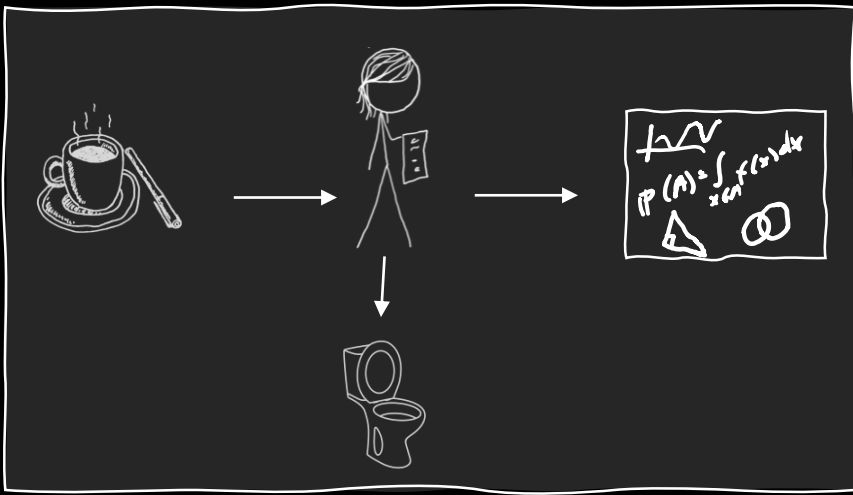
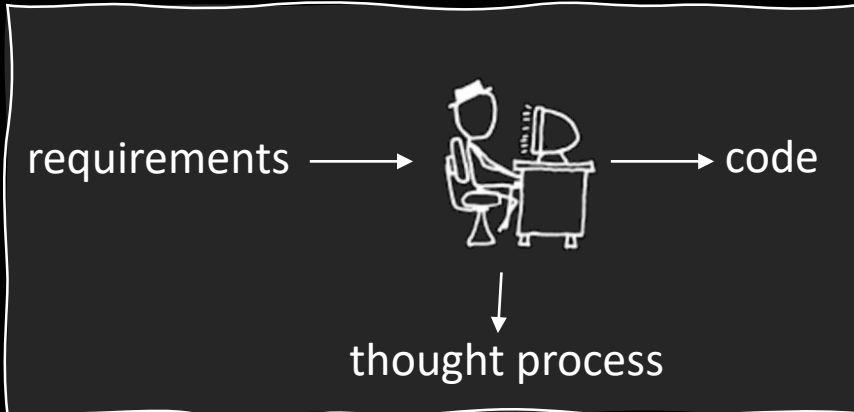


IA Scientific Computing

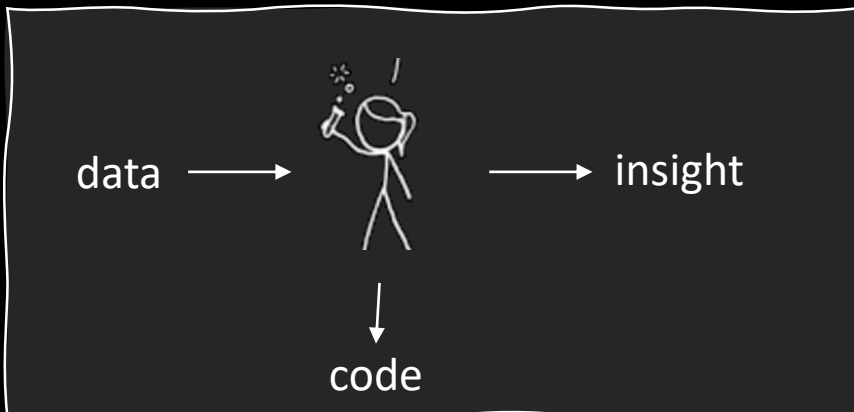
Dr Damon Wischik



“A mathematician is a machine for turning coffee into theorems” – Erdős / Rényi

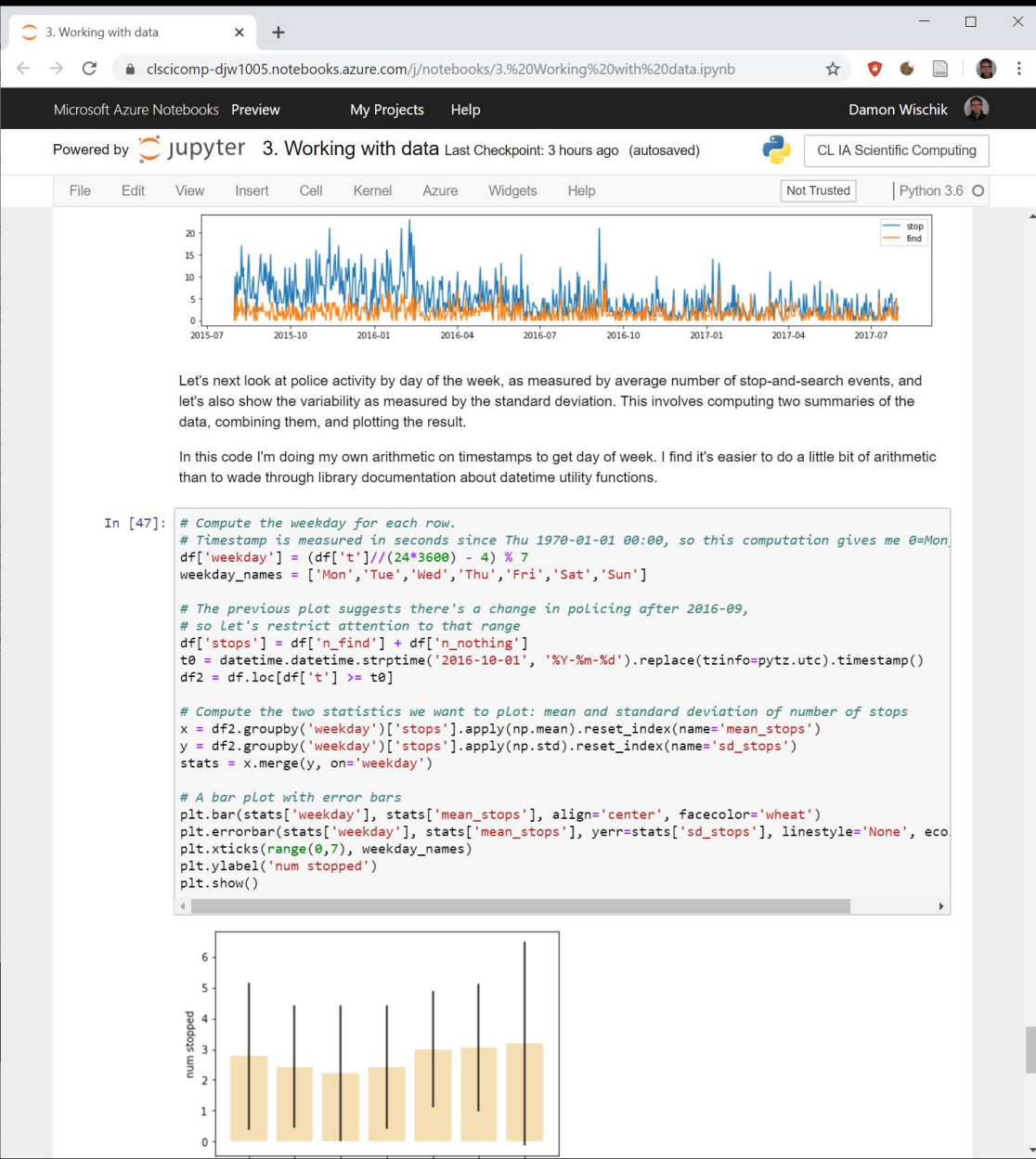


A software engineer



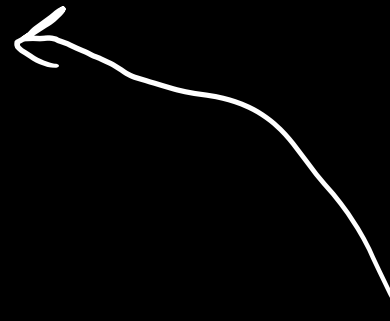
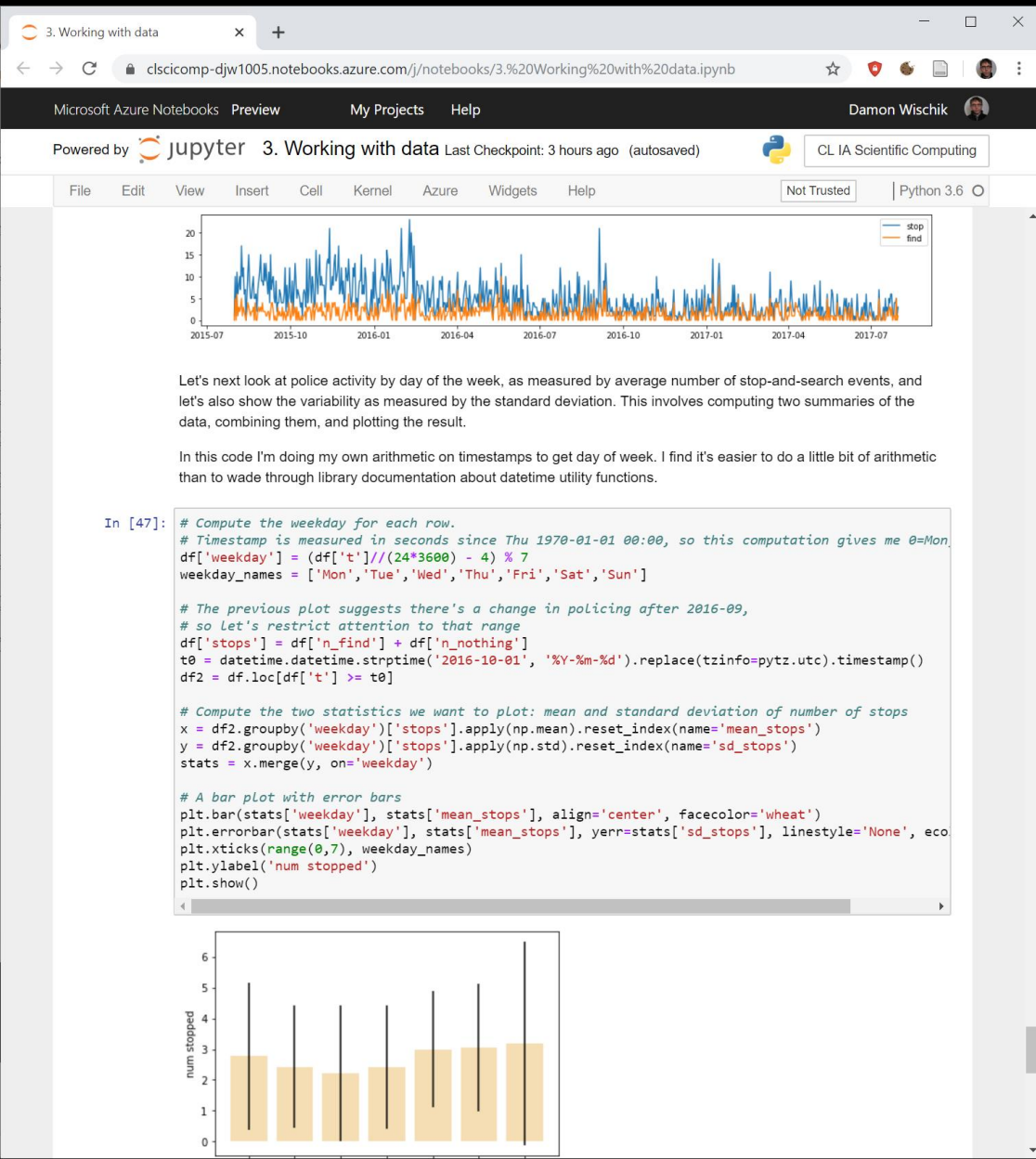
A scientist / machine learner / modeller

What is scientific computing?



Jupyter notebooks
Python
numpy + pandas + matplotlib

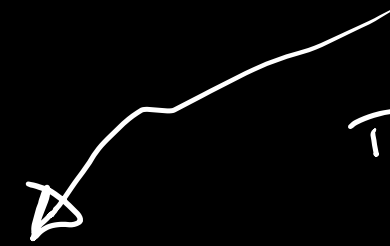
How to write bad code



first I ran this cell



and now this cell is producing strange answers



Then this cell, I think

What does good code look like?

What Not to Do

- Your ML has doubtless been one big file where you threw together all the functions and value declarations
- Lots of C programs look like this :-(
 - *We could emulate this in OOP by having one class and throwing everything into it*
- We can do (much) better

OOP Concepts

- OOP provides the programmer with a number of important concepts:
 - *Modularity*
 - *Code Re-Use*
 - *Encapsulation*
 - *Inheritance (lecture 5)*
 - *Polymorphism (lecture 6)*
- Let's look at these more closely...

Modularity and Code Re-Use

- You've long been taught to break down complex problems into more tractable sub-problems.
- Each class represents a sub-unit of code that (if written well) can be *developed, tested and updated independently* from the rest of the code.
- Indeed, two classes that achieve the same thing (but perhaps do it in different ways) can be swapped in the code
- Properly developed classes can be used in other programs without modification.

How to make effective use of notebooks



“Every line of code is written without reason, maintained out of weakness, and orphaned by chance.”

Jean-Paul Sartre



“Look at each line of your code and ask yourself: ‘does this spark joy?’ If not, don’t keep it.”

Marie Kondo

For this course, you should create your own notebooks from scratch.

- A notebook is not a logfile.
It’s a document, which you edit and reshape.
- A notebook is not a source file.
Text cells are more important than code cells.
- A notebook is for exploring and explaining.
One-off tasks.
Many of the questions are one- or two-liners.

<https://programmingisterrible.com/post/139222674273/write-code-that-is-easy-to-delete-not-easy-to-extend>

“Effective Jupyter notebooks” <https://news.ycombinator.com/item?id=19860955>

“JupyterCon: I don’t like Notebooks” <https://news.ycombinator.com/item?id=17856700>

Don't set traps for yourself with disorganized notebooks

- You should accumulate code during a work session, but trim it at the end of each session.
- Your reader should be able to understand your work by reading top-to-bottom.
Your code should all work if run top-to-bottom

while working

```
imports
experiment 1
debug code
tweaked experiment 1
experiment 2
update to experiment 1
for gotten import
```

after you've finished

```
imports
utility functions
run-once setup code
functions that implement
your solutions
import autograder
submit solutions to
autograder
```

Submitting your answers to the autograder

The image displays two overlapping windows. The background window is a Microsoft Azure Notebooks interface, showing a Jupyter Notebook with the following code:

```
In [ ]: !pip install ucaml --up
import ucaml

In [*]: GRADER = ucaml.autograd

log in

Waiting for you to log in

In [ ]:
```

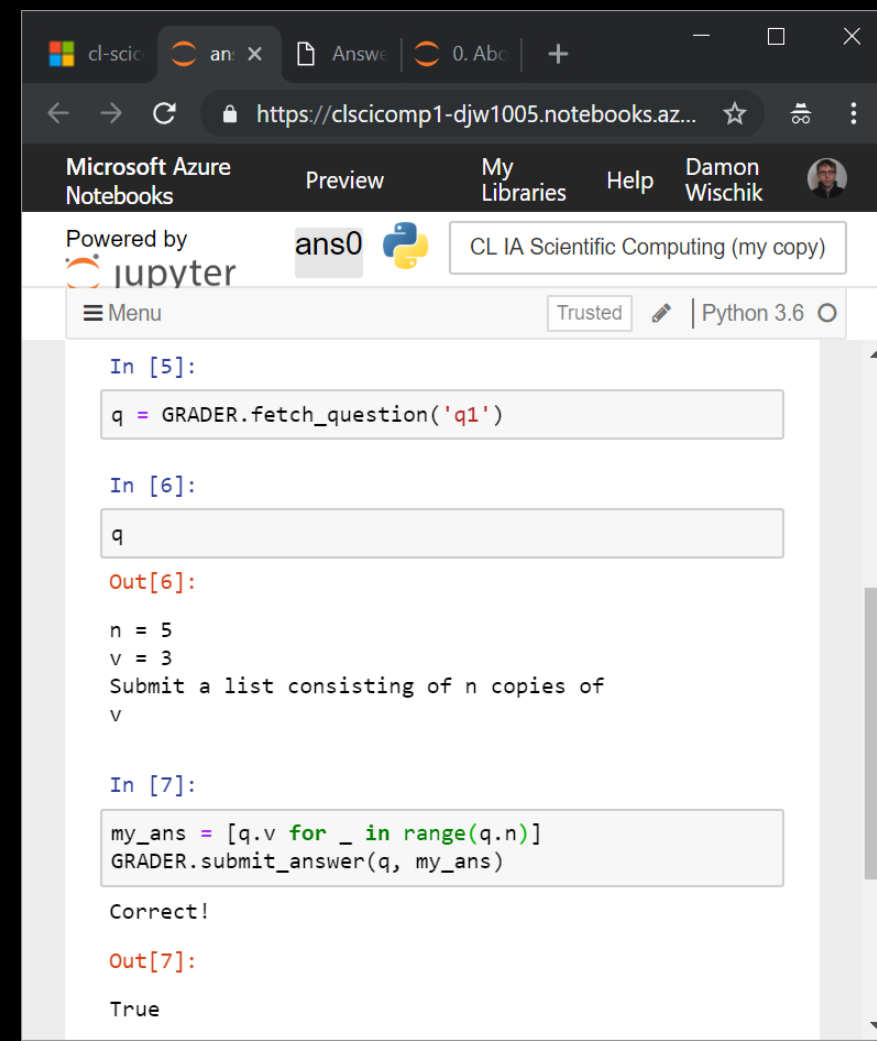
The foreground window is a web browser showing the <https://markmy.solutions/questions/...> page. The navigation bar includes links for [djw1005](#), [questions](#), [answers](#), and [logout](#). The main content area displays the course title **IA Scientific Computing** and a grid of assignment cards:

assignment2a	assignment2b
Assignment 2a	Assignment 2b
answered 4/4	answered 0/3

assignment3a	notes0
Assignment 3a	About the course
answered 3/7	answered 1/1 optional

Submitting your answers to the autograder

```
# Code to run in the preamble to the "submit solutions" section of your notebook:  
  
!pip3 install ucaml  
import ucaml  
GRADER = ucaml.autograder('https://markmy.solutions', course='scicomp').subsection('notes0')  
  
# To answer a question:  
  
q = GRADER.fetch_question('q1')  
my_ans = ... # using the parameters in q  
GRADER.submit_answer(q, my_ans)
```



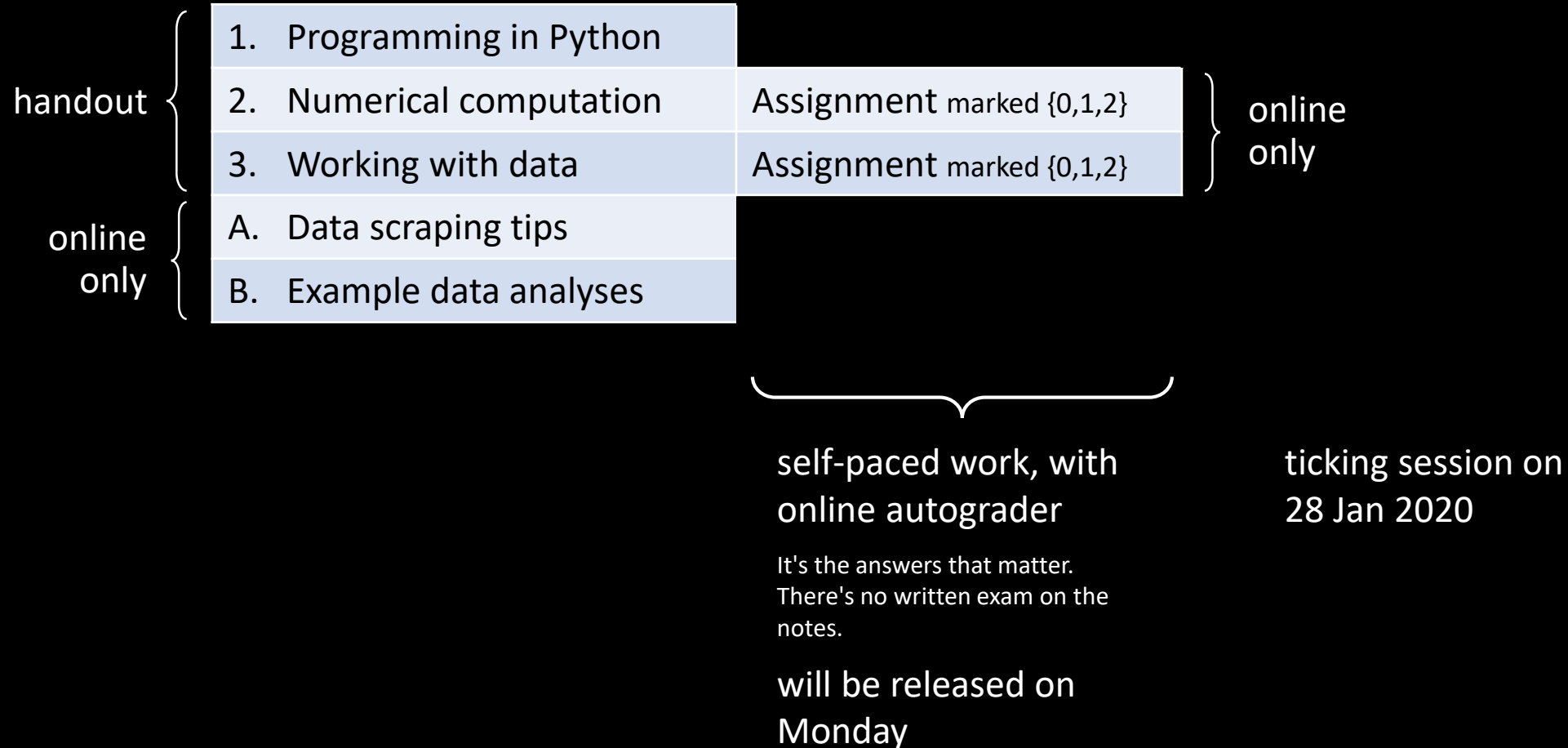
The screenshot shows a Jupyter Notebook interface in a web browser. The browser's address bar displays the URL `https://clscicomp1-djw1005.notebooks.azure.com/`. The notebook's title bar indicates it is "Powered by Jupyter" and "ans0". The notebook content shows three code cells:

- In [5]:** `q = GRADER.fetch_question('q1')`
- In [6]:** `q`
- Out[6]:**

```
n = 5  
v = 3  
Submit a list consisting of n copies of  
v
```
- In [7]:** `my_ans = [q.v for _ in range(q.n)]
GRADER.submit_answer(q, my_ans)`
- Out[7]:**

```
Correct!  
True
```

How is the course structured and examined?



marking academic work \neq testing code

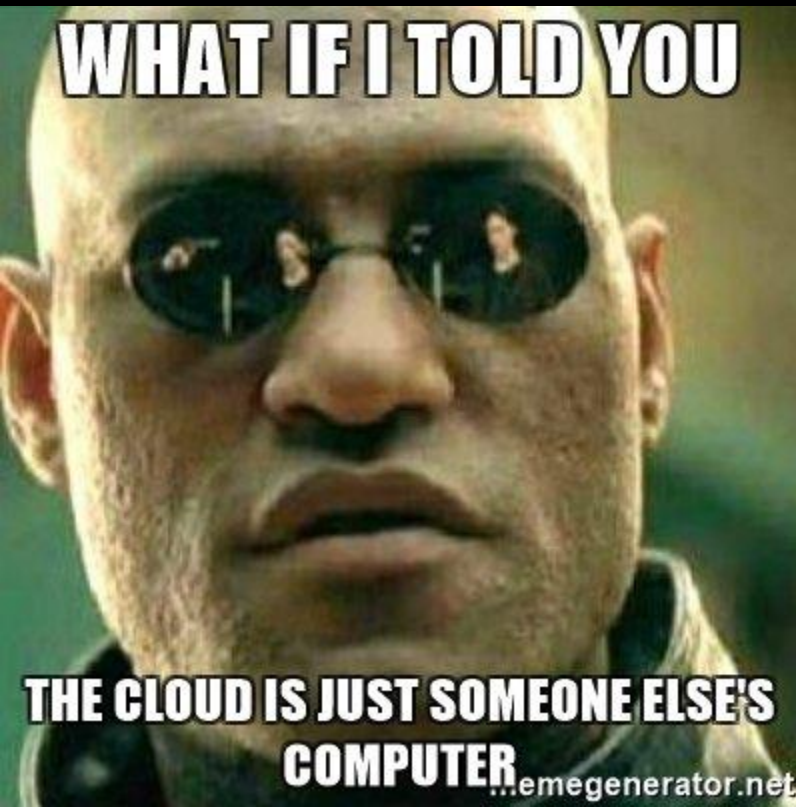
- It's *your job* to write your own unit tests.
- Nature's datasets are full of quirks, and nature is out to deceive you.
- As a data scientist you need to be ever vigilant, always checking your assumptions.

MY CODE PASSED QUESTION 2.
BUT IT WAS BUGGY CODE, AND IT
TOOK ME AGES TO DEBUG AND
PASS QUESTION 3.

YOUR GRADER SUCKS.



Running Jupyter

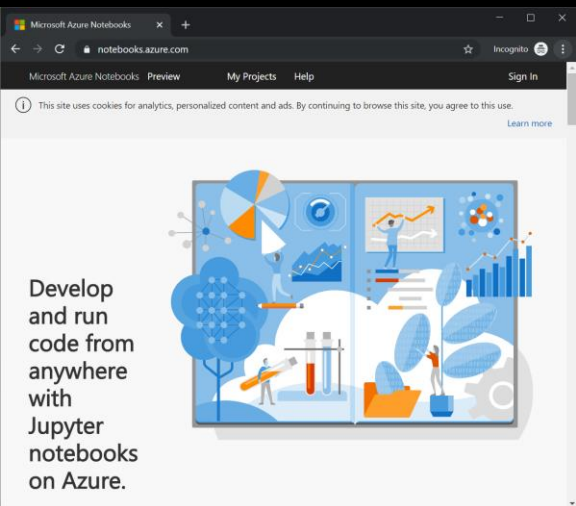


... and it will fail when you need it most.
Backup!

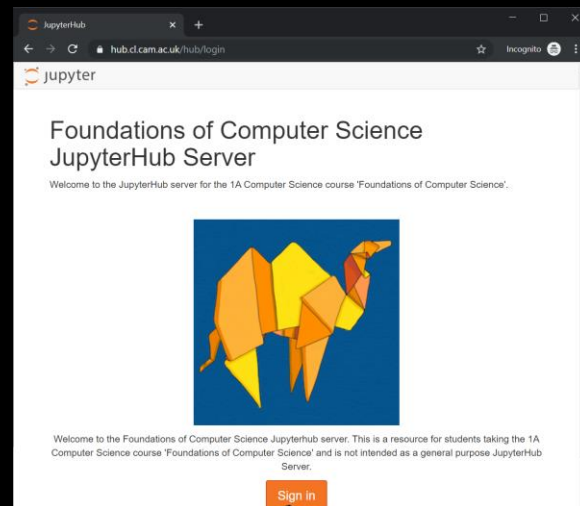
Running Jupyter

The autograder will run wherever you run Jupyter + Python3.

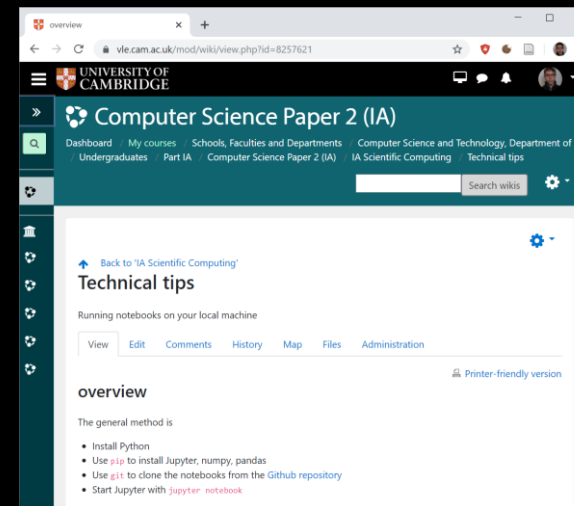
notebooks.azure.com



hub.cl.cam.ac.uk



your own machine



(installation tips on Moodle)

Course: Computer Science Paper 2

https://www.vle.cam.ac.uk/course/view.php?id=141182

UNIVERSITY OF CAMBRIDGE

Dashboard Courses Categories Course History Help About Moodle

Computer Science Paper 2 (IA)

Find a course

Computer Science Paper 2 (IA)

Participants

Grades

Announcements

Hardware Practical Ticking Times - 2018-19

Digital Electronics

Discrete Mathematics

Operating Systems

Software and Security Engineering

ML Vacation Tick

IA Scientific Computing 2017/2018

IA Scientific Computing

My favourite courses

Data Science: Principles and practice

LE49: Probabilistic Machine Learning 2018-19

Computer Science Paper 1 (IA)

IB Foundations of Data Science 2018-19

Computer Science Paper 2 (IA)

Dashboard ► My courses ► Schools, Faculties and Departments ► Computer Science and Technology, Department of ► Undergraduates ► Part IA ► Computer Science Paper 2 (IA)

IA Scientific Computing

Help forum

Library of course notebooks

- Course structure and assessment
- How to run notebooks on Azure
- Printout of the notebooks [pdf]

Technical tips

Running notebooks on your local machine

Notebook submission

To get your tick, you must upload your Jupyter notebooks. This is just a record-keeping requirement, not part of your grade, so please don't waste time tidying your code.

Timetable

- 22 Nov 2018: briefing lecture
- 17, 24 Jan 2019: help sessions
- 30 Jan 2019: autograder completion deadline
- 31 Jan 2019, 2--5pm: ticking session

About the ticking session

- You must have completed the online submission of answers by 00:05am on Thursday 31 January.
- Bring your computer, with your notebooks open and running. (If you don't have your own computer, your notebooks should be open and running on a lab machine).
- The ticker may ask you to re-run your code or make modifications. Make sure your code executes correctly.
- The ticker will ask you to explain your code. You don't need to memorise the exact syntax of all the functions you use, but you do need to be able to explain the purpose and the input and output datatypes.

Latest announcements

Add a new topic...

10 Oct, 11:09
Claire Chapman
Hardware Ticking Group Allocations
29 Jan, 13:24
Damon Wischik
Scientific Computing ticks
15 Jan, 19:15
Damon Wischik
Scientific Computing help sessions & deadline
1 Dec, 22:24
Damon Wischik
Scientific Computing autograder is now open
26 Oct, 07:32
Robert Harle
First hardware Ticking session for group EA Today
Older topics ...

Search forums

Go

[Advanced search](#)

Where do I go for help?

1. StackOverflow
2. Moodle help forum
3. Help sessions at the beginning of Lent term