Data Analysis and Visualisation

2810ICT/7810ICT — Software Technologies

School of ICT

Griffith University

Trimester 2, 2019

Due at the end of Week 12, Sunday 6th October, midnight

Assignment Description

This assignment will test your ability to use python to interact with databases, excel workbooks, perform data analysis using NumPy, and use matplotlib to generate graphs. Raw data will be provided and will need to be stored in a database. You will then need to write scripts to query the database and pull out subsets of the data. These subsets will then need to be further analysed and have graphs procedurally generated to display relevant information. All of the investigations should yield results that you can present in a professional report.

This is an individual assignment.

It is important to note that submission of this assignment is a requirement for passing the course. Late submissions will be marked according to Griffith University's assessment policy. 5% of the overall mark will be deducted for each business day late. After 5 days, no submissions will be accepted.

Submission Requirements

This assignment must be submitted online via L@G under the assessment page. Your submission should include;

- A word document (preferred), PDF Is also acceptable, other formats are not allowed. This should contain your results presented as a scientific report.
- .py files containing your code. You should submit a separate .py file for each problem.
- A readme.txt file for your scripts

You can submit a zipped file with all the above documents.

Problem Statement

As more and more industries are becoming data driven, being able to process a large volume of raw data and produce a concise and insightful summary is becoming more and more important. As a data consultant for a government agency, you are tasked with processing some food inspection and health violation data and producing a report summarising some of the information.

The raw data is provided in 2 excel spreadsheets: (A) Inspections.xlsx and (B) Violations.xlsx. You will need to complete the below tasks and present your results in a report.

For each python script, you should handle the case where the script has already been run and therefore the data already exists. This could mean checking to see if the table already existed in the database, or a specific workbook/worksheet already exists. In each case, you should decide what to do (display error? Create a book/sheet with a different name? Delete the existing version and re-run the script?).

You should write a readme.txt file to accompany your scripts. Prepare a brief usage guide, any requirements and assumptions, and document what each script does and any other important info (for example; how does each script deal with database tables/excel sheets already existing).

Task 1 – Access the workbooks and create a database

Create a Python script (createdb_food.py) to perform the following tasks:

- Open the excel files.
- Create a SQLite database with two tables, one for each excel file. Each column in the excel files should correspond to a column in the tables. Make sensible decisions for attribute types.
- Import the data from the excel files to the corresponding tables in the database.

Task 2 – Query the database

Create a Python script (sql_food.py) to perform the following tasks:

- List the distinctive businesses that have had at least 1 violation ordered alphabetically to the console and then write their name, address, zip code and city into a new database table called “Previous Violations”.

Print a count of the violations for each business that has at least 1 violation to the console along with their name ordered by the number of violations. *SQL Hint: Group By*

Task 3 – Excel via Python

Create a Python script (excel_food.py) to perform the following tasks:

- Create a new workbook named “ViolationTypes.xlsx”.
- Create a sheet named “Violations Types”.
- Query the database and calculate the number of each type of violation based on violation code.
- Write the relevant data into the worksheet you created. This should show the total number of violations, then list how that is broken down by violation code, including the description of the violation code. For example:

Code	Description	Count
F001	Dirty Floors	300
F002	Rotten Food	135
	Total Violations	435

Task 4 – Numpy in Python

In this task, we are interested in analysing the data points over the time period covered. You will need to create a Python script (numpy_food.py) to perform the following tasks:

- Use MatPlotLib to create a plot with the following data:
    - The number of violations per month for the postcode with the highest total violations
    - The number of violations per month for the postcode with the lowest total violations
    - The average number of violations per month for ALL of California (ALL postcodes combined and averaged). For example, If postcode 1111 has 5 violations during July, 2222 has 4 violations during July, and 3333 has 3 violations for July, then the average violations in July is 4 (12 violations/3 postcodes)

- Use MatPlotLib to create a plot with the following data:
    - The average number of violations per month for all McDonalds compared with the average number of violations for all Burger Kings. ***This will require a new query as it is not grouped by postal code.*** If there were 3 McDonalds stores with 4, 5 and 9 violations for July, then the average for July would be 6 (4+5+9/3 stores).

- ***For 7810ICT students:***
    - Query the Violations table and retrieve all distinct violation codes and descriptions
    - Using Regular Expressions (NOT SQL), filter the resulting data to print out a list of all violation codes and descriptions that involve the word 'food'. You should add this violation listing as an appendix to your report.

For the above tasks, violations per month means to sum the violations for each month that there is data for (July 15 – December 17) - approximately 30 points of data. This provides a look at the food violation trends over time.

The SQL queries required to select the correct data are not trivial and need careful consideration. You may choose to use more complex queries that select more refined data, or simpler queries and handle more of the data processing in python.

Since the original dataset is huge, you may consider to use partial data in your programming and testing to save time. Once your program works correctly, you can run it on the complete dataset for data analysis.

Task 5 – Report

After developing all of your scripts, you need to take the information and present it in a brief executive-style report. You may use the provided template as is, modify it, or come up with your own. Do not simply paste data from the console/excel into each section in the template – present the data in a professional manner.

Marking

This assignment is worth 30% of your final grade. The assignment will be marked out of 100 and marks will be allocated as follows:

- Task 1 - Creating the database and importing the data (20 marks)
- Task 2 - Querying the database (15 marks)
- Task 3 - Excel in Python (20 marks for UG, 15 marks for MA)
- Task 4 – NumPy and Matplotlib (25 marks for UG, 20 marks for MA)
- Additional tasks for MA (10 marks)
- Task 5 - Report - (20 marks)

Additionally, marks will be deducted for incorrect:

- Poor English/grammar

- Lack of comments in the code
- Poor presentation