Coursework 1

1 Introduction

This project implements a SQLite database system to manage and analyze energy consumption data across different regions. It handles both industrial-commercial and residential (domestic) energy usage data, tracking consumption patterns for various energy types including electricity, gas, coal, oil, and bioenergy.

2 Section 1.1 Data exploration

The dataset, London Energy and Greenhouse Gas Inventory (LEGGI), is an emissions inventory which quantifies greenhouse gas emissions released to the environment from activities in London. It is produced by the Greater London Authority (GLA) (, GLA). It is used under licence https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/.

In this part, we use Borough Energy consumption data in LEGGI 2022.xlsx as raw data. First I copy the frame to a new excel named Energy consumption.xlsx. Then I use pd.read_excel() to load it as a dataframe. I drop a useless row 'Year' before other data manipulations since all data is about the same year. Then I check for missing values in missing_data_check() function. I print the rows with missing values and find out that only one row named Unapportioned has missing values.

I explore the basic data structure in explore_data() function. First, I print the top 5 data and tail 5 data. Then I use df.info() to find out that the dataframe has 36 rows and 21 columns. All columns are in data type float64. I print out the index names next. There are 33 boroughs in London and the last 3 lines are not about the data in a specific borough.

I also use df.describe() to calculate statistics. I cannot use the whole dataframe because the last 3 lines are not about a borough. Hence, I use df.iloc[:-3,:] to drop the last 3 rows before I do df.describe(). I got the mean, std and percentile values of each column. For example, the mean of domestic electricity energy consumption is about 354059400. To explore domestic total energy consumption of each borough, I draw

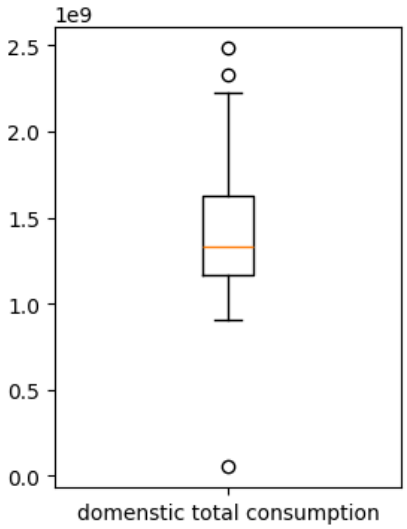


Figure 1: explore total energy consumption

a boxplot, figure 1. It is obvious that the total energy consumption of a borough mostly lies in range of $(1.0\times10^9, 1.7\times10^9)$.

3 Section 1.2 Data preparation

3.1 Target audience and 3 questions

Suppose our target audience are Urban Planners and Policy Makers. They are responsible for developing strategies to improve energy efficiency. The followings are questions they may ask.

- Which boroughs have the highest energy consumption? What is the main cause of it?
- Is there a correlation between electricity and other types of fuel in terms of energy consumption?
- Among all 33 areas, which type of energy consumption accounts for the largest share of the total energy consumption in the borough?

3.2 Question 1

Question 1 is: Which boroughs have the highest energy consumption? What is the main cause of it? The answer is Hillingdon and Transport.
First I sort values by Grand Total value and use ascending=False to display the most energy consumption borough on the top. It turns out to be Hillingdon. Then I drop useless columns to keep the total energy consumption of domestic, industrial and commercial, transport and total. I use df.div() to normalize each row by divide each column by the borough total energy consumption and get the percentage of energy. Transport accounts for 48.9% of consumption, which is the most. The prepared data is saved as q1.csv.

3.3 Question 2

Question 2 is: Is there a correlation between electricity and other types of fuel in terms of energy consumption? The answer is yes. The energy consumption of electricity and gas is positive correlated.
I use correlation_explore() function to explore the correlations between fuel types. I draw scatter plots, where x axis is electricity and y axis are gas, coal, oil,Bioenergy and waste*. Let's see figure 2 and figure 3, there is a positive correlation between gas and electricity while no other correlations are detected.

3.4 Question 3

Question 3 is: Among all 33 areas, which type of energy consumption accounts for the largest share of the total energy consumption in the borough? The answer is domestic.
I also use q1.csv, and draw a boxplot for the first three columns. This bar chart illustrates the percentage of different types of energy consumption in a borough, categorized into Domestic, Industrial and Commercial, and Transport. The y - axis represents the percentage of energy consumption, while the x - axis lists the three categories. From the chart, it is evident that the Domestic category has the highest percentage of energy consumption.

4 Section 2.1 Database Design

4.1 Tables and Attributes

4.1.1 1. Location Data Table (location_data)

Attribute	Data Type	Key Type
city_id	INTEGER	Primary Key
city_name	TEXT	UNIQUE NOT NULL

4.1.2 2. Sector Data Table (sector_data)

Attribute	Data Type	Key Type
sector_id	INTEGER	Primary Key
sector_name	TEXT	UNIQUE NOT NULL

4.1.3 3. Fuel Types Table (fuel_types)

Attribute	Data Type	Key Type
fuel_id	INTEGER	Primary Key
fuel_name	TEXT	UNIQUE NOT NULL

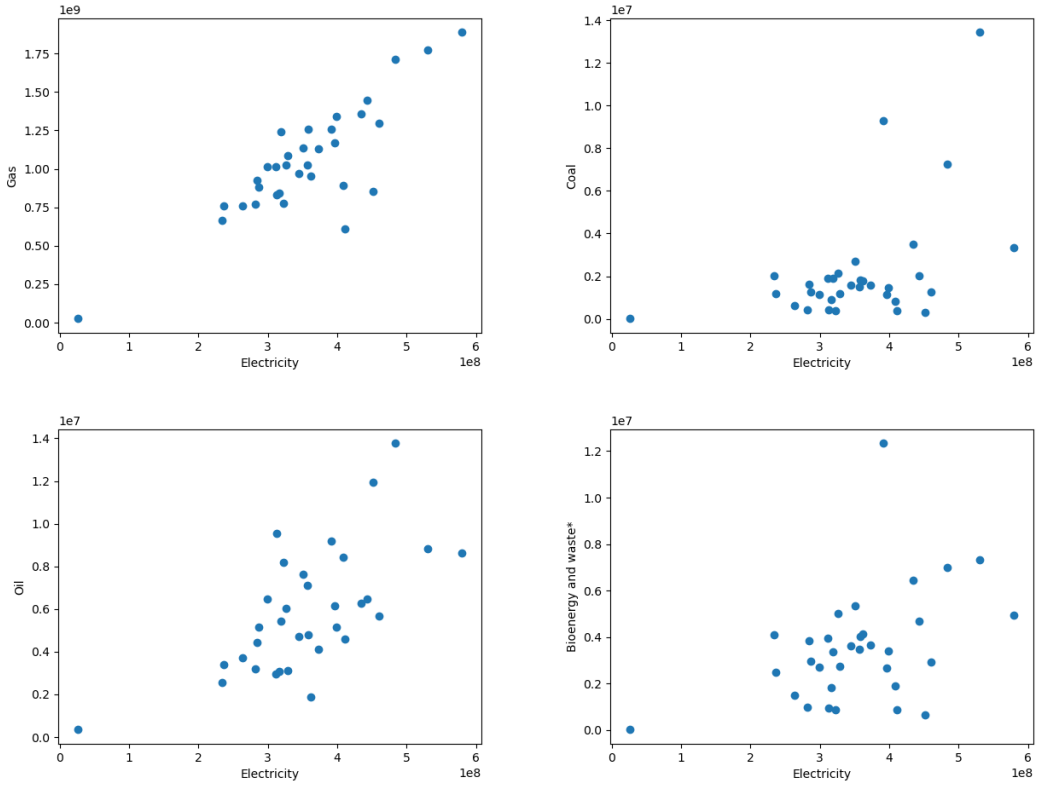


Figure 2: domestic energy consumption correlations

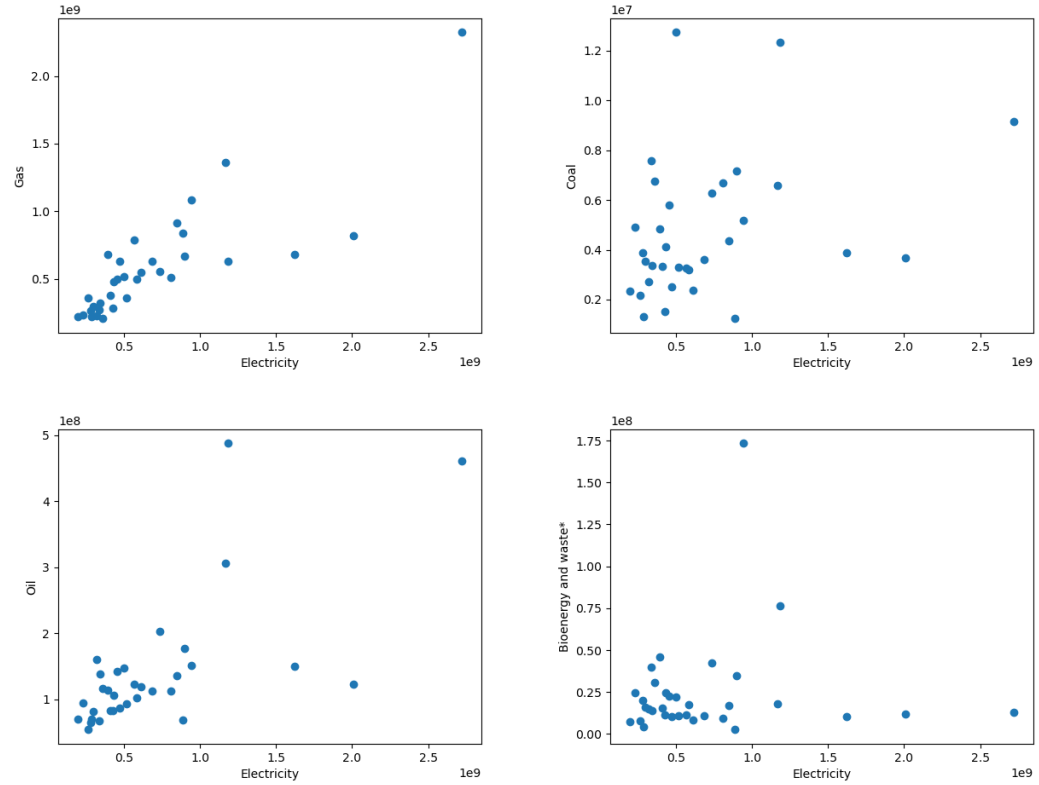


Figure 3: Industrial and commercial energy consumption correlations

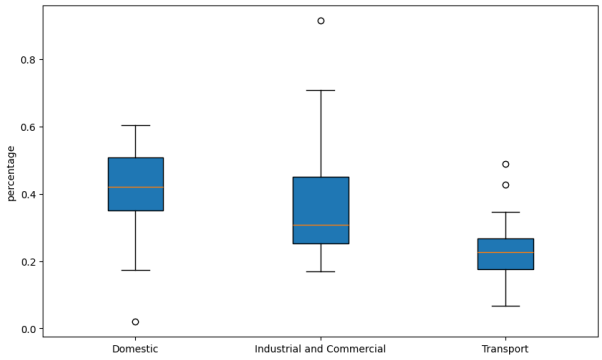
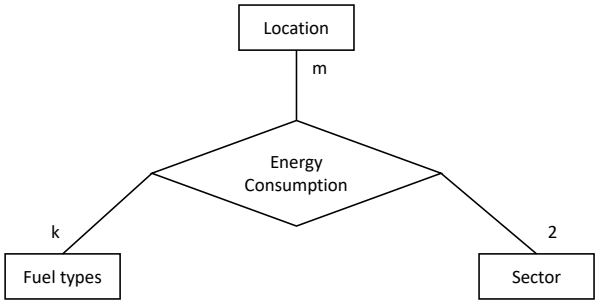


Figure 4: energy consumption percentage



Location Table

loc_id	loc_name
0	Hillingdon
1	Richmond
2	...

Sector Table

sector_id	sector_name
0	Domestic
1	Commercial

Fuel types Table

fuel_id	fuel_name
0	Gas
1	Coal
2	...

Figure 5: Database design.

4.1.4 4. Commercial Usage Table (commercial_usage)

Attribute	Data Type	Key Type
usage_id	INTEGER	Primary Key
city_ref	INTEGER	Foreign Key → location_data(city_id)
sector_ref	INTEGER	Foreign Key → sector_data(sector_id)
fuel_ref	INTEGER	Foreign Key → fuel_types(fuel_id)
usage_amount	REAL	-

4.1.5 5. Residential Usage Table (residential_usage)

Attribute	Data Type	Key Type
usage_id	INTEGER	Primary Key
city_ref	INTEGER	Foreign Key → location_data(city_id)
sector_ref	INTEGER	Foreign Key → sector_data(sector_id)
fuel_ref	INTEGER	Foreign Key → fuel_types(fuel_id)
usage_amount	REAL	-

4.2 Relationships

The database implements the following relationships where m represents the number of locations (33), 2 represents the number of sectors, and k represents the number of fuel types (5):

- location_data (m) → commercial_usage (n)
  - Each location can have multiple commercial usage records
  - Each commercial usage record belongs to one location
- location_data (m) → residential_usage (n)
  - Each location can have multiple residential usage records

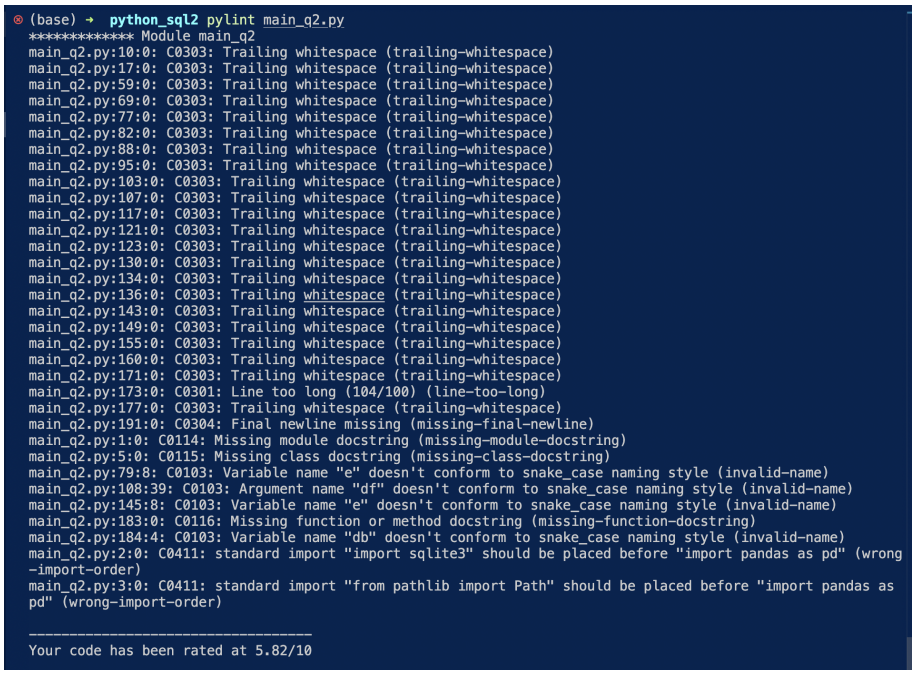


Figure 6: Before Linting

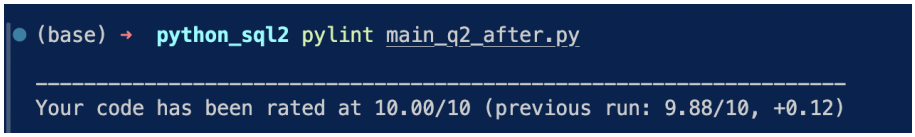


Figure 7: After Linting

  - Each residential usage record belongs to one location

- sector_data (2) → commercial_usage (n)
  - Two sectors (Commercial and Residential)
  - Each commercial usage record belongs to one sector

- sector_data (2) → residential_usage (n)
  - Two sectors (Commercial and Residential)
  - Each residential usage record belongs to one sector

- fuel_types (k) → commercial_usage (n)
  - k fuel types (Electricity, Gas, Coal, Oil, Bioenergy)
  - Each commercial usage record belongs to one fuel type

- fuel_types (k) → residential_usage (n)
  - k fuel types (Electricity, Gas, Coal, Oil, Bioenergy)
  - Each residential usage record belongs to one fuel type

5 Section 3 Tools

5.1 Setup

Please refer to 3_requirements.txt and 3_readme.md.

5.2 Linting

Changes made:

- Added module docstring
- Fixed import order (standard library imports first)
- Removed trailing whitespace
- Renamed variables: 'e' → 'database_error', 'df' → 'consumption_data', 'db' → 'database'
- Fixed line length issues by breaking long lines
- Added final newline
- Improved error handling with more specific exception types
- Improved variable naming for clarity
- Added consistent spacing throughout the code
- Removed all trailing whitespace after lines
- Added final newline at the end of file
- Removed extra blank lines between SQL table definitions
- Maintained consistent indentation
- Kept docstrings and code functionality intact

5.3 Source code control

For coursework 1, I use Git as the version control system. The main branch contains the stable code. Each commit includes a clear message describing what changes were made. Before merging changes back to the main branch, I run pylint to check code quality and ensure all tests pass. The project's dependencies are tracked in requirements.txt, making it easy for others to set up the same development environment. I use a basic .gitignore file to exclude database files and Python cache files from version control.

References

Greater London Authority (GLA). 2024. London energy and greenhouse gas inventory (leggi).

Statement of AI use: No.