Università degli Studi di Milano Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea di Data Science

FINANCIAL MARKET ANALYTICS

Exam Report

Luca Poli 852027 l.poli6@campus.unimib.it

Summary

Introduction

In this Financial Market Analytics project, I will develop a Factor model (also known as a Risk model) based on the provided dataset "Euro.xls." The dataset consists of 40 factors for approximately 800 stocks.

A Factor model is a statistical model based on the assumption that systematic price changes are primarily generated by certain factors. Since the dataset contains variables based on company fundamentals, I will construct a Fundamental model (as opposed to an Economic model based on macroeconomic variables).

The report will be divided into two main chapters. In the first chapter, I will analyse the provided data in more detail and perform univariate strategies, where only one factor is considered at a time, followed by an analysis of the results. In the second chapter, I will present different types of multivariate strategies that employ multiple factors. Additionally, I will propose my solution to address the correlation issue among the univariate strategies and analyse the results.

Chapter 1: Univariate Strategies

In this chapter, I will provide a more detailed description of the given data, the methodology used for data processing and analysis. Subsequently, I will perform univariate strategies on 10 factors (to identify the best ones) and present the results.

1.1 Methodology Description and Factors Used.

1.1.1 Methodology

The dataset consists of 40 factors for 797 equities. However, some of these equities have partial data, meaning that certain factors or price values are missing for specific dates. Therefore, I conducted checks to exclude such factors from the portfolio.

After some initial data processing and cleaning (described in the Appendix), I have shifted and transformed the factors to remove the look-ahead bias and to make sure that the higher values of the factors are better. In detail, I have shifted the factors related to returns and price of 1 month and those relative to company fundamentals of 3 months. Then, in the transformation process, factors were categorized into three groups based on their desirability:

- Factors that are better when higher, which were left unchanged.
- Factors that are better when lower, for which the inverse of the factor was used.
- Factors that are better when are less than a certain value, for which the difference between the factor and the target value was utilized.

The univariate strategies are implemented as follows:

- For each date, I rank the available stocks based on a given factor (by taking the transformed best), and I take the top 30 stocks to be included in the portfolio.
- I then calculate the returns of the portfolio for each date.
- After extracting the portfolio state at each period, I use it to adjust the returns with the fees.
- I then calculate the metrics and ratios for each strategy, and I compare them to the benchmark (made of the 800 stocks equally weighted) to evaluate the performance of the strategy.

1.1.2 Factors and Metrics Used.

For the implementation of univariate strategies, I initially chose a high number of factors (combining both technical and fundamental ones). The obtained results helped me determine a smaller set of factors to be used in the multivariate strategies. The analysed factors are as follows:

- RSI_14D: Momentum oscillator measuring speed and change of price movements.
- PE_RATIO: Valuation metric indicating investors' willingness to pay for earnings.
- PX_TO_BOOK_RATIO: Compares a company's market value to its book value, indicating relative valuation.
- BEST_EPS: Highest reported earnings per share value for a stock.
- WACC_COST_EQUITY: Measure of a company's cost of capital from equity sources.
- MOV_AVG_30D: 30-day moving average smoothing out short-term fluctuations.
- EBITDA_MARGIN: Ratio of Earnings Before Interest, Taxes, Depreciation, and Amortization to total revenue, indicating profitability.
- NET_DEBT_PER_SHARE: Measure of a company's financial leverage, calculated as net debt divided by the number of shares outstanding.
- NORMALIZED_ACCRUALS_CF_METHOD: Adjustments made to reported earnings to reflect the company's true economic performance.
- VOLATILITY_90D: Measure of how much the price of a security has fluctuated over the past 90 days, indicating market risk.

One of the most important metrics used to evaluate the performance of the strategies is the information ratio. The information ratio is a measure of the return of a portfolio in excess of the returns of a benchmark, adjusted for the additional risk taken. It is calculated as follows:

$$IR = \frac{R_P - R_B}{\sigma_P}$$

Where:

- IR is the information ratio.
- Rp is the return of the portfolio.
- Rb is the return of the benchmark.
- σp is the tracking error.

I evaluated each strategy using a series of metrics and ratios:

- Information ratio: The most important ratio that helps determine the excess return (compared to the benchmark) achieved by selecting that factor.
- Returns: Portfolio returns for each period, including the following calculations:
  - Mean: Mean return for each period.
  - STD: Standard deviation, useful as a risk measure.
  - Negative STD: Standard deviation of negative returns.
  - Total: Total portfolio return over the entire period.
- Alpha: Excess return of the portfolio compared to the benchmark, with the following calculations:
  - Mean: Mean alpha for each period.
  - Total: Total alpha generated by the portfolio.
- Risk-adjusted return: Return adjusted for risk, calculated as Return / Risk.
- Sharpe Ratio: Sharpe ratio, calculated as the risk-adjusted return net of the Risk-Free rate. It expresses the overall risk.
- Beta: Portfolio beta relative to the benchmark, it expresses the systematic risk.
- Treynor Ratio: calculated as the return net of the Risk-Free rate, adjusted for the portfolio's beta. It expresses systematic risk.
- Sortino Ratio: like the Sharpe ratio but only considering downside risk (i.e., negative standard deviation).

Note: Strategies are evaluated with and without fees (0.4% for a sell operation).

1.1.3 Benchmark

The benchmark is made of the 800 stocks equally weighted. It is used to compare the performance of the strategies and to evaluate the results.

As follows, we can see the benchmark statistics and returns:

Figure 1: Benchmark logarithmic return (at each month)

Figure 2: Benchmark compound return (over the entire period)

Statistics	**Value**
Return AVG (%)	7.8443
Return STD (%)	18.1806
Downside return STD (%)	14.5467
Return Total (%)	71.9063
Return Risk Adj (%)	43.1466
Sharpe Ratio	0.0617
Sortino Ratio	0.2671

Figure 3: Benchmark statistics

1.2 Overall analysis of the results for all univariate strategies

We can see the outcomes of the univariate strategies by examining the information ratio.

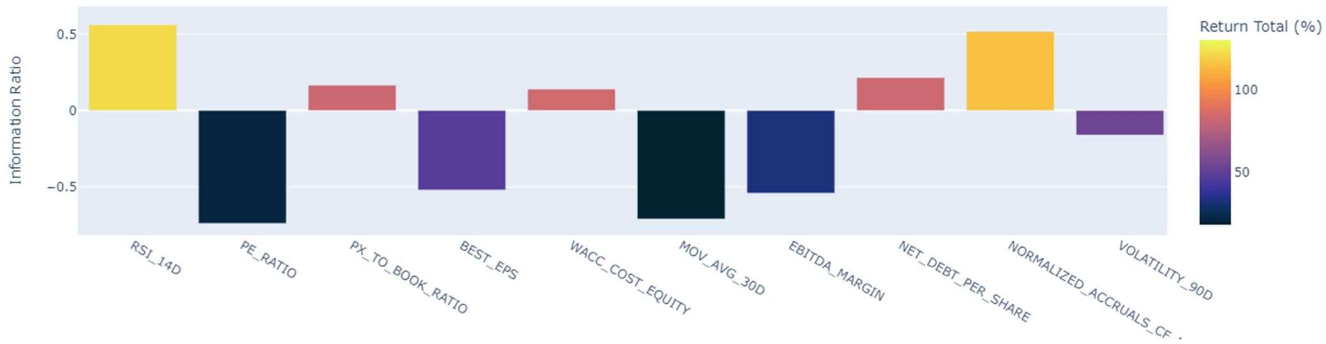

Figure 4: Information ratio for Univariate strategies (with fees)

We can see that the best strategies are RSI_14D, NORMALIZED_ACCRUALS_CF, NET_DEBT_PER_SHARE, PX_TO_BOOK_RATIO, and WACC_COST_EQUITY (for the other five is negative). These strategies have a positive information ratio, from 0.56 to 0.14, that means that they have a return higher than the benchmark for the same risk taken. But we must consider that is not high enough to be considered a solid strategy (as we expected), since those results are not statistically significant and could be due to luck.

The performance of those portfolios can be observed also by analysing the alphas.

Figure 5: Winning univariate strategies, Alpha at each month.

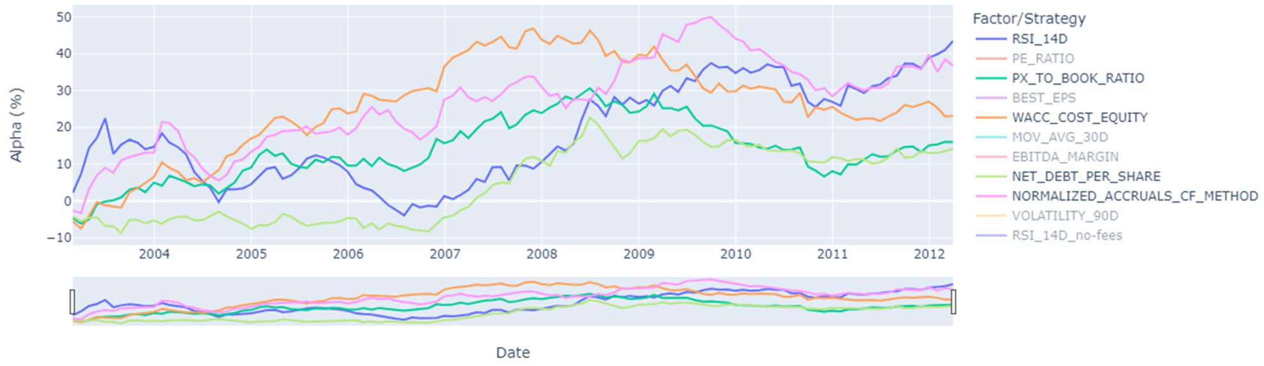

Figure 6: Winning univariate strategies, compound Alpha.

Strategies/Factors	Infor... Ratio	Return AVG (%)	Return STD (%)	Downside Return STD (%)	Return Total (%)	Alpha AVG (%)	Alpha Total (%)	Return Risk Adj (%)	Sharpe Ratio	Beta	Treynor Ratio	Sortino Ratio
RSI_14D	0.5604	13.3361	23.5328	17.9864	122.2479	4.7441	43.4878	56.6705	0.115	1.1925	0.0786	0.5213
NORMALIZED_ACCRUALS_	0.5169	12.576	23.1969	17.3362	115.2802	4.0043	36.7062	54.2142	0.1072	1.1872	0.0726	0.497
NET_DEBT_PER_SHARE	0.2153	9.0891	17.6789	14.4545	83.3169	1.5469	14.1799	51.4122	0.0838	0.9222	0.0556	0.3549
PX_TO_BOOK_RATIO	0.1657	9.0008	16.7167	13.0199	82.5076	1.743	15.9775	53.8432	0.0871	0.849	0.0594	0.3872
WACC_COST_EQUITY	0.1397	9.167	14.7014	11.3305	84.0313	2.5216	23.1151	62.3548	0.1023	0.6914	0.0753	0.4596
VOLATILITY_90D	-0.1609	5.9682	11.5833	9.4188	54.7083	0.0766	0.7026	51.5238	0.0501	0.4973	0.0404	0.2133
BEST_EPS	-0.5199	5.1905	17.8932	14.0844	47.5797	-2.4398	-22.3644	29.0083	0.0199	0.9449	0.013	0.0874
EBITDA_MARGIN	-0.5412	3.6144	15.0978	12.0803	33.1317	-3.2681	-29.9576	23.9397	-0.0066	0.7524	-0.0046	-0.0286
MOV_AVG_30D	-0.7103	2.0341	15.1495	12.6661	18.6456	-4.8234	-44.2144	13.4266	-0.0367	0.746	-0.0258	-0.152
PE_RATIO	-0.7389	2.2814	21.9859	17.0704	20.9131	-6.128	-56.1731	10.3768	-0.022	1.1455	-0.0146	-0.0983

Figure 7: Univariate strategies, statistics ordered by Information Ratio.

From the findings is evident that the winning strategies manage to defeat the benchmark, delivering a good alpha for both of technical and fundamental factor. But, as we can see from the statistics, the winning strategies have a higher standard deviation and a higher drawdown, meaning that they are riskier than the benchmark; this is confirmed by Sharpe ratio that is quite low. We can also remarkably observe that (despite a lower IR) the strategy WACC_COST_EQUITY has higher alpha of the third-best strategy, but with lower volatility; thus, providing a better risk-adjusted return.

1.3 Analysis of the 3 Best Strategies

We can delve deeper into the top strategies by examining the portfolio state during each period; in detail, for this part I will consider the two best strategies: RSI_14D (RSI), NORMALIZED_ACCRUALS_CF (NAC) and WACC_COST_EQUITY (WCE), for the reasons explained before.

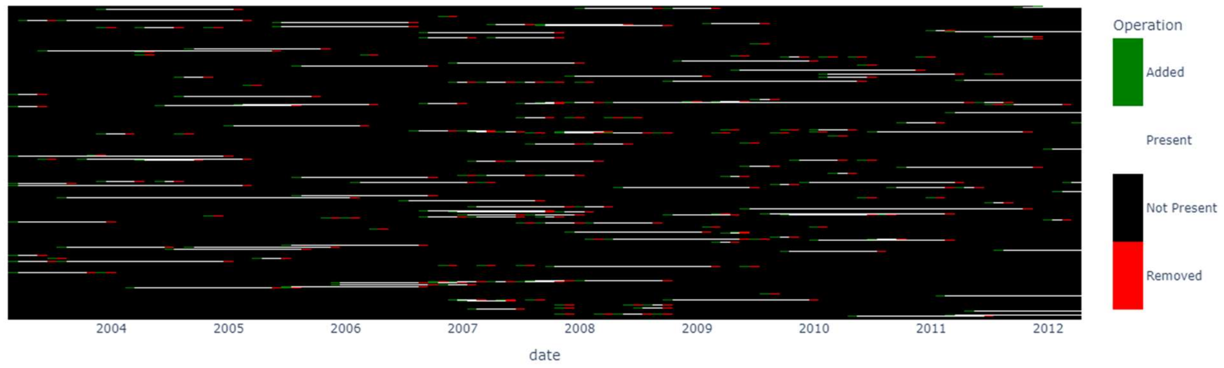

Figure 8: Portfolio state for RSI_14D

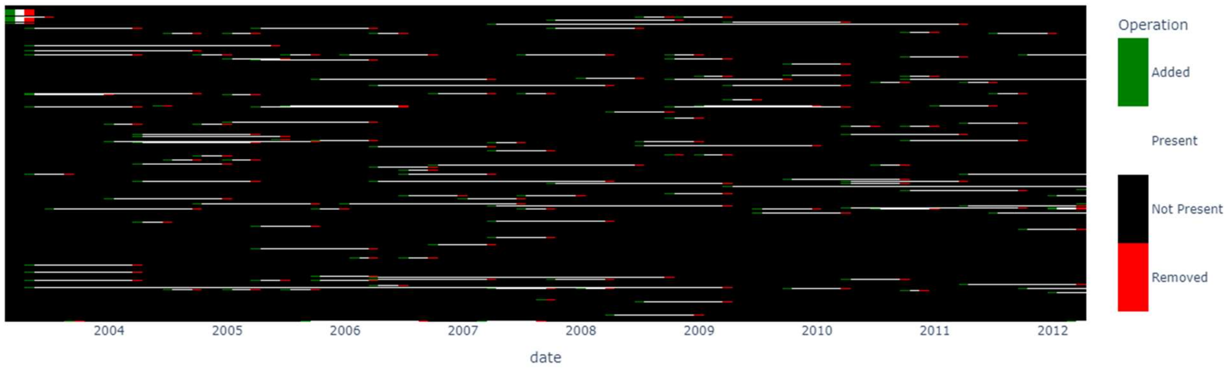

Figure 9: Portfolio state for NORMALIZED_ACCRUALS_CF

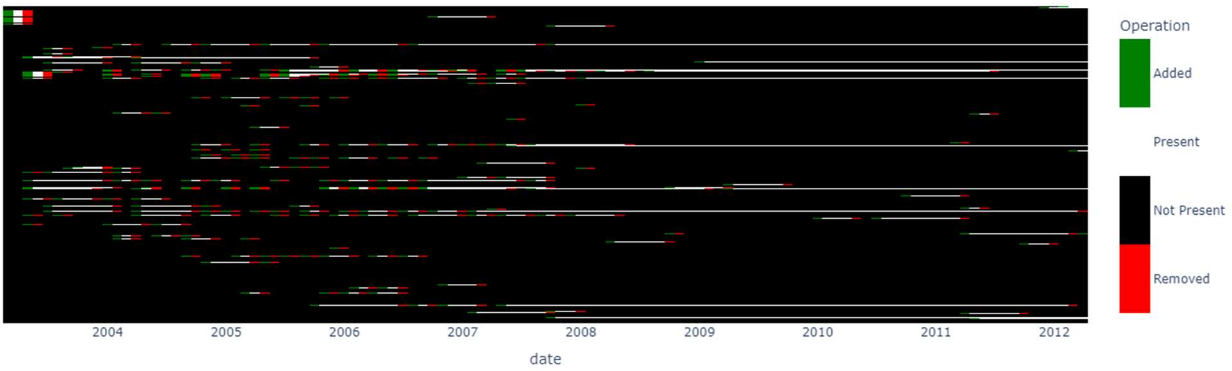

Figure 10: Portfolio state for WACC_COST_EQUITY

By looking at the portfolio state, we can see that RSI have a high turnover (typical for technical based strategy) meanwhile NAC and WCE have a lower turnover. This is a good sign, since a high turnover means higher fees and a lower return. We can also see that RSI and NAC have a stable turnover while WCE has a high turnover at the beginning and then it almost goes to zero; this is probably due to the fact that most of the missing data are at the beginning of the period, so the strategy is not able to take a strong position.

Chapter 2: Multivariate Strategies

In this chapter, I will present different types of multivariate strategies that employ multiple factors. Additionally, I will propose my solution to address the correlation issue among the univariate strategies and analyse the results.

Note: Fees are considered in the multivariate strategies.

2.1 Sequential screening

The first multivariate strategy I will present is the Sequential Screening. This strategy is based on the idea of selecting the best stocks by using a sequence of factors. The factors are used to rank the stocks, and the top stocks are selected for the portfolio. The factors are used in a specific order, and the stocks are selected based on the ranking of the previous factor.

The factors used in the sequential screening are the best 4 factors from the univariate strategies, used in order of their IR (from the worst to the best); in detail:

- At the first time we use NET_DEBT_PER_SHARE to select the top 400 stocks.
- Then we use PX_TO_BOOK_RATIO to select the top 200 stocks.
- Then we use NORMALIZED_ACCRUALS_CF to select the top 100 stocks.
- At the end we use RSI_14D to select the top 30 stocks.

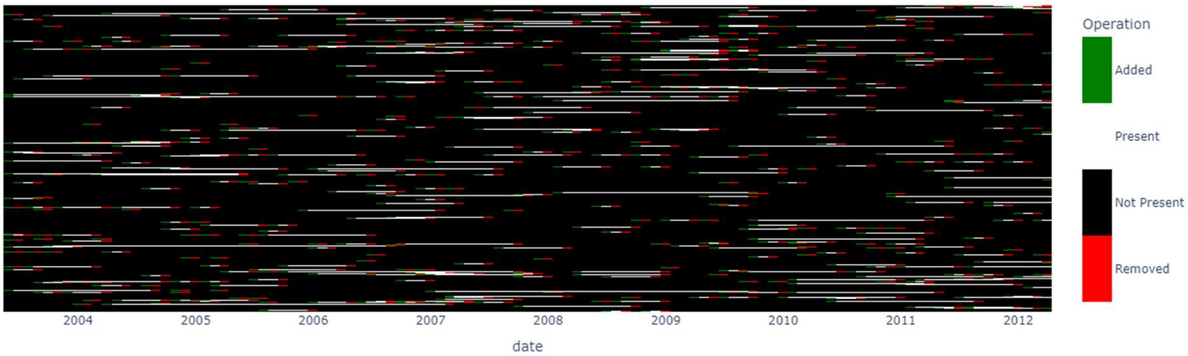

Figure 11: Portfolio state for Sequential screening

The result of the sequential screening is a portfolio with a high turnover, due to the high turnover of RSI, that provides an IR slightly higher than the best univariate strategy, with comparable metrics and slightly higher alpha and beta.

2.2 Simultaneous screening with ZScore

The second multivariate strategy I will present is the Simultaneous Screening with zscore, it is based on the idea of aggregating different factors to one, and then selecting the best stocks based on the ranking of the aggregated factor. The factors used in the simultaneous screening are the best 4 factors from the univariate strategies.

2.2.1 Simple

The first version of the Simultaneous Screening with zscore is the simple version; where I calculate the mean, at each period, of the zscore of the winning factors; then, I rank every equity based on the mean of the zscore and I select the top 30 stocks.

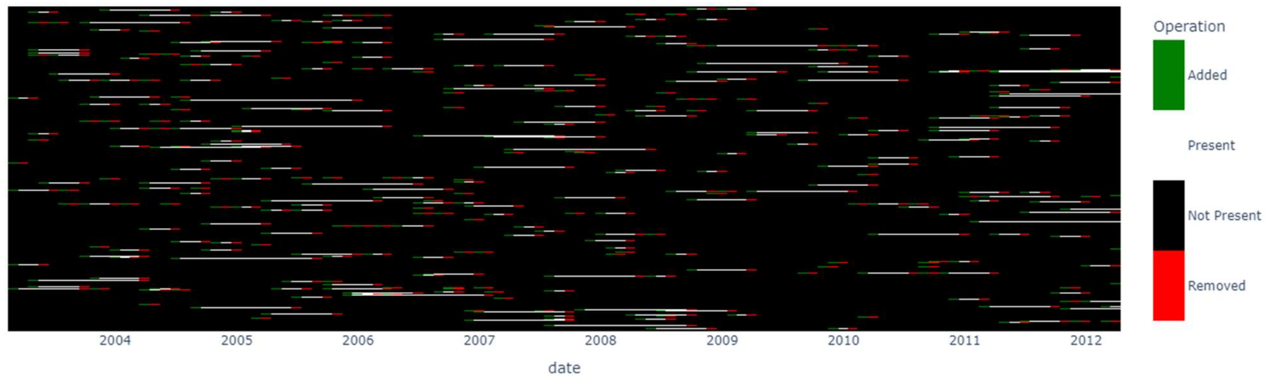

Figure 12: Portfolio state for zscore simple screening

The outcome of employing the straightforward z-score screening method is a portfolio with an even higher turnover than the sequential screening, but is better than it in every other aspect, with a higher IR, better metrics and a higher alpha and lower beta.

2.2.2 Weighted

In this version I have tried to resolve the correlation problem of the factors by using a weighted mean of the zscore, where the weights are the inverse of the correlation between the factors. In detail, the weights are calculated by computing the correlation matrix of the (zscore) factors over a rolling window of 12 months, then I compute the average for each factor; and at the end I use the inverse of the absolute average as weights.

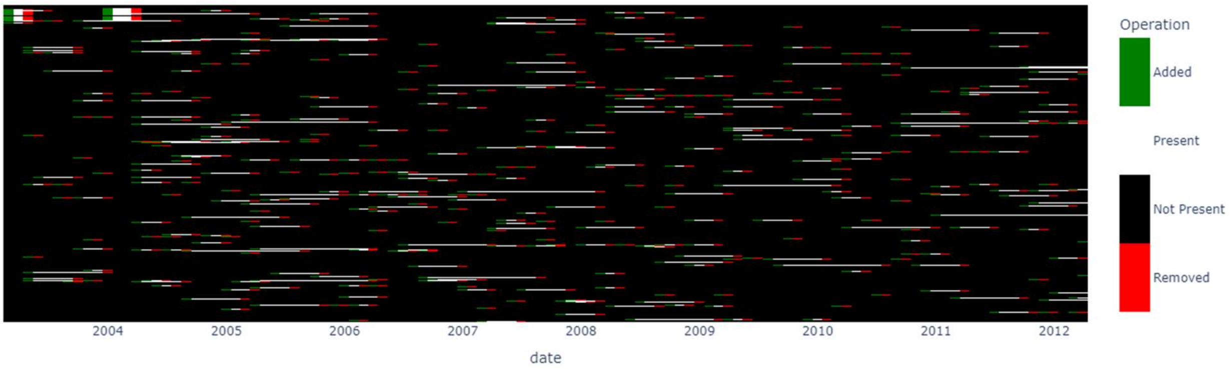

Figure 13: Portfolio state of zscore weighted screening

As we can see from the results it has a slightly lower turnover than the simple version and, given the instability of the weights at the beginning, it takes some strange positions at the start; but, after a few months, it stabilizes and provides a slightly higher alpha with a bit lower IR.

2.3 Overall analysis of the results for all multivariate strategies

We can see the outcomes of the multivariate strategies by examining the information ratio, the achieved alpha, and the statistics.

Figure 14: Information Ratio of multivariate strategies

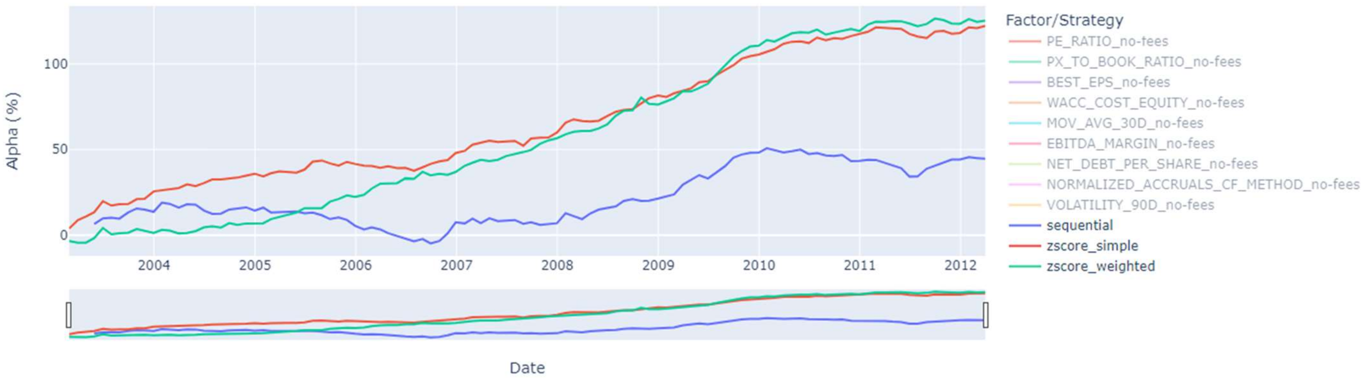

Figure 15: Compound alpha of multivariate strategies

Strategies/Factors	Infor... Ratio	Return AVG (%)	Return STD (%)	Downside Return STD (%)	Return Total (%)	Alpha AVG (%)	Alpha Total (%)	Return Risk Adj (%)	Sharpe Ratio	Beta	Treynor Ratio	Sortino Ratio
zscore_simple	2.043	20.9928	18.4625	14.1511	192.4337	13.3312	122.2024	113.705	0.2663	0.953	0.1787	1.2037
zscore_weighted	1.9701	21.7109	20.3453	15.4492	199.0168	13.6677	125.2869	106.7122	0.2519	1.0512	0.1689	1.149
sequential	0.6408	13.5252	23.6731	17.8589	120.5993	4.9991	44.5749	57.1329	0.1167	1.2121	0.0789	0.5358

Figure 16: Statistics of multivariate strategies

Observing the performance, it's evident that the z-score strategies outperform the sequential screening, which has a lower turnover, but its alpha and IR are way lower than the z-score strategies. In comparison, the simple version of the z-score strategy has a slightly higher IR than the weighted version and slightly lower alpha, but they are pretty similar in every other aspect (except for the beta which is higher for the weighted).

We can also look at the risk metrics (such as Risk adjusted return, Sharpe, Sortino and Treynor ratio) which confirm that the z-score strategies are better than the sequential screening.

Conclusion

In this project, I have developed a Factor model based on the provided dataset "Euro.xls." I have performed univariate strategies on 10 factors and presented the results. Subsequently, I have presented different types of multivariate strategies that employ multiple factors and proposed my solution to address the correlation issue among the univariate strategies. I have also analysed the results and compared the performance of the strategies to evaluate the results.

It can be concluded that the best univariate strategies are RSI_14D, NORMALIZED_ACCRUALS_CF, NET_DEBT_PER_SHARE, PX_TO_BOOK_RATIO, and WACC_COST_EQUITY. These strategies have a positive information ratio, from 0.56 to 0.14, that means that they have a return higher than the benchmark for the same risk taken. However, the results are not high enough to be considered a solid strategy.

I have used those factors to create multivariate strategies, and I have found that the best multivariate strategies are the simple and weighted versions of the Simultaneous Screening with zscore, which outperform the sequential screening. The simple and weighted zscore are quite similar in terms of risk and performance, but if we remove the first few months, the weighted version is better. The sequential screening has a lower turnover, but its performance lacks behind the z-score strategies.

In future development we could try to use some different combination of factors for multivariate strategies, instead of assuming that the best factor for the univariate strategies is also the best for multivariate ones, we could also try to address the correlation problem in different ways (such as using PCA). Another possible development, with more frequent data, could be the use of a machine learning model to learn the best combination of factors for the multivariate strategies.

Appendix: Technical Details

The code used for this report is written in Python and is divided in three main parts: dataset_extraction.ipynb responsible for the initial data processing; analysis.ipynb for the analysis of the strategies; app.py for the visualization of the results.

In the dataset_extraction.ipynb file, I have extracted the data from the provided dataset "Euro.xls" (renamed "data_initial.xlsx") and transformed it to a pandas DataFrame (multi-indexed by date and equity, and factors as columns) without losing or changing any information. This part is important, because in the analysis file I change the factors to use without any difficulty.

In the analysis.ipynb file, I have performed the univariate and multivariate strategies, and then I have evaluated the results. To be computationally efficient I have used a lot of vectorized operations (from pandas and numpy) and the numba library; however, this makes the code less understandable at first sight, so I have separated the code in different functions, and added a lot of comments to make it more readable. Every result is saved in the folder "data/output" on multiple csv files, to be used in the app.

In the app.py file, I have used the Dash and Plotly libraries to create an interactive web application to visualize the results, (it was not required, but it makes easier to analyse the results); to run the app, it is necessary to install the requisite libraries (requirements.txt) and execute the file.

There is also a fourth file, offline_plot.ipynb, that is used to generate similar plot to the ones in the app; but in a notebook environment, to remove the need of running the app to visualize the results (although the app is recommended).