Part 1: TechnicalAnalysis Trading System

1. Input Description

The system takes as input:

- **Stock historical data CSV files**, one per stock
- **Directory path** to the data folder
- **Backtest configuration parameters** like date ranges and model type

Key Input Fields:

- `Date`: Trading day
- `Price`: Closing price
- `Open`: Opening price
- `Vol.`: Volume (e.g., '1.2M')

2. Module & Class Architecture

```
Main Program (main block)
|— Loop over files in data folder
    |— TechnicalAnalysis(file_path)
        |— load_data()
        |— calculate_indicators()
        |— generate_signals()
        |— train_tree_model() [Optional]
        |— predict_tree_signal() [Optional]
        |— backtest_model_predictions()
        |— backtest()
        |— evaluate_strategy_results()
```

Class: TechnicalAnalysis

- **Constructor**
  - Initializes with CSV file path
  - Loads and processes data
- **load_data()**
  - Converts date strings to datetime
  - Parses and standardizes volume strings
- **calculate_indicators()**
  - Computes SMA, EMA, MACD, RSI
  - Adds candlestick patterns (bullish/bearish engulfing)
- **generate_signals()**
  - Creates boolean Buy/Sell signals based on indicators
- **train_tree_model()**
  - Optional: Trains XGBoost or LightGBM model
  - Uses previous indicators as features
- **predict_tree_signal()**

- Predicts Buy/Sell signals using trained model on out-of-sample data

- **backtest()**
  - Simulates trading strategy using signals
  - Updates portfolio value over time
  - Computes metrics like Sharpe ratio, drawdown, volatility
- **evaluate_strategy_results()**
  - Compares strategy vs. Buy & Hold
  - Outputs results to CSV
  - Saves portfolio value graph

3. Outputs Description

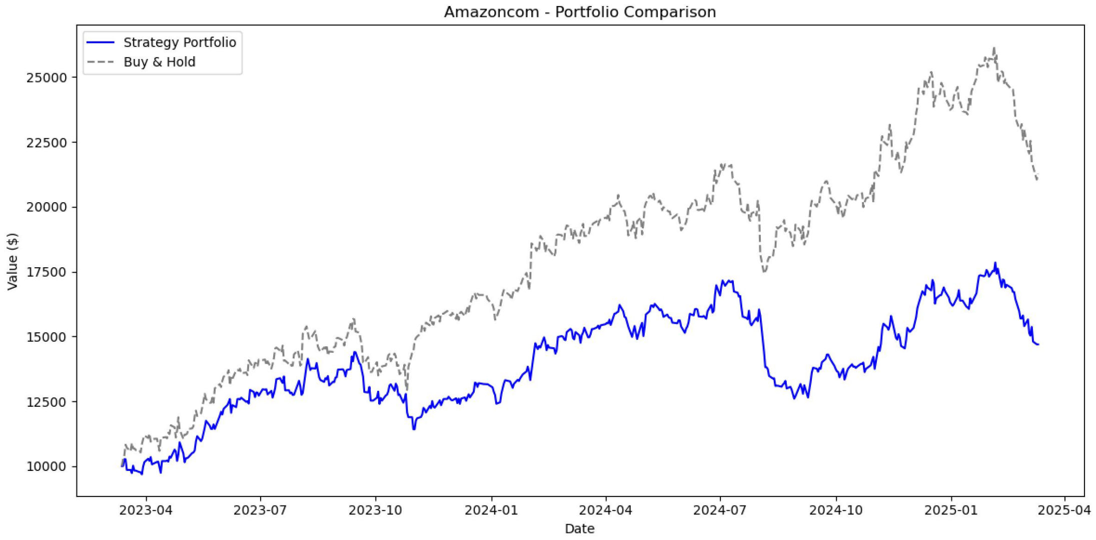
Each stock generates:

CSV file with evaluation metrics: final portfolio value, Sharpe ratio, etc.
PNG chart comparing strategy vs. Buy & Hold

Key Metrics:

- Annualized Return
- Sharpe Ratio
- Maximum Drawdown
- Volatility
- Buy & Hold benchmark results

Example Graph: `amazoncom_comparison.png`



X-axis: Time
Y-axis: Portfolio Value
Lines: Strategy vs Buy & Hold

4. Software Design Notes

Modular structure for ease of extension (e.g., adding new indicators)

OOP encapsulation using the `TechnicalAnalysis` class

Compatible with ML models (XGBoost/LightGBM) as optional enhancement

Reusable on any stock with proper CSV input

5. Suggested Improvements

Integrate live data fetching using yfinance or Alpha Vantage

Add GUI or CLI interface

Include transaction log export (e.g., trade dates, buy/sell prices)

6. Architecture Diagram

```
+----------------------------+
|        CSV Files           |
+----------------------------+
           |
           v
+----------------------------+
|   TechnicalAnalysis Class  |
|   - Load/Preprocess        |
|   - Indicators             |
|   - Signal Generation      |
|   - Optional ML Training   |
|   - Backtesting            |
|   - Metrics & Plotting     |
+----------------------------+
           |
           v
+----------------------------+
|   Evaluation Outputs       |
|   - Metrics CSV            |
|   - PNG Chart              |
+----------------------------+
```

Part 2:Multi-Factor Quantitative Model: Project Overview

1. Introduction

In this research project, we construct a **systematic multi-factor model** for stock selection and portfolio return prediction using both **fundamental** and **technical indicators.** The entire modeling pipeline spans data gathering, factor preprocessing, single-factor screening, multicollinearity analysis, factor combination, multi-factor model training, and finally **out-of-sample backtesting.**

The process is designed to emulate a real-world **quant equity research workflow**, from raw data to tradable strategy, with the goal of building a robust, interpretable, and predictive model for cross-sectional stock returns.

---

2. Project Objectives

- Identify alpha-generating factors from a large pool of fundamental and technical variables
- Clean, neutralize, and standardize factor data to ensure comparability
- Evaluate predictive power of individual factors via IC/RankIC and long-short simulations
- Detect and mitigate **collinearity** to improve model stability
- Combine selected factors into a parsimonious yet powerful multi-factor signal
- Train a cross-sectional **linear regression model** to predict returns
- Simulate a **daily long-short portfolio** based on the model and evaluate performance

---

3. Workflow Overview

Firstly, Gain the fundamental and technical factors described in CICC Report.

Then we can go to the full pipeline.

The full pipeline is structured into **seven modular steps**:

Module	Name	Purpose
1	Factor_Gainning.py	Load, reshape, and align raw factor data; extract clean stock pool
2	Data_Processing.py	Visualize, winsorize, z-score standardize, and neutralize all factors
3	Single_Factor_Testing.py	Analyze IC, Rank IC, and long-short performance of each factor
4	Colinearity_Analysis.py	Analyze factor collinearity using beta series and cross-sectional correlation
5	Factor_Combination.py	Combine redundant or thematically similar factors into composite indicators
6	Multi_Factor_Model.py	Train a linear model on INS, predict OOS returns, and evaluate IC + risk correlation
7	BackTesting.py	Simulate daily long-short portfolio and evaluate strategy performance

4. Final Deliverables

Clean and filtered INS/OOS factor datasets

IC/RankIC plots and factor selection summaries

Collinearity heatmaps and decomposition reports

Combined factor files

Multi-factor model predictions and return forecasts

Long-short backtest results including Sharpe ratio, max drawdown, and win rate

Fundamental and Technical Indicators

We should first obtain the fundamental and technical factors described in the CICC report:

https://research.cicc.com/frontend/recommend/detail?id=1758

I. Fundamental Factors

Fundamental factors provide insights into a company's financial health, operational quality, and future growth potential. In this study, we categorize fundamental indicators into five core groups:

1. Profitability Factors

These indicators reflect a company's ability to generate earnings relative to its revenue, assets, or capital.

- **Gross Profit Margin (GPM)**: Gross Profit / Revenue
- **Net Profit Margin (NPM)**: Net Income / Revenue
- **Return on Equity (ROE)**: Net Income / Shareholders' Equity
- **Return on Assets (ROA)**: Net Income / Total Assets
- **Return on Invested Capital (ROIC)**: Net Income / Invested Capital
- **Operating Profit Margin (OPM)**, **Gross Profit to Assets (GPOA)**, **Cash Flow to Assets (CFOA)**

2. Growth Factors

Growth metrics capture the rate of expansion in a company's operations or earnings over time.

- **Revenue Growth (QoQ)**: Quarterly change in total revenue
- **Operating Profit Growth (QoQ)**: Change in operating income
- **ROA Change (QoQ)**, **ROE Change (QoQ)**

3. Accrual Quality Factors

These factors measure the reliability and sustainability of reported earnings.

- **Accrual Ratio (APR)**: (Operating Income - Operating Cash Flow) / Operating Income
- **APR Difference (APRD)**: Change in APR from previous quarter
- **Cash Sales Ratio (CSR)**: Operating Cash Flow / Revenue

- **CSR Difference (CSRD)**: Change in CSR from previous quarter

4. Safety Factors

Safety indicators evaluate a company's financial stability, leverage, and liquidity.

- **Current Ratio (CR)**: Current Assets / Current Liabilities
- **Quick Ratio (QR)**: (Current Assets - Inventory) / Current Liabilities
- **Debt-to-Equity Ratio (DTE)**: Total Debt / Shareholders' Equity
- **Cash to Total Assets (CCR)**
- **Interest Coverage Ratio (ICR)**
- **Equity to Assets**, **Liabilities to Assets**, etc.

5. Operating Efficiency Factors

These metrics assess how efficiently a company utilizes its assets and capital in generating revenue.

- **Total Asset Turnover (TAT)**: Revenue / Total Assets
- **Fixed Asset Turnover (FAT)**: Revenue / Fixed Assets
- **Working Capital Turnover (WCT)**: Revenue / Working Capital
- **OCFA**: Residual from regression of operation cost on fixed assets, representing capacity utilization improvement

---

II. Technical Factors

Technical indicators are derived from price and volume data and are widely used in momentum and trend-based strategies.

1. Trend Indicators

Used to identify the direction and strength of price trends.

- **Moving Average (MA)**: Simple average over a fixed window (e.g., 20-day MA)
- **Exponential Moving Average (EMA)**: Gives more weight to recent prices
- **MACD**: Measures trend direction and momentum shifts
- **ADX (from DMI)**: Quantifies trend strength

- **SuperTrend**: Combines trend-following and volatility principles

2. Momentum Indicators

These indicators measure the speed or strength of price movements.

- **Relative Strength Index (RSI)**
- **KDJ (Stochastic Oscillator)**
- **Williams %R**
- **Momentum (MOM)**
- **Rate of Change (ROC)**

3. Volume Indicators

Volume-based metrics evaluate trading activity and confirm trends.

- **On-Balance Volume (OBV)**
- **Volume Moving Averagew**
- **Volume Ratio (VR)**
- **Money Flow Index (MFI)**

4. Volatility Indicators

These measure the degree of variation in a stock's price over time.

- **Average True Range (ATR)**
- **Bollinger Bands (BOLL)**
- **Price Standard Deviation (STD_N)**

Conclusion

The combination of fundamental and technical factors provides a holistic view of a company’s valuation, risk, growth potential, and market sentiment. By integrating these signals, we can construct more robust and data-driven investment strategies tailored for both value and momentum-oriented approaches.

Multi-Factor Modeling

Module	Name	Purpose
1	Factor_Gainning.py	Load, reshape, and align raw factor data; extract clean stock pool
2	Data_Processing.py	Visualize, winsorize, z-score standardize, and neutralize all factors
3	Single_Factor_Testing.py	Analyze IC, Rank IC, and long-short performance of each factor
4	Colinearity_Analysis.py	Analyze factor collinearity using beta series and cross-sectional correlation
5	Factor_Combination.py	Combine redundant or thematically similar factors into composite indicators
6	Multi_Factor_Model.py	Train a linear model on INS, predict OOS returns, and evaluate IC + risk correlation
7	BackTesting.py	Simulate daily long-short portfolio and evaluate strategy performance

Module 1: Factor Construction and Preprocessing

In the first stage of our multi-factor model pipeline, we focused on aggregating, cleaning, and aligning both fundamental and technical factors across a consistent stock universe and time frame. This process ensures a high-quality input matrix for downstream model training and validation.

1.1 Factor Acquisition and Merging

Data Source: The raw factor files were collected from two directories:

`Fundamental_Factors/` – e.g., ROIC, GPOA, accrual-based indicators

`Technical_Factors/` – e.g., RSI, ADX, moving averages

Parsing & Structuring: Each CSV file was reshaped into a multi-index DataFrame where columns were organized by (stock ticker, factor name). The factors were then concatenated along the column axis.

1.2 Stock Universe Filtering

Objective: Ensure all factors are calculated for a common set of stocks.

Method: We identified the intersection of non-null tickers across all factor files, yielding a consistent universe of tradable stocks.

Result: The number of overlapping stocks across all factors was reported and used to filter the merged factor set.

1.3 Time Period Selection

The factor data was filtered to include only trading dates from April 1st, 2024 onward.

Known U.S. market holidays (e.g., Independence Day, Thanksgiving) were excluded to ensure consistency with return data.

1.4 Dataset Split: In-Sample (INS) and Out-of-Sample (OOS)

INS: The first 131 trading days (approx. April to early October 2024).

OOS: The following 59 trading days (approx. mid-October to December 2024).

Each factor was saved separately into the respective `INS/` and `OOS/` folders for model training and backtesting purposes.

1.5 Return Calculation

Price Source: Adjusted close prices were downloaded from Yahoo Finance via the `yfinance` API for all stocks in the final universe.

Return Type: Daily log returns were computed and filtered to match the same post-April 1st window.

Export: Return matrices for both INS and OOS periods were saved as `returns.csv` in their respective directories.

1.6 Residual Volatility Estimation

Market Benchmark: Daily returns of the S&P 500 index (`^GSPC`) were downloaded.

Estimation Method: For each stock, a linear regression of stock returns on market returns was conducted. The standard deviation of residuals was used as a proxy for idiosyncratic risk (residual volatility or `resivol`).

Output: Daily constant `resivol` values were broadcast over time and saved as `resivol.csv` for both INS and OOS datasets.

Module 2: Factor Neutralization and Standardization

In this module, we focused on cleaning the raw factors by neutralizing them against risk exposures, analyzing their distributions, and applying statistical preprocessing steps such as winsorization and standardization. These steps are critical for ensuring that the input data is robust, comparable, and suitable for cross-sectional regression modeling.

2.1 Descriptive Analysis of Factor Distributions

Visualization: For each raw factor, we plotted its cross-sectional distribution using histogram and kernel density plots.

Statistical Summary: We computed key descriptive statistics (mean, standard deviation, percentiles, etc.) to inspect skewness, outliers, and overall factor quality.

2.2 Neutralization Using Residual Volatility

Rationale: To reduce the influence of common risk exposures, each factor was neutralized against *residual volatility (resivol)* using cross-sectional linear regression on each trading day.

Method:

For each date, a linear model was fitted:

$$Factor_i = \beta \bullet Resivol_i + \varepsilon_i$$

The residuals ε_i were retained as the *neutralized factor values*.

Scope: The neutralization was performed separately for all in-sample (INS) and out-of-sample (OOS) factor files.

2.3 Winsorization and Z-Score Standardization

Winsorization: We capped each factor's cross-sectional distribution at the 2.5th and 97.5th percentiles to mitigate the influence of extreme outliers.

Z-Scoring: The winsorized values were standardized to zero mean and unit variance for each date to ensure comparability across factors.

Batch Processing: These operations were applied on a per-date and per-factor basis for all neutralized factors.

2.4 Distribution Analysis of Cleaned Factors

After cleaning, each factor was again visualized to confirm the effect of winsorization and standardization. Most distributions now centered around zero and appeared well-behaved.

2.5 Output

Export: The final cleaned factors were saved as individual CSV files in the format `factor_name_clean.csv` for both INS and OOS sets.

File Structure: Each file represents a matrix with dates as rows and tickers as columns, ready for downstream modeling.

Module 3: Single-Factor Performance Testing

This section focuses on evaluating the individual predictive power of each factor through Information Coefficient (IC) analysis and long-short portfolio simulation. These tests help identify high-quality factors that demonstrate stable return-predictive relationships and consistent alpha generation.

3.1 Data Alignment

Each factor file (ending in `_clean.csv`) was aligned with the corresponding forward return data (`returns.csv`), ensuring consistency in dates and stock universe.

A one-day lag was applied to the factor values to simulate realistic trading scenarios (i.e., using yesterday’s factor values to form today’s positions).

3.2 Information Coefficient (IC) Analysis

IC Definition: The cross-sectional Pearson correlation between factor values and next-day returns.

Rank IC: The Spearman correlation (rank-based) was also calculated to assess monotonic but non-linear relationships.

Metrics Computed:

Mean and standard deviation of IC over time.

Information Ratio (IR): Defined as mean IC divided by standard deviation.

These IC measures provide insight into how consistently the factor ranks stocks relative to their future performance.

3.3 Long-Short Return Simulation

Portfolio Construction: On each date, stocks were sorted by factor value and divided into *n_groups = 5* quantile groups.

Strategy Logic:

Go long on the top 20% (highest factor values).

Go short on the bottom 20% (lowest factor values).

Daily return is computed as the mean of long positions minus the mean of short positions.

Performance Metrics:

Annualized Return

Volatility

Sharpe Ratio

Maximum Drawdown

Win Rate

Final Cumulative Return (PNL)

3.4 Visualization and Output

For each factor, a 4-panel performance report was saved:

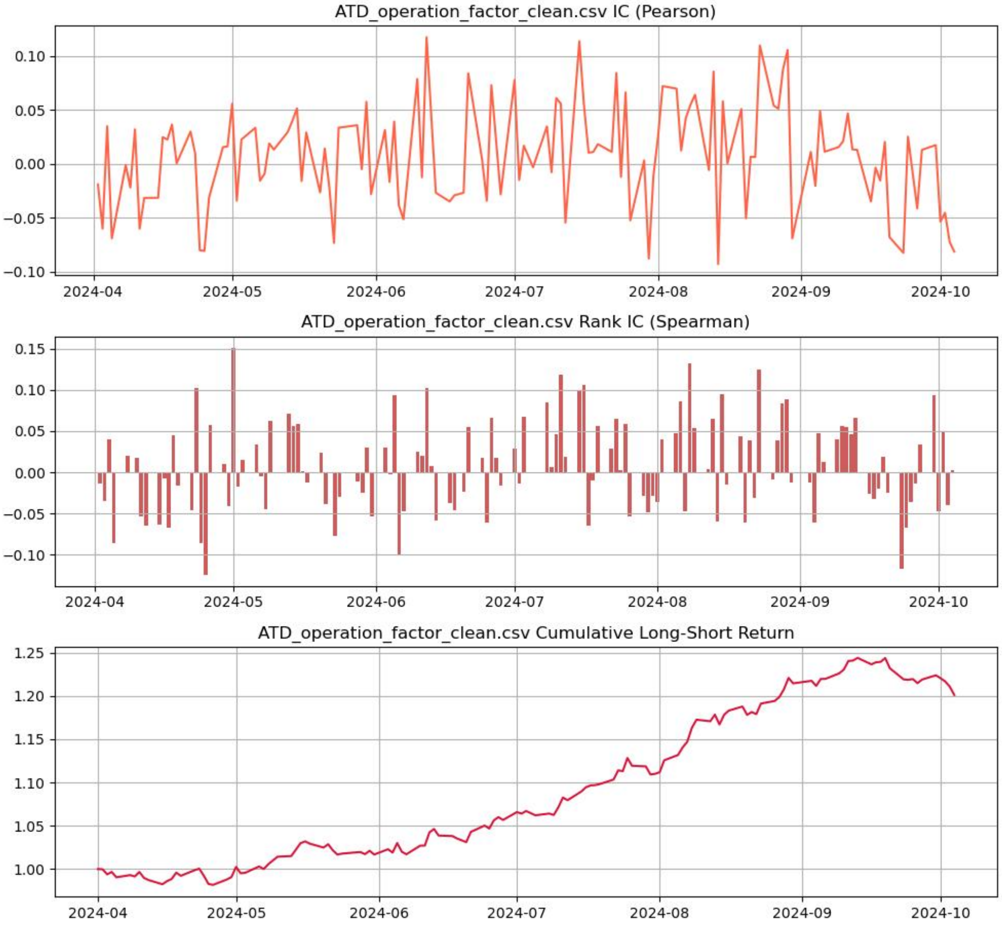
Line chart of daily IC values.

Bar chart of daily Rank IC values.

Cumulative long-short return over time.

Text summary of performance statistics.

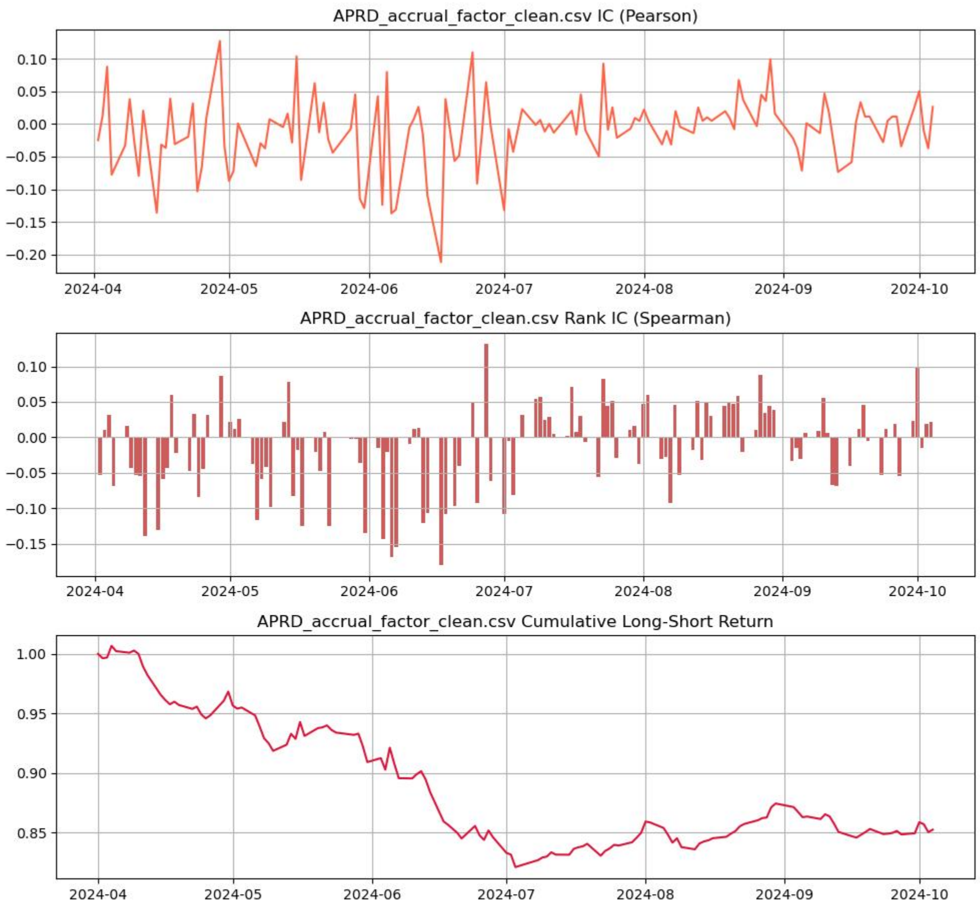
All performance plots were exported as PNGs to `Single_Factor_Testing_Report/`, and summary statistics were aggregated into a single CSV file: `IC_summary.csv`.


ATD_operation_factor_clean.csv IC (Pearson)

ATD_operation_factor_clean.csv Rank IC (Spearman)

ATD_operation_factor_clean.csv Cumulative Long-Short Return

```
Annual Return: 0.43259
Volatility: 0.09357
Sharpe Ratio: 3.84475
Max Drawdown: -0.03444
Win Rate: 0.60769
PNL Final: 1.20107
```



```
Annual Return: -0.26138
Volatility: 0.11211
Sharpe Ratio: -2.70080
Max Drawdown: -0.18436
Win Rate: 0.49231
PNL Final: 0.85255
```

Using the above figures, We filtered **15** candidate factors from an initial pool of **50.**

3.5 Insights and Factor Selection Basis

This module serves as the **screening stage** of the factor selection pipeline. Only factors with:

High and stable IC or Rank IC,

Robust long-short returns,

Acceptable drawdowns and volatility,

...will be passed on to the composite factor modeling phase.

Profitability Factors

ROIC_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.0095	0.07061	0.13453	0.0176	0.08431	0.00101	0.00865

CFOA_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00657	0.07203	0.09123	0.01664	0.07151	0.00105	0.00819

GPOA_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00808	0.05186	0.15571	0.01823	0.06804	0.00113	0.00649

Growth Factors

OP_QoQ_GrowthFactors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.01618	0.05385	0.30048	0.02796	0.07919	0.00202	0.00773

Revenue_QoQ_GrowthFactors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.01808	0.06645	0.27207	0.02781	0.05687	0.00288	0.00792

ROA_QoQ_GrowthFactors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.01437	0.05726	0.25103	0.01624	0.05035	0.0023	0.00705

Accrual Quality Factors

APR_accrual_factor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00264	0.03322	0.0795	-0.00096	0.04407	0.00053	0.00462

Safety Factors

CR_safety_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00571	0.05908	0.09665	0.00445	0.08261	0.00133	0.00765

CUR_safety_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00609	0.05708	0.10661	0.00235	0.09107	0.00118	0.00782

QR_safety_factors_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00609	0.05708	0.10661	0.00235	0.09107	0.00118	0.00782

Operating Efficiency Factors

AT_operation_factor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00344	0.04983	0.0691	0.01518	0.07275	0.00076	0.00657

ATD_operation_factor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00752	0.04683	0.16054	0.0081	0.05524	0.00143	0.00589

INVT_operation_factor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00698	0.05422	0.12873	0.03437	0.10921	0.00269	0.01162

RATD_operation_factor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.01129	0.06142	0.18384	0.01129	0.09166	0.00148	0.00906

Trend Factors

ADX_TrendFactor_clean.csv_analysis

IC_mean	IC_std	IR	RankIC_mean	RankIC_std	LS_mean	LS_std
0.00301	0.05266	0.05715	0.00468	0.05923	0.00061	0.00622

With the above analysis, we removed **3** factors with relatively low autocorrelation within each kind of factor.

Module 4: Collinearity Analysis of Selected Factors

This module examines the linear dependencies and redundancy among the selected factors. Factor collinearity can reduce model interpretability, lead to overfitting, and distort regression-based weighting methods. Thus, this step ensures the final factor set remains sufficiently diversified.

4.1 Selected Factor Set

A total of **12 factors** were selected based on prior single-factor tests and business intuition. These factors span multiple categories:

Profitability: ROIC, GPOA

Growth: OP QoQ, Revenue QoQ, ROA QoQ

Accrual Quality: APR

Safety: CR, CUR, QR

Operational Efficiency: INVT, RATD

Trend: ADX

All factor files were renamed from `_clean.csv` to `_final.csv` and copied into the working directories (`INS` and `OOS`) for consistency in processing.

4.2 Beta Time Series Construction

To assess how each factor correlates with future returns over time, we estimate **daily cross-sectional betas** by regressing daily forward returns on each factor individually. This provides:

A **time series of factor betas** per factor.

A **correlation matrix of factor betas**, measuring co-movements across time.

This analysis helps ensure the selected factors have differentiated signals rather than capturing the same risk premium.

Beta_Corr_Table

	ADX_TrendFactor	APR_accrual_factor	CR_safety_factors
ADX_TrendFactor	1	-0.051499741	-0.11640831
APR_accrual_factor	-0.051499741	1	-0.080786569
CR_safety_factors	-0.11640831	-0.080786569	1
CUR_safety_factors	-0.022237315	0.130715082	0.106327817
ROA_QoQ_GrowthFactors	-0.049315133	-0.121471189	0.156007777
ROIC_factors	0.038548788	0.079546531	0.186371255
GPOA_factors	-0.163515194	0.090480338	0.033527601
RATD_operation_factor	0.17589835	0.053909511	-0.018593195
OP_QoQ_GrowthFactors	0.218997387	0.104476722	0.133234039
Revenue_QoQ_GrowthFactors	0.042744697	0.343042964	-0.100987336
QR_safety_factors	-0.022237315	0.130715082	0.106327817

INVT_operation_factor	-0.18221526	0.247361228	0.196114063

CUR_safety_factors	ROA_QoQ_GrowthFactors	ROIC_factors
-0.022237315	-0.049315133	0.038548788
0.130715082	-0.121471189	0.079546531
0.106327817	0.156007777	0.186371255
1	0.092366735	-0.077419182
0.092366735	1	0.156055924
-0.077419182	0.156055924	1
-0.160290569	0.048346803	0.03917552
-0.092716663	-0.135504282	0.0868211
-0.026249497	0.172895311	0.182950423
0.065255767	-0.268879646	-0.130783776
1	0.092366735	-0.077419182
0.113356742	0.020596038	0.189045189

GPOA_factors	RATD_operation_factor	OP_QoQ_GrowthFactors
-0.163515194	0.17589835	0.218997387
0.090480338	0.053909511	0.104476722
0.033527601	-0.018593195	0.133234039
-0.160290569	-0.092716663	-0.026249497
0.048346803	-0.135504282	0.172895311
0.03917552	0.0868211	0.182950423
1	-0.06823758	-0.006200302
-0.06823758	1	0.149623787
-0.006200302	0.149623787	1
-0.037564678	0.340032487	0.030302897
-0.160290569	-0.092716663	-0.026249497
0.07288214	-0.128618511	0.228329294

Revenue_QoQ_GrowthFactors	QR_safety_factors	INVT_operation_factor
0.042744697	-0.022237315	-0.18221526
0.343042964	0.130715082	0.247361228
-0.100987336	0.106327817	0.196114063
0.065255767	1	0.113356742
-0.268879646	0.092366735	0.020596038
-0.130783776	-0.077419182	0.189045189
-0.037564678	-0.160290569	0.07288214
0.340032487	-0.092716663	-0.128618511
0.030302897	-0.026249497	0.228329294
1	0.065255767	0.127786549
0.065255767	1	0.113356742
0.127786549	0.113356742	1

4.3 Cross-Sectional Correlation Analysis

We computed cross-sectional correlations of factor values across all stocks on each trading day:

> **Average Correlation Matrix**: Mean correlation between all factor pairs over the in-sample period.
>
> **Time-Series Correlation Panel**: Tracks daily correlations for each factor pair to assess temporal variation in dependency structures.

Highly correlated factor pairs (e.g., correlation > 0.9) are flagged as potentially redundant and may be candidates for removal or merging.

Factor_Corr_Mean

	ADX_TrendFactor	APR_accrual_factor	CR_safety_factors
ADX_TrendFactor	1	0.001084819	-0.032763965
APR_accrual_factor	0.001084819	1	0.021782401
CR_safety_factors	-0.032763965	0.021782401	1
CUR_safety_factors	-0.025407379	0.04323466	0.94095348
ROA_QoQ_GrowthFactors	0.018580072	0.001160321	-0.017192916
ROIC_factors	0.006551625	-0.012611551	-0.024589902
GPOA_factors	0.038224231	-0.008758904	-0.163102076
RATD_operation_factor	-0.001828914	-0.050345311	0.033240799
OP_QoQ_GrowthFactors	0.045213745	-0.095884955	-0.062763791
Revenue_QoQ_GrowthFactors	0.026390504	-0.012723943	0.06086625
QR_safety_factors	-0.025407379	0.04323466	0.94095348
INVT_operation_factor	-0.011974998	0.028673522	-0.050941681

	CUR_safety_factors	ROA_QoQ_GrowthFactors	ROIC_factors
ADX_TrendFactor	-0.025407379	0.018580072	0.006551625
APR_accrual_factor	0.04323466	0.001160321	-0.012611551
CR_safety_factors	0.94095348	-0.017192916	-0.024589902
CUR_safety_factors	1	-0.029351877	0.018677583
ROA_QoQ_GrowthFactors	-0.029351877	1	0.350633832
ROIC_factors	0.018677583	0.350633832	1
GPOA_factors	-0.154719297	0.066787966	0.156553295
RATD_operation_factor	0.034687343	0.047921046	0.05058649
OP_QoQ_GrowthFactors	-0.073171716	0.299872008	0.226876923

0.052262838	0.176177619	-0.004482945
1	-0.029351877	0.018677583
-0.127436039	0.034805809	0.117468946

GPOA_factors	RATD_operation_factor	OP_QoQ_GrowthFactors
0.038224231	-0.001828914	0.045213745
-0.008758904	-0.050345311	-0.095884955
-0.163102076	0.033240799	-0.062763791
-0.154719297	0.034687343	-0.073171716
0.066787966	0.047921046	0.299872008
0.156553295	0.05058649	0.226876923
1	0.039031166	0.100021736
0.039031166	1	0.139499805
0.100021736	0.139499805	1
0.086818653	0.23715474	0.285109861
-0.154719297	0.034687343	-0.073171716
0.01720636	0.04268379	0.064842834

Revenue_QoQ_GrowthFactors	QR_safety_factors	INVT_operation_factor
0.026390504	-0.025407379	-0.011974998
-0.012723943	0.04323466	0.028673522
0.06086625	0.94095348	-0.050941681
0.052262838	1	-0.127436039
0.176177619	-0.029351877	0.034805809
-0.004482945	0.018677583	0.117468946
0.086818653	-0.154719297	0.01720636
0.23715474	0.034687343	0.04268379
0.285109861	-0.073171716	0.064842834
1	0.052262838	0.047512549
0.052262838	1	-0.127436039
0.047512549	-0.127436039	1

4.4 Output

All results were compiled and exported to an Excel file:

```
/Users/a12205/Desktop/美国实习
/Colinearity_Analysis/factor_collinearity.xlsx
```

This file contains:

- `beta_corr`: Correlation of factor betas over time.
- `factor_corr_mean`: Mean daily cross-sectional factor correlation matrix.
- `cum_corr`: Time series of correlations for all unique factor pairs.

4.5 Conclusion

This module ensures that the **final factor set is diverse** in signal exposure and minimizes multicollinearity. The retained factors are sufficiently orthogonal to support stable estimation in subsequent multi-factor modeling and portfolio construction.

Module 5: Factor Combination and Aggregation

This module aims to **aggregate semantically similar factors** into composite indicators. The purpose of combining related factors is to reduce noise and improve the robustness and explanatory power of factor signals.

5.1 Objective

Individual factors, while informative, often carry overlapping information. By grouping and averaging similar factors, we construct higher-level composite factors that are:

- **More stable over time**
- **Less sensitive to outliers or individual factor errors**
- **Better at capturing thematic economic concepts (e.g., safety, profitability)**

5.2 Merge Strategy

The merging process is guided by a predefined `merge_dict`, which groups relevant factor names. For example:

```
merge_dict = {
    "safety_factors": [
        "CR_safety_factors",
        "CUR_safety_factors",
        "QR_safety_factors"
    ]
}
```

This example groups three liquidity-related safety indicators into one unified factor called `safety_factors_combined.csv`.

5.3 Implementation Details

All input files follow the naming convention `*_final.csv` and are loaded from both **in-sample (INS)** and **out-of-sample (OOS)** folders.

Each group in `merge_dict` is **equal-weight averaged** to produce the final composite factor.

Any remaining factors not included in the merging dictionary are saved individually as `{factor_name}_combined.csv`.

5.4 Output

For each folder (`INS` and `OOS`), the output includes:

`safety_factors_combined.csv` – A composite safety indicator.

Other standalone factors like `ROIC_factors_combined.csv`, `ADX_TrendFactor_combined.csv`, etc.

INS output path: `/Users/a12205/Desktop/美国实习/INS/`
OOS output path: `/Users/a12205/Desktop/美国实习/OOS/`

These `_combined.csv` files will be used in subsequent modeling modules.

5.5 Conclusion

This step enhances signal strength by leveraging grouped factor themes. It prepares the dataset for cross-sectional modeling, ensuring that final input features reflect high-level economic narratives while minimizing redundancy.

Module 6: Multi-Factor Model – Construction and Out-of-Sample Prediction

This module develops a **cross-sectional multi-factor return prediction model** using the cleaned and combined factors derived from previous steps. The model is trained in-sample (INS) and then applied out-of-sample (OOS) to generate return forecasts.

6.1 Objective

The primary goal is to use a **linear regression framework** to capture the relationship between standardized factor exposures and short-term stock returns. The model is expected to:

- Provide **robust in-sample signal quality**, measured via IC and IR
- Generalize to the **out-of-sample period**, maintaining low correlation with known risk factors (e.g., `resivol`)

6.2 Model Pipeline Overview

- **Load Factors**
  Load all factors ending with `_combined.csv` for both INS and OOS periods. Factors are stored in stacked (panel) format by stock-date.

- **Load Returns and Risk Factor (`resivol`)**

  - `returns.csv` provides 5-day forward returns for supervised training.

  - Only `resivol.csv` is retained as the primary risk control.

- **Train Cross-Sectional Regression Model**

  - A separate linear regression is fitted **each day** in the INS period.

  - Coefficients (`betas`) are stored, and predicted returns are generated for the INS stocks.

- Each day’s regression uses only stocks with non-missing values.

Predict OOS Returns

- Use the **average of all in-sample betas** to make OOS predictions.
- For each OOS day, apply the average beta vector to factor exposures.

Normalize Predictions (Scaling)

- Normalize each day’s predictions so that:
  - Long positions sum to **+0.5**
  - Short positions sum to **–0.5**
- This ensures market neutrality and equal risk contribution.

Evaluation Metrics

- **In-sample IC (Information Coefficient)** and IR are calculated.
- **OOS correlation** with `resivol` is assessed to ensure **risk neutrality**.

Save Outputs

- `oos_prediction.csv`: Scaled OOS prediction matrix by date and ticker
- `evaluation_report.csv`: Contains IC Mean, IC Std, IR, and OOS risk correlations

---

6.3 Key Results

Metric	Value
IC Mean	0.0872
IC Std	0.0371
IC IR	2.35
OOS Corr – resivol	0.0058

In-Sample Performance:
Measured by Spearman IC and IR (IC / IC Std)

Risk Control:
Low correlation between OOS predictions and residual volatility (`resivol`) confirms effective **neutralization of risk exposure**

6.4 Use of Predictions

The normalized predictions are used in the **final backtesting module** to construct long-short portfolios and assess out-of-sample investment performance.

The top-N and bottom-N stocks by predicted return are selected daily.

Their average realized returns constitute the backtest return series.

6.5 Conclusion

This module is the core of the predictive engine. It bridges cleaned factor inputs and final portfolio actions, ensuring both **predictive accuracy and risk control**. The normalized OOS predictions serve as the foundation for evaluating strategy profitability in a realistic setting.

Module 7: Backtesting — Portfolio Simulation and Performance Evaluation

This module conducts **out-of-sample backtesting** based on predicted return scores from the multi-factor model. The backtest simulates a **daily long-short strategy**, selecting stocks with the highest and lowest predicted alpha values, and evaluates the strategy’s effectiveness using industry-standard performance metrics.

7.1 Objective

To assess the **practical profitability and risk** of the factor model’s predictions using realistic trading rules:

Long the top-N predicted stocks

Short the bottom-N predicted stocks

Equal weight within each group

7.2 Strategy Setup

Prediction Source:
The model’s normalized OOS predictions (`oos_prediction.csv`)

Return Target:
Realized forward 5-day log returns (`returns.csv`)

Portfolio Construction Rule:

- Daily, select the top-N stocks with the highest predicted returns → long
- Select the bottom-N stocks with the lowest predicted returns → short
- Equally weight each stock in its group (no alpha weighting)
- **Market neutral**: Long and short positions have the same capital allocation

Default setting:
`N = 60` (can be adjusted via the `top_n` parameter)

7.3 Backtest Logic

Data Alignment
Ensure the predicted and realized returns align on both dates and tickers.

Daily Portfolio Returns
For each day:

- Take the average return of the long group
- Take the average return of the short group
- Compute net long-short return as `long - short`

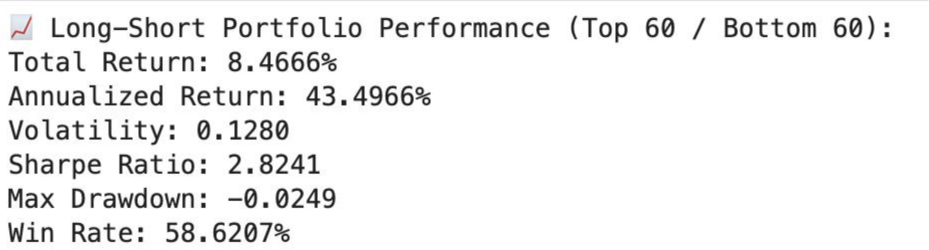
Performance Evaluation
Key metrics:

- **Total Return**
- **Annualized Return**
- **Annualized Volatility**
- **Sharpe Ratio**
- **Maximum Drawdown**
- **Win Rate** (percentage of positive daily returns)

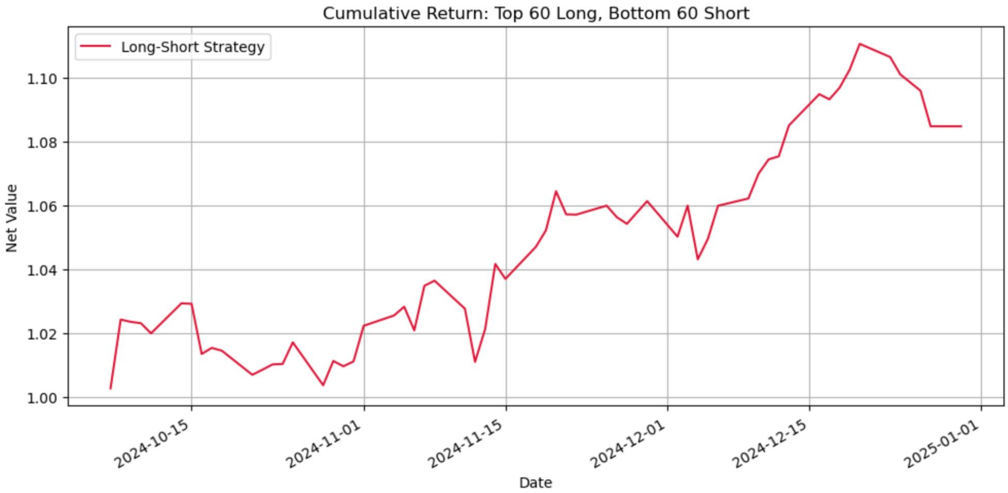
7.4 Result Outputs

Return Time Series:
`top60_long_bottom60_short_returns.csv` (daily returns of long, short, and long-short portfolios)

Summary Statistics:
`long_short_summary_top60.csv` (performance metrics)



```
📈 Long-Short Portfolio Performance (Top 60 / Bottom 60):
Total Return: 8.4666%
Annualized Return: 43.4966%
Volatility: 0.1280
Sharpe Ratio: 2.8241
Max Drawdown: -0.0249
Win Rate: 58.6207%
```



7.5 Insights

> This module provides a **quantitative validation** of the model's predictive power.
>
> The strategy is **fully out-of-sample**, using only information available prior to each trading day.
>
> It reflects a **capacity-neutral**, frictionless portfolio, suitable for comparing model performance under ideal conditions.

7.6 Conclusion

This backtesting framework completes the pipeline by transforming the model's signal into **tradable actions** and evaluating **risk-adjusted returns**. It provides a rigorous test of whether the factor model delivers meaningful and stable alpha in a realistic portfolio setting.