Summer 2022 Internships

ISG introduction to coding with Noteable

EDINA

THE UNIVERSITY of EDINBURGH

Noteable™

Summer 2022 Interns Workshop

Introduction to coding with Noteable

Welcome to the session ISG Summer Interns !

This session is hosted by **EDINA** and the **Learning, Teaching and Web** sections of ISG, the University of Edinburgh

In this workshop session you will

- Learn about the history of coding
- Hear from EDINA and LTW how coding at UoEdinburgh has evolved
- Learn about the Python programming language and the Noteable service to access programming environments
- Play and explore coding concepts during the workshop and learn how to access examples and activities

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

A brief history of code – the language for computers (and cookbooks)

- Humans speak in and with natural language, computers in binary and with transistors
- Coding has a long history of building recipes to ‘make stuff’ and codify it into actions that the coder or others can use
- A computer can follow a recipe as ‘Stubborn Chef’
- Modern computers mix languages, functions, inputs, outputs and more

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

Early days of modern computation – the 19th Century

- **1801:** Joseph Jacquard, French merchant invents a loom that uses punched wooden cards to automatically weave fabric designs. Early computers would use similar punch cards.

- **1821:** English mathematician Charles Babbage conceives of a steam-driven calculating machine that would be able to compute tables of numbers. Funded by the British government, the project, called the "Difference Engine" fails due to the lack of technology at the time.

1848: Ada Lovelace, English mathematician and daughter of poet Lord Byron, writes world's first computer program. Lovelace writes step-by-step description for computation of Bernoulli numbers with Babbage's machine.

1890: Herman Hollerith designs a punch-card system to help calculate the 1890 U.S. Census.

Modern computation – the 20th Century

- **1936:** Alan Turing, a British scientist and mathematician, presents the principle of a universal machine, later called the Turing machine, in a paper called "On Computable Numbers…"
- **1950s:** Development of early programming language such as COBOL and Fortran
- **1968:** Douglas Engelbart reveals a prototype of the modern computer in San Francisco. His presentation, called "A Research Center for Augmenting Human Intellect" includes a live demonstration of his computer, including a mouse and a graphical user interface (GUI)
- **1985**: As a response to the Apple Lisa's GUI, Microsoft releases Windows in November 1985
- **1999:** Wi-Fi, the abbreviated term for "wireless fidelity" is developed, initially covering a distance of up to 100 meters

The 21st Century

- De-centralisation of the internet and spread of computer operating systems to host browser-based internet
- In 2003 AMD's Athlon 64, the first 64-bit processor for personal computers, is released to customers
- In 2007, Apple releases the iPhone
- Python 3 released in 2008
- 2022: 700+ programming languages, global IT industry worth over £4 trillion

From the moon to your laptop

- Code can help humans monitor, design, build and execute ideas
- Lovelace's Algorithm & the rise of modern computers – all 1s and 0s
- Coding makes machine language readable by humans
- Different languages for different uses

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

Unlocking the language of computers

- Compiler – computer software that translates source code written in a high-level language (C++) into a set of machine-language instruction

- Language: a system of conventional spoken or written symbols by means of which human beings express themselves as members of a social group and participants in its culture

- Programming: the process of developing and implementing various sets of instructions to enable a computer to achieve a certain task

- Computation: to cause the instruction in (a programme) to be carried out, to execute

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

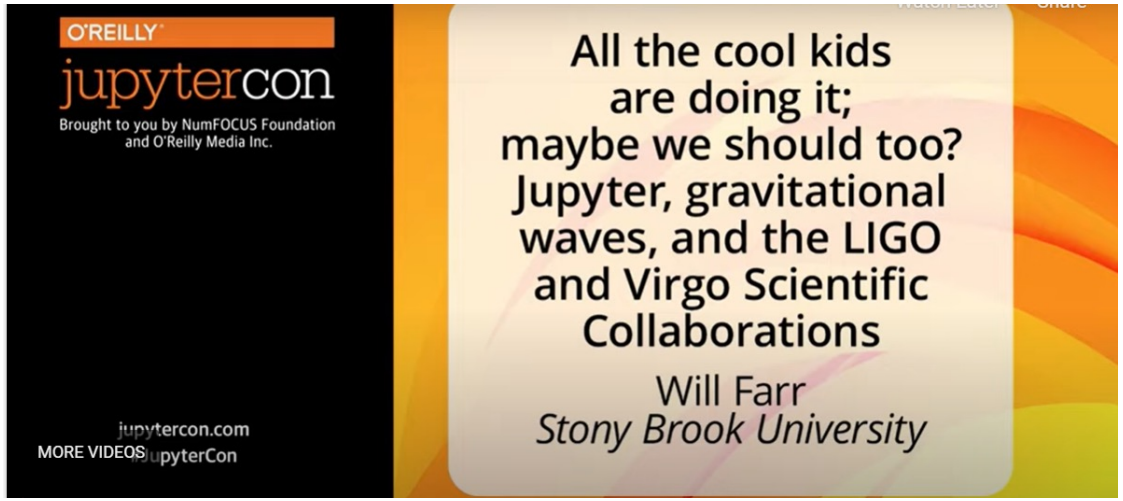
Coding – thinking computationally - LTW

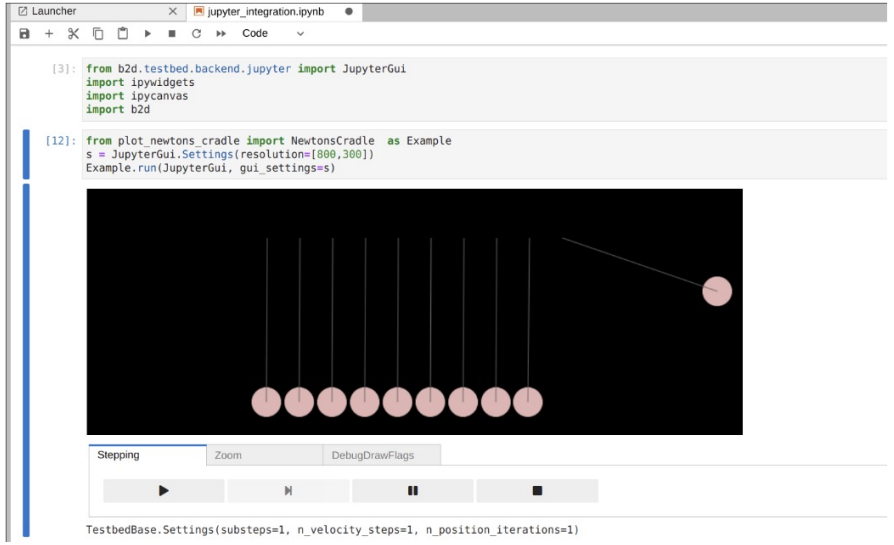
- Introduction to programming languages – Python, Julia, R
- Some uses of these programming languages at UoEdinburgh
- The journey to build a single platform for coding across the University
- Available tools and Noteable as a platform to host programming languages in virtual spaces

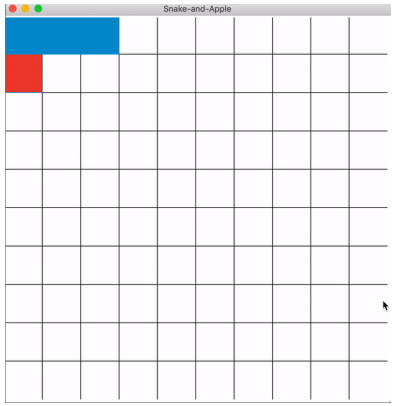
EDINA THE UNIVERSITY of EDINBURGH

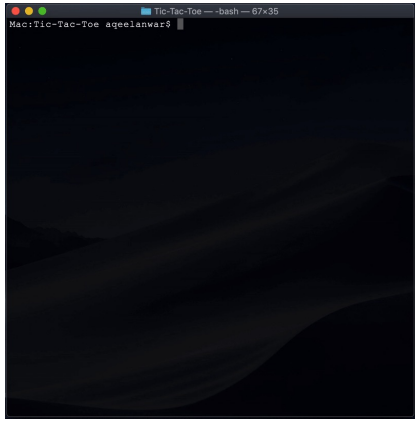
Noteable™

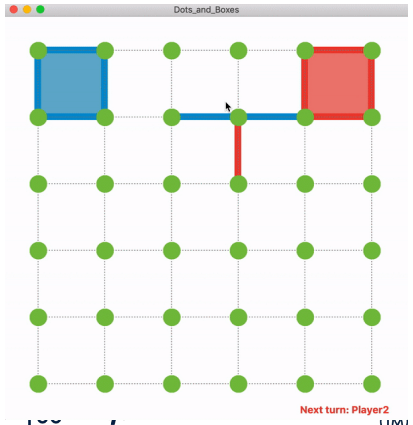
From Earth to Jupyter – no significant software comes out of nowhere











EDINA

THE UNIVERSITY of EDINBURGH

Noteable

The initial code – iPython to Jupyter

- The iPython notebook extends the console-based approach to interactive computing
- Web-based application over local installation of software
- Captures the computation process: developing, documenting and executing code
- Excellent tool for learning about code
- Communicate the results and reproduce all content visible
  - Inputs and outputs of computations
  - Explanatory text
  - Mathematics
  - Images
  - Rich media representations of objects

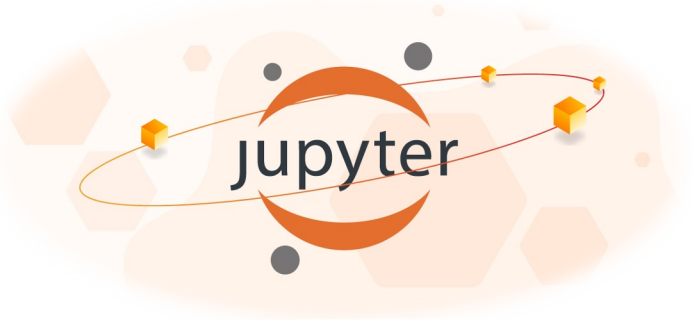
EDINA THE UNIVERSITY of EDINBURGH

Noteable™

Accessing coding environments

What is Noteable?

Noteable is provided by EDINA, UoEdinburgh as a service. Access hosted computational notebook servers integrated into VLE with tools to manage code



Computational notebooks are:

- An online tool to help you to gain data skills across disciplines
- A combined space for coding activities, explanations and collaboration
- One-stop-shop for data analysis, statistical simulations, Machine Learning fundamentals & more preconfigured by EDINA for class uses
- Extensively used across academia and industry

Noteable is a cloud-based platform to run, manage and teach with computational notebooks

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

What does the Noteable home page look like and how do I start coding?

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

jupyter Intro_Nat3_Part_1 (3) Last Checkpoint: 22/04/2021 (autosaved)

jupyter My_first_plot Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

My first plot

We will use our favorite libraries, **NumPy** and **Matplotlib**, to make a plot of a periodic function. First, our beautiful equation:

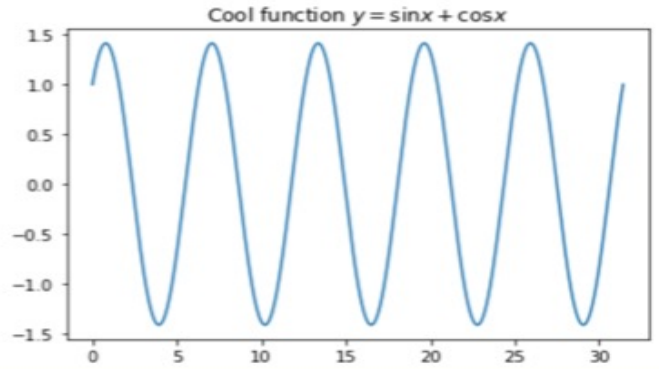
$$y = \sin x + \cos x$$

In [1]:
```
import numpy
from matplotlib import pyplot
%matplotlib inline
```

The `numpy.linspace()` function creates an array of equally spaced numbers.

In [2]:
```
x = numpy.linspace(0, 10*numpy.pi, 10**3)
y = numpy.sin(x) + numpy.cos(x)
```

In [3]:
```
pyplot.plot(x,y)
pyplot.title('Cool function $ y = \sin{x} + \cos{x} $');
```



Jupyter header and tool bar.

A markdown cell, with title, explanation, and equation.

A code cell, setting things up with needed libraries.

A short explanation.

Code cells assigning two array variables, then making a line plot.

Python 3

Disk

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

Example of coding in class: ObsPy

- ObsPy: geophysical package pre-installed in Noteable
- Access, process and visualise seismic data
- Widely used amongst geophysicists

jupyter SeismicNotebook Last Checkpoint: 18 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Memory CPU Disk

In [ ]:
```
# Define function to filter the seismograms (short-period instrument)
def filtering(filter_select, max_freq, min_freq):
    station = ['BILL', 'CCM']
    shortperiod = ['s_one', 's_two']
    starttime1 = [720, 960]
    endtime1 = [780, 1020]
    for i in range(0,2):
        shortperiod[i] = client2.get_waveforms('IU', station[i], '10', 'SH?', t+starttime1[i], t + endtime1[i])
        st_filt = shortperiod[i].copy()
        # Cut out frequencies above a certain frequency if highpass is selected
        if filter_select == 'highpass':
            st_filt.filter(filter_select, freq=max_freq,
                           corners=2, zerophase=True).plot(show=True, size=(900, 350))
        # Cut out frequencies below a certain frequency if lowpass is selected
        elif filter_select == 'lowpass':
            st_filt.filter(filter_select, freq=min_freq,
                           corners=2, zerophase=True).plot(show=True, size=(900, 350))
        # Cut out frequencies outside or within a frequency range if bandpass or bandstop is selected respectively
        else:
            st_filt.filter(filter_select, freqmin=min_freq, freqmax=max_freq,
                           corners=2, zerophase=True).plot(show=True, size=(900, 350))

# Create dropdown list widget to choose which option to use
filters = ['highpass', 'lowpass', 'bandpass', 'bandstop']
interact(filtering,
         # Dropdown widget for the filter option
         filter_select = widgets.Dropdown(options=filters, value='highpass',
                                          description='Type of filter', style={'description_width': 'initial'},
                                          disabled=False),
         # Slider widget for maximum frequency
         max_freq = widgets.FloatSlider(value=1, min=0.2, max=10, step=0.2,
                                        description='Max frequency', style={'description_width': 'initial'},
                                        disabled=False),
         # Slider widget for minimum frequency
         min_freq = widgets.FloatSlider(value=1, min=0.2, max=10, step=0.2,
                                        description='Min frequency', style={'description_width': 'initial'},
                                        disabled=False))

plt.show()
```

In [ ]:
```
# Define function to filter the seismograms (long-period instrument)
def filtering(filter_select, max_freq, min_freq):
    station = ['BILL', 'CCM']
    longperiod = ['l_one', 'l_two']
    starttime2 = [690, 960]
    endtime2 = [1200, 1230]
    for i in range(0,2):
        longperiod[i] = client2.get_waveforms('IU', station[i], '00', 'LH?', t+starttime2[i], t + endtime2[i])
        lt filt = longperiod[i].copy()
```

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

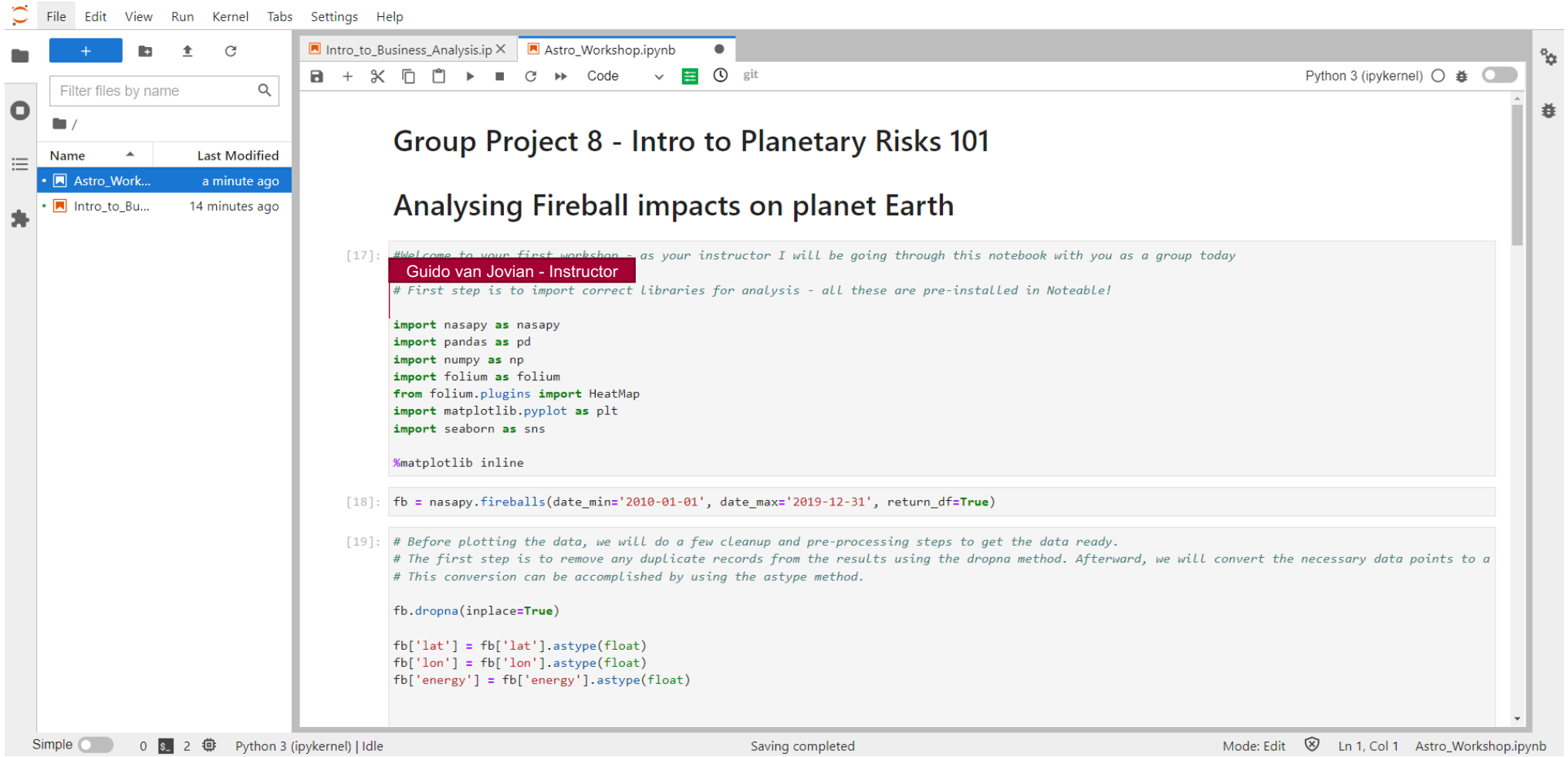
Workshop: accessing Noteable through a link to edit data in a notebook

1. Demo of access to Noteable in a playground LEARN course
   10 minutes
2. Accessing a Jupyter notebook to visualise Covid-19 case data
   15 minutes
3. Work in groups to edit parameters and list top Free Fringe events
   20 minutes
4. Run-through of available Python Lab materials and cloning them to Noteable
   10 minutes
5. Q&A and accessing materials in Noteable beyond today

EDINA THE UNIVERSITY of EDINBURGH

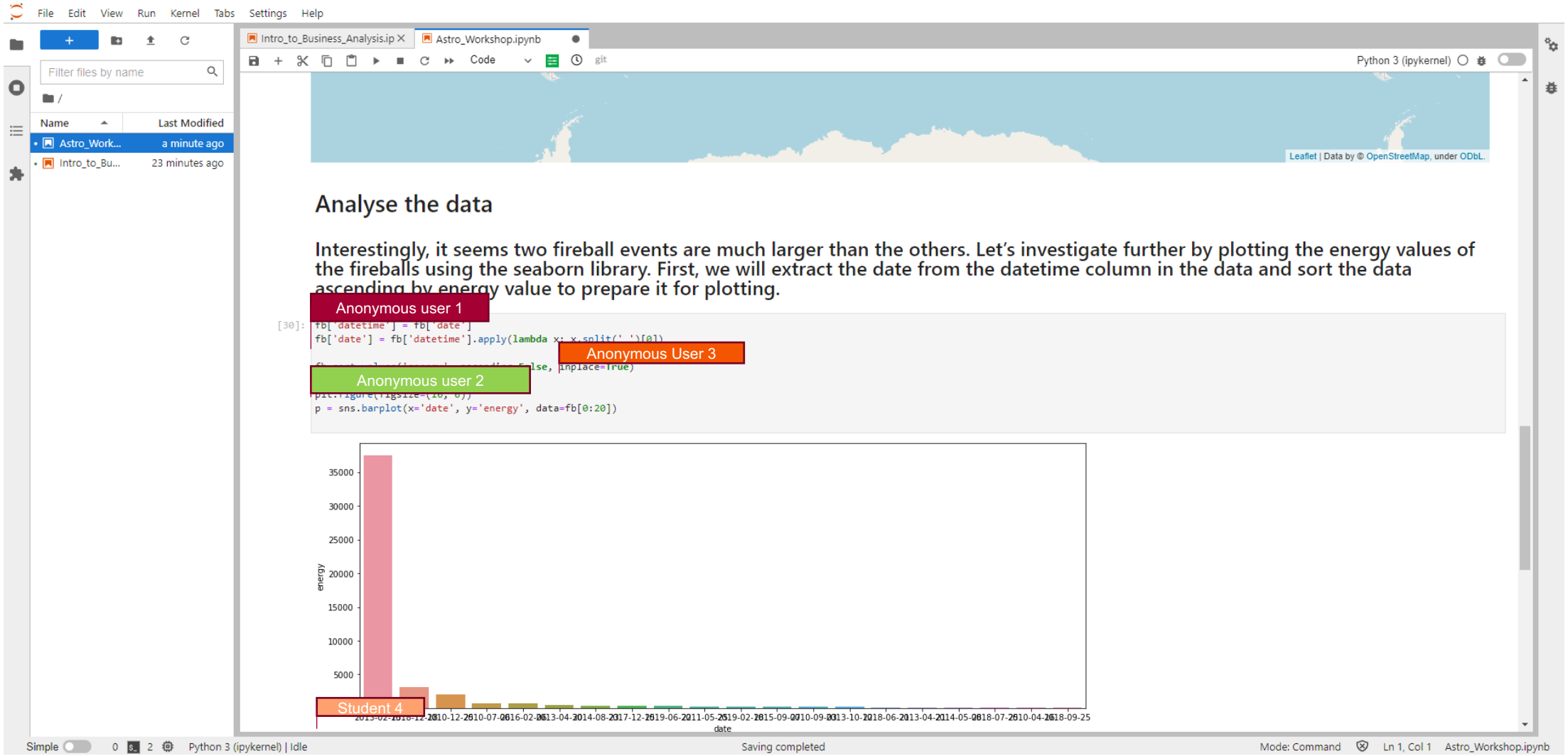
Noteable™

Coming September 2022:Collaborate coding with Noteable – Example:



EDINA THE UNIVERSITY of EDINBURGH

Noteable™



EDINA

THE UNIVERSITY of EDINBURGH

Noteable™

Takeaways

- Coding is the modern equivalent of building complex recipes to achieve desired outcomes and outputs
- Computational notebooks are an interactive platform for learning and executing code
- Noteable is a space to create, save and manage learning activities in notebooks
- You can install coding platforms locally; access in your browser through a VLE with Noteable

EDINA THE UNIVERSITY of EDINBURGH

Noteable™

Thank you for joining the session!

Do you see yourself using Noteable in your classes?

Have any questions or interest in any features?

Contact Us

edina@ed.ac.uk

EDINA THE UNIVERSITY of EDINBURGH

Noteable™