Féidearthachtaí as Cuimse
Infinite Possibilities

Week 6
Data analysis

Fundamentals of IoT

Dr. Eoin Rogers (eoin.rogers@tudublin.ie)

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

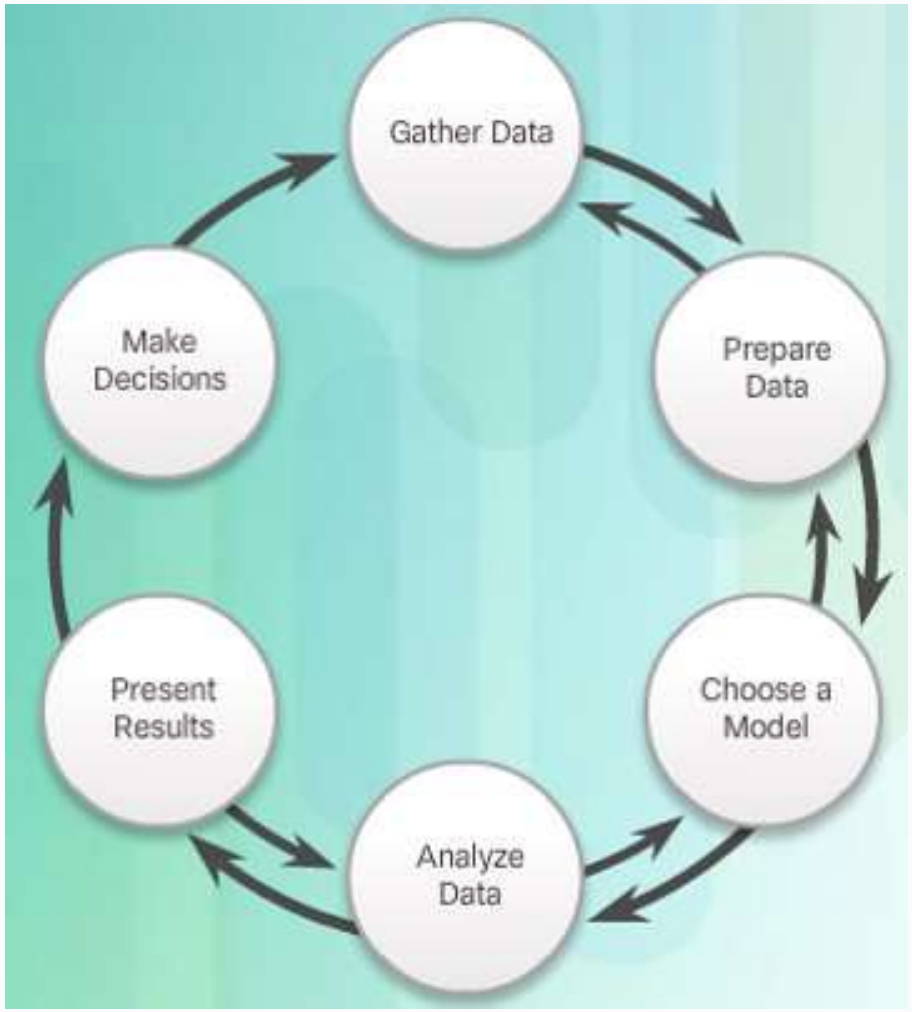
Lesson Outline

- This will be a basic introduction to data analysis: what is it and why is it useful
- We will cover both NumPy and Pandas, two Python libraries widely used for data analysis
- We will also talk briefly about common statistical models

What is data analysis?

- Modern systems often produce significant quantities of data
- This is particularly true for IoT!
- This data can be analysed to discover patterns and useful information
- This might be good for informed decision making

Six step analysis lifecycle



Notice that there is a vague similarity to the **scientific method**: we're trying to **gather data** and then **build models that explain** that data.

Big data

- IoT systems can be a significant source of **big data**. This can be useful for:
  - Decision making
  - Machine learning
  - Improving IoT systems
- But it also means we have to be mindful of **ethical issues**!

Terminology

There are some important pieces of terminology we should keep in mind

Data analysis can be:

- **Descriptive** – We might just want to know what happened
- **Predicitive** – We might want to build models to predict what will happen or to fill in gaps in the data
- **Prescriptive** – We might want to use the data to inform decision making and help us choose between alternatives

The four big Vs of data

- **Volume** – The quantity of data
- **Velocity** – The frequency of the data
- **Variety** – The unpredictability of data
- **Veracity** – The accuracy of the data

States of data

- Data can be
  - **In motion –** i.e. being transmitted over the network
  - **At rest –** i.e. being stored on a disk
  - **In use –** i.e. being processed

NumPy

Let’s look at some actual code!

ndarray

- The **core class** of NumPy – an **n-dimensional array**
- Similar to a Python list, but much more powerful
- Although everything in the array **must have the same type**, and the **dimensions along each axis must be uniform**

Attributes of ndarray

- ndim
- shape
- size
- dtype
- itemsize

Creating an ndarray

Use array() to create a Python list:

```
import numpy as np

my_array = np.array([[1, 2, 3],
                     [4, 5, 6]])
```

Creating an ndarray

Or create an array of all zeros:

```
import numpy as np

my_array = np.zeros(shape=(3, 3))
```

Creating an ndarray

Or of all ones:

```
import numpy as np

my_array = np.ones(shape=(3, 3))
```

Creating an ndarray

Use arange() to create an array containing a range:

```
import numpy as np

zero_to_nine = np.arange(10)

two_to_nine = np.arange(2, 10)

zero_to_nine_even = np.arange(0, 10, 2)
```

Creating an ndarray

Use linspace() to create an array containing evenly spaced numbers:

```
import numpy as np

twelve_numbers = np.linspace(0, 10, 12)
```

(Note this includes the 10)

Reshaping an array

```
four_by_three = twelve_numbers.reshape(4, 3)
```

Elementwise maths operations

If x and y are ndarrays of the same shape:

```
x + y

x - y

x * y # N.B. This isn't matrix

      # multiplication!
```

Matrix multiplication

Use the @ symbol for matrix multiplication:

```
x @ y
```

Array indexing and slicing

Very similar to how it works in Python:

- `a[0], a[-2], a[1:5]`
- `a[1:5:2], a[1:], a[:5]`
- `a[::2], a[::-1]`

Array indexing and slicing

But we can seperate dimensions by commas:

- `a[0, 5], a[1:3, 5]`
- `a[::2, 3], a[1::2, 2:8]`

Three useful methods

- Use **`ravel()`** to flatten arrays
- Use **`vstack()`** to stack arrays vertically
- Use **`hstack()`** to stack arrays horizontally

Pandas

This is another Python library widely used for data analysis

DataFrames

- The **DataFrame** is the core data structure used in Pandas
  - You can think of it as a **cross between a NumPy array and a database table**
  - Consists of rows of data each of which follows a **schema** specified in a columnar format

Creating a DataFrame

```
import pandas as pd

students = pd.DataFrame({
      'Name': ['John', 'Mary', 'Bob', 'Anne'],
      'Subject': ['Computer Science', 'Physics',
                  'History', 'Business'],
      'Average exam result': [70, 80, 65, 85],
})
```

Creating a DataFrame

```
import pandas as pd

students = pd.read_csv('students.csv')
```

Useful DataFrame methods and attributes

- `head()`
- `tail()`
- `index`
- `columns`
- `to_numpy()`
- `describe()`
- `T`
- `sort_index()`
- `sort_values(by='name')`

Access a column by name

```
students['Name']
```

Access a column by index (primary key)

```
students.loc[0]
```

Filter a row by column

```
students.loc[0, ['Name', 'Subject']]
```

DataFrames support slicing!

```
students.loc[0:2, ['Name', 'Subject']]
```

Boolean indexing (like an SQL query)

```
students[
  students['Average exam result'] >= 80
]
```

Data analysis techniques

This will be the last topic!

Important terms

- Averages
  - Mean
  - Median
  - Mode
- Standard deviation
- Percentiles

Dimensionality reduction

- Given a high-dimensional input, reduce it to low-dimensional data while minimising the amount of information that is lost
- Examples:
  - Linear discriminant analysis
  - Principle component analysis
  - Autoencoders
  - Information gain

Statistical correlation analysis

- A mathematical measure of the **similarity** between two variables
  - In other words, given that you know the value of variable A, how accurately can you predict the value of B?
- One measure in common use if **Pearson correlation**

Pearson correlation

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}}.$$

`x` and **`y`** are the two datasets, both of length **`n`**, and we use **`i`** as an index name

Summary

- IoT devices produce huge quantities of data, so it can be useful to know some basic data analysis techniques to deal with them
- Terminology like the four big Vs or data in motion/rest/use are important, even if they seem trivial
- Pandas and NumPy are the workhorse of the data analysis world
- Statistical and machine learning terms are important as well

That’s all for this week

Thanks for your attention!