Open Source and PhD: a match made in heaven

Luca Ferranti, University of Vaasa

Outline

WHO AM I?

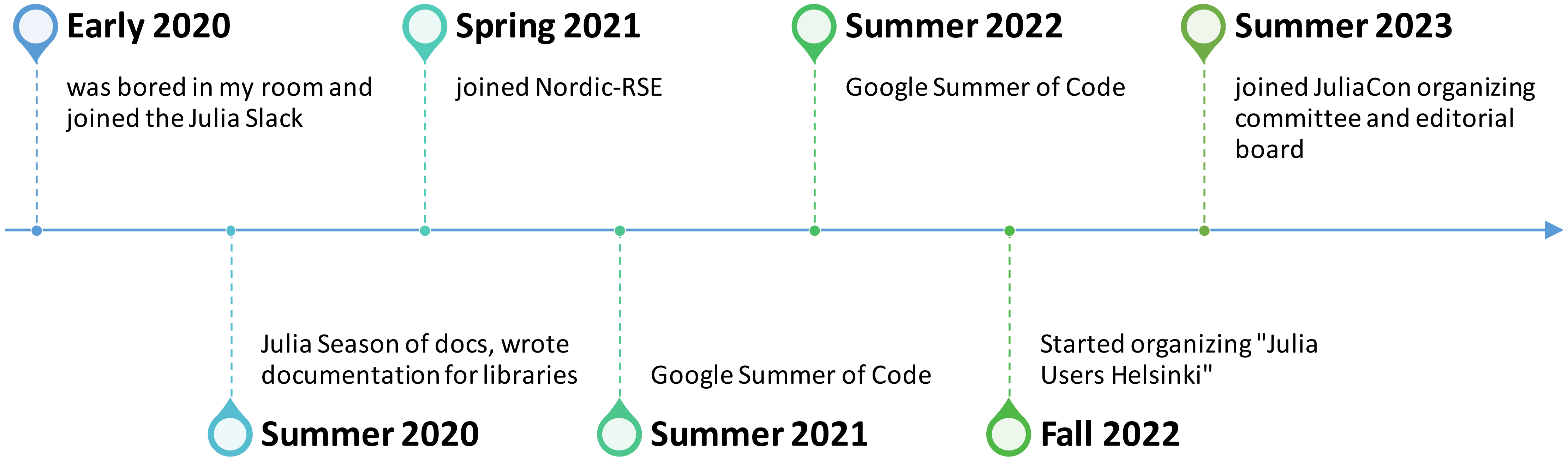
WHAT IS OPEN-SOURCE?

WHY IS IT IMPORTANT?

HOW CAN IT HELP YOUR PHD?

WHERE TO GO? HOW TO GET STARTED

My open-source timeline



Early 2020

was bored in my room and joined the Julia Slack

Summer 2020

Julia Season of docs, wrote documentation for libraries

Spring 2021

joined Nordic-RSE

Summer 2021

Google Summer of Code

Summer 2022

Google Summer of Code

Fall 2022

Started organizing "Julia Users Helsinki"

Summer 2023

joined JuliaCon organizing committee and editorial board

What is open source?

Open source is **NOT** only about coding

You can get involved even if you don’t know / don’t like coding! Why?

Because first and foremost, open source is **about community!**

Open source is building a community of people around a software which is public and free (as in free beer, and free of speech)

Why is open-source important?

We live in the Reproducibility crisis!

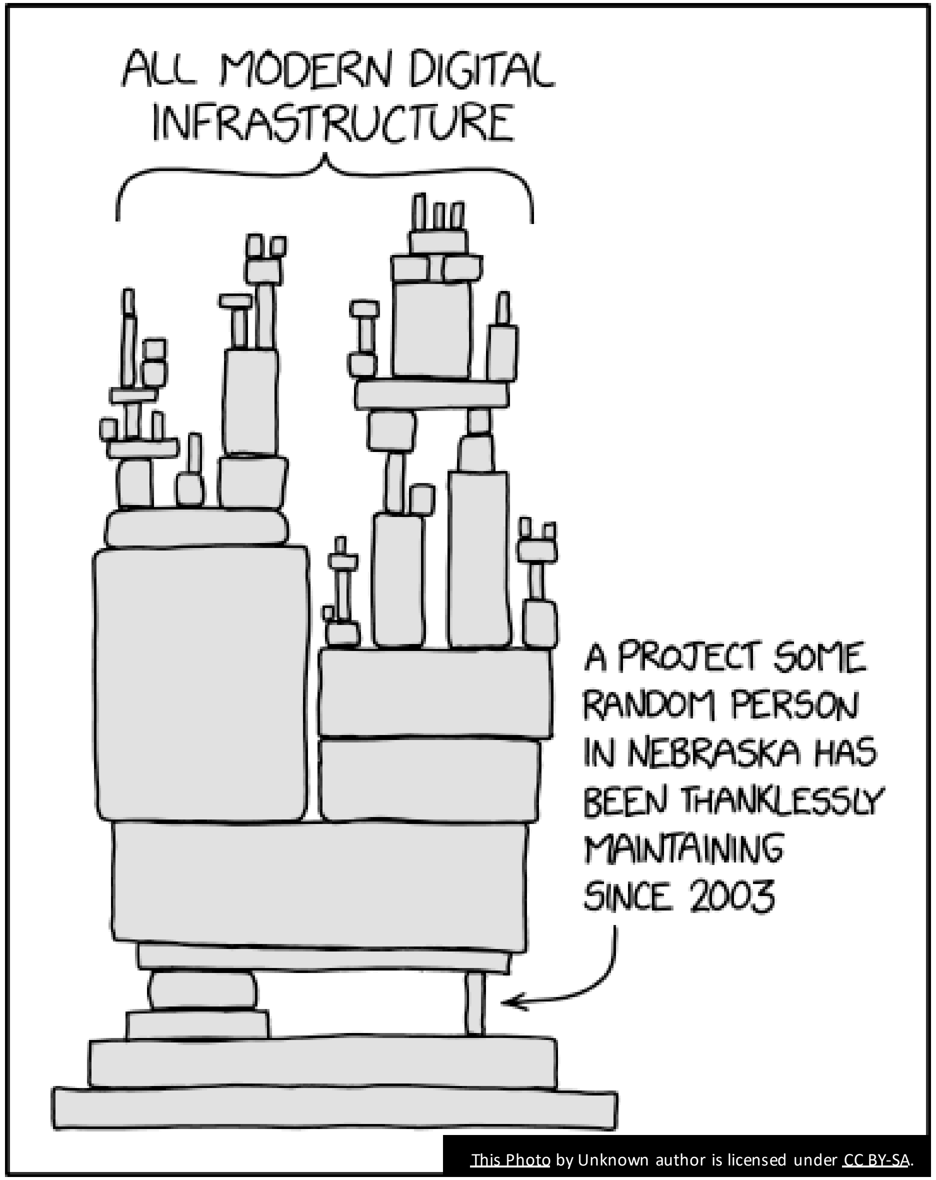
- Reproducing (numerical) experiments is getting harder and harder
- People use research software in their work, and researchers need it!

Why? Because researchers lack a lot of technical skills NOT THEIR FAULT!

- It works on my machine.... Yes, but you are not giving your machine to the world!
- Used libraries change versions, no versions pinned
- Sometimes behavior depends on operating systems
- Messy organization of experiments -> accidentally uses old data, side effects.

How much is your work depending on open source software?

- Several popular tools / languages are open source
- Python
- TensorFlow / keras
- Julia
- Sk-learn
- …
- ...


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Pros of doing open-source during your PhD

- You don't feel alone: meet a long of young researchers alike
- Amazing networking!
- Less hypocrisy, less b*llsh*t metawork
- In theory, possibility of higher impact of your work (if you write a software, others might actually use in their reseatch), but... see later
- Develop skills that help in industry transition

Risks / challenges of open-source

Fully volunteer, can do as much as you want, easy to get too excited

Risk of burn-out! It is ok to say no, remember to take your time to relax!

Maintaining research software is a longer commitment, can feel overwhelming (but is rewarding)

Extra work in addition to writing papers, academic system also pretty bad at rewarding this

Open source in the current academic system

Unfortunately, research software is not very well recognized

In theory, you can "just" archive your software versions / data in zenodo, to get a DOI and make it citable

In practice, unfortunately, sometimes the academic systems regards them as "second hand" research output

This is a pity, because research software should be more valuable than claiming to improve the state of the art by 1% in one specific case.

Software papers: write a paper associated with your software (since writing a book of documentation is not enough)

Journal of Open Source Software

https://joss.theoj.org/

Peer-review full open access, free of charge

As open as possible, reviews are carried as public github discussions

Write a short paper (1-2) associated with your software

Review actually focuses on the code, unlike several other journals

How to get started with open-source?

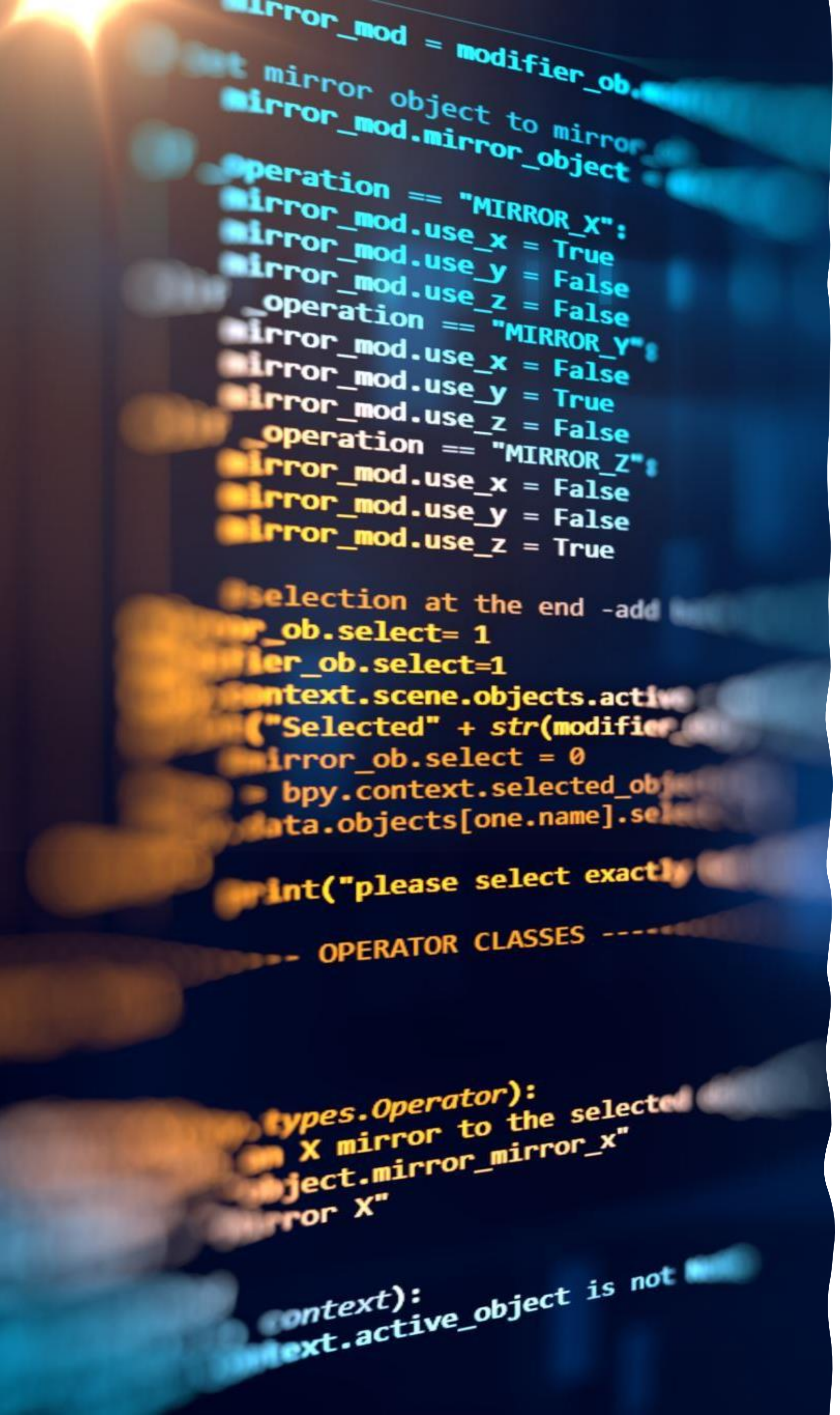
- Find a topic that makes you passionate, maybe a tool you use frequently and would like to contribute to
- I'll tell you a secret: **you don't need permission to get started, if you like it, just do it**
- Open-source contributions are **NOT** only code. We need more non-technical people, more tutorials, videos, blogs, etc. Etc.

Google summer of code

- https://summerofcode.withgoogle.com/
- Scholarship offered by google to contribute to open-source projects during summer.
- Join an open-source org, get a cool mentor, do cool stuff **and get paid**
- **Important:** very selective, generally requires you have already had some involvement in the organization.

How to get technical skills?

- **Carpentries:** https://carpentries.org/ non-profit organization that teaches fundamental of computing practices (git, shell, python/r for data analysis)
- **CodeRefinery:** https://coderefinery.org/ N ordic organization to teach good coding and research software practices to researchers and professionals. **Amazing material**, organizes workshops 1-2 a year.
- **Join a chat and ask questions when you are stuck, people are generally very helpful.** (remember to be mindful, everyone is volunteer, give enough info in your question)



Can this be a career path in academia?

- Researchers have already a huge workload, expecting them to learn git, software development etc. can be a burdain
- **Research software engineers (RSE):** professional figure that helps researchers developing and maintaining their code/data.
- There are a lot of RSEs without knowing it, if you write a lot of code for your research and help others, you are!
- Work in progress to get more recognition in academia

Research software engineers, nordic landscape

We have a nordic association! Come chat with us, great talks, great activities, great people https://nordic-rse.org/

Aalto has a very strong RSE group https://scicomp.aalto.fi/

If you write a lot of code in academia, you are not alone, join us!

I have a dream

- ***For code to be as important as papers***
- ***For research software to be considered research output***
- ***For researchers for getting help (training / support) with coding matters easily***
- ***For Research Software Engineers to be a viable and respected career path in academia***

You may say I am a dreamer, but I am not the only one