Great Expectations Validation Analysis Report

Generated on: 2025-10-07 14:45:21
Analysis Period: 20251005T180117.592126Z to 20251005T180117.592126Z

Executive Summary

This report analyzes **132** data quality expectations across **1** validation suites.

Key Metrics

- **Overall Success Rate:** 96.21%
- **Exception Rate:** 0.00%
- **Expectation Types Analyzed:** 15
- **Critical Issues:** 1 expectation types below 80% success rate

Critical Findings

Top Issues Requiring Attention

1. **expect_column_mean_to_be_between**: 58.3% success rate (7.0/12.0 expectations)

AI-Powered Analysis

Great Expectations Data Quality Report

Run ID: `20251005T180117.592126Z`
Date of report: October 7 2025

> **Scope** – Validation run on the *nyc_taxi_data_onboarding_suite_final* suite (132 expectation checks) covering a single-record snapshot of the NYC taxi dataset.

1. Executive Summary

Metric	Value
Total Expectations	132
Successful Expectations	127
Failure Rate	3.79 %
Success Rate	96.21 %
Exceptions	0
Suite Count	1

The validation run demonstrates **overall good data quality**. 96 % of expectations pass and there are **no runtime exceptions**, indicating that the underlying schema and ingest process are functioning as expected.

The sole source of concern is **mean-value expectations**:

- 12 mean expectations were defined, of which only 7 passed (58 % success).

- All maximum-value and median-value expectations passed at 100 %.

The mean-value failures suggest that **some columns have average values that lie outside the declared ranges** (e.g. `total_amount`, `trip_distance`, etc.). These deviations do not represent a systemic data-integrity problem (no schema violations, no missing columns), but they may affect downstream analytics that rely on statistically-derived metrics.

2. Critical Issues

Issue	Why it matters	Impact
Mean expectation failures (5 out of 12)	Indicates that the statistical distribution of key columns is not within the acceptable bounds defined by business rules or previous historical averages.	Possible mis-interpretation of fare amounts, trip metrics, or driver incentives; risk of incorrect business decisions or KPI reporting.
Single-timestamp data snapshot	The current run covers a single data point (2025-10-05 18:01:17). No temporal trend analysis is possible from this snapshot alone, limiting validation depth.	Inability to detect seasonality, drift, or gradual degradation across epochs.
No exceptions reported	While good for overall integrity, it may mask underlying data issues that do not trigger validation errors—for instance, unexpected nulls in hidden columns or foreign-key mis-references not covered by the suite.	Potential for silent data quality drift in future runs.

3. Trends Analysis

Trend	Observation	Notes
Consistent high success for structural checks	All column-maximum, median, and basic existence checks passed.	The schema is stable and consistent with expectations.
Inconsistent statistical checks	Mean checks have a 58 % pass rate whereas max/ median checks pass at 100 %.	Likely due to outlier or drift in underlying data values, not necessarily a structural issue.
Temporal limitation	Validation window is a single second; no historical comparison possible.	Future runs should span multiple days or weeks to surface performance trends.
Zero-exception rate	No test failures due to runtime exceptions.	Provides confidence that the validation harness is configured correctly.

4. Recommendations

Recommendation	Target	Action Steps	Owner
Adjust mean expectation ranges	Columns with failing mean checks	• Validate current bounds against recent historical data (last 7 days). • Re-calculate means for each column and update `expect_column_mean_to_be_between` thresholds accordingly.	Data Engineering
Implement dynamic expectation calibration	All statistical expectations	• Add a pre-validation step that calculates current statistics and automatically updates expectation ranges (e.g., using a percentile-based approach). • Schedule calibration weekly.	Data Ops
Expand benchmark windows	Validation suite	• Extend the `batch_kwargs` to fetch a 7-day window (or hourly buckets) rather than a single timestamp. • Add expectation types that detect trend drift (`expect_column_mean_to_change_by`).	Analytics
Introduce anomaly detection	Emerging patterns	• Add `expect_column_mean_to_be_increasing` or `expect_column_mean_to_be_decreasing` to detect unexpected reversals.	Data Quality Team
Schedule automated alerts	Mean expectation failures	• Configure Slack or email alerts when any mean expectation fails.	Data Ops
Review data ingestion pipeline	Data quality continuity	• Conduct a root-cause analysis of the mean outliers: data source changes, transformations, or data source integrity.	ETL Developers
Document expectation rationale	Governance	• Create documentation linking each expectation to a business requirement. This aids future stakeholders in understanding the purpose of bounds.	Knowledge Base

5. Risk Assessment

Risk	Likelihood	Impact	Mitigation
Statistical bias in downstream analytics	Medium	High	Tighten mean expectations, periodic calibration.
Undetected data drift	Low	Medium	Expand batch window and add drift-detection expectations.
Operational disruption due to false positives	Low	Low	Validate adjusted thresholds against stakeholders before deployment.
Increased maintenance overhead	Medium	Medium	Automate expectation updates via CI/CD pipeline.
Loss of stakeholder trust	Low	High	Communicate changes, provide evidence of improved accuracy.

6. Next Steps (Prioritized Action Items)

#	Action	Owner	Due Date
1	**Root-cause analysis of mean failures**	ETL Developers	2025-10-15
2	**Re-calculate and update mean thresholds**	Data Engineering	2025-10-20
3	**Implement automated calibration of statistical expectations**	Data Ops	2025-10-25
4	**Extend validation window to 7 days**	Analytics	2025-10-28
5	**Add drift-detection expectations**	Data Quality Team	2025-11-02
6	**Deploy alerting mechanism for mean failures**	Data Ops	2025-11-05
7	**Generate stakeholder briefing on updated expectations**	Knowledge Base	2025-11-07

Closing Remarks

The current validation shows that the **structural integrity of the dataset is intact**, but **statistical expectations on mean values require adjustment** to better reflect the observed data distribution. Proactively calibrating these expectations and expanding the temporal scope of validation will provide more robust, actionable insights and mitigate the risk of inaccurate reporting.

Data Catalog Summary

Data Assets: 1 **Expectation Suites:** 1 **Validation Runs:** 1 **Total Columns Monitored:** 15 ## Recommendations Based on the analysis, the following actions are recommended: 1. **Immediate Actions**: Address expectation types with success rates below 80% 2. **Monitoring**: Implement daily monitoring for critical data assets 3. **Expectation Review**: Review and update failing expectation types 4. **Process Improvement**: Establish data quality governance processes ## Technical Details - **Analysis Engine**: Great Expectations v0.18.22 - **AI Analysis**: Ollama LLM (gpt-oss:20b) - **Data Source**: Validation results from BirdiDQ/gx/uncommitted/validations - **Report Generated**: {datetime.now().isoformat()} ---

Appendix A: Detailed Suite Performance | Suite Name | Total Expectations | Successful | Success Rate | Exceptions | |------------|------------------|------------|--------------|------------| | nyc_taxi_data_onboarding_suite_final | 132.0 | 127.0 | 96.20% | 0.0 | ## Appendix B: Detailed Expectation Type Performance

Expectation Type	Total	Successful	Success Rate	Exceptions
expect_column_max_to_be_between	14.0	14.0	100.00%	0.0
expect_column_mean_to_be_between	12.0	7.0	58.30%	0.0
expect_column_median_to_be_between	12.0	12.0	100.00%	0.0
expect_column_min_to_be_between	14.0	14.0	100.00%	0.0
expect_column_proportion_of_unique_values_to_be_between	8.0	8.0	100.00%	0.0
expect_column_quantile_values_to_be_between	12.0	12.0	100.00%	0.0
expect_column_stdev_to_be_between	12.0	12.0	100.00%	0.0
expect_column_unique_value_count_to_be_between	8.0	8.0	100.00%	0.0
expect_column_value_lengths_to_be_between	1.0	1.0	100.00%	0.0
expect_column_values_to_be_between	14.0	14.0	100.00%	0.0
expect_column_values_to_be_in_set	8.0	8.0	100.00%	0.0
expect_column_values_to_match_regex	1.0	1.0	100.00%	0.0
expect_column_values_to_not_be_null	14.0	14.0	100.00%	0.0
expect_table_columns_to_match_set	1.0	1.0	100.00%	0.0
expect_table_row_count_to_be_between	1.0	1.0	100.00%	0.0

Appendix C: Data Catalog ### Data Assets Overview

Data Asset	Type	Table	Schema	Datasource	Columns	Suites
nyc_taxi_data	table	nyc_taxi_data	None	postgres_sql_nyc_taxi_data	15	1

Column Quality Summary

Data Asset	Column	Expectations	Success Rate	Exceptions
nyc_taxi_data	index	8	100.00%	0
nyc_taxi_data	passenger_count	11	100.00%	0
nyc_taxi_data	trip_distance	8	87.50%	0
nyc_taxi_data	store_and_fwd_flag	6	100.00%	0
nyc_taxi_data	payment_type	11	100.00%	0
nyc_taxi_data	fare_amount	8	87.50%	0
nyc_taxi_data	extra	11	100.00%	0
nyc_taxi_data	mta_tax	11	100.00%	0
nyc_taxi_data	tip_amount	8	87.50%	0
nyc_taxi_data	tolls_amount	11	90.91%	0
nyc_taxi_data	improvement_surcharge	11	100.00%	0
nyc_taxi_data	total_amount	8	87.50%	0
nyc_taxi_data	pickup	4	100.00%	0
nyc_taxi_data	dropoff	4	100.00%	0
nyc_taxi_data	congestion_surcharge	10	100.00%	0

--- *This report was automatically generated by the Great Expectations Validation Analysis system.*