COMP60711 - DATA ENGINEERING

Coursework Description

Week 1

Coursework Description

1. Data Description

You have been given access to two road-traffic data files in CSV format, **'rawpvr_2018-02-01_28d_1083 TueFri.csv'** and '**rawpvr_2018-02-01_28d_1415 TueFri.csv**'. Each file contains observations collected via Inductive Loops sensors planted on a particular site of Chester Road in the city of Manchester. For example, file **'rawpvr_2018-02-01_28d_1083 TueFri.csv'** contains observations collected from site 1083, while file '**rawpvr_2018-02-01_28d_1415 TueFri.csv**' contains observations collected from site 1415. The observations were collected during the month of February 2018. Note that each row of the file contains one observation, i.e., the properties associated with one detected vehicle. As a consequence, if you count the total number of records with the following timestamp (06/02/2018), you are able to estimate the total volume of traffic (i.e., the total vehicle count) on the 6th of February 2018.

Both data files present the same structure, composed of the following attributes or properties:

- ***Date*** is a timestamp containing day of the month and year and time of the day when a vehicle was detected, with the following format: dd/mm/yyyy HH:MM:SS.
- ***Lane*** is an identifier of a given lane of the road (a road may have multiple lanes and each lane has a unique identifier).
- ***Lane Name*** is the name given to a particular lane of the road. Each lane has a unique name.
- ***Direction*** identifies the direction followed by a road lane (e.g., North, South, etc.). Different lanes may follow the same direction.
- ***Direction Name*** is the name of the direction followed by a lane of the road.
- ***Speed (mph)*** is the speed with which the detected vehicle was moving at the time it was detected.
- ***Headway (s)*** is the time distance between two consecutive vehicles following the same route. More precisely, it is the time distance between the front bumper of one vehicle and the front bumper of the vehicle behind it.
- ***Gap (s)*** is also a time distance between two consecutive vehicles following the same route, but it indicates the time distance between the rear bumper of one vehicle and the front bumper of the vehicle behind it.
- ***Flags*** is a number that identifies the day of the week when a vehicle was detected.
- ***Flag Text*** is the text description of the day of the week when a vehicle was detected.

2. Developing your Coursework

This course unit's coursework is composed of a list of tasks, divided into three sub-lists. This week's sub-list is as follows:

Sub-list 1: This is the first sub-list and contains a single task (described in Section 3), which does not carry any marks, and so, it is NOT MANDATORY and so it does not need to be submitted via Blackboard or any other platform. However, you are free to develop it and self-assess it according to criteria that we will provide via a rubric and model answers.

The main aim of this task is to give you the opportunity to become familiar with the dataset, use case, the tools and programming language (PL) you are going to use to develop the course unit's exercises in the weeks to come. We suggest a few tools for you try and later choose from, to develop the remaining coursework for this course unit, as well as Python, as main PL. These are available from the Department's machines and can also be accessed from home, if you feel like installing them in your own machine. Please, go to Section 4 of this document to see the list of suggested Data Preparation/Analysis tools.

We are aware there are other tools and PLs you could use, but we would advise you to focus on the list we provide, because the tools and PL found on the list are widely used and more familiar to the people assessing your work. Also, some automatic marking of code may be used, which can only mark Python code.

To develop this first task, we would like you to use Python (PL), and choose one of Knime, OpenRefine and the MySQL Database Management System (as Data Preparation and Analysis tools). As you may be able to anticipate, to use MySQL, you need to transform the CSV file into a relational table and use SQL.

It may seem strange to you our suggestion to develop the same task 4 time using these different technologies. There are a few reasons: we want you to (1) have minimal experience with all of them, as part of the skill set we would like you to develop; (2) develop an intuition of which of these technologies seem more appropriate for this simple task and which would probably be more suitable for more complex tasks; (3) chose your favourite tool(s) for the remaining coursework for this course unit; and (4) self-assess your first coursework, knowing that any loss of marks will not affect your final mark for this course unit. The task is very simple and does not take much time to be completed (a few minutes for someone already familiar with the tools). The hardest part is really to try to use the tools you have never used before, following Web-available instructions and tutorials.

When developing the coursework, you should do the following:

1. Use file '**rawpvr_2018-02-01_28d_1083 TueFri.csv**' for all tasks, unless you are explicitly told in the task to use both files, **'rawpvr_2018-02-01_28d_1083 TueFri.csv'** and '**rawpvr_2018-02-01_28d_1415 TueFri.csv**'.

2. When asked to use a technology of your choice to develop your work, use the one you are most familiar with from the suggested ones in Section 4, to avoid steep learning curves.

3. Provide explanations of your Python code in the form of comments that you place within the code itself. You can add a paragraph of comment on top of one line or a group of lines of code that you deem to be associated (by functionality).

Identify and justify any data preparation steps.

For each comment, use a maximum of *50 words* of text. If the code plots a graph, then make sure that the plotted graph is added to a Word file, and that it is referred to in the code comments, using a distinct number that you are going to give to each graph/plot you generate. You should also be able to refer to these graphs/plots from any text-based essay-like answers you provide.

4. Do not ***arbitrarily*** discard/delete rows in the data files provided to you, unless explicitly required in the task question.

5. A model answer and marking rubric will be provided for you to self-assess your work. By self-assessing your work via a rubric provided by us combined with a possible solution/answer to the tasks described in Section 3, you will get to see some of the relevant requirements for developing your future coursework assignments as well as the coursework style.

3. Task Description

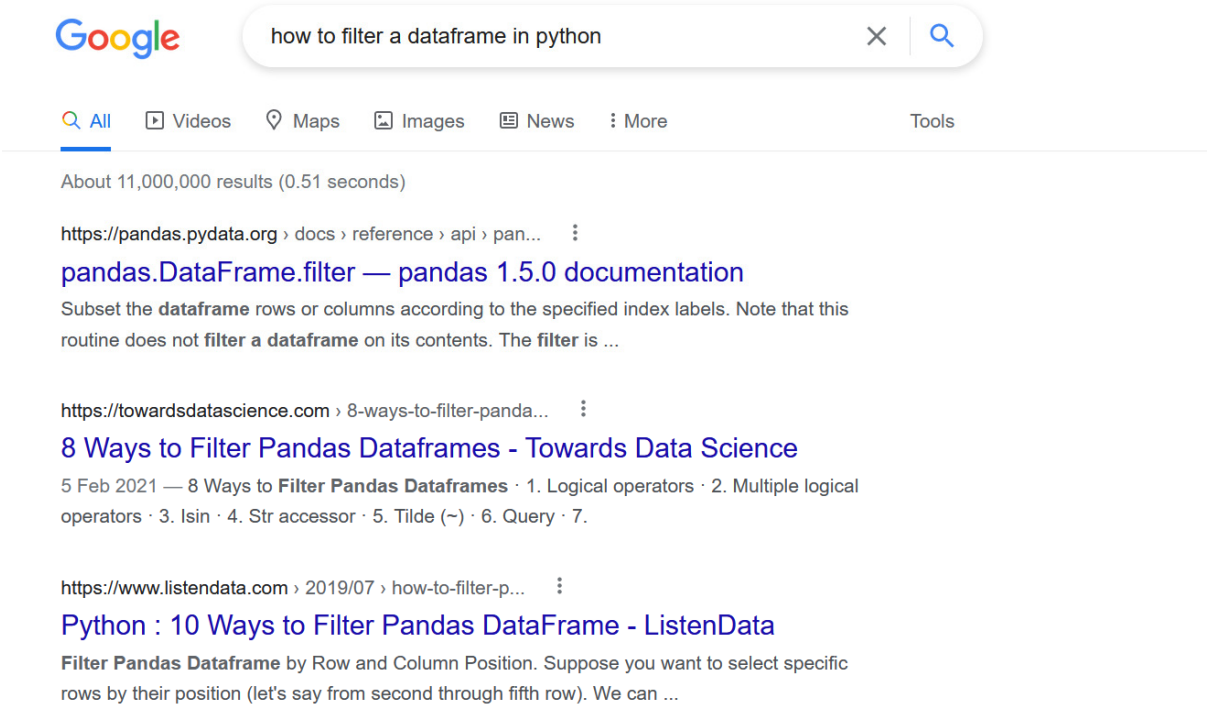
Week1	
Task1	A. Update the 'Flag' and 'Flag Text' columns of the file by creating an index for each day of the week. For example, rows where the observation date falls on a Tuesday should have, as value for 'Flags', 2, and for 'Flag Text', 'Tuesday'. B. Calculate the total traffic volume per weekday. Note that the total traffic volume should not be placed anywhere in the data file, but merely calculated/estimated. C. Answer the following question: Was any Data Preparation step necessary when performing A and/or B? If so, identify the Data Preparation step(s) in the PL code you developed (in the form of comments), explaining why it is a data preparation step. If not, then explain why no data preparation step was needed. **Output**: For A: (1) The total traffic volume of each per day of the week + a screenshot of the updated file, as it is output from the PL code you developed (there is no need to show the outputs obtained through the use of the other suggested tools). For B: (2) A step-by-step description of the development of the task using EACH of the data preparation tools (not the PL), associating with each step an explanation as to why you chose to execute certain function(s), while emphasising the functionality behind each function. For C:

	(3) An answer to the question.

4. Further Advice

The following list contains the PL and Data Manipulation/Preparation tools we suggest you use in the development of your coursework: *Python, Knime, OpenRefine and MySQL*. These are of easy access and free installation. You can access them from the machines in lab, or you can just download and install them in your machine.

If you are not familiar with the suggested programming language and/or tools, then you can search for commands in the Web, as shown below:



If using Python, for example, you will be interested in using commands from packages such as *pandas*, *numpy*, *datetime*, *os* and *calendar*, to handle *Date* related data types.

General tutorials for each of these can be found from the following links:

- **Knime** (https://www.knime.com/downloads/download-knime)
  Documentation: (https://docs.knime.com/)
  Tutorials: (https://www.youtube.com/watch?v=HEp9Cbql2hs, https://www.youtube.com/watch?v=5WAyOiIfHPg)

- **OpenRefine** (https://openrefine.org/download.html)
  (https://www.youtube.com/watch?v=WCRexQXYFrI),
  (https://www.youtube.com/watch?v=wGVtycv3SS0),
  (https://www.youtube.com/watch?v=wfS1qTKFQoI)

- **Python (**https://www.tutorialspoint.com/python/index.htm **)**

- **MySQL (**https://www.mysql.com/downloads/ **)**
  (https://www.tutorialspoint.com/mysql/index.htm **)**