SimFin Panel Builder - README

Overview

This script builds a monthly point-in-time (PIT) panel dataset for long-term stock selection research, following the methodology from **Wynne (2023) "Long-term Stock Selection using Random Forest and LSTM Models for Fundamental Analysis"**.

The output is a comprehensive dataset combining:

- Monthly stock prices and returns
- Quarterly fundamental data (income statement, balance sheet, cash flow)
- 70+ financial ratios across 7 categories
- Macroeconomic indicators from FRED
- Market capitalization classifications

Key Features

✅ **Point-in-Time Alignment** - Ensures no lookahead bias
✅ **Year-Adjusted Market Caps** - Following paper's methodology
✅ **70+ Financial Ratios** - Comprehensive fundamental coverage
✅ **Data Quality Filters** - Remove penny stocks, outliers, data errors
✅ **Winsorization** - Handle extreme outliers in ratios
✅ **Financial Exclusion** - Standard practice in asset pricing research
✅ **FRED Macro Integration** - GDP, inflation, interest rates

---

Prerequisites

1. Install Required Packages

```bash
pip install pandas numpy pandas-datareader
```

2. Download SimFin Data

You need the following CSV files from SimFin:

```
simfin_cache/
├── us-shareprices-daily.csv
├── us-income-quarterly.csv
```

```
├── us-balance-quarterly.csv
├── us-cashflow-quarterly.csv
└── us-companies.csv  (optional, for financial exclusion)
```

How to get SimFin data:

1. Create free account at https://simfin.com/
2. Go to "Bulk Data" section
3. Download "US - SharePrices - Daily" (Plus subscription required for full history)
4. Download "US - Income Statement - Quarterly"
5. Download "US - Balance Sheet - Quarterly"
6. Download "US - Cash Flow - Quarterly"
7. Download "US - Companies" (for industry classification)
8. Extract all files to a directory (e.g., `simfin_cache/`)

Quick Start

Basic Usage (Paper-Compliant Defaults)

```bash
python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel.csv \
    --add-macro
```

This will create a panel with:

- 1-month fundamental lag (PIT compliance)
- 1-month macro lag
- Financial companies excluded
- Winsorized ratios (5 MAD)
- Data quality filters applied
- Year-adjusted market cap categories

Expected Output

```
================================================================================
```

```
BUILDING SIMFIN PANEL WITH PAPER COMPLIANCE IMPROVEMENTS
================================================================================
Fundamental lag: 1 months
Macro lag: 1 months
Exclude financials: True
Winsorize ratios: True
Quality filters: True
Year-adjusted market caps: True
================================================================================

Building monthly prices...
  415318 monthly observations for 5835 tickers

Building quarterly fundamentals...
  182456 quarterly observations

Merging monthly prices with fundamentals...
Applied 1-month lag to fundamental publish dates for PIT alignment
  415318 rows after merge

Excluded 28934 rows from 487 financial companies

Applying data quality filters...
  Removed 12453 penny stock observations (price < $1)
  Removed 3421 observations with non-positive book equity
  Removed 8234 observations with market cap < $5M
  Removed 1823 observations with extreme returns (<-99% or >1000%)
Total filtered: 26931 rows (6.5%)

Applying year-adjusted market cap categories...
Median market cap by year (sample): {2015: 287450000, 2016: 312890000, ...}

Winsorizing 68 ratio columns at 5.0 MAD...

Adding FRED macro features...
Applied 1-month lag to macro features

================================================================================
SUMMARY
================================================================================
Output file: data/simfin_panel.csv
Shape: (361456, 128)
Date range: 2015-11-30 → 2024-09-30
Unique tickers: 5348
Rows with non-null 1yr_return: 287593

Market cap distribution:
```

```
Nano Cap     98234
Micro Cap    87123
Small Cap    76543
Mid Cap      54321
Large Cap    32109
Mega Cap     13126
Name: cap, dtype: int64
```

Command-Line Options

Required Arguments

Argument	Description
`--cache-dir`	Directory containing SimFin CSV files
`--out`	Output CSV file path

Point-in-Time Alignment

Argument	Default	Description
`--fundamental-lag-months`	1	Lag fundamental data by N months to ensure availability at investment date
`--macro-shift-months`	1	Lag macro features by N months to reduce lookahead risk

Why lag data? In real investment scenarios, you need to ensure data was actually published before your decision date. A 1-month lag is conservative and recommended by the paper.

Macro Features

Argument	Description
`--add-macro`	Fetch and merge FRED macroeconomic data
`--macro-cache`	Path to cache FRED data (default: `data/fred_macro_cache.csv`)
`--macro-rebuild`	Force rebuild of macro cache

Macro features added:

- `FEDFUNDS` - Federal Funds Rate

- `DGS10` - 10-Year Treasury Yield
- `GDP` - Gross Domestic Product
- `USACPIALLMINMEI` - CPI (inflation)
- `1mo_inf_rate`, `1yr_inf_rate` - Inflation rates
- `1mo_GDP`, `1yr_GDP` - GDP growth rates

Data Quality Options

Argument	Default	Description
`--exclude-financials`	True	Exclude financial companies (SIC 6000-6999)
`--no-exclude-financials`	-	Include financial companies
`--apply-quality-filters`	True	Apply standard filters (penny stocks, etc.)
`--no-quality-filters`	-	Skip quality filters
`--winsorize`	True	Winsorize extreme ratio values
`--no-winsorize`	-	Skip winsorization
`--winsorize-mad`	5.0	Number of MADs for winsorization threshold

Quality filters remove:

- Penny stocks (price < $1)
- Negative or zero book equity
- Very small market cap (< $5M)
- Extreme returns (< -99% or > 1000%)
- Negative revenue

Market Cap Classification

Argument	Default	Description
`--year-adjusted-caps`	True	Use year-adjusted boundaries (paper methodology)
`--static-caps`	-	Use fixed boundaries for all years

Market cap categories:

- **Nano Cap**: < $50M (scaled)
- **Micro Cap**: $50M - $300M (scaled)
- **Small Cap**: $300M - $2B (scaled)
- **Mid Cap**: $2B - $10B (scaled)
- **Large Cap**: $10B - $200B (scaled)
- **Mega Cap**: > $200B (scaled)

Usage Examples

1. Paper-Compliant Build (Recommended)

```bash
python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel_compliant.csv \
    --add-macro
```

Uses all paper recommendations: 1-month lags, exclusions, winsorization, year-adjusted caps.

2. Raw Build (No Filters)

```bash
python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel_raw.csv \
    --no-exclude-financials \
    --no-quality-filters \
    --no-winsorize \
    --static-caps \
    --fundamental-lag-months 0 \
    --macro-shift-months 0
```

Minimal processing - includes all data, no lags, no filters.

3. Conservative Build (Maximum PIT Safety)

```bash
```

```
python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel_conservative.csv \
    --add-macro \
    --fundamental-lag-months 2 \
    --macro-shift-months 2
```

Uses 2-month lags to be extra cautious about data availability.

4. Quick Test Build (Faster, for Development)

```
bash

python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel_test.csv \
    --no-quality-filters \
    --no-winsorize \
    --static-caps
# Skip --add-macro to avoid FRED API calls
```

Faster build for testing - skips expensive operations.

5. Include Financials (Special Research)

```
bash

python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel_with_financials.csv \
    --add-macro \
    --no-exclude-financials
```

Keep financial companies if studying them specifically.

Output Dataset Structure

Core Columns

Column	Type	Description
TICKER	str	Stock ticker symbol
gvkey	int	SimFin unique identifier

public_date	datetime	Month-end date (investment decision date)
MthPrc	float	Month-end price
Shares Outstanding	float	Shares outstanding
MthCap	float	Market capitalization
divyield	float	Dividend yield (TTM dividends / price)
1yr_return	float	**TARGET VARIABLE** - Forward 12-month return
cap	str	Market cap category (Nano/Micro/Small/Mid/Large/Mega)

Financial Statement Data (Quarterly)

Column Group	Examples
Income Statement	Revenue, Cost of Revenue, Gross Profit, Operating Income (Loss), Net Income, etc.
Balance Sheet	Total Assets, Total Equity, Total Current Assets, Cash, Inventories, etc.
Cash Flow	Net Cash from Operating Activities, Change in Fixed Assets & Intangibles, Dividends Paid
TTM Aggregates	All flow items have _ttm versions (trailing 12 months)

Financial Ratios (70+ ratios)

Valuation (11 ratios)

- bm, ptb, ps, pcf, pe_inc, pe_exi, pe_op_basic, pe_op_dil, evm, dpr, divyield

Profitability (11 ratios)

- npm, gpm, roa, roe, roce, opmbd, opmad, ptpm, cfm, efftax, GProf

Solvency (8 ratios)

- de_ratio, debt_at, debt_assets, debt_capital, capital_ratio, intcov, intcov_ratio, dltt_be

Liquidity (4 ratios)

- curr_ratio, quick_ratio, cash_ratio, cash_conversion

Efficiency (9 ratios)

- `at_turn`, `inv_turn`, `rect_turn`, `pay_turn`, `sale_invcap`, `sale_equity`, `sale_nwc`, `rd_sale`, `lt_ppent`

Financial Soundness (20+ ratios)

- `cash_lt`, `invt_act`, `rect_act`, `short_debt`, `curr_debt`, `lt_debt`, `profit_lct`, `ocf_lct`, `cash_debt`, `fcf_ocf`, `accrual`, etc.

Other

- `debt_ebitda`, `int_debt`, `int_totdebt`, `capex_ttm`, `fcf_ttm`, etc.

Macroeconomic Indicators (if `--add-macro`)

Column	Description
`DATE`	Macro observation date
`FEDFUNDS`	Federal Funds Rate (%)
`DGS10`	10-Year Treasury Yield (%)
`USACPIALLMINMEI`	CPI Index
`1mo_inf_rate`	1-month inflation rate
`1yr_inf_rate`	1-year inflation rate
`GDP`	Gross Domestic Product
`1mo_GDP`	1-month GDP growth rate
`1yr_GDP`	1-year GDP growth rate

Understanding Point-in-Time (PIT) Alignment

The Problem

Without proper PIT alignment, you risk **lookahead bias**:

Good approach: Use Q3 2020 data published October 30, 2020 ✅

How This Script Handles It

```
python

# 1. Fundamental data is merged "backward"
#    At each month-end, find the most recent published statement

# 2. Apply conservative lag (default 1 month)
#    Ensure data was available before investment date

# 3. Never use exact-match if lag > 0
#    Prevents using data published on decision date
```

Timeline Example

```
2020-10-30: Q3 2020 earnings published
2020-11-30: Month-end prices (with 1-month lag, can use Q3 data)
2020-12-31: Investment decision date (Q3 data definitely available)
2021-01-30: Q4 2020 earnings published (too late for Dec 31 decision)
```

Data Quality Best Practices

1. Missing Data Strategy

The script **does not impute** missing fundamental data. This is intentional:

- Missing data is handled in the modeling phase
- Allows flexibility in imputation methods
- Preserves information about data availability

2. Outlier Handling

Winsorization (recommended):

- Caps extreme values at median ± 5 MAD
- Preserves relative ordering
- More conservative than dropping outliers

Why outliers occur:

- Near-zero denominators (e.g., P/E with tiny earnings)
- Corporate actions (mergers, spinoffs)
- Data errors

3. Target Variable

The `1yr_return` is the **forward** 12-month return:

```
python

return = (Price_t+12 - Price_t) / Price_t
```

Important notes:

- Returns > 1000% are filtered (likely data errors)
- Returns < -99% are filtered (bankruptcy/delisting)
- Missing returns occur for most recent 12 months

Troubleshooting

Issue: "No module named 'pandas_datareader'"

Solution:

```
bash

pip install pandas-datareader
```

Issue: "FileNotFoundError: us-shareprices-daily.csv"

Solution:

- Download SimFin data (see Prerequisites)
- Verify file paths match `--cache-dir`
- Check file names use semicolon (`;`) separators

Issue: "All training data has missing values"

Causes:

1. TTM requirements too strict (need 4 consecutive quarters)
2. Early date range (insufficient history)

3. Too many quality filters

Solutions:

- Use `--no-quality-filters` for testing
- Start analysis from 2017+ (needs 2015-2016 for training)
- Check data completeness in SimFin files

Issue: "Fetching FRED data fails"

Causes:

- Network issues
- FRED API rate limits
- Outdated pandas-datareader

Solutions:

```bash
# Update pandas-datareader
pip install --upgrade pandas-datareader

# Use cached data
python script.py --cache-dir ./simfin_cache --out output.csv --add-macro
# (Will use cache if exists)

# Skip macro for testing
python script.py --cache-dir ./simfin_cache --out output.csv
# (No --add-macro flag)
```

Issue: Very few rows after quality filters

Check:

```bash
# See how many rows each filter removes
python simfin_build_panel_v4_2.py \
  --cache-dir ./simfin_cache \
  --out data/test.csv \
  --apply-quality-filters
```

Review console output:

```
Applying data quality filters...
  Removed 12453 penny stock observations (price < $1)
  Removed 3421 observations with non-positive book equity
  ...
```

Performance Tips

1. Cache FRED Data

First run:

```bash
python script.py --cache-dir ./simfin_cache --out output.csv --add-macro
# Creates data/fred_macro_cache.csv
```

Subsequent runs reuse cache automatically.

2. Test with Smaller Date Range

Modify script to filter dates:

```python
# In build_monthly_prices(), after loading:
px = px[px['Date'] >= '2020-01-01']  # Test with recent data only
```

3. Profile Memory Usage

For very large datasets:

```python
# Use chunks or reduce date range
# Monitor with: top, htop, or Activity Monitor
```

Integration with Random Forest Model

After building the panel:

```bash
```

```
# 1. Build the panel
python simfin_build_panel_v4_2.py \
    --cache-dir ./simfin_cache \
    --out data/simfin_panel.csv \
    --add-macro

# 2. Run the Random Forest backtest
python rf_stock_selection.py
```

The RF script expects:

- `data/simfin_panel.csv` location (or modify path)
- Column names matching output structure
- `1yr_return` as target variable
- `public_date` for time-based splits

Citation

If you use this script for research, please cite:

bibtex

```
@mastersthesis{wynne2023stock,
  title={Long-term Stock Selection using Random Forest and LSTM Models for Fundamental Analysis},
  author={Wynne, Morgan},
  year={2023},
  school={University of Bristol}
}
```

Support & Contributing

For issues, questions, or contributions:

1. Check the Troubleshooting section

2. Review SimFin documentation: https://simfin.com/
3. Check FRED documentation: https://fred.stlouisfed.org/

License

This script is provided as-is for academic and research purposes. Please ensure compliance with SimFin and FRED data usage terms.

Changelog

Version 4.2 (Current)

- ✅ Added year-adjusted market cap categories
- ✅ Implemented proper PIT alignment with configurable lags
- ✅ Added averaged balance sheet items for ratios
- ✅ Stricter TTM requirements (min_periods=4)
- ✅ Comprehensive data quality filters
- ✅ Winsorization for extreme outliers
- ✅ Financial company exclusion
- ✅ All paper-compliant features

Version 4.1 (Previous)

- Basic panel construction
- Static market cap categories
- Simple TTM aggregation

Happy researching! 📊📈