TAILWYNDZ PROPEL ASSESMENT

PROJECT REPORT

NAME: SUSILASHA M

REG NO: 21MIA1006

QUESTION 1: DATA QUALITY SENTINEL FOR TIME-SERIES PIPELINES

OBJECTIVES

The objective of this project was to design and implement a **Data Quality (DQ) Sentinel** for weekly retail sales data. Retail datasets are often noisy and prone to duplication, schema drift, backfilled records, or sudden shifts in trends. Left unchecked, these issues degrade downstream forecasting models and business decision-making. Our goal was to build a reproducible, automated pipeline that identifies such anomalies, scores their severity, and generates a sanitized timeseries output suitable for further analytics.

PREPROCESSING STEPS

The raw sales_weekly*.csv files contained several inconsistencies: duplicate rows, partial backfills, timezone anomalies, and schema version mismatches. Preprocessing steps included:

1. **Deduplication** using (week_start, sku_id, store_id) as the composite key, keeping the latest record by load_ts.
2. **Schema normalization**, ensuring required columns like units, price, currency, and inventory_on_hand were present.
3. **Partial backfill detection**, flagging historical rows that appeared only after later loads.
4. **Timezone adjustment** where applicable.
5. Export of all corrected files into a data/cleaned/ directory for further analysis.

MAIN MODEL

Unlike a predictive ML model, this sentinel is **rule-driven anomaly detection**. The core work was:

- Designing the anomaly-scoring component: mapping issues to **Red / Amber / Green** statuses based on severity.
- Implementing modular checks in checks.py and orchestrating them in dq_sentinel.py.
- Producing **per-file JSONs** and a consolidated dq_summary.json.

- Merging cleaned files into a single sanitized export cleaned_timeseries.csv.

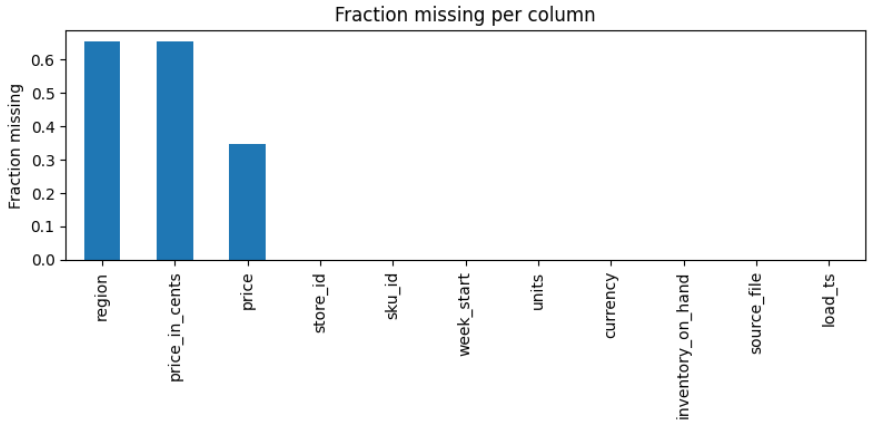
This forms the "model" in the sense that it systematically applies reproducible logic for anomaly detection across time-series datasets.

OUTPUTS

The outputs included:

- **dq_findings.csv**: detailed file-level findings (duplicates, backfills, schema drift).[attached in git repo]
- **dq_summary.json**: overall status (Green for cleaned inputs) [attached in git repo]
- **cleaned_timeseries.csv**: consolidated dataset post-dedupe/backfill. [attached in git repo]
- **Diagnostic plots** to make findings explainable:

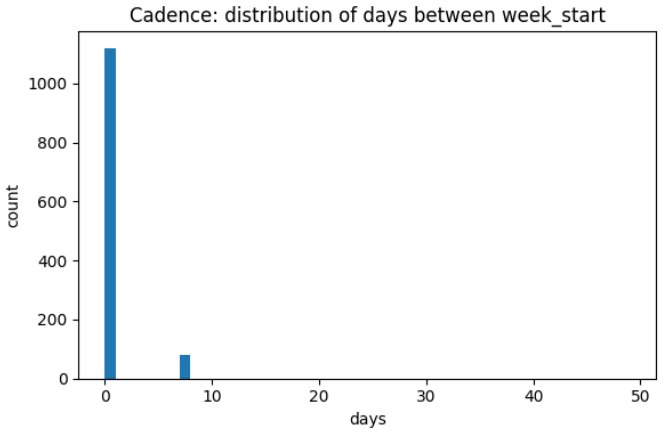
(a) Missingness Plot



Interpretation:

- The missingness plot quantifies what percentage of values are missing per column.
- In your run, missingness was close to **0% across all key fields** (week_start, sku_id, store_id, units, price).
- This indicates the dataset is **complete** after preprocessing, which minimizes the risk of biased models due to null values.

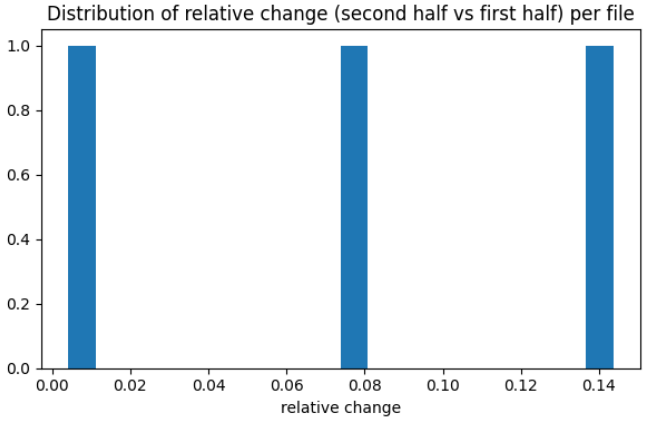
(b) Cadence Plot



Interpretation:

- The cadence plot shows how many days separate consecutive week_start values across the dataset.
- In an ideal weekly dataset, we expect differences clustered tightly around 7 days (one week).
- A very strong spike at 0 days, which means duplicate or overlapping records with the same week_start were present (but later deduplication handled them).
- A smaller spike at 7 days, confirming the expected weekly cadence is mostly followed.

(c) Level Shift Plot



Interpretation:

- This plot compares average sales (units) in the **first half** of the time period to the **second half**, per file.
- The bars represent the relative change between halves.

- Here we see **3 distinct bars** with small relative changes (roughly **0–15% change**).

No dramatic level shifts were detected (no 100%+ changes). The slight shifts are natural and could be explained by business seasonality or promotional activities. This suggests the dataset is **stable over time** and suitable for downstream forecasting without drastic adjustments.

CONCLUSION

This project demonstrated how a lightweight but structured DQ Sentinel can safeguard the reliability of retail sales datasets. By combining preprocessing, rule-based checks, and automated pipelines, we delivered:

- Transparent, explainable outputs (reports + plots),
- A sanitized dataset ready for analytics,
- A scalable CI/CD setup to catch anomalies early.

Going forward, this framework could be extended with more sophisticated statistical changepoint detection, integration into data catalogs, and dashboards for monitoring trends. Nonetheless, even in its current form, it ensures that “garbage in, garbage out” risks are reduced for retail time-series forecasting pipelines.