Fundamentals of Data Science Workshop 4

Week 8: 18/03 – 22/03

Aims of the Workshop

We have learned a lot about data cleaning, analysis, and statistics. We will now see how to visualise our data sets. This is useful for spotting patterns in data sets, and also for presenting our work in the best-possible way.

Workshop Timetable

The workshops in Fundamentals of Data Science will be significantly different to other modules. We will use the time not only for coding, but also for discussion, small group activity, and tutorial activity. For this week, the timetable is given below. Please note that this timetable should be taken as indicative only. We will modify 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.

Time	**Activity**
15:00-15:15	Discussion on Exercises
15:15-16:30	Exercises
16:30-16:45	Discussion / Tutorial on Exercises
16:45-17:25	Exercises
17:25-17:30	Wrap Up and Reflections

Useful Information

Throughout this workshop you may find the following useful.

Python Documentation

https://docs.python.org/3.8/

This allows you to look up the core language features of Python 3.8 as well as tangential information about the Python language. We will refer you back to the Programming module by Brian Tompsett for more notes on this.

Jupyter Notebook Basics

Jupyter itself offers some basic documentation for people new to the editor. These can be found at https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html

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: https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html

15:00 (and 16:30) Discussion

The first discussion this afternoon is to think what the key figures might you produce for your project? What figures will you need to address the goals of the project?

15:15 Exercises

Arguably the most obvious figure that we need is one that shows the ages of the population.

Exercise 1
Using your project data, make a histogram (using either matplotlib or seaborn – take your pick!) of the ages of the population.

Take your time and choose sensible bin widths for the histogram. How did you decide this? Make a post in the chat and compare notes with others!

Exercise 2
We want a bit more detail than just the overall age of the population.

Make two further histograms: one for 'male' and one for 'female' ages. Once again, consider what bin width to use here.

Exercise 3(i)
Demographers often talk about an *age pyramid* to characterise ages versus gender. Let's have a go at making one ourselves. To do this, we are going to use some REAL data from the UK.

You will need the two files from Canvas that are called:
'2019estimatesMale.csv' and '2019estimatesFemale.csv'

[Both of these files are estimated projections from the Office for National Statistics for 2019.]

Firstly, we will need to read them both in (in Pandas maybe) and check that everything looks good.

Let's try to put the data in the format that we need.

First step: multiply all 'male' counts by -1 (i.e., so that they are negative numbers). The reason for doing this will become obvious very soon.

Exercise 3(ii)
Import the following into your code in addition to Pandas:

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

We will also need the male and female data in a single dataframe (not strictly 100% necessary but will help with the next task). Create a new dataframe in Pandas that has the format:

Age Group, number of males (negative), number of females

(note the negative value is what you did in Exercise 3(i))

You should end up with a dataframe that is similar in format to the following *(NB: feel free to use the following as an example in Exercise 3(iii) initially if you are having problems creating the histogram data from the csv files.)*

```
age_p = pd.DataFrame({'Age': ['100+', '90-99', '80-89', '70-79', '60-
69', '50-59', '40-49', '30-39', '20-29', '10-19', '0-9'],
'Male': [-6, -17, -92, -180, -290, -427, -544, -506, -545, -515, -
427],
'Female': [4, 15, 76, 176, 314, 487, 634, 642, 579, 535, 447]})

AgeClass = ['100+', '90-99', '80-89', '70-79', '60-69', '50-59', '40-
49', '30-39', '20-29', '10-19', '0-9']
```

Exercise 3(iii)

Let’s plot our age pyramid now using seaborn barplotting.

Try this:

```
age_pyramid = sns.barplot(x='Male', y='Age', data=age_p,
order=AgeClass, color=('mediumblue'), label='Male')

age_pyramid = sns.barplot(x='Female', y='Age', data=age_p,
order=AgeClass, color=('darkorange'), label='Female')

age_pyramid.legend()
plt.title('Age Pyramid')
```

Where ‘Male’ is the name of the column that contains the actual number of males with an age stored in the column ‘Age’ (and similar for ‘Female’). You might need to do some wrangling to get the data in to this format.

Let’s also do some labelling:

```
age_pyramid.set(xlabel='Population Count', ylabel='Age Group')
```

Exercise 3(iv)

Evaluate how your plot looks. Does it communicate the information well? Are there any improvements you can think of?

We will talk about Exercise 3(iv) in our **16:30 discussion.**

Exercise 4

Produce an age population pyramid for your **Project Data.**

NB: This exercise assumes that you have cleaned your 'Ages' data from last week. If not, simply use a "dropna" command to get rid of the missing data so that you can at least plot it. You can return to the cleaned data later.

You will need to put your data into bins, and this may take a bit of time to do / think about! Feel free to post your solution to this to the Teams channel. Look once again at the example dataframe given in Exercise 3(ii) to help with this.

Exercise 5

Science.

Does the age pyramid from your project look anything like the age pyramid from the government statistics? What are the differences (if any)? Why do you think those differences arise and are they significant?

Exercise 6 and Extension Work

Science and coding.

We will have to make some kind of evaluation about **commuters** in the Project data set.

By "commuter" we mean someone who might live in the town that the census has been taken in, but works in one of the nearby cities, rather than within the town itself.

Question for discussion / chat: how would you identify a commuter?

This is not an easy question as the data given didn't record this in any systematic way.

(Aside: modern era censuses directly ask this).

But! Could we infer the existence of commuters somehow?

Are there certain classes of people in the census who must be commuters?

Are there certain types of people who are more likely than others to be commuters? Why? Justify your answer.

Once you have decided how you might infer who are commuters, try to determine how many such commuters exist in the data set by finding people who match your criteria.

As a fraction of the population, how many people work outside the town and therefore need to travel significant distances for work?

This exercise requires a few assumptions and guess work. But can you make a rigorous argument about it?