COMP9321 25T1 Assignment 1 v1.2 (15 marks)

Changelog

V1.2 - 4th March 2025 – fixed output shape for Q5

V1.1 - 3rd March 2025 – clarified edge cases for Q4

V1.0 - 3rd March 2025 – released

Introduction

The NSW FuelCheck dataset is maintained by the NSW Government. It allows motorists to access historical and live information about fuel prices across NSW. We have downloaded the "FuelCheck Price History Jan 2025" file in February 2025 and are storing it locally for your use as **fuel.csv.**

A dataset of Australian postcodes, with their latitudes and longitudes, is also available from the user "Elkfox" on Github. This has been stored locally for your use in February 2025 as **postcodes.json.**

You're encouraged to review these and explore these datasets prior to attempting the assignment. Please note that this data is publicly available, and we are not responsible for political correctness, or other data inaccuracies, as this is purely a learning exercise for data services engineering. Accompanying each question below you'll find:

Output: This includes the expected shape of the DataFrame to return.

Marking Considerations: These, along with the expected output, are provided to help you stay on track. They serve as marking guidance to help you understand how we mark the assessment.

Code Template

You **must** use the code template and rename the code template with your zID.

You **must not** modify the code template, except where indicated. For example, you must not:

- Import additional third-party libraries.
- Modify the function signature for each question_X function.
- Modify the main function.
- Modify the log function.
- Disable or modify the calls to the log and plt.savefig functions within each question_X function.

- Add any global variables.

You **must** setup a virtual environment as per the provided requirements.txt and instructions here.

You **must** use either Python 3.11 (which is installed on CSE) or Python 3.13 (which is the current release).

You **must** use pandas features to solve all question.

You **must not** iterate over the rows of any DataFrame. For clarity, any `for` or `while` loop that runs proportionally to the number of rows in a DataFrame is iterating over the rows of that DataFrame. This also holds for any Series, for example, iterating over one column of a DataFrame. However, you **may** iterate over DataFrame.columns, if you feel it necessary.

You **must not** convert any DataFrame to a native data type (e.g. a list or dict) to process the data.

You **must not** hard code any file paths within the code you write. These are specified in the main function, which you must not modify, and are passed as parameters to the question_X functions that you will complete. You must use these local variables instead of hard-coded values.

You **must not** display any plots (e.g. using plt.show), as the code template is configured to save your plot to disk.

You **must not** manually edit any datasets. During marking, a copy of the original CSV and JSON files will be used.

You **may** import and use any of the Python 3.11 or 3.13 standard libraries. You may not use any third-party libraries outside of what is provided in the virtual environment setup.

You **may** write helper functions in the relevant code template section.

Part 1: Data Ingestion and Cleaning (4 marks)

Question 1: (1 marks)

Load the NSW FuelCheck dataset from the file **fuel.csv** into a Pandas DataFrame **df1**.

Handle the CSV input correctly, ensuring that erroneous data does not affect the structure of the DataFrame.

Hint: Investigate why an extra column appears without any data in the header.

Return the DataFrame as **df1**.

Output: (60151, 8)

Marking considerations:

- [0.5 marks] The data in the DataFrame is correct.
- [0.5 marks] The shape of the DataFrame is correct, including the header.

Question 2: (1 marks)

Perform the following on the DataFrame from Question 1 **df1**, and return a new DataFrame **df2** with the following:

- Rename the column header “ServiceStationName” to “Name”.
- Ensure the text in the “Suburb” column is all upper case.
- Remove any rows for addresses not in NSW.

Output: (59256, 8)

Marking considerations:

- [0.5 marks] The above cleaning steps are performed correctly.
- [0.5 marks] The remaining data is not malformed in any way.

Question 3: (1 marks)

Load the postcodes dataset from the file **postcodes.json** into a new Pandas DataFrame **df3**. Include all columns except for the “accuracy” column.

Return the DataFrame as **df3**.

Output: (16875, 6)

Marking considerations:

- [0.5 marks] The data in the DataFrame is correct.
- [0.5 marks] The header and removal of “accuracy” in the DataFrame is correct.

Question 4: (1 marks)

With the DataFrame **df2** as the basis, create a column called “Latitude”, and another column named “Longitude”. Populate these columns with the corresponding values as per the postcodes DataFrame **df3.**

Use the “Postcode” and “Suburb” columns in **df2** to match with **df3**.

Where such a match doesn't exist, instead match only on postcode.

Should that produce multiple matches, select the match whose suburb name is alphabetically first. For example, 2018 is shared by Rosebery and Eastlakes, so Eastlakes would be selected.

If there are no matches at all, leave the “Latitude” and “Longitude” fields blank.

Save the final DataFrame **df4** as a CSV file **df4.csv**. Do not write the row names (index) to the CSV file.

Return the final DataFrame as **df4**.

Output: (59256, 10)

Marking considerations:

- [0.5 marks] The data in the DataFrame is correct.
- [0.5 marks] The CSV file has the correct data and is named correctly.

Part 2: Data Exploration (2 marks)

Question 5: (2 marks)

Using the data in **df4**, create a new DataFrame **df5**, which will provide a summary of the average fuel price per postcode and fuel type.

df5 should have a hierarchical index, where “Postcode” is the primary index and “FuelType” is the secondary index.

The DataFrame should include the following:

- Index: Postcode
- Index: FuelType
- Column: AveragePrice

Return the final DataFrame as **df5**.

Output: (4048, 1)

Example:

Postcode	FuelType	AveragePrice
2088	**E10**	**123.45**
	PDL	**234.56**
	ULP91	**345.67**
...		
2090	**E10**	**456.78**
	PDL	**0.00**
	ULP91	**567.89**

The postcodes should be sorted numerically, and the “FuelType” sorted alphabetically within each postcode.

The final "AveragePrice" for each postcode and fuel type can be found by calculating:

- The average per-station and per-type daily fuel price, where there may be multiple fuel readings per day for a particular fuel type and service station, which are now aggregated to the service station level.
- The average per-postcode and per-type daily fuel price, where the daily averages for each service station and fuel type are now aggregated to the postcode level.
- The final average per-postcode and per-type fuel price, where the daily averages for each postcode and fuel type are now aggregated to a single value covering the duration of the dataset.

If there are days without a calculated value for a particular fuel type and postcode, exclude those from your final calculation.

In the final DataFrame **df5**, all “FuelType”s should be returned for each postcode, even if that fuel is not sold in that postcode. The “AveragePrice” in that case would be 0.00.

Note that the “AveragePrice” should be calculated to 2 decimal places (2 d.p.). In this case, 2 d.p. must always include all digits, even if they are .00.

Marking considerations:

- [0.25 marks] The columns have the correct headers
- [0.25 marks] The columns are sorted as requested.
- [0.5 marks] The indexes are correct.
- [1 mark] The “AveragePrice” calculation is correct and rounded to 2 d.p.

Part 3: Data Manipulation (2 marks)

Question 6: (1 marks)

Using the DataFrame **df4** as the basis, create a column called "PriceChangeAverage". Populate the column with a 2 decimal place (2 d.p.) percentage difference of that row's fuel price with the corresponding average fuel price from **df5**. Note that in this case, 2 d.p. must always include all digits, even if they are .00.

Return the final DataFrame as **df6**.

Output: (59256, 11)

Example:

Assuming the following row in the previous question from **df5**:

2088, PDL, 234.56

Assuming your data in **df4** contains the following row (some columns omitted for example's sake):

BP Mosman (9542),2088,PDL,207.9

Your **df6** value for the "PriceChangeAverage" column would be **-11.37.**

Marking considerations:

- [0.25 marks] The column is added correctly.
- [0.25 marks] The sign (-/+) and decimal places are correct.
- [0.5 marks] The calculated "PriceChangeAverage" values are correct.

Question 7: (1 marks)

Using the DataFrame **df6** as the basis, create a column called "PriceChangePrevious". Populate the column with the 2 d.p. difference of that row's fuel price, with the previous timestamp entry for that exact same fuel type and address. For the first row per unique service station and fuel type in the DataFrame, set the field as zero. Note that 2 d.p. must always include all digits, even if they are .00.

Return the final DataFrame as **df7**.

Output: (59256, 12)

Example:

Assuming your data in **df6** contains the following row (some columns omitted for example's sake):

Astron Grafton,"105 BENT ST, SOUTH GRAFTON NSW 2460",SOUTH GRAFTON,2460,ASTRON,P98,2025-01-01 09:09:09,184.9

Astron Grafton,"105 BENT ST, SOUTH GRAFTON NSW 2460",SOUTH GRAFTON,2460,ASTRON,P98,2025-01-01 10:09:09,188.9

Your **df7** value for the “PriceChangePrevious” column would be **4.00.**

Marking considerations:

- [0.25 marks] The column is added correctly.
- [0.25 marks] The sign (-/+) and decimal places are correct.
- [0.5 marks] The calculated “PriceChangePrevious” values are correct.

Part 4: Data Visualisation (7 marks)

In this section, the most effective visualisations provide deep insights into the dataset. You should also aim to achieve:

- Suitable choice of visualisation.
- Appropriate use of scale and colour.
- Appropriate inclusion of title, labels, legend, with suitable sizing.
- Visualisation is self-explanatory and informative.
- Image is correctly saved to disk, rather than shown.

Question 8: (3.5 marks)

Using the DataFrame **df7,** create a visualisation to answer the question: "Is it cheaper to patron independent or franchised service stations?"

Save your visualisation as **a PNG file (already setup in the code template)** and explain in a return variable **answer8** what insights you have uncovered from this visualisation.

Note: you may use the "Brand" column to distinguish between Independent and all other (Franchised) service stations.

Marking considerations:

- [1.5 marks] The visualisation provides deep insights into the research question.
- [1 mark] The visualisation is error free and of professional standard.
- [1 mark] The discussion has thorough and sound reasoning explaining how the visualisation answers the research question.

Question 9: (3.5 marks)

Using the DataFrame **df7,** create a visualisation to answer the question: "Are consumers in certain regions of NSW being unfairly charged for fuel compared to other regions?"

Save your visualisation as **a PNG file (already setup in the code template)** and explain in a return variable **answer9** what insights you have uncovered from this visualisation.

Marking considerations:

- [1.5 marks] The visualisation provides deep insights into the research question.
- [1 mark] The visualisation is error free and of professional standard.
- [1 mark] The discussion has thorough and sound reasoning explaining how the visualisation answers the research question.