

WELCOME

PROGRAMMING WITH PYTHON

Lucy Whalley / l.whalley@northumbria.ac.uk.

lucydot.github.io/slides

lucydot.github.io/python_novice

HI! I'M LUCY

- University of Birmingham: MSci Theoretical Physics
- Birmingham City University: PGCE Mathematics
- Imperial College London: PhD in Materials Science
(with the CDT-PV)
- Imperial College London: Research Associate in
Solar Cells
- University of Northumbria: Vice-Chancellor's Fellow

HI! I'M LUCY

- University of Birmingham: C++ / Python
- Birmingham City University: HTML / static websites
- Imperial College London: High-performance computing / Software Carpentry / Fellow @ Software Sustainability Institute
- University of Northumbria: ???

WORKSHOP OUTLINE

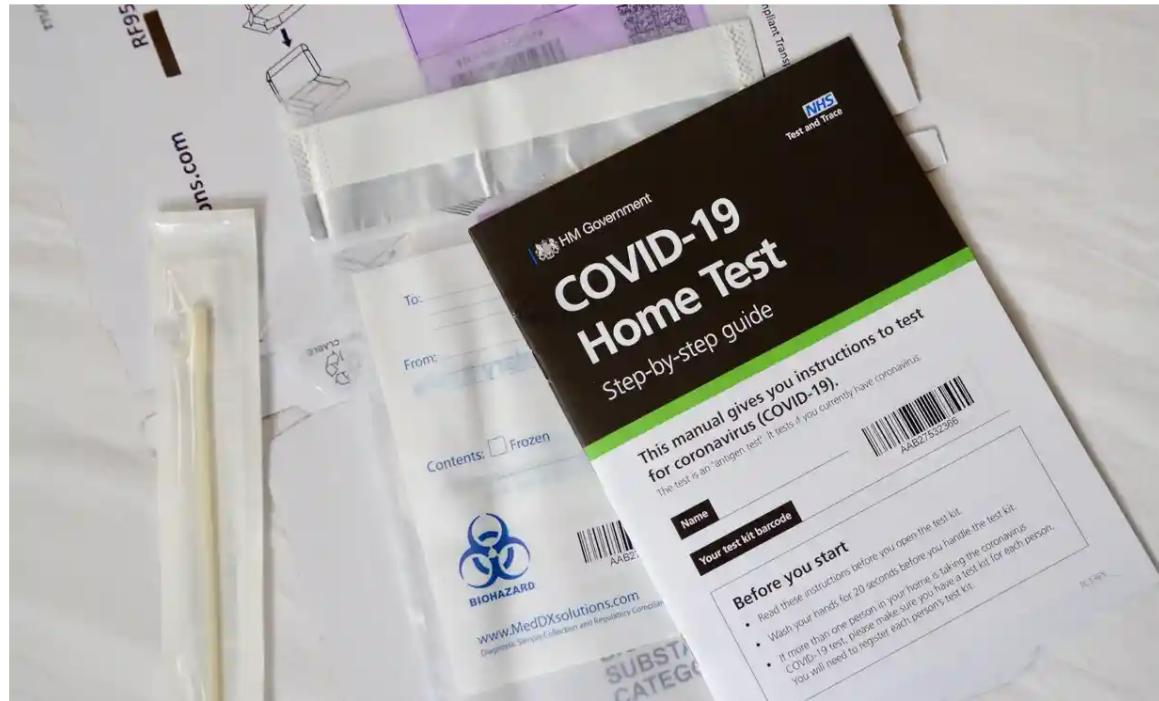
- Day One: Python basics (variables, data types, for loops, conditionals, functions)
- Day Two: Data analysis and plotting

...but why is this important?

Covid: how Excel may have caused loss of 16,000 test results in England

Public Health England data error blamed on limitations of Microsoft spreadsheet

- [Coronavirus - latest updates](#)
- [See all our coronavirus coverage](#)



WHY PROGRAMMING? REPRODUCIBILITY

Reproducibility is a major principle of the scientific method. It means that a result obtained by an experiment or observational study should be achieved again with a high degree of agreement when the study is replicated with the same methodology by different researchers.

WHY PROGRAMMING? REPRODUCIBILITY

The work below relates to sections **Hot polaron states** and **Carrier cooling: heat transfer to the lattice** of the 2017 publication *Slow cooling of hot polarons in halide perovskite solar cells*. The notebook was written by Lucy D. Whalley.

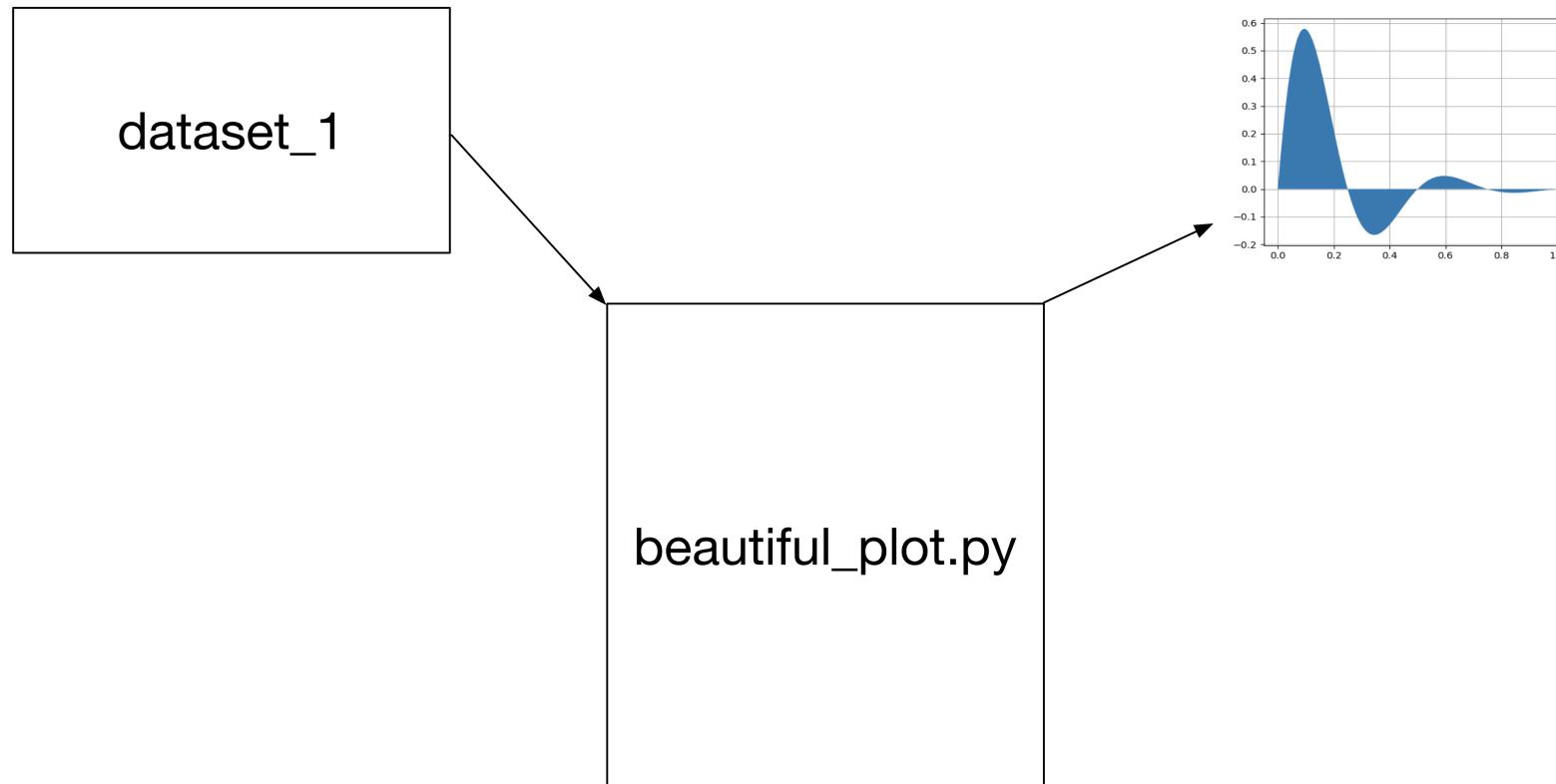
```
# import libraries
import math
import scipy
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import scipy.integrate as integrate
%matplotlib inline

# constants in SI
m = 9.10938291E-31 # mass of electron
q = 1.60217657E-19 # charge of electron
epsilon_0 = 8.85418782E-12 # permittivity of free space
h = 6.62606957E-34 # plancks magical number
hbar = 1.054571800E-34 # reduced planck
Rydberg = 2.1787E-18 # Rydberg energy (in joules)
avogadro = 6.02214179E23

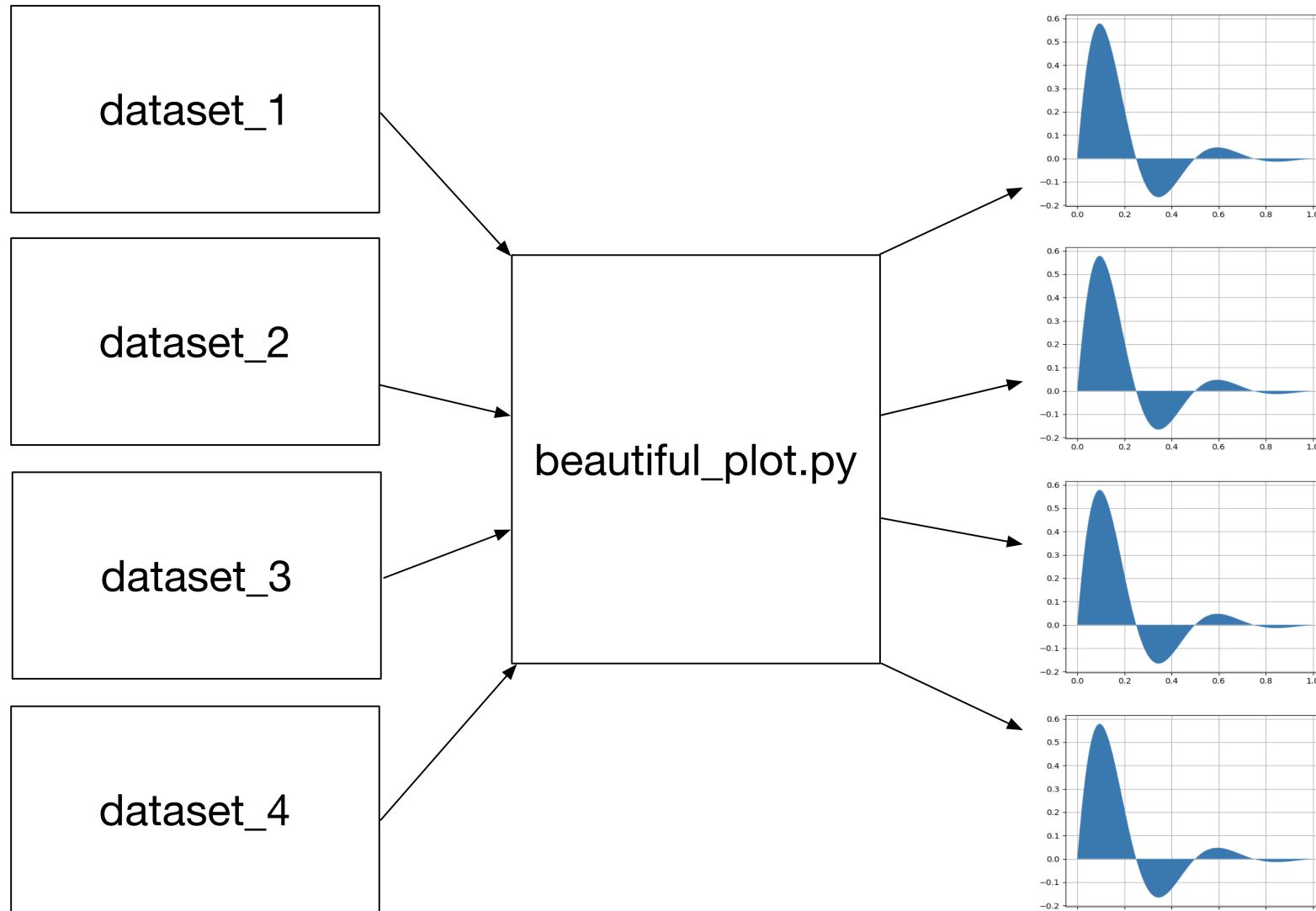
# units in SI
Angstrom = 1E-10 # m to angstrom
k = 1.38064852E-23
```

research paper: [Slow Cooling of Hot Polarons in Halide Perovskite Solar Cells](#)
analysis code: github.com/WMD-group/hot-carrier-cooling

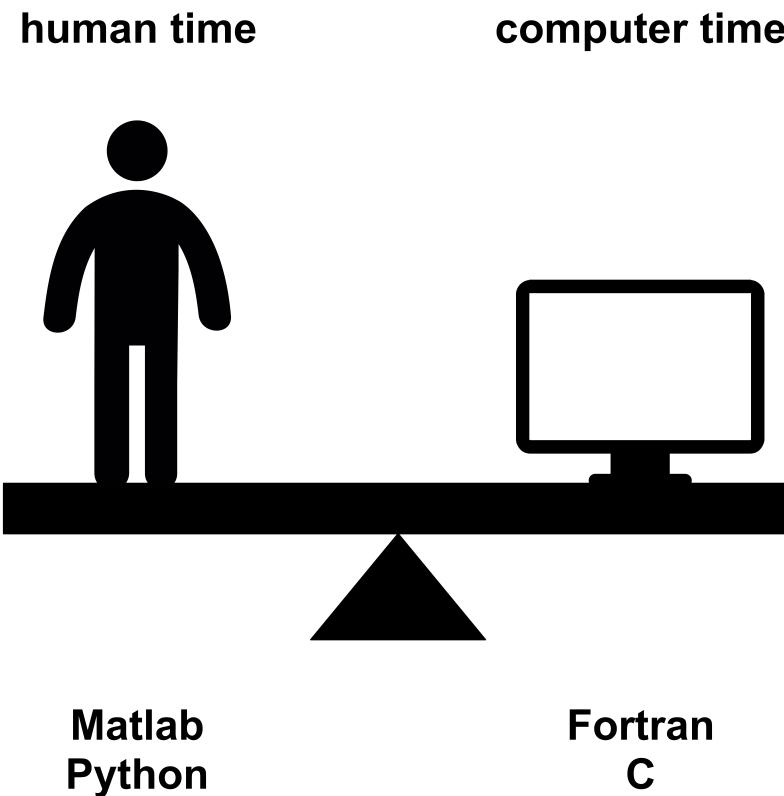
WHY PROGRAMMING? REPEATABILITY



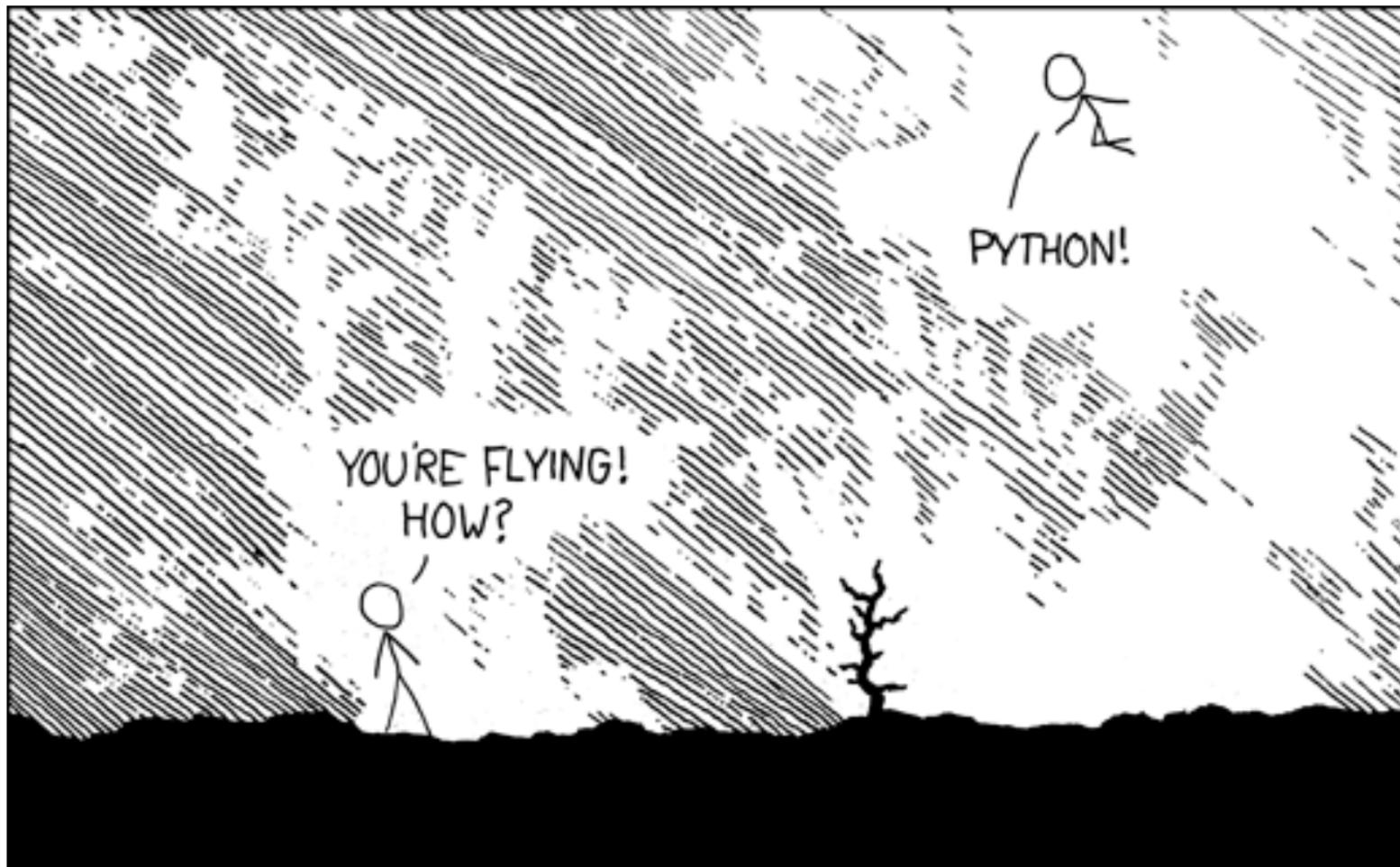
WHY PROGRAMMING? REPEATABILITY



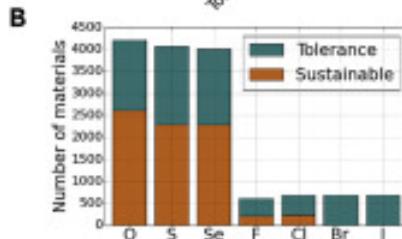
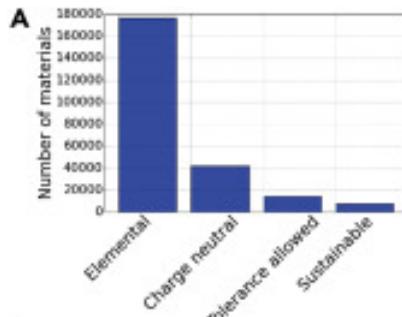
WHY PYTHON? THE TRADE-OFF



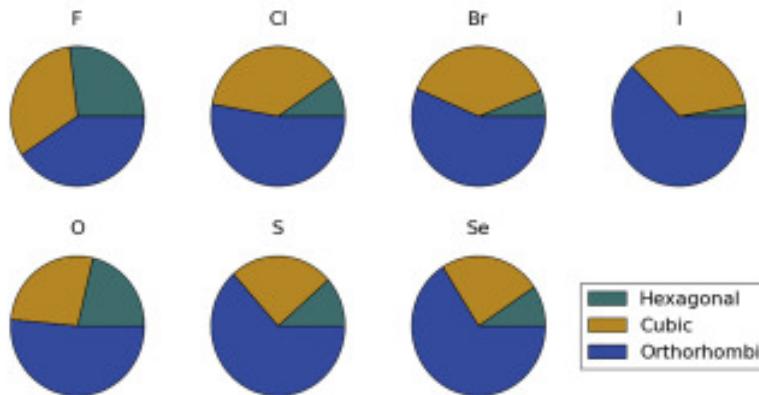
WHY PYTHON? IT GIVES YOU WINGS



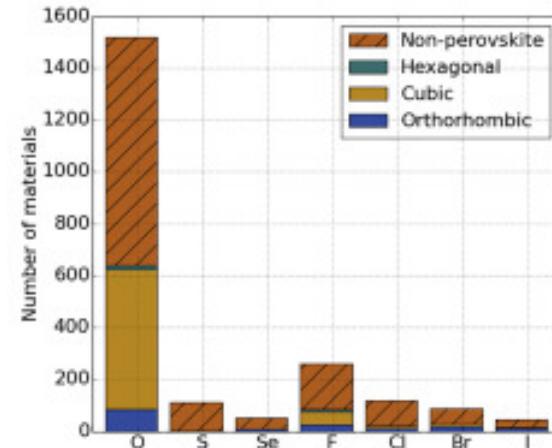
WHY PYTHON? PUBLICATION-READY PLOTS



C



D



Computational Screening of All Stoichiometric Inorganic Materials, D.W. Davies et al.

WHY PYTHON?

- readable
- free to use
- cross-platform
- well documented
- widely used

TODAY'S OUTLINE

1. running python code
2. variables
3. data types
4. functions, help and errors
5. lists
6. for loops
7. if statements

PLAIN TEXT VS. JUPYTER NOTEBOOK

- *Plain text approach:*
 - write code in a text editor
 - save with a `.py` extension
 - run code using a terminal
- *Jupyter notebook approach:*
 - write code in a jupyter notebook
 - run code in a jupyter notebook
 - save with a `.ipynb` extension

SETUP (THUMBS UP WHEN COMPLETE)

- Add your name to the etherpad
<https://etherpad.wikimedia.org/p/cdt-renu>
- Open your Jupyter Notebook (open your terminal / git bash --> jupyter notebook)
- Arrange your screen so you can see my screen (via Teams) and your Jupyter Notebook at the same time

TASK (5 MIN)

Use your Jupyter notebook to...

- link to your favourite webpage
- calculate $3624357/325$
- make a bullet pointed ToDo list with heading "ToDo list"

--> Thumbs up when you're done please

a = 65

- letters, digits and _
 - cannot start with a digit
 - _ start has special meaning
 - case sensitive

TASK (2 MIN)

Can you predict what the final value of position is
for the code block below? --> Vote at
<http://etc.ch/qoHR>.

```
initial = 'left'  
position = initial  
initial = 'right'
```

TASK (2 MIN)

What do you think the following code will print? -->

Vote at <http://etc.ch/qoHR>.

```
first = 1
second = 5*first
first=2
print('first is', first, 'and second is', second)
```

DATA TYPES

Data type	Python name	Definition	Example
integer	int	positive or negative whole numbers	-256
float	float	real number	-3.16436
string	str	character string	"20 pence."
list	list	a sequence of values	['frog', 2, 8]

+ boolean, dict, tuple, complex, None, set

TASK (2 MIN)

Which of the following will print 2.0? --> Vote at
<http://etc.ch/qoHR>.

```
first = 1.0
second = "1"
third = "1.1"
```

1. first + float(second)
2. float(second) + float(third)
3. first + int(third)
4. first + int(float(third))
5. int(first) + int(float(third))
6. 2.0 * second

TODAY'S OUTLINE

1. **running python code:** Jupyter Notebooks, markdown basics
2. **variables:** variable names, variable assignment, `print()`, execution order
3. **data types:** integer, float, string, list, `len()`, string operations/indexing/slicing, type conversion: `int()`, `str()`, `float()`
4. **functions, help and errors:** `min()`, `max()`, `round()`, `help()`, runtime errors (exceptions), syntax errors
5. **lists**
6. **for loops**
7. **if statements**

LISTS

Data type	Python name	Definition	Example
integer	int	positive or negative whole numbers	-256
float	float	real number	-3.16436
string	str	character string	"20 pence."
list	list	a sequence of values	['frog', 2, 8]

FOR LOOPS

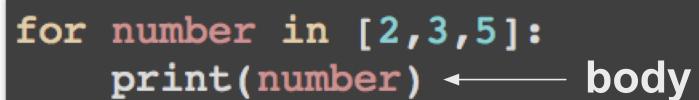
```
print(2)
print(3)
print(5)
```

```
for number in [2,3,5]:
    print(number)
```

FOR LOOPS

```
print(2)  
print(3)  
print(5)
```

loop variable sequence



```
for number in [2,3,5]:  
    print(number) ← body
```

PAIRS TASK (3 MIN)

I want to sum the integers from 1 to 10. What is wrong with this code? How can I fix it? --> Thumbs up when you are ready to share your answer.

```
total = 0
for number in range(10)
    total = total + number
print(total)
```

CONDITIONALS

```
mass = 4.2

if mass > 3:
    print(mass, ' is large')

if mass < 2:
    print(mass, ' is small')

if 2 <= mass <= 3:
    print(mass, ' is just right')
```

PAIRS TASK (3 MIN)

What is wrong with the code? Fix the code so that it works as intended -> Thumbs up when you are ready to share your answer.

```
grade = 95

if grade >= 70:
    print("grade is C")
elif grade >= 80:
    print("grade is B")
elif grade >= 90:
    print("grade is A")
```

SUMMARY

1. **running python code:** Jupyter Notebooks, markdown basics
2. **variables:** variable names, variable assignment, `print()`, execution order
3. **data types:** integer, float, string, list, `len()`, string operations/indexing/slicing, type conversion: `int()`, `str()`, `float()`
4. **functions, help and errors:** `min()`, `max()`, `round()`, `help()`, runtime errors (exceptions), syntax errors
5. **lists:** sequence type, immutable vs mutable, list method `append`, `del`
6. **for loops:** dummy variable, loop syntax, index from 0
7. **if statements:** if, elif, else, ordering

These slides available at: lucydot.github.io/slides

Workshop materials are available at: lucydot.github.io/python_novice

Back tomorrow at 9am for Part Two.

WELCOME

PROGRAMMING WITH PYTHON, DAY TWO

Lucy Whalley / l.whalley@northumbria.ac.uk

lucydot.github.io/slides

lucydot.github.io/python_novice

TODAY'S OUTLINE

1. functions
2. variable scope
3. libraries
4. cleaning data with pandas
5. analysing data with numpy
6. plotting data with matplotlib
7. running code as a Python script
8. programming good practice

FUNCTIONS

```
def print_greeting():
    print ("Hello!")
```

FUNCTIONS

```
def print_personalised_greeting(name):  
    print ("Hello "+name)
```

TASK (2 MIN)

Fill in the blanks to create a function that takes a list of numbers as an argument and returns the first negative value in the list --> Vote at <http://etc.ch/qoHR>

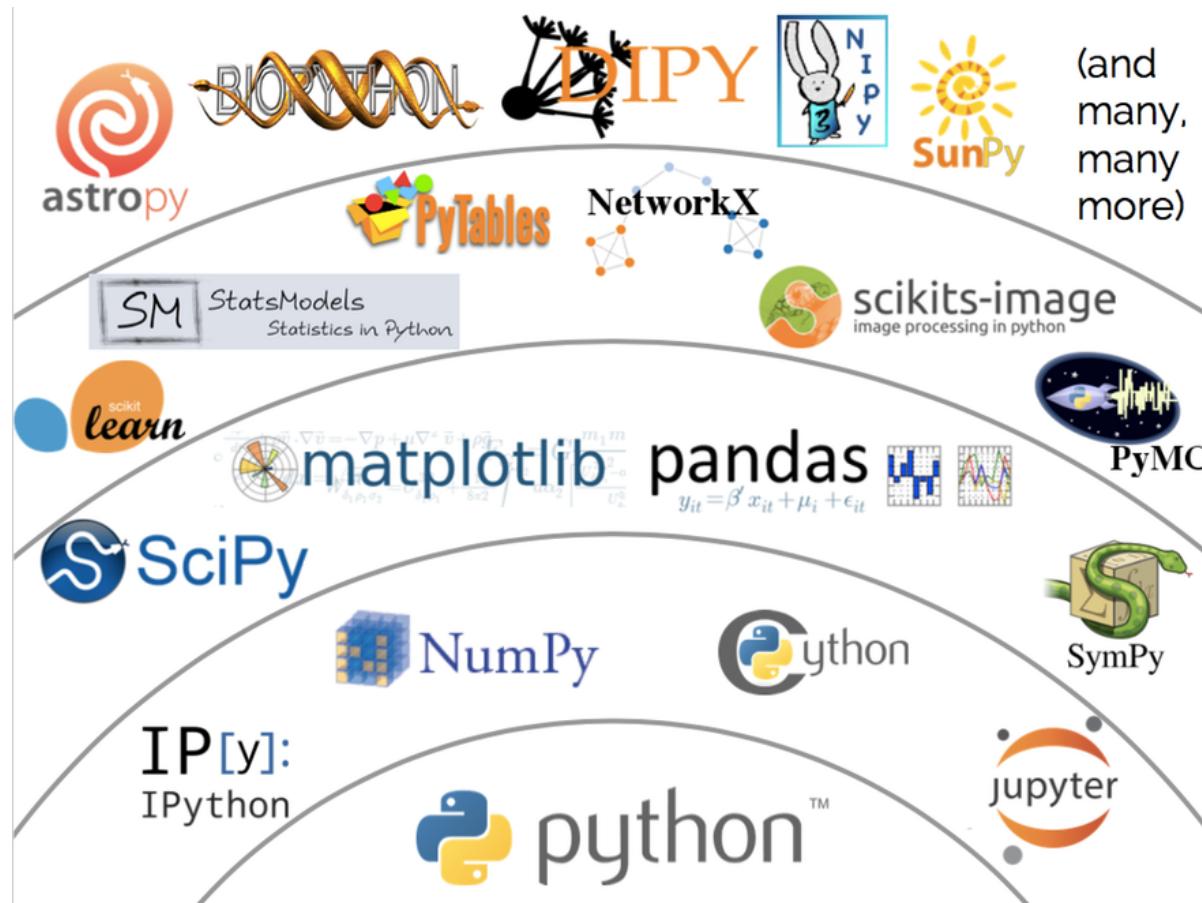
```
def first_negative(values):
    for v in ____:
        if ____:
            return ____
```

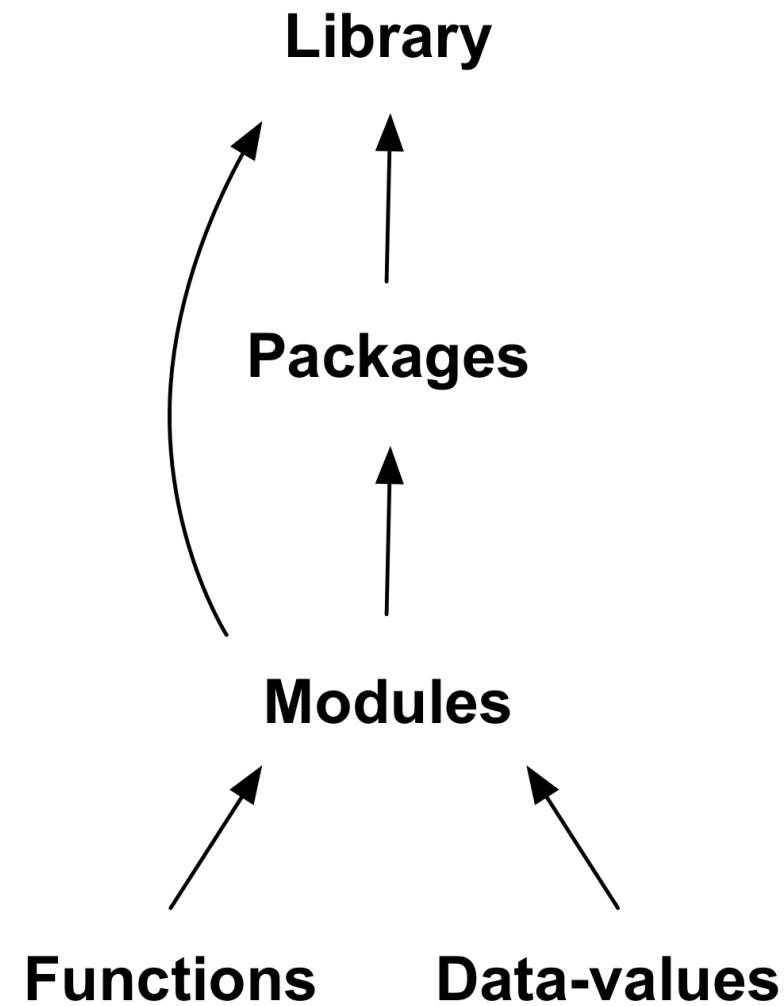
VARIABLE SCOPE

```
pressure = 103.9

def adjust(temperature):
    new_temperature = temperature*1.43/pressure
```

PYTHON SCIENTIFIC LIBRARIES





PAIRS TASK (15 MIN)

You want to select a random character from a string.

```
base = "ATCHAGHRASG"
```

1. which standard library module could help you?
2. which function could you select from that module?
3. try to write a program that uses that function

Feel free to look online (search for "Python standard library") -> Thumbs up when you are ready to share your answer.

TODAY'S OUTLINE

1. **functions:** function syntax, return statement, parameters and arguments
2. **variable scope:** local and global variables
3. **libraries:** modules, packages, libraries, import statements, aliases
4. **cleaning data with pandas:**
5. **analysing data with numpy:**
6. **plotting data with matplotlib:**
7. **running code as a Python script:**
8. **programming good practice:**

INDEXING ARRAYS

0 1 2
↓ ↓ ↓
`data = [[A, B, C], [D, E, F], [G, H, I]]` ← 0
 |← 1
 |← 2

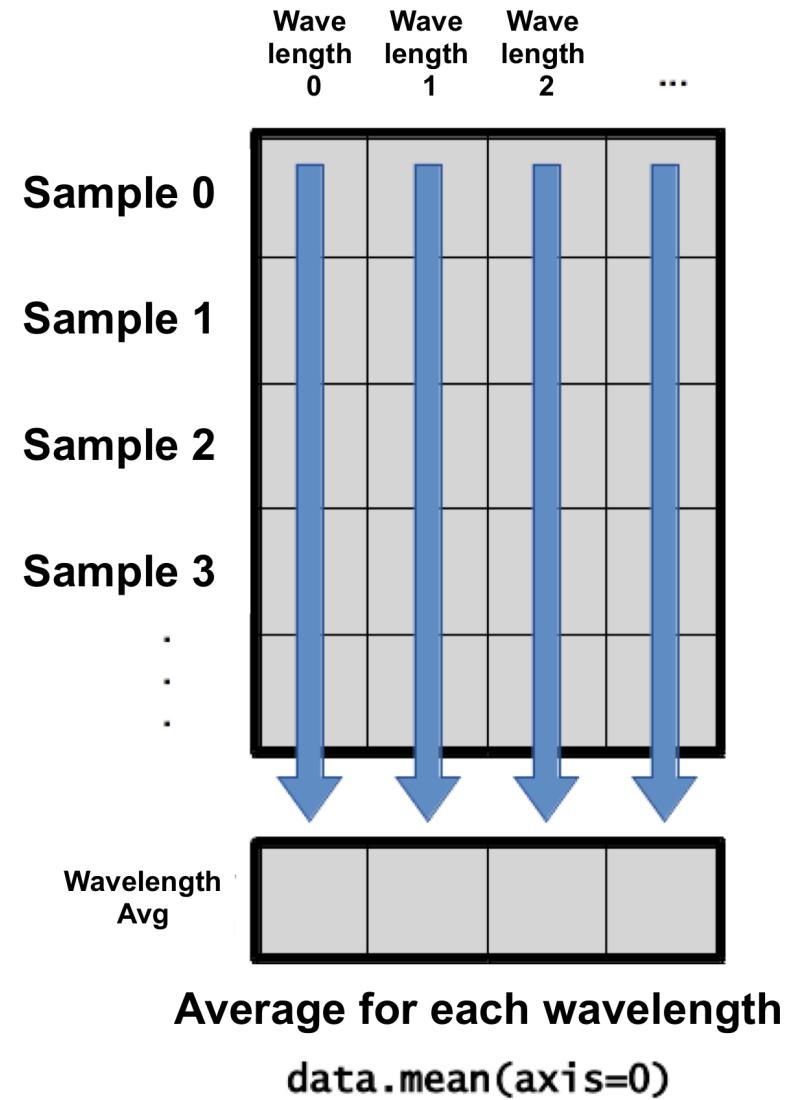
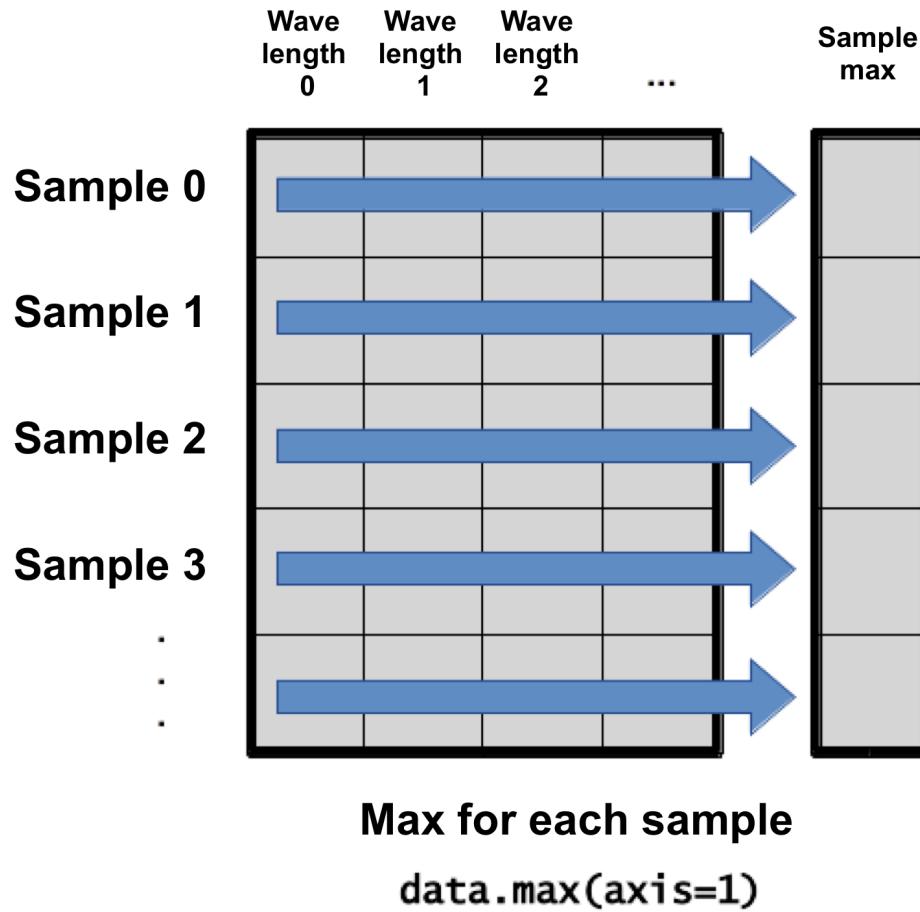
`data[0][0] = A` `data[1][0] = D` `data[2][0] = G`
`data[0][1] = B` `data[1][1] = E` `data[2][1] = H`
`data[0][2] = C` `data[1][2] = F` `data[2][2] = I`

TASK (2 MIN)

Crack the code -> Thumbs up when you are ready to share your answer.

```
letters = np.array([[r,y,c,t],  
                   [u,o,x,e],  
                   [d,p,i,n]])  
  
letters[0,2] letters[2,0] letters[0,3] -  
letters[0,0] letters[1,3] letters[3,3] letters[0,1]
```

OPERATIONS ACROSS AXES



PUTTING IT ALL TOGETHER

bit.ly/plot_example

PROGRAMMING GOOD PRACTICE

Follow standard Python style

- focus on readability (consistency is key!)
- maximum line length of 79 characters
- whitespace:

```
spam(ham[1], {eggs: 2})
```

```
spam( ham[ 1 ] , { eggs: 2 } )
```

- clear, meaningful variable names

PROGRAMMING GOOD PRACTICE

Use assertions to check for errors

```
def calc_bulk_density(mass, volume):  
    assert volume > 0  
    return mass / volume
```

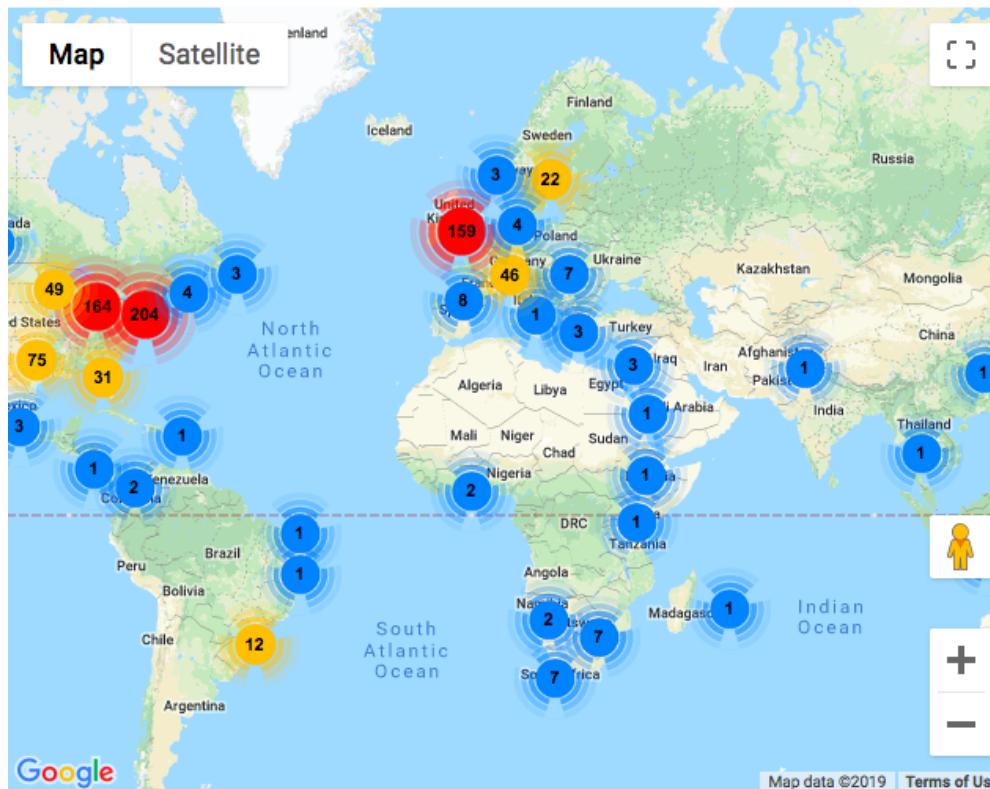
PROGRAMMING GOOD PRACTICE

Document your code with docstrings

```
def calc_bulk_density(mass, volume):
    "Return dry bulk density = powder mass / powder volume."
    assert volume > 0
    return mass / volume
```

PROGRAMMING GOOD PRACTICE

Use version control - attend a git SWC



software-carpentry.org

SUMMARY

1. **functions:** function syntax, return statement, parameters and arguments
2. **variable scope:** local and global variables
3. **libraries:** modules, packages, libraries, import statements, aliases
4. **cleaning data with pandas:** `pandas.read_csv`, `DataFrames`, `pandas.to_csv`
5. **analysing data with numpy:** `numpy.loadtxt`, N-dimensional arrays, attributes
6. **plotting data with matplotlib:** `%matplotlib inline`, `plot()`, `xlabel()`,
`ylabel()`, `show()`, `savefig()`
7. **running code as a Python script:** `%%writefile filename.py`, `python3 filename.py`
8. **programming good practice:** Python style, assert statements, docstring

1. **running python code:** Jupyter Notebooks, markdown basics
2. **variables:** variable names, variable assignment, `print()`, execution order
3. **data types:** integer, float, string, list, string operations/indexing/slicing, type conversion
4. **functions, help and errors:** `min()`, `max()`, `round()`, `help()`, errors
5. **lists:** sequence type, immutable vs mutable, list method `append`, `del`
6. **for loops:** dummy variable, loop syntax, index from 0
7. **if statements:** if, elif, else, ordering
8. **functions:** function syntax, return statement, parameters and arguments
9. **variable scope:** local and global variables
10. **libraries:** modules, packages, libraries, import statements, aliases
11. **cleaning data with pandas:** `pandas.read_csv`, `DataFrames`, `pandas.to_csv`
12. **analysing data with numpy:** `numpy.loadtxt`, N-dimensional arrays, attributes
13. **plotting data with matplotlib:** `plot()`, `xlabel()`, `show()`, `savefig()`
14. **running code as a Python script:** `%%writefile filename.py`, `python3 filename.py`
15. **programming good practice:** Python style, `assert` statements, docstring

Before you hang up, please complete the workshop
survey (link in etherpad -

<https://etherpad.wikimedia.org/p/cdt-renu>)

=====Thank-you=====