COMP63301-Data Engineering Concepts Coursework 1

Instructions

This coursework is **100% formative**, meaning it is designed to support your learning and skill development.

- **Marks awarded** are for **feedback purposes only** and **do not contribute** to your final course grade.
- You are strongly encouraged to engage fully with the tasks, as they align closely with the learning outcomes of the module and will prepare you for summative assessments.

This coursework addresses key activities within the Data Engineering Lifecycle, specifically:

- Data Exploration & Profiling: Understanding the structure, patterns, and quality of raw time series data.
- Data Quality Improvement: Applying basic techniques to assess and enhance data completeness.
- Data Summarisation: Using Python to generate descriptive statistics and summaries.

You will work with unrefined time series datasets, applying Python-based methods to explore, clean, and summarise the data.

All tasks must be completed using the Python programming language. Minimum Python version: 3.10 or higher. **You can use the ONLY the Python libraries specified in the code stub for this exercise**.

You are required to complete **five data-related requests**, each designed to simulate a real-world data engineering scenario. The requests involve reading and understanding the dataset, while applying appropriate Python techniques.

Submission Requirements

You are required to submit the single Python code stub provided for this exercise. This file will serve as the container for your solutions to each of the five data requests outlined in the coursework.

- Please ensure that your responses are written directly within the designated sections of the code stub, following the instructions and comments embedded in the file.
- Do not rename or restructure the file, as it is configured to work with our Gradescope automated marking system.
- The submission link is provided our Canvas space (where this coursework information is provided).

You may submit this coursework multiple times. However, we **strongly recommend** that you achieve a **successful submission**—one that is fully processed and marked by our

Gradescope automarker, with feedback provided—**well in advance** of attempting your next summative coursework.
This will allow you to:

- **Practise submitting your coursework** early to become familiar with the Gradescope platform.
- Ensure your submission is in the **correct format** (i.e., the original stub with your completed code).
- **Submit as many times as needed**—only your most recent submission will be considered.
- Aim to achieve a **successful submission** (i.e., one that is fully marked by the Gradescope automarker and returns feedback) **well before** you begin your next summative coursework.
- You receive timely feedback to inform your learning.
- You avoid technical issues or delays close to summative deadlines.

Need Help?

If you experience any issues with the code stub, the automarker, or the submission process

- Post your question in the course discussion forum.
- Attend the laboratory drop-in session of the following week.

Exercises

CSV Data Cleaning and Value Imputation using Python

1. The dataset contains a column named `end_location_name`, which includes missing (null/NaN) entries. Your task is to:
    a. Identify all missing values in the `end_location_name` column.
    b. Replace each missing entry with the placeholder string `'Stop St.'`, which serves as a meaningful default value.
    c. After performing the replacements, compute and return the total number of occurrences of the value `'Stop St.'` in the `end_location_name` column.

    Important:

    - Your final output should be a single integer representing the count.
    - Do **not** include any labels, column names, or additional formatting in your output.

2. The dataset contains a column named `start_location_name`, which includes missing (null/NaN) entries. Your task is to:
    a. Identify all missing values in the `start_location_name` column.
    b. Replace each missing entry with the placeholder string `'Start St.'`, which serves as a meaningful default value.

c. After performing the replacements, compute and return the total number of occurrences of the value `'Start St.'` in the `start_location_name` column.

Important:

- Your final output should be a single integer representing the count.
- Do **not** include any labels, column names, or additional formatting in your output.

Deriving Missing Values and Targeted Data Retrieval

3. The dataset contains a column named `DURATION`, which includes missing (null/NaN) entries. Your task is to:
   a. Identify all missing values in the `DURATION` column.
   b. Derive and fill each missing value using the difference between the corresponding `ended_at` and `started_at` timestamps.
   c. Ensure that the resulting `DURATION` values are formatted as strings and they appear in the `hh:mm:ss` format.

   Once the missing values have been filled, return the following:

- The total number of remaining `NaN` values in the `DURATION` column, as a single integer.
  Do not include any labels, column names, or additional formatting in this output.
- A table displaying the `trip_id` and `DURATION` columns for **only** the rows with the following `trip_id` values:
  `1821126`, `1821158`, `1821204`, `1821289`, `2047623`
- *Ensure that the column names (`trip_id` and `DURATION`) are included in the output.*

Geospatial Data Enrichment and Aggregation

4. The dataset `scooter.csv` contains geographic information in the form of location names but lacks coordinate data necessary for mapping and spatial analysis. A supplementary dataset, `geocodedstreet.csv`, contains coordinate information for various street names. Your task is to:

   a. Identify the five most frequent values in the `start_location_name` column of `scooter.csv`.
   b. Using these five values, perform a merge between `scooter.csv` and `geocodedstreet.csv` to enrich the trip data with geographic coordinates.
   c. After merging, return a table containing the following columns and their corresponding values and in the order shown below::

      1. `address`: the full address from `scooter.csv`
      2. `count`: the number of times each `address` value appears in `scooter.csv`
      3. `street`, `x`, `y`: columns from `geocodedstreet.csv` representing the street name and its coordinates, in this order.

Important:

- Ensure that the output includes the specified column names: `address`, `count`, `street`, `x`, `y`.
- Only include rows corresponding to the five most frequent `start_location_name` values.
- For the values to be returned of columns '`x`' and '`y`', ensure you use all the digits of the values, as they appear in file `geocodedstreet.csv`.

Statistical Summary of Trip Durations by Location Pair

5. The dataset `scooter.csv` contains trip data, including the columns `start_location_name`, `end_location_name`, and `DURATION`. Your task is to:

    a. For each distinct location pair ⟨`start_location_name`, `end_location_name`⟩, compute the five-number summary statistics for the `DURATION` column:
        - Minimum
        - First Quartile (Q1)
        - Median (Q2)
        - Third Quartile (Q3)
        - Maximum

    b. Prior to computing the statistics, ensure that all `DURATION` values are converted to **seconds**.

    c. Return results **only** for the first five location pairs that have **more than one trip** associated with them.

    Important:

    - All summary statistics must be reported in **seconds**. Use the nearest value rounding method (standard rounding), with a decimal precision of 2 digits, as shown in the following examples:
        - A value like 12.3456 should be returned as 12.35.
        - A value like 12.344 should be returned as 12.34.
    - The output should clearly indicate the location pair (start, end) and the corresponding five-number summary (min, Q1, Q2, Q3 and max).
    - The following labels should be used for each "value" returned in the results set, and in the following order:

        - start
        - end
        - min
        - Q1
        - Q2
        - Q3
        - max