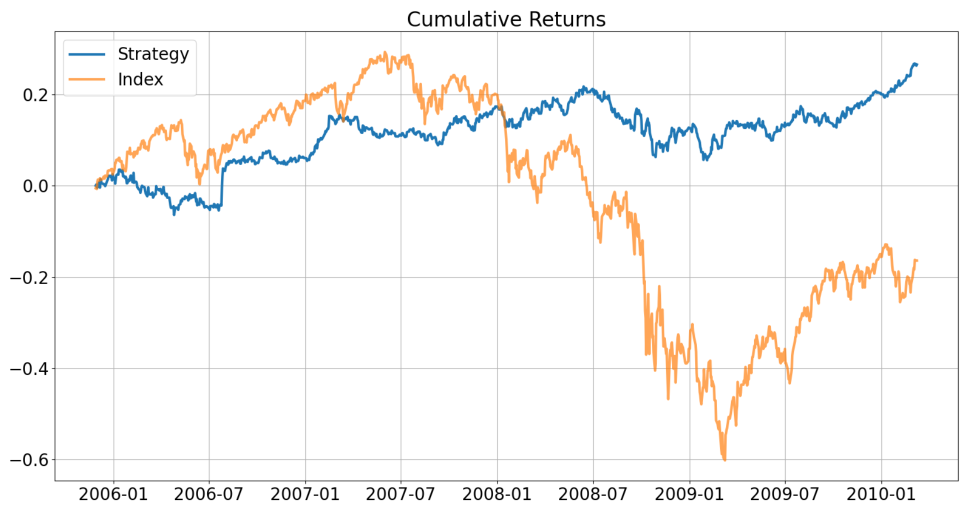
The cointegration Alpha : enhanced index tracking and long-short equity market neutral strategies - Application on the CAC40

Zakaria Magdoul

2024/2025



Contents

1 Commentary on the Paper

1.1 Introduction

In their 2002 publication, Alexander and Dimitriu introduce a sophisticated method for optimizing portfolios, employing cointegration analysis as an alternative to conventional correlation techniques. This innovative strategy presents notable opportunities for improved index tracking and the creation of market-neutral strategies within the domain of quantitative finance.

1.2 Conceptual Framework: Cointegration vs. Correlation

Traditional portfolio construction is largely based on correlation analysis of asset returns, a method that, as the authors point out, has several shortcomings. Correlation analysis necessitates stationary variables, which in turn requires the differencing of price series data. This process removes valuable information about shared trends in prices. Moreover, correlation measurements tend to be unstable, particularly during times of market volatility.

Cointegration provides distinct advantages by leveraging the complete dataset available in level financial variables. When applied to non-stationary stock prices, cointegration is present when there exists at least one stationary linear combination of these prices. This can be interpreted as mean reversion in price spreads, a characteristic particularly advantageous for trading strategies. It suggests that, regardless of the system's position, prices will maintain their long-term relationship.

The mathematical basis of cointegration can be articulated as follows: For a vector of $I(1)$ time series X_t, cointegration is confirmed if there exists a vector β such that:

$$\beta' X_t \sim I(0)$$

This implies that while individual asset prices may exhibit random walk behavior, specific linear combinations of these prices are stationary and mean-reverting. This provides a solid foundation for trading strategies.

Stability and Information Advantage

A particularly persuasive element of the cointegration method is its enhanced stability relative to correlation. The authors rightly highlight that while correlation offers a short-term measure of co-dependency, cointegration captures long-term equilibrium relationships. This stability leads to reduced portfolio rebalancing and decreased transaction costs, addressing a key practical issue in the implementation of quantitative strategies.

1.3 Trading Strategies and Empirical Results

The paper details three key applications of cointegration-based optimization: index tracking, long-short market-neutral strategies, and a combination of both. Each strategy capitalizes on the unique attributes of cointegration to meet specific investment goals.

1.3.1 Index Tracking Strategy

The cointegration-based index tracking strategy seeks to replicate benchmark returns and volatility by establishing a stable, long-term relationship between the portfolio and its target index. Unlike traditional methods that concentrate on minimizing tracking error variance, the cointegration approach ensures any deviation from the benchmark is temporary and mean-reverting.

1.3.2 Long-Short Market Neutral Strategy

The paper presents its most compelling findings in the context of the long-short market-neutral strategy. Using data from DJIA constituents between 1995 and 2001, the authors report their most successful self-financing statistical arbitrage strategies yielded approximately 10% annual returns (net of transaction and repo costs) with roughly 2% annual volatility and negligible correlation with the market.

1.4 Relevance to Modern Quantitative Finance

Despite its publication in 2002, the paper's methodology remains highly relevant to contemporary quantitative finance, primarily due to:

1. The proliferation of factor investing and smart beta strategies has increased interest in alternative portfolio construction methodologies.
2. Market-neutral strategies continue to be a cornerstone of many quantitative hedge funds.
3. The limitations of correlation-based approaches have become more widely recognized, especially following the 2008 financial crisis.

1.5 Limitations and Future Research Directions

While the paper offers significant contributions, there are several limitations and potential extensions that can be considered:

1. The empirical analysis is limited to DJIA constituents, which may affect its generalizability to other markets and time periods.
2. The authors acknowledge the need for further work on stock selection rules to complement cointegration results.
3. Optimal rebalancing rules represent another area for potential improvement.

In Conclusion, the paper by Alexander and Dimitriu makes a significant contribution to quantitative finance by highlighting the advantages of cointegration-based portfolio optimization for index tracking and market-neutral strategies. Their method provides a more stable and theoretically robust foundation compared to traditional correlation-based approaches.

2 Application on CAC 40 Daily Data

2.1 Preliminary Steps

2.1.1 Reconstracting the index

Per the paper, it is prudent to first reconstruct the index using the latest member stocks. This approach mitigates issues arising from disappearing and reappearing stocks, ensuring data consistency. This is particularly important as we employ a linear regression method, which is known to be highly sensitive to missing data.

We begin by simulating the forward evolution of stock weights over time using the following formulas:

$$\text{Market\_Cap}_{j,0} = \text{Weight}_{j,0} \cdot \text{Price}_{j,0}$$
$$\text{Market\_Cap}_{j,i+1} = \text{Weight}_{j,i} \cdot \frac{\text{Price}_{j,i+1}}{\text{Price}_{j,0}}$$
$$\text{Weight}_{j,i+1} = \frac{\text{Market\_Cap}_{j,i+1}}{\sum_{k=1}^{N} \text{Market\_Cap}_{k,i+1}}$$

As for the divisor, we used the following formula:

$$\text{Divisor} = \frac{\sum_k \text{Price}_{k,T} \cdot \text{Weight}_{k,T}}{\text{Cac40}_T}$$

```python
Weights = pd.DataFrame(columns=Prices.columns[1:], index=Prices.index) #Table containing the evolution of weights

iw = Initial_Weights.weights #Initial weights provided by Pr.Attali
Weights.iloc[0] = iw.values

Market_cap = Weights.iloc[0] * Prices.iloc[0, 1:]
for i in range(1, len(Weights)):
```

```
        Market_cap = Market_cap * Prices.iloc[i, 1:] / Prices.iloc[i-1, 1:]
        Weights.iloc[i] = Market_cap / Market_cap.sum()


Dividor = (Prices.iloc[-1, 1:] * Weights.iloc[-1]).sum() / Prices.iloc[-1, 0]
```

Resulting index

2.1.2 Is our Data I(1) ?

The researchers of the original papers stated that their reconstructed DJIA index was Integrated of order one. For that we Implement the Augmented dickey fuller Test.
Here are the p-values for our tests, a p-value less than 1%, means that time series is not stationnary, we also test the log of the indexes as it is what we actually will need for the cointegration based linear regression later.

	Original	**Differentiated**
CAC 40	26%	$5.10^{-18}\%$
Log(CAC 40)	35%	$7.10^{-14}\%$
Reconstructed CAC 40	47%	$2.10^{-18}\%$
Log(Reconstructed CAC 40)	48%	$5.10^{-18}\%$

Table 1: Results of ADF Tests

Our TimeSeries are indeed I(1).

2.2 Index Tracking

Remark : all the subsequent code snippets are a simplifed version of methods of a class we created that can be found on this **GitHub Repo**

2.2.1 Stock Selection

The first stage, of index tracking is stock selection, we must choose a number *Stock_number* (hyper-parameter) of stocks by some technic, many models can be used, but we will use the same one used in the paper, which is the price ranking of the stocks in the index at the moment of the construction (and eventually at each rebalancing).

```
def Select_stocks(self, start_date):
    stocks = set() #set of selected stocks
    temp = self.data.iloc[start_date, : ] #table containing all prices of stocks at
        the time of rebalancing
    for _ in range(self.Stock_number):
      stocks.add(temp.idxmax()) #we add the stock with biggest price present in temp
        to our set
      temp = temp.drop(temp.idxmax()) # we remove it from temp
    self.stocks = stocks
    return
```

2.2.2 Portfolio Calibration

The second stage, has to do more with cointegration, let's first give a simple definition.

Definition If several time series are individually integrated of order d (meaning they require d differences to become stationary) but a linear combination of them is integrated of a lower order, then those time series are said to be cointegrated.

This stage of index tracking is determining the portfolio holdings in each of the stocks selected in the previous stage.The stocks weights in each portfolio are estimated based on the ordinary least square (OLS) coefficients of the cointegration equation that regresses the index log price on the portfolio stocks log prices over a given calibration period prior to the portfolio's construction moment.

$$log(index_t) = c_1 + \sum_{k=1}^{n} c_{k+1} * log(P_{k,t}) + \epsilon_t$$

How does cointegration intervene ?
It is established that OLS cannot be applied to non stationary variables,unless there is a cointegration between log index and the tracking portfolio $\sum_{k=1}^{n} c_{k+1} * log(P_{k,t})$. The way we will check this condition is by checking whether our OLS residual is stationary.
After that, we normalize our coefficients, to obtain weights.

```
def Calibrate(self, start_date):
    self.Select_stocks(start_date= start_date) #on call our previously defined stock selection method
    Prix_stocks = self.data.loc[self.data.index[self.Calibration_period]: , :]
    X = np.log(self.data.loc[self.data.index[start_date]:self.data.index[start_date + self.Calibration_period],list(self.stocks)].values) # log of # Bloomberg's stock prices for the entire calibration period
    X = np.column_stack((np.ones(len(X)), X)) # we add ones to get an intercept .
    y = self.log_index.iloc[start_date:start_date+self.Calibration_period+1].values # log price of the index over the calibration period
    A, residuals, rank, s = np.linalg.lstsq(X, y, rcond=None)
    resid = y - np.dot(X, A) #residual
    #let's test to see if the residual is stationarry
    if adfuller(resid)[1] < 0.05: #we fix a confidence level of 5%
      self.weights = A[1:] / np.sum(A[1:])# test succesful, we normalize data
    return
```

After the first Calibration, we let the portfolio untouched for a period *rebalancing_period*, where its value evolves solely off the stocks price evolution.

$$\pi_{T+x} = \pi_{T-1} \sum_{k=1}^{n} \frac{W_{k,T}}{P_{k,T}} P_{k,T+x}$$

```
def price_evolution(self, start_date):
    Prix_stocks = self.data.loc[self.data.index[self.Calibration_period]: , list(self.stocks)] #stock's price
    prix_initiale = self.prices.iloc[start_date-1] #initial price

    t = start_date
    for t in range(start_date , min(start_date + self.rebalancing_period, len(self.prices)))  :
      temp = self.weights * Prix_stocks.iloc[t,:] /  Prix_stocks.iloc[start_date-1,:] #prices are updated
      self.prices.iloc[t] = prix_initiale * np.sum(temp)
    self.rang_rempli += 1
    return
```

Taking Transaction Costs (TC) Into Account: A percentage of every transaction is taken from our portfolio, we will note it pct in our code, and the way the fees are accounted for in the paper is through this formula:

$$TC_T = pct * \textstyle\sum_{k=1}^{n} |w_{k,T} - w_{k,T-10}| P_{k,T}$$

As taking the fees into account is more technical, and was integrated in all our functions, we let you check the code out directly on our repo mentioned above.

2.2.3 Results

Here under, are different results for different choices of number of stocks in portfolio, calibration period and transaction fees.

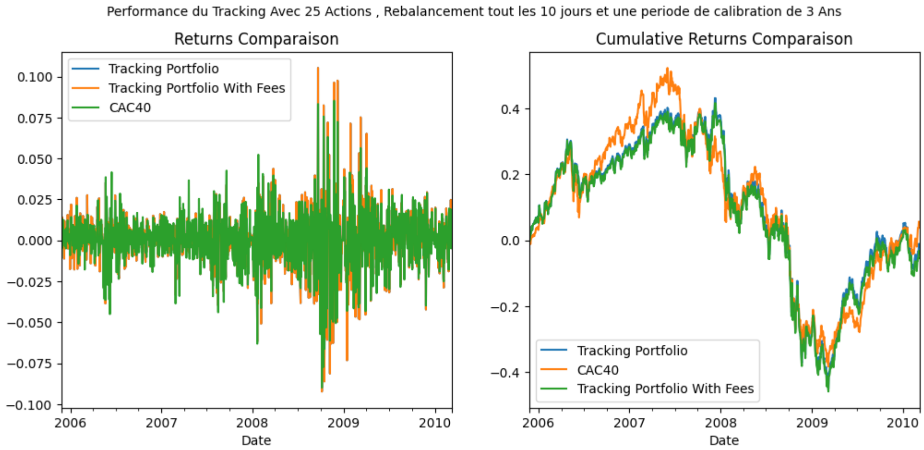


Figure 1: Performances with : $stock\_number = 22$, rebalancing period = 10, $calibration = 3, 5$ yrs, $fees = 0.1$

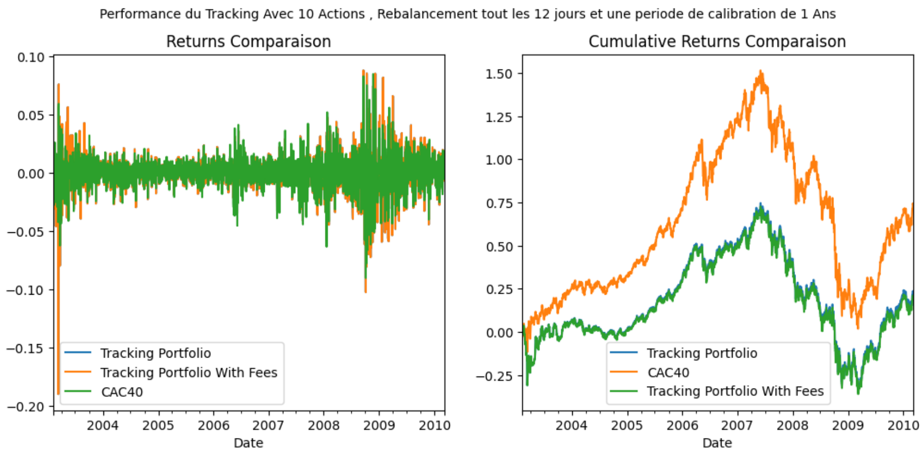


Figure 2: Performances with : $stock\_number = 22$, rebalancing period = 10, $calibration = 3, 5$ yrs, $fees = 0.1$

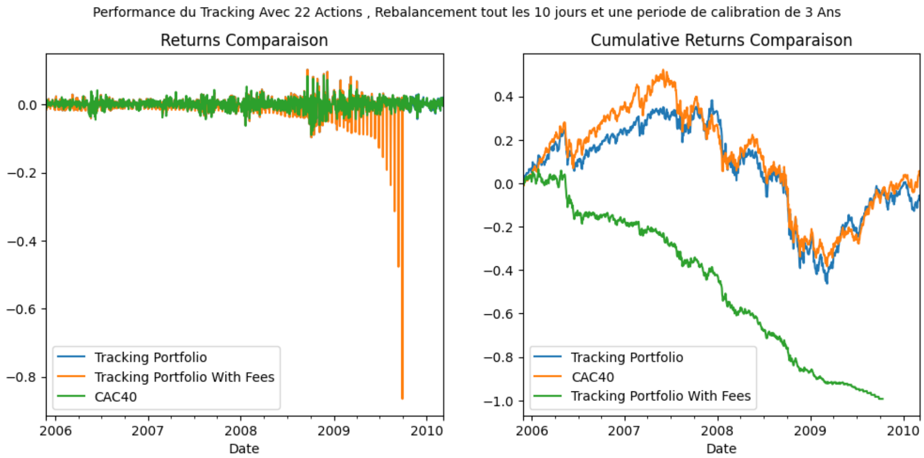


Figure 3: Performances with : $stock\_number = 22$, rebalancing period $= 10$, $calibration = 3,5$ yrs, $fees = 0.1$

We notice that the more stocks we include, and the less fee percentage, and longer calibration period , the better we track the index.

2.2.4 Interesting Findings and remarks

How Transaction costs evolved over time

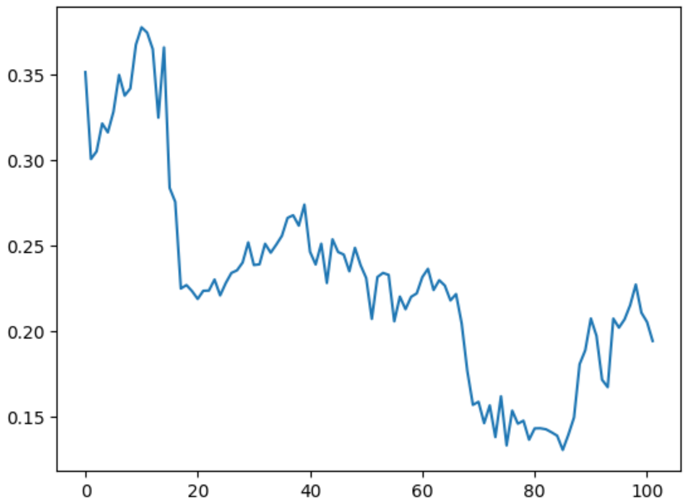


Figure 4: Evolution of Transaction fees

It is apparent that during the subprime crisis, our portfolio took less positions.Which also can be seen as a sign of stability of holdings

When the stock selection doesn't produce cointegrated time series

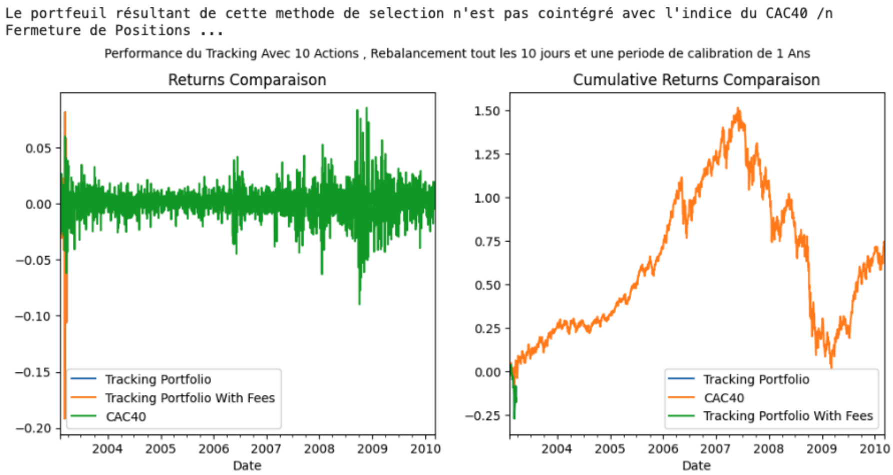


Figure 5: The Portfolio stops evolving after hitting a non cointegrated selection of stocks

The portfolio stopped evolving, a couple of months after initiation, having met a non-cointegrated selection of stocks

We now are capable of tracking an index using cointegration, let's now use this to build a market neutral strategy.

2.3 Long-short strategy

It is apparent that within the crisis of subprimes, our strategy took fewer positions, and therefore we paid the least amount of TC's.

Once the index tracking method based on cointegration is implemented, this tool can be used to develop a long-short market-neutral strategy. The goal is to capture excess performance relative to the CAC 40 while minimizing volatility and correlation with the market.

2.3.1 Strategy Construction

The method used for portfolio construction is as follows:

1. We create synthetic indices based on the CAC 40 with an overperformance of +/- x

```python
cac40Ret = Reconst_CAC40.pct_change().dropna()
prime=0.15
nDays=252
benchPlusRet = cac40Ret+prime/nDays
benchPlus = [Reconst_CAC40.iloc[0]]
for r in benchPlusRet:
    benchPlus.append(benchPlus[-1] * (1 + r))

benchPlus = pd.Series(benchPlus, index=Reconst_CAC40.index)

benchMinusRet = cac40Ret-prime/252
benchMinus = [Reconst_CAC40.iloc[0]]
for r in benchMinusRet:
    benchMinus.append(benchMinus[-1] * (1 + r))

benchMinus = pd.Series(benchMinus, index=Reconst_CAC40.index)
```

2. We create a portfolio P_+ to track the + index and a portfolio P_- to track the - index using the cointegration method. For this we must create two instances from the our classe, based on bench + and -. Using 30 stocks, a calibration period of 3 years and a half, and a rebalancing period of 10 days, as well as a fee pct of 0.002

Resulting portfolios

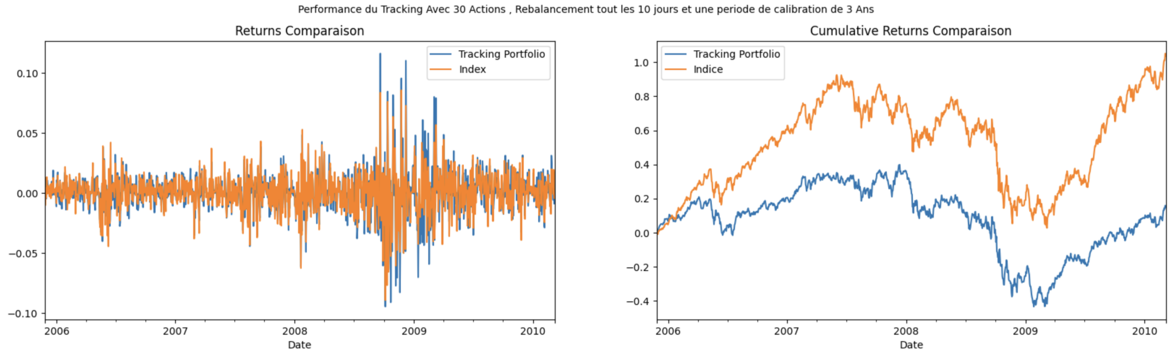


Figure 7: Comparison of benchmark +15% and tracking portfolio, using 30 stocks, rebalancing every 10 days and 3 years of calibration

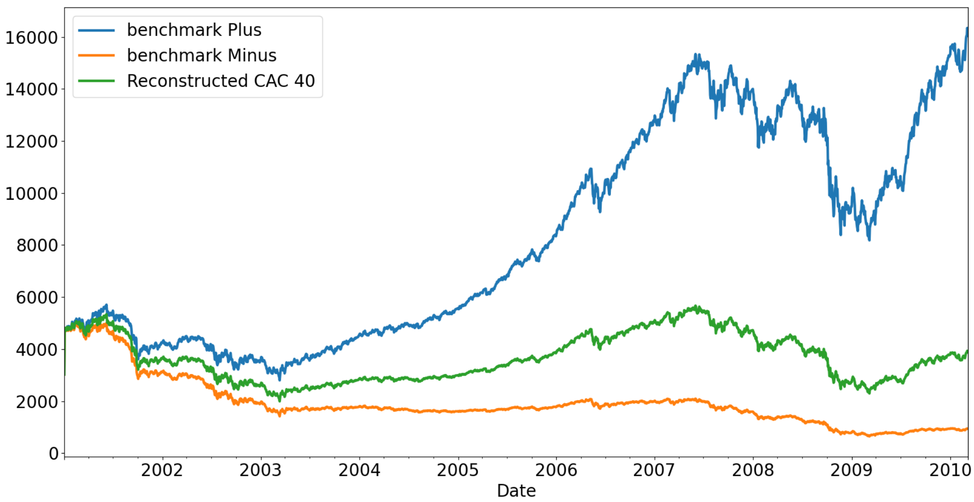


Figure 6: Comparison of returns of CAC40 with returns of benchmarks based on CAC 40 +/- 15% of annual return

3. We take a long position in P_+ and a short position in P_-. These two portfolios are strongly correlated with the benchmark, but their tracking errors are weakly correlated with the market. The difference between the two should produce a portfolio that is relatively insensitive to the market.

4. We consolidate ("Netting") the positions across the different stocks in order to reduce transaction costs. This impact becomes more significant when rebalancing occurs frequently.

This approach is thus expected to create a portfolio following a long-short strategy with low volatility due to its market neutrality, while still capturing excess return relative to the CAC 40.

```
transaction_costs.iloc[0]=coef_TC*np.sum(np.abs(portfeuilPlus.transaction.iloc[0]-
    portfeuilMinus.transaction.iloc[0])*Prices_without_cac.loc[portfeuilMinus.
    transaction.index[0]])
i = 11
while i < len(portfeuilPlus.transaction) :
  transaction_costs.iloc[i]=coef_TC*np.sum(np.abs((portfeuilPlus.transaction.iloc[i]-
      portfeuilMinus.transaction.iloc[i])-(portfeuilPlus.transaction.iloc[i-11]-
      portfeuilMinus.transaction.iloc[i-11]))*Prices_without_cac.loc[portfeuilMinus.
      transaction.index[i]])
  i+=11
transaction_costs.plot()

```

```
totalPrices_wo_TC=Initial_Budget+portfeuilPlus.prices-portfeuilMinus.prices
totalPrices=totalPrices_wo_TC-transaction_costs
plt.figure()
totalPrices_wo_TC.plot()
totalPrices.plot()
plt.show()
```

2.3.2 Empirical Results

We evaluated the strategy's results using several indicators:

1. Strategy return
2. Annualized volatility
3. Correlation with the CAC 40
4. Skewness and kurtosis
5. Sharpe ratio

Let's analyze these results. First, the portfolio indeed shows relatively low correlation with the CAC40. This value seems to stabilize around 0.3

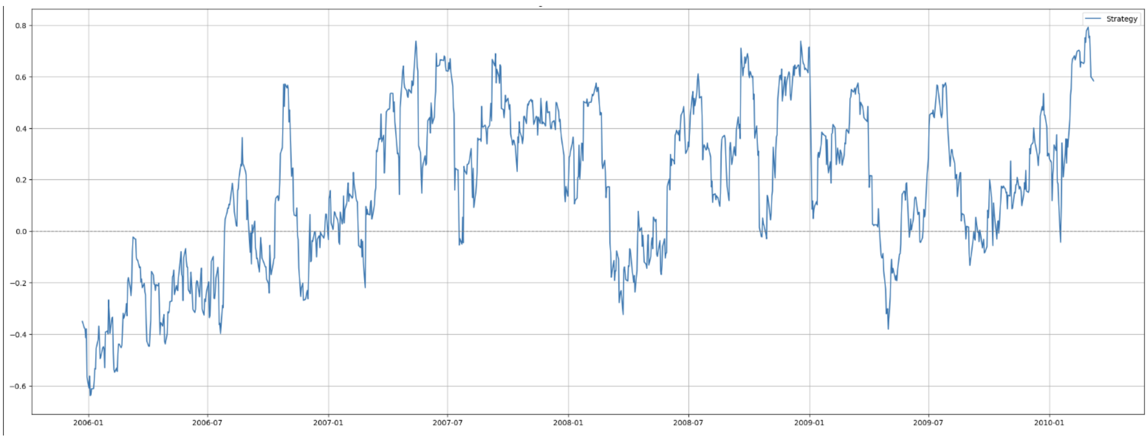


Figure 8: Rolling Correlation with index 20d

Furthermore, the strategy provides consistent returns, although sometimes lower than those of the CAC 40.

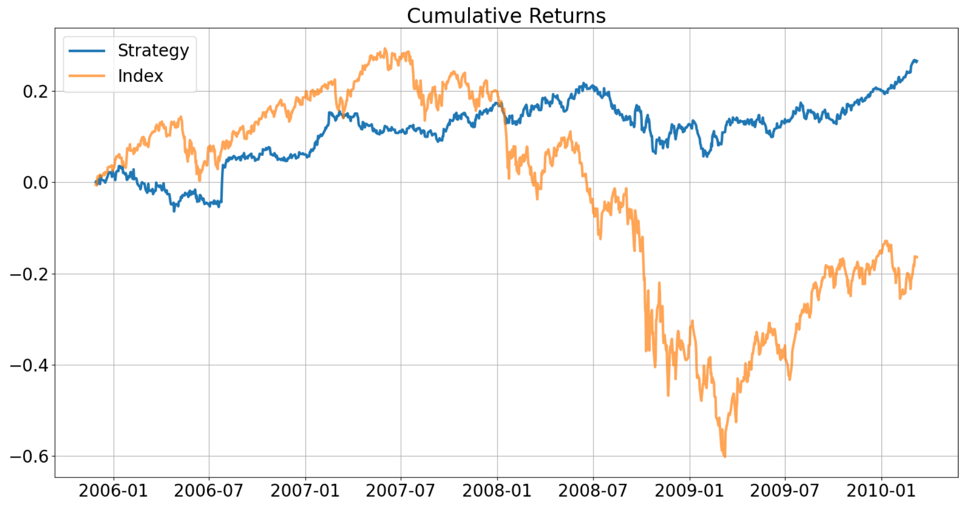


Figure 9: Comparison of cumulative returns between the strategy and CAC40

However, the volatility of the strategy is significantly lower than the volatility of the CAC40, especially during the global financial crisis. Low volatility was one of the objective of this strategy and it seems very well fulfilled.

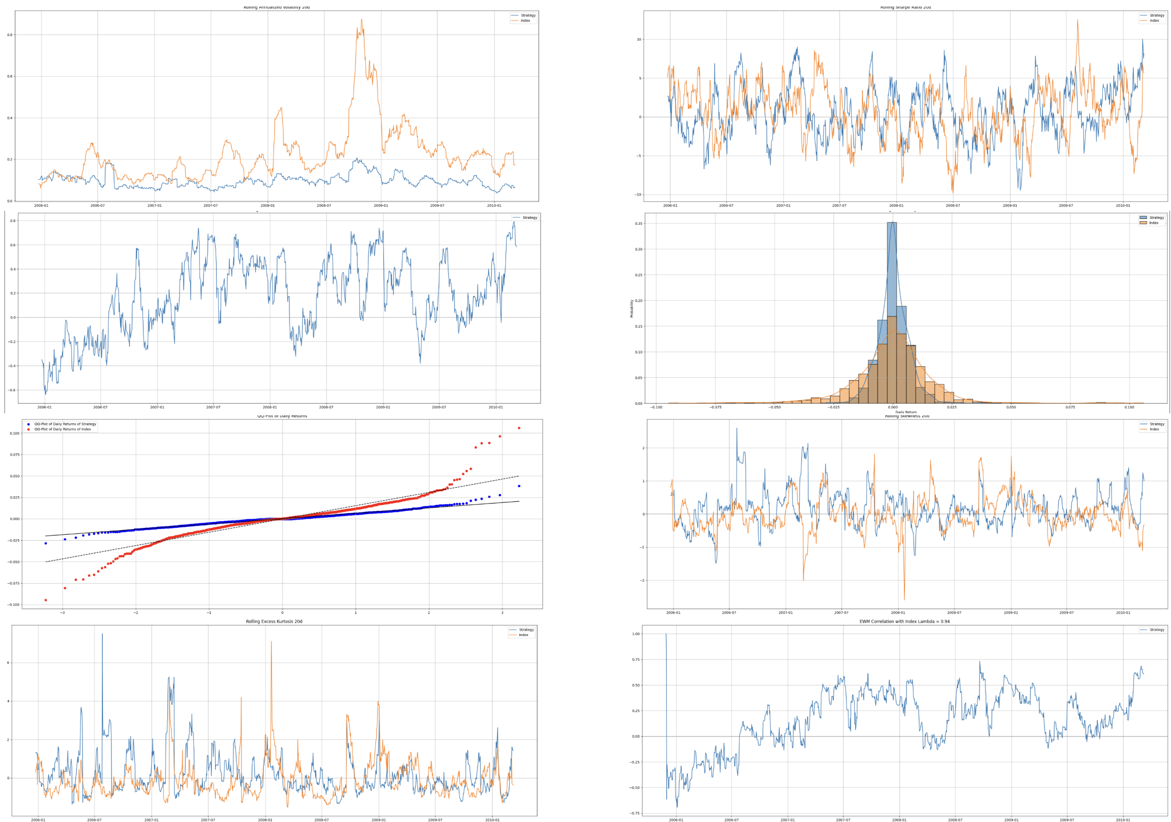


Figure 10: Statistical Plots on our Long-short Strategy

- The Sharpe ratio is generally positive, though occasionally negative. It is generally at the same level as the CAC40. However, whereas the CAC40 shows great returns and great risks, the strategy shows lower returns and lower risks.
- Moreover, we observe that the distribution of daily returns is clearly leptokurtic, with a much

higher frequency of returns very close to zero compared to CAC40.
When comparing the distributions of daily returns with the normal distribution, It seems the strategy gives a distribution slightly more normal than the distribution of the CAC40. However in both cases, we clearly observe the fat tail phenomenon.

- Skewness of returns seems generally the same between the strategy and the index, however the strategy seems to exhibit a slightly higher kurtosis.

3 Conclusion

In this report, we examined the employment of cointegration techniques to create advanced index tracking and market-neutral strategies, expanding on the methodology presented by Alexander and Dimitriu (2002). By utilizing their approach on daily data from the CAC 40, we validated that cointegration-based methods provide a resilient alternative to traditional correlation-based models, specifically for non-stationary financial time series.

The index tracking strategy revealed that, when cointegration exists, a meticulously calibrated portfolio can closely replicate the performance of the benchmark. This is achievable while needing fewer assets and demonstrating greater stability, notably during volatile times such as the 2008 financial crisis. Our results suggested that performance enhances with a higher number of selected stocks, extended calibration periods, and reduced transaction costs, all of which contribute to more efficient tracking.

Subsequently, we broadened this framework to construct a long-short market-neutral strategy by generating synthetic benchmarks with adjusted performance targets. The ensuing portfolio displayed diminished volatility and lower market correlation in comparison to the CAC 40, achieving its risk management objectives while upholding competitive Sharpe ratios. Moreover, the empirical evidence indicated that this strategy sustained stability and robustness across varied market conditions.

We can say that this project underlines the practical significance of cointegration in contemporary quantitative finance, serving as a robust tool for creating tangible trading strategies. Further research could investigate other indexes, more dynamic stock selection rules, alternative cointegration detection techniques and fine-tuning parameters of the method.

References

[1] Carol Alexander, Anca Dimitriu (2002). The Cointegration Alpha: Enhanced Index Tracking and Long-Short Equity Market Neutral Strategies. *THE BUSINESS SCHOOL FOR FINANCIAL MARKETS*.