Project Description

This supplement describes the data provided for your group project. These are real-world datasets. To protect the data provider's proprietary information, the structure of these datasets, the locations of the tanks therein, and the invoices have been obfuscated so as *not* to reflect the real information of the data provider.

The datasets chronicle over a year's fuel purchases (by the gas station owners) and sales at all city gas stations.

Data Dictionary

Locations.csv

This dataset lists all the gas station locations and contains the following columns:

- Gas Station Location: The unique ID of the gas station
- Gas Station Name: The gas station name
- Gas Station Address: The gas station address
- Gas Station Latitude: The gas station latitude
- Gas Station Longitude: The gas station longitude

Tanks.csv

Each gas station location may have more than one tank. This dataset contains information about these tanks and their attributes

- Tank ID: A unique ID of each tank in the system
- Tank Location: Gas station this tank is located at
- Tank Number: ID of each tank in a specific location
- Tank Type: The type of fuel this tank is used for: U for regular gas, D for Diesel, and P for premium gas. You can consider D and P as Gas.
- Tank Capacity: Capacity of the tank in liters

Invoices.csv

Each gas station purchases different fuel types from its supplier(s). Every delivery of each fuel type to all tanks of a location generates one invoice. The Invoices.csv dataset contains information about these invoices over time and has the following columns:

- Invoice Date: Date of the purchase
- Invoice ID: Unique ID of the invoice
- Invoice Gas Station Location: Gas station location
- Gross Purchase Cost: Total Canadian Dollar (CAD) paid for the purchase
- Amount Purchased: Total number of fuel liters purchased
- Fuel Type: Purchased fuel type

Fuel_Level_Part_1.csv and **Fuel_Level_Part_2.csv**

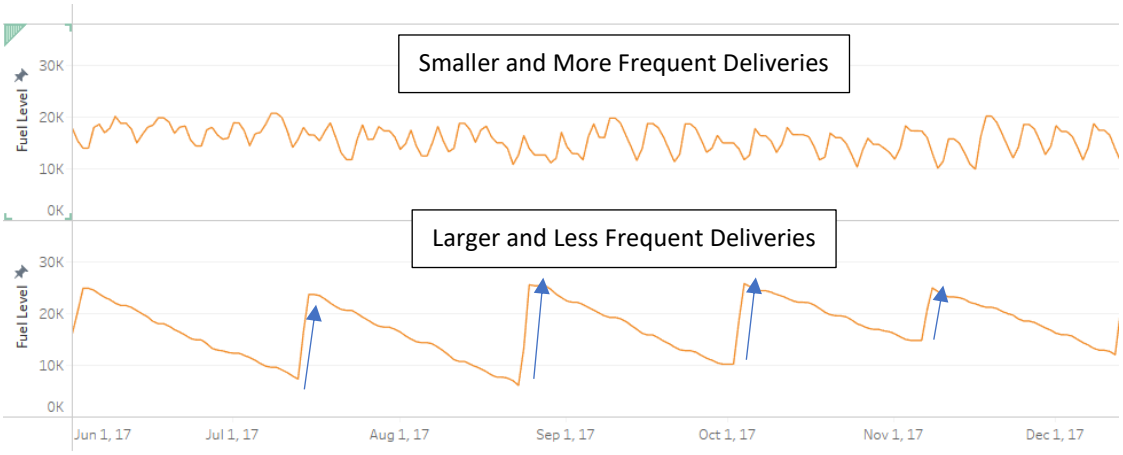
These two datasets contain fuel level information in each tank at frequent and mostly regular time stamps. These two datasets contain the following columns:

- Tank ID: ID of the tank
- Fuel Level: The amount of remaining fuel (inventory in liters)
- Time Stamp: The time of inventory reporting

Problem Description

A gas station purchases fuel in bulk (thousands of liters) to meet the needs of its customers. Each gas station may offer different fuel types (regular gas, premium gas, diesel), with each type stored in one or more underground tanks. The number and capacity of these tanks are influenced by factors such as available space, city regulations, proximity to suppliers, and demand.

At any given time, a gas station might carry tens of thousands of liters of fuel. Decisions regarding fuel replenishment frequency and fuel replenishment quantity are crucial for profitability and operational efficiency. Frequent replenishments in small quantities reduce cash tied up in inventory but may miss out on quantity discounts offered by suppliers. Conversely, larger, less frequent deliveries often qualify for discounts but require higher upfront costs (see Figure 1). Note that quantity discounts are applied separately for each gas station location and fuel type; purchases cannot be aggregated across locations or fuel types.



The supplier offers the following quantity discounts:

Purchase quantity (liters)	Discount per liter
0-15000	0
15000-25000	2 cents
25000-40000	3 center
40000+	4 cents

Business Questions

Your team is responsible for thoroughly exploring the provided datasets, analyzing inventory replenishment patterns, and proposing a cost-effective inventory policy. Your recommendations should aim to reduce purchasing costs while maintaining adequate fuel levels to avoid stockouts. The analysis will be conducted using Python and its data analysis libraries. Delivery costs can be ignored in this project.

Your analysis should address the following questions:

1. Evaluate Current Inventory Management Practices

Analyze how well the gas stations manage their fuel inventory and order quantities.

- Data Preprocessing
  - Concatenate Fuel_Level_Part_1.csv and Fuel_Level_Part_2.csv.
  - Merge the Tanks.csv, Locations.csv, and Invoices.csv datasets.
  - Clean datasets and handle missing values.
- Visualize Fuel Levels
  Create visualizations for each tank to assess inventory trends over time. Identify stations with effective inventory management practices and those at risk of stockouts.
- Quantify Current Performance
  Analyze the Invoices.csv dataset to evaluate current purchasing patterns. Calculate cost savings achieved via discounts for stations 1–8.

2. Recommend Improved Ordering Strategies

Propose strategies to improve fuel order quantities and save costs.

- Quantify Maximum Potential Savings:
  - Use the total tank capacity for each location and fuel type to determine the highest applicable discount rate.
  - Calculate a 7-day inventory threshold based on the average daily consumption for each location and fuel type.
  - Compute potential savings by optimizing order quantities to maximize discounts while maintaining sufficient inventory.
- Provide Recommendations:
  Offer specific recommendations to improve inventory policies and estimate potential cost savings for each location.

3. Identify the Best Day for Fuel Orders

Analyze whether a specific day of the week consistently offers lower fuel prices.

- Examine the distribution of price-per-liter rates by day of the week.
- Identify the day with the lowest historical rates.
- Calculate additional savings if purchases were consistently made on the identified day.

4. Evaluate the Feasibility of Adding Tanks

Assess whether increasing tank capacity at certain locations would be beneficial.

- Assume the cost of adding a new tank and the average inflation rate.
- Estimate the cost-benefit of increased capacity over a 5-year period.
- Identify locations that would benefit most from additional tanks.

In addition to addressing the outlined questions, your analysis should explore other relevant aspects of inventory management to ensure a thorough understanding of cost structures and operational efficiency.

Group Report Requirements

Each team must submit a Python notebook with your code and Markdown annotations explaining the project, methodology, and results.

- Your project will be assessed on:
    - Data Processing: Effective cleaning, merging, and preprocessing of datasets.
    - Clarity of Code: Well-documented code with meaningful comments.
    - Exploration of Datasets: Insightful analysis and overview of the data.
    - Business Questions: Identifying and answering critical business questions.
    - Report Quality: Clear, well-structured, and thorough documentation.
    - Visualization: Professional and informative graphs.
    - Academic Integrity: Original work with appropriate citations if external resources are used.
    - Logical Soundness: Use of sound reasoning and methodology in deriving conclusions.

We will validate your analysis by running your submitted code and comparing results with your report.