Data Corrector: Error Analysis

CARLOS JAVIER GALÁN ZAVALA*

February 8, 2026

Abstract

The following report proposes a framework complementary to the existing DataCurator module to treat financial data integrity issues. The financial data errors provided (through the error log and the complete database) suggest that errors are systemic rather than random; tickers which demonstrate certain errors have common characteristics. This report recommends a conservative approach to data error detection and imputation. Empirical evidence shows that error detection through statistical methods results in an overwhelming amount of false positives.

This framework argues that data provider errors should be strictly detected using hard constraints derived from financial logic. Even with "unbreakable" rules, checks derived from these should be audited. For this reason, the proposal is a module separate from the KaxaNuk DataCurator which is capable of visualizing and auditing data quality corrections.

*inquiries.carlosgalan@gmail.com

Contents

1 Introduction

Data provider issues are a rather common problem that financial analysts need to address. Data quality translates to model quality, model quality translates into alpha. The purpose of this challenge is to first gain an understanding of the problems by analyzing the problems themselves, and their underlying causes and patterns. This in order to propose better informed solutions to data correction or imputation instead of blindly proposing machine learning algorithms.

Once the error is identified, an appropriate imputation method should be implemented. This approach explored deterministic and statistical detection and imputation. Regardless of the methodology, error correction in financial data should always be accompanied by an error log forensic analysis in order to minimize false positives. The forensic analysis in this work was done using a large language model (Gemini 3) to scrape financial news sites for events that may be related to the observed error logs. After thoroughly examining the forensic analysis, false positive flagging should always be under the practitioner's criteria.

1.1 Summary of findings

- Financial data errors are systemic phenomena
- Sanity checks should be able to handle edge cases where "logic" may not properly apply
- Statistical methods yield too many false positives
- Auditing the error detection process is paramount for its validity

1.2 Proposal

Although the sanity check functions could be implemented as custom calculations within the DataCurator module, logging corrections and log visualization is paramount for the practitioner's understanding and choice over which fixes to implement. This report recommends the following approach:

- A module separate from the DataCurator which is capable of detecting and correcting data provider errors.
- Disable the quality checks which sparked the errors from the DataCurator in order to delegate the initial sanity check into this new module.
- A thorough sanity check based **only** in financial logic and equivalencies.
- Error logging and subsequent forensic analysis of the error logs.
- Visualization to aid the evaluation of the forensic analysis.

Alternative: Implement this sanity check as a series of custom calculations within the DataCurator. An working implementation of this framework can be seen in the github repo from this project.

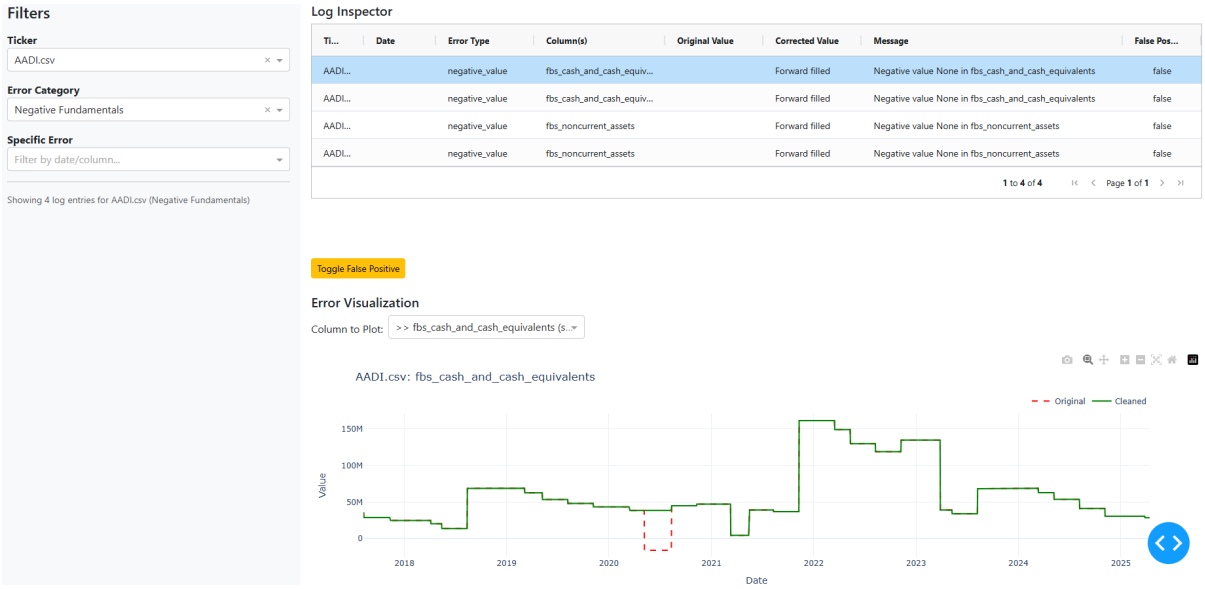

Figure 1: Screenshot of the module developed to tackle this challenge.

2 Provided Error Log Analysis

Financial Modeling Prep's data errors are **not isolated incidents but systemic issues** concentrated in three categories: non-standard securities (preferred shares, warrants, senior notes), companies undergoing corporate actions, and illiquid micro-cap stocks. The errors stem from data architecture that struggles with securities that deviate from standard common stock data structures.

2.1 The "Sorted by Date" Errors

The 74 tickers generating "FundamentalData.rows not correctly sorted by date" errors reveal the same patterns. Nearly **40% are non-common equity securities**—preferred shares, warrants, or senior notes—that have fundamentally different data reporting requirements than common stock. The B. Riley Financial family alone contributes 8 tickers (RILY, RILYG, RILYK, RILYL, RILYN, RILYT, RILYZ, RILYP), spanning common stock, preferred shares, and tradeable senior notes. Federal Agricultural Mortgage (Farmer Mac) adds 8 more (AGM and seven preferred series), while Presidio Property Trust contributes common stock, preferred shares, and warrants.

Corporate actions create data discontinuities across this list. At least 12 tickers underwent mergers, acquisitions, or name changes in 2024–2025:

Table 1: Corporate Actions Creating Data Discontinuities (2024–2025)

Ticker	Event	Date
FARO	Acquired by AMETEK	July 2025
IVAC	Acquired by Seagate	March 2025
SASR	Acquired by Atlantic Union	April 2025
APDN	Rebranded to BNBX	October 2025
MICS	Became RIME	September 2024
ATON	Rebranded AlphaTON Capital	September 2025
NBP	Former I-Mab, now NovaBridge	October 2025

Several companies are in financial distress: B. Riley Financial suspended dividends and faces Nasdaq delisting risk after **435–475M quarterly losses**; Ideanomics (IDEX) filed Chapter 11 bankruptcy in December 2024 following SEC fraud settlements; Staffing 360 Solutions (STAF) was delisted to OTC.

2.2 Market Cap Distribution

The market cap breakdown exposes another pattern: **approximately 35–40% of affected tickers are micro-cap stocks** (under $300M market capitalization). These include XELB ($8–11M), EVTV ($20M), SOTK ($40M), TPCS ($35M), and DLPN ($50M). Micro-cap stocks typically have less rigorous data reporting, lower analyst coverage, and more frequent data quality issues due to limited institutional oversight.

Table 2: Market Cap Distribution of Error Tickers

Market Cap Category	Percentage of Error Tickers
Micro-cap (<$300M)	∼35–40%
Small-cap ($300M–$2B)	∼25–30%
Mid-cap ($2B–$10B)	∼20%
Large-cap (>$10B)	∼15%

The large-cap tickers that appear—DOV ($27B), DG ($22–24B), JBL ($18B), RBA ($19.5B)—likely experience errors due to corporate actions rather than data quality. RBA (RB Global) completed a major merger with IAA in 2023, and MTZ (MasTec) faced shareholder lawsuits creating reporting complexities.

2.3 Preferred Shares Dominate the "No Data Returned" Errors

All 12 tickers generating "No data returned by unadjusted market data endpoint" errors are **preferred shares or eliminated share classes**:

- **PEI series** (PEI-PB, PEI-PC, PEI-PD): Pennsylvania REIT preferred shares—company emerged from Chapter 11 bankruptcy in 2020 with restructured capital

- **PSB series** (PSB-PX, PSB-PY, PSB-PZ): PS Business Parks preferred depositary shares—parent company was acquired by Blackstone for $7.6 billion in 2022, delisting common stock but potentially leaving preferred shares trading
- **NRZ series** (NRZ-PA, NRZ-PB, NRZ-PC): Rithm Capital (formerly New Residential Investment) fixed-to-floating rate preferreds experiencing LIBOR transition complications
- **STZ-B**: Constellation Brands Class B stock—**eliminated entirely in November 2022** when the Sands family exchanged their super-voting shares for $64.64 cash plus Class A shares
- **PNC-PP**: PNC Financial Series P preferred with complex fixed-to-floating rate structure
- **ALP-PQ**: Appears to be an invalid or delisted ticker

The pattern is unmistakable: FMP's unadjusted market data endpoint cannot handle preferred share structures, depositary shares, or securities that no longer trade but retain historical data.

2.4 Warrants Create Impossible Price Relationships

The three tickers with "MarketDataDailyRow low $>$ high" errors are all **SPAC warrants trading at near-zero prices**:

Table 3: SPAC Warrants with OHLC Violations

Ticker	Company	Current Price	Status
UWMC-WT	UWM Holdings	∼$0.01	NYSE delisting proceedings initiated December 19, 2025
BFLY-WT	Butterfly Network	∼$0.02	Extremely illiquid, ∼44K average daily volume
ML-WT	MoneyLion	∼$0.26	Thinly traded, expires September 2026

When securities trade at fractions of a penny with minimal volume, bad tick data becomes inevitable. Wide bid-ask spreads, stale quotes, and erroneous trade reports create situations where recorded daily lows can exceed daily highs. UWMC-WT is actively being delisted for "abnormally low selling price"—the security is essentially worthless.

2.5 Negative Shares Outstanding Traces to Corporate Restructuring

The three "Negative shares outstanding" errors (HELE, QLGN, ELDN) correlate directly with significant corporate events:

HELE (Helen of Troy, error date May 1, 2017): No stock splits, but the company operates on a February fiscal year-end. The error date falls during fiscal year transitions when share counts from buyback programs may create calculation discrepancies across FMP's data sources.

QLGN (Qualigen Therapeutics, error date November 14, 2025): This company has undergone **two reverse stock splits** (1-for-10 in 2022, 1-for-50 in 2024), was acquired by Faraday Future as a 55% stakeholder, and rebranded to AIxCrypto Holdings in November 2025—all creating massive data discontinuities.

ELDN (Eledon Pharmaceuticals, error date November 14, 2025): A **$50 million dilutive offering** closed around November 12, 2025, adding 15+ million shares plus warrants, increasing share count by over 100% year-over-year. The error date coincides exactly with this offering.

2.6 The Single Negative Price Error Points to Data Corruption

ASB-PF (Associated Banc-Corp Series F Preferred) showing a negative low price is simply **data corruption**. Preferred stocks have complex ex-dividend adjustments, and a calculation error in FMP's dividend adjustment pipeline likely produced an impossible negative value. The security trades normally around $21 with a 6.65% yield.

2.7 Conclusions: Systemic Issues with Non-Standard Securities

These errors demonstrate FMP has **architectural limitations handling three categories of securities**:

Non-standard security types: Preferred shares, warrants, senior notes, and depositary shares have different data structures, reporting requirements, and pricing mechanics than common stock. FMP's fundamental data infrastructure appears designed primarily for common equity.

Corporate action transitions: Mergers, acquisitions, reverse splits, name changes, and bankruptcies create data discontinuities. When Constellation Brands eliminated STZ-B or Blackstone acquired PSB, historical data must be handled differently—FMP's pipeline struggles with these transitions.

Illiquid and penny securities: When UWMC-WT trades at $0.0098 with minimal volume, standard data validation breaks down. The "low $>$ high" errors are essentially the data provider acknowledging bad tick data from nearly untradeable securities.

The pattern suggests it would be beneficial to disable the native lock the data curator has that yielded this error log, and implement robust handling taking into account non-common equity and apply different validation rules capable of handling illiquid instruments while logging the errors detected. For users, these errors serve as a useful audit tool to identify overall quality of a data provider.

3 Sanity Check

The previous log analysis and data correction strategies were consequent to a thorough forensic examination of the critical errors that kept the DataDurator from extracting the data. This next section documents the functions used to detect and correct error types found in the set of 6000+ tickers, including the errors found in the log and those that did not yield an error that prevented DataCurator from downloading the data. Still, financial logic dictates the encountered values should not be possible, so these values were all treated with a deterministic filter and imputation strategy.

These functions act as a post-download validation and correction layer. The data arrives intact but contains logical inconsistencies, impossible values, or violations of financial accounting identities. Left uncorrected, these issues propagate through backtests, valuations, and risk models—often without triggering obvious failures.

The functions below implement deterministic, auditable corrections with full logging of every modification made to the source data.

These functions could be implemented as custom calculation within the DataCurator module (without explicit error logging) or by separate within the EDA module (with explicit error logging).

3.1 `sort_dates`

3.1.1 Error Addressed

Out-of-order fundamental data rows. Financial statements may arrive with dates that are not chronologically sorted due to amended filings (10-K/A, 10-Q/A) inserted after original statements, fiscal year-end changes creating overlapping periods, or data provider ingestion timing mismatches.

3.1.2 Approach

1. Edit the Data Curator so the dates do not need to be sorted for data to be downloaded. Ingest the uncurated data.
2. Establish a date hierarchy: primary date column (`m_date`) $\rightarrow$ filing date (`f_filing_date`)
3. Preserve null-dated rows in their original positions (they cannot be sorted)
4. Sort only valid-dated rows while maintaining relative positions
5. Deduplicate by keeping earliest or latest filing per period
6. Log all position changes for audit

3.1.3 Correction Method

Reordering only—no values are modified. Null rows remain untouched. Deduplication uses filing date to determine which version to keep.

3.2 `fill_negatives_fundamentals`

3.2.1 Error Addressed

Negative values in fundamental data columns where negatives are impossible. Examples include negative shares outstanding, negative total assets, and negative revenue (in contexts where it should be gross revenue). These typically result from data entry errors, sign convention mismatches, or incorrect aggregation.

3.2.2 Approach

1. Scan specified columns for values < 0
2. Replace negatives with `NULL`
3. Apply forward-fill to propagate the last valid (non-negative) value
4. Log every replaced value with ticker, date, and original value

3.2.3 Correction Method

Forward-fill from last known good value. Assumes temporal continuity: if shares outstanding was 100M yesterday and shows -5M today, yesterday's value is more trustworthy. Since fundamental data follows a stepwise function, interpolation would introduce values which never existed. This is why forward-fill is generally preferred for fundamental data.

3.3 `fill_negatives_market`

3.3.1 Error Addressed

Negative prices in market data. A stock price cannot be negative, but data corruption from dividend adjustment calculation errors, bad tick data propagation, or corporate action misapplication can produce impossible negative values in OHLC or VWAP columns.

3.3.2 Approach

1. Identify negative values in specified columns
2. Gather up to 4 previous valid (non-negative) data points
3. Fit a **backward-looking cubic spline** to extrapolate a replacement value
4. If fewer than 3 prior points exist, fall back to last valid value (if no such value exists, default to zero)
5. If spline produces negative or non-finite result, fall back to last valid value

3.3.3 Correction Method

Cubic spline interpolation (backward-looking only). Explicitly avoids look-forward bias for back-testing integrity. Preserves original null positions. Falls back gracefully when insufficient history exists.

3.4 `zero_wipeout`

3.4.1 Error Addressed

Zero values in share-related columns when trading volume is positive. This paradox indicates data corruption: if volume > 0, trading occurred; if shares outstanding $= 0$, the company has no equity. Both cannot be true simultaneously. Common cause: placeholder zeros inserted during data pipeline failures.

3.4.2 Approach

1. Identify rows where ANY target column equals 0 AND `m_volume` > 0
2. Replace the zero with `NULL`
3. Apply forward-fill to restore continuity
4. Log all affected rows

3.4.3 Correction Method

Conditional forward-fill. Only triggers when the logical impossibility (zero shares + positive volume) is detected. Leaves legitimate zeros (pre-IPO, delisted) untouched if volume is also zero.

3.5 `mkt_cap_scale_error`

3.5.1 Error Addressed

10× or greater jumps in market cap or shares outstanding. These typically indicate unit conversion errors (shares in units vs. thousands vs. millions), data source switches mid-series, or incorrect corporate action adjustments.

This function is meant to detect scale errors in shares outstanding. Since market cap = shares outstanding × close, genuine market events (like splits) would be cancelled in the previous equation yielding the same market cap. Anomalous market caps are, therefore, caused by an anomalous value in either of the factors. Since market data is under higher scrutiny and less prone to scale errors than fundamental data (and particularly shares outstanding), this function attributes market cap anomalies to anomalous shares outstanding values.

The 10× jump parameter was arbitrarily chosen; this value should be adjusted based on the investment universe: it is impossible for large caps to jump 5× overnight (largest jump in history was Volkswagen in 2008 with a ∼4× jump), but in small caps it is a reasonable outlier. Placing a 10× threshold on a universe that contains only small caps would be prone to false positives. Whereas a 100× threshold on a universe that contains only large caps would be prone to false negatives. A future implementation would individually analyze each ticker's market cap tier and place the threshold accordingly.

3.5.2 Approach

1. Compare each row's value to the previous row
2. Flag rows where value $\geq 10\times$ prior value
3. Detect **correlated jumps**: if both market cap AND shares outstanding jump together, the error likely spans multiple rows
4. For correlated jumps, identify the entire error span (values within 20% of jumped value)
5. Apply forward-fill to replace the corrupted span

3.5.3 Correction Method

Forward-fill with span detection. Single-row spikes: replace with prior value. Multi-row plateaus: identify the elevated region and replace entirely. Recalculate market cap from corrected shares outstanding. Logs include error type classification.

3.6 `ohlc_integrity`

3.6.1 Error Addressed

Violations of OHLC mathematical constraints:

- High $<$ max(Open, Close, Low) $\rightarrow$ High should be the maximum
- Low $>$ min(Open, Close, High) $\rightarrow$ Low should be the minimum
- VWAP outside [Low, High] $\rightarrow$ VWAP must fall within the day's range

These violations break technical indicators, volatility calculations, and charting.

3.6.2 Approach

Validates three column groups independently: (1) Raw OHLC (`m_open`, `m_high`, `m_low`, `m_close`, `m_vwap`), (2) Split-adjusted OHLC, and (3) Dividend-and-split-adjusted OHLC.

For each group: compute actual max/min of OHLC values, compare against declared High/Low, and check VWAP bounds.

3.6.3 Correction Method

Table 4: OHLC Integrity Corrections

Violation	Correction
High $<$ actual max	Set High = max(O, H, L, C)
Low $>$ actual min	Set Low = min(O, H, L, C)
VWAP outside range	Set VWAP = (O + H + L + C) / 4

Uses OHLC centroid as VWAP replacement (simple average, not volume-weighted, but mathematically valid).

3.7 `validate_financial_equivalencies`

3.7.1 Error Addressed

Violations of fundamental accounting identities:

Hard Filters (corrected):

- Assets $\neq$ Current Assets + Noncurrent Assets
- Liabilities $\neq$ Current Liabilities + Noncurrent Liabilities

Soft Filters (flagged only):

- Stockholder Equity $\neq$ Common Stock + APIC + Retained Earnings + Other Equity
- Period End Cash $\neq$ Cash and Cash Equivalents
- Assets $\neq$ Liabilities + Equity + Noncontrolling Interest

3.7.2 Approach

1. Compute component sums for each identity
2. Compare against declared totals with configurable tolerance (default 5%)
3. For hard filters: apply proportional scaling to force balance
4. For soft filters: set `data_warning` flag without modifying values

3.7.3 Correction Method (Hard Filters)

Proportional Scaling:

$$\text{Factor} = \frac{\text{Total}}{\text{Current} + \text{Noncurrent}} \tag{1}$$

$$\text{Corrected\_Current} = \text{Current} \times \text{Factor} \tag{2}$$

$$\text{Corrected\_Noncurrent} = \text{Noncurrent} \times \text{Factor} \tag{3}$$

Edge Case: If components sum to 0 but total $\neq 0$, the entire total is assigned to the noncurrent bucket (residual plug). **Limitation:** This works under the assumption that "Total" is the ground truth. This should be corroborated through external means prior to calculation.

3.8 Limitation

Soft filter violations are logged but not corrected because: equity components may have legitimate "other" buckets not captured; cash timing differences may reflect intra-period movements; balance sheet identity failures may indicate complex structures (e.g., variable interest entities); and a sample of 500 tickers yielded. This validation logic yielded an overwhelming amount of false positives, therefore a more comprehensive context-aware logic is required.

3.9 `validate_market_split_consistency`

3.9.1 Error Addressed

Inconsistency between raw market data and split-adjusted market data. The relationship should be deterministic:

$$\text{Adjusted\_Price} = \text{Raw\_Price} \times K \tag{4}$$

$$\text{Adjusted\_Volume} = \text{Raw\_Volume}/K \tag{5}$$

where K is the cumulative split adjustment factor. When K_{implied} (from raw/adjusted) $\neq K_{\text{expected}}$ (from split events), the data is internally inconsistent.

3.9.2 Approach

1. Calculate daily split factor: factor $=$ denominator$/$numerator (1.0 if no split)
2. Calculate cumulative K: $K_{\text{expected}} = \text{cumulative\_product}(\text{daily\_factors})$

3. For each price column pair: $K_{\text{implied}} = \text{adjusted}/\text{raw}$
4. For volume: $K_{\text{implied}} = \text{raw}/\text{adjusted}$ (inverse relationship)
5. Flag rows where $|K_{\text{implied}} - K_{\text{expected}}| > \text{tolerance} \times |K_{\text{expected}}|$

3.9.3 Correction Method

Recalculate adjusted values from raw values using K_{expected}:

- **Prices:** corrected_adjusted $= \text{raw} \times K_{\text{expected}}$
- **Volume:** corrected_adjusted $= \text{raw}/K_{\text{expected}}$

Table 5: Validated Column Pairs for Split Consistency

Raw Column	Adjusted Column	Relationship
`m_open`	`m_open_split_adjusted`	Price ($\times K$)
`m_high`	`m_high_split_adjusted`	Price ($\times K$)
`m_low`	`m_low_split_adjusted`	Price ($\times K$)
`m_close`	`m_close_split_adjusted`	Price ($\times K$)
`m_vwap`	`m_vwap_split_adjusted`	Price ($\times K$)
`m_volume`	`m_volume_split_adjusted`	Volume ($\div K$)

3.10 Summary of Deterministic Corrections

Table 6: Summary Matrix of Sanity Check Functions

Error Type	Detection Method	Correction Method	
`sort_dates`	Unsorted rows	Date comparison	Reorder in place
`fill_negatives_fundamentals`	Negative fundamentals	value < 0	Forward-fill
`fill_negatives_market`	Negative prices	value < 0	Cubic spline / forward-fill
`zero_wipeout`	Zero shares + positive volume	shares $= 0$ AND volume > 0	Forward-fill
`mkt_cap_scale_error`	$10\times$ jumps	value $\geq 10\times$ prior	Forward-fill (span-aware)
`ohlc_integrity`	OHLC constraint violations	H $<$ max, L $>$ min, VWAP bounds	Set to computed bounds
`validate_financial_equivalencies`	Accounting identity failures	$\lvert\text{Total} - \text{Sum}\rvert >$ tolerance	Proportional scaling
`validate_market_split_consistency`	Split adjustment mismatch	$K_{\text{implied}} \neq K_{\text{expected}}$	Recalculate from K_{expected}

For future implementations of this work, more financial logic checks could be implemented for a data ingestion process that is more rigorous.

4 Sanity Check Forensic Log Audit

4.1 Notable patterns

Common mistakes and causes can be seen al throughout the error log stemming from the sanity check. This is a summary of the observed patterns with an example ticker for each.

Ticker	Company Name	Sector	Error Type	Anomaly Characteristic	Prone State / Risk Factor
ADN	Advent Technologies	Energy / Tech	Negative fundamentals (Sign Flip)	Exact match of positive cash balance reported as negative.	De-SPAC: "Reverse Recapitalization" causes data feed to invert signs from non-standard S-4 tables.
AADI	Aadi Bioscience	Biotech	Negative fundamentals (Sign Flip / Mapping)	Negative value matches positive filing data or Asset line.	Reverse Merger: Parsing "Predecessor" private company financials from Proxy Statements.
ADV	Advantage Solutions	Marketing	Financial equivalencies (Mapping Error)	Value corresponds to financing cash flows or net debt.	De-SPAC: Leveraged buyouts/mergers confusing "Net Cash" calculations with "Cash Asset" fields.
AEVA	Aeva Technologies	Auto Tech (Lidar)	Financial equivalencies (Flow-to-Stock)	Value corresponds to operating burn/loss.	De-SPAC: Pre-revenue startups where "Net Loss" is the dominant figure, scraped as "Cash."
ALIT	Alight, Inc.	Business Services	Financial equivalencies (Mapping Error)	Value corresponds to transaction costs or investing outflows.	De-SPAC: Complex "Sources and Uses" tables in merger filings leading to line-item confusion.
ACET	Adicet Bio	Biotech	Financial equivalencies (Mapping Error)	Value corresponds to merger adjustments.	Reverse Merger: Integration of private biotech financials into public shell history.
AMRX	Amneal Pharma	Pharma	Financial equivalencies (Mapping Error)	Value corresponds to pro-forma adjustments.	Merger/Integration: Large scale integration of two entities causing restatement friction.
AIV	Aimco	Real Estate (REIT)	Financial equivalencies (Mapping Error)	Value corresponds to Spin-off distributions.	Spin-Off: Separation of assets creating negative pro-forma adjustments in Equity/Cash Flow statements.
ANSCU	Agriculture & Natural Solutions	SPAC	Financial equivalencies (Liab. as Asset)	Negative value matches working capital deficit.	Active SPAC: Shell companies often have high accrued liabilities and zero operating cash (outside Trust).

Ticker	Company Name	Sector	Error Type	Anomaly Characteristic	Prone State / Risk Factor
AAON	AAON, Inc.	Industrial	Negative fundamentals (Book Overdraft)	Negative value reflects outstanding checks > bank balance.	Small Cap / Historical: Legacy data handling of "Book Overdrafts" as negative assets rather than liabilities.
ALJJ	ALJ Regional	Conglomerate	Negative fundamentals (Book Overdraft)	Negative value reflects liquidity timing.	Small Cap / Distressed: Tight working capital management leading to reporting anomalies in data feeds.
AEL	American Equity	Insurance	Financial equivalencies (Liab. as Asset)	Value matches "Payable for Securities".	Financial Services: Netting of "Settlement Liabilities" against Cash assets (Statutory vs GAAP).
AB	AllianceBernstein	Asset Mgmt	Financial equivalencies (Liab. as Asset)	Value matches "Due to Brokers".	Financial Services: Complex cash management netting causing negative reporting.
APCX	AppTech Payments	Fintech	Financial equivalencies (Mapping Error)	Value likely matches small operating loss.	Micro-Cap/OTC: Non-standard reporting formats in OTC markets often break standard parsers.

This analysis highlighted certain edge cases which were not properly handled with the existing logic. Future implementations of this work will take these edge cases into account.

4.2 False Positives

Since the sanity check covers values that are not possible given the nature of the specific feature, false positives should be extremely unlikely. Nevertheless, the forensic analysis yielded several entries which were, in fact, false positives. This highlights the need for a thoroughly refined logic when dealing with financial data sanity checks. Future implementations of this work should accommodate the sanity check logic to be capable of handling these edge cases.

The categorization presents which company types were found to trigger false positives within the sanity check:

Table 8: Company Archetypes Triggering False Positives

Company Archetype	Example Tickers	Primary Anomaly Detected	Accounting / Structural Root Cause	State of the Company
The "Float" Optimizers	ADV, BJ, ALJJ	Negative Cash (Book Overdraft)	Utilization of Zero Balance Accounts (ZBA) and vendor float. Cash is netted against uncleared checks.	**High Efficiency / Mature:** Strong credit allows them to operate with negative working capital.
The Transformers	ADN, AEVA, ALIT, ANSCU	Asset Discontinuity / Pricing Spikes	De-SPAC mechanics, Warrant Liability restatements (SEC 2021), and Shell-to-OpCo history splicing.	**Speculative / Transitional:** Undergoing radical capital structure changes.
The "Burn" Runners	ACET, AEVA	Negative Cash / Solvency Flags	Confusion between "Accumulated Deficit" (Equity) and Cash. Reverse merger history mismatch.	**High Growth / Pre-Revenue:** Funded by equity raises, not operations. Structurally "insolvent" by GAAP income metrics.
The Regulatory Outliers	AEL, AB	Deficiency in Assets / Overdrafts	Statutory Accounting (SAP) "Deficiencies," Cash Collateral Liabilities, IFRS vs. GAAP netting.	**Regulated / Global:** Subject to non-standard accounting regimes (Insurance, Banking, International).
The Corporate Action Vectors	AAON, AMRX	Price Crashes / Cap Jumps	Unadjusted Stock Splits, Reverse Morris Trust Mergers.	**Restructuring:** Changing share count or legal entity structure.

5 Statistical Filter

This section explores a generalized approach to error detection by identifying probabilistic outliers; values that are technically possible but extremely improbable given the surrounding data context. These outliers typically result from data transmission errors, calculation precision loss, or provider-side aggregation bugs that don't trigger logic violations but distort quantitative analysis. This is an important nuance to emphasize: this approach has a nonzero

probability of false positives (theoretically 0.0001, although empirical evidence demonstrates otherwise in this case). With millions of entries spanning several thousand tickers, it is virtually impossible to not have false positives. This is why, this and every approach for treating financial data, should be combined with forensic log auditing in order to detect and correct false positives.

The functions below implement statistical outlier detection using four complementary methods, each selected for different data characteristics and error patterns. All imputation methods preserve backtesting integrity through backward-looking analysis and provide full audit logging.

5.1 `rolling_z_score`

5.1.1 Outlier Pattern Addressed

Time-series drift outliers in price and moving average data. Market prices and technical indicators exhibit trending behavior—a stock trading at $50 may drift to $100 over months. Standard deviation calculated on the entire history treats $100 as an outlier relative to early $50 data, creating false positives. This method adapts to local trends using a rolling window.

Common causes: flash crash recovery artifacts (partial tick data correction), intraday quote consolidation errors creating spurious OHLC values, and moving average calculation errors during corporate actions.

5.1.2 Approach

1. For each target column, calculate rolling mean and standard deviation using a **21-day backward-looking window** (excludes current row via `.shift(1)` to prevent look-forward bias)
2. Compute Z-scores: $Z = (\text{value} - \text{rolling\_mean})/\text{rolling\_std}$
3. Flag outliers with $0.0001 > \alpha$ and sufficient window data exists (`min_periods` = 10)
4. Treat outliers as missing values and fit a **cubic spline** on remaining valid data
5. Interpolate replacement values at outlier positions
6. Log all corrections with Z-score, rolling statistics, and interpolation method

5.1.3 Correction Method

Primary: Cubic spline interpolation (fits smooth curve through clean data). **Fallback 1:** Nearest valid value (if spline fails due to geometric constraints). **Fallback 2:** Last valid value (if spline produces non-finite result).

All interpolation uses only historical data (positions before the outlier) to maintain backtest validity.

5.1.4 Columns Processed

Time-series market data (OHLC prices, VWAP): Raw (`m_open`, `m_high`, `m_low`, `m_close`, `m_vwap`—split adjusted when necessary); Split-adjusted (`m_open_split_adjusted`, etc.); Dividend-and-split-adjusted (`m_open_dividend_and_sp` etc.).

Technical indicators (moving averages): Simple Moving Averages (`c_simple_moving_average_5d_close_*`, 21d, 63d, 252d variants); Exponential Moving Averages (`c_exponential_moving_average_5d_close_*`, 21d, 63d, 252d variants). The `*` suffix indicates both `split_adjusted` and `dividend_and_split_adjusted` variants.

5.2 `mahalanobis_filter`

5.2.1 Outlier Pattern Addressed

Multivariate outliers in fundamental data violating cross-sectional relationships. A company reporting $1B revenue with $10B operating expenses may pass univariate checks but violates industry norms. This method detects combinations of values that are improbable relative to sector peers.

Common causes: unit scaling errors affecting multiple related fields (assets in millions, liabilities in thousands), partial quarterly restatements creating temporary inconsistencies, and wrong fiscal period data assigned to a calendar quarter.

5.2.2 Approach

1. **Peer Group Identification:** Query metadata to identify all tickers in the same sector
2. **Robust Standardization:** For each quarter across all peer history, calculate robust Z-scores using median and MAD (Median Absolute Deviation)
3. **Pooled Covariance Estimation:** Fit a single robust covariance matrix (MinCovDet) on the pooled Z-score matrix of all peers across all time
4. **Distance Calculation:** Compute Mahalanobis distance from the sector centroid for the target ticker:

$$D^2 = (\mathbf{Z} - \mu)^\top \Sigma^{-1} (\mathbf{Z} - \mu) \tag{6}$$

5. **Outlier Detection:** Flag quarters where $D^2 > \chi^2(p, 1-\alpha)$ threshold (Chi-squared distribution with degrees of freedom = number of columns)
6. **Quarterly Imputation:** Replace entire flagged quarter with forward-filled values from the last clean quarter

5.2.3 Correction Method

Quarterly forward-fill: When a quarter is flagged, all fundamental values for that quarter are replaced with the previous valid quarter's values. Assumes fundamental data changes are gradual quarter-to-quarter; sudden multivariate shifts indicate corruption. If no prior clean quarter exists, values remain unchanged (no correction possible).

Limitation: Forward fill is not appropriate in every column regarding fundamental data. Ratio imputation would be a better approach for dealing with flows (fis, fcf columns).

5.2.4 Columns Processed

Balance Sheet (`fbs_` prefix): Core aggregates (`fbs_assets`, `fbs_current_assets`, `fbs_noncurrent_assets`, `fbs_liabilities`, `fbs_current_liabilities`, `fbs_noncurrent_liabilities`, `fbs_stockholder_equity`); Cash components; Operating assets; Receivables/Payables; Debt components; Equity components; and Other items.

Cash Flow Statement (`fcf_` prefix): Core flows (`fcf_net_cash_from_operating_activities`, `fcf_net_cash_from_in` `fcf_net_cash_from_financing_activities`, `fcf_free_cash_flow`); Operating activities; Investing activities; Financing activities; Tax and interest; and Cash reconciliation.

Income Statement (`fis_` prefix): Revenue/Costs; Operating expenses; Operating income; Non-operating items; Pre-tax/Tax; Net income; and Per share metrics.

Calculated Valuation Ratios (`c_` prefix): Per-share metrics; Valuation multiples; Trailing metrics; and Market cap.

5.3 mad_filter

5.3.1 Outlier Pattern Addressed

Univariate spikes in bounded or sparse data. Volume and technical oscillators often exhibit extreme spikes (10×–100× normal) that break standard deviation assumptions. RSI and similar indicators are bounded [0, 100], making them resistant to Gaussian modeling. MAD (Median Absolute Deviation) provides robust outlier detection for these distributions.

Common causes: end-of-day volume adjustments creating artificial spikes, oscillator calculation errors when price data contains gaps, and dividend record date misalignments creating spurious dividend values.

5.3.2 Approach

1. Calculate median and MAD for each column: $\text{MAD} = \text{median}(|X - \text{median}(X)|)$
2. Compute Modified Z-scores: $M = 0.6745 \times (X - \text{median})/\text{MAD}$
3. Flag outliers where $0.0001 > \alpha$
4. Treat outliers as missing and fit **cubic spline** on valid data
5. Interpolate replacement values
6. Log all corrections with Modified Z-score, median, and MAD

5.3.3 Correction Method

Primary: Cubic spline interpolation. **Fallback 1:** Nearest valid value (if spline fails). **Fallback 2:** Last valid value (if spline produces non-finite result).

Same interpolation strategy as `rolling_z_score`, but operates on the full history (not windowed) since volume/oscillators don't drift.

5.3.4 Columns Processed

Volume and traded value: Volume (`m_volume`, `m_volume_split_adjusted`, `m_volume_dividend_and_split_adjusted`); Traded value (`c_daily_traded_value`, `c_daily_traded_value_sma_5d`, 21d, 63d, 252d variants).

Technical oscillators and indicators: RSI (`c_rsi_14d_*`); MACD (`c_macd_26d_12d_*`, `c_macd_signal_9d_*`); Chaikin Money Flow (`c_chaikin_money_flow_21d_*`).

Dividend amounts (magnitude checks): Declaration date, Ex-dividend date, Record date, and Payment date dividend values.

5.4 garch_residuals

5.4.1 Outlier Pattern Addressed

Volatility-conditional price outliers. A 5% daily return is normal during market crashes but anomalous during calm periods. Standard Z-scores fail to account for volatility clustering (periods of high volatility beget more high volatility). GARCH models dynamic volatility to detect returns that are extreme *given the current market regime*.

Common causes: partial adjustment of stock splits or reverse splits (e.g., price halved but volume not doubled), timestamp errors causing end-of-day vs. intraday price mismatches, and consolidated tape errors during low-liquidity periods.

5.4.2 Approach

1. Calculate percentage returns: $r = (P_t - P_{t-1})/P_{t-1} \times 100$
2. Fit **GARCH(1,1)** with Student's t-distribution:

$$\sigma_t^2 = \omega + \alpha \cdot \varepsilon_{t-1}^2 + \beta \cdot \sigma_{t-1}^2 \tag{7}$$

3. Extract conditional volatility σ_t and compute standardized residuals: $z_t = r_t/\sigma_t$
4. Calculate dynamic threshold from t-distribution: threshold $= t^{-1}(1-\alpha/2, \nu)$ where ν = degrees of freedom
5. Flag outliers where $|z_t| >$ threshold
6. **Validate stationarity:** If $\alpha + \beta \geq 1$, the model is non-stationary; skip correction
7. Interpolate using **cubic spline** on clean data (outliers masked)
8. Log corrections with standardized residual and threshold

GARCH operates on **returns** (first differences), not levels, to handle non-stationary price series. Outliers are identified in return space but corrections are applied to price levels.

5.4.3 Correction Method

Primary: Cubic spline interpolation on price levels (using indices where return outliers were detected). **Fallback 1:** Nearest valid price (if spline fails). **Fallback 2:** Last valid price (if spline produces non-finite result).

The method requires ≥ 100 valid observations to fit GARCH reliably and skips columns where returns have near-zero variance (e.g., delisted stocks with constant prices).

5.4.4 Columns Processed

Returns: Daily log returns (`c_log_returns_dividend_and_split_adjusted`); Intraday log range (`c_log_difference_high_`

Volatility measures (annualized rolling standard deviation of log returns): 5-day, 21-day, 63-day, and 252-day variants (`c_annualized_volatility_*d_log_returns_dividend_and_split_adjusted`).

5.5 Summary of Statistical Filters

Table 9: Summary Matrix of Statistical Filter Functions

Function	Target Data Type	Detection Method	Correction Method	Min Data Required
`rolling_z_score`	Time-series (prices, MAs)	Rolling Z-score (63d window, threshold=3.5)	Cubic spline / nearest / last valid	10 valid points in window
`mahalanobis_filter`	Fundamental data (quarterly)	Mahalanobis distance on robust Z-scores (χ^2 threshold)	Quarterly forward-fill	$\geq$ ($5 \times$ columns) peer observations
`mad_filter`	Spiky univariate (volume, oscillators)	Modified Z-score via MAD (threshold=3.5)	Cubic spline / nearest / last valid	4 valid points
`garch_residuals`	Volatility/returns	GARCH(1,1) standardized residuals (t-distribution)	Cubic spline / nearest / last valid	100 valid returns

6 Statistical Filter Forensic Analysis

This is the most important part of this section. In order to discern anomalous market events from data provider errors, we need to evaluate the subset of statistical outliers detected with the previous methods by researching market events that may have caused these particular outliers. If no market event is found, it is more likely this outlier is due to a data provider error. If a market event is found, false positive flagging should be under control of the analyst.

The provided database has over 30 million entries across all tickers. Every statistical filter was programmed to have a false positive probability (confidence) of 0.0001. Nevertheless, distributional assumptions for each method are not always met. So the actual ratio is expected to be greater.

As a tool used to automate the forensic analysis, the error log was fed into a large language model (Gemini 3) tasked to scrape financial data sources (e.g., SEC EDGAR) and news sites for possible causes to these outliers. Afterwards, the model was asked to rate (from 1 to 100) how likely each data point is to be caused by the researched market event, rather than a financial data provider error. **This output should be thoroughly examined by the practitioner**, who will flag false positives accordingly. Unflagged errors will be corrected.

6.1 Research Output

Research shows that most of the outliers detected by these methodologies are false positives. Further highlighting the importance of grounding data quality checks in financial logic rather than simple generalized statistics (even if these are context aware).

These are some of the examples that were deemed to be a false positive:

Ticker	Date	Original	Corrected	Z-Score	Classification	Confidence	Evidence Summary
AAGRW	2024-09-17	0.0042	0.0058	-6.28	False Positive	Very High	Delisting confirmed (Sep 26, 2024); "Position Closing Only" status explains liquidity gap.
AACIU	2025-10-15	10.92	10.69	1.71×10^7	False Positive	Very High	Price verified by 3rd party sources; Precedes Evernorth merger announcement (Oct 20).
AACI	2023-01-26	8.56	9.95	-31.52	False Positive	High	Date matches exactly with extended Redemption Deadline; drop consistent with liquidity squeeze.
AAIC-PB	2018-01-31	22.97	24.91	-15.00	False Positive	High	Macro-correlation with Jan 2018 yield spike and dividend payment date.
AACIW	2023-01-10	0.4149	0.0861	9.59	False Positive	High	Volatility matches speculative trading prior to Extension Vote approval.
AACIW	2025-08-13	0.7035	0.3966	28.21	False Positive	Med-High	Daily price history confirms trading range in \$0.60s; correction to \$0.39 is erroneous.
AACI	2021-11-26	9.34	9.79	-20.68	Uncertain	Low	Significant drop below NAV without specific news catalyst; could be valid liquidity gap or true error.

Due to the overwhelming amount of confirmed false positives (more than 50% of the obtained 29908 logs), statistical and machine learning approaches to data provider error detection were discarded. **A generalized approach is no substitute for financial logic**

7 Limitations

This section documents the known limitations of the proposed framework, encompassing methodological constraints, mathematical assumptions that may not hold in practice, and structural issues that practitioners should consider before implementation.

7.1 Statistical Methodology Limitations

7.1.1 MAD Scaling Factor

The MAD filter applies the scaling factor 0.6745 to compute modified Z-scores:

$$M = \frac{0.6745 \times (X - \text{median})}{\text{MAD}} \tag{8}$$

This constant derives from the relationship $\text{MAD} = \Phi^{-1}(0.75) \cdot \sigma \approx 0.6745\sigma$ under normality. For heavy-tailed distributions, this calibration produces conservative thresholds (fewer detections); for bounded distributions like RSI, the opposite applies.

7.1.2 Mahalanobis Distance Assumptions

The cross-sectional peer comparison methodology assumes:

- Firms within a sector are drawn from a common multivariate distribution
- The covariance structure is stationary across time
- Multivariate near-normality (required for the χ^2 threshold)

In practice, sector heterogeneity (e.g., growth vs. value firms within Technology), time-varying correlations, and the bounded nature of financial ratios invalidate these assumptions. The MinCovDet estimator improves robustness but does not completely resolve the fundamental distributional misspecification.

7.2 Correction Method Limitations

7.2.1 Cubic Spline Extrapolation Instability

The cubic spline is fitted to 4 historical points and extrapolates to the corrupted value. This approach was chosen to preserve the "continuity" of the function governing the treated marked dat feature. Nevertheless, this approach has mathematical limitations:

- A cubic polynomial through 4 points is exactly determined (zero degrees of freedom), providing no smoothing
- Polynomial extrapolation beyond the data range is numerically unstable, particularly for cubic and higher-order polynomials
- Edge effects at series boundaries can produce extreme or non-physical values

Linear interpolation could be a more stable alternative, but it does not preserve continuity that will be useful for models fitted on this data.

7.2.2 Balance Sheet Proportional Scaling Assumptions

The correction for balance sheet identity violations:

$$\text{Factor} = \frac{\text{Total Assets}}{\text{Current Assets} + \text{Noncurrent Assets}} \tag{9}$$

implicitly assumes the *total* is correct and the *components* are erroneous. Without external validation (e.g., cross-referencing SEC filings), there is no mathematical basis for preferring one assumption over the other.

7.3 Threshold and Parameter Limitations

7.3.1 Arbitrary Detection Thresholds

Several critical thresholds are set without empirical calibration:

Table 11: Arbitrarily chosen Thresholds

Parameter	Value	Limitation
Market cap jump threshold	10×	Does not adapt to volatility or market cap tier
Rolling Z-score window	21 days	May be too short for illiquid securities
Balance sheet tolerance	5%	May flag legitimate rounding in small firms
GARCH minimum observations	100	Excludes recently listed securities
Spline fallback points	4	Unstable cubic interpolation

7.4 Structural and Architectural Limitations

7.4.1 Point Estimates Without Uncertainty Quantification

All corrections produce point estimates without confidence intervals or uncertainty flags. Output could be augmented with correction confidence scores or distributional estimates to enable more sophisticated downstream handling.

7.4.2 Limited Scope of Financial Logic Rules

The deterministic sanity checks, while superior to statistical methods for the covered cases, address only a subset of possible data errors:

- **Not covered:** Currency mismatches, segment-level vs. consolidated confusion, GAAP vs. IFRS inconsistencies
- **Partially covered:** Corporate action adjustments (splits handled; spinoffs, mergers, and rights offerings not fully addressed)

Extending the rule set to cover additional financial logic constraints would improve coverage.

7.4.3 Forensic Analysis Scalability

The current LLM-assisted forensic analysis does not scale smoothly to extremely large error logs like in this experiment:

- LLM output mentioned only the first several hundred instances (by ticker, in alphabetical order) of each error, casually neglecting most of the entries.
- News and filing availability varies by security (micro-caps have sparse coverage)

A possible fix for this problem could be batching the log research into chunks that are more easily digestible for the LLM.

7.5 Generalizability Limitations

7.5.1 Data Provider Specificity

The error patterns documented (Section 2) are specific to Financial Modeling Prep's data architecture. Other providers may exhibit different error modalities:

- Bloomberg: Different corporate action handling and adjustment methodologies
- Refinitiv: Alternative fundamental data mapping conventions
- Quandl/Nasdaq: Varying coverage of preferred shares and warrants

The sanity check functions are portable, but the error prevalence analysis and threshold calibration may require provider-specific tuning.