XBRL and Compustat, a Comparative Study

Guillaume Eymery

November 2025

Contents

1 Overview

This project builds a fully reproducible pipeline that maps raw XBRL filings into standardized financial variables comparable to S&P' Golbal's Compustat. Although XBRL offers granular, firm-reported data, it is not directly usable: firms apply different tags, custom extensions, inconsistent period definitions, and varying aggregation practices. To address this, we design a rule-based mapping system, apply systematic filtering and quarter reconstruction, and validate the resulting data against Compustat using relative error metrics and fill-rate comparisons.

Prior study has attempted to compare XBRL-reported values with Compustat (Tallapally et al. (2011), "Data Differences – XBRL versus Compustat."). However, the paper tests only a single variable (sales/revenue) for a small sample of 30 firms and does not implement any systematic parsing, tag disambiguation, or automated mapping. As a result, it provides no generalizable methodology for translating XBRL filings into a Compustat-style dataset and is now outdated given the evolution of the US–GAAP taxonomy.

In contrast, the present work provides a complete mapping framework, a robust parsing and cleaning pipeline, and quantitative validation across a wide set of financial variables.

2 Overview of XBRL

2.1 What is XBRL?

XBRL (eXtensible Business Reporting Language) is an XML–based language designed for the electronic communication of business and financial data. Much like HTML, which structures and annotates documents for web display, XBRL uses tagged elements to encode the meaning of financial concepts. For example, an HTML snippet may contain `<h1>Revenue</h1>`, while XBRL uses `<us-gaap:Revenues contextRef="CurrentYear" unitRef="USD">` ... In both languages, the tags do not compute values; they label them. The difference is that HTML expresses presentation semantics, whereas XBRL expresses accounting meaning, units, periods, and entity context.

Because XBRL is a descendant of XML, it inherits its structure, extensibility, and syntactic rules. Details can be found through the U.S. Securities and Exchange Commission website [2].

2.2 The US–General Accepted Accounting Principles Taxonomy

Public firms reporting to the SEC must use the US–GAAP Taxonomy, a large, formal dictionary of permissible XBRL tags. Each tag encodes a specific accounting concept (`us-gaap:Assets`, `us-gaap:NetIncomeLoss`), along with relationships to other concepts (summations, parent–child hierarchies). As in HTML, where the tag set is standardized (`<div>`, `<table>`), the taxonomy provides a common vocabulary. However, unlike HTML, firms may also create extensions. They are custom tags(`acme:SubscriptionRevenue`) when no existing US–GAAP concept fits. When custom tags and extensions can be useful at times, Debreceny & al. [3] showed that a majority of them are redundant and could have been filed under an already existing tag. The same economic concept may be tagged differently across firms or even across years for the same firm. For this reason, the language is in constant transformation, making it increasingly difficult for robust data parsing. Every year, the SEC updates the list of all the verified tags (as well as their occurrence) on `https://www.sec.gov/data-research/taxonomies-schemas/taxonomy-files-annual-updates`.

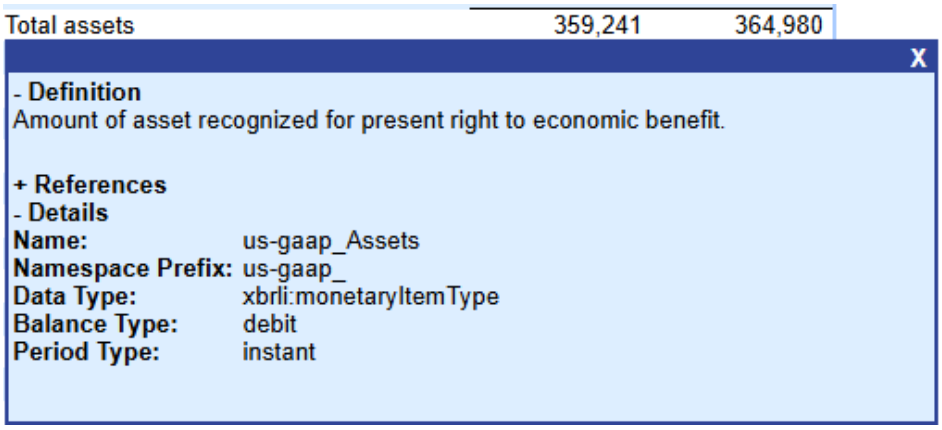


Figure 1: XBRL tag info for 2024—2023 Apple 10-K

All the tags for any publicly traded company can be retrieved from the SEC filings. [1]

2.3 Challenges from the Language Perspective

Although XBRL provides rich metadata, its flexibility creates several challenges for constructing standardized financial variables.

1. Synonymy (multiple tags for the same idea). Firms may report revenue using `us-gaap:Revenues`, or `us-gaap:RevenueFromContractWithCustomerExcludingAssessedTax`. The latter has been adopted as the standard US GAAP taxonomy for Revenues across the filings in 2019 (and is still widely used in 2025). However, a data scientist interested in retrieving data prior to this date must also account for `Revenues`.

2. Polysemy (same tag, $\neq$ meaning): `RevenueFromContractWithCustomerExcludingAssessedTax` is also frequently misused in XBRL filings: some firms tag Product Sales with it when also representing Total Revenues, creating ambiguity in data construction and parsing.

3. Context and period complexity: XBRL requires explicit definition of periods, start and end dates, units, and reporting context. Two filings may both contain the tags but refer to different fiscal periods.

4. Extensive use of extensions: mentioned in the previous section

3 Methodology: End-to-End Data Extraction Pipeline

The full pipeline transforms SEC filings into standardized, Compustat-style financial variables. The objective is to provide a hands on object oriented Python script, where the user would only need to enter a ticker (or a list of tickers) and the output would be a ready to use excel file. The columns should represent the variables and the lines the (company, year, quarter) tuple. The methodology consists of five major components described below.

3.1 Filing Identification and Retrieval

For each ticker, we first identify the firm's CIK and its recent filing history using the SEC submissions API. This step is handled by the `EdgarClient` class developped for this project:

```
client = EdgarClient(ticker, user_name, user_adress)
```

[1] `https://www.sec.gov/cgi-bin/viewer?action=view&cik=320193&accession_number=0000320193-25-000079#`

The `user_name` as well as `user_adress` are mandatory as the EDGAR API requires anyone using their service to identify themselves before making any requests. The purpose of the first API request is to look for the user's ticker in the `.json` file containing all company data general info and retrieve the CIK number [2]. This saves the user the effort of doing so manually.


```
{
  "0": {
    "cik_str": 1045810,
    "ticker": "NVDA",
    "title": "NVIDIA CORP"
  },
  "1": {
    "cik_str": 320193,
    "ticker": "AAPL",
    "title": "Apple Inc."
  },
  "2": {
    "cik_str": 789019,
    "ticker": "MSFT",
    "title": "MICROSOFT CORP"
  }...}
```


3.2 Access Raw XBRL Files

`EdgarClient` also provides a method to retrieve the accession numbers of any filing. These filings (10–K, 10–Q, 8–K, etc.) constitute the entry point to the SEC's XBRL disclosure system. Once the CIK is known, the full filing log of a company can be accessed through the SEC submissions API.[3] This response includes the description of each filing, its location in the EDGAR archive, the accession number, and the corresponding primary HTML or XML documents.

Using this information, the raw XBRL filing can be retrieved directly from the SEC's EDGAR archive via the pattern[4]:

`https://www.sec.gov/Archives/edgar/data/CIK/AccessionNumber/primary_doc.htm`

Although these HTML filings can be parsed with `BeautifulSoup` to extract numerical values, this approach is computationally costly and significantly slower than using the dedicated SEC XBRL API. For this reason, the present project relies primarily on the `companyconcept` API calls for numerical extraction.

However, retrieving the full HTML/XBRL filing remains important for a different dimension of analysis. As discussed later in the report, it can be interesting to integrate modern large language models (LLMs) to enable the extraction of qualitative, non-numeric information that cannot be obtained through the API alone.

3.3 Parsing XBRL Concepts

The core of the data retrieval is performed by the SEC's `/api/xbrl/companyconcept` endpoint, accessed through the `EdgarAPIParser`[5]:

`https://data.sec.gov/api/xbrl/companyconcept/CIK{cik}/us-gaap/{tag_name}.json`

[2] This json file can be accessed through: `https://www.sec.gov/files/company_tickers.json`

[3] Example request: `https://data.sec.gov/submissions/CIK0000320193.json` — Apple Inc.

[4] `https://www.sec.gov/Archives/edgar/data/1318605/000156459021004599/tsla-10k_20201231.htm`: Tesla , Inc. 10-K (2024)

[5] `https://data.sec.gov/api/xbrl/companyconcept/CIK0001318605/us-gaap/NetIncomeLoss.json`, Tesla Inc, Net Income

For each tag, the parser returns all available observations across time, including:

- the reported value,
- the unit of measurement,
- the period start and end dates,
- the fiscal period (Q1, Q2, Q3, FY),
- the filing date.


```
  {
"cik": 1318605,
"taxonomy": "us-gaap",
"tag": "NetIncomeLoss",
"label": "Net Income (Loss) Attributable to Parent",
"description": "The portion of profit or loss for the period, net of income taxes, which is
attributable to the parent.",
"entityName": "Tesla, Inc.",
"units": {
  "USD": [
    {
      "start": "2009-01-01",
      "end": "2009-12-31",
      "val": -55740000,
      "accn": "0001193125-12-081990", #accession number not required anymore
      "fy": 2011,
      "fp": "FY",
      "form": "10-K",
      "filed": "2012-02-27"
    },...}
```


This makes the dataset self-contained without referencing the original HTML filing. The class `EdgarAPIParser` automates the retrieval of these values.

3.4 Harmonization Through Rule-Based Mapping

A major challenge in XBRL is that different firms use different tags for the same financial concept. To solve this, we develop a rule-based mapping engine defined in a JSON file (`filing_tags_new.json`). Each Compustat variable is mapped to a sequence of rules:

1. direct match: use the first matching tag,
2. sum: combine multiple components (for instance raw materials + WIP + finished goods for inventory),
3. substract: remove contra-accounts (useful when the company reports gross revenues instead of net).
4. fallback: last-resort replacements (almost never used).


```
{
  "INCOME_STATEMENT": {
    "xsgaq": {
      "rules": [
        {
          "method": "direct",
          "tags": ["OperatingExpenses"]
        },
```

```
      {
        "method": "sum",
        "tags": ["SellingGeneralAndAdministrativeExpense",
        "ResearchAndDevelopmentExpense",
        "SellingAndMarketingExpense"],
        "_comment" : "Compustat integrates R&D in SG&A
        - xsgaq includes all operating expenses"
      },
      {
        "method": "fallback",
        "tags": ["GeneralAndAdministrativeExpense"]
      }
    ]
  },...}...}
```

When some specific tags do not exist, returns 0. The objective of this `.json` file is to create a link between Compustat variables and the ones extracted from XBRL files. However, some values need additional transformations to better map the Compustat benchmark.

3.5 Filtering & De-duplication Logic

The output of the mapping stage contains overlapping observations, amended filings, and year-to-date rows that must be removed. This is handled by the `SmartFiltering` class.

Removal of amended filings and Duration Checks. Some filings report twice the same information. It is the case of the 10-K/A (which is an amended 10-K), but has to be filtered out to not count the same year twice. Similarly, some quarterly filings can report 6-months results or YTD results in their Income Statement in addition to the traditional quarter result. To avoid accounting for these, we decide to calculate the duration for each value (end date - start date) and remove all the income statement values that has a duration greater than 100 days (92 days = 1 quarter, on which we add a buffer) or less than 355 days.

This ensures the dataset reflects genuine quarter or fiscal-year values.

3.6 Quarter Reconstruction for Q4

Many firms do not report Q4 directly; instead, they provide a full-year (FY) value. To compute Q4, we apply the convention:

$$Q4 = FY - (Q1 + Q2 + Q3)$$

This is automated in the `QuarterMapping` class. It produces a complete quarterly time series even for firms that never disclose Q4 explicitly. In the early years (2005 - 2014), quarterly reports are not always available for some companies; in this instance we decide to keep FY data and not to compute Q4.

This process intends to yield a standardized, Compustat-like dataset constructed entirely from public filings.

4 Empirical Evaluation

This section evaluates the quality of the XBRL-derived dataset by comparing it to Compustat, following two dimensions: overall data coverage and structure, and numerical accuracy of each mapped financial variable.

4.1 Data Overview

The final dataset is constructed by running the full extraction and mapping pipeline for a representative set of large public firms. In the main retrieval script (`main.py`), we process 100 widely followed tickers.

The resulting file contains a panel of quarterly and annual observations, with one column per standardized variable (`saleq`, `cogsq`, `niq`, `actq`, `dlcqy`, etc.). The dataset is therefore directly comparable to the structure of Compustat's database.

To support evaluation, we also load the WRDS template (`CompustatDataTemplate.csv` for example) and align variable names, units, and date formats so both sources share:

- a common period identifier (`end`),
- a common firm identifier (`ticker`),
- the same variable naming convention.

4.2 Validation Against Compustat

We evaluate accuracy through two diagnostics implemented in `data_comparison.py`: relative error computation, and fill-rate comparison.

Nearest-date matching. Because XBRL filings may differ slightly in reporting dates relative to Compustat's normalized calendar, we match each XBRL observation to its closest Compustat value within a $\pm$10-day tolerance window:

Relative error computation. For each variable present in both datasets, we compute the relative deviation:

$$\text{Error} = \frac{|v_{\text{XBRL}}| - |v_{\text{Compustat}}|}{|v_{\text{XBRL}}|} \times 100.$$

The script computes this for all overlapping variables and stores the results in a DataFrame object. Histograms for every variable are automatically generated, clipped to the 1st–99th percentile range, and exported as a PDF file. This provides a visual assessment of the magnitude and dispersion of errors across firms and periods.

Fill-rate comparison. Accuracy must be evaluated jointly with coverage. We therefore compute the ratio of non-missing, non-zero observations in the XBRL dataset relative to Compustat:

$$\text{FillRate}_{i,j} = \frac{\#\{XBRL_{i,j} \neq 0\}}{\#\{Compustat_{i,j} \neq 0\}},$$

for ticker i and variable j. The resulting table reveals which variables are well populated in public XBRL filings and which ones show inconsistencies or sparse tagging.

To further analyze coverage, the fill-rate results can be reorganized into two pivot tables.

1. Fill rates per variable, showing which accounting concepts are reliably disclosed across firms and which have systematic gaps.
2. Fill rates per company, highlights differences in reporting completeness across issuers—some firms tag a broad and consistent set of US–GAAP items, while others rely heavily on extensions or omit certain components.

Together, these two pivot perspectives provide a clear view of where the XBRL reporting is strong and where structural inconsistencies remain.

5 Results

5.1 Overall Data Quality and Coverage

Our validation of XBRL-extracted financial data against Compustat reveals a dataset with strong overall coverage but identifiable structural inconsistencies. The aggregate fill rate across all variables and companies is 0.83 (indicating that XBRL filings capture approximately 83% of the financial metrics available in Compustat). This suggests that while XBRL reporting is substantially complete for most firms and periods, some gaps remain in certain accounting concepts and company disclosures.

The evolution of fill rates over time, depicted in Table 2 (Fill Rate by Year and Quarter), shows an upward trend from 2010 to 2025. The earliest quarters (2010 Q1–Q2) record fill rates of approximately 0.60–0.67, reflecting early-stage XBRL adoption and incomplete tagging standards. By 2017–2018, fill rates stabilize around 0.75–0.86, and continue to improve through 2022, reaching 0.88. This trajectory suggests increasing compliance with XBRL reporting requirements as financial institutions refined their tagging practices over the decade. [6]

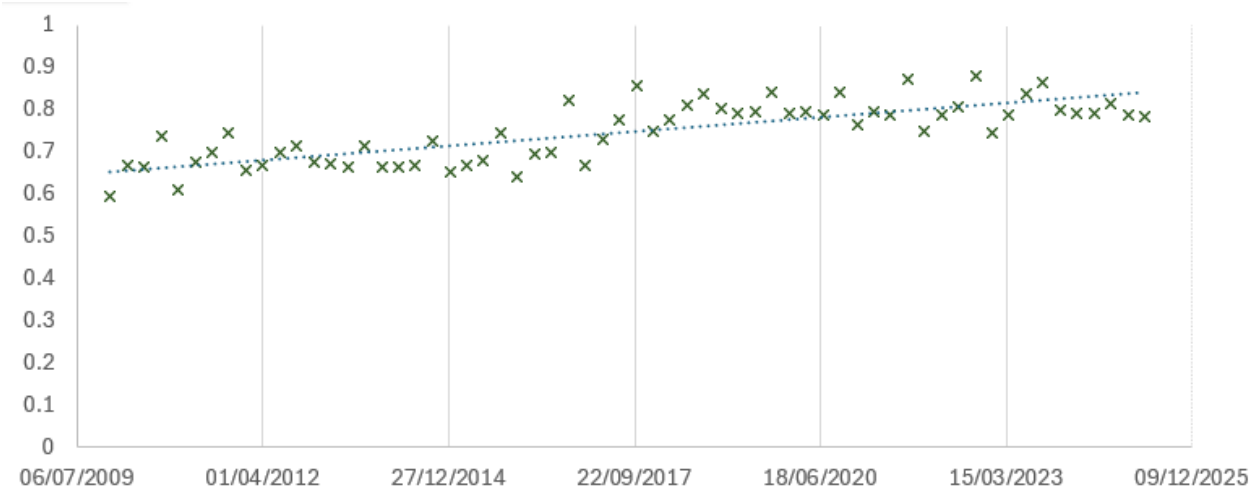


Figure 2: Fill Rates per Year - Evolution

5.2 Variable-Level Analysis: Strengths and Vulnerabilities

Table 2 (Average Fill Rate by Compustat Variable) reveals considerable variation in coverage across accounting line items. Variables with fill rates exceeding 0.95 include fundamental balance sheet items: total current assets (actq: 0.98), total current liabilities (lctq: 0.98), total assets (atq: 0.97), total liabilities and equity (lseq: 0.97), intangible assets (intanq: 0.94), and equity method investments (ivaeqq: 0.97). These high fill rates reflect the importance of these items to financial reporting and their consistent tagging in XBRL filings since 2010.

In contrast, several variables exhibit low coverage. Specifically cash flow and income statement items : short-term investment changes (ivstchy: 0.26), pre-tax income (piq: 0.38), and preferred stock (pstkq: 0.42). Extraordinary items such as income from discontinued operations (xidoq: 1.00) show fill rates near unity but reduced practical value given their infrequent occurrence.

These gaps likely reflect inconsistent or discretionary XBRL tagging practices, as firms may employ different taxonomies for specialty items (or eventually not report them at all on their statements) or rely on custom US–GAAP extensions rather than standard tags.

[6] 2025 data should be disregarded due to incomplete collection: Q3 and Q4 show anomalously low Compustat values (735 and 0 observations, respectively), indicating data collection was interrupted.

5.3 Company-Level Disparities

Table 3 (Average Fill Rate by Company) exposes variation in XBRL reporting. Most companies cluster around the 0.70–0.85 range, consistent with the aggregate 0.83 mean. However, several firms deviate from this distribution.

BlackRock (BLK: 0.10), Disney (DIS: 0.39), and Broadcom (AVGO: 0.45) all show fill rates below 0.50. These low values mostly reflect gaps in their historical XBRL filings rather than parsing issues. For instance, BlackRock has no SEC XBRL statements before 12/31/2022, which naturally lowers its coverage. Broadcom[7] only began consistent XBRL reporting around 2018, so its Compustat history extends far beyond what is available in EDGAR. As a result, their low fill rates are driven by reporting limitations, not extraction errors.

Some companies (CI: 0.47, LIN: 0.49, JPM: 0.52, PLD: 0.54, GS: 0.56, WFC: 0.59, PNC: 0.44) fall into a secondary tier of moderate underreporting (0.47–0.59), suggesting that industry-specific accounting practices influence their XBRL disclosures. Financial services firms (JPM, GS, WFC, PNC) are particularly present in this group. This pattern is consistent with the fact that banks and insurance companies follow reporting frameworks that differ from those used in non-financial sectors, which the current JSON mapping does not yet fully incorporate.

5.4 Temporal Stability and Consistency

The quarter-by-quarter analysis (Table 3) documents two patterns:

- Seasonal patterns within years: Q4 consistently shows higher fill rates (0.74–0.88) relative to Q1–Q3 (typically 0.61–0.80). This pattern is robust across all years from 2010 to 2024, suggesting that year-end filings receive more comprehensive XBRL tagging than interim reports. The difference ranges from 0.08 to 0.16 percentage points, a gap indicating either improved regulatory scrutiny of annual filings or greater institutional effort applied to year-end disclosures.
- Long-term improvement trajectory: The increase from 2010 (0.60–0.74) to 2022 (0.75–0.88) reflects the maturation of XBRL adoption and increasing standardization. The leveling off around 0.79–0.88 across 2020–2024 indicates a new equilibrium, where further improvements would require addressing the gaps identified in the variable-level analysis.

5.5 Error Distribution and Relative Accuracy

To assess accuracy beyond fill rates, we analyze the relative error between XBRL-extracted values and their Compustat counterparts. The error histograms reveal two main patterns: a large group of variables with errors tightly centered around zero, and another group with heavy-tailed or asymmetric errors.

- Variables with Near-Zero Error Distributions: The majority of core accounting items produce histograms tightly concentrated around zero, with minimal dispersion. On pages 1–2 of the histogram set, variables such as actq (Assets Current), lctq (Liabilities Current), atq (Total Assets), ceqq (Ordinary Equity), txpq (Income Taxes Payable)... show extremely narrow spikes at zero, indicating that XBRL and Compustat values match almost (if not) exactly for most observations. These variables correspond to the highest fill-rate items identified earlier and reflect standardized, frequently disclosed GAAP line items. Their accuracy confirms that the parsing pipeline interprets core numerical tags reliably on a large scale.
- Heavy-Tailed or Asymmetric Error Patterns: A second subset of variables displays heavy dispersion, long tails, or asymmetric error distributions. These patterns appear frequently in operating expense, cost variables and "Other" (cogsq — Cost of Goods Sold, lcoq — Other Liabilities Current).

[7] `https://data.sec.gov/api/xbrl/companyconcept/CIK0001730168/us-gaap/Assets.json`

Why Some Variables Cannot Be Reconciled Exactly to Compustat A closer inspection of specific items (for instance "Other Current Liabilities" (LCOQ) and its annual counterpart (LCO)) reveals that some disparities are due to differences in the construction of financial aggregates. Compustat derives these items as residual categories using internal standardization rules, often combining components that are not individually disclosed in the firm's XBRL filings. By contrast, the XBRL taxonomy only provides what issuers explicitly report; if a company aggregates multiple heterogeneous components into a single line item, the underlying detail needed to reproduce Compustat's definition is not available.

For example, Compustat defines LCOQ as:

> *"a residual item representing those current liabilities that are not debt, trade accounts payable, or income taxes payable. This item is not available for banks. It is the sum of Accrued Expense (XACC) and Current Liabilities–Other–Sundry (LCOX), plus additional industry-specific components."*

In practice, this category may include dozens of elements such as customer deposits, billings in excess of cost, film rights payable, unredeemed gift certificates, estimated litigation claims, and other reserves. Many of these components are never separately tagged in XBRL; they appear only as part of a firm-reported "Other current liabilities" line.

This definitional mismatch is visible when comparing firm filings to Compustat. Consider the following excerpt from a recent SEC 10-K filing (Apple Inc., USD millions):

Current liabilities (excerpt)	**2025**	**2024**
Accounts payable	69,860	68,960
Other current liabilities	66,387	78,304
Deferred revenue	9,055	8,249
Commercial paper	7,979	9,967
Term debt	12,350	10,912
Total current liabilities	165,631	176,392

In this example, "Other current liabilities" is provided as a single consolidated line item. However, Compustat's construction of LCOQ requires separating this line into multiple conceptual buckets (accrued expenses, deferred taxes payable, sundry accounts...). Since the filing does not disclose these components individually, it is mathematically impossible to reconstruct Compustat's value exactly from XBRL alone. The parser can extract the raw reported value, but cannot infer the unobserved decomposition required by Compustat's standardized methodology.

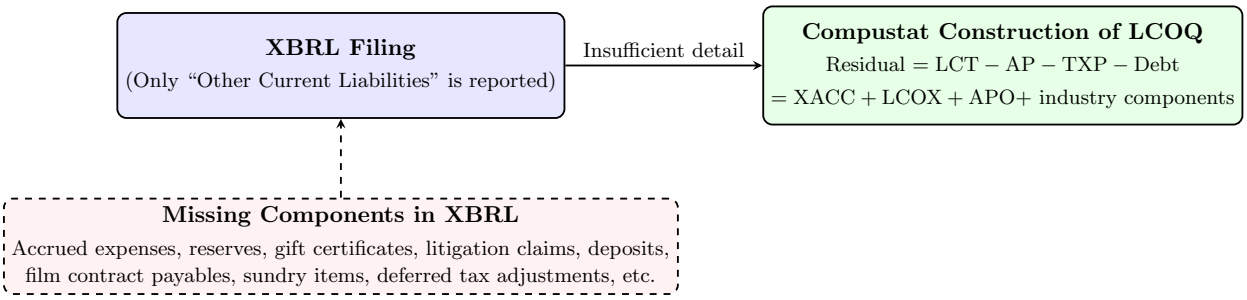


Figure 3: Conceptual mismatch between reported XBRL line items and Compustat construction of LCOQ.

This diagram highlights that even when the parser extracts all available XBRL tags correctly, the absence of the granular components that Compustat uses prevents exact reconciliation. Compustat's methodology effectively embeds industry knowledge, aggregation rules, and multi-year consistency checks that cannot be replicated using only the raw XBRL tags disclosed in firm filings.

This diagram shows that even with perfect XBRL extraction, missing granular components make it impossible to fully match Compustat's values. Compustat applies industry rules, aggregation choices, and historical

adjustments that cannot be reproduced from the raw tags firms report.

These irreconcilable aggregates explain why variables such as `lcoq`, `cogsq`, and several cash-flow items produce heavy-tailed error distributions. Their values rely on Compustat's internal adjustments or residual calculations, not on information available in XBRL. As a result, large relative deviations reflect structural differences—not parsing errors.
This distinction is essential for interpreting error histograms: systematic deviations in these categories reflect limitations in available public data, not deficiencies in the extraction pipeline.

The observations in Appendix indicate that the parsing pipeline performs well for standardized GAAP concepts. However, improving and updating the JSON tag mapping remains essential to limit the remaining discrepancies.

6 Discussion

The results highlight an important distinction between the availability of financial data in XBRL filings and the standardization applied by commercial databases such as Compustat. While the EDGAR XBRL infrastructure provides rapid, efficient access to firm-reported financial statements, it is inherently constrained by the level of detail that firms choose to disclose. Many line items—particularly residual categories such as "Other Current Liabilities", investment subcomponents, and non-operating adjustments—are reported only in aggregated form. As a consequence, certain Compustat variables cannot be reproduced exactly, even when the XBRL parsing is technically flawless.

At the same time, XBRL offers capabilities that Compustat does not (besides being free for use). Beyond numerical values, filings contain rich narrative information: risk factor discussions, footnotes, disclosures, and managerial commentary. These textual components can be obtainable through XBRL's structured HTML and XML layers. As large language models continue to advance, it becomes increasingly feasible to automatically parse, classify, and evaluate these narrative disclosures at scale. This opens new possibilities for research and analytics and sentiment extraction.

7 Conclusion

The empirical analysis demonstrates that the quality of XBRL-reported financial data has improved over time, approaching a steady state of reliable coverage for core accounting variables. For foundational items such as revenues, assets, liabilities, and major cash-flow components, XBRL-based extraction closely matches Compustat in both completeness and accuracy. However, structural limitations persist for specialized, infrequent, or residual items whose components are either undisclosed or inconsistently tagged across firms. These discrepancies arise not from issues in parsing, but from differences in conceptual construction: Compustat applies proprietary standardization rules that cannot always be replicated using publicly available XBRL tags.

Despite these limitations, XBRL remains a powerful resource. Its structured format, instant availability through the EDGAR API, and access to textual disclosures make it highly valuable for empirical research and automated analysis. As filer compliance improves and SEC taxonomy standards evolve, the gap between firm-reported XBRL data and standardized databases should continue to shrink.

In summary, XBRL is good at providing broad, timely, and text-rich disclosures, while Compustat provides harmonized and research-ready financial aggregates. The two systems are best viewed as complementary: XBRL for accessibility and detail, Compustat for consistency and standardization.

8 Appendix

Table 1: Compustat Variable Definitions

Statement	Variable	Compustat Definition
Income Statement	saleq	Quarterly Sales / Revenues
	cogsq	Cost of Goods Sold
	xsgaq	Selling, General, and Administrative Expense
	oibdpq	Operating Income Before Depreciation
	dpq	Depreciation and Amortization
	oiadpq	Operating Income After Depreciation
	xintq	Interest and Related Expense
	nopiq	Nonoperating Income (Expense)
	piq	Pretax Income (Income Before Extraordinary Items)
	txtq	Income Taxes
	ibq	Income Before Extraordinary Items
	dvpq	Preferred Dividends
	xidoq	Income from Discontinued Operations
	niq	Net Income
Cash Flow Statement	ibcy	Net Income (YTD)
	dpcy	Depreciation and Amortization (YTD)
	xidocy	Income from Discontinued Operations (YTD)
	txdcy	Deferred Taxes (YTD)
	esubcy	Equity in Earnings of Unconsolidated Subsidiaries (YTD)
	sppivy	Special Items (YTD)
	fopoy	Other Funds from Operations (YTD)
	recchy	Change in Receivables (YTD)
	invchy	Change in Inventories (YTD)
	apalchy	Change in Accounts Payable and Accrued Liabilities (YTD)
	txachy	Change in Income Taxes Payable (YTD)
	oancfy	Net Cash from Operating Activities (YTD)
	ivchy	Capital Expenditures for Investments (YTD)
	sivy	Proceeds from Sales of Investments (YTD)
	capxy	Capital Expenditures (YTD)
	sppey	Proceeds from Sale of PPE (YTD)
	aqcy	Acquisitions (YTD)
	ivstchy	Change in Short-Term Investments (YTD)
	ivacoy	Other Investing Activities (YTD)
	ivncfy	Net Cash from Investing Activities (YTD)
	sstky	Sale of Common and Preferred Stock (YTD)
	prstkccy	Purchase of Common Stock (YTD)
	dvy	Dividends Paid (YTD)
	dltisy	Long-Term Debt Issued (YTD)
	dltry	Long-Term Debt Retired (YTD)
	dlcchy	Change in Short-Term Debt (YTD)
	fiaoy	Other Financing Activities (YTD)
	fincfy	Net Cash from Financing Activities (YTD)
	exrey	Effect of Exchange Rate Changes (YTD)
	chechy	Change in Cash and Equivalents (YTD)
Assets	chq	Cash and Cash Equivalents
	ivstq	Short-Term Investments
	rectq	Accounts Receivable
	invtq	Inventories
	acoq	Other Current Assets
	actq	Total Current Assets
	ppentq	Net Property, Plant, and Equipment

Statement	Variable	Compustat Definition
	ivaeqq	Equity Method Investments
	ivaoq	Other Long-Term Investments
	intanq	Intangible Assets (including Goodwill)
	aoq	Other Noncurrent Assets
	atq	Total Assets
Liabilities	dlcq	Current Debt
	apq	Accounts Payable
	txpq	Income Taxes Payable
	lcoq	Other Current Liabilities
	lctq	Total Current Liabilities
	dlttq	Long-Term Debt
	loq	Other Long-Term Liabilities
	txditcq	Deferred Taxes (Net)
	mibq	Minority Interest (Noncontrolling Interest)
	ltq	Total Liabilities
Stockholders' Equity	pstkq	Preferred Stock
	ceqq	Common Equity / Total Equity
	lseq	Total Liabilities and Equity

Table 2: Average Fill Rate by Compustat Variable

Variable	Average Fill Rate
acoq	0.48
actq	0.98
aoq	0.29
apalchy	6.29
apq	0.75
aqcy	0.87
atq	0.97
capxy	0.74
ceqq	0.91
chechy	0.50
chq	0.90
cogsq	0.49
dlcchy	1.14
dlcq	0.52
dltisy	0.61
dltry	0.50
dlttq	0.61
dpcy	0.61
dpq	0.49
dvpq	0.49
dvy	0.84
exrey	0.47
fiaoy	0.75
fincfy	0.89
fopoy	0.47
ibcy	0.96
ibq	0.38
intanq	0.94
invchy	0.78
invtq	0.73
ivacoy	0.73
ivaeqq	0.97
ivaoq	0.98
ivchy	0.88
ivncfy	0.89
ivstchy	0.26
ivstq	0.58
lcoq	0.70
lctq	0.98
loq	0.80
lseq	0.97
ltq	0.97
mibq	1.12
niq	0.91
oancfy	0.90
oiadpq	0.74
oibdpq	0.84
piq	0.38
ppentq	0.88
prstkccy	1.06
pstkq	0.42

Variable	Average Fill Rate
recchy	0.58
rectq	0.78
saleq	0.55
sivy	0.61
sppey	1.79
sppivy	0.91
sstky	0.34
txdcy	1.34
txditcq	0.75
txpq	0.87
txtq	0.98
xidocy	1.49
xidoq	1.00
xintq	0.76
xsgaq	0.80
Average Total	0.83

Table 3: Average Fill Rate by Company

Company	Average Fill Rate
AAPL	0.81
ABBV	0.70
ABT	1.81
ACN	0.83
ADBE	0.94
ADP	0.80
AMAT	0.71
AMD	0.88
AMGN	0.85
AMZN	1.07
AVGO	0.45
AXP	0.68
BA	0.69
BAC	0.50
BK	0.62
BKNG	0.65
BLK	0.10
C	0.75
CAT	0.74
CI	0.47
CL	0.73
CMCSA	0.78
COP	0.88
COST	0.84
CRM	1.08
CSCO	0.81
CSX	0.74
CVS	0.85
CVX	0.64
DE	0.71
DHR	1.85
DIS	0.39

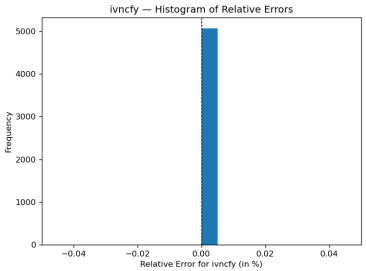
Company	Average Fill Rate
DUK	0.67
ELV	0.83
FIS	0.77
GE	0.65
GILD	0.82
GOOGL	0.54
GS	0.56
HD	0.70
HON	0.69
IBM	0.68
INTC	1.10
INTU	0.81
ISRG	0.81
JNJ	0.72
JPM	0.52
KO	0.88
LIN	0.49
LLY	0.78
LMT	0.79
LOW	0.67
MA	0.78
MCD	0.78
MDLZ	0.83
MDT	0.57
META	0.75
MMC	0.70
MMM	0.68
MO	0.83
MRK	0.78
MS	1.13
MSFT	0.78
NEE	1.17
NFLX	0.93
NKE	0.72
NOW	0.69
NVDA	2.08
ORCL	0.74
PEP	0.67
PFE	0.78
PG	1.77
PLD	0.54
PM	1.83
PNC	0.44
PYPL	0.87
QCOM	0.85
REGN	0.79
RTX	0.76
SBUX	0.85
SCHW	0.65
SLB	0.80
SO	0.61
SPGI	0.83
T	0.69

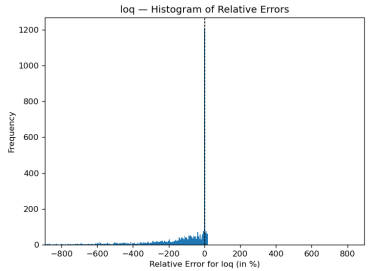
Company	Average Fill Rate
TMO	0.87
TSLA	0.68
TXN	0.73
UNH	1.96
UNP	1.33
UPS	0.88
V	0.93
VRTX	0.81
VZ	0.76
WFC	0.59
WMT	0.82
XOM	0.61
ZTS	1.49
Av. Fill Rate	**0.83**

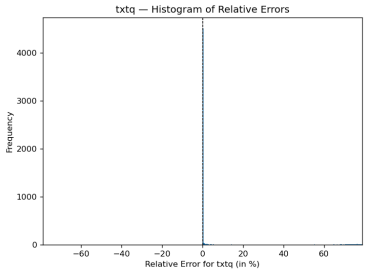
Table 4: Fill Rate by Year and Quarter

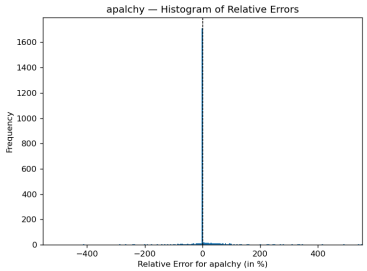
Year	Quarter	Parsing Collected Values	Compustat Collected Values	Fill Rate Ratio
2010	Q1	2718	4567	0.60
2010	Q2	3102	4654	0.67
2010	Q3	3071	4636	0.66
2010	Q4	3566	4834	0.74
2011	Q1	2799	4596	0.61
2011	Q2	3159	4684	0.67
2011	Q3	3252	4660	0.70
2011	Q4	3623	4865	0.74
2012	Q1	3095	4723	0.66
2012	Q2	3180	4757	0.67
2012	Q3	3354	4802	0.70
2012	Q4	3511	4916	0.71
2013	Q1	3206	4740	0.68
2013	Q2	3227	4802	0.67
2013	Q3	3207	4830	0.66
2013	Q4	3563	4984	0.71
2014	Q1	3154	4759	0.66
2014	Q2	3185	4805	0.66
2014	Q3	3244	4843	0.67
2014	Q4	3625	4986	0.73
2015	Q1	3164	4837	0.65
2015	Q2	3280	4896	0.67
2015	Q3	3349	4915	0.68
2015	Q4	3742	5012	0.75
2016	Q1	3092	4831	0.64
2016	Q2	3400	4889	0.70
2016	Q3	3439	4919	0.70
2016	Q4	4112	4992	0.82
2017	Q1	3230	4838	0.67
2017	Q2	3562	4882	0.73
2017	Q3	3796	4891	0.78
2017	Q4	4286	4995	0.86
2018	Q1	3621	4827	0.75
2018	Q2	3760	4855	0.77

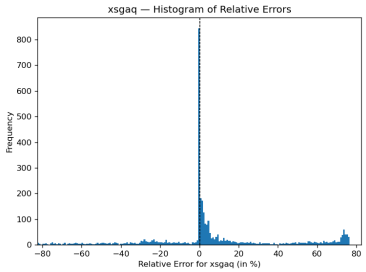
Year	Quarter	Parsing Collected Values	Compustat Collected Values	Fill Rate Ratio
2018	Q3	3948	4877	0.81
2018	Q4	4171	4980	0.84
2019	Q1	3897	4845	0.80
2019	Q2	3865	4876	0.79
2019	Q3	3907	4909	0.80
2019	Q4	4216	5006	0.84
2020	Q1	3840	4845	0.79
2020	Q2	3893	4904	0.79
2020	Q3	3875	4920	0.79
2020	Q4	4214	5019	0.84
2021	Q1	3693	4825	0.77
2021	Q2	3869	4859	0.80
2021	Q3	3851	4887	0.79
2021	Q4	4347	4986	0.87
2022	Q1	3617	4822	0.75
2022	Q2	3834	4869	0.79
2022	Q3	3961	4909	0.81
2022	Q4	4386	4988	0.88
2023	Q1	3610	4839	0.75
2023	Q2	3852	4894	0.79
2023	Q3	4116	4913	0.84
2023	Q4	4315	4981	0.87
2024	Q1	3867	4831	0.80
2024	Q2	3859	4878	0.79
2024	Q3	3884	4907	0.79
2024	Q4	4056	4981	0.81
2025	Q1	3798	4818	0.79
2025	Q2	3821	4868	0.78
2025	Q3	3848	735	5.24
2025	Q4	204	0	0.00

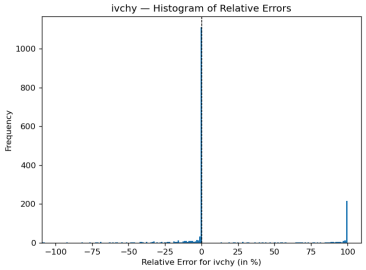

ivncfy — Histogram of Relative Errors
Frequency
Relative Error for ivncfy (in %)

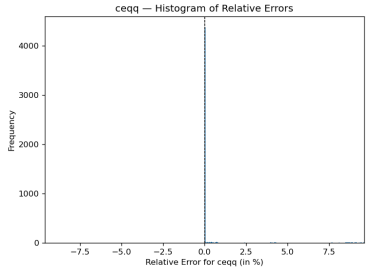

loq — Histogram of Relative Errors
Frequency
Relative Error for loq (in %)

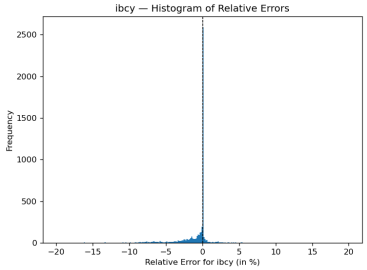

txtq — Histogram of Relative Errors
Frequency
Relative Error for txtq (in %)

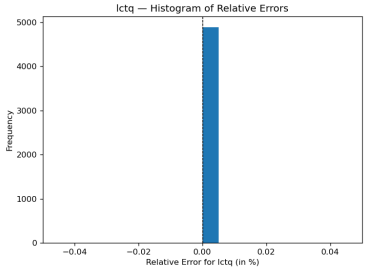

apalchy — Histogram of Relative Errors
Frequency
Relative Error for apalchy (in %)

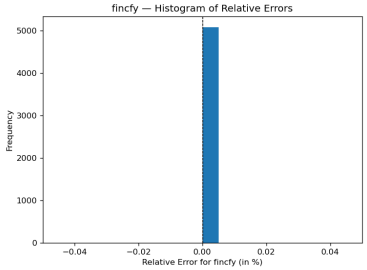

xsgaq — Histogram of Relative Errors
Frequency
Relative Error for xsgaq (in %)

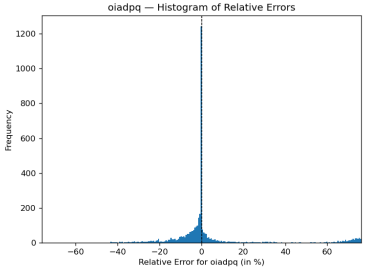

ivchy — Histogram of Relative Errors
Frequency
Relative Error for ivchy (in %)

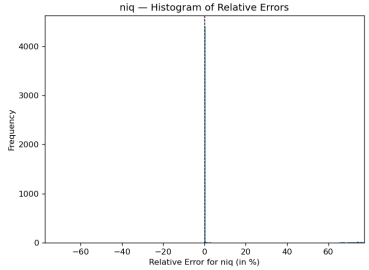

ceqq — Histogram of Relative Errors
Frequency
Relative Error for ceqq (in %)

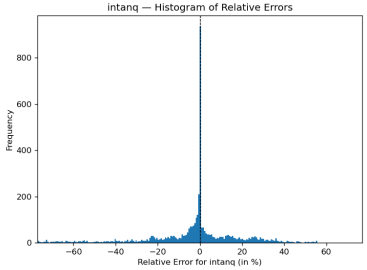

ibcy — Histogram of Relative Errors
Frequency
Relative Error for ibcy (in %)

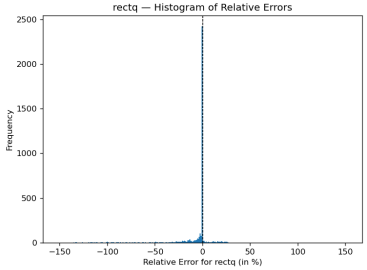

lctq — Histogram of Relative Errors
Frequency
Relative Error for lctq (in %)

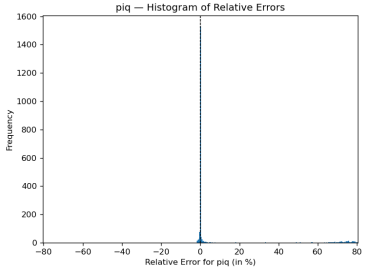

fincfy — Histogram of Relative Errors
Frequency
Relative Error for fincfy (in %)

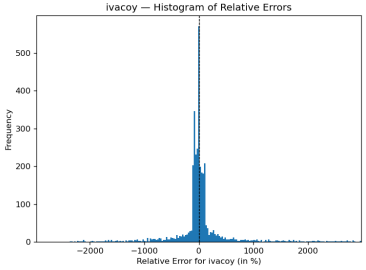

oiadpq — Histogram of Relative Errors
Frequency
Relative Error for oiadpq (in %)

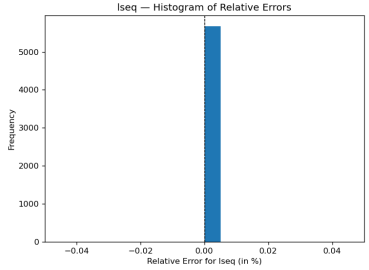

niq — Histogram of Relative Errors
Frequency
Relative Error for niq (in %)

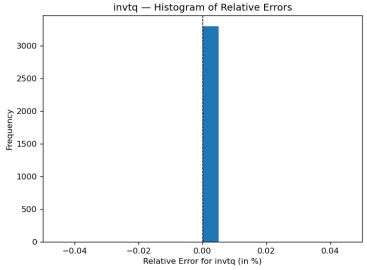

intanq — Histogram of Relative Errors
Frequency
Relative Error for intanq (in %)

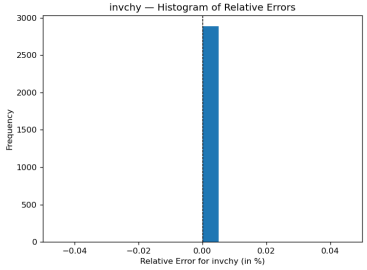

rectq — Histogram of Relative Errors
Frequency
Relative Error for rectq (in %)

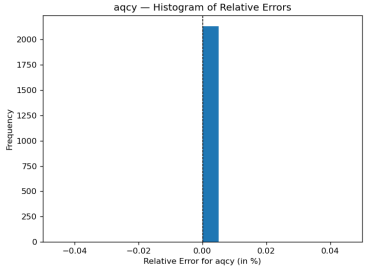

piq — Histogram of Relative Errors
Frequency
Relative Error for piq (in %)

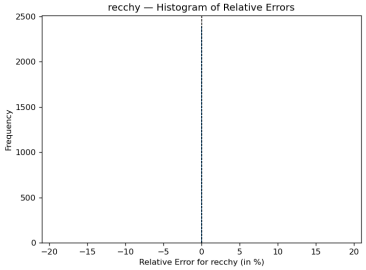

ivacoy — Histogram of Relative Errors
Frequency
Relative Error for ivacoy (in %)

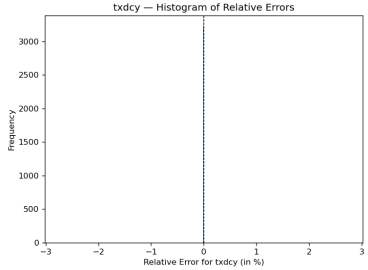

lseq — Histogram of Relative Errors
Frequency
Relative Error for lseq (in %)

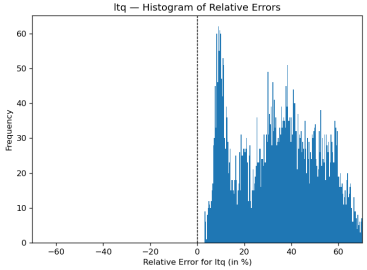

invtq — Histogram of Relative Errors
Frequency
Relative Error for invtq (in %)

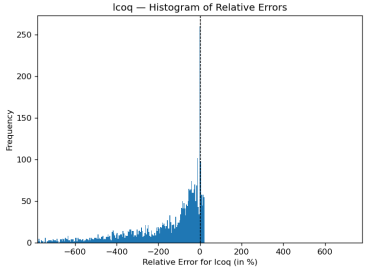

invchy — Histogram of Relative Errors
Frequency
Relative Error for invchy (in %)

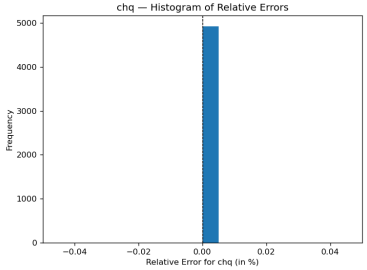

aqcy — Histogram of Relative Errors
Frequency
Relative Error for aqcy (in %)

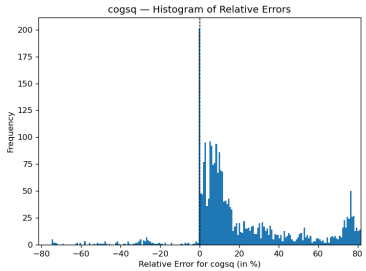

recchy — Histogram of Relative Errors
Frequency
Relative Error for recchy (in %)

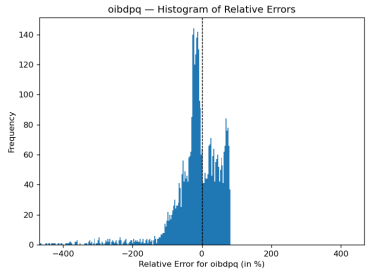

txdcy — Histogram of Relative Errors
Frequency
Relative Error for txdcy (in %)

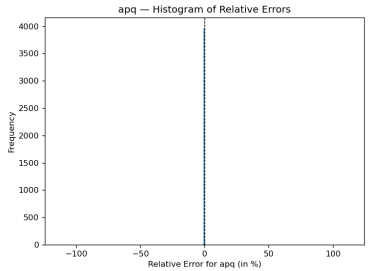

ltq — Histogram of Relative Errors
Frequency
Relative Error for ltq (in %)

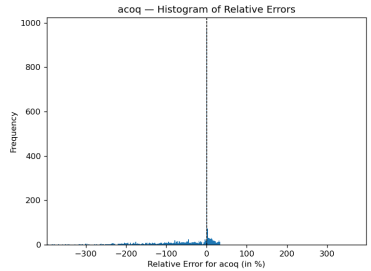

lcoq — Histogram of Relative Errors
Frequency
Relative Error for lcoq (in %)

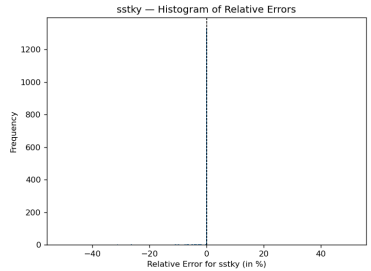

chq — Histogram of Relative Errors
Frequency
Relative Error for chq (in %)

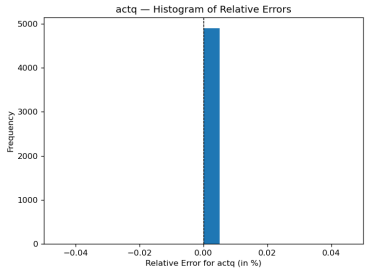

cogsq — Histogram of Relative Errors
Frequency
Relative Error for cogsq (in %)

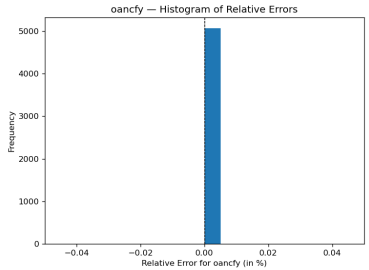

oibdpq — Histogram of Relative Errors
Frequency
Relative Error for oibdpq (in %)

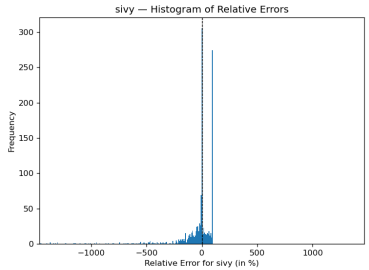

apq — Histogram of Relative Errors
Frequency
Relative Error for apq (in %)

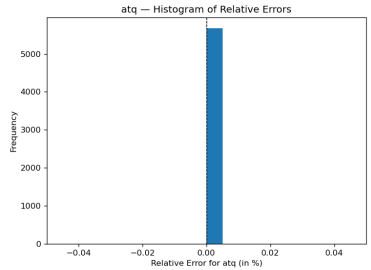

acoq — Histogram of Relative Errors
Frequency
Relative Error for acoq (in %)

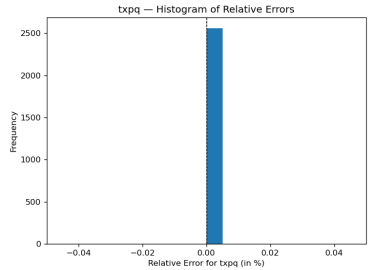

sstky — Histogram of Relative Errors
Frequency
Relative Error for sstky (in %)

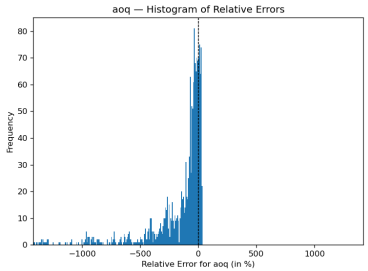

actq — Histogram of Relative Errors
Frequency
Relative Error for actq (in %)

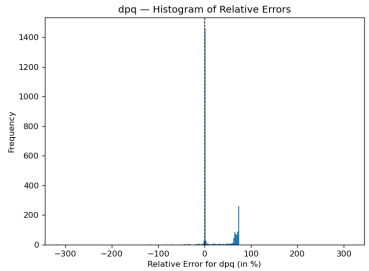

oancfy — Histogram of Relative Errors
Frequency
Relative Error for oancfy (in %)

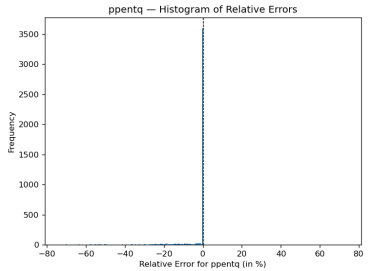

sivy — Histogram of Relative Errors
Frequency
Relative Error for sivy (in %)

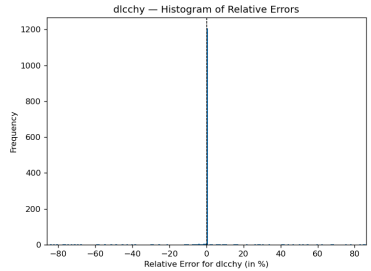

atq — Histogram of Relative Errors
Frequency
Relative Error for atq (in %)

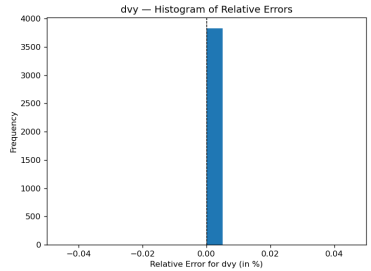

txpq — Histogram of Relative Errors
Frequency
Relative Error for txpq (in %)

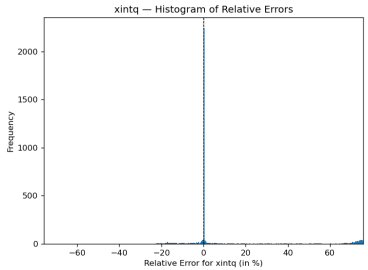

aoq — Histogram of Relative Errors
Frequency
Relative Error for aoq (in %)

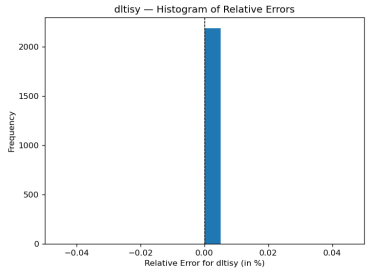

dpq — Histogram of Relative Errors
Frequency
Relative Error for dpq (in %)

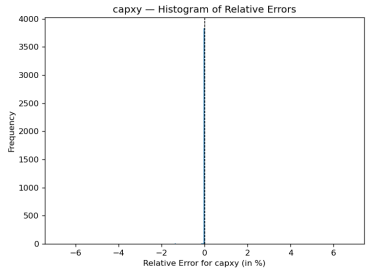

ppentq — Histogram of Relative Errors
Frequency
Relative Error for ppentq (in %)

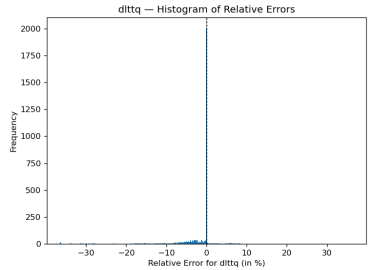

dlcchy — Histogram of Relative Errors
Frequency
Relative Error for dlcchy (in %)

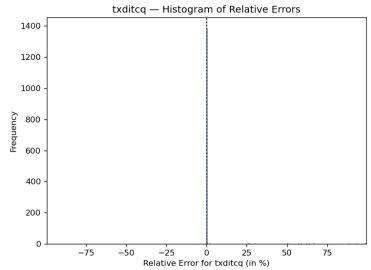

dvy — Histogram of Relative Errors
Frequency
Relative Error for dvy (in %)

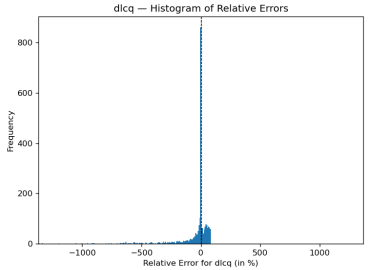

xintq — Histogram of Relative Errors
Frequency
Relative Error for xintq (in %)

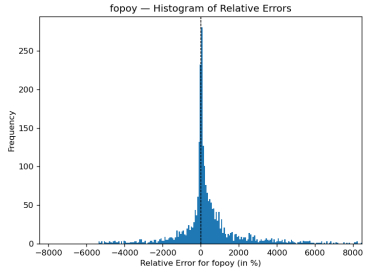

dltisy — Histogram of Relative Errors
Frequency
Relative Error for dltisy (in %)

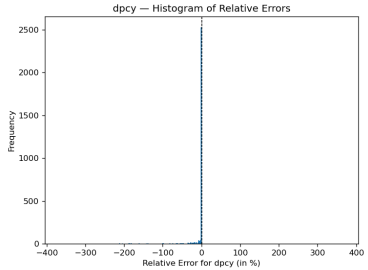

capxy — Histogram of Relative Errors
Frequency
Relative Error for capxy (in %)

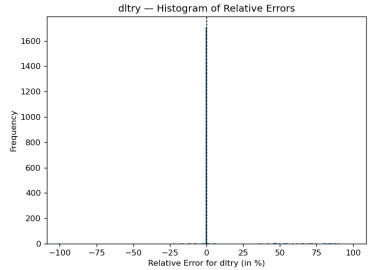

dlttq — Histogram of Relative Errors
Frequency
Relative Error for dlttq (in %)

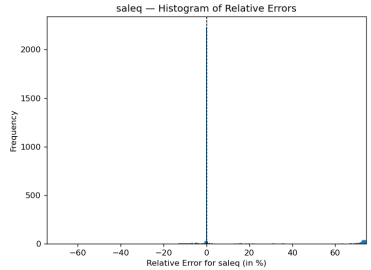

txditcq — Histogram of Relative Errors
Frequency
Relative Error for txditcq (in %)

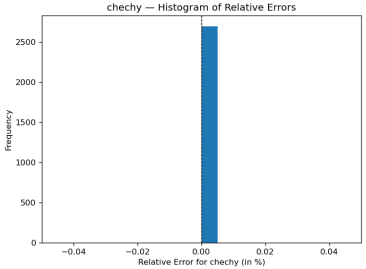

dlcq — Histogram of Relative Errors
Frequency
Relative Error for dlcq (in %)

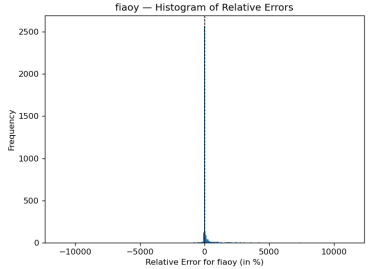

fopoy — Histogram of Relative Errors
Frequency
Relative Error for fopoy (in %)

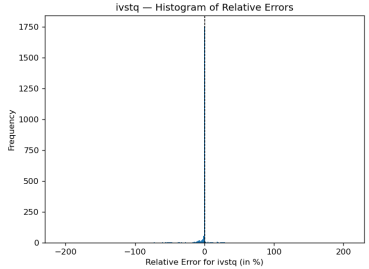

dpcy — Histogram of Relative Errors
Frequency
Relative Error for dpcy (in %)

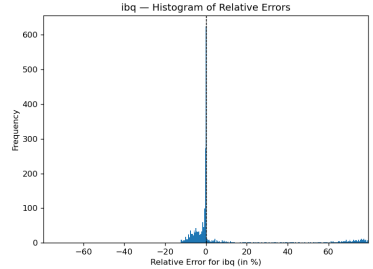

dltry — Histogram of Relative Errors
Frequency
Relative Error for dltry (in %)

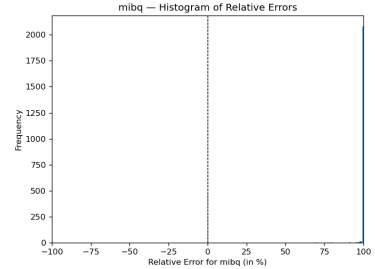

saleq — Histogram of Relative Errors
Frequency
Relative Error for saleq (in %)

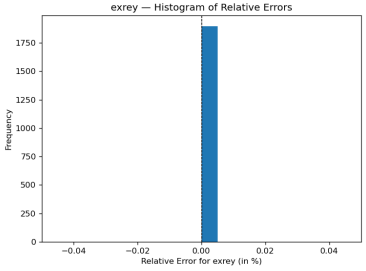

chechy — Histogram of Relative Errors
Frequency
Relative Error for chechy (in %)

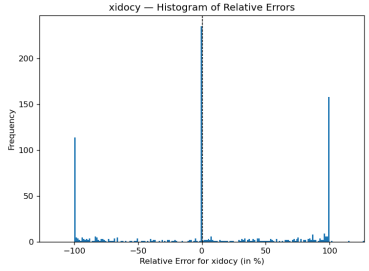

fiaoy — Histogram of Relative Errors
Frequency
Relative Error for fiaoy (in %)

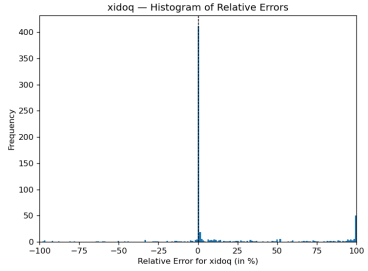

ivstq — Histogram of Relative Errors
Frequency
Relative Error for ivstq (in %)

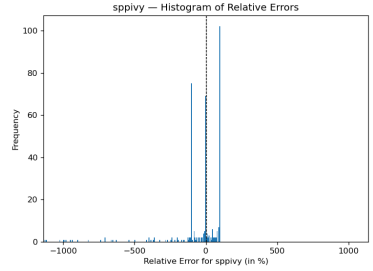

ibq — Histogram of Relative Errors
Frequency
Relative Error for ibq (in %)

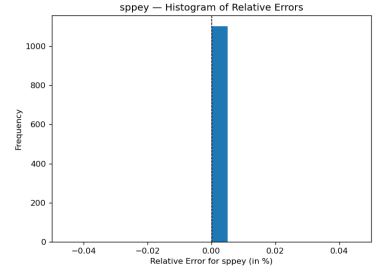

mibq — Histogram of Relative Errors
Frequency
Relative Error for mibq (in %)

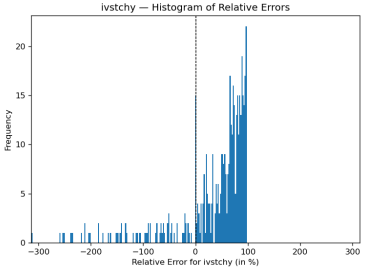

exrey — Histogram of Relative Errors
Frequency
Relative Error for exrey (in %)

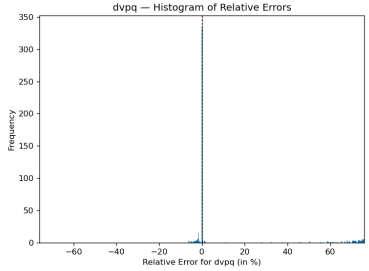

xidocy — Histogram of Relative Errors
Frequency
Relative Error for xidocy (in %)

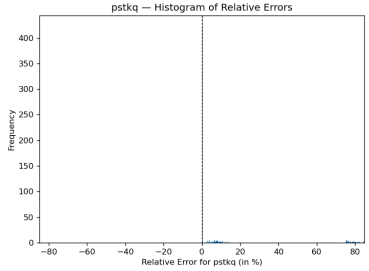

xidoq — Histogram of Relative Errors
Frequency
Relative Error for xidoq (in %)

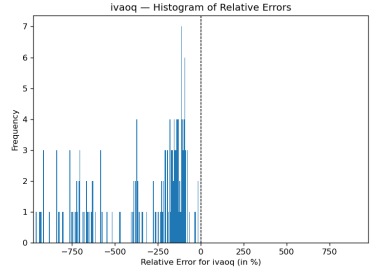

sppivy — Histogram of Relative Errors
Frequency
Relative Error for sppivy (in %)

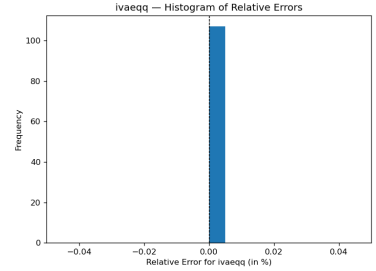

sppey — Histogram of Relative Errors
Frequency
Relative Error for sppey (in %)

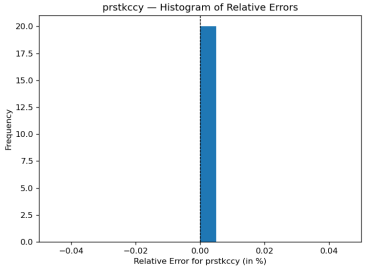

ivstchy — Histogram of Relative Errors
Frequency
Relative Error for ivstchy (in %)


dvpq — Histogram of Relative Errors
Frequency
Relative Error for dvpq (in %)


pstkq — Histogram of Relative Errors
Frequency
Relative Error for pstkq (in %)


ivaoq — Histogram of Relative Errors
Frequency
Relative Error for ivaoq (in %)


ivaeqq — Histogram of Relative Errors
Frequency
Relative Error for ivaeqq (in %)


prstkccy — Histogram of Relative Errors
Frequency
Relative Error for prstkccy (in %)

References

[1] Tallapally, P., Luehlfing, M. S., & Motha, M. (2011). Data Differences — XBRL versus Compustat. *Journal of Technology Research*. Available at: `https://www.aabri.com/manuscripts/11798.pdf`

[2] U.S. Securities and Exchange Commission (SEC). *EDGAR XBRL Guide*. Washington, D.C. Available at: `https://www.sec.gov/files/edgar/filer-information/specifications/xbrl-guide.pdf`

[3] Debreceny, R., Farewell, S., Piechocki, M., Felden, C., & Gräning, A. (2011). Flex or Break? XBRL's Uneasy Tension Between Standardization and Adaptation. *International Journal of Accounting Information Systems*, 12(2), 93–111.