

Study sheet: Module 1, alternative data in finance

This session introduces alternative data as an asset class and situates ALT/FNDATA's auction and resale pricing dataset within it. Students move from the abstract question of what counts as alternative data to a concrete, hands-on look at a real institutional dataset: 10M+ transaction records spanning watches, jewelry, gems, handbags, fine art, works of art, wine and whisky, automobiles, and more, drawn from over 100 auction houses and resale marketplaces, with history reaching back to the late 1990s. By the end of the class, students will have queried live records in a no-code sandbox and connected what they found to a broader conversation about how professional investors source signal outside of traditional filings and price feeds.

What you should be able to do

1. Define alternative data and distinguish it from traditional market and fundamental data sources.
2. Describe at least three categories of alternative data used by institutional investors, with auction and resale pricing data as one example.
3. Explain what a physical luxury asset transaction record contains and why point-in-time correctness (using the actual sale date) matters for research integrity.
4. Query a live dataset in a no-code SQL environment to answer a simple pricing question.
5. Articulate one hypothesis for how alternative pricing data could complement or contradict a signal from public equity markets.

Key terms

- **Alternative data:** information used for investment or research decisions that falls outside traditional filings, price feeds, and analyst estimates.
- **Point-in-time data:** a record that reflects only information knowable as of a specific date, such as the actual `sale_date` of a lot rather than the date it was later added to a database.
- **Sandbox:** the no-code, browser-based environment at sandbox.altfndata.com where students query live data without an API key.
- **Pre-sale estimate:** the auction house's forecast of a lot's likely price range before the sale, captured in `sale_estimates_high_usd_price`.
- **Realized price:** the actual amount paid for a lot, captured in `usd_price_decimal`, recorded only when status is sold.
- **Vendor:** the auction house or marketplace that listed the lot, useful for understanding how concentrated a result set is in a single seller.
- **Coverage browser:** the sandbox tab that shows how many brands and vendors exist within a category before a student writes a single query.
- **Revealed preference:** what a buyer's actual purchase demonstrates about their willingness to pay, as distinct from a stated preference expressed in a survey.

Core sandbox query

The query the module is built around. Run it, then change the brand or category and compare.

```
SELECT designer, item_title, sale_date, usd_price_decimal
FROM all_watches_data
WHERE designer LIKE '%Rolex%'
      AND status = 'sold'
ORDER BY usd_price_decimal DESC
LIMIT 25;
```

Common mistakes to avoid

1. Assuming a designer LIKE '%Rolex%' filter catches every relevant row. Partial-name matches can pull in unrelated listings or miss lots where the designer field is entered differently than expected, so always sanity-check the row count against the coverage browser.
2. Dropping the status = 'sold' filter and treating asking prices or unsold lots as if they were prices buyers actually paid.
3. Reading one quarter, or one sale, as evidence of a trend. A single record-setting lot says something about that lot, not about the whole brand.
4. Confusing sale_estimates_high_usd_price, which is a forecast set before the sale, with usd_price_decimal, which is what was actually paid.
5. Treating a result pulled from one or two vendors as representative of the entire market for a brand, without checking how many different auction houses contributed to the row count.
6. Comparing two brands' price ranges without first checking whether one query returned far fewer rows than the other.

Full lesson, more queries, and the homework are in Module 1 at academy.altfndata.com. Questions: info@altfndata.com.