Fundamentals of Data Science Workshop 1

Week 5 (Week 9 for LS): 24/02/2025 (LS: 24/03/2024)

Aims of the Workshop

In this introductory week we started exploring the concept of the scientific method, we have looked at a specific data science example (sports analytics) using a limited data set. We have seen how to fit lines to data sets (using linear regression) and we have started to think about different types of data. In the week ahead we will focus on data cleaning in more detail and progress towards data management.

Workshop Timetable

The workshops in Fundamentals of Data Science are different to those of other modules. We will use the time not only for coding but also for discussion, small group activities, and tutorial activities. Additionally, your engagement with these exercises will be assessed by the **Canvas Quizzes** for each week. The outline for this week is given below. Please note that this should be taken as indicative only. We will adjust the times according to how quickly things progress. Also note that we do not assign deadlines to these workshops as the activity is very open-ended. You can use the MS Teams channel for any questions/requests for help.

Time	**Activity**
15:00-15:05	Introductions
15:05-15:20	Group Discussion / Tutorial on Sports Analytics Task
15:20-18:00	Exercises

Useful Information

Throughout this workshop you may find the following useful.

Python Documentation

You can look up the core language features of Python 3.8 here as well as tangential information about the Python language. You may also need to review the content from the Programming module earlier in the course.

Jupyter Notebook Basics

Jupyter itself offers some basic documentation for people new to the editor. These can be found here.

Jupyter can also use markdown cells for text input which can be useful for making notes on your code e.g., if you wish to annotate each cell as to which Exercise it belongs to. You can see how to do this here.

Reminder

We encourage you to discuss the content and any findings you gather from this session with the teaching team in their office hours. Note that workshops are not isolated undertakings - if you have questions from previous weeks or lecture content, please talk to us.
The content of this workshop is not intended to be 100% completed within the session; as such it is expected that some of this work will be completed outside of the session. The exercises represent examples of what to do and you should feel free to expand upon them.

15:00 Introductions

The Lecturers for the Fundamentals of Data Science Workshops are:
Dr. Tareq Al Jaber (t.al-jaber@hull.ac.uk); Dr Aarzoo Dhiman (a.dhiman@hull.ac.uk);
Additionally, there are also GTAs that will assist during your workshop session.

15:05 Discussion (mini-tutorial / study group) on Sports Analytics Task

Earlier, we introduced you briefly to some data from the Australian Football League (AFL).

In this workshop, we'd like to discuss this exercise before starting on some coding tasks. This will get you used to thinking like a data scientist in a given domain.

Open Questions:

a) How would more data have helped you decide which team has a better chance of winning?

b) What sort of extra data would you have liked access to? And importantly: why?

c) Is it possible to even derive an answer to the question posed with the data supplied? Why or why not?

d) Further contextualisation: If you owned a betting or gambling business, would you take bets on the game's outcome? If so, what factors would be important and why? How would you compute what odds to give your customers?

15:20 Exercises

For the coding tasks below, create a new notebook and name it something appropriate and/or memorable such as "DataScienceWorkshop1". I have tried to divide up these tasks in to "science" and "coding", although the difference between the two will get fuzzier as we progress.

Exercise 1

It can be helpful to create random data to use for testing your code. In science, this is often done so that people can test their code on fake input data to make sure it works before applying it to real world data. The first exercise will show you how to generate some fake data. To do this, we are going to use the "random" module.

Coding.

Let's use the random package to generate 100 random integers between 1 and 100.The example below shows how to generate a list containing 20 random integers. Modify this code to create 100 random integers.

```
import random
x1 = []
for i in range(20):
    x1.append(random.randint(1,100))
```

Exercise 2

Coding.

Look up how to use the random module to create floating point random numbers, rather than integers. Generate 100 random numbers with one decimal place and store them in another list (e.g., in a list called “x2”).

Exercise 3

Coding.

Let’s generate 100 random coordinates. Use the “random” module to create 100 (x,y) coordinates, where x and y can range between 1 and 100, have one decimal place.

Exercise 4

Science.

If we tried to fit a line of best fit to the coordinates obtained in Exercise 3, what would you expect the result to be? Answer this either by yourself, in a small group, or talk to a demonstrator about it before doing anything else.

Once you have answered this, justify your answer by attempting a linear regression on this data. Does it agree with your prediction? Why or why not?

Exercise 5

We’re going to go a bit beyond linear regression now. We will produce some random data that should be fit by a parabola instead (i.e., an equation that looks like $y = ax^2 + bx + c$, where a, b, and c are all constants). Try this:

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)
X = 2 - 3 * np.random.normal(0, 1, 20)
Y = X - 2 * (X ** 2) + np.random.normal(-3, 3, 20)
plt.scatter(X,Y, s=10)
plt.show()
```

In the plot, you should see a set of random data points that look like a curve.

Exercise 6

Coding.

We will cover this process in more detail in upcoming lectures. However, for today’s workshop, I’d like you to attempt to fit a parabola to the above data.

This can be achieved with *polynomial regression.*

```
# First import the following modules:
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import leastsq
```

```
# Now we have to define some function that describes a parabola:
def func(params, x):
    a, b, c = params
    return a * x * x + b * x + c

# Define an error function:
def error(params, x, y):
    return func(params, x) - y

# And a function to find the solution parameters:
def solvePara():
    p0 = [10, 10, 10]
    Para = leastsq(error, p0, args=(X, Y))
    return Para

# Finally, the solution can be done as follows:
def solution():
    Para = solvePara()
    a, b, c = Para[0]
    print("a=",a," b=",b," c=",c)
    print("The equation of the curve is:")
    print("y="+str(round(a,2))+"x*x+"+str(round(b,2))+"x+"+str(c))
```

If you now call “solution()” it should output the parameters of the parabola.
If you’ve used the same random number seed as the example above the solution will be:

```
a= -2.1096712526292816
b= 0.8908751717649521
c= -2.078090020446448
```

Check if you can recover these values.
Extension Exercise: Add a curve to a graph of your points using the above solution.

Exercise 7

Coding.
How would you modify the above code to fit a third order polynomial?
(A third order polynomial would have the form $y = ax_3 + bx_2 + cx + d$).

Exercise 8

Science.
How would you expect the fit to change as you added (or removed) more of those random points that were generated in Exercise 5 and fit in Exercise 6?

Exercise 9

Science.
How would you identify “outliers” in the data prior to fitting a function to it?

Coding.
Add in a clearly spurious point to the data from Exercise 5. Repeat Exercise 6 but modify your code so that it can detect this clear outlier. Feel free to discuss the ways that you might do this on the Teams channel or with a teacher.