IA Scientific Computing

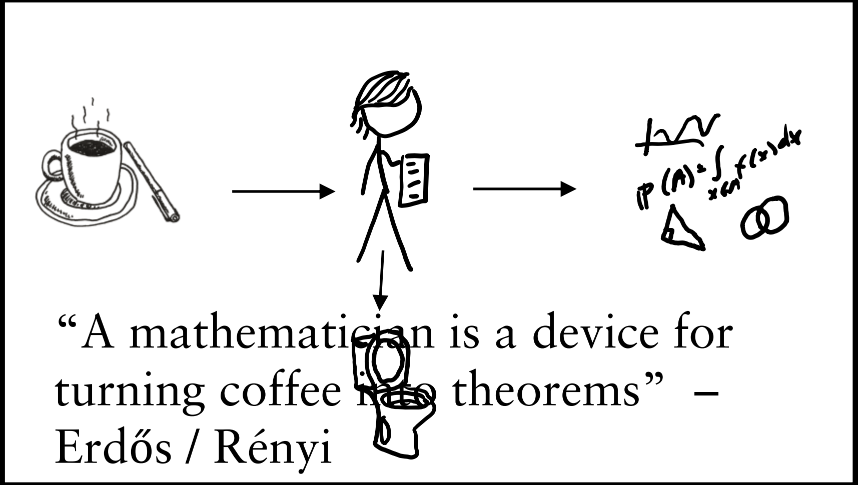
BRIEFING LECTURE

Scientific computing

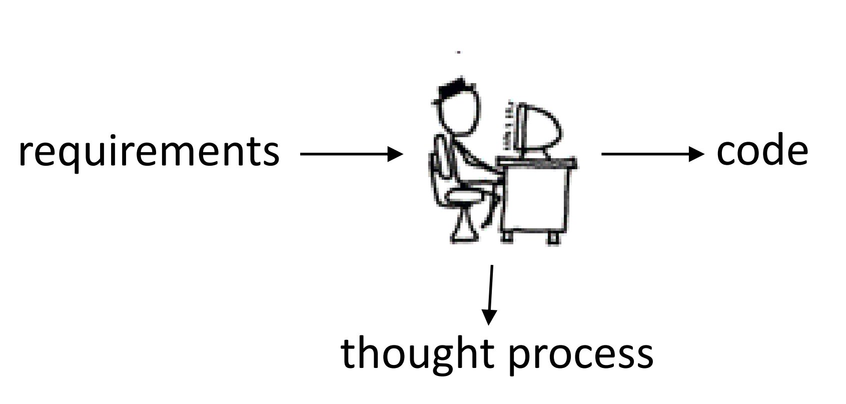
✦ computing as a tool for doing science

Computer science

✦ the study of computation

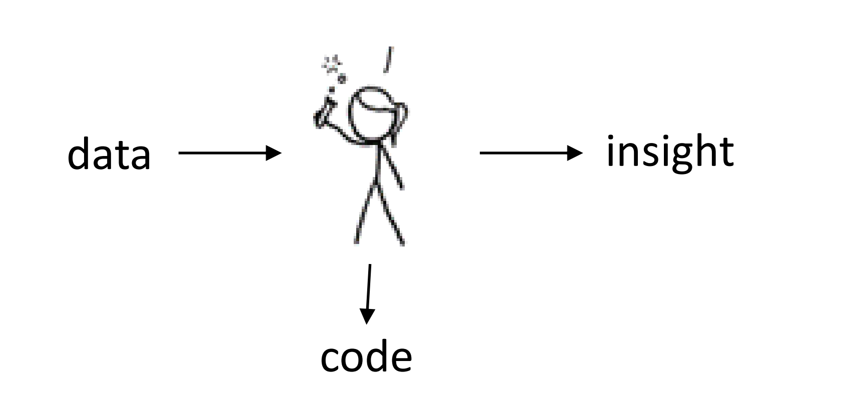


“A mathematician is a device for turning coffee into theorems” – Erdős / Rényi



requirements → code

thought process



data → insight

code

SCIENTIFIC COMPUTING

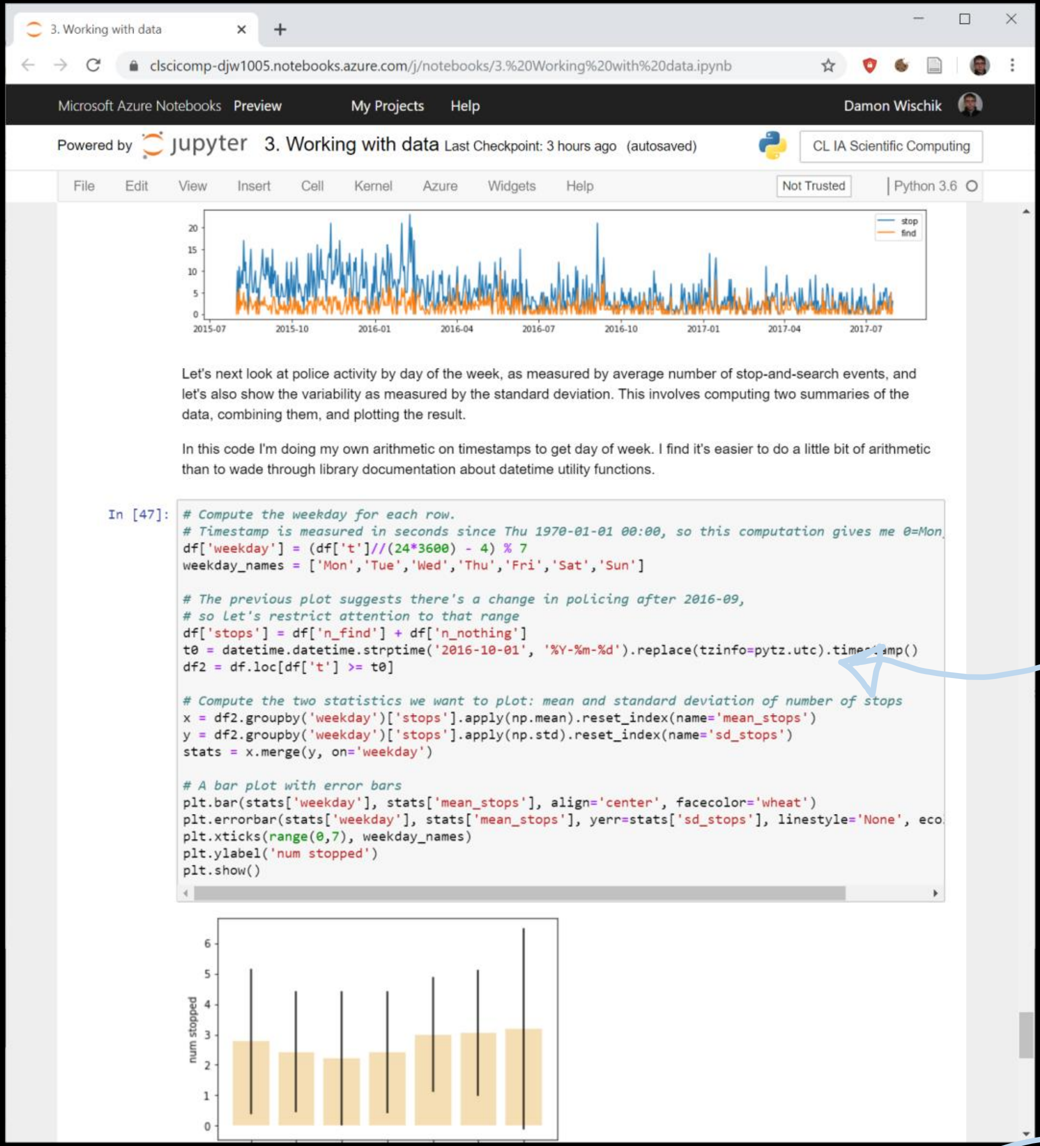
Try out an idea ✦ see what happens ✦ refine your idea ✦ try something else ✦ iterate … ✦ share what you’ve learnt

CODE AT THE SPEED OF THOUGHT

- Concise one- or two-liners for one-off tasks
- Rich, expressive libraries & glue code

Scientific computing

= Python + numpy + plotting + pandas
+ Jupyter notebooks



Let's next look at police activity by day of the week, as measured by average number of stop-and-search events, and let's also show the variability as measured by the standard deviation. This involves computing two summaries of the data, combining them, and plotting the result.

In this code I'm doing my own arithmetic on timestamps to get day of week. I find it's easier to do a little bit of arithmetic than to wade through library documentation about datetime utility functions.

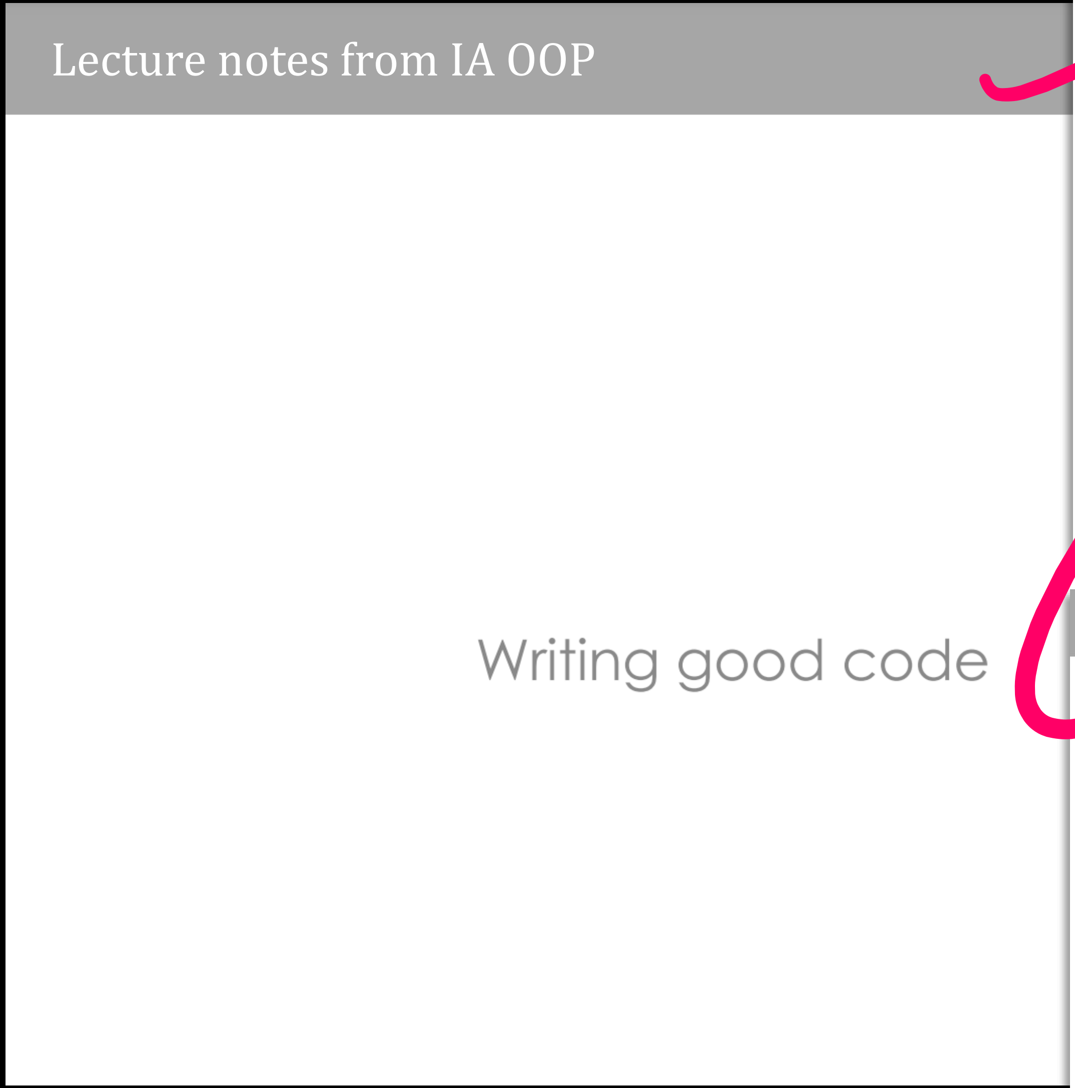
In [47]:

```
# Compute the weekday for each row.
# Timestamp is measured in seconds since Thu 1970-01-01 00:00, so this computation gives me 0=Mon
df['weekday'] = (df['t']//(24*3600) - 4) % 7
weekday_names = ['Mon','Tue','Wed','Thu','Fri','Sat','Sun']

# The previous plot suggests there's a change in policing after 2016-09,
# so let's restrict attention to that range
df['stops'] = df['n_find'] + df['n_nothing']
t0 = datetime.datetime.strptime('2016-10-01', '%Y-%m-%d').replace(tzinfo=pytz.utc).timestamp()
df2 = df.loc[df['t'] >= t0]

# Compute the two statistics we want to plot: mean and standard deviation of number of stops
x = df2.groupby('weekday')['stops'].apply(np.mean).reset_index(name='mean_stops')
y = df2.groupby('weekday')['stops'].apply(np.std).reset_index(name='sd_stops')
stats = x.merge(y, on='weekday')

# A bar plot with error bars
plt.bar(stats['weekday'], stats['mean_stops'], align='center', facecolor='wheat')
plt.errorbar(stats['weekday'], stats['mean_stops'], yerr=stats['sd_stops'], linestyle='None', eco
plt.xticks(range(0,7), weekday_names)
plt.ylabel('num stopped')
plt.show()
```

First I ran this cell
up here

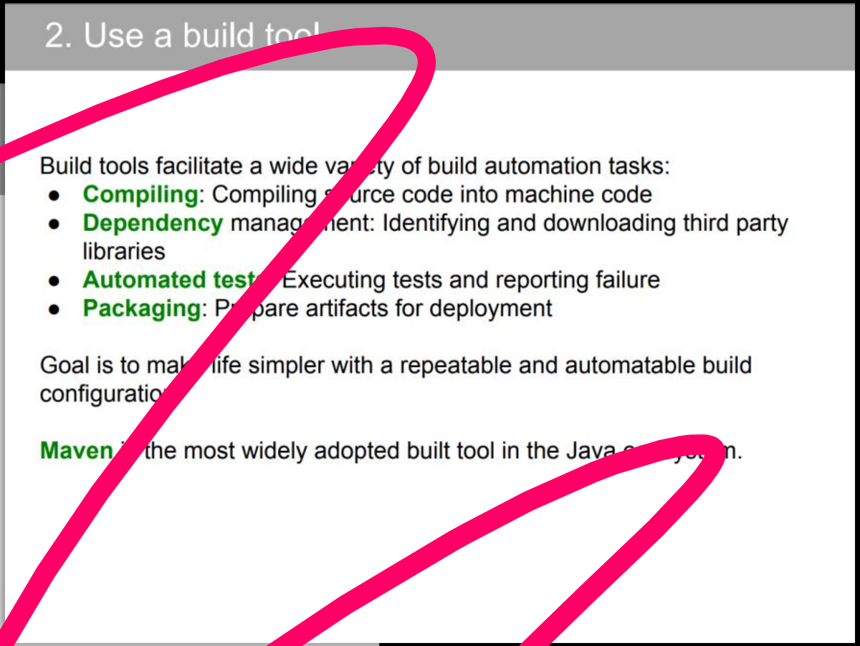
And now this
cell is producing
strange answers

Then this one,
I think.



Lecture notes from IA OOP

Writing good code



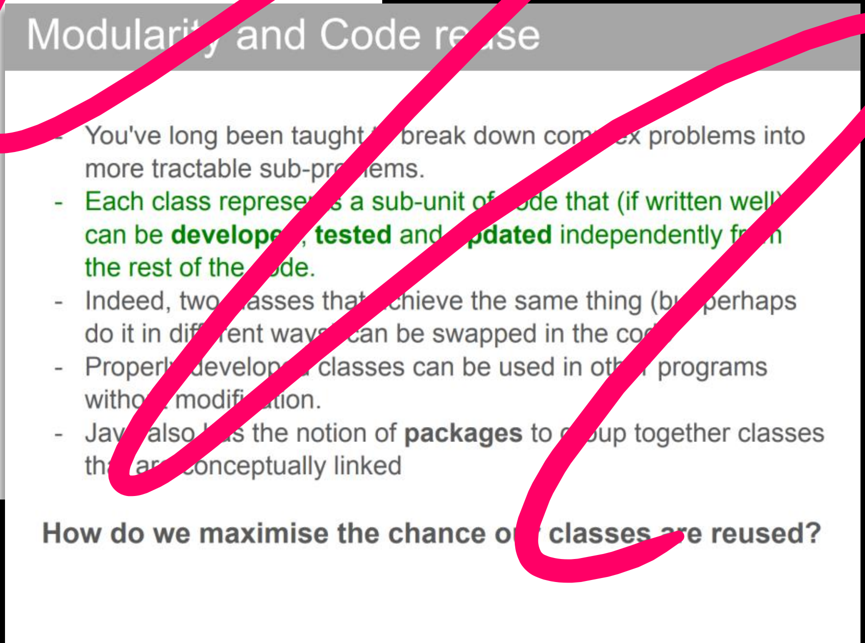
2. Use a build tool

Build tools facilitate a wide variety of build automation tasks:

- **Compiling**: Compiling source code into machine code
- **Dependency** management: Identifying and downloading third party libraries
- **Automated tests**: Executing tests and reporting failure
- **Packaging**: Prepare artifacts for deployment

Goal is to make life simpler with a repeatable and automatable build configuration.

Maven is the most widely adopted built tool in the Java ecosystem.



Modularity and Code reuse

- 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.
- Java also has the notion of **packages** to group together classes that are conceptually linked

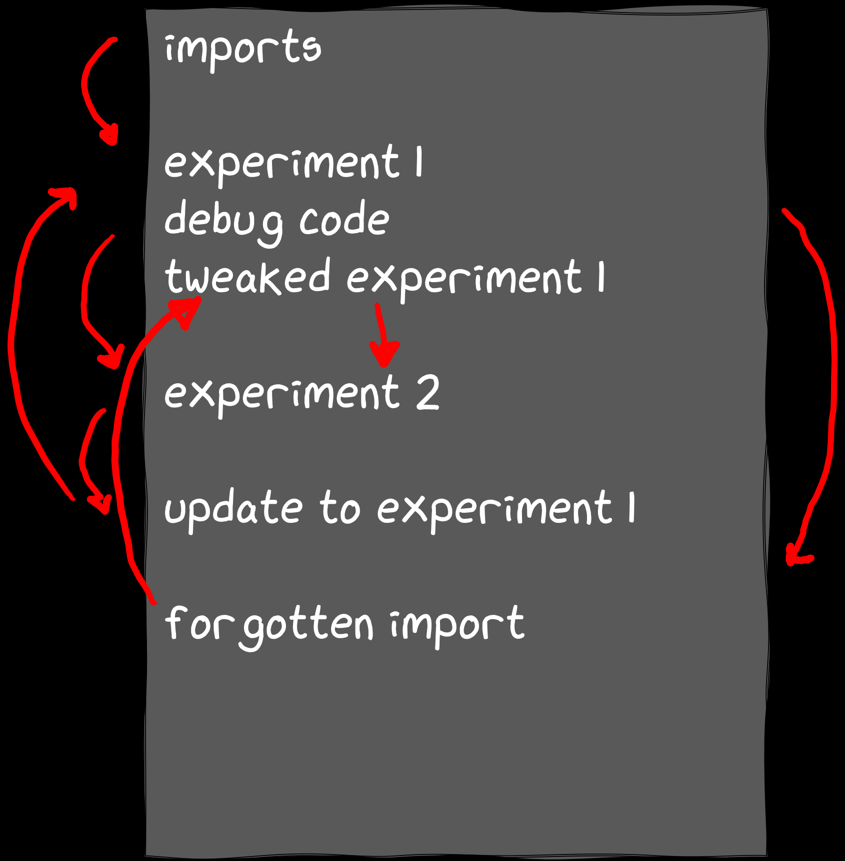
How do we maximise the chance our classes are reused?

Bad advice for scientific computing

Marie Kondo,
de-cluttering guru

Look at each line of your code and ask yourself: ‘does this spark joy?’ If not, delete it.

while working



imports

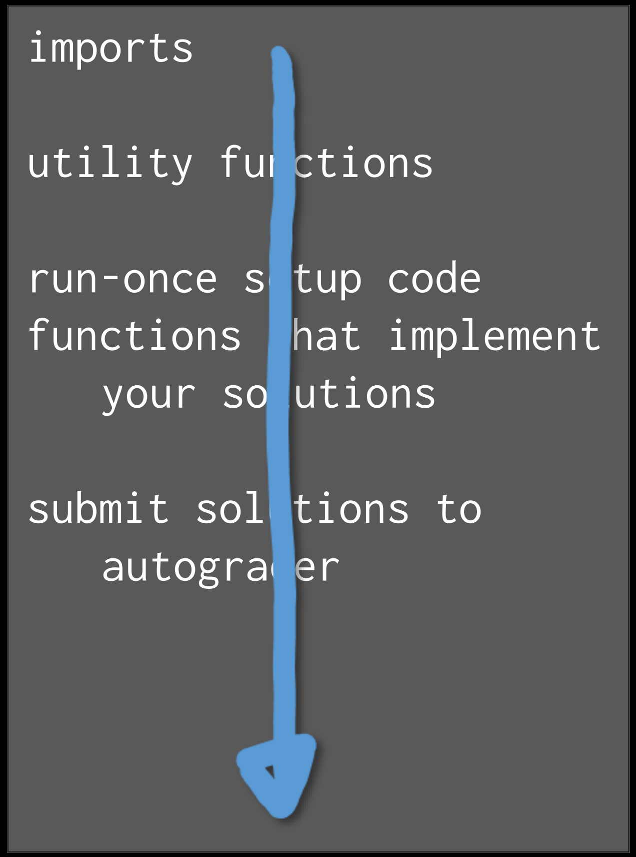
experiment 1
debug code
tweaked experiment 1

experiment 2

update to experiment 1

forgotten import

after you've finished



imports

utility functions

run-once setup code
functions that implement
    your solutions

submit solutions to
    autograder

TUTORIALS

0. Programming in Python
   language quirks

1. Numerical computation
   `numpy`

2. Plotting data
   `matplotlib`

3. Working with data
   pandas

A. Data scraping recipes

ASSESSMENT

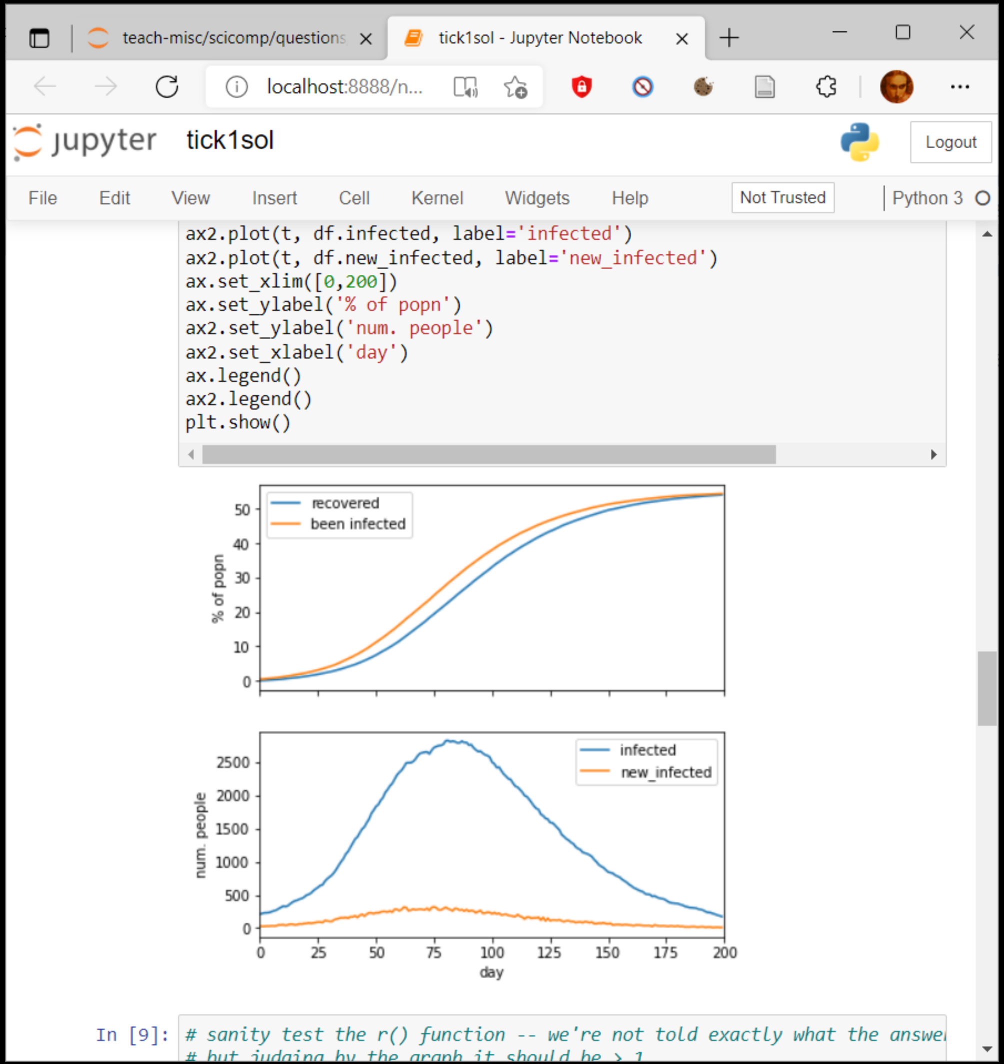
(maths paper mark = 92% exam + 8% Scientific Computing ticks)

No written exam

Four ticks, each marked pass/fail
Ticks 1 and 2: pass the autograder & submit notebook by 22 Jan
Ticks 3 and 4: submit pdfs and notebook by 29 Jan

Some of you will have a viva.

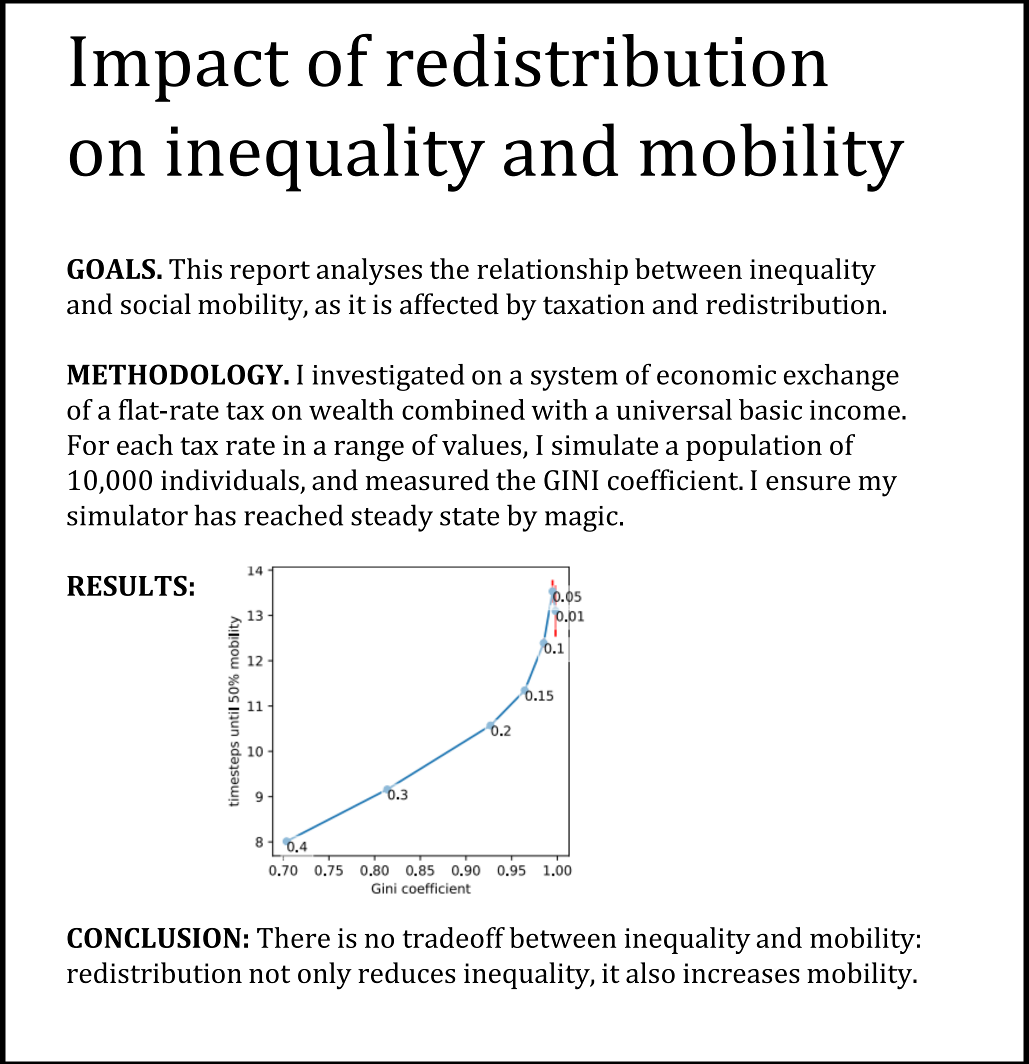
Tick 1,2: Econo-physics simulator (with answers checked by autograder)



```
ax2.plot(t, df.infected, label='infected')
ax2.plot(t, df.new_infected, label='new_infected')
ax.set_xlim([0,200])
ax.set_ylabel('% of popn')
ax2.set_ylabel('num. people')
ax2.set_xlabel('day')
ax.legend()
ax2.legend()
plt.show()
```

```
In [9]: # sanity test the r() function -- we're not told exactly what the answe
        # but judging by the graph it should be > 1
```

Tick 3: plots
Tick 4: One-page scientific report



Impact of redistribution on inequality and mobility

GOALS. This report analyses the relationship between inequality and social mobility, as it is affected by taxation and redistribution.

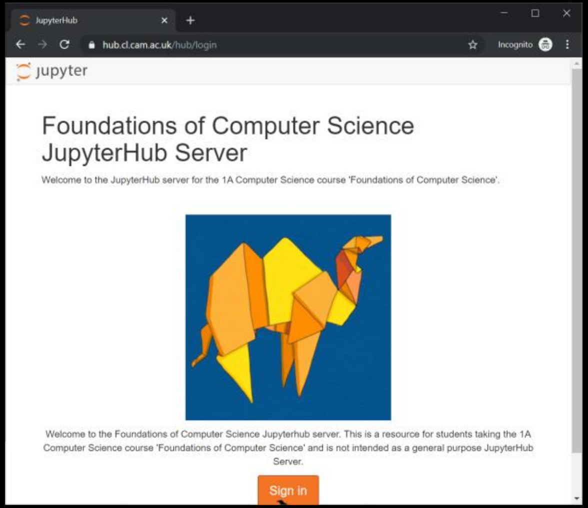
METHODOLOGY. I investigated on a system of economic exchange of a flat-rate tax on wealth combined with a universal basic income. For each tax rate in a range of values, I simulate a population of 10,000 individuals, and measured the GINI coefficient. I ensure my simulator has reached steady state by magic.

RESULTS:

CONCLUSION: There is no tradeoff between inequality and mobility: redistribution not only reduces inequality, it also increases mobility.

The autograder will run wherever you run Python3

hub.cl.cam.ac.uk

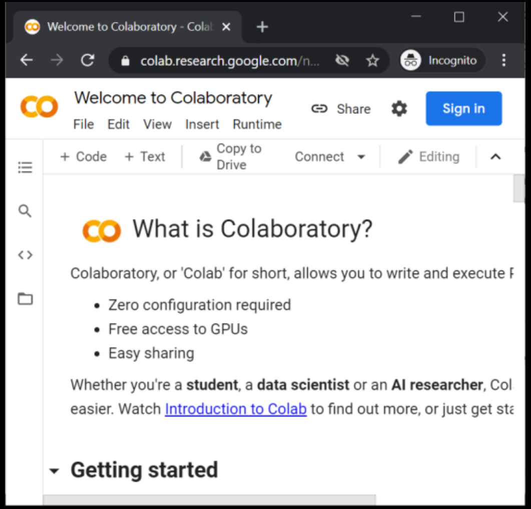


Foundations of Computer Science JupyterHub Server

Welcome to the JupyterHub server for the 1A Computer Science course 'Foundations of Computer Science'.

Welcome to the Foundations of Computer Science Jupyterhub server. This is a resource for students taking the 1A Computer Science course 'Foundations of Computer Science' and is not intended as a general purpose JupyterHub Server.

Google colab



What is Colaboratory?

Colaboratory, or 'Colab' for short, allows you to write and execute F

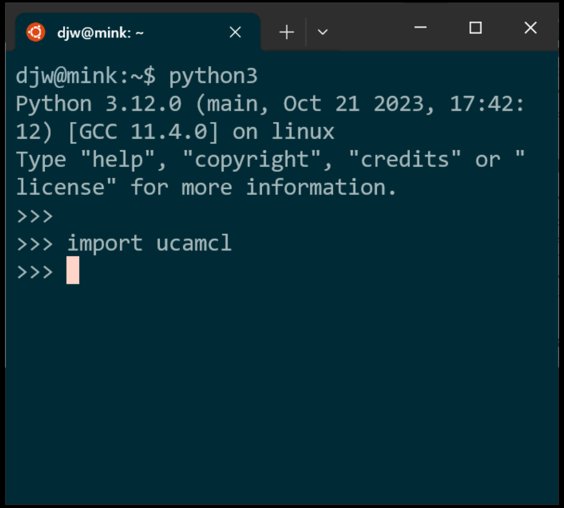
- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Col easier. Watch Introduction to Colab to find out more, or just get sta

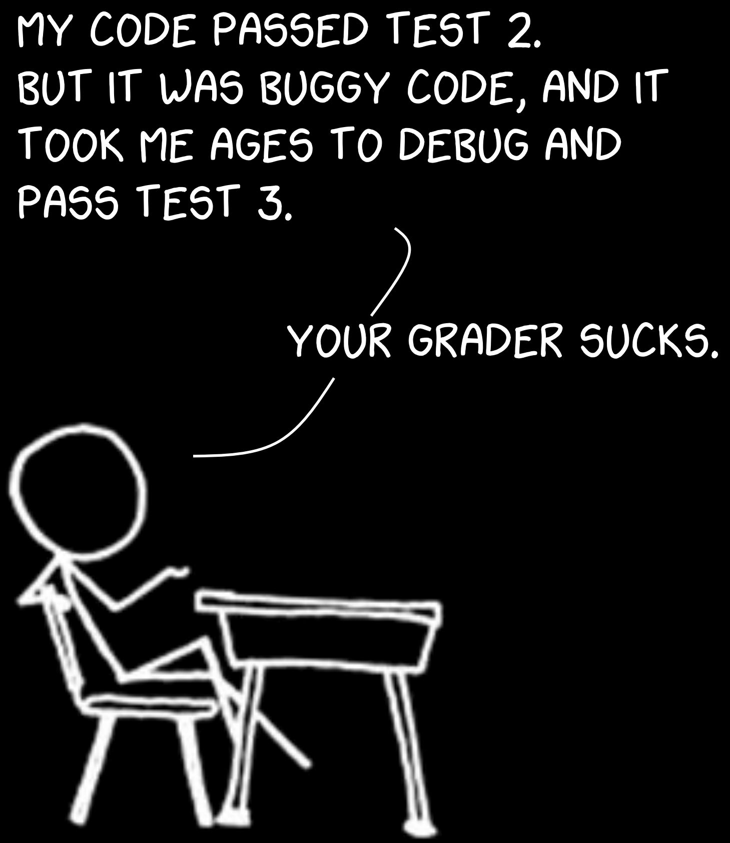
Getting started

VSCode

Command line



```
djw@mink:~$ python3
Python 3.12.0 (main, Oct 21 2023, 17:42:
12) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "
license" for more information.
>>>
>>> import ucamcl
>>> 
```


MY CODE PASSED TEST 2.
BUT IT WAS BUGGY CODE, AND IT TOOK ME AGES TO DEBUG AND PASS TEST 3.
YOUR GRADER SUCKS.

Scientific computing isn’t about meeting requirements, it’s about discovery.

✦ chart your own path ✦ write your own tests

✦ invent a few small test cases ✦ work them out with pen and paper ✦ make sure your code agrees

Can I use ChatGPT?

HM, HOW DO SCINTIC COMPUTING PROBLEM?
HM, HOW DO' E SICI COMUTING PReBLEM?
HM, HOW DO SOLVE THIS ENTHSC SCOIBUTICS SCIENTIC C COMPTENG = ?
WOW, AI IS LIKE A TUTOR!!
WOW, - Chat G(F?) - Chat G(s) - Chat
THIS = HOMEWORK LIKE A TUTOR!!

Can I use ChatGPT?

Yes, feel free.

Can I use ChatGPT to save me time and effort?

Unlikely.

Can I use ChatGPT to sharpen my thinking?

Yes !!!👍🔥👍🔥👍

Help and support

- Moodle help forum
- Helpdesk sessions early in Lent term
- Optional hints-and-tips lecture early in Lent term