

Principles of Computing

This report will go over the development of the python application which reads in over 3 years of South Wales Police Data and will manipulate and report on the data. Throughout this document the word application will be regarded as app.

This application will use several modules which would need to be installed via pip, these are as follows: *glob2, matplotlib, pandas, tkinter* (2 of these are built into python). The application can be run from the terminal, idle, spyder or vscode. *The main development was created within VS Code and running the python file from a terminal on Ubuntu 20.04.1.*

1 – Data

Before even starting with thinking about an algorithm or what type of analysis that can be done, it is important to do a bit of some Exploratory Data Analysis (EDA). Whilst analysis is within the name, for this part it is going to be looking at the layout of the data and seeing what data could be useful for some analysis.

To begin with, the data was accessed from <https://data.police.uk/data> (Data Police, 2020) which is the which has been published under the Open Government Licence which allows for copying, publishing, adapting, and exploiting the data (OGL, 2020).

The data downloaded for this project was downloaded from October 2017 to September 2020 for the South Wales Police force. When looking at the database and the layout, there are a lot of rows here which will not be particularly useful. For example, CrimeID, Reported by (we are only looking at South Wales Police), Falls Within, LSOA code, LSOA name, Last outcome category and Context. Whilst it seems like we are dropping a lot of this data, for the type of analysis planned, this would not be extremely useful.

The important data from the database would be, Month, Latitude, Longitude and Crime Type. A few things stand out with these columns, having a lat long is good for plotting data onto a map, this could be explored with heat maps or 3d charts on a map. The month and crime type could be used to look at the type of crimes, count of crimes, how does the crime change overtime.

There are a few ideas which stand out from those columns, some will be explored in the next section of the report.

2 – App Development

2.1 – Script

After looking through the data in the last section of the report, this section will go over the development of the app (starting as a script or algorithm) and how it has developed overtime. Before starting with getting all the data, for a project this size it is a promising idea to start with one dataset (one CSV) and try to get a feel for starting the algorithm.

The initial algorithm process went something like this,

Begin

```
Import Module,  
Import Data,  
Read Data as a Data Frame,  
Only grab Month, Lat, Long, Crime Type  
Sum Total Crimes  
    print Total Crimes  
Sum Total Crime Per Type.  
    Print total crime per type
```

End

Whilst this is quite a big task for the beginning, it is important to know that a lot of this was covered previously. But after getting used to Python and how it can be used to analyse data, it started to take shape and there was potential to import multiple files. This is where there were a bunch of modules to help with this task. Glob was used to import multiple files with for example '*.csv' which would import all files which ended with .csv.

This was then used for each of the years within the dataset, starting with 2017. Which then stored all the file names which had '2017*.csv' and stored them in a list. This list was then used to loop through each of the names and read the csv files into the next module, pandas data frame which will be referred to as 'df.' Pandas df's were useful here to store like an excel file within python, which made exploring and manipulating the data a lot easier.

After getting all the files in a df, the next step was to only take the data which was required. This turned out to be month, lat, long and crime type, whilst the initial idea was to use the crime type, the other data was picked for later exploration.

The next step here was to get a total crime count for 2017, with pandas this was made quite easy by referencing the df, then the column within the df and a function called count(). This count would simply count all the entries within that df. After getting this to work, the next initial idea was to count the crime types. This was done by creating a dictionary of the crime types within the df and counting each of the values with this dictionary compared to the original df, which was then printed out to the command line displayed each crime then the count.

The next initial idea was to plot some of the data on a chart using matplotlib, the idea behind this was to plot the number of crimes and the count. Whilst prior experience helped with creating a plot,

it can also prove to be difficult depending on the data at this point. Whilst getting a plot to work, this has opened the other types of plots which could be used for the data.

One was to have a normal bar plot which was straight forward to get to grips with, whereas the pie chart proved to be more difficult. Due to this being the case, a pie chart was only used for the last dataset (2020) to get a little feel for how it could be used.

After getting this to work, the idea of doing it again for the other types of years was not so daunting. Whilst there is a clean and clever way to do it as a loop, this was the first-time using python and would have a few not so elegant ways of doing things. After getting all of 2017 sorted into functions, this layout was used to re-create the functions for the rest of the years.

2.2 – Graphical User Interface (GUI)

After getting the main bulk of the code down, next was the challenge of learning how to get a GUI and implement the initial code with it. For this part there is a module called Tkinter, whilst the documentation is bad for the module, there were other 3rd party websites which gave good examples which were referenced a lot.

Starting the GUI proved to be difficult, but after getting the main window started and finding out how to put some other labels and buttons on the main screen, it started to come together quite quickly.

The rest of the GUI development was focusing on getting a title, some information on some labels and figuring out how to get buttons to run functions on click. Whilst other experience in programming with GUI's was here, using it within Python was quite different.

One of the biggest issues was with the plots, as these needed to be global variables to allow the other functions the ability to call them, after changing these variables to global getting year 2017 it allowed for an easier way of plotting the data. Whilst this works, there is a small bug which only worked after getting the total crime count, to show this there was some try catch methods used, which will show a pop-up screen if there is an error.

After getting several buttons to work, there was some development on getting a new window to appear with the crime type and count, this was a bit difficult to understand until getting stuck in and that is the best way to figure out the code sometimes.

After getting to grips with the GUI and getting the buttons to call the already coded functions, the development stage here started to fly by. This was due to a lot of the code already used could be reused for the different years.

The key features which stand out for the GUI are the use of multiple buttons and labels, which were simple widgets used with a more advanced button being the quit and destroys button (quit closes the entire applications, whereas destroy only closes the one windows). There was also a little pop up which was implemented with the use of a built-in function as a message box, this appears when there is a bug within the program.

This has been the main development process when creating the GUI and trying to implement it with the rest of the code which was carried out before starting the GUI.

The main design idea here was as follows:

Begin

 Load main window (root)

 if button pressed

 print x

 else

 break.

 if plots pressed

 open plot

 else

 break.

End

Where most of the actions are waiting for an input from the user.

3 – Conclusion

Whilst a lot of Python programming was new to me, there were enough resources online and from the tutorials prior to carrying out the development to see me through. Whilst there were a lot more ideas going through my head whilst carrying out the development, it seems with the issues that were popping up, a lot of these ideas had to be put on hold. One of the main ideas was going to be mapping the data, which was explored using Jupyter Notebook, whilst it is great, the map does not work when using Spyder or a GUI.

Overall, the assignment was fun, but I may have spent a lot more time on the GUI than I should have.

4 - Software

Anaconda Navigator - 1.10.0

Spyder - 4.1.5

Python – 3.8.5

Modules:

glob2 – 0.7

matplotlib – 3.3.2

pandas – 1.1.3

tkinter

VS Code – 1.15.1

5 - References

Data Police, 2020, *Data downloads*, Available at: <https://data.police.uk> (Accessed: 23/10/20)

Glob, 2020, *glob* — *Unix style pathname pattern expansion*, Available at: <https://docs.python.org/3/library/glob.html> (Accessed: 23/10/20)

Matplotlib, 2020, *Matplotlib: Python plotting*, Available at: <https://matplotlib.org/> (Accessed: 23/10/20)

Open Government Licence (OGL), 2020, *Open Government Licence*, Available at: <https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/> (Accessed: 08/12/20)

Pandas, 2020, *pandas – Python Data Analysis Library*, Available at: <https://pandas.pydata.org/> (Accessed: 23/10/20)

Tkinter, 2020, *tkinter* — *Python interface to Tcl/Tk*, Available at: <https://docs.python.org/3/library/tkinter.html> (Accessed: 01/12/20)

Tutorials Point, 2020, *Python - Python - GUI Programming (Tkinter)*, Available at: https://www.tutorialspoint.com/python/python_gui_programming.htm (Accessed 01/12/20)

6 – Appendix

6.1 – GUI 1



6.2 – GUI 2



6.3 – GUI 3

