LUISS

Department of Economics and Management

Course of Asset Pricing

Profitability of algorithmic Pairs Trading strategy: an empirical application on the Italian Stock Exchange using Cointegration approach

Candidate:

Kevin Graziani

Supervisor:

Prof. Paolo Porchia

ACADEMIC YEAR 2020/21

Abstract

The following work illustrates the development and results of a pairs trading strategy with a cointegration approach applied to the FTSE MIB, the main index of "Borsa Italiana" (Italian Stock Exchange), in period 01.01.2008 – 31.12.2020. With an average annual return of 32.91% and a Sharpe ratio of 1.29, the results obtained are interesting and consistent over time in terms of risk and return, demonstrating that a pairs trading algorithm can still be profitable despite early adopters' unlucky history.

In the first part of the dissertation, the concepts of Statistical Arbitrage are uncovered, from definition to history, different strategies, and statistical approaches. In the second part, the research work is presented, illustrating the different steps followed to develop the algorithm and finally digging into the results, with emphasis on excess returns, market correlation and possible future developments of the strategy.

A complete transcription of the Python code developed is provided in Appendix 1.0

Questions and further analysis requests can be sent to kevin.graziani@studenti.luiss.it

Index

Introduction

At the beginning of the 80s, the first trading desks to adopt statistical arbitrage strategies reached remarkable successes, thanks to the audacity of their leaders, the use of consistent financial leverage, and the *first comer advantage*, since the strategies deployed were practically unknown. These achievements, however, were soon compensated by equally spectacular failures, with the consequence of the financial community taking the distance from statistical arbitrage. In the first years of 21st century, statistical arbitrage, and particularly the underlying strategy of pairs trading, were not considered as *sound trading strategies*, and the returns achieved only a couple of decades ago were at the end just a short parenthesis of the financial history.

The aim of the dissertation is to demonstrate that a pairs trading strategy can still be a profitable investment strategy, and that the failures of the traders of the eighties are not a proof of the inefficacy of pairs trading, but rather an evidence of the limits of emotions in trading choices and a reckless use of leverage. In this respect, using the programming language Python, a pairs trading algorithm has been developed, able to remove human factor in the day-to-day operations and to obtain consistent results over time in terms of risk and return.

For the purpose of its deployment, the algorithm takes into account the forty stocks composing the FTSE MIB, the major index of the Italian Stock Exchange. It performs two different cointegration tests, Engle-Granger Test and Augmented Dicky-Fuller Test, over a period of six months, with the aim of selecting the cointegrated pairs that will be traded in the following period of equivalent duration. This approach can be fairly considered as a simple Machine Learning approach, given that the algorithm uses the data it computes in the training period to perform actions in the trading period.

During the trading period, the algorithm automatically opens long and short positions based on the abnormal movement of the z-index of the pairs. Without using stop-loss or take-profit, positions are closed when z-index comes back to a range of tolerable values. The use of financial leverage is not contemplated.

The algorithm is tested over a period of 13 years, from 2008 to 2020 included, testing different stressful financial periods, such as the subprime crisis of 2008, the fall of financial markets of 2018 and the pandemic crisis in 2020. The results obtained are promising: the average annual return is 32.91% while the annualized volatility is 22.55%. Sharpe ratio is 1.29. An average of 38 trades are opened each year, and the annual returns are always positive: minimum annual return is 0.07% in 2010 and the maximum is 93.72% reached in 2016. Correlation with the index is insignificant, with an average value of 0.07, while the returns of the algorithm are able to beat the index returns 11 of 13 years. Excess returns are positive each year except for 2010.

The exposition of the results are then followed by possible further developments that could contribute to reach better performances and increase efficiency, such as the optimization of portfolio weights and of pairs composition.

The dissertation is successful in its attempt to practically prove the efficacy of pairs trading thanks to the cointegration approach, to the absence of the human emotions in trading decisions, and to the absence of financial leverage.

Part One

1. Concept of Statistical Arbitrage

To grasp the different aspects of statistical arbitrage, it is important to start by understanding the concepts at the basis, inspecting some of the different definitions provided by the literature, its modern history and the various strategies deployed by the arbitrageurs.

1.1. Definition

The first definition of *arbitrage opportunity* comes from the well-known theory of *market efficiency,* in which market players are ready to take advantage of every mispriced asset, thereby rebalancing the market and reinstating the correct valuation. In particular, the opportunity of arbitrage arises when two *equivalent assets are mispriced* in terms of one to the other. The arbitrageurs profit from the difference between such prices and, by doing so, they narrow the price of the equivalent assets until the point in which market efficiency theory holds again.

A simplistic example of arbitrage opportunity is with cross-border listed companies. Company X, listed on both NYSE and Canada's TSX, is trading respectively at $8.00 USD and $10.00 CAD. Given that the current CAD/USD exchange rate is 1.10, a trader could purchase stocks on the NYSE for $8.00 USD and at the same time selling them on the TSX for $10.00 CAD (equal approximately to $9.09 USD), therefore realizing a profit of $1.09 for each stock traded.[1]

The main aspect of arbitrage opportunity is that the trade of the arbitrageur is profitable and, in the meantime, it doesn't involve any direct risk. This is the main difference with statistical arbitrage, as in the latter risks exist and they are statistically assessed.

---

[1] Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/arbitrage/

Statistical arbitrage (SA), as pointed out by Lazzarino *et al*, has been broadly defined by the literature from different points of views, with lexical, conceptual, abstract, and operational definitions.[2] For example, among the conceptual definitions, Connor and Lasarte define SA as *a zero-cost portfolio where the probability of a negative payoff is very small but not exactly zero*. Among the operational definitions, Ledoit defines SA as an investment strategy with a Sharpe ratio above a constant and strictly positive level.

The issue with all the definitions provided is that they are not able to incorporate all the different strategies and approaches of SA.

Lazzarino *et al* tried to offer a conceptual definition to fill that gap:

SA strategy is a relative value strategy with a positive expected excess return and an acceptably small potential loss.

Analyzing the definition provided, it is possible to find the key features of SA. The term "relative value strategy" refers to strategies that aim to find mispricing using historical relationships and combining the underlying securities (that can or cannot belong to the same asset class) in a long-short portfolio, thereby isolating and hedging different risks.

By referring to "positive expected excess return", the definition admits possible negative outcomes that are tolerated only when there are *expectations of higher than the risk-free returns*. Finally, "acceptably small potential loss" helps to differ SA from other investment strategies, given that the loss profile must be constrained.

For the purpose of the analysis, the definition provided above includes all the aspects needed to take into account, giving to the arbitrageurs the chance of suit the criteria with their risk measures and utility function.

---

[2] Lazzarino, M., Berrill, j. And Šević, A. (2018) What Is Statistical Arbitrage? *Theoretical Economics Letters*, 8, 888-908. https://doi.org/10.4236/tel.2018.85063

1.2. History

The history of statistical arbitrage can be traced back to the beginning of the 80s, when a group of researchers at Morgan Stanley, guided by Nunzio Tartaglia, created a program able to buy and sell stocks in pair combinations. The group gained huge attention (and profits) in the upcoming fifteen years and "pairs trading" officially entered the financial lexicon.[3]

The basis of pairs trading is as easy as effective: when a pair of stocks show similar historical price behavior, it is possible to bet on convergence of price as soon as the prices themselves diverge. But the term "pairs trading" became inadequate as an enormous amount of sophistication came into the playground. In the early 1990s the term "statistical arbitrage" started to be used by practitioners.

From the ultra-secretive group, some members soon found their independent way out of the investment bank. D.E Shaw and Double Alpha are just two of the firms created by former analysts of Tartaglia's team.

Even if Morgan Stanley was the place in which statistical arbitrage was born, perhaps its most intriguing story traces its roots back at Salomon Brothers' fixed income arbitrage group. Here, a smart and young bond trader named John Meriwether (JM) learned to surf the pricing differences between correlated bonds and to profit from their convergence, and soon made its way in the bank's hierarchy to set up the Arbitrage Group of the investment bank. Given his academic background, Meriwether started to hire different scholars from Eric Rosenfeld to Victor J. Haghani and Lawrence Hilibrand, every one of them composing the future dream team of Long-Term Capital Management (LTCM), the firm Meriwether opened in 1994.[4]

After more than a decade, JM found himself involved in a scandal at Salomon Brothers that pushed him out of the bank. The break of this relationship prompted Meriwether to open his own hedge fund under the name of Long-Term Capital Management. Members of the fund were in essence some of the scholars he previously hired in his former Arbitrage Group, but also other academics like Myron Scholes, Robert Merton, and Eugene Fama.

---

[3] A. Pole, Statistical Arbitrage: Algorithmic Trading Insights and Techniques - 2007. Wiley Finance
[4] R. Lowenstein, When Genius Failed: The Rise and Fall of Long-Term Capital Management – 2000. Random House

LTCM soon proved its value by reaching an average annual return of around 40% in the first 4 years. The academics of the fund, trading mainly in the fixed income market, were able to secure tiny profits by applying statistical arbitrage strategy of convergence trading. Their cold-blooded approach, mixed with a huge leverage generously provided by banks, were able to transform these tiny profits into huge gains for partners and investors of the fund.

Troubles came during the summer of 1998, when Russian financial crisis led to government default on its domestic bonds. Also, the liquidation of similar positions owned by other market players further stressed the positions taken by LTCM, that suffered margin calls by all the banks that had provided leverage for its operations. A bailout was organized by the Federal Reserve Bank of New York, and the partners ultimately wiped out all their investments inside the fund.

As Lowenstein recalls in his book "When Genius Failed", the strategy of Long-Term Capital Management could be seen as *picking nickels in front of a bulldozers* as the likely small profits of the fund were balanced against a small chance of an extreme loss.

In the beginning of the 21st century, the profits coming from statistical arbitrage's strategies started to shrink. Regarding the public opinion, the decline of performance was the direct cause of the growing number of market players that started to deploy these strategies, and of a change of the fundamentals of the market itself.

Nonetheless, prominent authors and researchers, fund managers and industry insiders believe that the fall of statistical arbitrage can be considered the bottom of an economic cycle and with further study and development of the strategy class, statistical arbitrage can conquer back its past glory.

1.3. Strategies

Following the analysis of Duarte Longstaff and Yu[5] and the already mentioned work of Lazzarino *et al*, a comprehensive list of the different strategies is presented.

1.3.1 Pairs Trading

As already seen, pairs trading is the most common SA strategy. Pairs trading is based on the idea that, thanks to statistical methods, it is possible to identify the long-term equilibrium in the spread between the components of a given pair of stocks. Regardless of the market trends, the strategy's trade is profitable whether the spread reverts to the mean. It can be applied to commodities as well as to portfolio, and the long-short positions that are opened to profit from when the spread widens can be combined in a ratio that makes the trade market-neutral. The different approaches to pairs trading are reviewed later in the dissertation.

1.3.2 Term Structure Arbitrage

Term structure arbitrage involves the use of a relative value analysis that suggest the opening of market-neutral long-short positions at different points of a term structure. Naturally, the trade is closed as soon as the mispricing disappears. It is a common strategy in fixed income (yield curve arbitrage) and commodities.

1.3.3 Volatility Arbitrage

Generally, volatility arbitrage is a bet on volatility that is structured to be independent from the underlying's performance. It can be categorized into different strategies: *gamma trading* focuses on implied vs historical volatility on the same asset; *volatility surface arbitrage* plays the implied volatility of a given asset in different points of the volatility surface; *cross-asset volatility trading* acts on the implied volatilities of different assets with a long-short positions' approach; *dispersion trading* analyses and trades the volatility of an index versus the volatilities of its components.

---

[5] Duarte, J., Longstaff, F.A. and Yu, F. (2006) Risk and Return in Fixed Income Arbitrage: Nickerls in Front of a Steamroller? The Review of Financial Studies

1.3.4 Swap Spread Arbitrage

Swap spread arbitrage is a SA strategy that plays on the difference between a floating and a fixed yield. The arbitrageurs bet that the floating yield exceeds the fixed yield by entering a par interest rate swap that pays a fixed coupon rate and receiving the floating LIBOR rate, at the same time buying a treasury bond with the money borrowed through a repo. The cash flow of the trade is the difference between the LIBOR (L_t) and the repo rate (r_t) minus the difference between the fixed coupon rate (SR) and the treasury rate (TR).[6]

1.3.5 Mortgage Arbitrage

Mortgage arbitrage's strategy is based on the purchase of mortgage-backed securities and the hedge of their interest rate exposure using derivatives. The bet is sustained by the fact that the yield of mortgage-backed securities is usually higher than that of comparable treasury bonds.

1.3.6 Capital Structure Arbitrage

Capital structure arbitrage consists in long-short positions in different instruments of the same company's capital structure. It can be further divided into Credit and Convertible arbitrage. The former aims to exploit mispricing between a company's equity and its credit default swap. Convertible arbitrage consist in buying a portfolio of convertible bonds hedging the position with a short selling of the underlying stocks.

---

[6] Krishnamurthy, A. (2008) Fundamental Value and Limits to Arbitrage

1.4 Pairs Trading Approaches

Given its importance for the following empirical application and for the general statistical arbitrage's landscape, the different approaches of pairs trading are reviewed in this chapter. The work of Christopher Krauss "Statistical arbitrage pairs trading strategies: Review and outlook" is considered to provide a comprehensive yet quick overview of pairs trading approaches, but it must be noted that different other studies have been considered by the academic.

Based on Gatev et al. (2006), pairs trading follows a two-period process: in the first period (training period) the objective is to find two securities with similar historical prices movements; the second period (trading period) involves the monitoring of the spread between the prices. Whenever the spread widens, two positions are opened: short position on the overvalued stock and a long position on the undervalued one. The positions are then closed as soon as the spread reverts to its historical mean.

1.4.1. Distance Approach

As for the name, distance approach employs distance metrics during the training period to identify the comoving assets. Then, during the trading period, simple nonparametric threshold rules give birth to trading signals. With Gatev, Goetzmann and Rouwehorst, a baseline approach has been developed with the use of Euclidean squared distance for the pairs' identification. Not performing a cointegration testing over the identified correlated pairs may result in spurious results (Alexander, 2001). Pearson correlation can further improve results by spotting other pairs opportunities missed by the previous correlation method.

1.4.2 Cointegration Approach

The cointegration approach computes cointegration tests in the training period to identify the comoving pairs, resulting in a more reliable equilibrium relationship of identified pairs when compared to distance approach. As for the latter, the trades in the trading period are usually triggered by applying simple nonparametric thresholds. In the literature landscape, the pioneer of this approach is Vidyamurthy with the 3-step process: (1) select potentially cointegrated

pairs based on fundamentals; (2) test for tradability with cointegration tests, such as Engle-Granger test and Johansen Test; (3) trade the pairs with nonparametric rules.

1.4.3 Time Series Approach

The literature on the time series approach focuses on optimizing trading signals by using different methods of time series analysis, specifically by modeling the spread as a mean-reverting process. The training period is ignored by the authors of this approach. Without entering in detail, Elliott et al. (2005) are the most followed authors, with their explicit description of the spread as a mean-reverting Gaussian Markov chain in Gaussian noise.

1.4.4 Stochastic Control Approach

As in the previous approach, even in the stochastic control approach the training period is ignored. This approach consists in identifying the optimal portfolio in the legs of a pairs trade compared to other assets available.

1.5 Analysis of Cointegration approach

The analysis and strategy deployed in the second part of the dissertation are based on the cointegration approach mentioned before. It is therefore necessary to dig more into it.

1.5.1. Mathematics of cointegration approach

Cointegration occurs when two or more nonstationary time series (1) have a long-run equilibrium, (2) move together such that their linear combination results in a stationary time series and (3) share an underlying common stochastic trend.

Consider a group of time series, Y_t , composed of three separate nonstationary time series:

$$y_1 = (y_{11}, y_{12}, \dots, y_{1t})$$
$$y_2 = (y_{21}, y_{22}, \dots, y_{2t})$$
$$y_3 = (y_{31}, y_{32}, \dots, y_{3t})$$

Based on cointegration, the time series can be structured to form a linear combination that is stationary:

$$\beta Y_t = \beta_1 y_{1t} + \beta_2 y_{2t} + \beta_3 y_{3t} \sim I(0)$$

Where β is the cointegrating vector of the time series. It is important to consider it as non-unique since there could be multiple ways of cointegrating the same time series. Therefore, imposing identification restrictions is mandatory to normalize the cointegrating vector for estimation.

The cointegrating vector can be normalized by setting $\beta = (1, -\beta_2, \dots, -\beta_N)$, so that the previous linear combination can be rewritten as:

$$\beta Y_t = y_{1t} - \beta_2 y_{2t} - \beta_3 y_{3t} \sim I(0)$$

11

Or directly in a standard regression form:

$$y_{1t} = \beta_2 y_{2t} + \beta_3 y_{3t} + u_t \sim I(0)$$

Where u_t is a stationary cointegrating error component. The latter represents the short-term deviations of the linear combination from the long-run equilibrium.

1.5.2. Why cointegration exists among stocks

There are multiple reasons why it is possible to encounter cointegrated stock pairs in the market. Here some of these reasons are indicated, but the list should not be considered as exhaustive as its goal is just to provide some examples of possible causes of cointegration.

Two stocks can be cointegrated because of the industry in which the underlying companies operate. It is fairly easy to find cointegrated stocks in the same industry because of the common trend followed by the stocks themselves. Else, the two companies can be in two different but related industries, such as consumer software development and consumer technology hardware, that often move in the same directions.

Another factor that makes stocks cointegrated is their presence in the same portfolio. In different conditions and situations, two stocks can present cointegration because they are traded in the same fashion, i.e. bought and sold at the same time by different actors in the market without reasons related to the fundamentals of these stocks. An example can be the selling of specific stocks to face a margin call hitting the portfolios of multiple investors.

Cointegrated patterns can be observed in stocks of companies that have related interests in monetary and fiscal policy. Stocks of companies that depend on import-export often move in similar directions after changes of monetary and fiscal policies in home country as well as in the related foreign countries.

The common use of the assets can create cointegration between them. Gold and Bitcoin are considered as "safe heavens" and therefore traded in similar fashion. Recently, trends started to be noticed by the market and this relationship can bring to cointegration in the future.

Finally, psychology as always plays an important role. Cointegration can be observed among assets or data that the market perceives as related, even if there is no evident correlation among them. Jim Simons and its Medallion Fund found an interesting relation between weather and stock price's movements, though not enough profitable to keep it secret.

1.5.3. Applications of cointegration approach

As seen before, the error component indicates whether a deviation from the equilibrium is encountered in time t. This is an essential aspect when trading a cointegrated pair of stocks in Statistical Arbitrage. In fact, the component itself offers the chance to construct a signal that suggest when the positions should be opened (like the z-score in the strategy described in Part Two).

Cointegration is a preferred approach to statistical arbitrage for a simple reason: it offers an analysis of the existence of a long-run relationship between non-stationary time series, as opposed to correlation, the latter being able only to spot linear relationships between stationary variables.

Therefore, to avoid spurious results, one should prefer cointegration tests as opposed to correlation tests when looking for statistically significant asset pairs to trade with Statistical Arbitrage strategies.

Part Two

In the second part of this dissertation, an empirical application of a pairs trading strategy is presented. The pairs trading strategy deployed follows the cointegration approach of Caldeira and Valle Moura (2013), with necessary differences explained below.

2. Pairs trading strategy on the Italian Stock Exchange

The pairs trading strategy developed aims at testing the profitability of such a statistical arbitrage strategy on the Italian Stock Exchange (Borsa Italiana) in the range period 2008-2020. With the use of Python and Microsoft Excel, a cointegration approach has been carried out over the 40 stocks currently composing the FTSE MIB Index, and signals are created based on z-score of cointegrated pairs and thresholds suggested by Caldeira et al.

Each year is divided in two different periods of same duration of six months: the training and the trading period. The algorithm's choices of trading are based on the computations performed during the training period, where stock pairs are selected through a process of cointegration tests, and no computations are made during trading period. This approach follows the rules of Machine Learning development, where the algorithm is free to test data previously computed, without any new input able to change them. Six months are here considered as a period long enough to avoid selecting pairs that are not truly cointegrated. Also, as market conditions and dynamics are prone to frequent changes, it appears as a judicious choice not to trade for a long period on the same computations. Therefore, the trading period has been set to six months.

The choice of selecting the stocks of the Italian index is mainly based on two considerations: first, these 40 stocks are the most traded stocks in Borsa Italiana, making them the perfect candidates for opening short selling positions without incurring in possible limits related to market volume, therefore removing this constraint in the analysis. Also, the analysis is trying to prove that pairs trading is a sound investment strategy in the Italian equity market, and the FTSE MIB has been especially designed to track the general trends of the Italian stock market.

2.1 Preliminary Remarks

Some constraints faced and simplifications adopted during the development of the strategy must be presented before entering the details of the strategy itself.

The first limit has been the impossibility of finding the components of the FTSE MIB Index for each year of the strategy. Given this constraint, only the 40 stocks composing the index in 2021 are used to carry out the cointegration tests, as indicated by the official website of Borsa Italiana. The risk-free rate used to calculate the annual excess return of the strategy is the value of the *Italy 10-year Government Bond* as of July of each year (the month in which trading period begins).[7] The index annual return is the return of the index during the trading periods.

Given that a pairs trading strategy is a cash-free strategy, i.e. the long position is funded with the short position, no initial capital is provided, therefore the results are computed in absolute terms. No trading fees are applied, and no stop-loss or take-profits are implemented.

For each trading period, the return of the strategy is a weighted average among the returns of each pair. The reason of this approach is that an equal amount of capital is allocated over each pair selected by the training period.

It may happen that two pairs have one same stock traded, resulting in a possible synergy in terms of cost savings between the two pairs trading. Given that trading fees are not applied, the model doesn't take advantage of the potential synergies.

---

[7] https://www.marketwatch.com/investing/bond/tmbmkit-10y?countrycode=bx

2.2 Training Period

During the training period, all the closing prices of the stocks composing the FTSE MIB Index 2021 are considered. Given that not all the 40 stocks composing the 2021 version of the Index were listed in the years of the strategy deployment, not all the 780 possible combinations are available for every year (Table 1.0).

The dataset of closing prices of the stocks are split in windows of 126 days, approximately 6 months of trading. In the first period, the model finds the cointegrated pairs to be traded in the second period.

Table 1.0 Number of possible pairs for each year in the range 2008-2020

Year	# of stocks tested	# of possible pairs
2008	31	465
2009	31	465
2010	31	465
2011	31	465
2012	31	465
2013	31	465
2014	33	528
2015	34	561
2016	37	666
2017	38	703
2018	39	741
2019	40	780
2020	40	780

The pairs combinations are assessed with two different cointegration tests.

The first one deployed is the Engle-Granger test, which constructs residuals based on the static regression. The test uses the residuals to see if unit roots are present using ADF test or similar tests. The residuals will be practically stationary if the time series is cointegrated.[8] The number of lags is chosen to minimize the corresponding information criterion.

If a set of variables are cointegrated, then there exists a valid error correction representation of the data, and viceversa.[9]

The two-step process of Engle-Granger test includes the estimation of the long-run equilibrium (EG 1) and the estimation of the Error Correction Model (EG 2).

$$Y_t = \delta_0 + \delta_1 x_t + u_t$$

Where u_t represents the OLS residuals. The output of the test are the parameters a and b (respectively δ_0 and δ_1), the p-value and the t-statistic. Only the pairs that show a t-statistic less than the critical value at 5% can proceed to the subsequent test.

The pairs selected by the Engle-Granger test are used to build an OLS regression model.

$$Y_t = a + bX_t + \varepsilon_t$$

Where *Y* and *X* are the stocks composing each pair. The results of the OLS regression are then used to compute the second test.

The second test is the well-known Augmented Dicky-Fuller test (ADF), a unit root test that uses an autoregressive model and optimizes an information criterion across multiple different lag values.[10] Given that the E-G test already deployed the ADF test in the first step of its process, now the spread of the stocks is used, calculated as follows:

$$\varepsilon_t = Lt - aSt$$

---

[8] https://www.statisticshowto.com/engle-granger-test/
[9] Engle and Granger (1987) Cointegration and Error Correction: Representation, Estimation and Testing.
[10] https://machinelearningmastery.com/time-series-data-stationary-python/

Where L_t and S_t are the two closing prices of the stocks in the pairs at time t, ε_t is the spread at time t, and a is the coefficient parameter resulting from the OLS regression seen previously. Also in this case, the only pairs that are selected for the trading period are the ones that can pass the cointegration test with a critical value of 5%.

After performing the two tests process on all the possible combinations available, the result is a set of pairs for each year that are ready to be traded in the trading period.

The pairs are ranked by t-statistic in ascending order, given that the more negative is the value, the more likely the dataset is stationary. Finally, it is possible to select only the first 20 pairs that are going to be traded in the trading period.

Table 2.0 Number of resulting pairs from the cointegration tests in the range 2008-2020

Year	Engle-Granger test	ADF test
2008	14	2
2009	37	2
2010	61	15
2011	40	17
2012	46	2
2013	24	2
2014	62	19
2015	45	4
2016	72	8
2017	52	20
2018	28	10
2019	63	2
2020	122	9
Total	666	112
Average	51,23	8,62

In Table 2.0 one can see the number of stock pairs that passed the first test (for example, 34 in 2009) and the number of them that passed also the second test (also for 2009, only 2) . Given

the relatively low number of possible combinations in the range 2008-2013, it was expected a relatively low number of cointegrated pairs available for trading, compared to the subsequent years. This expectation is fulfilled: an average of 6.7 pairs are traded in the range 2008-2013, against an average of 10.3 pairs for the period 2014-2020. For the whole period, an average of 8.62 pairs are traded each year.

Only in one year the gap of 20 pairs has been reached: in the training period of 2007, the pairs that resulted from the ADF test were 29. Without the cutting of the last 9 pairs, the return of the year would have been slightly lower (25.06% instead of 27.06%). The decision to limit the tradable pairs to only 20 comes from Caldeira et al., even if the approach used to rank the pairs differ from the one used in Caldeira's strategy involving the in-sample Sharpe Ratio.

The procedure explained above is repeated in a rolling fashion for every training period of each year in the range 2008-2020.

2.3 Trading Period

Having selected the cointegrated pairs, the trading period can be carried out. The trading period lasts approximately 6 months, from the beginning of July until the end of December of each year. To perform the trading, a set of signals are created to indicate the moments in which the pair is above (or below) the long-run equilibrium.

2.3.1 Signals

The signals are based on z-score, that is the measure of the distance to the long-term mean in units of long-term standard deviation.

$$z_t = \frac{\varepsilon_t - \mu_\varepsilon}{\sigma_\varepsilon}$$

In the formula above, ε_t is the value of the spread at time t. The rule suggested for opening and closing positions by Caldeira *et al* are followed without any other changes.

When the z-score hits the 2 standard deviation thresholds from above or from below, a signal is created and the position is opened. When the z-score of the pair, composed of stocks *L* and *S*, hits the -2 threshold, it suggests that the pair is below its long-run equilibrium value, so buying the pair (i.e. buying stock *L* and selling stock *S*) is the right choice.

Below the trading rules for the signals:

Open short position if $z_t > 2.00$	*Close short position if* $z_t < 0.75$
Open long position if $z_t < -2.00$	*Close long position if* $z_t > -0.50$

The signals are stored in a table that keep tracks of all the positions opened during the period. It is worth to remember the there are no settings for stop-loss or take-profit, and the signals can last for all the period without any limit of days, in contrast with the approach of Caldeira *et al.*

Table 3.0 Example of Signals created from 30/06/17 to 01/08/17 for Stock Pair ATL.MI – BZU.MI

Signals 2017 for Stock Pair ATL.MI - BZU.MI

Date	ATL.MI	BZU-MI	Z-Score	Signals Short Open	Short Positions Opened	Short Positions Close	Signals Long Open	Long Positions Opened	Long Positions Close
30/06/17	24,64	21,79	-1,2014	0	0	0	0	0	0
03/07/17	24,76	22,65	-1,7652	0	0	0	0	0	0
04/07/17	24,84	22,51	-1,6101	0	0	0	0	0	0
05/07/17	24,81	22,43	-1,5712	0	0	0	0	0	0
06/07/17	24,70	22,45	-1,6594	0	0	0	0	0	0
07/07/17	24,51	22,50	-1,8225	0	0	0	0	0	0
10/07/17	24,60	22,91	-2,0557	0	0	0	1	1	0
11/07/17	24,58	22,56	-1,8194	0	0	0	0	1	0
12/07/17	24,99	22,90	-1,7936	0	0	0	0	1	0
13/07/17	24,94	22,78	-1,7403	0	0	0	0	1	0
14/07/17	24,83	22,61	-1,6899	0	0	0	0	1	0
17/07/17	24,93	22,45	-1,5059	0	0	0	0	1	0
18/07/17	25,30	22,15	-1,0303	0	0	0	0	1	0
19/07/17	25,67	21,93	-0,6059	0	0	0	0	1	0
20/07/17	25,80	22,29	-0,8017	0	0	0	0	1	0
21/07/17	25,39	21,73	-0,6375	0	0	0	0	1	0
24/07/17	25,50	22,10	-0,8560	0	0	0	0	1	0
25/07/17	25,66	21,78	-0,4920	0	0	0	0	1	1
26/07/17	25,70	21,87	-0,5372	0	0	0	0	0	0
27/07/17	25,70	21,74	-0,4319	0	0	0	0	0	0
28/07/17	25,45	21,28	-0,2251	0	0	0	0	0	0
31/07/17	25,69	21,40	-0,1575	0	0	0	0	0	0
01/08/17	25,90	21,66	-0,2282	0	0	0	0	0	0

Table 3.0 shows an example of how the signals are created and managed by the algorithm. The Z-Score of 10/07/2017 is -2.0557, reaching the threshold of -2.0 and thereby suggesting to enter a long position, i.e. buy the first stock of the pair (ATL.MI) and sell the second one (BZU.MI). In the table, the column "Signals Long – Open" shows the number 1, indicating that one position is opened, while the column "Long Positions – Opened" takes track of all the positions opened at the same time.

The position is held until 25/07/2017 when the closing threshold of -0.5 is reached by a Z-Score of -0.4319. In this case the column "Long Positions – Close" displays the number 1, indicating that one long position is closed. The next trading day has no signals and, subsequently, no opened positions.

2.3.2 Trading

Given the signals created, it is possible to proceed to trade the pairs which have signals during the trading period.

Following the stocks pair shown in the example in table 3.0, here it is reported a new table (Table 4.0) in which the returns are computed based on the signals previously created.

Table 4.0 Example of trades carried out from 30/06/17 to 01/08/17 for Stock Pair ATL.MI – BZU.MI

Trading 2017 for Stock Pair ATL.MI - BZU.MI

Date	Prices		Percentage Change		Returns of Positions		Daily Returns	Cumulative Returns
	ATL.MI	BZU.MI	ATL.MI	BZU.MI	Long	Short		
30/06/17	24,64	21,79	/	/	/	/	/	/
03/07/17	24,76	22,65	0,4870%	3,9468%	0,00000%	0,00000%	0,000000%	0,000000%
04/07/17	24,84	22,51	0,3231%	-0,6181%	0,00000%	0,00000%	0,000000%	0,000000%
05/07/17	24,81	22,43	-0,1208%	-0,3554%	0,00000%	0,00000%	0,000000%	0,000000%
06/07/17	24,70	22,45	-0,4434%	0,0892%	0,00000%	0,00000%	0,000000%	0,000000%
07/07/17	24,51	22,50	-0,7692%	0,2227%	0,00000%	0,00000%	0,000000%	0,000000%
10/07/17	24,60	22,91	0,3672%	1,8222%	0,36720%	-1,82222%	-1,455024%	-1,455024%
11/07/17	24,58	22,56	-0,0813%	-1,5277%	-0,08130%	1,52772%	1,446416%	-0,008608%
12/07/17	24,99	22,90	1,6680%	1,5071%	1,66802%	-1,50709%	0,160929%	0,152322%
13/07/17	24,94	22,78	-0,2001%	-0,5240%	-0,20008%	0,52401%	0,323936%	0,476257%
14/07/17	24,83	22,61	-0,4411%	-0,7463%	-0,44106%	0,74627%	0,305208%	0,781465%
17/07/17	24,93	22,45	0,4027%	-0,7077%	0,40274%	0,70765%	1,110391%	1,891856%
18/07/17	25,30	22,15	1,4842%	-1,3363%	1,48415%	1,33631%	2,820459%	4,712316%
19/07/17	25,67	21,93	1,4625%	-0,9932%	1,46245%	0,99322%	2,455679%	7,167994%
20/07/17	25,80	22,29	0,5064%	1,6416%	0,50642%	-1,64159%	-1,135165%	6,032829%
21/07/17	25,39	21,73	-1,5891%	-2,5123%	-1,58915%	2,51234%	0,923197%	6,956026%
24/07/17	25,50	22,10	0,4332%	1,7027%	0,43324%	-1,70272%	-1,269475%	5,686551%
25/07/17	25,66	21,78	0,6275%	-1,4480%	0,62745%	1,44796%	2,075413%	7,761964%
26/07/17	25,70	21,87	0,1559%	0,4132%	0,00000%	0,00000%	0,000000%	7,761964%
27/07/17	25,70	21,74	0,0000%	-0,5944%	0,00000%	0,00000%	0,000000%	7,761964%
28/07/17	25,45	21,28	-0,9728%	-2,1159%	0,00000%	0,00000%	0,000000%	7,761964%
31/07/17	25,69	21,40	0,9430%	0,5639%	0,00000%	0,00000%	0,000000%	7,761964%
01/08/17	25,90	21,66	0,8174%	1,2150%	0,00000%	0,00000%	0,000000%	7,761964%

Given the prices of the period, the daily gains and losses are computed as percentage difference between price at time *t* and price at time *t – 1* for each stock. As seen in Table 3.0, a signal to enter a Long Position is created in 10/07/2021. Accordingly, the same day it is recorded a long position for stock ATL.MI and a short position for stock BZU.MI, reporting respectively a gain and a loss. The cumulative result of the day is reported in column "Daily Returns", with a value of -1.46%, rounded to the second decimal. The column "Cumulative Returns" computes the cumulative gain (or loss) of the positions opened and closed in the period. For example, cumulative return in day 12/07/17 is 0.15%, the sum of the previous day cumulative return (-0.01%) and the current daily return (0.16%). When position is closed in 25/07/17, the

"Cumulative Returns" column saves the last value obtained and reports it until the end of the period. In this case, one can say that the strategy's result in 01/08/17 has been 7.76% so far.

The algorithm can handle multiple long and short positions at the same time, effectively computing the daily and cumulative returns.

Graph 1.0 Prices and Z-Score of Stock Pair ATL.MI – BZU.MI in Trading Period 2017
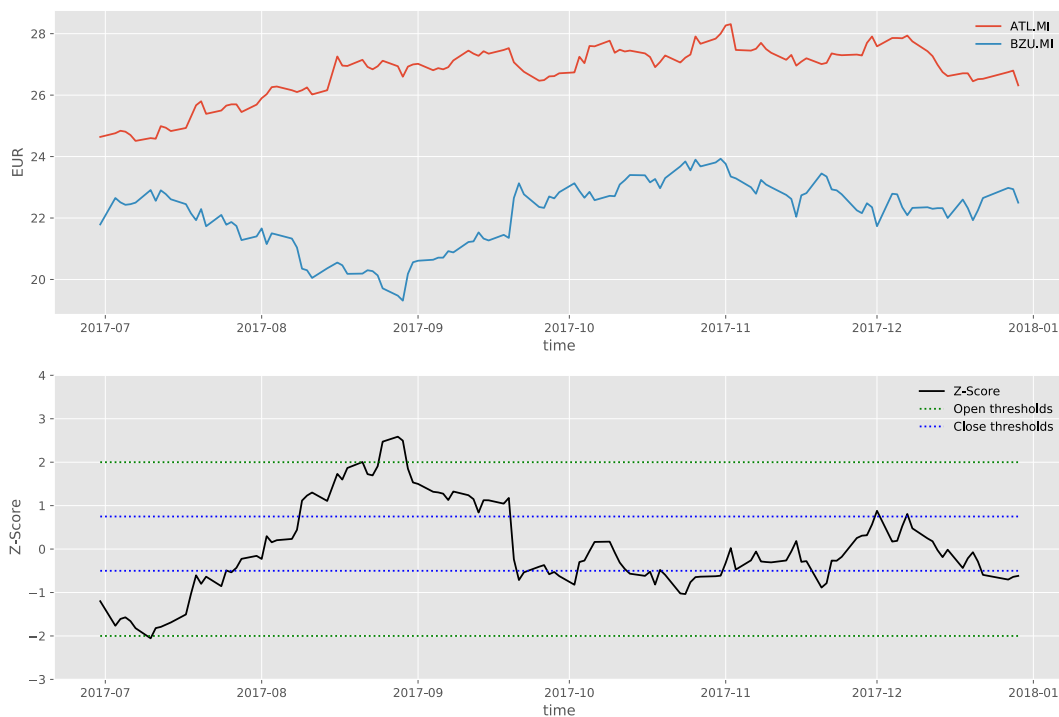
In Graph 1.0, prices (up graph) and Z-Scores (down graph) of the stocks ATL.MI and BZU.MI for the trading period 2017 are displayed. Z-Score's thresholds for open positions are in green, while for short positions are in blue. As it can be seen in the graph downside, the lower open threshold is reached almost immediately in the period (10/07/17) and then the following close threshold is hit in proximity of July (25/07/17). From the graph of the prices, is easy to infer that the strategy spotted an *uptrend* for ATL.MI stock, and a *downtrend* for BZU.MI stock.

It is worth to remember that, when a Z-Score open threshold is hit more than one day, one position per day is opened, therefore adding to the daily returns and, consequently, to the cumulative returns. In the case of ATL.MI-BZU.MI pair, this situation is presented by the end of August 2017: the following table (table 5.0) shows the number of positions opened in the

same period (from 1 until a total of 4 in 29/08/17). All the positions are closed when the close threshold is reached. "Returns of Positions" values are computed as the product of the percentage change of the day and the number of positions opened.

The resulting effect could be considered as using a soft financial leverage.

Table 5.0 Example of trades carried out from 18/08/17 to 21/09/17 for Stock Pair ATL.MI – BZU.MI

Trading 2017 for Stock Pair ATL.MI - BZU.MI

Date	Prices		Percentage Change		N. of Positions	Returns of Positions		Daily Returns	Cumulative Returns
	ATL.MI	BZU.MI	ATL.MI	BZU.MI		Long	Short		
18/08/17	26,95	20,18	-0,0371%	-1,3685%	0	0	0	0	7,761964%
21/08/17	27,15	20,19	0,7421%	0,0496%	1	-0,74211%	0,04956%	-0,692556%	7,069408%
22/08/17	26,92	20,30	-0,8471%	0,5448%	1	0,84714%	0,54482%	1,391962%	8,461370%
23/08/17	26,84	20,27	-0,2972%	-0,1478%	1	0,29718%	-0,14778%	0,149399%	8,610769%
24/08/17	26,94	20,13	0,3726%	-0,6907%	1	-0,37258%	-0,69068%	-1,063262%	7,547507%
25/08/17	27,12	19,71	0,6682%	-2,0864%	2	-1,33631%	-4,17288%	-5,509182%	2,038325%
28/08/17	26,94	19,47	-0,6637%	-1,2177%	3	1,99115%	-3,65296%	-1,661811%	0,376514%
29/08/17	26,60	19,31	-1,2621%	-0,8218%	4	5,04826%	-3,28711%	1,761152%	2,137666%
30/08/17	26,93	20,18	1,2406%	4,5054%	4	-4,96240%	18,02177%	13,059363%	15,197029%
31/08/17	27,00	20,56	0,2599%	1,8830%	4	-1,03973%	7,53219%	6,492465%	21,689494%
01/09/17	27,02	20,61	0,0741%	0,2432%	4	-0,29630%	0,97278%	0,676482%	22,365976%
04/09/17	26,81	20,64	-0,7772%	0,1456%	4	3,10882%	0,58222%	3,691041%	26,057017%
05/09/17	26,88	20,71	0,2611%	0,3391%	4	-1,04438%	1,35658%	0,312201%	26,369219%
06/09/17	26,84	20,71	-0,1488%	0,0000%	4	0,59522%	0,00000%	0,595223%	26,964442%
07/09/17	26,91	20,92	0,2608%	1,0140%	4	-1,04321%	4,05603%	3,012816%	29,977258%
08/09/17	27,13	20,88	0,8175%	-0,1912%	4	-3,27015%	-0,76484%	-4,034985%	25,942273%
11/09/17	27,45	21,22	1,1795%	1,6284%	4	-4,71805%	6,51341%	1,795365%	27,737638%
12/09/17	27,35	21,24	-0,3643%	0,0943%	4	1,45720%	0,37701%	1,834212%	29,571850%
13/09/17	27,28	21,53	-0,2559%	1,3654%	4	1,02376%	5,46141%	6,485172%	36,057022%
14/09/17	27,43	21,33	0,5499%	-0,9289%	4	-2,19941%	-3,71576%	-5,915167%	30,141855%
15/09/17	27,35	21,27	-0,2917%	-0,2813%	4	1,16660%	-1,12517%	0,041439%	30,183294%
18/09/17	27,47	21,45	0,4388%	0,8463%	4	-1,75501%	3,38506%	1,630043%	31,813337%
19/09/17	27,53	21,35	0,2184%	-0,4662%	4	-0,87370%	-1,86481%	-2,738509%	29,074828%
20/09/17	27,07	22,65	-1,6709%	6,0890%	4	6,68363%	24,35596%	31,039589%	60,114417%
21/09/17	26,91	23,13	-0,5911%	2,1192%	0	0,00000%	0,00000%	0,000000%	60,114417%

It is possible to call this increase of stake as leverage because of the cashless nature of the pairs trading strategy deployed: opening a long and a short position of the same size means that the algorithm can place the long order using the money received with the short position. Theoretically speaking the strategy is cashless, but it is worth noting that on a practical ground there are costs such as commissions, spreads and fees that make the strategy not completely cashless.

The reasoning behind the use of this leverage is that the Z-Score reaching the threshold more than one day in the same period can be considered as a strong signal for a profitable trade. In this way, the algorithm is not influenced by the already opened position in the portfolio: in contrast with human psychology, it loads up when the trade is going well and closes the whole cumulative position when the closing threshold is reached. This approach is followed for each of the selected cointegrated pairs in trading periods from 2008 to 2020.

3. Findings

3.1 Results of the Strategy

3.1.1 Annual and Excess Returns

The results obtained by the pairs trading strategy for period 2008-2020 suggests a positive performance, with positive excess returns for 12 out of 13 years of trading.

Table 6.0 Annual Returns, Risk-Free Rate and Excess Return of Period 2008-2020

Year	Annual Returns	Risk-Free Rate	Excess Return
2008	33,79%	5,10%	28,69%
2009	38,30%	4,34%	33,96%
2010	0,07%	4,01%	-3,94%
2011	29,90%	5,68%	24,22%
2012	48,45%	6,05%	42,40%
2013	7,31%	4,39%	2,92%
2014	34,69%	2,69%	32,00%
2015	35,08%	1,93%	33,15%
2016	93,72%	1,18%	92,54%
2017	27,06%	2,17%	24,89%
2018	33,74%	1,61%	32,13%
2019	11,93%	1,11%	10,82%
2020	33,75%	0,72%	33,03%

As mentioned earlier, risk-free rates are the value of *Italy 10-year Government Bond* as of July of each year.

Annual compounded return is computed considering the result of the previous period invested again the subsequent period. As in the table below, each euro invested in 2008 could have brought € 34.36 in 2020 (without considering trading fees and taxes on capital gain).

Table 7.0 Annual Growth of €1.00 in Pairs Trading Strategy in period 2008-2020

Annual Compounded Returns

2007	100,00%	1,00 €
2008	133,79%	1,34 €
2009	185,03%	1,85 €
2010	185,16%	1,85 €
2011	240,52%	2,41 €
2012	357,06%	3,57 €
2013	383,16%	3,83 €
2014	516,08%	5,16 €
2015	697,12%	6,97 €
2016	1350,46%	13,50 €
2017	1715,89%	17,16 €
2018	2294,83%	22,95 €
2019	2568,60%	25,69 €
2020	3435,51%	34,36 €

3.1.2 Risk Metrics

The pairs trading strategy developed shows good metrics of risk, with an annualized volatility of 22.55% and a standard deviation from excess returns of 0.2313.

Sharpe ratio is calculated as follow:

$$Sharpe\ Ratio = \frac{Average\ Excess\ Returns}{Standard\ Deviation\ of\ Excess\ Returns}$$

The Sharpe Ratio measures the excess return obtained by holding a riskier asset rather than the safe one, after adjusting for its risk. The value obtained is 1.29 and it can be considered as a good value.

The maximum daily drawdown is -11.47%. This is a huge loss for a single day of trading, and possible optimization are discussed in the next paragraph. Standard deviation is computed with daily returns; the value is 0.0165.

Table 8.0 Metrics for Risk and Return's Assessment

Average Annual Return	32,91%
Average Daily Return	0,26%
STD of Excess Returns	0,2313
Sharpe Ratio	1,2863
Largest Daily Return	20,02%
Lowest Daily Return	-11,47%
Standard Deviation	1,65%
Annualized Volatility	22,55%

Standard deviation (σ) and volatility (v) are computed for each year of the period. The results are shown in Table 9.0. Annualized volatility is 22.55%, a substantially good value given the Average Annual Return of 32.91%.

Volatility is calculated as follow:

$$v = \sqrt{252} \cdot \sigma$$

Table 9.0 Standard Deviation and Volatility for period 2008-2020

|	Standard Deviation	Annual Volatility
2008	0,0237	0,3757
2009	0,0203	0,3220
2010	0,0060	0,0946
2011	0,0083	0,1312
2012	0,0317	0,5025
2013	0,0057	0,0913
2014	0,0108	0,1707
2015	0,0119	0,1883
2016	0,0267	0,4236
2017	0,0080	0,1263
2018	0,0123	0,1954
2019	0,0060	0,0952
2020	0,0136	0,2154

3.1.3 Index Correlation

Inspecting the correlation with FTSE MIB daily returns, the average value is 0.070, suggesting a non-existing correlation between the strategy and the index.

Table 10.0 Pairs Trading Strategy vs FTSE MIB

| Year	Portfolio Strategy		FTSE MIB		
	Average Daily Returns	Annual Returns	Average Daily Return	Annual Return	Correlation
2008	0,27%	33,79%	-0,26%	-33,26%	0,166
2009	0,30%	38,30%	0,17%	21,73%	0,041
2010	0,00%	0,07%	0,06%	7,24%	0,069
2011	0,23%	29,90%	-0,19%	-24,42%	0,189
2012	0,38%	48,45%	0,12%	14,87%	-0,032
2013	0,06%	7,31%	0,18%	21,96%	0,015
2014	0,28%	34,69%	-0,09%	-11,47%	0,016
2015	0,28%	35,08%	-0,02%	-3,05%	0,001
2016	0,73%	93,72%	0,14%	18,26%	0,220
2017	0,21%	27,06%	0,05%	6,31%	0,105
2018	0,27%	33,74%	-0,12%	-14,77%	0,004
2019	0,10%	11,93%	0,08%	10,02%	0,207
2020	0,26%	33,75%	0,12%	15,17%	-0,088

From Table 10.0 it is possible to infer that the annual returns of the portfolio strategy are independent from the ones of the index, with a maximum correlation of 0.22 reached in 2016, the best year of the pairs trading strategy in terms of annual returns (93.72%).

Graph 2.0 Pairs Trading Strategy and FTSE MIB Correlation
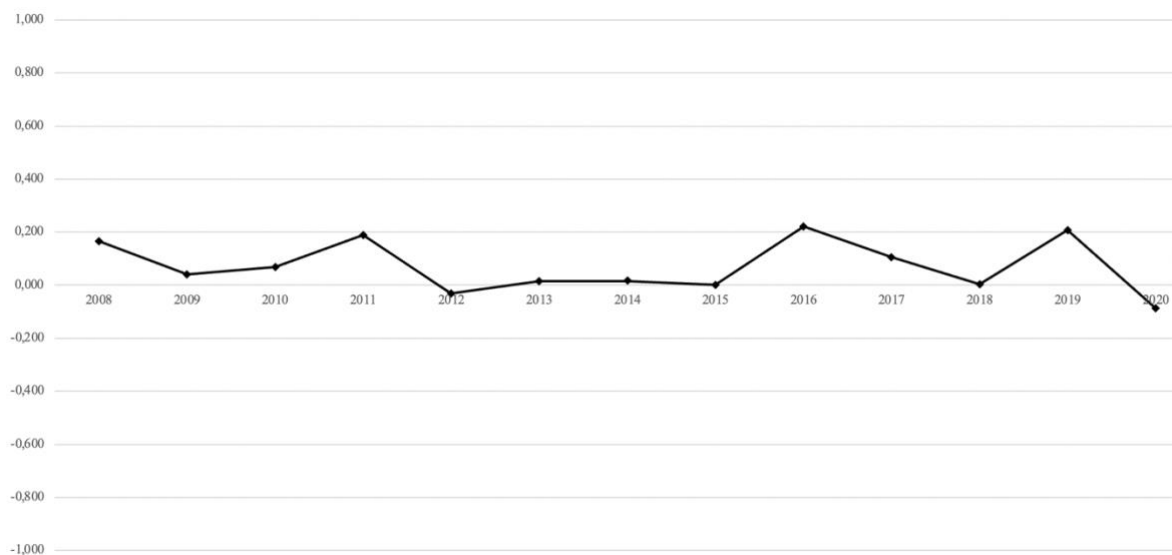
The annual returns of the algorithm's pairs trading portfolio strategy consistently beat the FTSE MIB annual returns every year except for 2010 and 2013, as shown in Graph 3.0

Graph 3.0 Pairs Trading Strategy and FTSE MIB Annual Returns
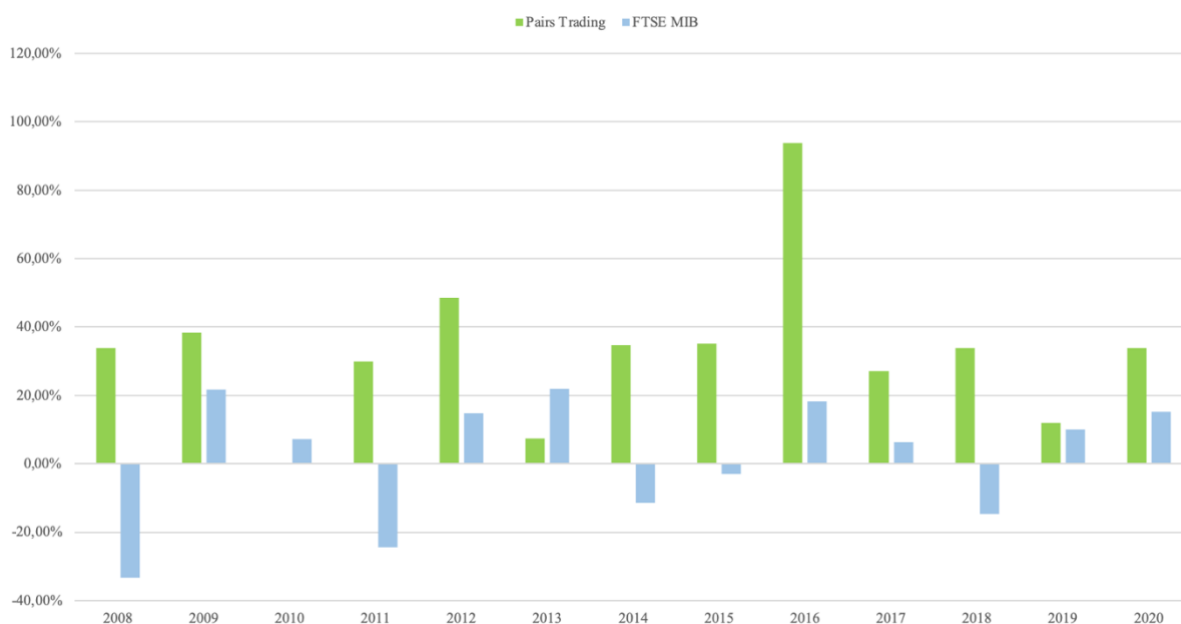
3.2 Further Developments

Overall, findings seem to confirm that pairs trading strategy is a valid approach to equity trading. Nonetheless, it must be considered that the algorithm developed has opportunity of improvement and refinement that are key of better results in terms of risks and returns.

3.2.1 Take-profit and Stop-loss

The algorithm was not developed to consider take-profit and stop-loss limits for the trades. The latter is of great value to cut the losses before they become too big to be recovered, but at the same time an overconservative stop-loss could close profitable trades that show a difficult start. Next, it should be noted that each pair traded in the same period employs a portion of the capital allocated to the strategy, in case of the algorithm the capital is evenly distributed. Stop-loss could be set at pairs level or at portfolio level, considering the capital allocated to each pair or the overall capital of the portfolio.

3.2.2 Portfolio Weights

As mentioned above, capital could be distributed in different ways to each pair. For example, a test has been carried out to distribute more capital to the pairs that show better cointegration results. The weights were based on ranking length: if the cointegrated pairs are 10, then a weight of 10 is given to the first pair, a weight of 9 to the second pair and so on. The result of such a capital distribution is slightly better in terms of average annual returns (+0.86%), but it increases the standard deviation of excess returns resulting in a decrease of the Sharpe ratio.

Table 11.0 Excess Returns and Sharpe ratio of Portfolio with incremental weights on pairs

Year	Annual Return	Risk-Free Rate	Excess Return
2008	0,225294679	5,096%	17,433%
2009	0,284385057	4,342%	24,097%
2010	0,056983104	4,007%	1,691%
2011	0,2339063	5,676%	17,715%
2012	0,645896568	6,050%	58,540%
2013	0,09730291	4,391%	5,339%
2014	0,332174283	2,686%	30,531%
2015	0,359737445	1,926%	34,048%
2016	1,123971289	1,183%	111,214%
2017	0,249176025	2,169%	22,749%
2018	0,314344503	1,612%	29,822%
2019	0,132981216	1,106%	12,192%
2020	0,333877268	0,717%	32,671%

STD of Excess Returns	0,2821
Sharpe Ratio	1,0852

3.2.3 Pairs Composition

Pairs traded are selected over a 6-months training period in which cointegration tests are computed. Better results are possibly reached by reducing/increasing the training period. A test was carried out with a 3-months training period and considerably better results were reached.

Table 11.0 and Annual Retur and Compounded Return and number of Pairs Traded with a 3-months training period

Year	Annual Return	Pairs Traded	Compounded Annual Return
2008	0,4082	12	1,4082
2009	0,7189	2	2,4206
2010	0,2997	6	3,1460
2011	0,2825	9	4,0347
2012	0,2755	1	5,1463
2013	0,0000	0	5,1463
2014	0,2920	11	6,6490
2015	0,4104	5	9,3778
2016	0,5423	17	14,4634
2017	0,6081	10	23,2585
2018	0,4664	20	34,1063
2019	0,5346	5	52,3396
2020	1,0394	3	106,7414
Mean	0,4522	7,77	

Interesting to note that reducing the training period by half leads to an average number of pairs traded of 7.77, not so different from the previous average of 8.62 pairs. The average annual return increases considerably to 45.22%.

Similar tests could be carried out, without being caught in *statistical overfitting*, to find the length of the training period that maximize both risks and returns.

3.2.4 Redundant Trades

The algorithm does not optimize the trades it opens across pairs. An example must be provided to understand the concept of this kind of optimization:

Consider two stock pairs, the first one composed by stock *A* and stock *B*, while the second one composed by stock *B* and stock *C*. At the same point in time, the signals suggest buying the first pair (i.e., sell stock *A* and buy stock *B*) and also the second pair (i.e., sell stock *B* and buy stock *C*). In this case, a redundancy should be spotted by the algorithm, therefore avoiding the positions on stock *B* and opening only the ones for stocks *A* and *C*.

The effect of such an optimization is not consistent with the study presented since the result of it would be fewer costs of transactions, here not considered.

3.2.5 Fees and Short Selling

Of particular interest would be to apply fees to the trades opened. Approximately an average of 38 trades are opened each trading period, the lowest number in 2019 with 2 trades and the highest in 2014 with 104 trades, suggesting that the cost of such a strategy would not be game changing.

Same conclusions could be draw for short selling, with a different consideration to be made: the algorithm does not take into account the possibility that a pair in a stock could not be sold short, a condition that can be found in the market. Moreover, the cost of selling a stock short could be as high as to make the trade result meaningless or even negative.

3.2.6 Assessment Tests

Advanced assessment tests can further inspect the statistical soundness of the results reached. Having followed Caldeira *et al* and reached to similar conclusions, the tests should positively

confirm that the algorithm is unbiased. Withe's Reality Check and Hansen SPA's Test are the two tests the academics have applied to their findings.

3.2.7 Daily Compounding and Leverage

The cashless state of pairs trading gives traders the possibility to leverage their positions heavily. While considering leverage in the form of multiple positions opened for the same pair (see Table 5.0), the algorithm could deploy higher level of leverage based on different parameters, both technical and fundamental (price oscillation, companies belonging to the same industry…).

Moreover, daily compounding is not considered by the algorithm, which invest the same sum of capital for each trade. It could be possible to pile up the gains of each trade and reinvest them, amplifying the returns of the strategy.

4. Conclusions

The algorithm developed works on the 40 stocks composing the FTSE MIB, the major Italian Stock Exchange index, organized in pairs. It is then able to process advanced cointegration tests during a 6-month training test and select the pairs that show cointegrated behaviors. It proceeds by trading the pairs based on Z-Score for the remaining half of the year. Such a process is deployed for each of the years in the range 2008-2020, resulting in an average annual return of 32.91% and annualized volatility of 22.55%, with a Sharpe ratio of 1.29 and a promising market correlation of 0.07.

As discussed in the preceding paragraph, the algorithm provides plenty of room for improvements, both in the training and the trading periods. It has been entirely developed in Python, partly following *Caldeira et al* paper on pairs trading over Brazilian stock market.

Based on the findings here illustrated, pairs trading can still be seen as a valid strategy, despite the falls of early adopters like LTCM and Morgan Stanley's arbitrageurs. Hedge funds still rely upon pairs trading, with more accurate and sophisticated algorithms able to consider substantial amount of heterogenous data and to produce higher returns and lower risks.

Possible developments of the algorithm presented can further optimize results and prepare it for a deployment on the market.

References

I. Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/arbitrage/

II. Lazzarino, M., Berrill, j. And Šević, A. (2018) What Is Statistical Arbitrage? *Theoretical Economics Letters*, 8, 888-908. https://doi.org/10.4236/tel.2018.85063

III. A. Pole, Statistical Arbitrage: Algorithmic Trading Insights and Techniques - 2007. Wiley Finance

IV. R. Lowenstein, When Genius Failed: The Rise and Fall of Long-Term Capital Management – 2000. Random House

V. Duarte, J., Longstaff, F.A. and Yu, F. (2006) Risk and Return in Fixed Income Arbitrage: Nickerls in Front of a Steamroller? The Review of Financial Studies

VI. Krishnamurthy, A. (2008) Fundamental Value and Limits to Arbitrage

VII. https://www.marketwatch.com/investing/bond/tmbmkit-10y?countrycode=bx

VIII. https://www.statisticshowto.com/engle-granger-test/

IX. Engle and Granger (1987) Cointegration and Error Correction: Representation, Estimation and Testing.

X. https://machinelearningmastery.com/time-series-data-stationary-python/

Appendix

Appendix 1.0 Transcription of the algorithm developed on Python

```python
import yfinance as yf
import pandas as pd
from statsmodels.tsa.stattools import coint
import itertools
import statsmodels.api as sm
import numpy as np
from numpy import arange
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.stattools import adfuller
from matplotlib.backends.backend_pdf import PdfPages
plt.style.use('ggplot')

pd.set_option("display.max_rows", None, "display.max_columns", None)

means = pd.DataFrame()

# this is the dataframe in which I have to store the cumulative results for everyday

final_returns = pd.DataFrame(columns=['cumulative_returns'], dtype='float')

tickers = ["A2A.MI", "AMP.MI", "ATL.MI", "AZM.MI", "BGN.MI", "BAMI.MI",
        "BMED.MI", "BPE.MI",
        "BZU.MI", "CPR.MI", "CNHI.MI", "DIA.MI", "ENEL.MI",
        "ENI.MI", "EXO.MI", "RACE.MI", "FBK.MI", "G.MI", "HER.MI",
        "IP.MI", "ISP.MI", "INW.MI", "IG.MI", "LDO.MI", "MB.MI",
        "MONC.MI", "NEXI.MI", "PIRC.MI", "PST.MI", "PRY.MI", "REC.MI",
        "SPM.MI", "SRG.MI", "STLA.MI", "STM.MI", "TIT.MI", "TEN.MI",
        "TRN.MI", "UCG.MI", "UNI.MI"]

for year in np.arange(2007, 2021):

    yearfig, yearaxis = plt.subplots(1, figsize=(15, 10))

    # total_returns is the yearly database in which I have the returns for
    # every pair traded. I use it to compute the mean return of the year

    total_returns = pd.DataFrame()

    total_daily_returns = pd.DataFrame()

    start = str(year)+"-01-01"
    end = str(year+1)+"-01-01"

    stocks = yf.download(tickers, start=start, end=end, show_errors=False)

    stocks.columns = stocks.columns.to_flat_index()
    stocks.columns = pd.MultiIndex.from_tuples(stocks.columns)
    stocks.swaplevel(axis=1).sort_index(axis=1)

    close = stocks.loc[:, "Close"].copy()

    # close['Date'] = pd.to_datetime(close['Date'])
```

```python
# close = close.set_index('Date')

close = pd.DataFrame(data=close, index=pd.DatetimeIndex(close.index))

# select the stocks with not many nan values, resulting in a total of 30 stocks
# close = close.dropna(thresh=len(close) - 2000, axis=1)

# create return dataframe

# create empty dataframe to store cointegration results for pairs
coint_results = pd.DataFrame(index=["Pair"])
coint_results_ADF = pd.DataFrame(index=["Pair"])


# delete columns and rows with nan

window = 126

long_open = -2.0  # t < -2
long_close = -0.5  # t < -0.5
short_open = 2.0  # t > 2
short_close = 0.75  # t < 0.75

close_train_period = close[0:window].dropna(
    axis="columns", how="any")
close_test_period = close[window:].dropna(axis="columns", how="all")
# create list with all possible pairs
pairs = close_train_period.columns[1:]
combinations = list(itertools.combinations(pairs, 2))

# if t-stat (egtest[0]) is greater then 1% confidence interval (egtest[2][0])
# I can reject the null meaning that the pair is cointegrated

for pair in range(len(combinations)):
    X = close_train_period.loc[:, combinations[pair][0]]
    y = close_train_period.loc[:, combinations[pair][1]]
    egtest = coint(y, X, trend='ct', method='aeg', autolag='AIC')
    egoutput = pd.Series(egtest[0:3], index=[
                'Test Statistic', 'p-value',
                'Critical Values 1%,5%,10%'])
    if egtest[0] < egtest[2][1]:
        reg = sm.OLS(y, sm.add_constant(X))
        rs = reg.fit()
        a = rs.params[0]
        b = rs.params[1]
        results = np.array(
            [combinations[pair][0], combinations[pair][1], a, b, egtest[0]])
        # rank by last column (t-stat) ascending order
        coint_results = coint_results.append(pd.DataFrame(results).T)
    else:
        pass

coint_results = coint_results.reset_index().dropna().drop("index", axis=1)

if not coint_results.empty:

    coint_results.columns = ['stock_1', 'stock_2', 'a', 'b', 't_stat']
    coint_results['a'] = pd.to_numeric(coint_results['a'])
    coint_results['b'] = pd.to_numeric(coint_results['b'])
    coint_results['t_stat'] = pd.to_numeric(coint_results['t_stat'])
    coint_results.sort_values(
        't_stat', ascending=True, inplace=True, ignore_index=True)
```

```
    for pair in range(0, len(coint_results)):
        stock1 = coint_results.loc[pair][0]
        stock2 = coint_results.loc[pair][1]
        a = coint_results.loc[pair][2]
        b = coint_results.loc[pair][3]
        t_stat = coint_results.loc[pair][4]
        spread = close_train_period[stock1] - a*close_train_period[stock2]
        adf = adfuller(spread)

        if adf[0] < adf[4].get('5%'):
            coint_results_ADF = coint_results_ADF.append(
                coint_results.loc[pair], ignore_index=True).dropna()

        if len(coint_results_ADF) > 20:
            coint_results_ADF = coint_results_ADF[:20].dropna()

# until now, I performed the Engle-Granger test and the ADF test to
# pick the stock pairs (max 20) that are most reliable to the test-period

if not coint_results_ADF.empty:

    for pair in range(0, len(coint_results_ADF)):

        stock1_ADF = coint_results_ADF.loc[pair][0]
        stock2_ADF = coint_results_ADF.loc[pair][1]

        def zscore(series):
            return(series - series.mean())/np.std(series)

        signals = pd.DataFrame()
        signals['stock_1'] = close_test_period[stock1_ADF]
        signals['stock_2'] = close_test_period[stock2_ADF]
        ratios = signals.stock_1 / signals.stock_2

        # calculate z-score
        # create signals dataframe
        # it renders 1 on signals' columns when z hits thresholds
        # it calculates positions as cumulative sums of hitted tresholds
        # it renders no. of closed positions when thresholds are met

        signals['z'] = zscore(ratios)
        signals['z up open'] = 2.0
        signals['z down open'] = -2.0
        signals['z up close'] = 0.75
        signals['z down close'] = -0.5

        signals['open_sig_1'] = 0
        signals['close_sig_1'] = 0
        signals['position_1'] = 0
        signals['close_1'] = 0

        signals['open_sig_2'] = 0
        signals['close_sig_2'] = 0
        signals['position_2'] = 0
        signals['close_2'] = 0

        signals['open_sig_1'] = np.select(
            [signals['z'] > signals['z up open']], [1])
        signals['position_1'] = (signals['position_1'].cumsum()
                    + signals['open_sig_1'].cumsum()).copy()

        for t in range(0, len(close_test_period)):
            if signals['z'][t] < signals['z up close'][t]:
```

```python
        if signals['position_1'][t] > 0:
            signals.close_1[t] = signals.position_1[t].copy()
            for t2 in range(t, len(close_test_period)-1):
                signals.position_1[t2+1] = 0

    signals['open_sig_2'] = np.select(
        [signals['z'] < signals['z down open']], [1])
    signals['position_2'] = (signals['position_2'].cumsum()
                    + signals['open_sig_2'].cumsum()).copy()

    # print("number of trades in "+str(year))
    # print(signals['open_sig_1'].astype(bool).sum(
    #     axis=0) + signals['open_sig_2'].astype(bool).sum(axis=0))

    for t in range(0, len(close_test_period)):

        if signals['z'][t] > signals['z down close'][t]:
            if signals.position_2[t] > 0:
                signals.close_2[t] = signals.position_2[t].copy()
                for t2 in range(t, len(close_test_period)-1):
                    signals.position_2[t2+1] = 0

    fig, plots = plt.subplots(2, figsize=(15, 10))
    plots[0].plot(signals.index, signals["stock_1"])
    plots[0].plot(signals.index, signals["stock_2"])
    fig.suptitle('Price and Z-Score of Pair '+stock1_ADF
            + '-'+stock2_ADF+' at year '+str(year))
    plots[0].set_xlabel('time')
    plots[0].set_ylabel('EUR')
    plots[0].legend([stock1_ADF, stock2_ADF])
    plots[1].plot(signals.index, signals['z'], color='black')
    plots[1].plot(signals.index, signals['z up open'],
            color='green', ls=':')
    plots[1].plot(signals.index, signals['z up close'],
            color='blue', ls=':')
    plots[1].plot(signals.index, signals['z down open'],
            color='green', ls=':')
    plots[1].plot(signals.index, signals['z down close'],
            color='blue', ls=':')
    plots[1].set_ylim(-3, 4)
    plots[1].set_xlabel('time')
    plots[1].set_ylabel('Z-Score')
    plots[1].legend(['Z-Score', 'Open thresholds', 'Close thresholds'])

    # fig.savefig('priceandzscore-'+str(year)+'-'
    #         + str(stock1_ADF)+'-'+str(stock2_ADF)+'.pdf')

    returns_pair = pd.DataFrame()

    returns_pair['stock_1'] = signals['stock_1']
    returns_pair['stock_2'] = signals['stock_2']
    returns_pair['returns_stock1'] = returns_pair['stock_1'].pct_change()
    returns_pair['returns_stock2'] = returns_pair['stock_2'].pct_change()
    returns_pair['daily_return_s1'] = - signals['position_1'] * \
        returns_pair['returns_stock1'] + \
        signals['position_2'] * returns_pair['returns_stock1']
    returns_pair['daily_return_s2'] = signals['position_1'] * \
        returns_pair['returns_stock2'] - \
        signals['position_2'] * returns_pair['returns_stock2']

    returns_pair['cum_daily_return'] = returns_pair['daily_return_s1'] + \
        returns_pair['daily_return_s2']
    returns_pair['cumulative_returns'] = returns_pair['cum_daily_return'].cumsum()
```

```python
    yearaxis.plot(
        signals.index, returns_pair['cumulative_returns'], label=stock1_ADF+'-'+stock2_ADF)
    yearaxis.legend(bbox_to_anchor=(1.05, 1), loc='upper left', ncol=1)

    #returns_pair.to_csv('returns_pair-'+str(year)+'-'
    #        + str(stock1_ADF)+'-'+str(stock2_ADF)+'.csv')

    total_returns = total_returns.append(
        returns_pair.iloc[-1])

    daily_returns = returns_pair.cum_daily_return
    daily_returns = daily_returns[1:]

    if len(total_daily_returns) == 0:

        total_daily_returns = daily_returns.copy()
        total_daily_returns = total_daily_returns.div(
            len(coint_results_ADF))
        total_daily_returns = pd.DataFrame(
            {'Date': total_daily_returns.index, 'cum_daily_return': total_daily_returns.values})

    else:

        total_daily_returns["cum_daily_return"] = total_daily_returns["cum_daily_return"] + \
            (daily_returns.div(len(coint_results_ADF))).values

# plt.close('all')
# yearfig.tight_layout()
# yearfig.savefig('trades_return-'+str(year)+'.pdf')

if len(final_returns) == 0:
    total_daily_returns.set_index(
        total_daily_returns['Date'], inplace=True)
    total_daily_returns = total_daily_returns.drop(columns=[
                                'Date'])
    final_returns = total_daily_returns.copy()

else:
    total_daily_returns.set_index(
        total_daily_returns['Date'], inplace=True)
    total_daily_returns = total_daily_returns.drop(columns=[
                                'Date'])
    final_returns = final_returns.append(total_daily_returns)

final_returns = final_returns.round(4)

    # printing in pdf

    '''

    pd.set_option("display.max_rows", None,
            "display.max_columns", None)

    from matplotlib.backends.backend_pdf import PdfPages

    fig, ax = plt.subplots(figsize=(12, 4))
    ax.axis('tight')
    ax.axis('off')
    the_table = ax.table(cellText=signals.values,
                colLabels=signals.columns, loc='center')

    pp = PdfPages("signals-"+str(pair)+".pdf")
```

```
        pp.savefig(fig, bbox_inches='tight')
        pp.close()

        fig, ax = plt.subplots(figsize=(12, 4))
        ax.axis('tight')
        ax.axis('off')
        the_table = ax.table(cellText=returns.values,
                    colLabels=returns.columns, loc='center')

        pp = PdfPages("returns-"+str(pair)+".pdf")
        pp.savefig(fig, bbox_inches='tight')
        pp.close()

        '''

if not coint_results_ADF.empty:

    total_returns = total_returns.drop(
            columns=['stock_1', 'stock_2', 'returns_stock1', 'returns_stock2',
                'daily_return_s1', 'daily_return_s2', 'cum_daily_return'])

    tot_ret_mean = total_returns['cumulative_returns'].mean()
    # print(total_returns)
    total_returns.loc['mean'] = tot_ret_mean
    means = means.append(total_returns.loc['mean'])
    means = np.around(means, decimals=4)

    # print(means)

# final_returns['cum_daily_return'] = final_returns['cum_daily_return'].round(6)
# final_returns.to_csv("final_returns.csv")
```