COMP20008 Project Phase 1

V1.0: 15th March 2017

Due Date

- Phase 1 (15%): 11:59pm 6th April. Submission is via the LMS.

Phase 1: Warmup - Python Exercises

In this phase, you will practice your Python data cleaning and visualisation skills with a publicly available dataset. The dataset includes information about New York City Yellow Taxi trips[1]. The trip include features recording pick-up and drop-off date/time, trip distance, payment amount, and passenger count. Table 1 shows the description of each feature.

Field Name	Description
pickup_datetime	The date and time when the taxi meter was engaged
dropoff_datetime	The date and time when the taxi meter was disengaged
trip_distance	The elapsed trip distance in miles reported by the taximeter
passenger_count	The number of passengers in the vehicle (a driver-entered value)
payment_amount	The total amount charged to passengers. Does not include cash tips

Table 1: Summary of Feature Description

You will be working with three sets of data in this phase:

- *raw-january.csv:* a sample dataset of around 100k trip records in January 2016. You need to use this dataset to answer questions 1.1 to 1.4
- *raw-june.csv:* a sample dataset of around 100k trip records in June 2016. You need to use this dataset to answer questions 1.1 to 1.3
- *clean-january.csv:* a different set of trip records (approx 89k) in January 2016 for which data cleaning has been applied to eliminate noisy or inconsistent values and to which a duration field has been added. You need to use this dataset to answer questions 2.1, 2.2 and 2.3

Libraries to use are Pandas and Matplotlib. You will need to write Python 3 code and work with Series and DataFrames discussed in workshop 1 (week 2) and data cleaning and basic visualisations that will be discussed in workshops 2 and 3 (weeks 3 and 4). If you are using other packages, you need to provide an explanation in your code about why it is necessary.

[1] https://www.kaggle.com/nyctaxi/yellow-taxis

1 Question 1 - Data Preprocessing 9 marks

For this section use the raw datasets: raw-january.csv, raw-june.csv.

1.1 Adding a new column to the table (2 marks)

Read each file of raw data into a pandas dataframe. For each dataframe, using the values in the *pickup_datetime* and *dropoff_datetime* columns, compute the duration of each trip (in minutes) and record this information in a new column called 'duration'. After transforming *pickup_datetime* and *dropoff_datetime* columns to duration, create two new DataFrames (one for January and one for June) that each have the following schema and then use these to perform the rest of the exercises in Section 1.

Index	Duration	Trip_distance	Passenger_count	Payment_amount

tip1: pandas.to_datetime() converts a string to a datetime object.
tip2: The time and date information are formatted as '%d/%m/%y %H:%M'.

1.2 Basic statistics for each column (1.5 marks)

What is the mean trip distance, mean trip duration and mean trip passenger count in January? What is the mean trip distance, mean trip duration and mean trip passenger count in June? Your code should print out the results with the following format:

```
****************************** STAT ******************************
January: ###
   June: ###
******************************************************************
```

Where STAT is the title of the statistic that you are reporting and ### is its value.

1.3 Visualisation using boxplots (2 marks)

Draw three plots called *Duration*, *Distance* and *Trip Fare*. Each plot should consist of two boxplots, one boxplot to show the distribution of one of the above 3 features in January, and one boxplot to show the distribution of the same feature in June.

tip: pandas.DataFrame.boxplot() draws a boxplot for each column of the DataFrame object

1.4 Data Imputation (3.5 marks)

Using the raw data from *raw-january.csv*, for each feature, i.e., Duration, Trip_distance, Passenger_count and Payment_amount:

1. Choose a 'normal' range for its values; for example a normal range for a person's age could be

$$R_{age} = [1, 99]$$

Via comments in your code, record your choice for each range and justify in 2-3 sentences.

2. Compute the ratio of values that do not fall within your defined normal range; for example, using a hypothetical example of a person's age, we could compute the ratio as:

$$Ratio = \frac{(number\ of\ instances < 1) + (number\ of\ instances > 99)}{total\ number\ of\ instances}$$

The output of this step should be a message for each of the four features, printed with the following format:

```
********************************************************************************
XXX% of instances in 'ColumnName' are noisy! (normal range is between AAA and BBB)
********************************************************************************
```

Where ColumnName is among the four above-mentioned features, XXX is the computed ratio and AAA and BBB are the lower and upper bounds of your chosen normal range.

3. Write code that will replace each of these noisy values by the mean of the given column.

2 Question 2: Analysis for a clean dataset 6 marks

For this section use the clean dataset: clean-january.csv

Hint: for these questions consider making use of pandas DatetimeIndex and groupby functionality.

1. (1 mark) Read in the file and create a new column *isWeekend* that indicates whether a trip was started (i.e. the meter was engaged) on a weekday or weekend. It should have value 0 for weekday and value 1 for weekend. Calculate and print out the percentage of weekend trips.

   The output of this step should look like

   ```
   ********************************************************************
   % of weekend trips=XXX
   ********************************************************************
   ```

   where XXX is the value you calculate.

2. (3 marks) Create a new column called *hour* which records the hour in which the trip began (i.e. the meter was engaged). It should have value 0 if it was begun 0:00-0:59, value 1 if it was begun 1:00-1:59, ..., value 23 if it was begun 23:00-23:59.

   Now plot two histograms showing the frequency of taxi trips over hours of the day. One histogram for weekends and one histogram for weekdays. The x axis of each histogram should use 6 bins with the ranges [0-6),[6,9),[9,12),[12,16),[16,20),[20,24). The y axis of the histogram should show the frequency.

3. (2 marks) Create a new column called *income_efficiency*, which for each trip is equal to the payment_amount divided by the trip_duration. Create a bar plot showing the mean income_efficiency (y axis) for each hour of the day (x axis). The x axis should label each of the bars: 0, 1, 2, ... 23.

Marking scheme

- Labelling provided on plots and figures
- Correctness of code (indentation, logic, etc)
- Completeness of the answers to questions
- Code modularity and flexibility
- Code commenting and clarity
- Comprehensiveness of discussions for 1.4.1

Submission Instructions

Submit a jupyter notebook (Strongly preferred. An empty notebook "notebook-for-answers.ipynb" is provided in the folder with the datasets) or a ".py" file (less preferred) with the code. The material should be submitted through the LMS.

Other

Extensions and Late Submission Penalties: If requesting an extension due to illness, please submit a medical certificate to the lecturer. If there are any other exceptional circumstances, please contact the lecturer with plenty of notice. Late submissions without an approved extension will attract a penalty of 10% of the marks available per 24hr period (or part thereof) that it is late. E.g. A late submission will be penalised 1.5 marks if 4 hours late, 3 marks if 28 hours late, 4.5 marks if 50 hours late, etc.

Phase 1 is expected to require 16-19 hours work.

Academic Honesty

You are expected to follow the academic honesty guidelines on the University website `https://academichonesty.unimelb.edu.au`

Further Information

A project discussion forum has also been created on the subject LMS. Please use this in the first instance if you have questions, since it will allow discussion and responses to be seen by everyone. The Phase 1 project page will also contain a list of frequently asked questions.