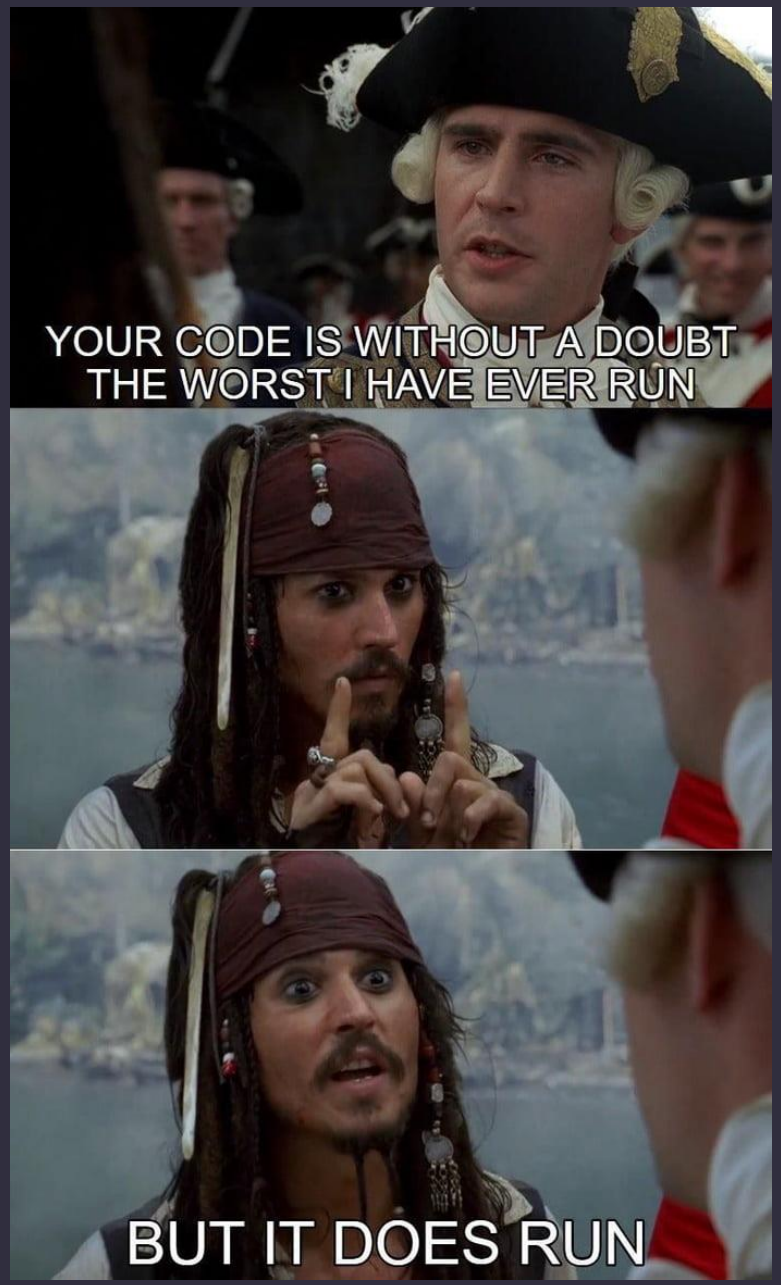
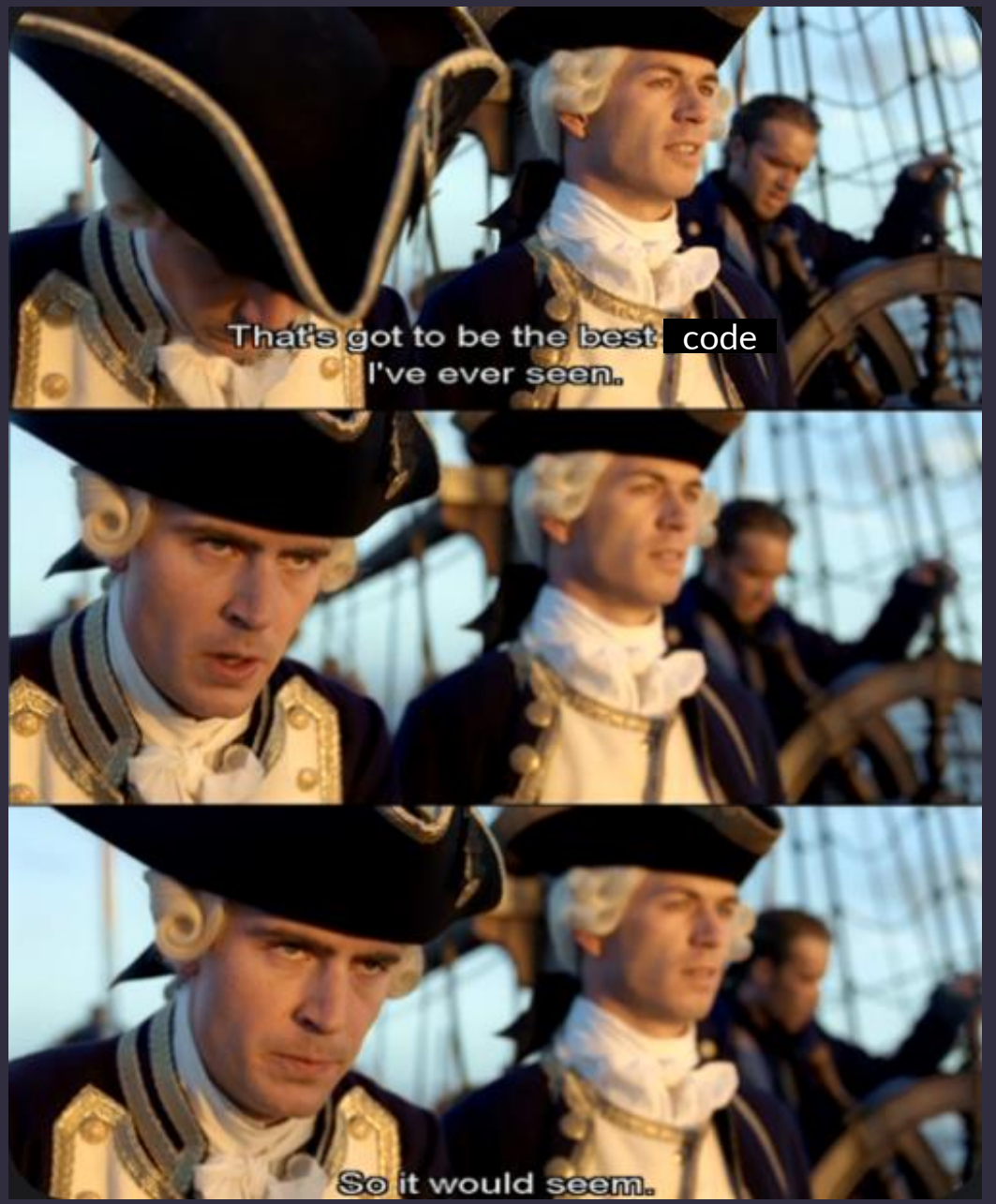
Good Scientific Code

A WORKSHOP ON WHAT, WHY, HOW

BEFORE:



AFTER:



Outlook

- Version control with git
- Clear code
- Software developing paradigms
- Collaboration & publishing code
- Documentation
- Scientific project reproducibility

This is a six-blocks workshop, developed over 3 years, combining textbooks, other workshops, online tutorials from field experts, blog posts, personal experience developing and documenting 10+ software, and research on how to make reproducible science.

Obvious disclaimers:

- These slides describe the ideal scenario
- Writing good code is not instant
- Writing good code is a ***craft***...

1. Iterative process: you constantly improve your skills, and learn more!
2. While general rules guide most of it, details depend on personal opinions!

Housekeeping

- This is a (mostly) language-agnostic workshop, meaning that the principles are about general coding. Examples and exercises will be in Julia and Python
- Workshop has 6 blocks, each with “lecture” and “exercises” parts intertwined
- Bring your own code base, for a small project (e.g., 2-3 plots of recent paper) which we will hopefully transform from something shameful to something prideful
- Workshop materials are here: https://github.com/JuliaDynamics/GoodScientificCodeWorkshop
- This workshop has ~120 slides, most with unique information
  - This is baptism by fire. You are not expected to retain everything. You will need to revisit things!

- ***If at any point you wish to ask a question, or discuss what is being presented, please feel free to immediately interrupt and ask away!***

What is the purpose of code?

- Technically, code is instructions for the computer…
- But, in the end of the day, your code is a mean to solve a problem!
- *"Programs are meant to be read by humans and only incidentally for computers to execute."*, Donald Knuth

- Code is written and read by humans! It is just another form of ***technical writing***, similar to a paper: must have ***crystal clear communication*** of the problem solving!

- Good Scientific Code =

  Clear, Easy to understand, Well-documented, Reproducible, Testable, Reliable, Reusable, Extendable, Generic

Not only you will learn how to achieve all of these in this workshop, but it will be made clear that they aren't costly.

Choose Your Weapon

- ***Choose what language fits what you want to achieve***
  - Expressivity (how fast and flexibly you can put your ideas to code)
  - An existing library that you want to use
  - Performance is absolutely critical
  - Re-usability and composability with other packages is important
  - I will need to do a lot “science” (plotting, modelling, testing, querying, interactiveness)

- My strong recommendation (because its good for ***all*** of the above bullet points)