

Programming in MUDE

Robert Lanzafame, Tom van Woudenberg

Lecturers Civil Engineering and Geosciences

Python in Teaching, TU Delft, March 14, 2024

Python, Programming in Education...

...A Reflection from Civil Engineering and Geosciences

When did you acquire your programming/computer skills?

→ Probably not in your “Intro to Programming” BSc course!!!

We expect our students to learn everything here!

Programming problems:

- Students expected to do a lot
- Teachers unsatisfied with capabilities
- Rarely taught/used after BSc course
- We don't teach students how to share
- A growing problem?

Typical Colleague?

“I don't have time to learn Python”
“Find data, do analysis. Good luck!”
“What is a .ipynb file? I can't read this!”
“Students can't code”

We are part of the problem!

MUDE = Our Playground

Digital generation ≠ ability to use a computer



What is MUDE?

- CEGM1000: Modelling, Uncertainty and Data for Engineers (MUDE)
- 12 ECTS over Q1 and Q2
- 1st year MSc, all students in Civil Engineering and Geosciences (~300)
- Focus is on fundamental topics
- Programming and computer skills essential
- Many digital formats (videos, PDF, Jupyter Notebook, etc)
- **Challenge: diverse set of materials, students and teachers!**



Diversity of Topics

Q1

1. Modelling Concepts
2. Propagation of Uncertainty
3. Observation theory
4. Numerical Modelling
5. Continuous Distributions
6. Risk and Reliability Intro

Q2

1. Finite Volume Method
2. Finite Element Method
3. Signal Processing
4. Time Series Analysis
5. Optimization
6. Machine Learning
7. Extreme Value Analysis
8. Risk and Reliability, Part 2

Requires programming

Diversity of Students

Survey based on 261 students (100% response rate)

- 74%: programming experience limited to 1 BSc course
- 60% have only used Python
- 18% have only used Matlab
- 17% have used other languages (overlap with Python/Matlab)
- 7% have zero programming experience (19 students)
 - Almost entirely international students (~30% of total)

Typical feedback:

As an international student I have little programming experience and found it difficult to contribute to my group. Sometimes my group members taught me things, but I could tell that I was a burden on them. I want to learn to use Python, but there isn't enough time.

Diversity of Students

Our experience:

- Our BSc course prepares students to use numpy, matplotlib, etc
- Our BSc course does *not* teach students:
 - How to use a computer
 - To be comfortable with non-numpy data structures (e.g., dictionaries!)
 - How to communicate, collaborate or share effectively
 - The importance of reading code and writing readable code
 - Python \neq Jupyter notebook!!!!
- There are exceptions (CS minor – we hire them as TA's 😊)

These issues should be addressed at the curriculum level

We start by addressing them in MUDE

Weekly Schedule

Monday	Tuesday	Wednesday	Thursday	Friday
8:45 Lecture				Project Sessions Multiple rooms
10:30 10:45	Open Question Hours 1 room	In-Class Workshops Multiple rooms		

Everything builds towards the project on Friday. Repeat for every week.

Instructions: README.md
Analysis: Jupyter notebook
Submit: Report.md
(via GitLab)

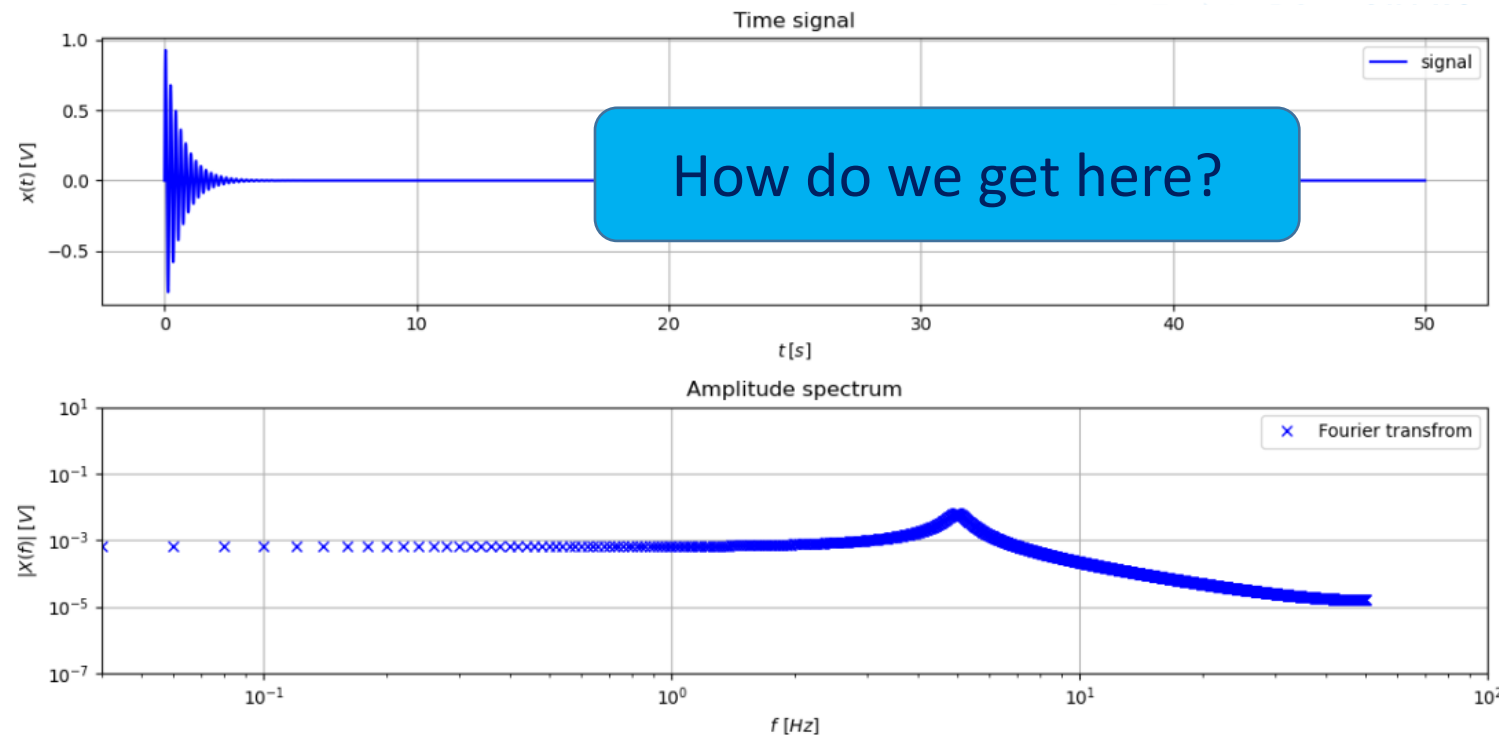
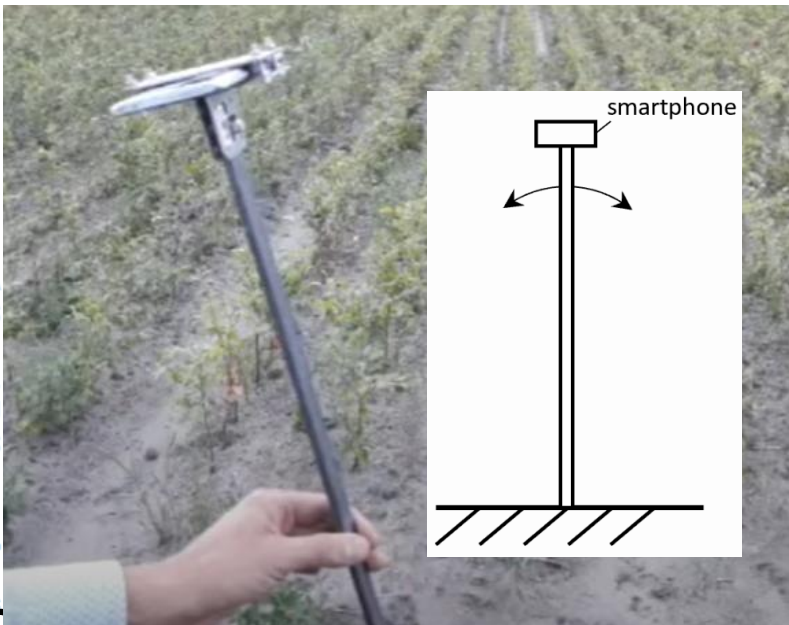
Learn material: read and interact with online textbook.

Programming Assignments: complete before Friday session. Submit via GitLab.



Example: Signal Processing (Q2, Week 3)

- Focus on analysis of signals in the frequency domain
→ Fourier Transform
- Project: apply and interpret for synthetic and real signals



Example: Siana

Programming Assignment

Overview of Assignment

This assignment will address iterating in Python, illustrating simple, but will be used throughout the course.

Specifically, we will look at:

- iterables and iterable objects
- iterables range, enumerate
- the modulo operator %
- plot type stem

```
team = ['green', 'red', 'blue']
score = [5, 9, 7]
```

```
# for YOUR_CODE_WITH_enumerate_HERE:
#     print(f'Team {i} has {j} points.'
```

```
for i, j in enumerate(score):
    print(f'Team {team[i]} has {j} points.'
```

Team green has 5 points.
Team red has 9 points.
Team blue has 7 points.

You may have noticed that enumerate is a bit verbose. You can define an unnecessary iteration index to accomplish things easier:

Task 4:

Use zip to print out the summary of points for each team.

```
team = ['green', 'red', 'blue']
score = [5, 9, 7]
```

```
# for YOUR_CODE_WITH_zip_HERE:
#     print(f'Team {i} has {j} points.'
```

```
for i, j in zip(team, score):
    print(f'Team {i} has {j} points.'
```

Team green has 5 points.
Team red has 9 points.
Team blue has 7 points.

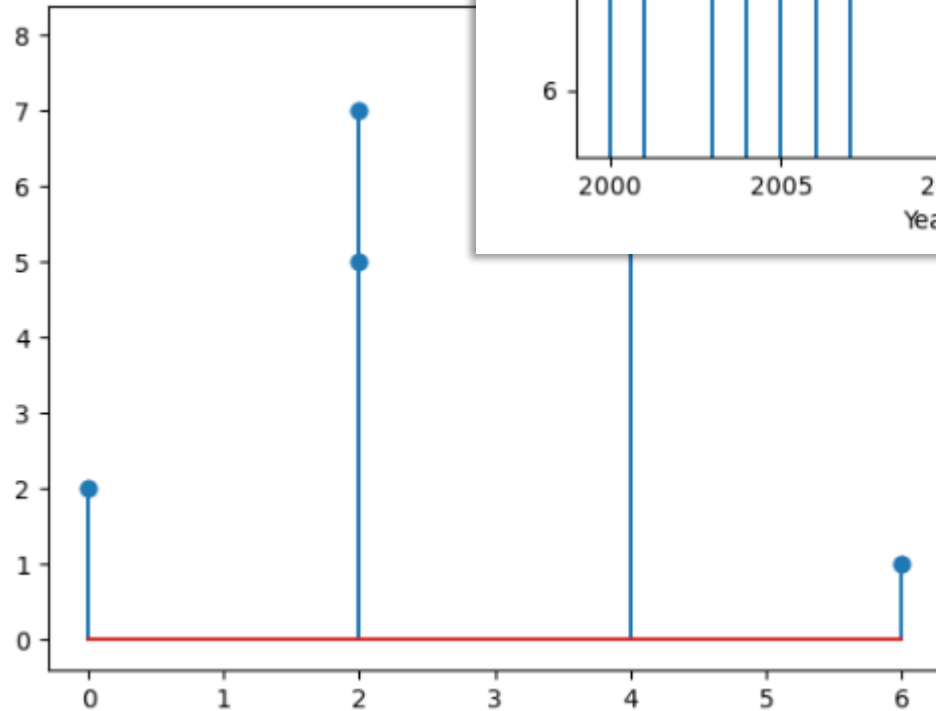
That's really compact!

You have probably used the matplotlib plot types plot, hist and scatter frequently; another type is stem, which is useful for indicating the magnitude of various points along a number line. As with a scatter plot, the values are easy to handle.

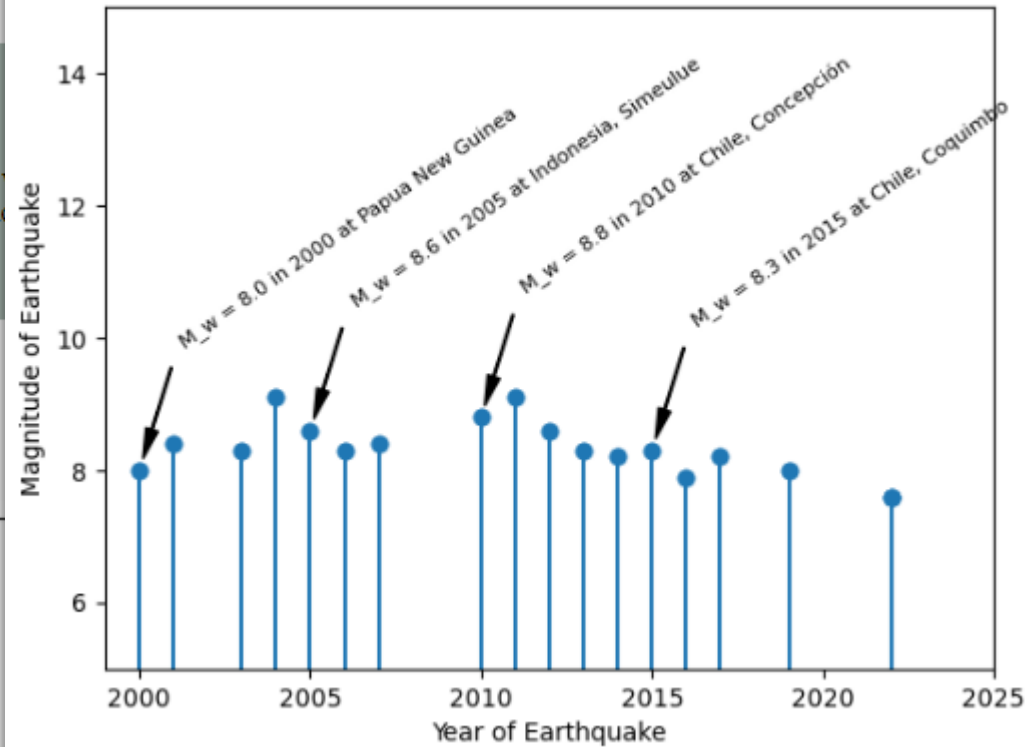
Task 6:

Run the cell below and play with the stem plot type. Do you see how the index is used?

```
value = [2, 7, 5, 1, 8]
index = [0, 2, 2, 6, 4]
plt.plot(index, value, 'o')
plt.stem(index, value);
```



Biggest earthquakes each year since 2000 on Ring of Fire

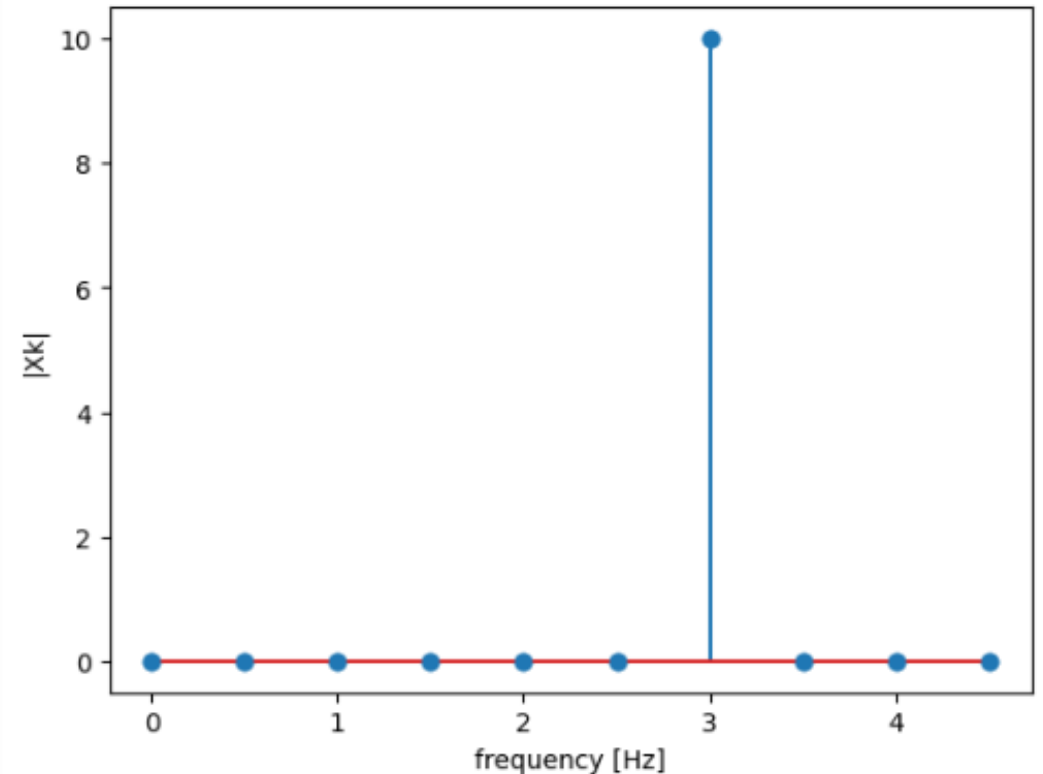


Example: Signal Processing (Q2, Week 3)

Workshop: make a DFT.
Easy after the Programming Assignment!

```
### SOLUTION
```

```
abs_fft = np.abs(np.fft.fft(xt))  
freq=np.arange(0,fs,1/T)  
plt.stem(freq[:int(N/2)], abs_fft[:int(N/2)])  
plt.plot(freq[:int(N/2)], abs_fft[:int(N/2)], 'o')  
plt.ylabel('|Xk|')  
plt.xlabel('frequency [Hz]');
```



Example: Signal Processing (Q2, Week 3)

Programming concepts are...

- Simple
- Used immediately
- Separated from theory
- Fun!

Theory in the book

$$X_k = \Delta t \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi}{N} kn}$$

This is the discrete Fourier transform (DFT), typically implemented in software packages as `fft` (in Python, we will use `numpy.fft.fft`).



Our TeachBook

MUDE

Search Ctrl + K

Welcome to the MUDE Textbook

Q1 Topics

- 1. Modelling Concepts
- 2. Propagation of Uncertainty
- 3. Observation theory
- 4. Numerical Modelling
- 5. Continuous Distributions
- 6. Risk and Reliability Intro

Q2 Topics

- 1. Finite Volume Method
- 2. Finite Element Method
- 3. Signal Processing
- 4. Time Series Analysis

Welcome to the MUDE Textbook

Welcome to the MUDE textbook. This is where assigned reading is located, along with interactive exercises to practice and study the module material.

Content in this textbook will be added throughout the semester, and changes will be announced and documented as needed. When new content is added, it will be done in large chunks. Changes will not be made retroactively that impact the scope of the exams (in other words, you will never need to re-study material because it changed prior to the exam). Notifications of errors, corrections and suggestions for improvement are gladly welcomed via the [Answers platform](#) and/or via the MUDE email address.

Interactive Pages—Use Python in your Browser!

This online textbook has a number of pages that are set up to be used interactively. On such pages you can use the "Live Code" button under the Rocket Ship icon in the top right to activate the interactive features and use Python interactively!

Sometimes the interactivity will involve completing an exercise, whereas on other pages it might simply provide the opportunity to edit the contents of code cells and execute it to explore the page contents interactively. Other pages may provide interactive figures (e.g., widgets).

This is a new feature that we are actively working to incorporate in the MUDE textbook, so please provide us feedback via the [Answers platform](#) if things go well or you have suggestions for improvement.



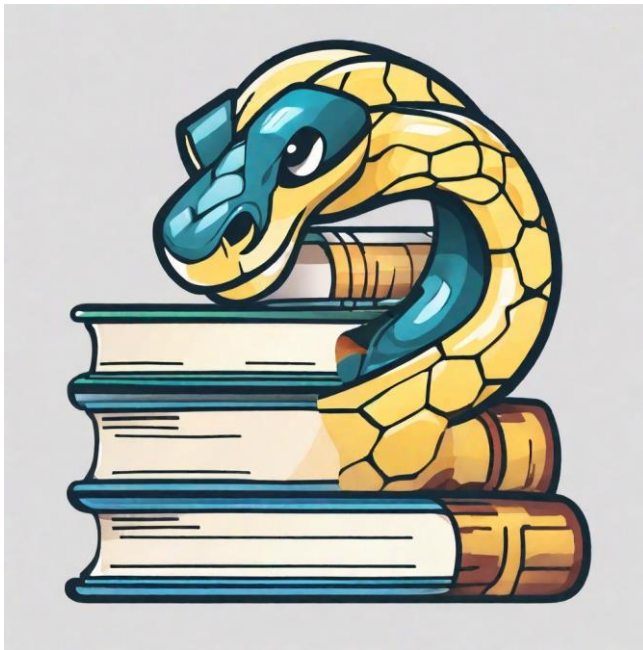
Our TeachBook

The interactive Python features were illustrated using 3 demonstration videos that are not included in this PDF, but can be viewed at the following YouTube Links:

- Sympy: <https://youtu.be/X0zrIwUKja4>
- Quiz Questions: https://youtu.be/eUmdEu_Z5us
- Neural Network: <https://youtu.be/8AeYnKn4Tcg>
- Confidence Intervals: <https://youtu.be/qCYA8z-u9DE>

**As of March, 2024 the book is only accessible with a TU Delft account, but this will change in the future.*

TeachBooks

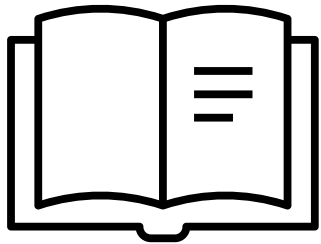


Teachers' Educational
Assistance for interaCtive
Hands-on Browser-based Online
Open Knowledge for Students

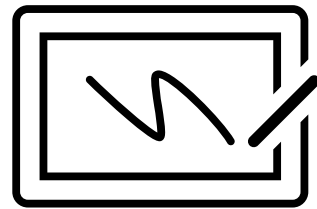


interactivetextbooks.citg.tudelft.nl

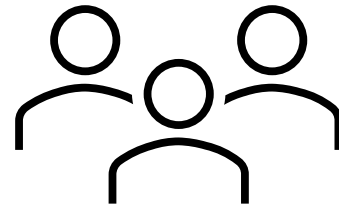
TeachBooks



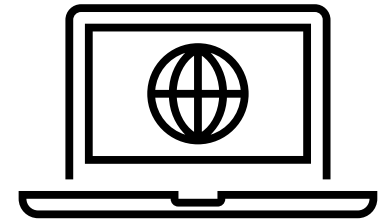
Manual on
collaboration
and interactivity



Template for
your own book in
15 clicks



Share content,
features and
advice within
community,
supported by
TAs



Your online book
via GitHub or
GitLab

Example – Python in browser (client-side)

Search Ctrl + K

Jupyter Book Manual

Getting started!

Installation and setup for different types of users ∨

Workflows ∨

Features

Basic Jupyter Book Features ∨

TeachBooks Features ∧

Hypothesis

Grasple questions

Exclude parts of book from published book

Page download options

Adding interactive h5p elements

Python-enabled interactivity ∧

Widgets

Exercise checking using check-answer button



Python-enabled interactivity

Our book has been able to run Python code live in the browser (thanks Max!). This page contains some installation instructions and the other sections show how to use this functionality to create interactive figures and feedback on code

Setting up Python live coding

To set up the Python live coding you need to add our [own sphinx-thebe extension](#) to your book. This extension doesn't rely on a 3rd party like Binder and it supports local python execution and custom features. Therefore, you need to add some lines to `requirements.txt` and `_config.yml`

For `requirements.txt` add the following lines:

```
--extra-index-url https://gitlab.tudelft.nl/api/v4/projects/11239/packages/pypi/simple
sphinx-thebe ~= 0.9.9
```

This will download the correct version of the sphinx extension when the book is build on the server (which loads the required packages from `requirements.txt`)

Afterwards, this sphinx extension needs to be enabled in your book. This can be done by adding the following lines to `_config.yml`:

```
launch_buttons:
  thebe: true
```

Setting up Python live coding

Instructions: local build

Custom cell tags

Additional packages

TeachBooks

Over 10 books actively
used in education

Over 1000 students
impacted

3 faculties

All of this with our “local”
CEG support (TA’s)

Imagine what we can do
together with a pool of
TA’s at the university!

We want You to be Involved



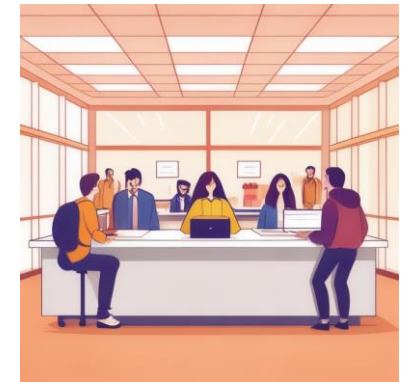
Feedback from
student-view



(Unexperienced)
teachers
promoting
interactivity

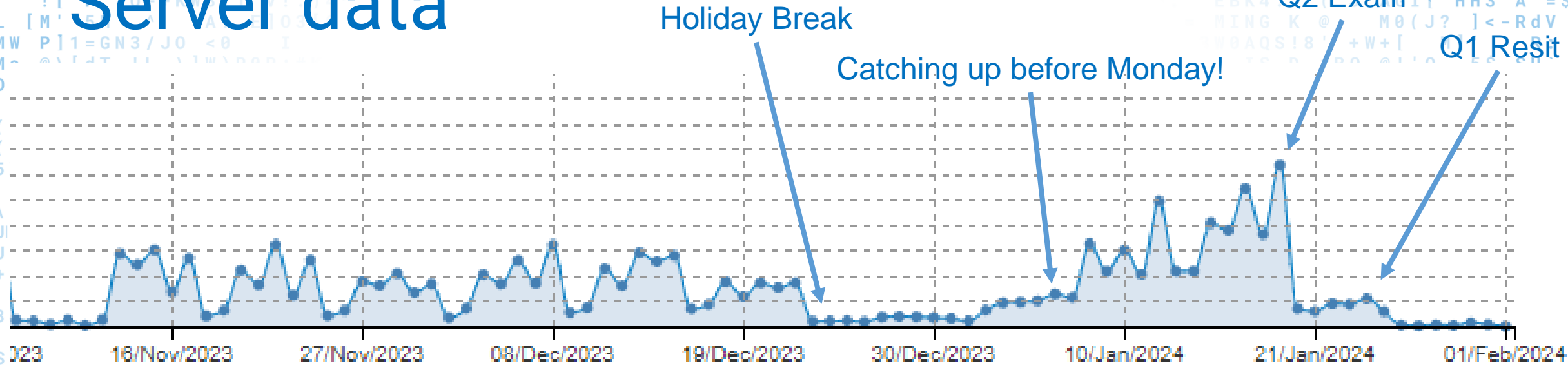


Experts sharing
content, features
and learn us how
to do this



Administrators
referring to our
platform

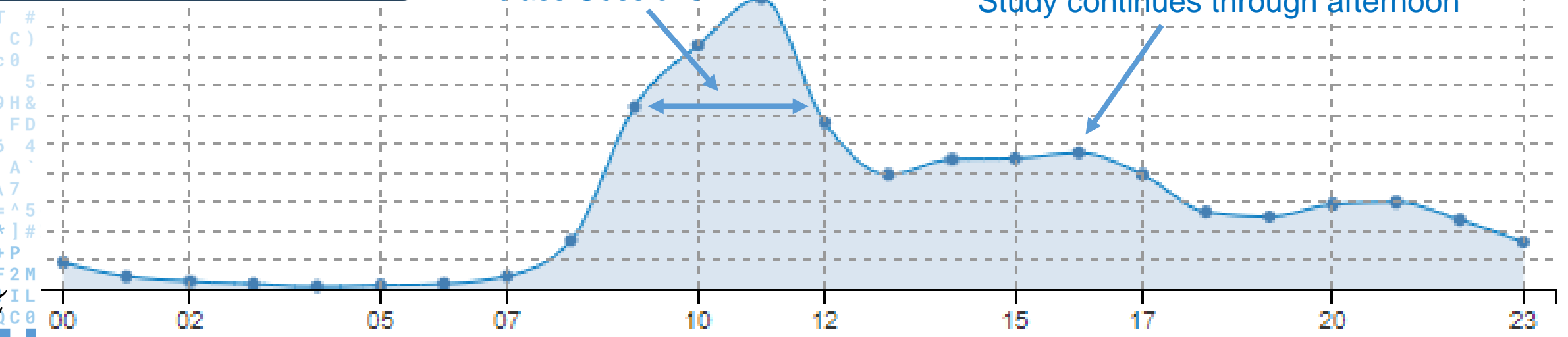
Server data



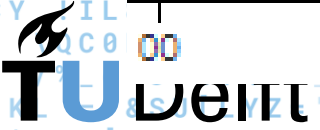
Vertical scales are relative measures of activity for 260 active students.

Usage of website during Q2

In-class Sessions Study continues through afternoon



Usage of website by hour of day (all days)



How students perceive the book (100% response rate)

Strongly Disagree Disagree Neutral Agree Strongly Agree

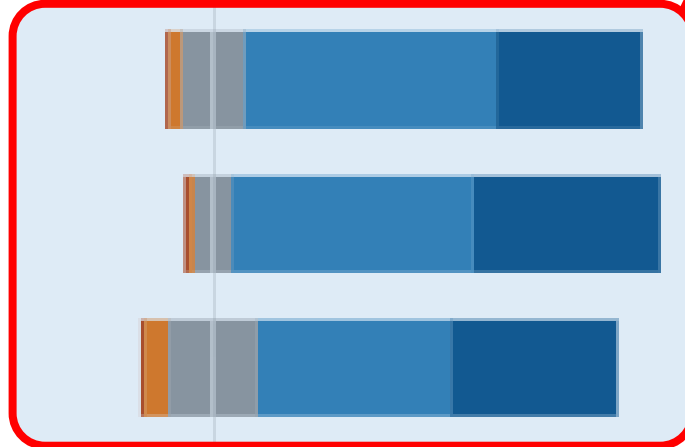
A website is better than a paper book



A website is better than a PDF book



Videos in the book were useful



In-line quiz questions were useful



Python-enable pages were useful



More courses should use this type of book



Interactivity is appreciated!



How students perceive the book (100% response rate)

Strongly Disagree Disagree Neutral Agree Strongly Agree

A website is better than a paper book

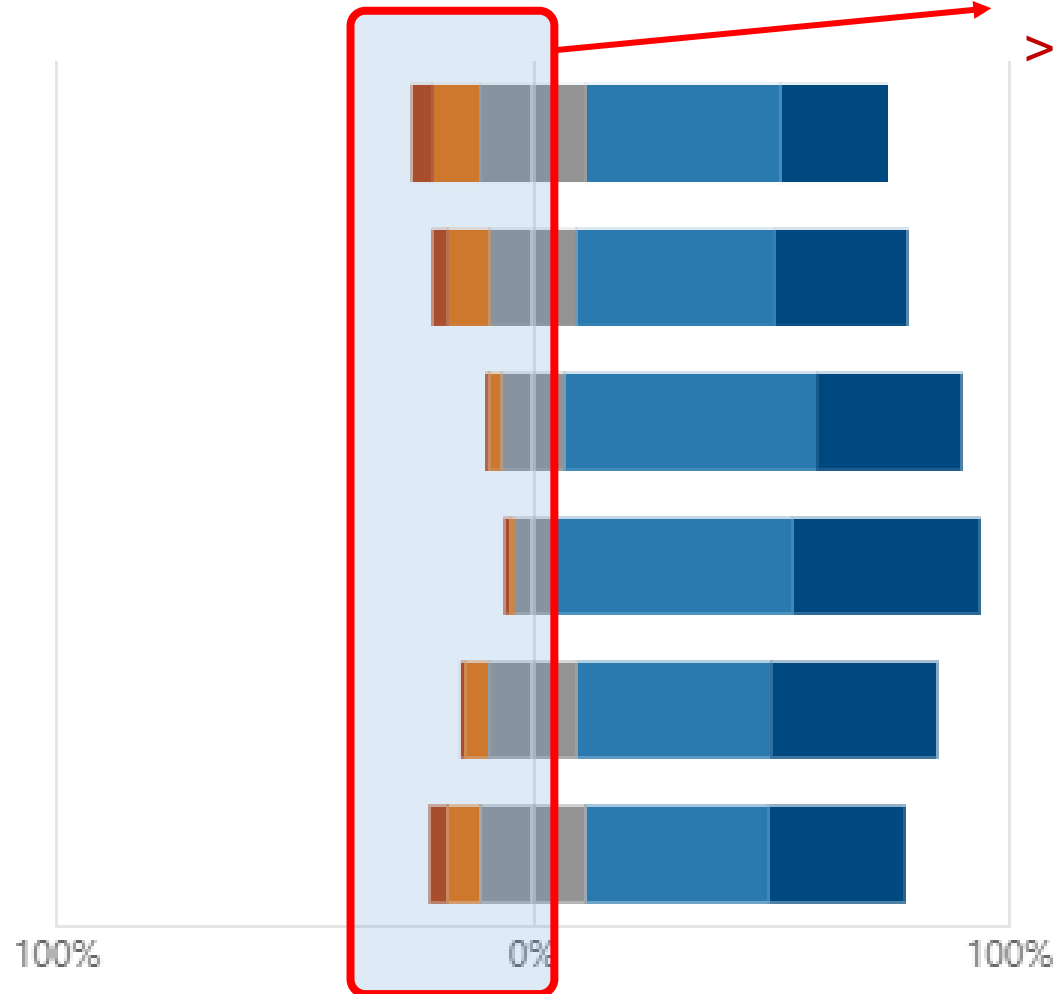
A website is better than a PDF book

Videos in the book were useful

In-line quiz questions were useful

Python-enable pages were useful

More courses should use this type of book



Students want their "own" version to study

>>> We will work on this in March, get in touch if interested!



Book the one and only solution?



Book is nice, but not everything

- How to work with your own Python environment?
- How to work with your own file system?
- Which IDE to use? And how to use it?
- How to collaborate with git?
- How to communicate using code?



Programming on own device with open-source tools



eum



Assignments via git

- MUDE group 2023-2024
 - Personal groups for students
 - One repository per assignment
 - Student group
 - One repository per assignment

- Repository are forks of central repository
- Linked to online file explorer with improved rendering



git

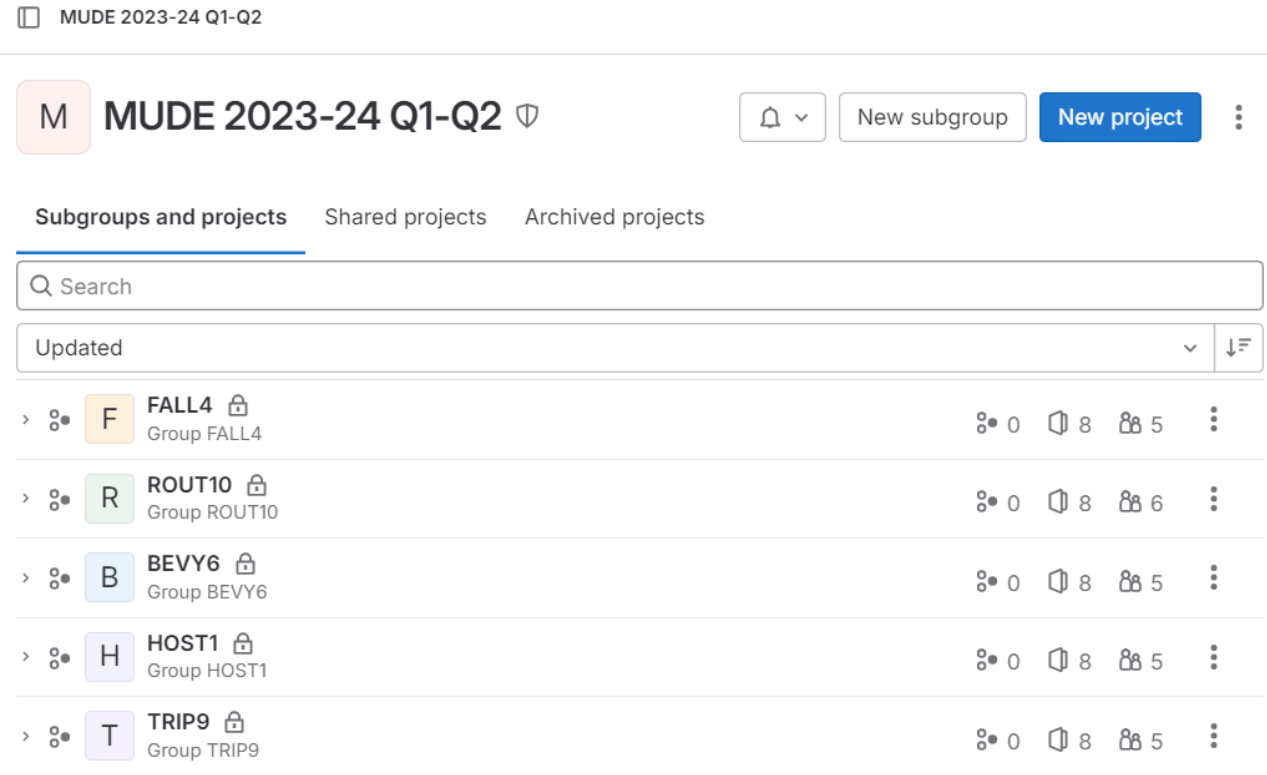


GitLab

Distribution via git

Distribution via python scripts using GitLab API

- Only 2 people could use them on fixed moments in time
- NetID is used for students
- Preferred git environment TU Delft



Assignments via git

GitHub Classroom

- Students can start at any time
- GUI for administration and student overview
- Requires GitHub accounts for students
- Requested whether this workflow allow GitHub over GitLab

Coding the Matrix Method in Python - Week 1

Individual assignment • Active

https://classroom.github.com/a/3jKvq3_M

Edit

Download

Assignment Details

Students total 3

2 Rostered 1 Added students

Accepted assignments 3

3 Students

Assignment submissions 3

2 Submitted 1 Not submitted

Filter by unlinked accounts




Filter by accepted

Filter by passing

Sort

Filters Search for an assignment

Classroom roster

	id_luri Not submitted @ibcmrocha 0 commits	Repository Feedback
	id_Tom Submitted @Tom-van-Woudenberg Latest commit 2 months ago 1 commit	Repository Feedback
	rlanzafame Submitted Link to student Latest commit 2 months ago 1 commit	Repository Feedback



Feedback via git?

Solution as:

- Branches
- Additional files

Feedback:

- Scripted automatic grading for programming assignments
- In a markdown file in their repo, or feedback via Brightspace



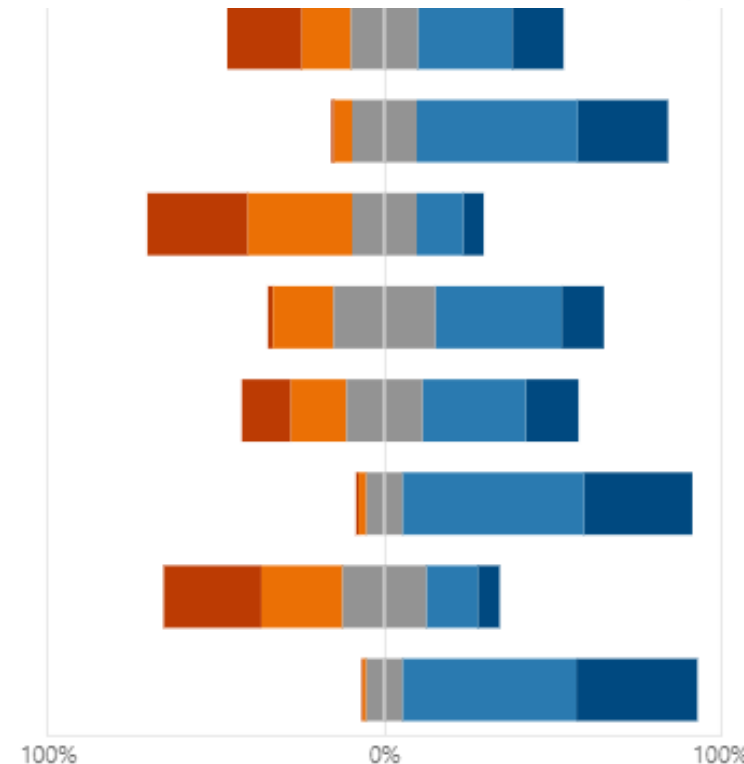
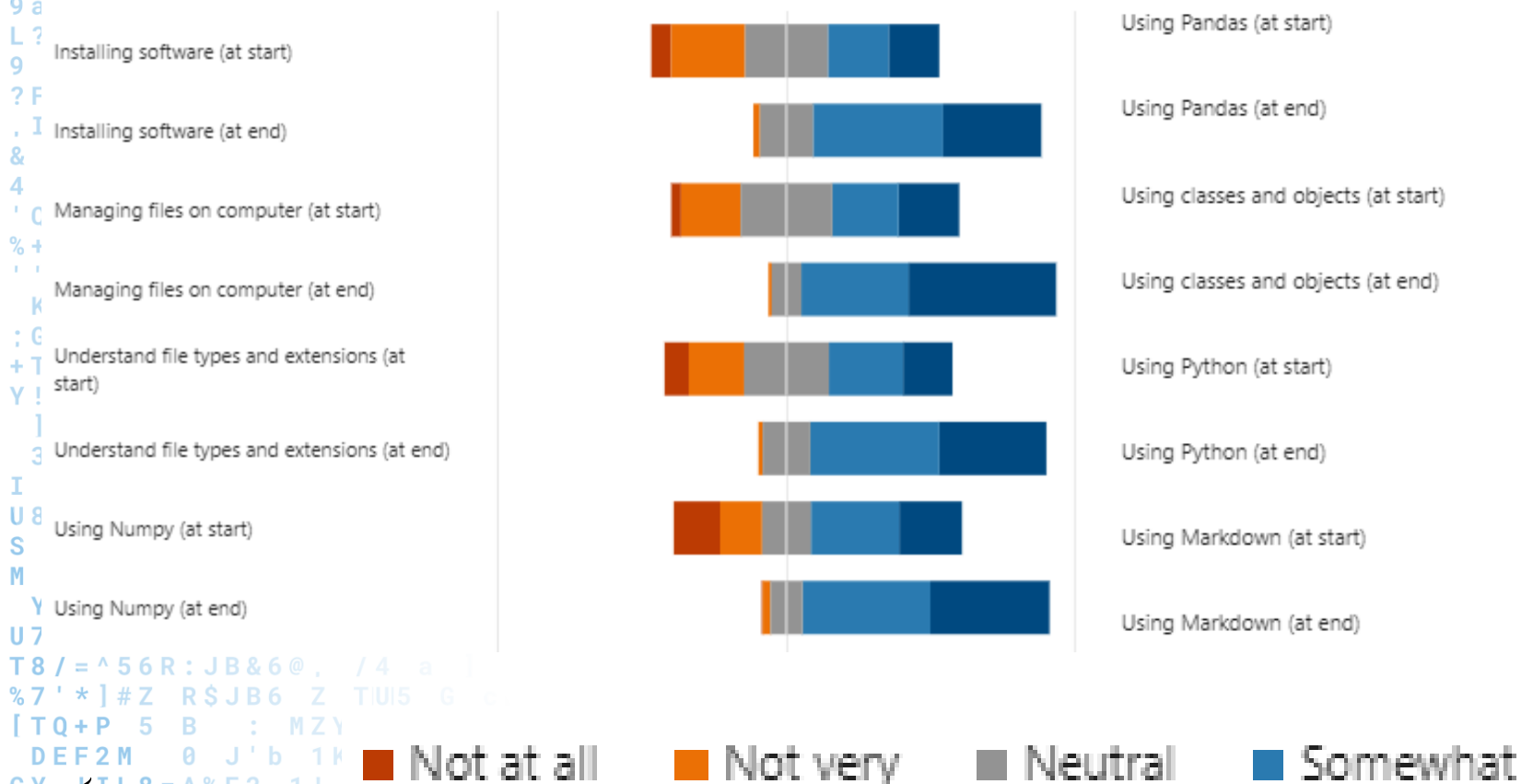
What do the students think?

- Survey asked paired questions:

Describe comfort level with _____ at beginning of Q1 and end of Q2

N = 261, response rate = 100%

How comfortable with topic _____, before/after?



■ Not at all
 ■ Not very
 ■ Neutral
 ■ Somewhat
 ■ Very



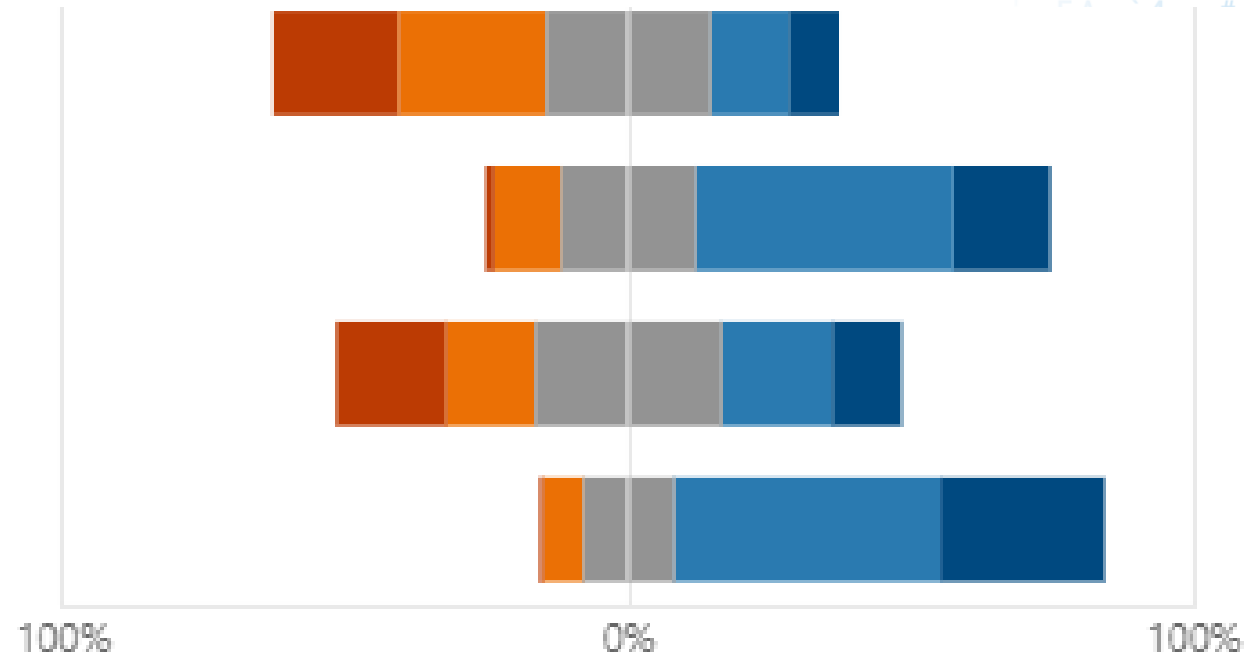
How comfortable applying for _____ that lists Python as a requirement, before/after?

Job (start of MUDE)

Job (end of MUDE)

Thesis (start of MUDE)

Thesis (end of MUDE)



■ Not at all
 ■ Not very
 ■ Neutral
 ■ Somewhat
 ■ Very



Thoughts for the future

Teaching materials
work for students
and teachers! 😊

- Supporting students: consider computer/programming context
- Supporting teachers: instructions, best practices, tech “support”
- Integration/alignment throughout curriculum: make information accessible
- Documentation + Communication is useful
- We will maintain links and resources on Digital Skills page
- Python: contribute examples to our book?
- Improving TeachBooks...with you!

Can you help us here?

- ...platform for communication / collaboration?
- ...license for edu material: only CC-BY?
- ...funding for book TA's for all faculties?

Dreams for the future

Frustrating:

- Explaining to administrators what we need as teachers
- Software and programming-related decisions lacking teacher perspective

→ Community of teachers collaborating on programming-related:

- Content
- Teaching
- Workflows



Contact and Resources

Contact:

- Robert/Tom: TeachBooks@tudelft.nl
- MUDE: R.C.Lanzafame@tudelft.nl

Information websites:

- teachbooks.tudelft.nl
- github.com/teachbooks
- mude.citg.tudelft.nl/2023/overview

Our Python book:

- github.com/TeachBooks/learn-python
- teachbooks.github.io/learn-python

Our Prob/Stats book:

- github.com/TUDelft-CITG/learn-probability
- tudelft-citg.github.io/learn-probability
- May move to GH/TeachBooks soon....