

Q U A N T S P A C E

JSE Factor Library

Complete User Guide & Tutorial

From Data to Factor Analytics in Ten Builds

Proprietary System | Built for JSE Institutional Use | Version 1.0 | 2025

# 1. What Is the QuantSpace JSE Factor Library?

The QuantSpace JSE Factor Library is a production-grade quantitative research and portfolio analytics pipeline built specifically for JSE-listed equities. It answers the question that sits at the heart of systematic investing in South Africa:

*Which factor signals are actually driving JSE equity returns right now — and how should that change how we construct and monitor our portfolios?*

Most investment teams can describe factor performance after the fact. Far fewer have a systematic, repeatable pipeline that ingests raw JSE market and fundamental data, constructs investable factor signals, validates their predictive power, and translates them into portfolio exposure analytics every month. The QuantSpace JSE Factor Library is built to solve exactly that problem.

It processes monthly South African equity data — prices, fundamentals, and macro series — through a ten-stage build pipeline to automatically compute factor signals, measure their IC, attribute portfolio factor exposures, estimate ex-ante risk, and calculate transaction costs.

## Why JSE-Specific Factor Infrastructure Matters

South African equity markets have structural characteristics that make generic global factor models a poor fit:

- Resources sector concentration means the JSE is highly sensitive to commodity cycles that drive cross-sectional return dispersion between sectors
- Rand volatility creates a persistent macro factor that interacts with every other signal — value stocks in rand-sensitive sectors behave very differently from value stocks in offshore-revenue businesses
- A small investable universe of approximately 150-200 liquid equities means factor construction choices — composites, sector-neutralisation, liquidity screens — matter more than in deep developed markets
- Fundamentals reporting lags and ASX-like dual-class structures require careful point-in-time data handling that generic pipelines do not address

## What Makes This Different

| Traditional Approach  | QuantSpace Factor Library                        |
|-----------------------|--------------------------------------------------|
| Ad hoc factor screens | Production pipeline with explicit data contracts |

| Traditional Approach         | QuantSpace Factor Library                                          |
|------------------------------|--------------------------------------------------------------------|
| Single-factor analysis       | 14 factor signals across 5 families with sector-neutral variants   |
| No IC validation             | Build 8 IC Engine — monthly IC series and ICIR per signal          |
| Manual portfolio attribution | Automated factor exposure and attribution (Build 5)                |
| No ex-ante risk              | Ledoit-Wolf factor covariance and portfolio risk (Build 9)         |
| No cost awareness            | Transaction cost engine with net-of-cost factor returns (Build 10) |
| Static snapshots             | Full 60-month rolling pipeline with parquet artifact versioning    |

## 2. The Ten Build Stages

The pipeline is organised into ten sequential stages. Each stage reads validated upstream artifacts and writes structured parquet and JSON outputs. Running `python run_build.py --stage all` executes all ten stages in dependency order.

| Stage    | Name                 | Purpose                                                                                   | Depends On    |
|----------|----------------------|-------------------------------------------------------------------------------------------|---------------|
| Build 1  | Data Ingestion       | Loads, validates, and normalises all raw CSV inputs                                       | Raw CSV files |
| Build 2  | Universe Engine      | Constructs monthly investable universe snapshots with liquidity screens                   | Build 1       |
| Build 3  | Factor Engine        | Computes 14 factor signals with cross-sectional normalisation and sector-neutral variants | Build 1, 2    |
| Build 4  | Factor Return Engine | Builds long-short factor portfolios and computes monthly factor return series             | Build 1, 2, 3 |
| Build 5  | Portfolio Exposure   | Maps holdings to factors and computes weighted-average exposures and attribution          | Build 1, 3, 4 |
| Build 6  | Portfolio Monitoring | Generates drift alerts, mandate breach checks, and monitoring state                       | Build 5       |
| Build 7  | Performance Engine   | Computes realised portfolio and benchmark returns, rolling performance, and linkage       | Build 5, 6    |
| Build 8  | IC Engine            | Computes IC series, rolling IC, and ICIR summary per factor variant                       | Build 3, 4    |
| Build 9  | Ex-Ante Risk         | Estimates factor covariance matrix and portfolio tracking error                           | Build 4, 5    |
| Build 10 | Transaction Costs    | Computes rebalance cost estimates and net-of-cost factor returns                          | Build 2, 4, 5 |

## 3. The Fourteen Factor Signals

Build 3 computes 14 factor signals across five factor families. Each family includes one or more raw signals plus a composite (equal-weighted average of z-scores) and a sector-neutral composite (demeaned within GICS sector before aggregation).

| Family   | Signals                              | Sector-Neutral Variant |
|----------|--------------------------------------|------------------------|
| Value    | book_to_price, value_composite       | value_composite_sn     |
| Momentum | momentum_12_1, momentum_composite    | momentum_composite_sn  |
| Quality  | roe, quality_composite               | quality_composite_sn   |
| Low Risk | trailing_vol_12m, low_risk_composite | low_risk_composite_sn  |
| Size     | size_composite                       | size_composite_sn      |

All signals are computed point-in-time: fundamental data is only used once it has been published (report\_date), and market data uses only the prior month-end observation. This eliminates look-ahead bias from all factor constructions.

## Factor Construction Rules

Factor computation is governed by config/factor\_rules.yaml. Key parameters:

- min\_valid\_obs: minimum cross-sectional observations required before a signal is computed for a given month (default: 5)
- winsorise: [1.0, 99.0] — raw signals are winsorised at the 1st and 99th percentile before normalisation
- momentum\_windows: [12, 6] — momentum is computed over 12-month and 6-month windows with the most recent month excluded

## 4. System Requirements and Installation

### System Requirements

| Requirement | Specification                                        |
|-------------|------------------------------------------------------|
| Python      | 3.10 or higher                                       |
| RAM         | 4 GB minimum, 8 GB recommended for full JSE universe |
| Disk        | 500 MB for demo data and artifacts                   |
| OS          | Windows (PowerShell), macOS, or Linux                |

### Installation

#### Step 1 — Create a virtual environment

```
python -m venv .venv
```

#### Step 2 — Activate the environment

```
# Windows PowerShell
.venv\Scripts\Activate.ps1
# macOS / Linux
source .venv/bin/activate
```

#### Step 3 — Install dependencies

```
pip install -r requirements.txt
pip install -r requirements_dashboard.txt
```

## 5. Data Setup

### Option A — Synthetic Demo Data (Recommended for First Run)

The project includes a synthetic data generator that produces a fully realistic but fictional JSE dataset covering 80 equities across 5 years (2020–2024). This is suitable for demonstrations, client presentations, and pipeline validation.

```
python generate_demo_data.py
```

This writes six files to `data/raw/` and prints a validation summary. All ten checks should print PASS before proceeding.

| File                                           | Description                                     | Rows   |
|------------------------------------------------|-------------------------------------------------|--------|
| <code>security_master.csv</code>               | 80 fictional JSE equities with full schema      | 80     |
| <code>market_monthly.csv</code>                | Monthly adjusted prices and returns (2020–2024) | 4,800  |
| <code>fundamentals_annual.csv</code>           | Annual financial statements per security        | 400    |
| <code>benchmark_macro_monthly.csv</code>       | Macro and benchmark return series               | 60     |
| <code>portfolio_holdings_sample.csv</code>     | 2 sample portfolios (active and passive)        | ~5,700 |
| <code>benchmark_constituents_sample.csv</code> | ALSI and SWIX constituent weights               | 9,600  |

### Option B — Real JSE Data

To use live JSE market data, place CSV files in `data/raw/` following the exact schemas required by the ingestion loaders. Column names must be lowercase snake\_case. The minimum file set for a full pipeline run is:

- `security_master.csv`
- `market_monthly.csv`
- `fundamentals_annual.csv`
- `benchmark_macro_monthly.csv`
- `portfolio_holdings_sample.csv`

#### `security_master.csv` — Required Columns

| Column                     | Type   | Notes                        |
|----------------------------|--------|------------------------------|
| <code>security_id</code>   | string | Unique identifier, e.g. S001 |
| <code>ticker</code>        | string | JSE ticker symbol            |
| <code>issuer_name</code>   | string | Company name                 |
| <code>security_type</code> | string | 'Equity' for all             |
| <code>sector</code>        | string | JSE/GICS sector              |
| <code>industry</code>      | string | Sub-industry                 |
| <code>listing_date</code>  | date   | YYYY-MM-DD format            |

| Column               | Type    | Notes                    |
|----------------------|---------|--------------------------|
| delisting_date       | date    | Blank if active          |
| is_dual_listed       | boolean | True or False            |
| primary_listing_flag | boolean | True or False            |
| free_float_pct       | float   | 0 to 1 (not 0 to 100)    |
| shares_outstanding   | float   | Total shares             |
| active_flag          | boolean | True if currently active |

### market\_monthly.csv — Required Columns

| Column           | Type   | Notes                             |
|------------------|--------|-----------------------------------|
| date             | date   | Month-end date, e.g. 2024-01-31   |
| security_id      | string | Must match security_master        |
| close_price_adj  | float  | Adjusted close price in ZAR       |
| monthly_return   | float  | Decimal total return, e.g. 0.032  |
| volume_shares    | float  | Total shares traded in the month  |
| traded_value     | float  | Total ZAR value traded            |
| market_cap_full  | float  | Full market capitalisation in ZAR |
| market_cap_float | float  | Float-adjusted market cap in ZAR  |

### fundamentals\_annual.csv — Required Columns

| Column              | Type   | Notes                                                          |
|---------------------|--------|----------------------------------------------------------------|
| period_end          | date   | Fiscal year-end date                                           |
| report_date         | date   | Date results were published — used for point-in-time alignment |
| security_id         | string | Must match security_master                                     |
| book_value_equity   | float  | Total book value of equity in ZAR                              |
| net_income          | float  | Net income after tax in ZAR                                    |
| earnings_per_share  | float  | EPS in ZAR                                                     |
| gross_profit        | float  | Gross profit in ZAR                                            |
| operating_cash_flow | float  | Operating cash flow in ZAR                                     |
| total_assets        | float  | Total assets in ZAR                                            |
| total_liabilities   | float  | Total liabilities in ZAR                                       |
| shareholders_equity | float  | Shareholders equity in ZAR                                     |
| debt_total          | float  | Total debt in ZAR                                              |

### benchmark\_macro\_monthly.csv — Required Columns

| Column                  | Type  | Notes                                  |
|-------------------------|-------|----------------------------------------|
| date                    | date  | Month-end date                         |
| jse_market_proxy_return | float | Market-cap weighted return of universe |
| usd_zar                 | float | ZAR/USD spot rate at month-end         |
| gold_return             | float | Gold monthly return                    |

| Column              | Type  | Notes                                      |
|---------------------|-------|--------------------------------------------|
| platinum_return     | float | Platinum monthly return                    |
| brent_return        | float | Brent crude monthly return                 |
| sa_10y_yield_change | float | Change in SA 10-year yield in basis points |

## 6. Running the Pipeline

### Common Commands

| Command                                                   | Action                                     |
|-----------------------------------------------------------|--------------------------------------------|
| python run_build.py --stage all                           | Run all 10 stages in dependency order      |
| python run_build.py --stage build3                        | Run a single stage                         |
| python run_build.py --stage all --log-level DEBUG         | Run with verbose debug logging             |
| python run_build.py --stage all --log-file logs/build.log | Write log output to file                   |
| python run_build.py --stage all --allow-partial-load      | Allow partial outputs for exploratory runs |

### What a Successful Run Looks Like

After running `--stage all` on the synthetic demo dataset, you should see the following key metrics in the log output:

- Build 1: security\_master 80 rows, market\_monthly 4800 rows — all validation checks PASS
- Build 2: 4789 investable rows across 60 months, avg\_investable 79.7
- Build 3: 4789 factor snapshot rows, 14 factors, 60 months
- Build 4: 1680 factor return series rows, 124308 bucket membership rows
- Build 5: 5728 holdings mapped, 1080 exposure rows, status SUCCESS
- Build 6: 826 alerts, 432 mandate breaches, 120 state rows
- Build 7: 120 realized return rows, 98.3% forward return coverage
- Build 8: 14 IC variants, 840 IC series rows, 778 rolling IC rows
- Build 9: Ledoit-Wolf covariance, 5 factors, 120 risk summary rows
- Build 10: 60 dates processed, weighted\_avg\_cost\_bps: 20.04

*Tip: The build summary JSON files in data/processed/ are the first place to inspect if any stage produces unexpected results. Each stage writes a structured summary immediately after completion.*

## 7. Production Readiness

The following audit findings were remediated prior to the Version 1.0 release. This section documents what was fixed and why it matters for production use.

| Item                                  | What Was Fixed                                                                                                                                                                                                                                           | Impact                                                                         |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Build 8 IC signal column mapping      | IC Engine now resolves signal columns from signal_col metadata in config/factor_portfolio_rules.yaml instead of synthesising column names from factor_name and signal_variant. Previously caused all IC rows to be skipped with signal_column_not_found. | CRITICAL — IC analytics were non-functional. Now fully operational.            |
| Stage input/output schema contracts   | Builds 2–10 validate required columns on all input and output DataFrames through src/pipeline/contracts.py before any analytics run. Schema drift now fails early with a clear error message.                                                            | MAJOR — Previously schema drift could propagate silently through the pipeline. |
| Build 6 dashboard status              | Portfolio Monitoring build summary now emits an explicit status field. Dashboard reads this field consistently. Previously Build 6 showed UNKNOWN in the dashboard status panel.                                                                         | MAJOR — Monitoring stage health was misrepresented in the dashboard.           |
| numpy declared in requirements.txt    | numpy is now explicitly declared as a direct dependency. Previously relied on transitive installation.                                                                                                                                                   | MINOR — Reproducibility and lockfile generation improved.                      |
| Performance rules configuration drift | Build 7 linkage configuration is now consumed by typed performance rules. Unused status-label config sections were removed from portfolio_performance_rules.yaml.                                                                                        | MINOR — Config changes now have predictable runtime effect.                    |

*All five audit blockers are resolved in Version 1.0. The pipeline is cleared for website and dashboard public demonstration with the synthetic demo dataset.*

## 8. Dashboard Walkthrough

### Launching the Dashboard

```
streamlit run app/dashboard.py
```

Opens at <http://localhost:8501>. After any pipeline re-run, press R in the browser to refresh, or stop and restart the Streamlit process.

### Data Upload Page

The dashboard includes a Data Upload page for loading raw CSV files without using the command line. Navigate to this page to upload real JSE market data or simulation files.

The upload page enforces canonical filenames — only the five expected filenames are accepted. An unexpected filename is rejected so a random CSV cannot silently overwrite a pipeline input. After uploading, run `python run_build.py --stage all` from the terminal to process the new data.

- Accepted: security\_master.csv, market\_monthly.csv, fundamentals\_annual.csv, benchmark\_macro\_monthly.csv, portfolio\_holdings\_sample.csv
- Uploaded files are saved directly into data/raw/ and immediately available to the pipeline runner

### Pipeline Status Page

The first page shows the health of all ten build stages. Each stage shows a status indicator (SUCCESS, PARTIAL, UNKNOWN, FAILED), the timestamp of the last run, and key metrics. This is the primary monitoring view and the first thing to check after any pipeline execution.

Key metrics to verify after a full run on demo data:

- Build 2: investable\_rows should be approximately 4789, observation\_months 60
- Build 4: n\_factors 14, n\_months 60, n\_return\_rows 1680
- Build 6: alert\_count and breach\_count will be non-zero — this is expected behaviour with synthetic data
- Build 9: risk\_model\_status should read 'ok', n\_factors 5

## Factor Overview Page

Shows the 14 factor signals computed by Build 3, including composite and sector-neutral variants across all five factor families. Displays monthly cross-sectional coverage, signal distribution, and factor diagnostics. Low coverage warnings appear when fewer than min\_valid\_obs securities have valid data for a signal in a given month.

## IC Analysis Page

Information Coefficient analytics from Build 8. The IC measures the rank correlation between a factor signal at formation date  $t$  and the forward return at  $t+1$ . A positive, stable IC above 0.05 indicates a signal with genuine predictive power over the observation window.

This page shows:

- IC time series per factor variant — how predictive each signal has been month by month
- Rolling 12-month IC — smoothed view of signal quality over time
- ICIR summary table — mean IC divided by IC standard deviation, a signal-to-noise ratio

*Note: With synthetic random-return data, IC values will be near zero by construction. This is expected. With real JSE historical data, IC above 0.05 and ICIR above 0.5 should emerge in the value, momentum, and quality families based on published academic evidence for emerging markets.*

## Portfolio Risk Page

Ex-ante risk summaries from Build 9. Shows:

- Portfolio tracking error relative to benchmark — annualised
- Factor volatility decomposition — contribution of each composite factor to total portfolio risk
- Ledoit-Wolf factor covariance matrix — 5x5 matrix across the composite factor families
- Risk model status — flags insufficient history if fewer than 24 months of factor returns are available

The current implementation uses the 5 composite factor families (value, momentum, quality, low\_risk, size) for covariance estimation. Atomic factors (book\_to\_price, roe, momentum\_12\_1, trailing\_vol\_12m) are tracked in exposures but excluded from the covariance model. This is noted in the risk\_model\_status field.

## Transaction Costs Page

Cost analytics from Build 10. Shows:



- Weighted average cost in basis points per rebalance date
- Total cost drag across the observation window
- Net-of-cost factor return series — factor returns after deducting estimated transaction costs

*Important: The current transaction cost model is a first-order approximation using active exposure turnover and synthetic liquidity estimates. For production use, replace with holdings-level security turnover and actual ADV and spread data from your market data provider.*

## 9. Key Output Artifacts

All outputs are written to data/processed/ (parquet and JSON) and data/interim/ (CSV mirrors). Parquet files are the authoritative outputs — CSV mirrors are provided for inspection convenience.

| File                                     | Stage    | Description                                                                                                                     |
|------------------------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------|
| universe_investable_monthly.parquet      | Build 2  | Monthly investable universe snapshots — one row per security per month                                                          |
| factor_snapshots_monthly.parquet         | Build 3  | Cross-sectional factor signals per security per month — 14 signals plus composites                                              |
| factor_return_series_monthly.parquet     | Build 4  | Long-short factor portfolio returns — 1680 rows (14 factors x 60 months x 2 schemes)                                            |
| factor_bucket_membership_monthly.parquet | Build 4  | Quintile bucket assignment per security per factor per month                                                                    |
| portfolio_factor_exposures.parquet       | Build 5  | Portfolio weighted-average factor exposures with benchmark and active exposures                                                 |
| portfolio_factor_attribution.parquet     | Build 5  | First-order factor attribution — active exposure times factor return                                                            |
| portfolio_monitoring_state.parquet       | Build 6  | Monthly monitoring state per portfolio — OK, WATCH, BREACH, or CRITICAL                                                         |
| portfolio_realized_returns.parquet       | Build 7  | Realised portfolio and benchmark returns with status flags                                                                      |
| ic_series_monthly.parquet                | Build 8  | Monthly IC per factor variant — 840 rows                                                                                        |
| ic_summary.parquet                       | Build 8  | ICIR and mean IC summary — 14 rows, one per factor variant. Includes signal_col_used and skipped_reason fields for diagnostics. |
| portfolio_risk_summary.parquet           | Build 9  | Ex-ante risk per portfolio per month — tracking error and factor contributions                                                  |
| factor_returns_net.parquet               | Build 10 | Net-of-cost factor returns after transaction cost deduction                                                                     |

## 10. Running Tests

### Full Test Suite

```
pytest tests/ -v
```

## Targeted Test Files

```
pytest tests/test_factor_pipeline.py -v
pytest tests/test_portfolio_exposure_pipeline.py -v
pytest tests/test_portfolio_performance_pipeline.py -v
pytest tests/test_ic_pipeline.py tests/test_pipeline_contracts.py -v
```

## With Coverage Report

```
pip install pytest-cov
pytest tests/ -v --cov=src --cov-report=term-missing
```

The test suite contains 45 test files covering ingestion, universe construction, factor computation, portfolio exposure, monitoring, performance, IC analytics, pipeline contracts, and integration. Notable additions in the current release include `test_ic_pipeline.py` (Build 8 stage-level tests) and `test_pipeline_contracts.py` (schema contract enforcement for Builds 2–10).

# 11. Troubleshooting

| Symptom                                         | Cause                                                                                                         | Resolution                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Build 1 fails: Required columns missing         | CSV file columns do not match the required schema                                                             | Check column names are lowercase snake_case. Run <code>generate_demo_data.py</code> and compare column lists.                                                                                                                                                                              |
| Build 1 fails: File not found                   | CSV file is missing from <code>data/raw/</code>                                                               | Ensure all five required CSV files exist in <code>data/raw/</code> before running.                                                                                                                                                                                                         |
| Build 6 shows UNKNOWN in dashboard              | Fixed in current release — Build 6 now emits an explicit status field and the dashboard reads it consistently | Re-run <code>python run_build.py --stage build6</code> . The dashboard should show SUCCESS after refresh.                                                                                                                                                                                  |
| Build 8 IC variants show as skipped             | Signal column cannot be resolved from <code>factor_portfolio_rules.yaml</code>                                | Open <code>ic_summary.parquet</code> and inspect the <code>skipped_reason</code> and <code>signal_col_used</code> columns. Verify <code>signal_col</code> entries in <code>config/factor_portfolio_rules.yaml</code> match column names in <code>factor_snapshots_monthly.parquet</code> . |
| Build 8 shows insufficient_history              | Fewer months of data than the IC rolling window requires                                                      | Use <code>--allow-partial-load</code> to proceed. IC statistics require a production-like history window. Tiny sample datasets legitimately produce this result.                                                                                                                           |
| Build 9 warns: factors not in covariance matrix | Atomic factors have no composite-level return series in Build 4 output                                        | Expected behaviour. Only the 5 composite family factors enter the covariance model. Atomic factors appear in exposures only.                                                                                                                                                               |
| Build 8 shows 0 significant factors             | Synthetic data produces near-zero IC by construction                                                          | Expected with demo data. Significant IC emerges with real JSE historical data.                                                                                                                                                                                                             |
| Dashboard not refreshing after pipeline re-run  | Streamlit is serving cached JSON summaries                                                                    | Press R in the browser or stop and restart: <code>streamlit run app/dashboard.py</code>                                                                                                                                                                                                    |
| Build 4 is slow                                 | 14 factors x 60 months x 2 weighting schemes computed in full                                                 | Expected: 15-20 seconds on demo data, 3-8 minutes on a full JSE universe. Normal behaviour.                                                                                                                                                                                                |
| OOM error on large universe                     | Full in-memory parquet operations on large datasets                                                           | Use <code>--allow-partial-load</code> for exploratory runs. Chunked processing for production-scale universes is on the roadmap.                                                                                                                                                           |

| Symptom                                | Cause                                                       | Resolution                                                                                                                                                          |
|----------------------------------------|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unexpected filename rejected on upload | Dashboard Data Upload page only accepts canonical filenames | Rename your file to match exactly: security_master.csv, market_monthly.csv, fundamentals_annual.csv, benchmark_macro_monthly.csv, or portfolio_holdings_sample.csv. |

## 12. File Structure Reference

| Path                                    | Contents                                                                          |
|-----------------------------------------|-----------------------------------------------------------------------------------|
| app/dashboard.py                        | Streamlit dashboard — five-panel analytics interface                              |
| config/settings.py                      | Path configuration and environment settings                                       |
| config/factor_rules.yaml                | Factor computation parameters — winsorisation, lookback windows, min observations |
| config/factor_portfolio_rules.yaml      | Factor portfolio construction — quantiles, weighting schemes, enabled factors     |
| config/portfolio_exposure_rules.yaml    | Exposure computation rules — signal column mapping, factor families               |
| config/portfolio_monitoring_rules.yaml  | Monitoring thresholds — drift limits, mandate breach rules, alert severity        |
| config/portfolio_performance_rules.yaml | Performance computation rules — benchmark, rolling windows, concentration penalty |
| config/universe_rules.yaml              | Universe construction filters — minimum traded value, market cap, security types  |
| config/validation_rules.yaml            | Raw data validation rules                                                         |
| data/raw/                               | Input CSV files — place your data here before running Build 1                     |
| data/interim/                           | CSV mirrors of all processed pipeline outputs                                     |
| data/processed/                         | Parquet and JSON build artifacts — authoritative pipeline outputs                 |
| data/sample_portfolios/                 | Sample portfolio holdings files for pipeline testing                              |
| src/ingestion/                          | Raw data loaders for all four input datasets                                      |
| src/universe/                           | Universe construction, membership, and diagnostics                                |
| src/factors/                            | Factor signal computation, IC engine, normalisation, and rules                    |
| src/portfolio/                          | Holdings loading and mapping                                                      |
| src/exposure/                           | Exposure computation, benchmark comparison, attribution, and contributors         |
| src/monitoring/                         | Risk alerts, mandate breach checks, drift, concentration, and escalation          |
| src/performance/                        | Realised returns, benchmark comparison, rolling performance, and linkage          |
| src/risk/                               | Ex-ante factor risk and Ledoit-Wolf covariance estimation                         |
| src/pipeline/                           | Stage orchestration, input/output contracts (contracts.py), and pipeline runners  |
| src/utils/                              | IO utilities, logging configuration                                               |
| src/models/                             | Canonical schema definitions and validation logic                                 |
| tests/                                  | 45 test files covering all major pipeline components                              |

| Path                       | Contents                                              |
|----------------------------|-------------------------------------------------------|
| generate_demo_data.py      | Synthetic JSE demo dataset generator                  |
| run_build.py               | Pipeline runner entry point — all stage orchestration |
| requirements.txt           | Core pipeline dependencies                            |
| requirements_dashboard.txt | Dashboard-specific dependencies (Streamlit, Plotly)   |

## 13. Glossary

| Term                                             | Definition                                                                                                                                                                                           |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Factor Signal                                    | A cross-sectional ranking of securities based on a measurable characteristic (e.g. book-to-price) that historically predicts future returns.                                                         |
| IC (Information Coefficient)                     | The Spearman rank correlation between a factor signal at formation date $t$ and the forward return at $t+1$ . A positive IC indicates predictive value.                                              |
| ICIR (Information Coefficient Information Ratio) | Mean IC divided by the standard deviation of IC over the observation window. A measure of signal consistency. ICIR above 0.5 is generally considered meaningful.                                     |
| Factor Return                                    | The return of a long-short portfolio that is long the top quintile and short the bottom quintile of a factor signal. Also called a factor-mimicking portfolio return.                                |
| Long-Short Return                                | The return spread between the top and bottom quintile factor portfolios.                                                                                                                             |
| Factor Exposure                                  | The weighted-average factor signal score of a portfolio, measuring how much factor risk the portfolio is taking.                                                                                     |
| Active Exposure                                  | Portfolio factor exposure minus benchmark factor exposure. Positive active exposure means the portfolio is tilted toward that factor versus the benchmark.                                           |
| Factor Attribution                               | The estimated contribution of each factor tilt to portfolio active return — calculated as active exposure multiplied by factor return.                                                               |
| Ex-Ante Risk                                     | Forward-looking portfolio risk estimated before returns are realised, using a factor covariance model.                                                                                               |
| Tracking Error                                   | The annualised standard deviation of active returns (portfolio return minus benchmark return). Measures how much the portfolio deviates from its benchmark.                                          |
| Ledoit-Wolf Shrinkage                            | A covariance matrix estimation technique that improves on the sample covariance matrix by shrinking extreme eigenvalues toward the grand mean — reduces estimation error for small samples.          |
| Sector-Neutral Signal                            | A factor signal that is demeaned within each GICS sector before cross-sectional normalisation, removing the sector allocation component of the signal.                                               |
| Point-in-Time Data                               | Market or fundamental data that reflects only what was publicly available at the observation date — prevents look-ahead bias in historical backtests.                                                |
| Look-Ahead Bias                                  | The error introduced when a model uses information that would not have been available at the time of the historical observation. The pipeline eliminates this by using report_date for fundamentals. |
| Parquet                                          | A columnar storage file format used for all pipeline artifacts. More efficient than CSV for analytical workloads and supports schema enforcement.                                                    |
| ALSI                                             | JSE All Share Index — the benchmark broad market index for South African equities.                                                                                                                   |
| SWIX                                             | Shareholder Weighted Index — an alternative JSE benchmark that weights securities by their free-float shares held by South African residents.                                                        |

| Term   | Definition                                                                                                                                                               |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADV    | Average Daily Value — a measure of liquidity, representing the average daily rand value of shares traded. Used in universe construction and transaction cost estimation. |
| Reg 28 | Regulation 28 of the Pension Funds Act — sets maximum asset class exposure limits for retirement fund portfolios in South Africa.                                        |

**QuantSpace (Pty) Ltd**

[george@quantspace.co.za](mailto:george@quantspace.co.za) | [www.quantspace.co.za](http://www.quantspace.co.za)

*This library is proprietary software developed by QuantSpace (Pty) Ltd. All factor methodologies, pipeline architecture, and analytics engines are the intellectual property of QuantSpace. Proprietary — Built for JSE Institutional Use — Version 1.0 — 2025*