THE UNIVERSITY of EDINBURGH
School of Mathematics

THE UNIVERSITY of EDINBURGH
Edinburgh Futures Institute

Teaching an Accessible MSc Level

Introduction to Data Science Course Without Prerequisites

SERVEH SHARIFI - STUART KING

SCHOOL OF MATHEMATICS – EDINBURGH FUTURES INSTITUTE

JUNE 2024

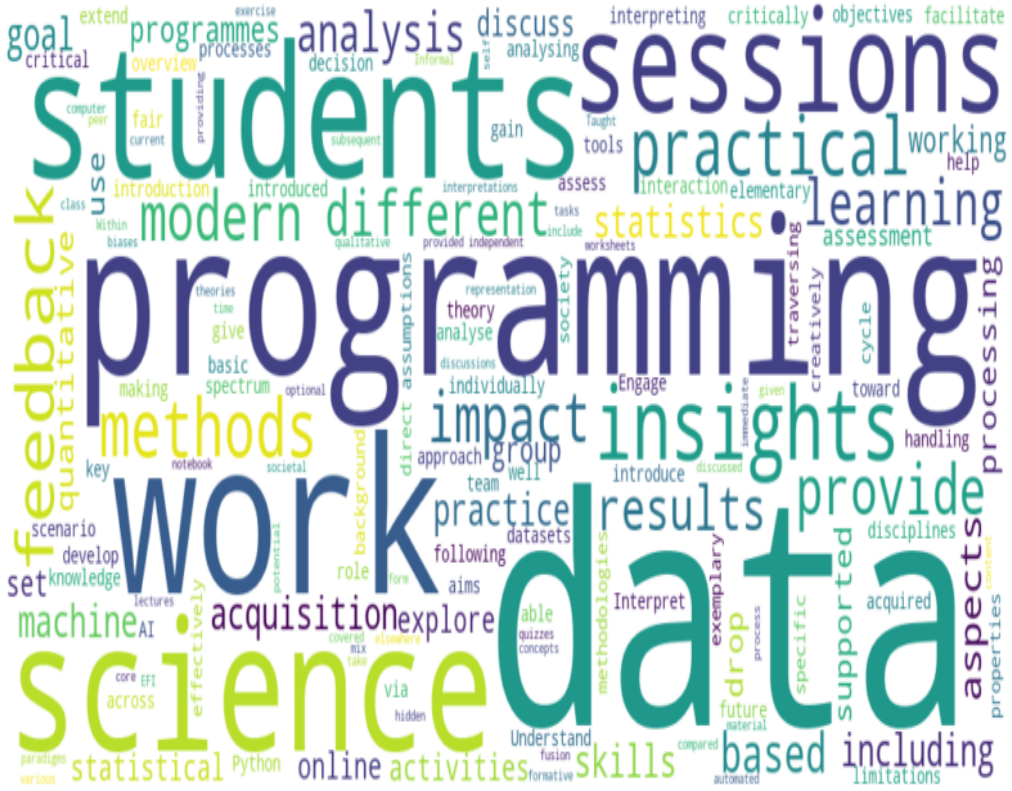
Edinburgh Futures Institute (EFI)

A new futures-focused space for learning, research, and innovation at the University of Edinburgh.

-  Circular Economy
-  Creative Industries
-  Data and Artificial Intelligence Ethics
-  Data, Inequality and Society
-  Education Futures
-  Future Governance
-  Interdisciplinary Futures (MA Hons)
-  Narrative Futures: Art, Data, Society
-  Planetary Health
-  Service Management and Design
-  Sustainable Lands and Cities

Insights Through Data

- 10 credit core course for all students in the EFI MSc programmes (an option between two courses)
- Fusion format of teaching (online and onsite students- online and onsite teaching every other week)
- 90 Students (75 onsite- 15 online), 2 lecturers and 2 TAs
- Covered topics:
  - Introduction to programming in Python
  - Introduction to Statistical modelling (linear regression)
  - Introduction to Machine Learning (classification, clustering)
- Assessment:
  Individual Programming Practical Tasks (40%)
  Group Critical Data Analysing Project (60%)



Students' prior experience

Covered topics

Week 1 - 2 - 3	**Getting started with working with data and Python (data cycle, data ethics, missing data, pandas, seaborn)**
Week 4 – 5 – 6	**Statistics (summary statistics, normal linear regression model, optional: logistic regression model, assessment of models)**
Week 7 – 8	**Machine Learning (classification, clustering)**
Week 9	Limitations, bias in algorithms and modelling, ethics
Week 10 – 11	Working on the group project

Introduction to Python and Data

- Python as a calculator
- Types of variables
- Indexing
- Loading data
- Data cleaning …

Removing columns/indices

Similar to above, it is easy to remove entries. This is done with the `drop()` method and can be applied to both columns and indices:

In [4]:
```
import numpy as np
# define new DataFrame using a function from NumPy (don't worry about what this line does just now)
data = np.reshape(np.arange(9), (3,3))

df = pd.DataFrame(data, index=['a','b','c'],
                  columns=['Edinburgh', 'Glasgow', 'Dundee'])
(df)
```

Out[4]:

	Edinburgh	Glasgow	Dundee
a	0	1	2
b	3	4	5
c	6	7	8

1. **Python as a calculator**: Python can be used as a calculator for simple arithmetical operations. See some of them in the table below:

Symbol	Task	Example	Result
+	Addition	4 + 3	7
-	Subtraction	4 - 3	1
/	Division	7 / 2	3.5
*	Multiplication	4 * 3	12
**	Power of	7 ** 2	49

```
here is the way we would drop two columns:
='columns')
```

Let's try them:

```
print(51/7)
print(round(51/7, 2))
print(21*21)
print(2**5)
```

Statistics and machine learning

- Example based lectures/videos
- Concept and code
- Revision of code in workshops and extra exercises
- Extra optional topics in videos and notebooks

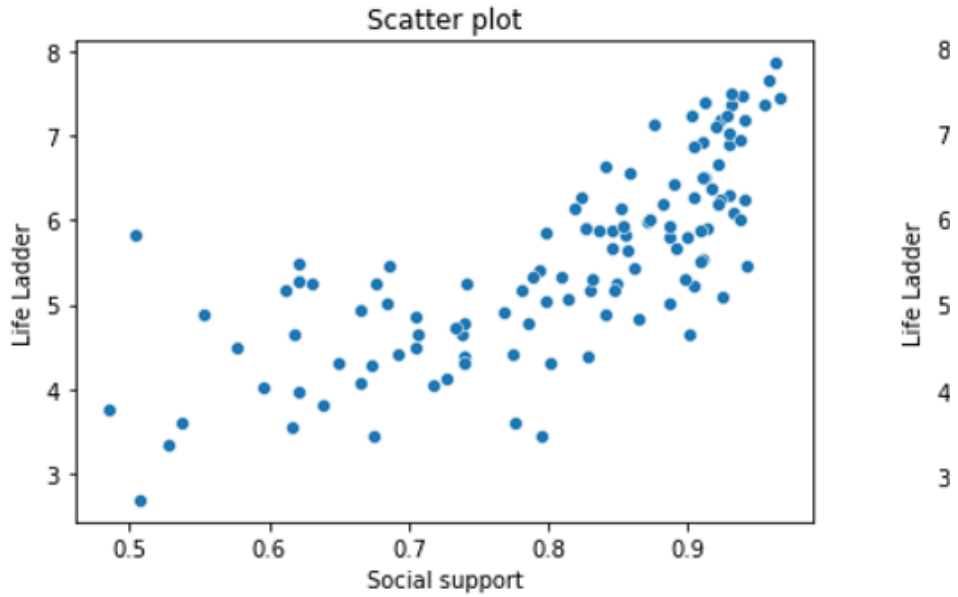
World happiness data

Six key variables are used to explain the variation of happiness across countries.

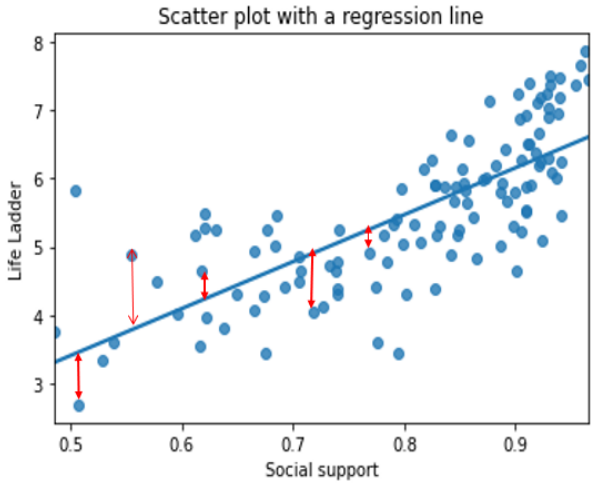
	Country name	Year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect
10	Afghanistan	2018	2.694303	7.494588	0.507516	52.599998	0.373536	-0.084888	0.927606	0.424125	0.404904
21	Albania	2018	5.004403	9.412399	0.683592	68.699997	0.824212	0.005385	0.899129	0.713300	0.318997
28	Algeria	2018	5.043086	9.557952	0.798651	65.900002	0.583381	-0.172413	0.758704	0.591043	0.292946
45	Argentina	2018	5.792797	9.809972	0.899912	68.800003	0.845895	-0.206937	0.855255	0.820310	0.320502
58	Armenia	2018	5.062449	9.119424	0.814449	66.900002	0.807644	-0.149109	0.676826	0.581488	0.454840
...	...	...	...	...	...	...	...	...	...	...	...
1641	Uzbekistan	2018	6.205460	8.773365	0.920821	65.099998	0.969898	0.311695	0.520360	0.825422	0.208660

```
import scipy.stats
scipy.stats.pearsonr(happy_2018["Social support"], happy_2018["Life Ladder"])[0]
```

```
0.7521436886148463
```



Statistics and machine learning



Life Ladder = a + b * Social support

$$y = a + b * x$$

a and b are found such that sum of squares of errors is minimised.

```
from statsmodels.formula.api import ols
model_ss = ols('Q("Life Ladder") ~ Q("Social support")', data=happy_2018).fit()
model_ss.params
```

```
Intercept             -0.014001
Q("Social support")    6.855793
dtype: float64
```

Life Ladder = *-0.014* + *6.856* * Social support

The coefficient value “b” indicates, given a one-unit increase in Social Support, the mean of the Life Ladder increases by 6.856 units.

Statistics and machine learning

KNN

- The choice of 3 or 5 points can change the outcome – more points overcome more noisiness in the data.

An example

- We'll start with a made-up example with some synthetic data.
- This one is useful as it has only two features and we can therefore visualize it
- The dataset comes in two parts – training data, and test data. Both have data points and the associated class labels. The train data is to use to construct out KNN model, the test data labels are used to see whether we get the correct answer.


```
knn_class = nei.KNeighborsClassifier(n_neighbors=3, metric='euclidean')
knn_class.fit(class_data[['xs','ys']],class_data['yc'])
```

Python with Jupyter notebooks on Noteable

An introduction to Pandas

pandas is a module which allows the construction of a *dataframe*, this principally by a column name and row name/number). It also includes

Again, the website for Pandas is good and contains the main set of do is more dense to read. Here is the main website.

- The main documentation for Pandas is here.
- There is a quick introduction to Pandas here
- There is a fantastic tutorial (also in Jupyter) here (under Lessons introduction to the basic concepts in Pandas.

Here are some basic examples to getting started with pandas, the dat them in our code.

```
# Common pandas import st
import pandas as pd
```

seaborn library

Seaborn is a library/module/package for making statistical graphics in P Seaborn has a nice structure which makes plotting easier using three ty "categorical" plots).

The general command is:

```
sns.---plot(data=---, x="---", kind="---")
```

and you can add appropriate optional elements to it:

```
sns.---plot(data=---, x="---", y="---", kind="--", hue="-
```

lary and

Machine Learning

The Scikit Learn module

The scikit learn module is an open source module dedicated to weeks to explore these methods and apply them to some data.

There are particular sections of the module dedicated to superv examples and where we do not respectively). Here we are goin classes (denoted 1 and 0), we want to use that data to construc

Firstly we are going to need to import some modules and functi decision trees):

```
: import pandas as pd
  import sklearn as skl
```

Regression model for Life Ladder as the response variable and S

The `lmplot` from Seaborn module makes a scatter plot with a regression line fitted to the

```
sns.lmplot(data=happy_2018, x="Social support", y="Life Ladder", ci=None)
plt.title("Scatter plot with a regression line")
plt.ylim(0, )
plt.xlim(0, )
plt.show()
```

We use the Python module statsmodels to fit regression models. There are other Python m complete outputs which help with understanding the process. See this link for extra regres

The function `ols` from `statsmodels.formula.api` is imported to fit the regression mod used in the formula because there is a space included in the variable names, otherwise yo

Exercises

Here is a new dataset that contains 4 columns in the relevant dataframe point.

```
: operation_data = pd.read_csv('haberman.csv')

  print(operation_data.head())
  print(operation_data.keys())
```

```
   age  yoo  npan  class
0   30   64     1      0
```

Help with coding

- Programming workshops on even weeks
- Optional drop-in Q&A sessions on odd weeks
- Pair-programming
- Rubber ducks
- Q&A channel on Teams
- Introduced online books
- Practice on CodeRunner

Weiwei Yan 06/10/2023 15:21 Edited

Hi! I have a question for w3 practical -
store log of prices in the data file
housedata["logprice"] = np.log(housedata["price"])

See more

```
In [19]: # import the data
housedata = pd.read_csv("kc_house_data.csv")

# store log of prices in the data file
housedata["logprice"] = np.log(housedata["price"])

# the first 5 rows of the data
housedata.head()
```

Out[19]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	flo
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	

https://pairprogramming.ed.ac.uk/

https://en.wikipedia.org/wiki/Rubber_duck_debugging

CodeRunner on Moodle

Create a function **temp_convert** that takes as an argument the temperature in degrees Fahrenheit T_f and converts it it to degrees Centigrade T_c. The formula you will need to use is:

$$T_c = \frac{5(T_f - 32)}{9}.$$

Your function should print out the new temperature in degrees Centigrade rounded to 2 decimal places, it doesn't need t
value.

For example:

Test	Result
temp_convert(59)	15.0

Answer: (penalty regime: 0 %)

Reset answer

```
1 def temp_convert(deg_far):
2     # Insert your code here
3     deg_cel = 5*(deg_far - 32)/9
4     print(round(deg_cel, 2))
5
```

	Test	Expected	Got	
✗	temp_convert(59)	15.0	15.0 15.0	✗
✓	temp_convert(32)	0.0	0.0	✓
✓	temp_convert(80)	26.67	26.67	✓
✓	temp_convert(0)	-17.78	-17.78	✓
✓	temp_convert(100)	37.78	37.78	✓

Your code must pass all tests to earn any marks. Try again.

Show differences

Assessment: Nbgrader on Noteable

- 3 Assessments: Individual Programming Practical Tasks (40%)
- In Jupyter notebooks and familiar to students
- Combination of auto-marking and manual-marking

ID: cell-0b457726beb21601 Autograded answer

```
### BEGIN SOLUTION
np.sqrt(reg_model.mse_resid)
### END SOLUTION


#  assign the value to answer_q5; for example answer_q5 = 106.1234

answer_q5 =
```

Points: 10

```
assert isinstance(answer_q5, float)

### BEGIN HIDDEN TESTS
assert round(answer_q5, 2) == 0.34
### END HIDDEN TESTS
```

Points: 20 ID: cell-q6 Manually graded answer

```
### BEGIN SOLUTION

new_data = pd.DataFrame({"log_sqft_living": [mean_log_sqft_living], "log_sqft_living15": [mean_log_sqft_living15],
                         "view": [2], "grade": [8], "bedrooms": [3], "bathrooms": [2]})
predictions = reg_model.predict(new_data)
new_data["prediction"] = predictions
log_pred = new_data.iloc[0, 6]
pred = np.exp(log_pred)
pred

### END SOLUTION
```

Assessment: Group project

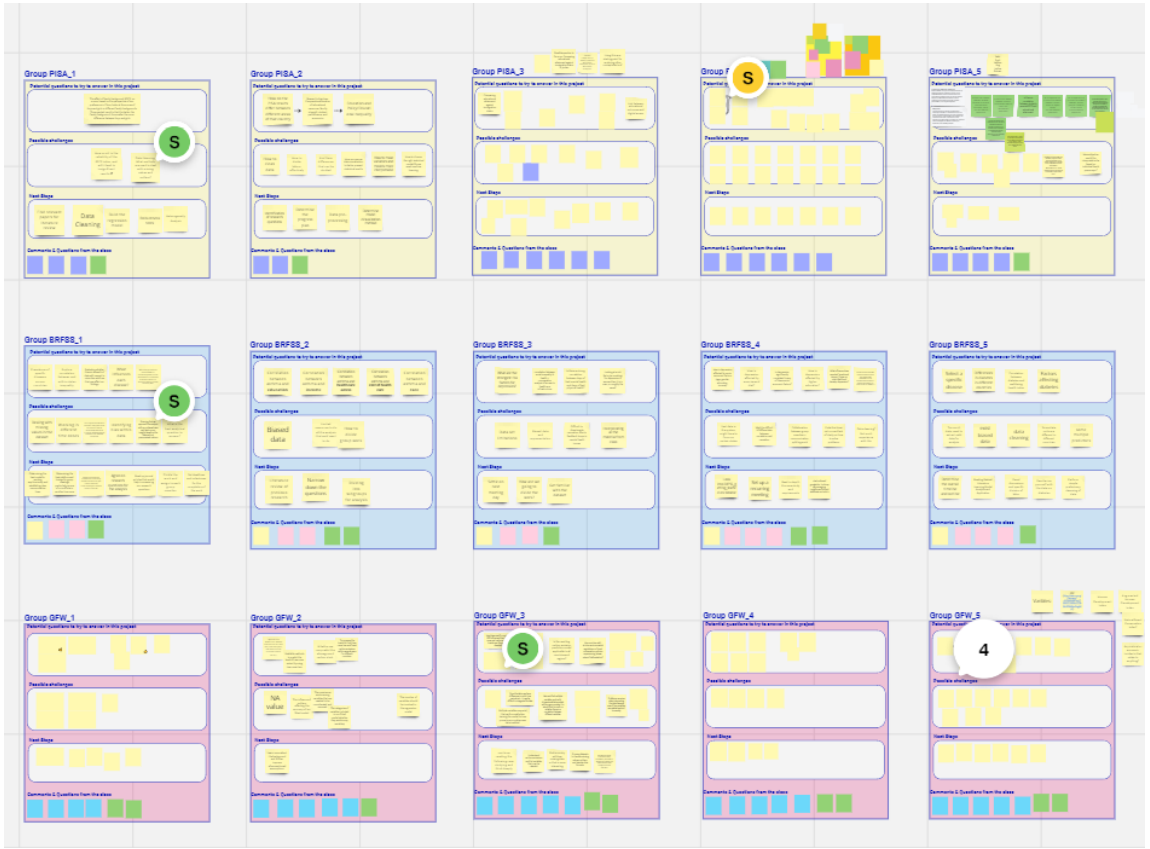
A maximum 3000 word report + A Jupyter notebook

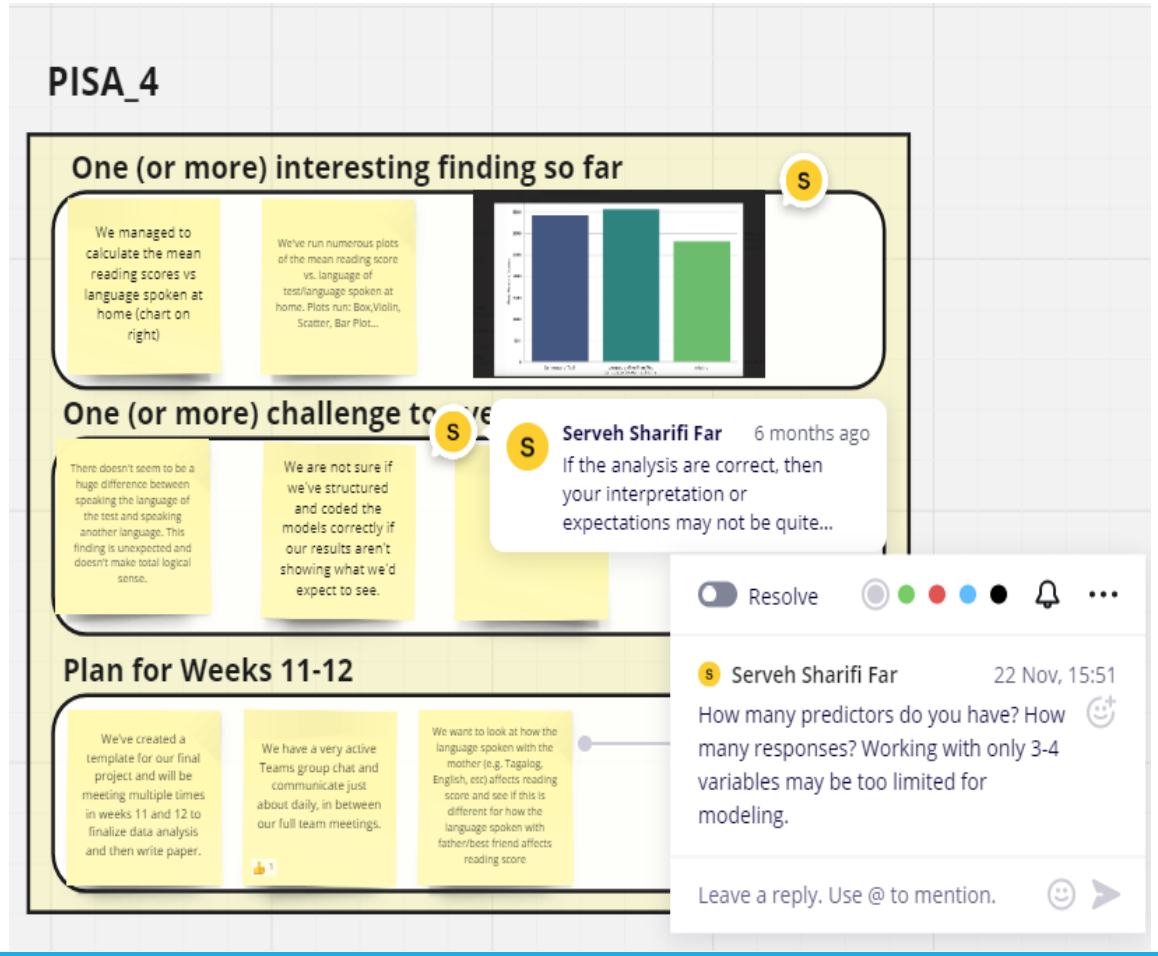
- Quality of data analysis
- Quality of insights and reflections
- Quality of presentation

Datasets related to the UN Sustainable Development Goals

- Education Data
- Health Data
- Environment Data

Assessment: Checking the progress on Miro





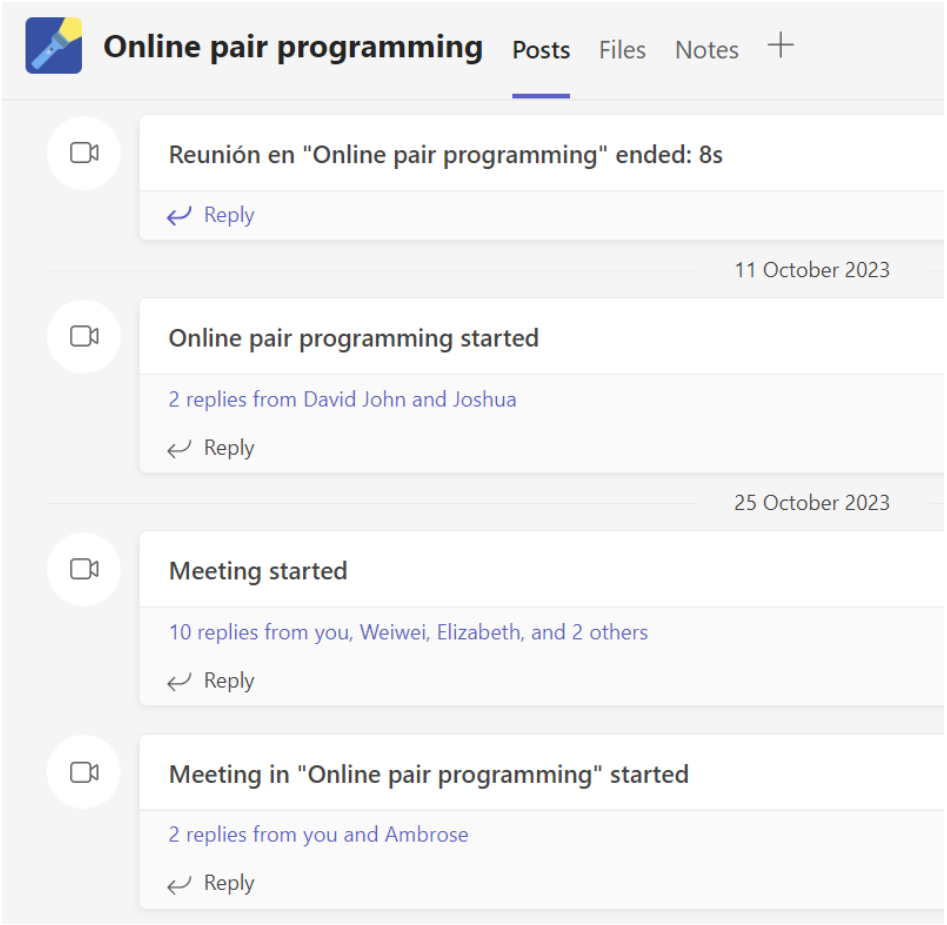
Challenges

- Ensuring that the course content is useful for all levels:
  - Statistics
  - Machine Learning
  - Python programming
  - The 60% final project on multidisciplinary topics (educational, environmental, health data)
- Group projects and guiding students to focus on appropriate questions and analyses
- Providing support for asynchronous students
- Effectively communicating the importance of learning some programming to work with data
- Assuming no programming background from students and spending enough time on building up confidence
- Scaling up the course for a larger number of students

Challenges

- Programming workshops on Teams for online students:

- The need for at least one lecturer/TA for supervising and helping
- Extra time for forming groups and starting calls
- Different experience of pair-programming by sharing screens
- Pair-programming often turned into group-programming



Thank You!