via IFTTT
  https://ifttt.com/github



Github project management

My own advice:

- Task/issue should
  - Have an action (implement, write)
  - Be specific , what does “done” look like?
  - Doable in one “sitting”
- Do not start many tasks at the same time before finishing (“doing” column)

TODO before Thursday

Complete Github exercise (group does not matter here yet) and submit git log to LearnIT

Email vech@itu.dk if one or more people missing from your group

Thursday project 1 starts due to illness / schedule change