Documentation for the Python Script for anomaly detection based on transaction amount

Introduction

The provided Python script is designed to analyze transactional data and detect anomalies based on various statistical methods. It utilizes pandas for data manipulation and datetime for handling date-related operations. The goal is to identify transactions that deviate significantly from expected patterns, which could indicate potential fraud, errors, or unusual behaviour

Functions used

- **Calculate_thresholds:** This function calculates statistical thresholds (mean, standard deviation, z-score threshold, and IQR upper threshold`) for transaction amounts within each category. These thresholds serve as benchmarks to identify outliers.

    - **Mean (mean):** The average transaction amount for a specific category.
    - **Standard Deviation (std_dev):** Measures the amount of variation or dispersion in the transaction amounts.
    - **Z-score Threshold:** Typically calculated as mean + k * std_dev where k is a multiplier (e.g., 3 in the script), representing how many standard deviations away from the mean a value is considered an outlier.

Transactions with amounts exceeding this threshold are flagged as potential anomalies.

- **IQR threshold**: This threshold identifies outliers based on the spread of transaction amounts within a category, specifically focusing on the middle 50% of the data

  .

  Q1 (First Quartile): The 25th percentile of transaction amounts.
  Q3 (Third Quartile): The 75th percentile of transaction amounts.
  IQR (Interquartile Range): The range between Q1 and Q3 (IQR = Q3 - Q1).
  Upper IQR Threshold: Calculated as Q3 + k * IQR, where k (typically 1.5)

Transactions with amounts exceeding this threshold are considered as outliers

- **calculate_mean_frequency**: This computes the mean daily transaction frequency per category. This helps in identifying anomalies related to transaction frequency.
- **is_zscore_outlier:** This determines if a transaction is a zscore outlier for its category, If the amount exceeds its threshold, it is outlier, used to identify transactions with amounts that significantly deviate from average plus a certain number of standard deviations within their category

- **is_high_amount_outlier:** This determines if a transaction amount is an unusually high amount outlier for its category. Compares the transaction's amount against a threshold (2 * category_data["mean"]). If the amount exceeds twice the mean transaction amount for its category, it's flagged as an outlier. Used to detect transactions with unusually high amounts compared to the average transaction amount within their category.

- **is_iqr_outlier:** This determines if a transaction amount is an IQR outlier for its category. Compares the transaction's amount against the upper IQR threshold (category_data["iqr_upper_threshold"]). If the amount exceeds this threshold, it's considered an outlier. Used to identify transactions with amounts that fall outside the upper range of the Interquartile Range (IQR) for their category.

- **get_historical_transactions:** This retrieves historical transactions for a given category within a specified window period. Used in is_time_series_outlier function to fetch past transactions for calculating moving averages and standard deviations to detect anomalies based on temporal patterns.

- **is_time_series_outlier:** This checks for transaction anomalies based on deviation from the moving average within a category. Retrieves historical transactions (category_history) using get_historical_transactions.Calculates moving average (rolling_average) and standard deviation (rolling_std_dev) of transaction amounts within the specified window_size. Compares the transaction's amount against a threshold (`rolling_average + 2 * rolling_std_dev`). If the amount exceeds this threshold, it's flagged as an anomaly.

Detect anomalies function

Constructs an anomaly report (`anomaly_report`) containing details of transactions flagged as anomalies, including transaction ID, date, category, amount, and reason for anomaly, after iterating through each transaction from the csv data file to detect anomalies based on the outliers. It will print the anomaly report if there are any, and if there aren't any, it will print anomaly not found.

Anomaly Report:

```
Anomaly Report:
{'transaction_id': 4, 'date': '2023-06-04', 'category': 'A', 'amount': 60, 'reason_for_anomaly': 'Time series anomaly', 'category_anomaly': False}
{'transaction_id': 8, 'date': '2023-06-08', 'category': 'A', 'amount': 100, 'reason_for_anomaly': 'Z-score anomaly', 'category_anomaly': False}
{'transaction_id': 12, 'date': '2023-06-04', 'category': 'B', 'amount': 160, 'reason_for_anomaly': 'Time series anomaly', 'category_anomaly': False}
{'transaction_id': 16, 'date': '2023-06-08', 'category': 'B', 'amount': 500, 'reason_for_anomaly': 'Unusually high transaction amount', 'category_anomaly': False}
{'transaction_id': 23, 'date': '2023-06-07', 'category': 'C', 'amount': 25, 'reason_for_anomaly': 'Time series anomaly', 'category_anomaly': False}
{'transaction_id': 24, 'date': '2023-06-08', 'category': 'C', 'amount': 60, 'reason_for_anomaly': 'Unusually high transaction amount', 'category_anomaly': False}
{'transaction_id': 32, 'date': '2023-06-08', 'category': 'D', 'amount': 100, 'reason_for_anomaly': 'Unusually high transaction amount', 'category_anomaly': False}
{'transaction_id': 33, 'date': '2023-06-09', 'category': 'A', 'amount': 54, 'reason_for_anomaly': 'High frequency anomaly', 'category_anomaly': True}
{'transaction_id': 34, 'date': '2023-06-09', 'category': 'A', 'amount': 56, 'reason_for_anomaly': 'High frequency anomaly', 'category_anomaly': True}
{'transaction_id': 35, 'date': '2023-06-09', 'category': 'A', 'amount': 57, 'reason_for_anomaly': 'High frequency anomaly', 'category_anomaly': True}
{'transaction_id': 36, 'date': '2023-06-09', 'category': 'A', 'amount': 55, 'reason_for_anomaly': 'High frequency anomaly', 'category_anomaly': True}
{'transaction_id': 37, 'date': '2023-06-09', 'category': 'A', 'amount': 58, 'reason_for_anomaly': 'High frequency anomaly', 'category_anomaly': True}
```