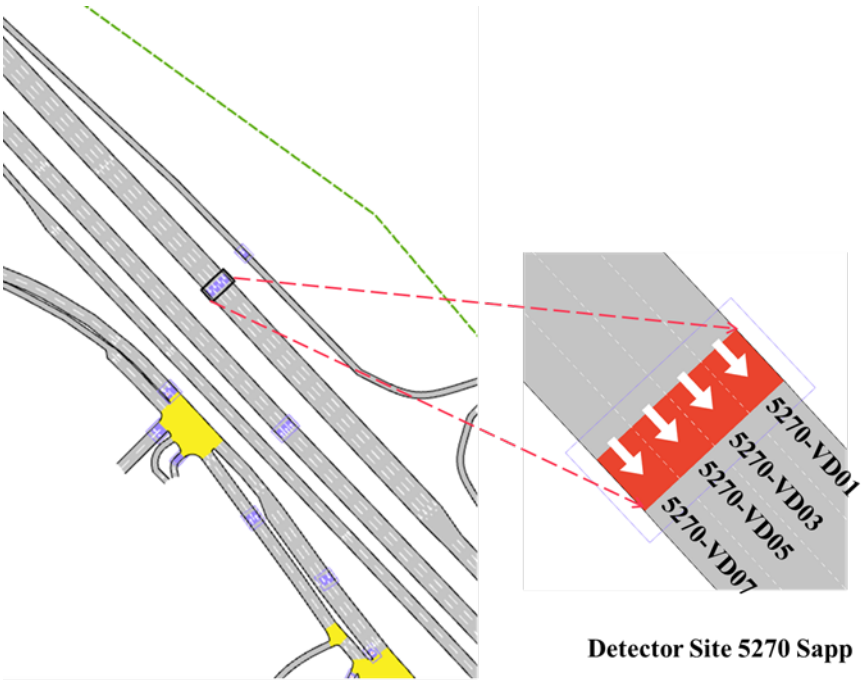
EGB103 Assignment 2
Data Processing using Python and Pandas

Individual only (strictly no group work or collaboration allowed)
Due: Friday 7th October (Week 10)
Worth: 30%

In this assignment you will create a Jupyter notebook and use Python and Pandas to analyse and visualize traffic data from the M3 motorway between Brisbane and the Gold Coast:



A Site from Logan City Road Network

You have been provided with a CVS file named 5270app.csv [1] which you must load and analyse using Pandas.

The road network is widely equipped with inductive loop detectors to monitor the flow of traffic. These are metallic loops installed under the road surface and work on the principle of induction. They are calibrated to detect the movement of vehicles. Each lane on the motorway has a separate detector to measure traffic travelling in that specific lane. Each row in the data is from a particular detector and covers a 1-minute period of time. The columns in the CSV file are as follows:

- **Time_AEST:** start time of the 1-minute period (in Australia Eastern Standard format)
- **Detector_ID:** Unique ID of the detector
- **Volume:** The number of vehicles detected over the loop during that 1-minute time interval
- **Occupancy:** The percentage of time the loop was occupied with a vehicle. For instance, during the one-minute detection, if the loop was occupied with a vehicle for 40 seconds, then the occupancy is 66.7
- **Speed:** The speed of the vehicles in km/h.

[1] The data is provided by Queensland Department of Transport and Main Roads and should be only used for research and teaching purposes.

The provided Solution Template provides a skeleton for what your Jupyter notebook should look like (with some plots deliberately hidden). You need to generate all of the 13 graphs listed (in the order specified) and you should aim to make them look as similar as possible to the provided plots (i.e., types of plots, titles, labels, legends).

One of the first analysis steps required is to combine the data from each lane detector to provide a summary of the traffic across all lanes at each point in time. For example, at time 2021-05-01 00:01:00 we have:

[9]:

	Time_AEST	Detector_ID	Volume	Occupancy	Speed
1	2021-05-01 00:01:00	5270-VD01	0	0.0	NaN
44615	2021-05-01 00:01:00	5270-VD03	8	3.3	92.625
89229	2021-05-01 00:01:00	5270-VD05	8	3.4	94.750
133843	2021-05-01 00:01:00	5270-VD07	2	0.7	99.500

To summarize traffic across all lanes during that time period we take the "*average*" of each measure. The average Volume is just the arithmetic mean $\frac{0+8+8+2}{4} = 4.5$ vehicles per minute per lane. Similarly, the average Occupancy is simply the arithmetic mean $\frac{0.0+3.3+3.4+0.7}{4} = 1.85\%$. For the average speed we can't simply compute the arithmetic mean because each lane has a different number of vehicles. So, we could instead compute a *weighted arithmetic mean* where the Speed is weighted by the Volume: $\frac{0 \text{ x } \text{NaN}+8 \text{ x } 92.625+8 \text{ x } 94.75+2 \text{ x } 99.5}{0+8+8+2} = 99.3333\ km/h$. That would be better, however, because the speeds that we averaging are "*rates*" (speed = kilometres travelled per hour), so traffic engineers tend to instead *average* speed data by using the *weighted harmonic mean* (see here for an explanation of why it is used for average speed). In our case the traffic Volume is used as the weights and the Speed is the variable being averaged. This same process (of using harmonic mean) applies to both averaging the speeds from all lanes for a given 1-minute time interval, as well as for averaging Speed data over longer time periods such as hours or days.

One of your first tasks is to create a Python function that will compute the *weighted harmonic speed* for a group of rows. Your function should be implemented using Pandas methods and operations rather than using low level Python loops. The provided Solution template illustrates how this harmonic mean function can then be applied to both average all speeds associated with each detector and to create a combined lanes data frame (with one row per time period).

1. The first set of graphs then visualizes this combined lane data, but only for the week from 2021-05-10 to 2021-05-16.
2. We then visualize each of our variables by day of week using box plots.
3. Next, we analyse these variables for each hour of the day, showing the averages for each day of the week as well as for all days combined using line plots.
4. To analyse the relationship between variables we start with a simple scatter plot of Volume vs Speed for the combined lane data.
5. In the next set of line plots, we explore the relationship between Volume, Speed and Occupancy by grouping the variables shown on the x-axis into fixed sized bins (just like we did in assignment 1). For example, in the Volume vs Speed plot, the first bin covers the Speed range from 0 to 10 km/h, the second from 10 to 20 km/h etc. The y-axis then shows the *average* for each bin. The plots show the relationships between these variables for each lane/detector separately, as well as for the combined data.

Each plot must be followed by markdown to summarize your observations/conclusions for that plot.

Your code should be written so that it works unchanged on a different data set (which may have a different number of lanes) by simply changing the csv file name imported. i.e., don't hard-code the values of the detector-ids used at this particular detection site.

Data Cleaning

In the Speed vs Occupancy plot you may notice some outliers (with high Occupancy and Speed). Our transport engineer believes these may be detector malfunctions. You have been asked by our transport engineer to retrieve the raw data that led to those outliers on the graph, and to develop a method to "*clean*" the data of those faulty readings (if the engineer decides based on your report that they are actually detector faults).

Requirements

Everything should be included in a single Jupyter notebook. The Python code included should follow best practices as outline in the lectures, including using well chosen identifier names, writing clear simple code, and not repeating yourself [2]. All data processing should be done using the Pandas library and should make use of the following Pandas features:

- Reading input data files.
- Parsing dates.
- Filtering rows by condition.
- Three different kinds of plots (line, scatter and box)
- Group By to explore relationships between variables.
- All plots should be appropriately titled, and axis appropriately labelled (as per solution template).
- User friendly axis labels, e.g., Mon, Tue, Wed (as per solution template).
- Appropriately sized figures that are large enough to easy read.
- Binning of values so as to investigate relationships between variables.
- Using Python functions to compute aggregate information for groups of rows.

All of these features are covered in the lectures and/or practical exercises, so there should be no need to use Python or Pandas features outside of what has been covered in class. Note, you will get Zero marks if you use some other programming language, library, or system such as R, MATLAB or Excel.

[2] Don't repeat yourself means creating Python functions so as to avoid repeating the very similar patterns of code for different plots