

Consider the following scenario:

You are a data engineer for a renewable energy company that operates a farm of wind turbines. The turbines generate power based on wind speed and direction, and their output is measured in megawatts (MW). Your task is to build a data processing pipeline that ingests raw data from the turbines and performs the following operations:

- Cleans the data: The raw data contains missing values and outliers, which must be removed or imputed.
- Calculates summary statistics: For each turbine, calculate the minimum, maximum, and average power output over a given time period (e.g., 24 hours).
- Identifies anomalies: Identify any turbines that have significantly deviated from their expected power output over the same time period. Anomalies can be defined as turbines whose output is outside of 2 standard deviations from the mean.
- Stores the processed data: Store the cleaned data and summary statistics in a database for further analysis.

Data is provided to you as CSVs which are appended daily. Due to the way the turbine measurements are set up, each csv contains data for a group of 5 turbines. Data for a particular turbine will always be in the same file (e.g. turbine 1 will always be in `data_group_1.csv`). Each day the csv will be updated with data from the last 24 hours, however the system is known to sometimes miss entries due to sensor malfunctions.

The files provided in the attachment represent a valid set for a month of data recorded from the 15 turbines. Feel free to add/remove data from the set provided in order to test/satisfy the requirements above.

Your pipeline should be scalable and testable; emphasis is based on the clarity and quality of the code and the implementation of the functionality outlined above, and not on the overall design of the application.

Your solution should be implemented in Python, using any frameworks or libraries that you deem appropriate. Please provide a brief description of your solution design and any assumptions made in your implementation.