Norwegian School of Economics
Bergen, Spring 2018

# Statistical Arbitrage Trading with Implementation of Machine Learning

An empirical analysis of pairs trading on the Norwegian stock market

Håkon Andersen & Håkon Tronvoll

Supervisor: Tore Leite

Master Thesis in Financial Economics

**Norwegian School of Economics**

## Abstract

The main objective of this thesis is to analyze whether there are arbitrage opportunities on the Norwegian stock market. Moreover, this thesis examines statistical arbitrage through cointegration pairs trading. We embed an analytic framework of an algorithmic trading model which includes principal component analysis and density-based clustering in order to extract and cluster common underlying risk factors of stock returns. From the results obtained we statistically prove that pairs trading on the Oslo Stock Exchange Benchmark Index does not provide excess return nor favorable Sharpe ratio. Predictions from our trading model are also compared with an unrestricted model to determine appropriate stock filtering tools, where we find that unsupervised machine learning techniques have properties which are beneficial for pairs trading.

## Acknowledgements

We would like to direct our appreciation towards our supervisor, Prof. Tore Leite, for valuable input and guidance throughout this process. Also, we would like show gratitude towards Bård Tronvoll for his reflections and insights.

Last, we would like to thank all our friends at NHH who have been supportive throughout the years.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Background

Has someone ever told you that *there is no such thing as a free lunch*? This is a knowledgeable proverb expressing the idea that it is impossible to get something for nothing. In mathematical finance, the term is used to describe the principle of no-arbitrage, which states that it is not possible to make excess profit without taking risk and without a net investment of capital (Poitrast, 2010). The recent development of modern technological novelties has aided the idea of no-arbitrage because it has disrupted the economic infrastructure and the working conditions in the financial markets. Two important interrelated technological shifts have been crucial to this progress. First, advanced computer technology has enabled investors to automate their trading through sophisticated trading algorithms. Second, stock exchanges have structured themselves in a completely computerized way which makes access to capital easier than ever (Avolio et al., 2002). These are all contributions which have led to increased market efficiency and transparency and, one can wonder if the progress has come so far that it has created a perfectly efficient market where the idea of no-arbitrage is, in fact, a striking reality?

Although this may be the new outlook, market inefficiencies may arise every now and then. The problem is just to identify such rare situations before they disappear. On these premises, financial institutions are now allocating huge resources to develop trading algorithms which

can locate such scarce market inefficiencies. As a consequence, learning to understand these algorithms can give you the skills to leverage market information in a way that disproves the notion of no-arbitrage. Therefore, by understanding and analyzing the intersection between trading algorithms and mathematical finance, *might there be a free lunch after all?*.

## 1.2 The Aim of this Thesis

In the attempt to investigate if there is a "free lunch" in the market, we want to identify stocks which have deviated from its relatively fundamental value. Moreover, we seek to develop an algorithmic trading model capitalizing on the notion of statistical arbitrage. This will be executed through the trading strategy known as pairs trading.

Pairs trading is a relative value statistical arbitrage strategy that takes a position in the spread between two stocks which prices have historically moved in tandem (Gatev et al., 2006). More specifically, one enters a long position in one stock and a short position in the other. Positions are executed simultaneously. The spread between the two stocks forms a stationary process, where trading signals are based on deviations from the long-term equilibrium spread. If the spread deviates from its historical equilibrium, we act and capitalize on the temporary inconsistency in the belief it will revert back in the nearest future. Since stock prices are assumed to follow a stochastic process, the strategy only needs to account for the relative price relationship between the stocks. This implies that the long position is entered in the understanding that the stock is relatively undervalued compared to the other and, must, therefore, be balanced by a short position (Harlacher, 2016). Hence, pairs trading gives no indications that stocks are mispriced in absolute terms because it bets on the relative relationship between two stocks to be mean-reverting (Harlacher, 2016).

In the last decades, various scholars have demonstrated unique methods of constructing a pairs trading strategy with declining profitability in recent time mainly due to improved technology. Nevertheless, the vast majority of the documented studies have been carried out in the US equity markets, thus leaving us with a scarce amount of research outside the region. This academic

ambiguity has further triggered our interest to undertake a study aiming to acquire knowledge on the profitability of such a strategy in Norway. To scrutinize the search of statistical arbitrage, we implement techniques of unsupervised machine learning to effectively filter stocks. The Oslo Stock Exchange Benchmark Index is the considered stock universe and has been used to measure the performance from the period of January 2013 to December 2017. Thus, to see if algorithmic pairs trading is profitable in Norway, we impose the following research question:

***Can algorithmic pairs trading with machine learning generate excess return?***

In the pursuit of the research question, we have constructed our research design as a four stage process. The interrelated stages are i) data management, ii) stock filtering, iii) identifying mean-reversion, and iv) trading setup and execution. Below is an outline of the study of these stages.

The first stage of the research design encompasses the process of data sampling and management, where we implement stock return and fundamental ratios. For the stock filtering process, we embed three different methods of unsupervised machine learning techniques to find natural clusters of common underlying risk factors. In the third stage, clustered stocks are tested for cointegration as a mean to identify mean-reversion. For the last, we instrument trading signals and thresholds for buy and sell.

## 1.3 Structure of Thesis

The thesis is organized as follows: Chapter 2 outlines the theoretical frameworks for this thesis. In Chapter 3, we present our research design and the methodology used. In Chapter 4, we present the empirical results obtained. From there, Chapter 5 discusses the results in light of the theoretical frameworks. Last, in Chapter 6, we conclude our findings.

# Chapter 2

# Theoretical Frameworks

## 2.1 The Efficient-Market Hypothesis

The "free lunch" principle is supported by the Efficient-Market Hypothesis (EMH), which states that transactions in an efficient market are at its correct value because all information is reflected in the price (Fama, 1970). The theory thus provides a theoretical indication of the outcome of our research question because one cannot exploit mispriced stocks when there are none. Moreover, Jensen (1978) contributes to the definition by defining market efficiency as,

*"A market is said to be efficient if it is impossible to make a profit by trading on a set of information, $\Omega_t$"*

From the definition, as outlined by Jensen (1978) and Fama (1970), market efficiency relies on, i) the information set adapted, $\Omega_t$, and ii) the ability to exploit this information. The former criteria postulate that market efficiency exists in three various forms on the basis of the information set, namely the weak, semi-strong and strong. If $\Omega_t$ only contains past information, the EMH is at its weak form (Timmermann and Granger, 2004). The semi-strong form indicates that the information set includes all past and present public information. Such information includes fundamental data such as product line, quality of management, balance sheet composition and earning forecasts (Bodie et al., 2014). Last, if $\Omega_t$ contains both public and private

information, the EMH is in its strong form (Malkiel, 2005). For a more detailed description of all forms of market efficiency, see Appendix 7.1.

For the second criteria, the EMH implies the idea of no-arbitrage. A situation where an investor is not able to obtain excess return from the information set because it reflects all relevant information. Moreover, Fama (1970) elevates the concept as,

$$E(\tilde{P}_{j,t+1} \mid \Omega_t) = [1 + E(\tilde{R}_{j,t+1} \mid \Omega_t)]P_{jt} \tag{2.1}$$

where E is the expected value, $P_{jt}$ is the price of stock $j$ at time $t$. $R_{j,t+1}$ is the stock return defined by $(P_{j,t+1} - P_{jt})/(P_{jt})$. The tildes indicates that $\tilde{P}_{j,t+1}$ and $\tilde{R}_{j,t+1}$ are random variables in the given time, conditional on the information set $\Omega_t$. Equation (2.1) simply states that the expected stock price at time $t+1$ is a function of the expected return and the price at time $t$. Fama (1970) then argues that this has a major empirical impact because it rules out any possibility to expect excess returns. Because of this, the expected value should, therefore, reflect the actual value and we can define,

$$Z_{j,t+1} = R_{j,t+1} - E(\tilde{R}_{j,t+1} \mid \Omega_t) \tag{2.2}$$

then,

$$E(\tilde{Z}_{j,t+1} \mid \Omega_t) = 0 \tag{2.3}$$

where equation (2.3) is the excess market return of the stock at $t+1$. Moreover, it is the difference between the observed return and the expected return with the property that the excess value is zero. Fama (1970) describes this condition as a *fair game*, an uncertain situation in which the differences between expected and actual outcomes show no systematic relations (Law and Smullen, 2008). If we describe $\alpha(\Omega_t) = [\alpha_1(\Omega_t) + \alpha_2(\Omega_t) + ... + \alpha_n(\Omega_t)]$ as the amount of capital invested in each of the $n$ available stocks based on the information $\Omega_t$, Fama (1970)

now argues that the excess market value at time $t+1$ is,

$$E(\tilde{V}_{j,t+1}) = \sum_{j=1}^{n} \alpha_j(\Omega_t) E(\tilde{Z}_{j,t+1} \mid \Omega_t) = 0 \tag{2.4}$$

where the total excess market value is a fair game with a value of zero. This is because rational investors are able to gather relevant information and thus, investors do not have any comparative advantage. Hence, the expected value thus equals the observed value, meaning that *there is no such thing as a free lunch.* As aforementioned, Fama (1970) outlines that stock prices fully reflect all information. Some information is expected and some unexpected. The unexpected portion of this information arrives randomly, and the stock price is adjusted according to this new information. Fama (1970) outlines this as a random walk. He describes successive price changes to be independent and identically distributed such that,

$$P_{jt} = P_{j,t-1} + \mu_t \tag{2.5}$$

where $\mu_t$ is a white noise process with a mean of zero and variance $\sigma^2$. This means, that under the rubric of the EMH, $P_{jt}$ is said to be marginal because the best forecast of all values $P_{j,t+1}$ is the current price $P_t$ (Pesaran, 2010). The random walk model then becomes,

$$f(R_{j,t+1} \mid \Omega_t) = f(R_{j,t+1}) \tag{2.6}$$

with $f$ indicating the probability density function. The model of the random walk is an important concept for our research because it states that we cannot determine the precise return in advance. Over the long run, stock returns are consistent with what we expect, given their level of risk. However, in the short-run, fluctuations can affect the long-run prospect (Fama, 1970).

Nevertheless, there is empirical evidence that sheds doubts about the efficiency of markets and the unpredictability of stock prices. Moreover, various scholars ample evidence of excess return predictably caused by a long-term equilibrium between the relative prices of two financial time series. Any deviations from this relative equilibrium state are coming from a temporary shock

or reaction from the market and thus, creating arbitrage opportunities (Bogomolov, 2013). In the 1980's, Poterba and Summers (1988) documented this contrarian-strategy, which indicates that underperforming stocks (losers) yielded substantially better returns than the overperformers (winners). This was an indication of mean-reversion, an idea that the stocks would revert back to its equilibrium form after an event. Moreover, the authors examined 17 different foreign equity markets, analyzing the statistical evidence bearing on whether transitory components justifies a large portion of the variance in common stock returns. They concluded that the explanation of the mean-reverting behavior was due to time-varying returns and speculative bubbles which caused stock prices to deviate from its fundamental values (Poterba and Summers, 1988).

Furthermore, mean-reversion can also be discovered in light of investment behavior. De Bondt and Thaler (1985) conducted a market research on investment behavior, analyzing the links between mean-reversion and overreaction in the market. Their hypothesis proclaimed that individuals tend to put more effort on news pointing in the same direction, resulting in the systematical mispricing of stock prices. This irrational behavior, as pointed out by De Bondt and Thaler (1985), is ameliorated by successive price movements in opposite course to its correct market value. Thus, acting on underpriced losers and overpriced winners yielded cumulative abnormal returns over the investment period. Thus, discovering a substantial weak form of market inefficiencies.

### 2.1.1 Pure Arbitrage vs. Statistical Arbitrage

We have now outlined the concept of the Efficient-Market Hypothesis where the idea of no-arbitrage is a central element. We, therefore, deliberate in greater detail the concept of arbitrage and its distinction from statistical arbitrage.

The concept of arbitrage, as outlined by Do and Faff (2010), is a strategic process that capitalizes on inconsistencies in assets prices without any form of risk or net investment. The notion refers to buying an asset in one market and simultaneously selling it in another at a higher price, thus profiting from the temporary differences in the two prices. In more detail, Björk (1998)

defines a pure arbitrage possibility as a self-financing portfolio $h$ where the value process has a deterministic value at t=0 and a positive stochastic value $V_t$ at time $t \geq 0$. If we let $h_i(t)$ denote the number of shares in the portfolio, and $S_i(t)$ the price of a stock which trades in continuous time, than the value of the portfolio will be,

$$V_t = \sum_{i=1}^{n} h_i(t) S_i(t) \tag{2.7}$$

Then the portfolio $\sum_{i=1}^{n} h_i(t)$ is said to be self-financing if,

$$dV_t = \sum_{i=1}^{n} h_i(t) dS_i(t) \tag{2.8}$$

The derived equation indicates that when new prices $S(t)$ are manifested at time t, one re-balances the portfolio (Björk, 1998). The re-balancing consists of purchasing new assets through the sale of old assets which already exists in the portfolio. Moreover, the self-financing portfolio can consist of long and short positions in several risky assets which results in a zero initial cost. This means that the portfolio is self-financing because there is no exogenous infusion or removal of money (Lindström et al., 2015). On the premise of self-financing strategies, Björk (1998) defines arbitrage as a portfolio $h$ with a cumulative discounted value such that,

$$V_0^h = 0 \tag{2.9}$$

$$P(V_T^h \geq 0) = 1 \tag{2.10}$$

$$P(V_T^h > 0) > 0 \tag{2.11}$$

where equation (2.9) states that the portfolio is self-financed and has a zero initial cost. The second property (2.10) states that there is a 100% probability of a portfolio value of zero or greater. Furthermore, (2.11) expresses that there is always a probability of obtaining a discounted cumulative terminal value of greater than zero. This means that arbitrage is considered a risk-free profit after transaction costs (Björk, 1998).

Nevertheless, in the financial markets, an investor looking for an arbitrage opportunity typically engages in a trade that involves some degree of risk. In the specific case where these risks are statistically assessed through the use of mathematical models, it is appropriate to use the term statistical arbitrage (Lazzarino et al., 2018). Following the definition of Hogan et al. (2004), a statistical arbitrage is where the overall expected payoff is positive, but there is a probability of a negative outcome. Only when the time aspect approaches infinity and we repeat the process continuously, the negative payoff will converge towards zero. Given a stochastic process of the trading value on a probability space $\{\Omega, F, P\}$, Hogan et al. (2004) outlines four conditions for a statistical arbitrage portfolio,

$$V_0^h = 0 \tag{2.12}$$

$$\lim_{t\to\infty} E[V_t^h] > 0 \tag{2.13}$$

$$\lim_{t\to\infty} P(V_t^h) < 0 = 0 \tag{2.14}$$

$$\lim_{t\to\infty} \frac{Var[V_t^h]}{t} = 0 \ \text{ if } P(V_t^h < 0) > 0, \ \forall_t < \infty \tag{2.15}$$

where the first property inherent in a zero initial cost strategy i.e it is self-financing (2.12). Furthermore the strategy, in the limit, has a positive expected discounted cumulative cash flow (2.13) and, a probability of a loss approaching zero (2.14). Last, the average variance (over time) is converging to zero if the probability of a loss does not become zero in finite time (2.15). The last equation is only employed if there is a positive probability of losing money, because if $P(V_t^h < 0) = 0$ , $\forall_t \geq T$ with $T < \infty$, it describes the basic arbitrage as outlined by Björk (1998). Hence, a statistical arbitrage will accumulative riskless profit in the limit.

## 2.2 The Arbitrage Pricing Theory

In the second theoretical point of departure, we will outline the Arbitrage Pricing Theory (APT) as a mean to discover the "free lunch". As first outlined by Ross (1975), the APT is based on the idea that stock returns can be predicted using a linear model of multiple systematic risk factors. Ross (1975) describes these factors as economic risk factors such as business cycles, interest rate fluctuations, inflation rules etc. According to Ross (1975), the exposure of these factors will affect a stocks risk and hence, its expected return. While pure arbitrage imposes restrictions on prices observed at a specific point in time, the APT seeks to explain expected returns at different points in time (Poitrast, 2010). Because of this, any deviation from the theoretical optimum can be seen as a mispriced stock. As described by Vidyamurthy (2004) and Harlacher (2016), the theory uncover the heart of pairs trading because stocks with the same risk exposure will provide the same long-run expected return and, therefore, the APT may serve as a catalyst to identify arbitrary opportunities.

This line of thinking will be the basis of our statistical arbitrage strategy. If we are able to identify stocks with similar risk profile, any deviation from the APT expectation will be an opportunity to capitalize on relative mispriced stocks. Moreover, Harlacher (2016) outlines the relationship as presented in Figure 2.1, where the co-movement between two stocks only exists due to their common relation to underlying factors.

In greater detail, the APT structure the expected return of a stock in the following way,

$$r_i = \beta_o + \sum_{j=1}^{k} \beta_{i,j} F_j + u_i \tag{2.16}$$

where $F_j$ can be seen as a factor, and $\beta_{i,j}$ as the risk exposure of that factor. The $\beta_o$ together with $u_i$ are interpreted as the idiosyncratic part of the observed return. In addition to this linear dependence structure as outlined by Harlacher (2016) and Ross (1975), there are other assumptions pertaining to this model:

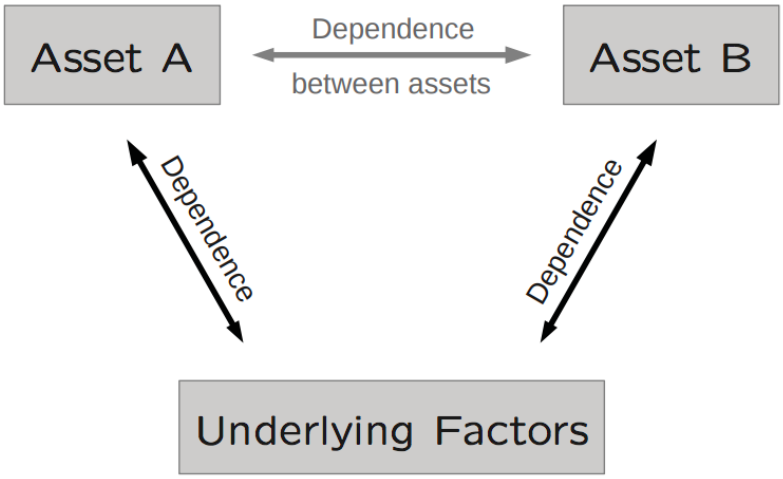



Figure 2.1: Dependency structure of two assets with the same underlying factors

1. Assumption: $E[u_i] = 0, \quad \text{for all } i$

2. Assumption: $E[u_i(F_j - E[F_j]) = 0, \quad \text{for all } i \text{ and } F_j$

3. Assumption: $E[u_i, u_h] = 0, \quad \text{for all } i \text{ and } h \text{ with } i \neq h$

From the first assumption, the expected mean of the residual is zero. This follows that the idiosyncratic risk is reflected in $\beta_o$. This means that if we are to expect that the factors are representing the systematic risk, then a stock with zero exposure to these factors should generate the same expected return as the risk-free rate (Harlacher, 2016). Then it holds true that $\beta_o$ is equal to the risk-free rate for all subsequent stocks, which again means that *there is no such thing as a free lunch*. Put in another way, if you are not willing to have any risk exposure, you cannot expect the stock return to be greater than the risk-free rate. Assumption two and three states that since the dependency of the assets is through the risk factors, the remaining part is created such that they are independent and uncorrelated from each other (Harlacher, 2016). This ensures that there is no multicollinearity among the factors (James et al., 2013).

Moreover, we can design the factors so that they are orthogonal to each other and to the residual $u_i$. This has the advantage of obtaining better parameter estimates (Shukla, 1997). In addition, this will help us to not select *ad hoc* risk factors (Shukla, 1997). In matrix form, we

can rewrite (2.16) as,

$$r_i = \beta_o + \beta_i^T F_j + u_i \tag{2.17}$$

where $u_i$ is a vector of random variables and $F_j$ is a $(k+1)$ vector of random factors. If we normalize the variables, we get $E[F_j] = 0$ and $E[u_i] = 0$, then the factor model implies $E[r_i] = \beta_o$. If this relationship truly exists with the underlying assumptions, Harlacher (2016) outlines the variance and the covariance of the stocks as,

$$Var(r_i) = \beta_i^T V \beta_i + \sigma_i^2 \tag{2.18}$$

$$Cov(r_i, r_h) = \beta_i^T V \beta_h \tag{2.19}$$

with $\sigma_i^2 = E[u_i^2]$, and $V$ being a $(k+1) \times (k+1)$ matrix with the covariance factor changes. As aforementioned, we will in this thesis utilize the APT to extract the underlying risk factors to better seek arbitrage opportunities. Moreover, we will follow the line of thinking of Chamberlain and Rothschild (1983) by using the principal component analysis to estimate and extract the common underlying factors by composing the eigenvectors. This will be the initial foundation when forming our research design for conducting a pairs trading strategy.

## 2.3 Pairs Trading

In the world of finance, pairs trading is considered the origin of statistical arbitrage (Avellaneda and Lee, 2008). It is a statistical arbitrage strategy which matches a long position with a short position of two stocks with relatively similar historical price movements. Even though there are several approaches to pairs trading, this thesis will analyze pairs trading through the concept of cointegration, as presented by Vidyamurthy (2004). Jaeger (2016) argues that cointegration between two stocks implies that there is a weak form of market efficiency because it opens for arbitrary situations based on historical information. This statistical relationship is, therefore, necessary to explain why there can be a presence of pairs trading (Jaeger, 2016).

If $Y_t$ and $X_t$ denotes the corresponding prices of two stocks with the same stochastic process, Avellaneda and Lee (2008) models the system on differentiated form as,

$$\frac{dY_t}{Y_t} = d\alpha_t + \beta\frac{dX_t}{X_t} + dS_t \tag{2.20}$$

Where $S_t$ is a stationary process and the cointegrated spread. This means that the spread between the two stocks do not drift apart too much and in the ideal case, the spread has a constant mean over time. If the spread deviates from its historical mean, we act and capitalize on the temporary inconsistency. During trading, limits are set on the top and bottom of the spread. If it ever goes below or above a particular normalized spread score, one will go long or short in the spread. In more detail, by entering a long position in the spread, the investor buys one unit of stock $Y_t$ and short $\beta$ units of stock $X_t$, which implies that the spread $S_t$ is below the equilibrium average (Avellaneda and Lee, 2008). Consequently, an opposite position will be entered if one goes short in the spread. This implies that one buy the relatively undervalued stock and sell the relatively overvalued stock in such portion that the total position is non-sensitive to overall market movements i.e the portfolio beta becomes zero (Harlacher, 2016). Once a trading signal like this occurs, a reversion to the historical mean is expected. The position will be closed when convergence is close to the mean (Kakushadze, 2014).

In light of the efficient-market hypothesis, if a cointegrated relationship between two stocks is identified we can expect two outcomes: i) the relationship may cease to exists and the weak form of market-efficiency holds true, which will result in a profit loss. This could be a result of news or shocks related to any of the stocks, and the recovery of such an event might last longer than the estimated trading period, or utmost, never at all. ii) The cointegrated relationship is, in fact, true and we will trade in the spread levels (Jaeger, 2016), which will result in a rejection of the weak-form of market efficiency.

In theory, pairs trading cannot be justified under the efficient market hypothesis. This needs to be violated through a mean-reverting behavior of stock prices, which ensures a relative long-term equilibrium between two stocks (Jaeger, 2016). The question is then if there are such

violations of the efficiency of markets, and what have scholars and investors historically done?

### 2.3.1 Empirical Evidence of Pairs Trading

It was not until the work of Gatev et al. (1999) that the first empirical strategy of mean-reversion was used on pairs trading. In the article, they employed a method named the distance approach, a technique engaging in deviations between normalized prices. In their study, they back-tested a pairs trading strategy on U.S equities, in the period of 1967 to 1997. Their strategy yielded excess return of 11%, robust for any transaction costs. Notwithstanding, the article of Gatev et al. (1999) extended the notion of its ancestors in that it deliberated the importance of mean reversion for generating pairs trading profits. The same article was reproduced in 2006 where they expanded the data period by five years, still with positive results.

Succeeding the study of Gatev et al. (2006), Do and Faff (2010) replicated the study by expanding the sample period by seven years. With the growing popularity of pairs trading and the technological advancement in the financial markets, they wanted to analyze whether the strategy could still produce excess return. Do and Faff (2010) argued that the increased competition among arbitrageurs would result in situations where even the smallest opportunity would be exploited. In addition, the arbitrageurs would face risks such as fundamental risk and synchronization risk which all work to prevent arbitrage. In their study, they revealed declining profits of pairs trading over the sample period. This was because of fewer convergence properties, higher arbitrage risk, and increased market efficiency. However, consistent with Gatev et al. (2006), Do and Faff (2010) claimed that pairs trading worked particularly well in time of financial crisis. This aspect is further investigated by Acharya and Pedersen (2005) which found that the profitability of pairs trading is negatively correlated with market liquidity.

In recent years, the advent of computer power and statistical methods has contributed to more advanced methods to the field of pairs trading. Some of these are through various machine learning techniques. The most cited article to include a principal component analysis (PCA) in pairs trading was conducted by Avellaneda and Lee (2008). In their training period, they used PCA as a mean to decompose and extract risk components, as a way to sort out the

idiosyncratic noise. The strategy yielded excess returns but received critique since the authors experimented with different threshold for entry and exit signals. PCA has also been used for portfolio optimization, as described in Tan (2012) where he gained positive results in terms of portfolio efficiency.

# Chapter 3

# Research Design and Methodology

This chapter introduces the main data and methodology used in our research design. The first sub-chapter gives a brief explanation of the different stages of our research design. Further on, we present each stage process with its respective theoretical and methodical concepts. All methodological work is conducted in the open source program Python and the python code for each stage are presented in Appendix 7.4.

## 3.1 Overview of the Research Design

For our research design, our main goal is to create an algorithm suitable for an efficient pairs trading strategy. To make our research design more transparent and easy to follow, we have designed it as an interrelated four stage process. The figure below describes the process of the different stages.


**Stage 1-3 are stages which seeks to find valid and suitable stock pairs. Stage 4 is the last stage process where we enter trading positions.**

Figure 3.1: Overview of research design

Stage one encompasses the process of data sampling and management. The data sample consists of daily returns and fundamental ratios of all companies at The Oslo Stock Exchange Benchmark Index. In this way, we can analyze price pattern and movements through the dimensions of price data and fundamental ratios.

In the second stage, we have structured our data through different unsupervised machine learning techniques. This is done so we can extract and cluster common underlying risk factors of stock returns. The first unsupervised method is a principal component analysis, a tool used for dimensionality reduction and factor analysis. We then apply a density-based clustering technique, a method for discovering unknown subgroups in the data. Last, we try to visualize the data through t-Distributed Stochastic Neighbor Embedding.

In the third stage of the research design, we seek to find mean-reversion among stock pairs in the clusters. This is done through the cointegration method, namely by following the procedure of the Engle-Granger two-step approach.

For the last stage, we implement the trading procedure. By generating trading signals on rolling z-scores, we test the strategy out-of-sample based on identified cointegrated stock pairs.

## 3.2 Stage 1: Data Management

The first stage in our research design encompasses the process of data sampling and management. The data set consists of daily historical closing prices adjusted for dividends and stock splits. This is also the common practice in the pairs trading literature. We use adjusted closing prices because corporate actions do not change the actual value to investors. This enables us to examine historical returns in an accurate way and we avoid false trading signals, as documented by Broussard and Vaihekoski (2012). Seeking to move beyond conventional return perspectives, we have decided to bring fundamental accounting ratios into our analysis as insightful and stabilizing indicators of mean-reversion. The fundamental ratios are: debt-to-equity, return on invested capital and revenue growth. By implementing fundamental values, we can create more robust clusters and stock pairs. The three fundamentals ratios are chosen due to its power of revealing companies profitability and financial health.

The universe considered is The Oslo Stock Exchange Benchmark Index (OSEBX). The sample period starts from January 2013 and ends in December 2017. During the sample period, we consider the 67 stocks that are listed on the exchange today, yielding 2211 possible pairs to trade. Since the index is revised semiannually, with changes implemented on 1 December and 1 June, the number of possible pairs will change during the sample period. Choosing OSEBX, which comprises the most traded stocks in Norway, ensures an acceptable level of liquidity. The liquidity in the stocks is an essential factor because pair trading strategy involves short-selling. Moreover, we do not include stocks that were de-listed during the sample period for multiple reasons. First, less liquid stocks may be difficult to short and add greater operational costs (bid-ask spread). Second, it is easier to work with and structure stock-data that has the full price-series. Last, stocks that have been listed for several years, are considered to be more solid and will most likely be possible to trade in the nearest future.

All price-series data are gathered from Yahoo Finance and verified by comparison with data from Bloomberg and Amadeus 2.0. The financial ratios are assembled from Bloomberg. In addition to adjusting for stock splits and dividends, we have cleansed missing data by using previous closing prices. We do this for facilitating an effortless back-testing.

## 3.3 Stage 2: Stock Filtering

In the second stage of our research design, we seek to filter the stocks into clusters suitable for pairs trading. We do this by extracting common underlying risk factors. In the process of stock filtering, we will use three different unsupervised techniques within machine learning. These are Principal Component Analysis, Density-Based Clustering, and t-SNE.

### 3.3.1 Machine Learning

The concept of machine learning refers to a set of tools for modeling, predicting and understanding complex datasets (James et al., 2013). The tools for understanding complex data can be classified as supervised or unsupervised (James et al., 2013). Supervised learning is defined as learning from examples, or past experiences. The notion is that for each variable, $x_i \quad i = 1, ..., n$ there is a comparable dependent variable $y_i$. The objective of supervised learning is therefore to fit a model that accurately can predict the response of future observations. Statistical models such as linear regression, logistic regression and support vector machines are all examples of supervised learning techniques.

In contrast, unsupervised learning describes a situation were every variable, $x_i \quad i = 1, ..., n$ , has no associated response or dependent variable $y_i$ (James et al., 2013). In these types of situations, it is not possible to fit a regression model (since there is no response variable to predict). This is referred to as an unsupervised situation. Clustering and principal component analysis are types of unsupervised learning.

### 3.3.2 Principal Component Analysis

In the search of exploiting statistical arbitrage, we will search for stocks with the same systematic risk-exposure. This is because they will generate the same long-run expected return according to the Arbitrage Pricing Theory (Ross, 1975). Any deviations from the theoretical expected stock return can therefore be seen as a mispriced stock and, help us to places trades accordingly. In the process of extracting these common underlying risk factors for each stock, we use the Principal Component Analysis (PCA) on stock returns as described by Jolliffe (2002).

In the PCA process, we create new variables known as principal components. These are constructed in a way that the first component accounts for as much of the variance of the data as possible. Then, the second component will try to explain as much of the remaining variability as possible, and so forth (James et al., 2013). As described by Avellaneda and Lee (2008), each component can be seen as representing a risk factor. Since the first component explains the most variance of the underlying data, it can be said that this factor represents the largest sources of systematic risk.

In the search for the principal components, we convert the stock-data to standardized returns in line with the process of Avellaneda and Lee (2008), in the following matrix,

$$A = Y_{ik} = \frac{R_{ik} - \bar{R}_{ik}}{\bar{\sigma}_i} \tag{3.1}$$

where $R_{ik}$ represents the stock returns

$$R_{ik} = \frac{S_{i(t_o-(k-1)\Delta t)} - S_{i(t_o-(k\Delta t))}}{S_{i(t_o-(k\Delta t))}} \qquad k = 1, \ldots, M, i = 1, \ldots, N \tag{3.2}$$

In equation (3.2) we use historical closing prices of N stocks at OSEBX, going back M days, where $S_{it}$ is the adjusted closing price of stock $i$ at time $t$ and $\Delta t = 1/250$ since we operate with 250 trading days per year.

Since PCA creates a new feature subspace that maximizes the variance along the axes, it makes sense to standardize the data. Even though we have common units, their variances may be very different, and scaling is therefore necessary (James et al., 2013). From matrix A, we compute the correlation matrix,

$$\rho_{ij} = \frac{1}{M-1} \sum_{k=1}^{M} Y_{ik} Y_{jk} \tag{3.3}$$

The reason we create a correlation matrix from the returns, and not from the raw price data, is that a return correlation-matrix gives us a better understanding of price co-movements. From the correlation matrix, we need to extract the eigenvectors and eigenvalues in order to create our principal components. The eigenvectors determine the directions of the new feature space and the eigenvalues of their variance (Raschka, 2017). There are two main forms of extracting the eigenvalues and eigenvectors, namely through an eigendecomposition (see Puntanen S. (2011) or through a Singular Value Decomposition (SVD). In our algorithm, the latter approach is incorporated for greater computational efficiency (Sandberg, 2004). The SVD is a standard tool in linear algebra and matrix analysis, see Goulb and Van Loan (1996) and Madsen et al. (2004) for details on its computation and properties. Trough the SVD theorem, we decompose our matrix A as follows,

$$A = USV^T \tag{3.4}$$

where U is an orthogonal matrix where the columns are the left singular vectors, S is a diagonal matrix with singular values, and V is the transposed orthogonal matrix which has rows that are the right singular vectors. By multiplying the matrix with the transposed matrix $A^T$, we get,

$$A^T A = VS^2V^T \tag{3.5}$$

The left-hand side of equation (3.5) is the same as our correlation matrix $\rho_{ij}$. Since the cor-

relation matrix $\rho_{ij}$ is symmetric to $A^TA$, the columns of V now contains the eigenvectors of $A^TA$ and the eigenvalues are the squares of the singular values in S. This tells us that the principal components of matrix A are the eigenvectors of $\rho_{ij}$, and by performing SVD on $A^TA$, the principal components will be in the columns of matrix V. Now, we project the eigenvectors onto the original return series. This projection will result in a new subspace which corresponds to our principal components,

$$F_j = \sum_{i=1}^{N} \phi_i^j R_{ik} \tag{3.6}$$

We refer to the $\phi_i^j$ as the loadings of the principal components. Correlation between the original variables and the factors is the key to understanding the underlying nature of a particular factor (Goulb and Van Loan, 1996). We have notated the principal components as $F_j$ for connecting them to equation (2.16), which outlined expected stock returns as a linear model of multiple systematic risk factors, as described in the Arbitrage Pricing Theory. However, the factors from equation (3.6) cannot be interpreted as economic risk factors, but as new factors which captures the underlying variance from the dataset. Moreover, the loadings make up the principal component loading vector, which we constrain so that their sum of squares is equal to one, ensuring non-arbitrarily large variance (James et al., 2013).

**Determining the number of Principal Components**

Since we are reducing the observations into principal components, we must analyze how much of the information in the data is lost by projecting the observations into a new subspace (James et al., 2013). We are therefore interested in knowing the proportion of variance explained (PVE) by each principal component. As outlined by James et al. (2013) the total variance in the data set is defined as,

$$\sum_{j=1}^{p} Var(R_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} R_{ij}^2 \tag{3.7}$$

and the variance of the $m$th component is

$$\frac{1}{n}\sum_{i=1}^{N} F_{im}^2 = \frac{1}{n}\sum_{i=1}^{N}(\sum_{j=1}^{p} \phi_{jm} R_{ij})^2 \tag{3.8}$$

Therefore, the PVE of the principal component is given by

$$PVE = \frac{\sum_{i=1}^{N}(\sum_{j=1}^{p} \phi_{jm} R_{ij})^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} R_{ij}^2} \tag{3.9}$$

We use the PVE equation (3.9) as outlined by James et al. (2013) to determine the number of principal components we want to use in our new subspace. Here, we want the number of principal components to be as small as possible. That is, if we can capture the variation with just a few components, it would bring a simpler description of the data. On the other hand, to avoid any loss of information, we want to capture as much variation as possible. Meaning that we must allow for many components. The question of how many principal components one need is still inherently ambivalent, and will depend much on the specific area of application and also on the data set used (James et al., 2013). Kim and Jeong (2005) outlines how the components can be described in three parts:

1. The first principal component captures the greatest variance and thus, represents the market risk.

2. The succeeding number of principal components represent synchronized fluctuations that only happens to a group of stocks.

3. The remaining principal components indicates random fluctuations in the stocks.

On this basis, we need to determine the number of components which enables us to capture part 1 and 2 but leaving out part 3. In the study of Avellaneda and Lee (2008), they selected the number of components that explained 55 percent of the total variance and argued that it would perform better than a predefined number of components. Others, such as Bai and Ng (2002) advocate a penalty function of selected factors to penalize for potential overfitting.

In this thesis, the number of components will be determined by the point at which the marginal proportion of variance explained from each principal component is small and insignificant, a technique known as *elbowing*, as described by James et al. (2013). It is worth noticing that this technique is *ad hoc*. Furthermore, we do not conduct all the PCA steps by hand, but as an integrated algorithm in Python. For further details on Python coding, see Appendix 7.4.2.

### 3.3.3 Density-Based Spatial Clustering of Applications with Noise

After the extraction of the principal components, we now seek to cluster the components, combined with the fundamental ratios, into regions of high density. This will help us to discover any hidden patterns, as we cluster stocks together with similar risk profiles. This process is done through the clustering technique Density-Based Spatial Clustering of Applications with Noise (DBSCAN), a technique developed by Ester et al. (1996). We use DBSCAN, as an alternative to K-nearest neighbors, because it does not require a predefined number of clusters in advance as described in Trana et al. (2012).

In DBSCAN, the goal is to identify dense regions, which can be measured by the number of objects close to a given point (Ester et al., 1996). The concept is based around density reachability, where a point $q$ is density reachable by another point $p$ if the distance between the points is below a certain threshold $e$, and $p$ is enclosed by sufficiently many points (Raschka, 2017). The DBSCAN algorithm consists therefore of two main parameters. The first parameter, $e$, reflects the radius of the neighbors around a given data point. The second parameter, *minPts*, represents the minimum number of points we want to have in our cluster. By definition, $q$ is considered to be density-reachable by $p$ if there exists a progression of $p_1, p_2, \ldots, p_n$, such that $p_1 = p$ and, $p_{i+1}$ is directly density-reachable from $p_1$ (Ester et al., 1996). As a general rule, the parameter of *minPts* can be derived from the number of dimensions (D) in the data set, as minPoints $\geq$ D + 1. However, the variable must at least contain three points, otherwise, the technique would yield the same as hierarchical clustering (Ester et al., 1996). For $e$, one must balance the choice between outliers and number of clusters formed.

The process of DBSCAN starts with an arbitrary data point, where the $e$-neighbors are gath-

ered. If the amount complies with *minPts*, a cluster is formed. Otherwise, it is classified as noise. Then iterate the process until all density-connected clusters are formed. The exact Python code is given in Appendix 7.4.2.

### 3.3.4 t-Distributed Stochastic Neighbor Embedding

Once we have clustered our data, we need to find a way to visualize it. The problem at hand is that we are dealing with a data set consisting of numerous dimensions and observations. Computers have no problem processing that many dimensions. However, we humans are limited to three dimensions utmost. Therefore, we seek to reduce the number of dimensions into a two-dimensional set, in a way that we can gain confidence in the DBSCAN output. We will do this by using the nonlinear dimensionality technique known as t-Distributed Stochastic Neighbor Embedding (t-SNE), as first introduced by van der Maaten and Hinton (2008). The unsupervised statistical learning algorithm is appropriate for embedding high-dimensional data in low-dimension visualization (Derksen, 2006).

We refer to the original high dimensional data set as X, where a data point is a point $x_i$. For our new (low) dimensional data set, we referred to this as Y, with a map point $y_i$. In the t-SNE process, we still want to conserve the structure of the data. In more detail, if two data points are close together we also want the corresponding map points to be close too. Therefore, let $|x_i - x_j|$ to be the distance among the data points, and $|y_i - y_j|$ the gap within the map points. We can now define the conditional probability of the data points as,

$$p_{j|i} = \frac{\exp\left(-\left|x_i - x_j\right|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\left|x_i - x_k\right|^2 / 2\sigma_i^2\right)} \tag{3.10}$$

The equation tells us how close the data points $x_i$ and $x_j$ are to each other, given a Gaussian distribution with variance $\sigma^2$. This means that the probability distribution is constructed in such a way that similar objects have a high probability of being picked, while non-similar objects have a low probability of being chosen (Jaju, 2017). From this, we generate the joint

probabilities,

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{3.11}$$

We also define a similar matrix of our map points,

$$q_{ij} = \frac{exp(|y_i - y_j|^2)}{\sum_{k \neq i} exp(|y_i - y_k|^2)} \tag{3.12}$$

The distribution of the map points is based on the same idea as earlier but uses a different distribution. The conditional probability of the low dimensional data set is based on t-distribution. The main difference between the two conditional probabilities is that $p_{ij}$ is fixed, while $q_{ij}$ depends on the map points. The objective now is the minimize the distance between the two probabilities. This is because we want the data points to yield comparable map points. The minimization process depends on the mismatch between the similarity of data and map points, that is $p_{ij} - q_{ij}$.

The process is done through the Kullback-Leiber divergence with a gradient descent,

$$KL(P||Q) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.13}$$

The equation tells us that if two map points are distant to each other whereas the corresponding data points are not, they will be drawn together (Derksen, 2006). The same will happen if they are nearby, they will be repelled. The process iterates until a final mapping is procured and the equilibrium is attained. From this, it creates a visualization of the clusters formed in a two-dimensional plane. For further mathematical computation, see van der Maaten and Hinton (2008). For Python coding, see Appendix 7.4.2.

## 3.4 Stage 3: Identifying Mean-Reversion

In the third stage of our research design, we seek to find mean-reversion among stocks in the clusters discovered in the previous stage. This will be identified by cointegration. Below is a description of the elements regarding cointegration.

### 3.4.1 The Cointegration Approach

In this thesis, the cointegration approach for pairs trading has been chosen, following the framework of Vidyamurthy (2004).

#### Stationarity

A time series is stationary when the parameters of the underlying data do not change over time (Wooldridge, 2009). Wooldridge (2009) describes stationary as a stochastic process $x_t \in \{t = 1, 2...\}$ where every moment in time, the joint distribution of $(x_1, x_2, ..., x_m)$ is the same as the joint distribution of $(x_{1+h}, x_{2+h}, ..., x_{m+h})$ for all integers $h \geq 1$. The definition is referred to as a strict stochastic stationary process. Nevertheless, a weaker form of stationarity is more commonly applied in finance due to its practical application on data samples. On this basis, this thesis will engage in a weak stochastic stationary process. The process is weak stationary if, for all values, the following is true:

- $E(Y_i(t)) = \mu$ (Constant mean)
- $var(Y_i(t)) = \sigma^2$ (Constant variance)
- $cov(Y_i(t), Y_{i+s}(t) = cov(y_t, y_{i-s}(t)) = \gamma$ (Covariance depends on s, not t)

The weak form of a stochastic stationary process needs to have constant mean and variance. The most important feature is that the process has a constant mean. If the spread between the stock pair deviates from the mean, we can capitalize by trading on this. Furthermore,

Wooldridge (2009) outlines that the covariance only concentrates on the first two moments of a stochastic process. Hence, the structure does not change over time.

Non-stationary time series is referred to as random walks or random walks with a drift (Wooldridge, 2009). Random walks slowly wander upwards or downwards, but with no real pattern, while random walks with a drift show a definite trend either upwards or downwards. As a rule of thumb, non-stationary time series variables, which stock price series often are, should not be used in regression models, in order to avoid spurious regression. However, most time series can be transformed into a stationary process. This is done by differencing the time series, in such a way that the values projects change and not absolute values. If a time series becomes stationary after d times, it is referred to as an I(d). If $Y_i(t)$ and $X_i(t)$ are non-stationary I(1) variables, and a linear combination of them $S_i(t) = Y_i(t) - \beta X_i(t) - \alpha$ is a stationary I(0) process (i.e. the spread is stationary), then the set of $Y_i(t)$ and $X_i(t)$ time series are cointegrated (Wooldridge, 2009).

**Cointegration**

Cointegration implies that $Y_i(t)$ and $X_i(t)$ share similar stochastic trends, and since they both are I(1) they never diverge too far from each other. The cointegrated variables exhibits a long-term equilibrium relationship defined by $Y_i(t) = \alpha + \beta X_i(t) + S_i(t)$, where $S_i(t)$ is the equlibrium error, which represents short-term deviations from the long-term relationship (Wooldridge, 2009).

For pairs trading, the intuition is that if we find two stocks $Y_i(t)$ and $X_i(t)$ that are I(1) and whose prices are cointegrated, then any short-term deviations from the spread mean, $\overline{S_i}$, can be an opportunity to place trades accordingly, as we bet on the relationship to be mean reverting. When testing the spread for cointegration, we define the spread as,

$$S_i(t) = Y_i(t) - \beta X_i(t) + \alpha_i \tag{3.14}$$

When testing for cointegration, the Engle-Granger two-step method has been used[1].

- **Two steps in the Engle-Granger method:**

  1. Estimate the cointegration relationship using OLS.
  2. Test the spread for stationarity

The Augmented-Dickey-Fuller (ADF) test[2] is used to verify cointegration. Below is the general formulation of the ADF test:

$$\Delta S_i(t) = \alpha + \beta t + \gamma S_{i(t-1)} + \delta_1 \Delta S_{i(t-1)} + \sum_{i=1}^{k} \theta_i \Delta S_{t-i} + \varepsilon_t, \tag{3.15}$$

In our approach, the model of order 1 has been kept for all stock pairs. In addition, the ADF statistics depends on whether an intercept and/or a linear trend are included (MacKinnon, 2010). The pairs trading strategy involves taking positions in the stock themselves, therefore the intercept term is excluded. Thus, the ADF formulation we use is the following:

$$\Delta S_t = \gamma S_{t-1} + \theta \Delta S_{t-1} + \varepsilon_t \tag{3.16}$$

The null hypothesis is that there is no cointegration, the alternative hypothesis is that there is a cointegrated relationship. If the p-value is small, below 5%, we reject the hypothesis that there is no cointegrated relationship.

- Test: $\gamma$ coefficient p-values:
  - If $\gamma$ p-value $< 0.05$: stationary time series with 95% of statistical confidence.
  - if $\gamma$ p-value $> 0.05$: time series non-stationary with 95% of statistical confidence.

[1]For a detailed description of the method, see (Engle and Granger, 1987)
[2]For detailed information see (Dickey and Fuller, 1979)

## 3.5 Stage 4: Trading Setup and Execution

In the final stage of our research design, we seek to execute the trading algorithm and evaluate the results. As stated in several studies, the general rule is to trade the positions when they exceed a certain threshold. In this thesis, we have followed a similar trading procedure as proposed by Caldeira and Moura (2013) and Avellaneda and Lee (2008).

### 3.5.1 Trading Signals and Execution

In the trading strategy, we focus on the spread-process $S_i(t)$ (3.14), but as aforementioned neglecting the intercept term $\alpha_i$. This is because we only take position in the stocks themselves, $Y_i(t)$ and $X_i(t)$ accordingly. As described by Perlin (2009), a z-score is generated,

$$Z_i = \frac{S_i(t) - \overline{S_i}}{\sigma_{eq,i}} \tag{3.17}$$

where,

$$\overline{S_i} = \frac{1}{w} \sum_{j=i-w}^{i-1} S_i(t) \tag{3.18}$$

The z-score tells us the distance from the equilibrium spread in units of the equilibrium standard deviation. The z-score is used for generating trading signals and positions. In this thesis, we will utilize a rolling z-score to capture shifts in the spread. According to Reverre (2001) the length of the rolling-window should be short enough to be reactive to shifts and long enough to appear reasonably efficient in stripping noise out. On this basis and the fact that we are trading on a six months basis, we have settled on a 10-day rolling z-score (w=10). Once 10 days of information is gathered, the trading algorithm begins. On that basis, we analyze if the z-score is inside or outside the trading thresholds. This means that the portfolio is re-balanced with 10 days of historical data.

The trading strategy is described as follows,

- **Long Spread Trade:**
  - Enter long position: Previous (Z-Score > - 2) $--$ > Current (Z-Score < -2).
  - Exit long position: Previous (Z-Score < +1) $--$ > Current (Z-Score > -1).
- **Short Spread Trade:**
  - Enter short position: Previous (Z-Score < + 2) $--$ > Current (Z-Score > +2).
  - Exit short position: Previous (Z-Score > +1) $--$ > Current (Z-Score < + 1).

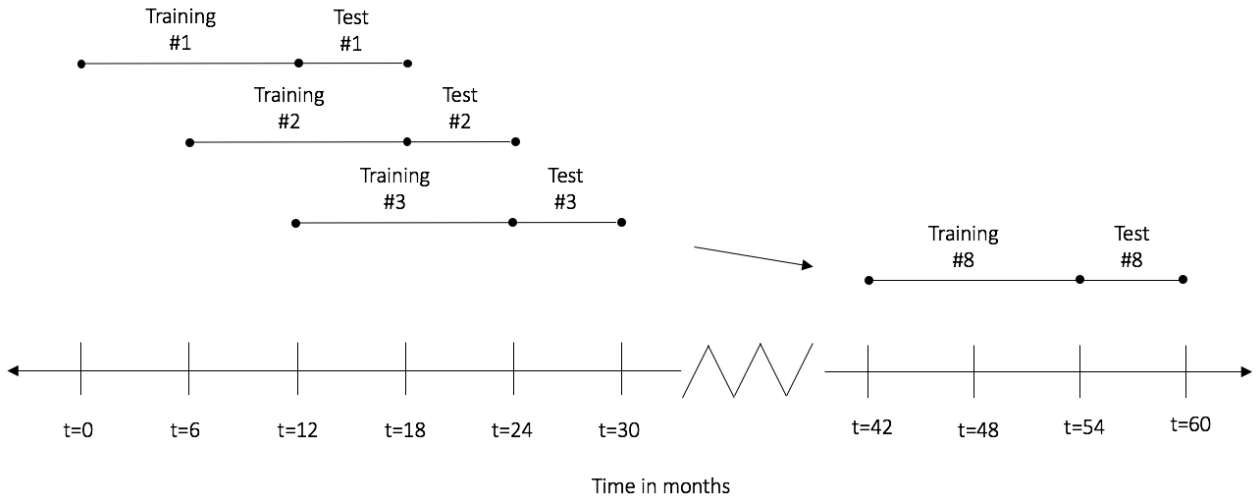
We enter a trade position when the z-score exceeds $\pm 2$ and closes the position when it drops below $+1$, and above $-1$ which is similar to Gatev et al. (2006), Andrade et al. (2005) and Do and Faff (2010). The reason for trading within these thresholds are as follows: when the z-score is far away from its equilibrium, we have reason to believe it will drift back again to its equilibrium (because the spread was stationary in the past). On the other side, closing the trade when the score passes the lower bound also makes sense since most stocks have some sort of deviations. The portfolio is rebalanced every day to maintain the desired level of asset allocation from our rolling z-score. If two stocks in a pair suddenly drift in opposite directions, the portfolio can end up with significant risk exposure. We, therefore, rebalance our portfolio as a mean to ensure that the portfolio is close to market-neutral, as documented by Do and Faff (2010). For algorithmic development, see Appendix 7.4.4.

### 3.5.2 Training and Testing Periods

When testing a trading algorithm on historical data, it is crucial to reserve a time period for testing purposes because it gives us an opportunity to see how the model would have performed in a real-life situation. That is why we have divided the sample period into training (in-sample) and testing (out-of-sample) periods.

In this thesis, we use a one year window for the training period and six months for the testing period. This sums up to eight training and testing periods in total. The length of training and backtesting periods is similar to the approach done by Caldeira and Moura (2013). When deciding the duration of the training and testing periods, there is no finite answer. One should make sure that the training period is long enough to determine if a cointegrated relationship exists between the stocks. Similarly, the testing period should be long enough for trading opportunities to occur.

The first training period is the year of 2013. Immediately after this period, the first test period starts, where we use the parameters computed in 2013. This continues on a rolling basis, meaning that pairs we trade are changed twice per year. Below is a description of the training and testing procedure, adapted from Andrade et al. (2005).




The figure shows the timing of training and testing periods. In the training period, valid mean-reverting stock pairs are identified. In the testing period, the trading strategy is executed.

Figure 3.2: Overview of training and testing periods

### 3.5.3 Transaction costs

In order to correctly assess the performance of the pairs trading strategy, the impact of transaction costs must be taken into consideration. According to Thapa and Sunil (2010), transaction costs are decomposed into commission, fees, and slippage. However, since we are trading on adjusted closing prices, we can disregard any slippage fees. Using the price model of Nordnet, one of the largest trading platforms in Norway, the commission is settled at 4.9 basis points (BPS) and fees at 0.25 BPS (Nordnet, 2018). In pairs trading, the simultaneous opening of a long- and short position means transaction costs are occurring twice, thus commission is settled at 9.8 BPS and fees at 0.5 BPS.

In addition, we have included short-sale rental costs as mentioned in Caldeira and Moura (2013). In the Norwegian equity market, the annual short fee rate is 450 BPS (Nordnet, 2018). There are 250 trading days per year, which gives us a daily shortage fee of 1.8 BPS. From the training period analysis, we estimated an average trading position to last four days. This gives us an average shortage fee of 7.2 BPS for each position. However, since shortage fees are only paid when closing a position, this cost should not be included when entering a position. Thus, we divide the shortage fee by two to represent the average trading cost[1]. This gives a rental fee of 3.6 BPS per trade. Assuming a continuation of this property for the testing periods, the total transaction cost is 13.9 BPS (0.139%) per trade for every stock pair.

In this thesis, we have decided to disregard the fixed costs of NOK 250 per short position (Nordnet, 2018) and tax implications. The fixed costs are in this model insignificant to the overall returns and, taxes will be pertaining any profit no matter what the revenue stream. Total transaction costs are presented in the table below.

| Commission | Fees | Rental | Total cost |
|---|---|---|---|
| 9.8 | 0.5 | 3.6 | 13.9 |

Table 3.1: Total transaction costs per trade for all stock pairs expressed in BPS

[1] 450 BPS annually rental cost/250 trading days) * 4 days average position)/2 = 3.6 BPS

### 3.5.4 Performance Measures and Hypothesis Testing

The return metric used is the nominally change in NOK of each stock pairs over the testing period. All stock pairs that were identified as cointegrated, will make up an equally weighted portfolio. In addition, the risk is defined as fluctuations in stock pairs returns measured by its standard deviation. This allows us to apply the Sharpe ratio (SR), as defined by,

$$SR_i = \frac{R_i - R_f}{\sigma_i} \tag{3.19}$$

with $R_i$ denoting the return of a stock pair, and $R_f$ the risk-free rate. The risk-free rate is incorporated as 10-year Norwegian government bond with 1,95 % p.a as of 18.04.2018 (Norwegian Central Bank, 2018). The standard deviation of a pair is denoted with $\sigma_i$. The Sharpe ratio is measuring the risk premium per unit of risk (Bodie et al., 2014)

Apart from Sharpe ratio, we also present the measure of alpha, $\alpha$, as obtained from the Capital Asset Pricing Model (CAPM). The theory states that one can only obtain excess return if one is willing to take risk. As according to Markowitz (1953), this can only be acquired by bearing systematic or market risk. The theory describes the relationship between risk and reward as,

$$E[R_i] - R_f = \beta_i(E[R_m] - R_f) \tag{3.20}$$

where $\beta_i = \frac{Cov(R_i,R_m)}{\sigma_M^2}$ and $E[R_m]$ denotes the expected market return. The $\beta_i$ denotes the covariance between our stock pair and the market, over the market variance. This tells us that $\beta_i$ describes the systematic risk captured by our stock pair. So, on the basis of CAPM, it gives us a relationship between the expected excess return and risk premium. Knowing this, we rearrange (3.20) to obtain the alpha,

$$\alpha_i = E[R_i] - R_f - \beta_i(E[R_m] - R_f) \tag{3.21}$$

The alpha tells us how much better or worse our pairs trading strategy performed relative to its benchmark i.e how well it performed relative to other securities with similar risk exposure.

That is why this metric can be used to determine whether our strategy is able to generate excess return. On this premises, to see if there are arbitrage opportunities, we impose the following hypothesis:

$H_0$: *There are no arbitrage opportunities*

$H_1$: *There are arbitrage opportunities*

We will utilize a one sample t-test to conclude whether there is arbitrage and to see if the strategy provided excess return. Since we are concerned with positive excess return only, we will employ a one-sided t-distribution.

## 3.6 Research Design Review

After a detailed description of our research design, a short summary of the main parts is presented in Table 3.2.

| **Stage 1: Data Management** | **Reasoning** |
|---|---|
| Historical adjusted closing prices converted to return series from January 2013 to December 2017. Fundamental ratios: ROIC, Debt-To-Equity & Sales Growth<br>All data gathered from Yahoo Finance and Bloomberg. | 1.1 Adjusted closing prices are used to avoid false trading signals<br>1.2 Fundamental ratios and daily returns series are included to analyze co-movement through several dimensions.<br>1.3 Financial ratios to reveal companies profitability and financial health. |
| **Stage 2: Stock Filtering** | **Reasoning** |
| In order to identify valid stocks, we restrict our research design to extract common underlying factors of stock returns.<br>This is done through the unsupervised techniques PCA and DBSCAN. Last, we visualize the clusters through t-SNE. | 2.1 PCA: Extract common underlying risk factors of stock returns.<br>2.2 DBSCAN: Cluster the components, combined with the fundamental values.<br>2.3 t-SNE: Visualize the clustered data. |
| **Stage 3: Identifying Mean-Reversion** | **Reasoning** |
| Use cointegration as a mean to identify mean-reversion and weak stationarity. Following the Engle-Granger two-step procedure with 5% significant level. | 3.1 Cointegration may demonstrate signs of mean-reversion in the future.<br>3.2 The Engle-Granger procedure is well known in statistics and econometrics.<br>3.3 No-intercept included because we are only taking positions in the stocks. |
| **Stage 4: Trading Setup and Execution** | **Reasoning** |
| Conducting the trades out-of-sample with eight test periods. A 10-day rolling z-score is created for the generation of trading signals and positions. All z-score parameters are updated daily. Long/short positions if z-score exceeds $\pm$ 2, close if z-score passes $\pm$ 1. | 4.1 A 10-day rolling window is used to avoid look-ahead bias. A daily rebalancing through an updated z-score.<br>4.2 Trading signals are estimated from training periods |

Table 3.2: Review of the Research Design

# Chapter 4

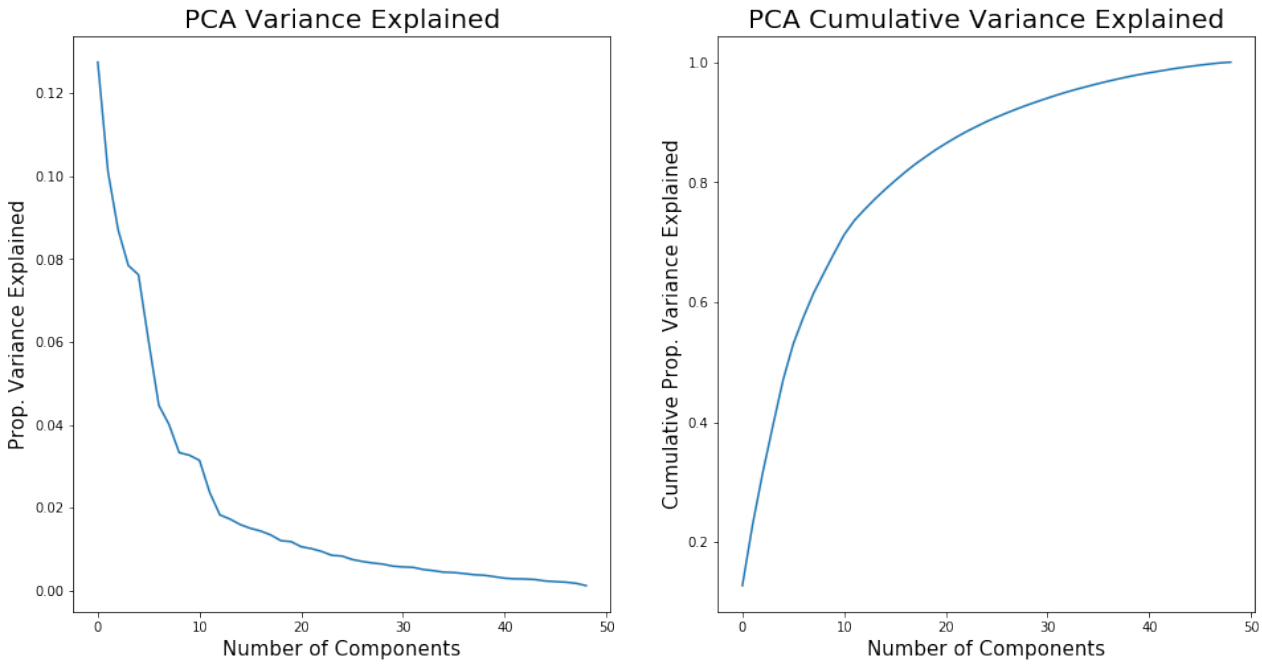
# Results

## 4.1 Determining the Number of Principal Components

The result from the PCA is presented in Figure 4.1. The figure describes the results from training period 1. In the screen plot to the left, we can see the proportion of variance explained by each of the principal components. On the right, we can see the cumulative proportion of variance explained.

Along the y-axis is the proportion of variance explained, as explained by equation (3.9). Remember that we standardized our data in the stock filtering process, which means that each variable has a mean of zero and variance of one. Because they have been standardized, each variable contributes to one unit of variance to the total variance in the data.

From Figure 4.1, we can see that the first five components explain roughly 40 % of the total variance, while 33 of the components (of the 50 total) is necessary to capture 95% of the variance. Not surprisingly, the proportion of variance explained exhibits a logarithmic relationship with the number of components which means that the marginal explanation of each component is diminishing. This is because each component can be seen as representing a risk factor and it is natural to think that different risk factors have different impact of the variance. Furthermore, the first component is similar to the market risk (Avellaneda and Lee, 2008). For further visualization of the first principal component, see Appendix 7.5.




From training period 1. In the screen plot to the left, we can see a plot illustrating the proportion of variance explained by each of the principal components. On the right, we can see the cumulative proportion of variance explained.

Figure 4.1: Determining the number of principle components

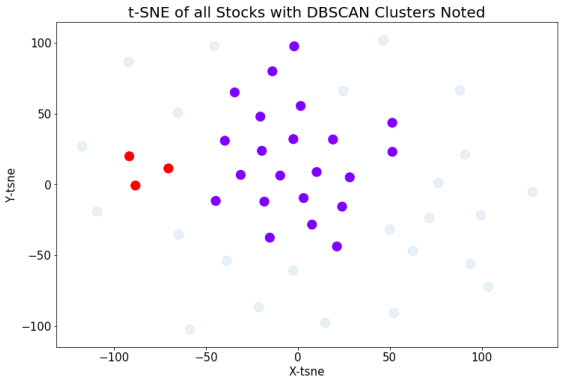
On the left side of Figure 4.1, we can see the proportion of variance explained from each principal component. From the figure, we can spot that the insignificant drop is around 12 components. Here is the elbow[1]. By settling 12 components, we are able to capture 75% of the variance explained. The same number of components have been chosen for the remaining periods, because of diminutive changes of variance explained across the training periods.

[1]Using the elbowing technique as explained in chapter 3.3.2

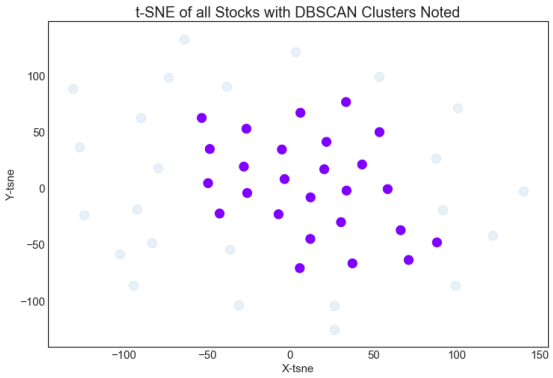
## 4.2 Cluster Discovering

Upon cluster analysis, we projected the new PCA subspace together with the financial ratios into regions of high density. The radius of each clusters $e$, where settled to 1.5. After trial and error, this number was chosen to find the point which minimized the maximum distance to all cluster points. For the minimum number of points, we settled *minPts* to 3. This was chosen because we wanted to ensure that stocks with similar risk profile was clustered together. Although the number of clusters will vary by the choice of $e$ and *minPts*, these values are chosen for all periods to be consistent.

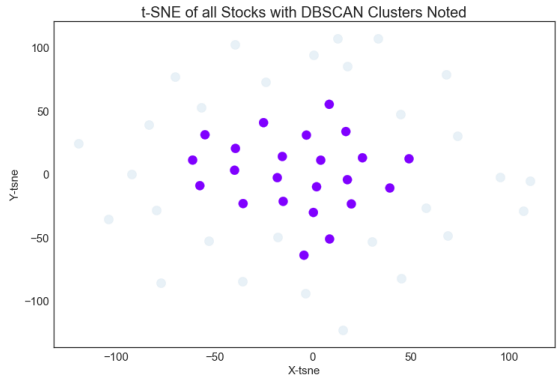
On the whole, we identified one or two clusters for every training period. The clusters did, however, differ in size and content from each training period. When visualizing the clusters, the t-SNE algorithm was implemented on the DBSCAN output, providing us with Figure 4.2. For a detailed description of all clusters formed, see Appendix 7.3.



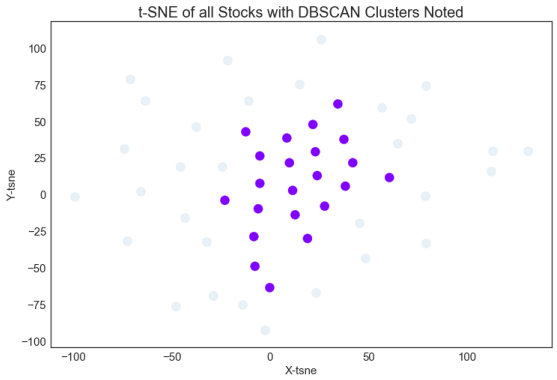

(a) clusters training period 1



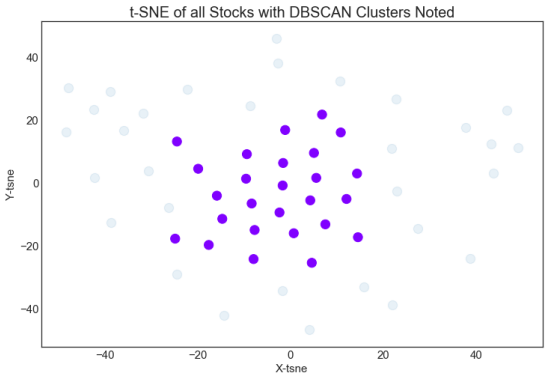

(b) cluster training period 2



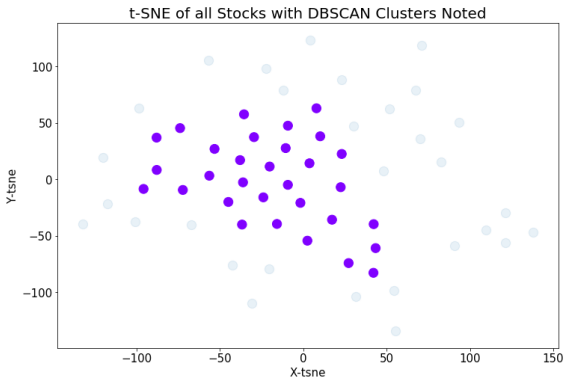

(c) cluster training period 3



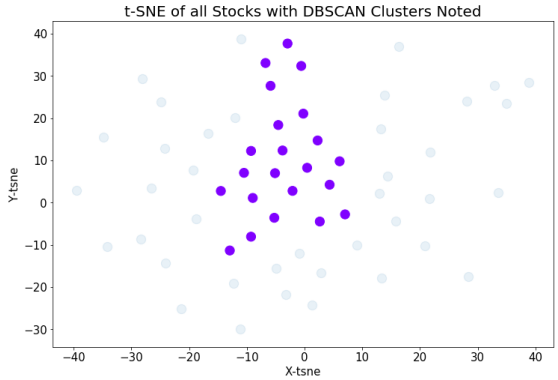

(d) cluster training period 4



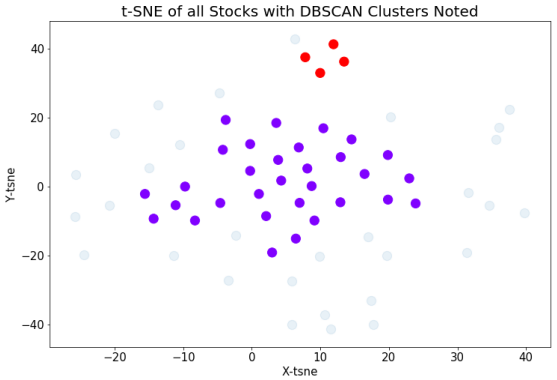

(e) cluster training period 5




(f) cluster training period 6




(g) cluster training period 7



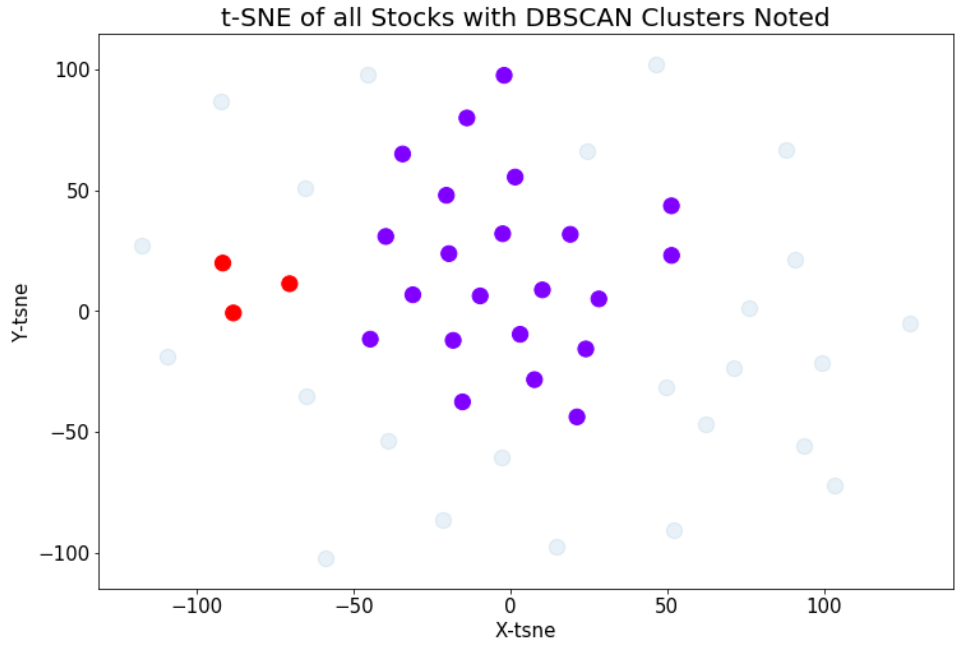

(h) clusters training period 8

Purple indicates cluster 1, red indicates cluster 2.

Figure 4.2: Clusters formed in all training periods

Analyzing the clusters in detail, we can see from Figure 4.3 and Table 4.1 the clustered stocks from training period 1. Cluster 1 is distinguished with the purple color, cluster 2 in red, and stocks that do not fit into a cluster are represented by the faded dots. From Table 4.1, we can see that cluster 1 is a combination of various stocks with similar risk exposure. From cluster 2, we obtained the stocks: DNB SpareBank 2 and Storebrand, all of whom are large financial banks in Norway.

For a detailed description of every cluster from all training periods, see Appendix 7.3.




Cluster visualization on DBSCAN output through t-SNE for training period 1. Cluster 1 in purple and cluster 2 in red.

Figure 4.3: Visualization of clusters formed from training period 1

| Stocks in cluster 1 | Stocks in cluster 2 |
|---|---|
| BAKKA, GJF, KOG, MHG, NHY, OLT, ORK, PGS, SALM, SCHA, SDRL, SNI, STL, SUBC, TEL, TGS, VEI, WWI, WWB, WWL, YAR. | DNB, SRBANK, STB |

Table 4.1: Stocks in clusters from training period 1

## 4.3 Summary of the Results

### 4.3.1 Summary of the Results Without Transaction Costs

Table 4.2 displays the overall results from our pairs trading strategy without transaction costs. The returns, standard deviation and Sharpe ratio are semi-annualized figures. This is because each testing period is six months. In the last three columns, the performance measures are from the Oslo Stock Exchange Benchmark Index (BM). In the last row of the table, the measures are annualized. For a detailed description of every test period, see Appendix 7.2.2.

| $Period$ | $Nb\,Pairs$ | $R_p$ | $\sigma_p$ | $SR_p$ | $R_{BM}$ | $\sigma_{BM}$ | $SR_{BM}$ |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 5.17% | 3.62% | 1.16[3] | 12.97% | 7.79% | 1.54 |
| 2 | 4 | -2.16% | 6.75% | -0.46 | -7.46% | 12.68% | -0.66 |
| 3 | 12 | 2.63% | 3.19% | 0.52 | 8.59% | 10.31% | 0.74 |
| 4 | 1 | -2.61% | 5.45% | -0.66 | -3.11% | 14.24% | -0.29 |
| 5 | 5 | 4.51% | 7.32% | 0.48 | 0.00% | 17.87% | -0.03 |
| 6 | 7 | 1.08% | 2.96% | 0.04 | 11.82% | 9.47% | 1.15 |
| 7 | 13 | 0.29% | 3.07% | -0.22 | 1.02% | 7.90% | 0.01 |
| 8 | 11 | 5.89% | 5.17% | 0.95 | 15.51% | 7.18% | 2.02 |
| Total[1] | 60 | 3.64% [2] | 6.81% | 0.25 | 10.42% | 16.18% | 0.52 |

Table 4.2: Summary statistics of portfolio without transaction costs

Over the investment horizon, our pairs trading strategy yielded a 3.64% compounded annualized return and a standard deviation of 6.81% p.a. By comparison, an investment in the OSEBX over the same period would have provided a return of 10.42% p.a on average with a volatility of 16.18%. An investment in the benchmark clearly outperformed the pairs trading strategy. The cumulative returns of the strategy and OSEBX are visualized in Figure 4.4.

[1] All numbers in the bottom row are annually figures

[2] Calculated as annually geometric mean: $\frac{1.1538}{1}^{(\frac{1}{4})} - 1 = 0.0364$

[3] Sharpe Ratio is calculated in semiannually terms, we use 6-months risk-free rate: $r_f = (1 + 0.0195)^{\frac{1}{2}} - 1$

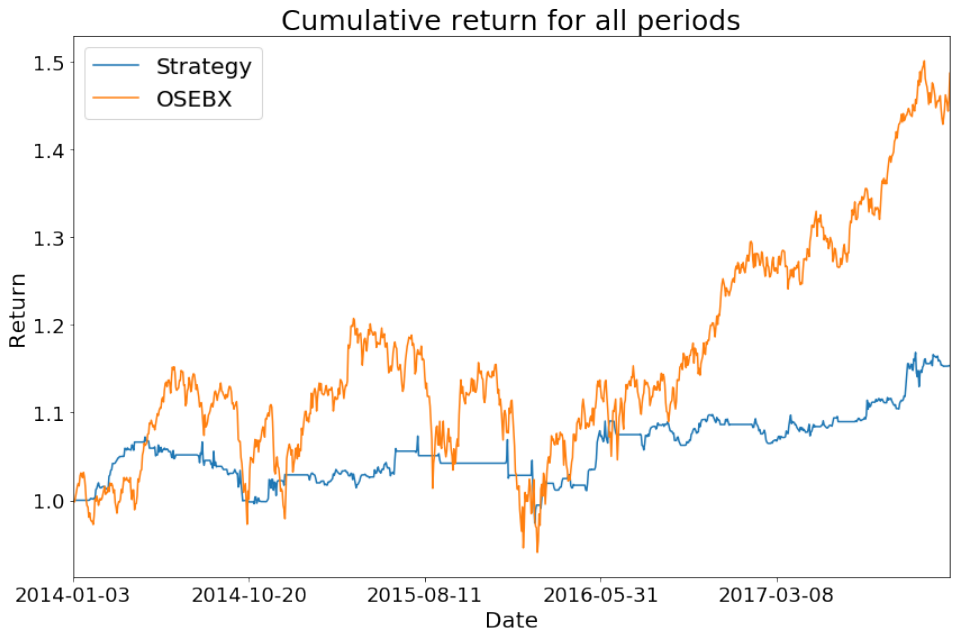



Figure 4.4: Value development for pairs trading strategy and benchmark index

From Table 4.2, our trading strategy obtained an overall positive Sharpe ratio of 0.25, which means that the strategy's risk premium is increasing by 0.25 per unit of risk. However, the Sharpe ratio of OSEBX (0.52) is superior.

To answer our research question on whether our pairs trading strategy could generate excess return, we imposed the following hypothesis on the alpha metric:

$H_0$: *There are no arbitrage opportunities*

$H_1$: *There are arbitrage opportunities*

Our aim is either to accept or reject the null hypothesis in order to determine whether or not our strategy provided excess return in form of arbitrage. The decision rule for the t-test is based on a one-sided t-test. This is because we are only interested in analyzing positive excess return and, therefore, we executed a one-sided t-test. In Table 4.3, we have the following output:

| Period | $\alpha$ Coefficient[1] | t | P-value |
|---|---|---|---|
| 1 | 6.12% | 1.76 | 0.04 |
| 2 | -2.87% | -0.43 | 0.34 |
| 3 | 2.27% | 0.71 | 0.24 |
| 4 | -3.73% | -0.67 | 0.25 |
| 5 | 3.55% | 0.52 | 0.30 |
| 6 | 0.33% | 0.10 | 0.46 |
| 7 | -0.68% | -0.19 | 0.42 |
| 8 | 7.39% | 1.48 | 0.07 |
| Total | 1.94% | 0.63 | 0.27 |

Table 4.3: One-tailed t-test on portfolio alpha without transaction costs

From Table 4.3, we can see all the alpha-coefficients from every test period. Looking at the average alpha for the entire period, it states that we obtained an annualized positive alpha of 1.94%. At a first impression, this means that our strategy yielded excess return. However, by looking at the p-value of 0.27 and using a confidence interval of 95%, we fail to reject the null hypothesis in favor of the alternative. The alternative hypothesis is in contrast to the null, stating that the pairs trading strategy did yield excess return. Even though our alpha was positive, we cannot conclude that this was due to our research design. Furthermore, a crucial assumption regarding hypothesis testing is whether our data is normally distributed. From Figure 4.5, we can see that the daily return is close to normally distributed.

[1]The alpha coefficients represent the $\pm$ excess return from each training period and are obtained from the CAPM-equation. The t-statistics are based on the average daily alpha in each period.

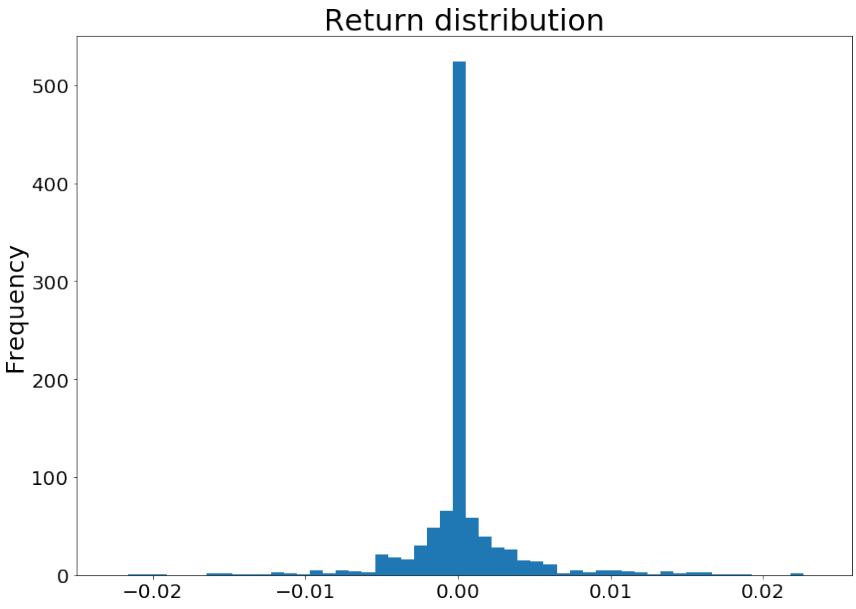



Figure 4.5: Daily return distribution over the sample period

Even though we did not generate excess return, the results obtained from our pairs trading strategy suggest that the risk-exposure is close to zero as presented in Table 4.4. Three out of eight betas have a p-value below 5%, which means they have a genuine effect on the dependent variable. This is also the case for the overall portfolio beta, which has a p-value of 4%. Nevertheless, with an overall portfolio beta of -0.03, our pairs trading strategy bears little systematic risk. From Table 4.4, we can see the summary statistics from all the betas during the eight testing periods, including the overall portfolio beta.

| Period | $\beta$ Coefficient | t | P-value |
|---|---|---|---|
| 1 | -0.16 | -3.72 | 0.00 |
| 2 | 0.03 | 0.70 | 0.48 |
| 3 | -0.08 | -2.58 | 0.01 |
| 4 | -0.04 | -1.31 | 0.19 |
| 5 | 0.01 | 0.26 | 0.79 |
| 6 | -0.02 | -0.59 | 0.56 |
| 7 | -0.04 | -1.23 | 0.22 |
| 8 | -0.17 | -2.51 | 0.01 |
| Total | -0.03 | -2.12 | 0.04 |

Table 4.4: Two-tailed t-test on portfolio beta

### 4.3.2 Summary of the Results with Transaction Costs

As discussed earlier in the previous chapter, the inclusion of transaction costs is of great importance. Our pairs trading strategy has multiple trades every month. Thus, transaction costs have a huge impact on the results obtained. Table 4.5, presents the summary statistics when we deduct for transaction costs.

| *Period* | *Nb Pairs* | $R_p$ | $\sigma_p$ | $SR_p$ | $R_{BM}$ | $\sigma_{BM}$ | $SR_{BM}$ |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 3.66% | 3.62% | 0.75 | 12.97% | 7.79% | 1.54 |
| 2 | 4 | -3.54% | 6.80% | -0.66 | -7.46% | 12.68% | -0.66 |
| 3 | 12 | 1.00% | 3.18% | 0.01 | 8.59% | 10.31% | 0.74 |
| 4 | 1 | -3.28% | 5.44% | -0.78 | -3.11% | 14.24% | -0.29 |
| 5 | 5 | 2.86% | 7.32% | 0.26 | 0.00% | 17.87% | -0.03 |
| 6 | 7 | -0.65% | 3.00% | -0.54 | 11.82% | 9.47% | 1.15 |
| 7 | 13 | -1.20% | 3.07% | -0.71 | 1.02% | 7.90% | 0.01 |
| 8 | 11 | 4.23% | 5.16% | 0.63 | 15.51% | 7.18% | 2.02 |
| Total | 60 | 0.69% | 6.82% | -0.07 | 10.42% | 16.18% | 0.52 |

Table 4.5: Summary statistics of portfolio with transaction costs

Once the deduction of transaction costs was implemented, the strategy yielded a compounded annualized return of 0.69% with a corresponding Sharpe ratio of -0.07. A negative Sharpe ratio implies that we obtained a negative risk premium per unit of risk.

Correspondingly, we also tested whether the strategy was able to generate excess return with transaction costs. Table 4.6 indicates the t-statistics of the portfolio alpha with transaction costs. With the inclusion of transaction costs, the annualized alpha is -1.01%. Moreover, the annualized alpha displays a p-value of 0.41 and, therefore, we fail to reject the same null hypothesis in favor of the alternative. This means that our pairs trading strategy were not able to generate excess return when deducted for transaction costs.

| Period | $\alpha$ Coefficient | t | P-value |
|---|---|---|---|
| 1 | 4.61% | 1.35 | 0.09 |
| 2 | -4.25% | -0.65 | 0.26 |
| 3 | 0.53% | 0.22 | 0.41 |
| 4 | -4.41% | -0.79 | 0.21 |
| 5 | 1.89% | 0.30 | 0.38 |
| 6 | -1.40% | -0.46 | 0.32 |
| 7 | -2.16% | -0.68 | 0.25 |
| 8 | 5.73% | 1.15 | 0.13 |
| Total | -1.01% | -0.22 | 0.41 |

Table 4.6: One-tailed t-test on portfolio alpha with transaction costs

Last, we display the cumulative return for all periods. This includes the pairs trading strategy with and without transaction costs, but also with the comparison of the benchmark index. This is presented in Figure 4.6.

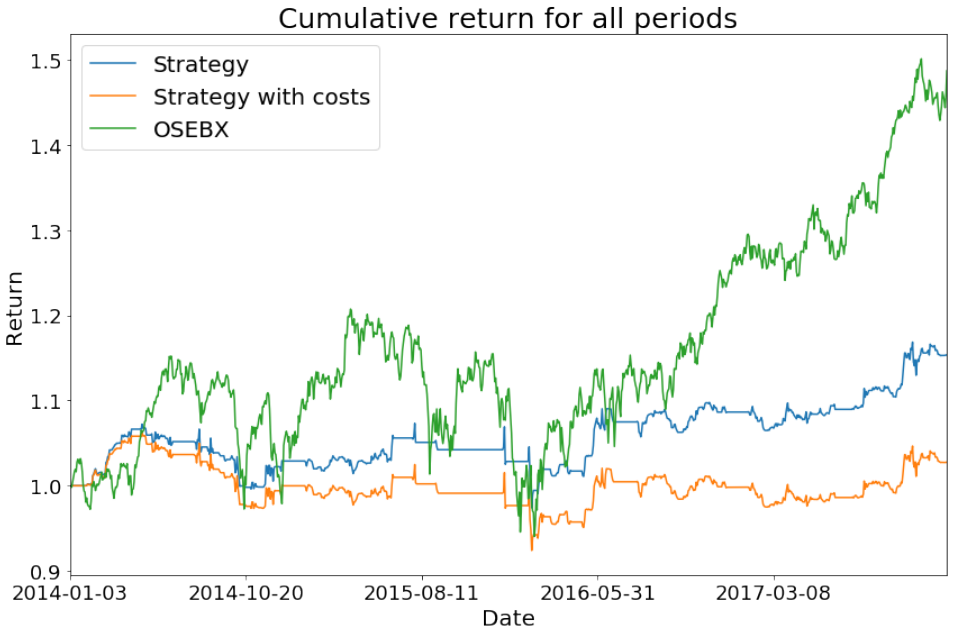



Figure 4.6: Strategy with and without transaction costs compared to benchmark

### 4.3.3 Comparing the Strategy with an Unrestricted Model

In order to verify our research design, we will compare it to an unrestricted model. As emphasized throughout this thesis, PCA and DBSCAN are implemented prior to the identification of mean-reversion. We thus created a restricted model with the intention of extracting and clustering stocks with similar risk exposure, in the belief they would be more suitable for a pairs trading strategy. For the unrestricted model, we have disregarded any sort of stock filtering and tested all stock combinations for cointegration. The results are presented in Table 4.7 and Figure 4.7.

| | **Unrestricted Model** | | | | | | **Restricted Model** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Period* | *Nb Pairs* | $R$ | $\sigma$ | $SR$ | $\beta$ | $\alpha$ | *Nb Pairs* | $R$ | $\sigma$ | $SR$ | $\beta$ | $\alpha$ |
| 1 | 26 | 2.10% | 3.66% | 0.31 | -0.12 | 2.57% | 7 | 5.17% | 3.62% | 1.16 | -0.16 | 6.12% |
| 2 | 9 | -22.48% | 78.05% | -0.30 | -0.17 | -24.88% | 4 | 2.16% | 6.75% | -0.46 | 0.03 | -2.87% |
| 3 | 30 | 6.24% | 9.91% | 0.53 | -0.13 | 6.26% | 12 | 2.63% | 3.19% | 0.52 | -0.08 | 2.27% |
| 4 | 35 | -5.71% | 8.82% | -0.90 | -0.16 | -7.33% | 1 | -2.61% | 5.45% | -0.66 | -0.04 | -3.74% |
| 5 | 40 | -5.29% | 19.14% | -0.41 | -0.16 | -5.41% | 5 | 4.51% | 7.32% | 0.48 | 0.01 | 3.55% |
| 6 | 60 | 19.68% | 9.12% | 1.83 | 0.05 | 18.17% | 7 | 1.08% | 2.96% | 0.04 | -0.02 | 0.33% |
| 7 | 85 | 2.09% | 6.95% | 0.16 | 0.05 | 1.18% | 13 | 0.29% | 3.07% | -0.22 | -0.04 | -0.68% |
| 8 | 41 | 2.44% | 5.62% | 0.26 | -0.10 | 2.92% | 11 | 5.89% | 5.17% | 0.95 | -0.17 | 7.39% |
| Total | 326 | -3.07% | 24.91% | -0.20 | -0.27 | -2.73% | 60 | 3.64% | 6.81% | 0.25 | -0.03 | 1.94% |

Table 4.7: Summary statistics of Unrestricted and Restricted model without transaction costs

In comparison, the number of stock pairs is over five times greater for the unrestricted model. This is no surprise, as we have penalized our restricted model with stern criteria through unsupervised machine learning techniques. Even though the unrestricted model created more stock pairs, the risk exposure is larger. This can be seen from the overall portfolio beta of -0.27 and standard deviation of 24.91%.

In the unrestricted model, the strategy yielded a negative annualized return of -3.07%, roughly six percentage point below the annualized return of the restricted model. This resulted in a negative Sharpe ratio of -0.20. Moreover, the annualized alpha was -2.73%. Once tested through a t-test, we again fail to reject the null hypothesis in favor of the alternative. Thus,

we can conclude that the unrestricted model did not produce any excess return.

Figure 4.7 outlines the strategy comparison between the restricted and unrestricted model.

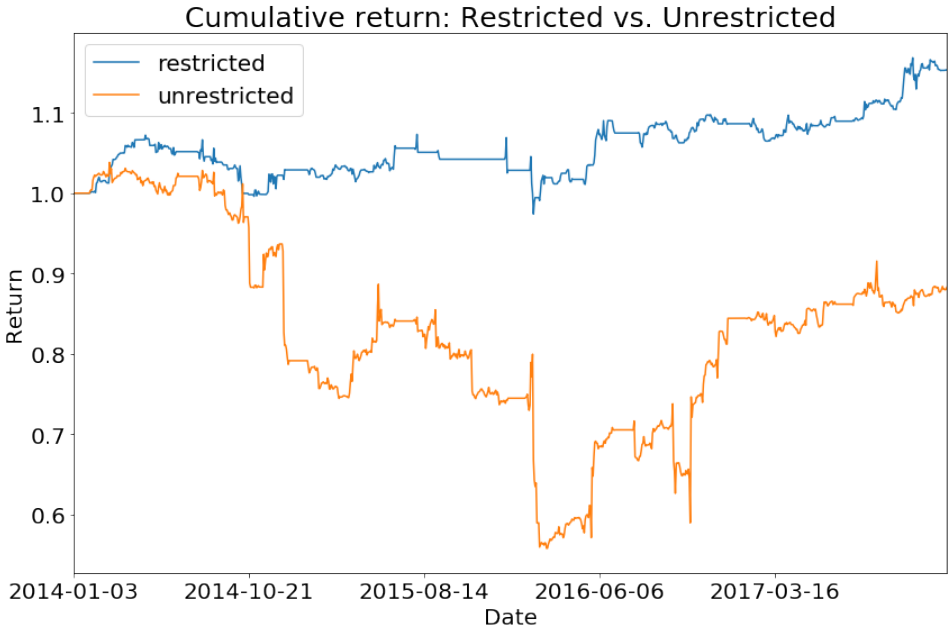



Figure 4.7: Restricted and Unrestricted model comparison

## 4.4 Empirical Summary

The number of principal components was settled at 12 in every training period. This allowed us to capture about 75% of the variation in the dataset. The principal components, combined with fundamental ratios, was then clustered in regions of high density where stocks with similar risk exposure where clustered together. This resulted in one or two clusters in every period.

From the results obtained, we were not able to statistically prove that the pairs trading strategy was able to generate excess returns from arbitrage. This was the result both with and without transaction costs. An unrestricted model was designed as a comparison to the original (restricted) model. Results showed that the returns where almost six percentage points under the restricted model. Moreover, this model was also not able to generate excess return.

# Chapter 5

# Discussion

In this chapter, we will try to reconnect the empirical findings with the outlined theoretical frameworks. The main theoretical frameworks are the efficient-market hypothesis, the arbitrage pricing theory, and pairs trading. We have therefore structured the discussion in this manner.

## 5.1 The Efficient-Market Hypothesis

The first theoretical point of departure was the efficient-market hypothesis (EMH), which stated that the value of a transaction is at its correct value because all of the information is fully reflected in the price (Fama, 1970). Moreover, the theory outlines the concept of no-arbitrage, stating that stock prices follow a random walk and, as a consequence, one cannot exploit mispricing of stocks. Our empirical results support this theory. The purpose of t-testing the alpha of the portfolio was to determine if our statistical arbitrage strategy were able to generate excess return on the Norwegian stock market. Even though we obtained an alpha of 1.94%, we failed to reject the null hypothesis in favor of the alternative and the alpha was therefore not statistically significant. Thus, our findings are in line with Fama (1970) and the EMH. This means that changes in stock prices cannot be reflected in algorithms, while excess return is gained as luck rather than an outcome of a correct prediction (Degutis and Novickyt, 2014).

Furthermore, our study leads to the questioning of exactly how efficient the Norwegian stock

market was during our sample period. As aforementioned, Fama (1970) outlines three kinds of market efficiency based on the information set, namely the weak, semi-strong and strong. The strong form postulates that all information, both public and private is reflected in the stock price. It is hard to argue for the latter because this implies that stock prices also includes inside information. Because of this uncertainty, the Norwegian stock market cannot be described as having a strong market efficiency. On the contrary, we can conclude that the OSEBX is an example of the weak-form efficiency since we were not able to generate excess return on historical data. The question is whether we can characterize it to be on the semi-strong form, where stock prices reflect both past and present public information. Bodie et al. (2011) argues that the semi-strong form is where all investors do not interpret the information in the same way, possibly leading to adverse situations from technical or fundamental analysis. We will, therefore, conclude that the Norwegian stock market is at a semi-strong form because even though we were not able to generate an overall positive portfolio alpha, there where some periods with positive significant results, which gives indications that there are arbitrage opportunities every now and then.

Looking at the empirical results from a broader perspective, we can see that investing in an actively managed portfolio will not outperform the market. Since we conclude that the Norwegian stock market is (semi-form) efficient, there is no incentive to spend resources to gather information and trade on it. Looking at the evidence, there is not much research in favor of active management. Some studies reveal that active management before cost may give the same return as passive investing. However, when deducting for costs associated with active management, it turns out that passive management gives the highest return (Fama and French, 2010), which is consistent with our results.

## 5.2 The Arbitrage Pricing Theory

Having discussed the results in light of the EMH, the second theoretical point of departure was the Arbitrage Pricing Theory. In our pairs trading strategy, we sought to extract common underlying risk factors of stock returns in the understanding that these could lead to more robust cointegrated stock pairs. This was because the APT uncovered the heart of pairs trading because it stated that stocks with similar risk exposure would yield the same return and, as a consequence, one could use the APT theory as a catalyst to find mispriced stocks and place trades accordingly (Harlacher, 2016).

In the pursuit of extracting the underlying risk factors, we used the principal component analysis to avoid ad hoc choices of factors. Nevertheless, in PCA, the components obtained are uninterpretable because completely new variables are created in a new subspace. This means that we have to trust that our research design is conducted in an adequate manner without any form of misspecifications. Moreover, when deciding the number of factors (principal components), we looked for the point at which the eigenvalues dropped and became insignificant, namely an ad hoc choice of the number of factors. This could, therefore, bias our results in a way that we captured too much or too little of the explained variance. Because almost half of the cointegrated pairs ceased to exist out-of-sample, it gives us an indication that the number and nature of the extracted factors were collected on false premises.

Notwithstanding, when we compared our strategy against the unrestricted model, it gives us confidence that PCA and DBSCAN were able, to a certain degree, to filter out stocks suitable for pairs trading. In the words of Avellaneda and Lee (2008), PCA decomposes and extract risk components, as a way to sort out the idiosyncratic noise. Clearly, this is evident in the restricted model when comparing it to the unrestricted model. Under those circumstances, we can infer that the cointegrated relationship between the stocks in the unrestricted model has further disintegrated out-of-sample compared to our restricted model. In this regard, we have reason to believe that filtering out stocks by using unsupervised machine learning techniques and including financial ratios are in fact beneficial when designing a statistical arbitrage strategy.

## 5.3 Pairs Trading

The last theoretical point of departure, was the empirical evidence of pairs trading. During our investment horizon, a total of 60 pairs were traded, 35 of which ended with positive returns. This proves that almost half of all pairs diverged more than desired out-of-sample and negatively affected the overall performance. This confirms the unfavorable profits due to diverging properties and advances in technology as documented by Do and Faff (2010). There may be various reasons for declining profits, but we will outline three distinct motives presented in previous literature, which we believe is relevant to our strategy.

1. First, a disintegration between two cointegrated stocks can be due to a structural shock or break, as first outlined by Hamilton (1992). This means that the stocks are still inherently cointegrated, but with different parameter values which our model is not able to capture.

2. Second, De Bondt and Thaler (1985) argue that stock returns have a tendency to reverse itself over the long run. They call it the reversal effect, an effect in which losers rebound and winners fade back. Thus, if our testing periods are too short, we may not be able to generate profit from stock pairs that take longer to converge back to their long-term mean.

3. Last, the original article of Gatev et al. (1999), which has been in circulation for a long time, may have contributed to a great enthusiasm for the use of pairs trading strategies among sophisticated investors. Increased competition coupled with the development of technology, might have vanished the profitability away.

In the light of pairs trading as a relative value investment, the strategy may, therefore, have seen its glory days. As more and more investors use similar strategies and chase the same price discrepancies, such mispricing will be eliminated more quickly, thus leading to lower returns. Therefore, it does not come as a surprise that we were not able to achieve excess return.

As presented in the results, we found that our pairs trading strategy is close to market neutral with an overall portfolio beta of -0.03, which is in accordance with Gatev et al. (2006). On the contrary, we saw great fluctuations in the OSEBX over the sample period. This is likely

to be linked with the historical volatility in oil prices. Since the Norwegian stock market is heavily exposed to the oil and gas industry, it is reasonable to assume that the high alteration in the oil and gas industry was probably rippling over to other sectors, resulting in negative repercussions and skepticism among equity investors at the OSEBX at the beginning of the sample period.

Nevertheless, our pairs trading risk-exposure remained stable throughout the turmoil. It also generated higher returns than the OSEBX in three out of the first five testing periods. This is to some extent consistent with what we mentioned in the empirical evidence of pairs trading. Namely that the strategy works particularly well in times of uncertainty and financial crises. During such adverse situations, there is usually a shortage of liquidity which can cause stock prices to deviate from its fundamental value. Stocks which usually follows the same stochastic process can suddenly diverge because liquidity is of dire constraints in the market. Such situations can be a good opportunity to increase pairs trading activities. By conducting pairs trading in these situations, one can take the role as an intermediary or liquidity provider. This is because we buy the underperforming stock and sell the stocks that do relatively well. Thus, we can see that the return from our pairs trading strategy was negatively correlated in times of financial turmoil, as outlined by Acharya and Pedersen (2005). From these findings, we believe pairs trading can be used as a mean to hedge systematic risk, especially in times of financial distress.

Although Do and Faff (2010) found declining profit in pairs trading, we hoped that the performance of our strategy would match the results obtained by Avellaneda and Lee (2008) and Avalon et al. (2017), which found favorable results with the use of unsupervised machine learning techniques. However, the studies of both (Avellaneda and Lee, 2008) and (Avalon et al., 2017) were conducted on the S&P 500. This is considered to be the most liquid and traded stock market in the world. It is, therefore, natural that such a strategy would yield different results on the Norwegian stock market. A market with more homogeneous companies. Conducting the algorithmic trading model as presented in this thesis may, therefore, be more beneficial for a larger stock universe. With a larger stock universe, PCA and DBSCAN would most likely result in a larger number of clusters, which again could provide more robust stock pairs.

## 5.4 Discussion Summary

Based on our analysis, the results are in accordance with the efficient-market hypothesis (EMH). Even though we obtained a relative market-neutral strategy with a beta close to zero, we were not able to gain a statistically significant excess return on the Norwegian stock market. With the rise of algorithmic trading and advanced computerized systems to analyze stock movements, we hoped to capitalize on market inefficiencies and asymmetric information on which such technology could bring. On the contrary, because of the improved technological novelties, it seems that the market is more efficient than we sought out. This means that the efficient-market hypothesis relevancy is more present than ever and, as a consequence, we can conclude that *there is no such thing as a free lunch.*

# Chapter 6

# Conclusion

## 6.1 Summary

At the beginning of this thesis, we outlined the idea of no-arbitrage saying that *there is no such thing as a free lunch*. Determined to use statistical algorithms to find a free lunch on the Norwegian stock market, we imposed the following research question:

**Can algorithmic pairs trading with machine learning generate excess return?**

Our main purpose was, therefore, to establish and create a pairs trading algorithm to capitalize on market inefficiencies. In the seek of the research question we used three unsupervised machine learning techniques to extract and cluster stocks with the same underlying risk factors. Thereafter, we identified cointegration among the stocks and placed trades accordingly. Based on the empirical results obtained, we can draw the following conclusion:

1. There seems not to be any arbitrage opportunities on the Norwegian stock market.
2. The stock market is efficient in the semi-strong form.
3. Pairs trading is a market-neutral strategy with almost no exposure to systematic risk
4. Unsupervised machine learning techniques have properties which are beneficial for a pairs trading strategy

We found that the strategy is close to market neutral with a statistically significant beta value of -0.03. Notwithstanding, with the implementation of transaction costs, the strategy only obtained an alpha of -1.02% and return of 0.69% p.a. Thus, we statistically concluded that pairs trading do not generate excess return on the Norwegian stock market in the sample period from January 2013 to December 2017. This is in line with the efficient-market hypothesis. Nevertheless, our strategy outperformed an unrestricted model based solely on cointegration. This reveals that extracting common underlying factors of stock returns and cluster them into regions of high density have properties that are beneficial for selecting stocks for a pairs trading strategy. On these premises, we can, therefore, inform and verify: *there is no such thing as a free lunch*.

## 6.2 Limitations and Future Work

One of the limitations of our research is that we have applied our research design on the Norwegian stock market over a relatively short period of time. We only conclude that there seems not to be any arbitrage on the Norwegian stock market in the given period. Longer or other sample periods could have provided different conclusions. Second, we concluded to use the same parameters in the machine learning algorithms for all periods. This could have biased our results as periods most likely changes throughout time.

Furthermore, during our pairs trading analysis, we identified aspects which can be considered in future work. We will outline the five aspects which we believe are the most relevant.

First, the strategy could be examined over a longer period or in two different decades. This could potentially give us an indication if there is declining profitability of pairs trading on the Norwegian stock market.

Second, since our strategy relies on cointegrated stocks in-sample, they may lose this relationship out of sample. As a result, adding adaptive trading rules could be taken into account. Adding a stop-loss function to our algorithm could probably make the strategy more robust.

Third, we assume that all stocks at the OSEBX are possible to short. This is not necessarily always the case in real life due to the liquidity in the stock themselves. Therefore, adding liquidity requirements to the model could be investigated in greater detail.

Fourth, in our thesis, we used adjusted closing prices since this is the common practice in the pairs trading literature. Implementing intra-day prices instead of closing prices would potentially give a more realistic view of the trade executions and transaction costs.

Last, the use of machine learning in pairs trading was very useful compared to an unrestricted model. However, it did not help us to achieve excess returns. It would, therefore, be interesting to see if other techniques would have provided better results. This could be through the use of the supervised techniques Support Vector Machines or Neural Networks. In greater detail, these techniques allow us to categorize unlabeled data and is one of the most widely used clustering algorithms in industrial applications.

# References


Acharya, V. and Pedersen, L. (2005). Asset pricing with liquidity risk. *The Journal of Financial Economics*, 77:375–410.

Andrade, S. C., Pietro, V. d., and Seasholes, M. S. (2005). Understanding the Profitability of Pairs Trading. *UC Berkeley Working paper.*

Avalon, G., Becich, M., Cao, V., Jeon, I., Misra, S., and Puzon, L. (2017). Multi-factor Statistical Arbitrage Model. *Stanford University.*

Avellaneda, M. and Lee, J.-H. (2008). Statistical arbitrage in the u.s. equities market. *Quantitative Finance*, 10(7):761–782.

Avolio, G. D., Gildor, E., and Shleifer, A. (2002). Technology, information production, and market efficiency. *Economic Policy for the Information Economy. Federal Reserve Bank of Kansas City.*

Björk, T. (1998). *Arbitrage theory in continuous time.* Oxford University Press Inc, Great Clarendon Street, Oxford OX2 6DP.

Bodie, Z., Kane, A., and Marcus, A. (2011). *Investments and Portfolio Management.* McGraw Hill/Irwin, New York.

Bodie, Z., Kane, A., and Marcus, A. (2014). *Investments.* McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121, 10th edition.

Bogomolov, T. (2013). Pairs trading based on statistical variability of the spread process. *Quantitative Finance*, 13(9):1411–1430.

Broussard, J. P. and Vaihekoski, M. (2012). Profitability of pairs trading strategy in an illiquid market with multiple share classes. *The Journal of International Financial Markets*, 22(5):1188–1201.

Caldeira, J. and Moura, G. (2013). Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy. *Rev. Bras. Financas (Online), Rio de Janeiro*, 11(1):49–80.

Chamberlain, G. and Rothschild, M. (1983). Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51:1281–304.

De Bondt, W. F. M. and Thaler, R. (1985). Does the stock market overreact? *The Journal of Finance*, 40(3):361–372.

Degutis, A. and Novickyt, L. (2014). The efficient market hypothesis: A critical review of literature and methodology. *Ekonomika*, 92(2).

Derksen, L. (2006). Visualising high-dimensional datasets using pca and t-sne in python. Retrieved from: https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association.*

Do, B.-H. and Faff, R. W. (2010). Does simple pairs trading still work? *The Financial Analyst Journal*, 66(4):83–95.

Engle, R. and Granger, C. (1987). Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 96:226–231.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

Fama, E. F. and French, K. R. (2010). Luck versus skill in the crosssection of mutual fund returns. *The Journal of Financial*, 65(5).

Gatev, E., Goetzmann, W. N., and Geert Rouwenhorst, K. (1999). Pairs trading: Performance of a relative value arbitrage rule. *Working paper*, Yale School of Managements International Center for Finance.

Gatev, E., Goetzmann, W. N., and Geert Rouwenhorst, K. (2006). Pairs trading: Performance of a relative value arbitrage rule. *Review of Financial Studies*, 19(2):797–827.

Goulb, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, 2715 North Charles Street.

Hamilton, J. D. (1992). *Time Series Analysis*. Princeton University Press, Princeton.

Harlacher, M. (2016). *Cointegration Based Algorithmic Pairs Trading*. PhD thesis, University of St. Gallen.

Hogan, S., Jarrow, R., Teo, M., and Warachka, M. (2004). Testing market efficiency using statistical arbitrage with applications to momentum and value strategies. *The Journal of Financial Economics*, 73(3):525–565.

Jaeger, M. (2016). Dynamic cointegration based pairs trading. Master's thesis, Copenhagen Business School.

Jaju, S. (2017). Comprehensive guide on t-sne algorithm with implementation in r and python. Retrieved from: https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/.

James, G., Witten, D., Travor, H., and Tibshirani, R. (2013). *An Introduction to Statistical Learning - with Applications in R*. Springer, New York.

Jensen, M. (1978). Some anomalous evidence regarding market efficiency. *The Journal of Financial Economics*, 6(2):95–101.

Jolliffe, I. (2002). *Principal Component Analysis*. Springer-Verlag, New York.

Kakushadze, Z. (2014). Mean-reversion and optimization. *The Journal of Asset Management*, 16(1):14–40.

Kim, D.-H. and Jeong, H. (2005). Systematic analysis of group identification in stocks markets. *The Physical Review*, 72:046133.

Larkin, J. (2017). Pairs trading with machine learning. Quantopian Inc.
Retrieved from: https://www.quantopian.com/posts/pairs-trading-with-machine-learning.

Law, J. and Smullen, J. (2008). *A Dictionary of Finance and Banking*. Oxford University Press, 4th edition.

Lazzarino, M., Berrill, J., and evi, A. (2018). What Is Statistical Arbitrage? *Theoretical Economics Letters*, 8:888–908.

Lindström, E., Madsen, H., and Nielsen, J. N. (2015). *Statistics for Finance*. Taylor and Francis Group.

Mackenzie, D. and Margenot, M. (2018). Introduction to pairs trading.
Retrieved from: https://www.quantopian.com/lectures/introduction-to-pairs-trading.

MacKinnon, J. G. (2010). Critical values for cointegration tests. *Queens Economics Department Working Paper No. 1227*.

Madsen, R. E., Hansen, L. K., and Winther, O. (2004). Singular value decomposition and principal component analysis. Technical report, Neural Networks. Retrieved from: http://www2.imm.dtu.dk/pubdb/p.php?4000.

Malkiel, B. (2005). Reflections on the efficient market hypothesis: 30 years later. *The Financial Review*, 40(1):1–9.

Markowitz, H. (1953). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Nordnet (2018). Prismodeller.
Retrieved from: https://www.nordnet.no/tjenester/prisliste.html.

Norwegian Central Bank (2018). Government bonds annual average. Retrieved from: https://www.norges-bank.no/en/Statistics/Interest-rates/Government-bonds-annual/.

Perlin, M. (2009). Evaluation of pairs trading strategy at the brazilian financial market. *The Journal of Derivatives and Hedge Funds*, 15:122–136.

Pesaran, M. H. (2010). Predictability of asset returns and the efficient market hypothesis. *Discussion papaer*, University of Cambridge.

Poitrast, G. (2010). Arbitrage: Historical perspectives. *Encyclopedia of Quantitative Finance*.

Poterba, J. and Summers, L. (1988). Mean reversion in stock prices: Evidence and implications. *The Journal of Financial Economics*, 22:22–59.

Puntanen S., Styan G.P.H., I. J. (2011). *Eigenvalue Decomposition. In: Matrix Tricks for Linear Statistical Models*. Springer, Berlin, Heidelberg.

Raschka, S. (2017). *Python Machine Learning: Learning and Deep Learning with Python, scikit-learn and TensorFlow*. Packt, New York, 2nd edition.

Reverre, S. (2001). *The Complete Arbitrage Deskbook*. McGraw-Hill Education, 1st edition.

Rosenius, N. and Sjőholm, G. (2013). Arbitrage opportunities on the omxs: How to capitalize on the ex-dividend effect. Master's thesis, Ume School of Business and Economics.

Ross, S. (1975). The arbitrage theory of capital asset pricing. *The Journal of Economic Theory*, 13(3):341–360.

Sandberg, H. (2004). *Model Reduction for Linear Time-Varing Systems*. Department of Automatic Control, Lund Institute of Technology (LTH).

Shukla, R. (1997). An empiricists guide to the arbitrage pricing theory. Syracuse University Syracuse, NY 13244.

Tan, J. (2012). Principal component analysis and portfolio optimization. Available at SSRN: https://ssrn.com/abstract=2213687 or http://dx.doi.org/10.2139/ssrn.2213687.

Thapa, C. and Sunil, S. P. (2010). International equity portfolio allocations and transaction costs. *The Journal of Banking and Finance*, 34(11):2627–2638.

Timmermann, A. and Granger, C. W. (2004). Efficient market hypothesis and forecasting. *The International Journal of Forecasting*, 20:15–27.

Trana, T. N., Nguyen, T. T., Willemsz, T. A., van Kessel, G., Frijlink, H. W., and van der Voort Maarschalk, K. (2012). A density-based segmentation for 3d images, an application for x-ray micro-tomography. *Analytica Chimica Acta*, 725:14–21.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9:2579–2605.

Vidyamurthy, G. (2004). *Pairs Tading: Quantitative Methods and Analysis*. John Wiley and Sons, Canada.

Wooldridge, J. M. (2009). *Introductory Econometrics*. South-Western Cengage Learning, 5191 Natorp Boulevard Mason, OH 45040 USA, 4th edition.

# Chapter 7

# Appendices

## 7.1 The different forms of market efficiency

| Market Efficiency | Assumption | Arbitrage opportunity |
|---|---|---|
| Strong efficiency | Public and private information reflected in stock prices | No arbitrage |
| Semi-strong efficiency | Public and past information reflected in stock prices | Only on non-public information |
| Weak efficiency | Historical prices reflected in stock prices | On fundamental and inside information |

**Source**: Rosenius and Sjőholm (2013)

Table 7.1: Different forms of market efficiency

## 7.2 Returns, Volatility and Sharp ratio

### 7.2.1 Return and volatility calculation

The Portfolio Return for each period is calculated as follows:

$$R_{port} = \sum_{i=1}^{N} \omega_i R_i, \tag{7.1}$$

where $w_i$ represent the weight allocated to each stock pair and $R_i$ represents the realized pair return.

The Portfolio Volatility for each period is calculated as follows:

$$\sigma_p^2 * days = \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_i \text{cov}(i,j) \omega_j * days = \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_i \sigma_{i,j} \omega_j * days, \tag{7.2}$$

where $\sigma_p$ represents the standard deviation of the portfolio in each period.

### 7.2.2 Return, Volatility and Sharpe Ratio for each stock pair

| Period 1 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| BAKKA-GJF | 0.31% | 11.84% | -0.06 | -1.35% | 11.93% | -0.19 |
| GJF-MHG | -1.73% | 3.00% | -0.90 | -2.82% | 3.02% | -1.25 |
| KOG-MHG | 5.09% | 12.33% | 0.33 | 3.36% | 12.29% | 0.19 |
| KOG-SALM | 6.07% | 3.26% | 1.56 | 4.90% | 3.32% | 1.18 |
| KOG-TEL | 8.45% | 6.43% | 1.16 | 7.25% | 6.41% | 0.98 |
| MHG-TEL | 14.23% | 7.06% | 1.88 | 12.5% | 7.01% | 1.64 |
| MHG-WWIB | 3.79% | 9.46% | 0.30 | 1.80% | 9.39% | 0.09 |
| Portfolio | 5.17% | 3.62% | 1.16 | 3.66% | 3.61% | 0.75 |

Table 7.2: Equal weighted portfolio period 1

| Period 2 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| AKER-OLT | -23.00% | 22.09% | -1.08 | -24.71% | 22.21% | -1.16 |
| MHG-OLT | 0.66% | 5.52% | -0.06 | -0.18% | 5.58% | -0.21 |
| OLT-ORK | 10.16% | 12.73% | 0.72 | 8.64% | 12.73% | 0.60 |
| WWI-WWIB | 3.52% | 4.49% | 0.57 | 2.08% | 4.56% | 0.24 |
| Portfolio | -2.16% | 6.75% | -0.46 | -3.54% | 6.80% | -0.66 |

Table 7.3: Equal weighted portfolio period 2

| Period 3 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| AFG-TEL | 11.95% | 10.58% | 1.04 | 10.11% | 10.51% | 0.87 |
| ATEA-GSF | 38.35% | 19.82% | 1.89 | 35.31% | 19.83% | 1.73 |
| ATEA-OLT | -1.79% | 8.64% | -0.32 | -3.54% | 8.53% | -0.53 |
| ATEA-TEL | -3.45% | 16.43% | -0.27 | -4.78% | 16.40% | -0.35 |
| ATEA-YAR | 6.54% | 5.20% | 1.07 | 5.08% | 5.11% | 0.80 |
| GSF-OLT | -2.29% | 10.16% | -0.32 | -4.04% | 10.16% | -0.49 |
| GSF-TEL | 3.18% | 9.20% | 0.24 | 1.47% | 9.22% | 0.05 |
| GSF-VEI | -4.97% | 7.66% | -0.78 | -6.02% | 7.61% | -0.92 |
| OLT-TEL | -4.41% | 11.05% | -0.49 | -5.72% | 11.02% | -0.61 |
| OLT-YAR | 5.61% | 6.11% | 0.75 | 4.14% | 6.20% | 0.51 |
| TEL-VEI | -16.92% | 11.36% | -1.57 | -18.53% | 11.42% | -1.71 |
| WWI-WWIB | -0.26% | 3.48% | -0.35 | -1.50% | 3.54% | -0.70 |
| Portfolio | 2.63% | 3.19% | 0.52 | 1.00% | 3.18% | 0.01 |

Table 7.4: Equal weighted portfolio period 3

| Period 4 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| WWI-WWIB | -2.61% | 5.45% | -0.66 | -3.28% | 5.44% | -0.78 |
| Portfolio | -2.61% | 5.45% | -0.66 | -3.28% | 5.43% | -0.78 |

Table 7.5: Equal weighted portfolio period 4

| Period 5 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| GSF-ORK | 8.58% | 8.89% | 0.86 | 7.38% | 8.89% | 0.72 |
| GSF-VEI | 11.51% | 8.18% | 1.29 | 9.37% | 8.14% | 1.03 |
| SALM-SCHA | 20.39% | 9.48% | 2.05 | 18.90% | 9.45% | 1.90 |
| SCHA-VEI | -18.72% | 27.94% | -0.70 | -20.06% | 27.87% | -0.75 |
| SNI-WWI | 0.80% | 13.60% | -0.01 | -1.29% | 13.65% | -0.17 |
| Portfolio | 4.51% | 7.32% | 0.48 | 2.86% | 7.32% | 0.26 |

Table 7.6: Equal weighted portfolio period 5

| Period 6 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| EKO-OLT | 3.56% | 6.00% | 0.43 | 1.57% | 5.90% | 0.10 |
| GJF-XXL | -10.72% | 5.71% | -2.06 | -11.96% | 5.81% | -2.23 |
| NHY-VEI | 6.87% | 7.51% | 0.78 | 5.39% | 7.53% | 0.59 |
| SNI-TGS | -0.53% | 9.21% | -0.16 | -2.59% | 9.32% | -0.38 |
| STB-VEI | 0.54% | 6.08% | -0.07 | -1.13% | 6.07% | -0.34 |
| STB-XXL | 13.22% | 10.24% | 1.20 | 11.03% | 10.29% | 0.98 |
| VEI-XXL | -5.38% | 8.54% | -0.74 | -6.83% | 8.70% | -0.90 |
| Portfolio | 1.08% | 2.96% | 0.04 | -0.65% | 3.01% | -0.54 |

Table 7.7: Equal weighted portfolio period 6

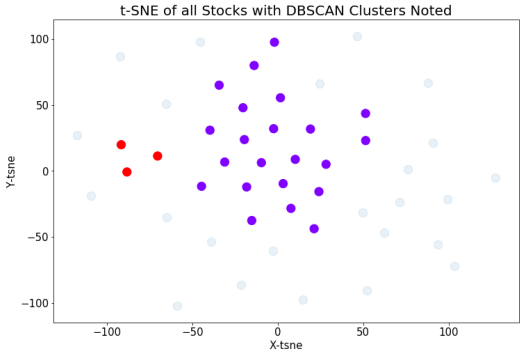
| Period 7 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| AFG-ENTRA | -3.01% | 5.47% | -0.74 | -4.39% | 5.40% | -0.99 |
| ATEA-TEL | 4.80% | 13.25% | 0.29 | 2.51% | 13.15% | 0.12 |
| BAKKA-EKO | -6.01% | 16.36% | -0.43 | -7.42% | 16.25% | -0.52 |
| EKO-ENTRA | 15.52% | 16.82% | 0.86 | 13.93% | 16.81% | 0.77 |
| EKO-MHG | 4.53% | 10.18% | 0.35 | 2.81% | 10.11% | 0.18 |
| EKO-NHY | -23.12% | 24.39% | -0.99 | -24.31% | 24.44% | -1.03 |
| ENTRA-OLT | -0.48% | 3.09% | -0.47 | -1.58% | 3.15% | -0.81 |
| EPR-GJF | -2.25% | 5.71% | -0.62 | -4.43% | 5.67% | -0.95 |
| EPR-XXL | 6.21% | 4.42% | 1.19 | 4.75% | 4.38% | 0.86 |
| GJF-TEL | -2.38% | 5.77% | -0.58 | -4.27% | 5.82% | -0.90 |
| GJF-XXL | -3.30% | 5.24% | -0.82 | -4.64% | 5.23% | -1.07 |
| MHG-NHY | 4.80% | 7.98% | 0.48 | 3.79% | 7.85% | 0.36 |
| ORK-XXL | 8.84% | 6.59% | 1.19 | 7.65% | 6.44% | 1.04 |
| Portfolio | 0.29% | 3.07% | -0.22 | -1.20% | 3.07% | -0.71 |

Table 7.8: Equal weighted portfolio period 7

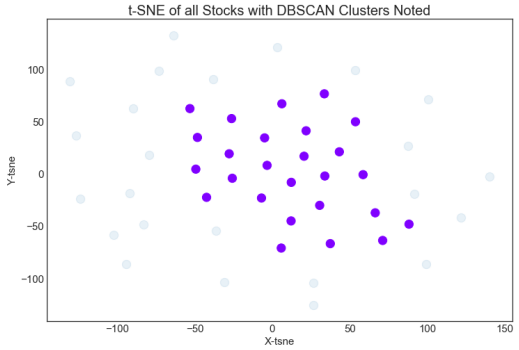
| Period 8 | Without costs | | | With costs | | |
|---|---|---|---|---|---|---|
| Pairs | $R_p$ | $\sigma_p$ | $SR_p$ | $R_p$ | $\sigma_p$ | $SR_p$ |
| AFG-EPR | 54.75% | 23.51% | 2.29 | 52.19% | 23.54% | 2.18 |
| AFG-TEL | -0.59% | 5.67% | -0.28 | -1.97% | 5.71% | -0.51 |
| AKERBP-STL | -5.80% | 14.92% | -0.45 | -6.83% | 14.88% | -0.52 |
| EPR-NPRO | -0.46% | 15.30% | -0.09 | -1.58% | 15.42% | -0.17 |
| KOG-TEL | -2.90% | 4.50% | -0.86 | -4.37% | 4.60% | -1.16 |
| NOD-NOFI | 0.72% | 7.75% | -0.03 | -0.95% | 7.71% | -0.25 |
| NOFI-VEI | 3.09% | 13.05% | 0.16 | 0.55% | 13.03% | -0.03 |
| OLT-TEL | 3.06% | 5.06% | 0.41 | 1.36% | 5.11% | 0.08 |
| SCHB-VEI | 5.59% | 44.97% | 0.10 | 3.84% | 45.00% | 0.06 |
| STL-YAR | 3.38% | 2.55% | 0.94 | 2.23% | 2.52% | 0.50 |
| WWI-WWIB | 3.93% | 3.79% | 0.78 | 2.07% | 3.77% | 0.29 |
| Portfolio | 5.89% | 5.17% | 0.95 | 4.24% | 5.16% | 0.63 |

Table 7.9: Equal weighted portfolio period 8

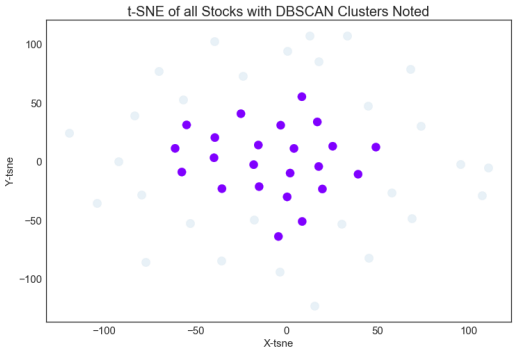
## 7.3 Clusters formed



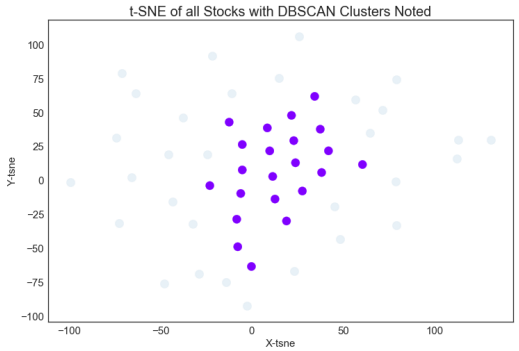

(a) clusters training period 1



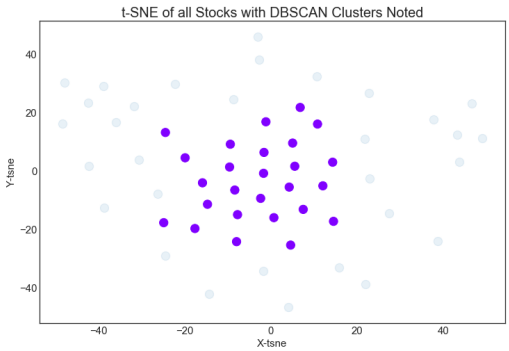

(b) cluster training period 2



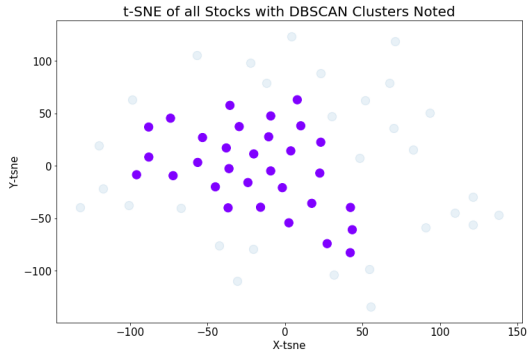

(c) cluster training period 3



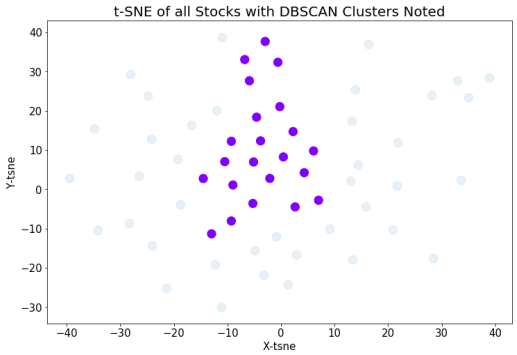

(d) cluster training period 4



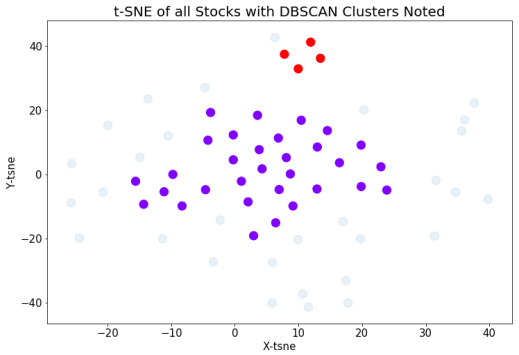

(e) cluster training period 5




(f) cluster training period 6




(g) cluster training period 7




(h) clusters training period 8

Purple indicates cluster 1, red indicates cluster 2.

Figure 7.1: Clusters formed in all training periods

Cluster period 1[1]

| | | | | |
|---|---|---|---|---|
| ATEA | BAKKA | GJF | KOG | MHG |
| NHY | OLT | ORK | PGS | SALM |
| SCHA | SDRL | SNI | STL | SUBC |
| TEL | TGS | VEI | WWI | WWIB |
| WWL | YAR | STB | STL | DNB |

Cluster period 2

| | | | | |
|---|---|---|---|---|
| AFG | AKER | ATEA | BAKKA | EKO |
| EVRY | GJF | KOG | MHG | NHY |
| NPRO | OLT | ORK | SALM | SCHA |
| SDRL | SNI | STL | SUBC | TEL |
| TOM | VEI | WWI | WWIB | WWL |
| YAR | | | | |

Cluster period 3

| | | | | |
|---|---|---|---|---|
| AFG | AKER | ATEA | BAKKA | EKO |
| GJF | GSF | KOG | NHY | NPRO |
| OLT | ORK | SALM | SNI | STB |
| STL | TEL | TOM | VEI | WWI |
| WWIB | WWL | YAR | | |

Cluster period 4

| | | | | |
|---|---|---|---|---|
| AFG | AKER | ATEA | EKO | GJF |
| KOA | GSF | KOG | NHY | NPRO |
| OLT | ORK | SNI | STB | STL |
| TEL | TGS | TOM | VEI | WWI |
| WWIB | WWL | YAR | | |

Cluster period 5

| | | | | |
|---|---|---|---|---|
| AKER | ATEA | BAKKA | BWLPG | EKO |
| ENTRA | GJF | GSF | KOA | KOG |
| MHG | NHY | NPRO | OLT | ORK |
| SALM | SCHA | SNI | TEL | TOM |
| VEI | WWI | WWIB | WWL | XXL |
| YAR | | | | |

Cluster period 6

| | | | | |
|---|---|---|---|---|
| AFG | AKER | ATEA | BAKKA | BWLPG |
| EKO | ENTRA | EPR | GJF | GSF |
| KOA | KOG | MHG | NHY | NPRO |
| OLT | ORK | SALM | SCHA | SCHB |
| SNI | STB | STL | SUBC | TEL |
| TGS | VEI | WWI | WWIB | XXL |
| YAR | | | | |

Cluster period 7

| | | | | |
|---|---|---|---|---|
| AFG | ATEA | BAKKA | EKO | ENTRA |
| EPR | GJF | GSF | KOG | MHG |
| NHY | NPRO | OLT | ORK | SALM |
| SCHA | SCHB | STB | TEL | VEI |
| XXL | YAR | | | |

Cluster period 8

| | | | | |
|---|---|---|---|---|
| AFG | AKER | AKERBP | ATEA | B2H |
| ENTRA | EPR | GJF | HEX | KIT |
| KOA | KOG | NHY | NOD | NOFI |
| NPRO | OLT | ORK | SCHA | SCHB |
| SNI | STB | STL | SUBC | TEL |
| TOM | VEI | WWI | WWIB | XXL |
| YAR | BAKKA | LSG | MHG | SALM |

[1]Stocks in purple are in cluster 1, stocks in red in cluster 2

## 7.4 Python Code

The Python code that follows are only a fraction of the entire code. The code below are for training and testing period 1 only. However, the same procedure are repeated for subsequent periods.

### 7.4.1 Stage 1: Data Management

```
# Importing Libraries and modules
import matplotlib.cm as cm
import matplotlib.pyplot as plt
%matplotlib inline  # We use Jupyter Notebook. Otherwise, use plt.show()

import numpy as np
import pandas as pd
import seaborn as sns

from sklearn.cluster import KMeans, DBSCAN
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn import preprocessing

from statsmodels.tsa.stattools import coint
from statsmodels.tsa.stattools import adfuller
import statsmodels.regression.linear_model as rg
from scipy import stats

# Importing our return and fundamentals dataset
stocks = pd.read_csv('OSEBX.csv', index_col='Date', sep=';')
benchmark = pd.read_csv('Benchmark.csv', index_col='Date')
benchmark = benchmark['Adj Close']
benchmark = benchmark['2014-01-01':'2014-06-30']
fundamentals = pd.read_excel('Fundamentals.xlsx',sheetname='2013')

# Training period:
training1 = stocks['2013-01-01':'2013-12-31'].dropna(axis=1)
# Createing an array of daily returns
returns1 = training1.pct_change()[1:] # Skip the first row (NaN)
```

### 7.4.2 Stage 2: Stock Filtering

In this section we use unsupervised machine learning techniques in order to filter out stocks that are suitable for pairs trading. We have followed the notebook written by (Larkin, 2017), and modified it to our needs.

#### Principal Components Analysis

```
pca = PCA(n_components = 12) # nr. of components is set to 12
pca.fit(returns1) # Extract the common underlying factors for each stock
pca.explained_variance_ratio_.cumsum() # determine  nr. of components
print('The shape of the array after PCA is:',pca.components_.T.shape)

# We add the three fundamental factors to the model
extracted_data = np.hstack(
    (pca.components_.T, fundamentals['ROIC'].values[:,np.newaxis],fundamentals['
    DEBT/EQUITY'].values[:,np.newaxis],fundamentals['SALES_GROWTH'].values[:,np.
    newaxis])
    )
extracted_data = preprocessing.StandardScaler().fit_transform(extracted_data)
print ('The shape of the array is now:', extracted_data.shape)
```

#### Density-Based Spatial Clustering of Applications with Noise

```
clustering = DBSCAN(eps=1.5, min_sample=3)  # eps can be changed. min_sample >=3
clustering.fit(extracted_data)cluster_group = clustering.labels_3
n_clusters = len(set(cluster_group)) - (1 if -1 in cluster_group else 0)
print ("\nClusters discovered: %d" % n_clusters) # prints the number of
    clusteres
clusteres = clustering.labels_ # Clustered = array of values >= -1

# clustered_series is stored as a pandas series object, where the stocks are the
    rows.
clustered_series = pd.Series(index = training1.columns,data = clusters.flatten()
    )
clustered_series_all = pd.Series(index = training1.columns, data = \clusteres.
    flatten()) # The list called clusteres is added.
```

```
clustered_series = clustered_series[clustered_series != -1]
cluster_limit = 50 # Limits the number of stocks in each cluster
counts = clustered_series.value_counts()
ticker_count_reduced = counts[(counts>1) & (counts<=cluster_limit)]
print ("Clusters formed: %s" % len(ticker_count_reduced))
print ("Pairs to evaluate: %s" % \(ticker_count_reduced*(ticker_count_reduced-1)
    ).sum())
```

### t-Distributed Stochastic Neighbour Embedding

```
# t-SNE is a tool to visualize high-dimensional data.
clusters_tsne = TSNE(learning_rate=200, perplexity=19).fit_transform(
    extracted_data)

# Determining the layout
plt.figure(1, facecolor='white',figsize=(8,6),)
plt.clf() # Clear the current figure
plt.scatter(
     clusters_tsne[(cluster_group!=-1), 0],
     clusters_tsne[(cluster_group!=-1),1],
     s=150,
     alpha=1,
     c=(cluster_group[cluster_group!=-1]),
     cmap=cm.rainbow
)
plt.scatter(
clusters_tsne[(clustered_series_all==-1).values, 0],
clusters_tsne[(clustered_series_all==-1).values, 1],
s=150,
alpha=0.1
)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.title('t-SNE of all Stocks with DBSCAN Clusters Noted',fontsize=20);
plt.xlabel('X-tsne',fontsize=15)
plt.ylabel('Y-tsne',fontsize=15)
```

### 7.4.3 Stage 3: Identifying Mean-Reversion

The function "*cointegrated_stocks*" is written by Mackenzie and Margenot (2018) and modified to our needs.

```
# The function below is copied from https://www.quantopian.com/lectures/
    introduction-to-pairs-trading. It is modified to our need

# This function takes one object (data). The object that will be added later, is
     the price data of all clusterd stocks.
def cointegrated_stocks(data):
    n = data.shape[1] # give us the number of stocks in cluster
    score_matrix = np.zeros((n, n)) # create an n*n array of zeros
    pvalue_matrix = np.ones((n, n)) # this array will be updated with
    cointegration p-values
    keys = data.keys() # store the ticker symbol of stocks
    pairs = [] # create an empty list

# The for loop below is a nested for-loop. When passing in a data object, the
    first for-loop will iterate through all stock which belongs to a cluster. The
     nested for-loop does the same, except it jumps one column to the right in
    the data frame.
    for i in range(n):
        for j in range(i+1, n):
            S1 = data[keys[i]]
            S2 = data[keys[j]]

# Description of the coint function: Uses the augmented Engle-Granger two- step
    cointegration test.
            result = coint(S1, S2, trend='nc') # no intercept included.
            score = result[0] # store result index[0]
            pvalue = result[1] # store p-value
            score_matrix[i, j] = values from score and p-value are stored in a
    matrix.
            pvalue_matrix[i, j] = pvalue
            if pvalue < 0.05: # If p-value below 5%, add pair to "pairs" list
                pairs.append((keys[i], keys[j]))
    return score_matrix, pvalue_matrix, pairs
```

```
# Create an empty dictionary consisting of the tree following keys: pairs, p-
    value_matrix and score_matrics, where values are a list of stock pairs in
    tuples, an array of p-values and score respectively.
cluster_dictionary = {}
# The for-loop iterate trough all stocks in clusters and extract the stock pairs
     that has a p-value below 5%
for i, which_clust in enumerate(ticker_count_reduced.index):
    stock_ticks = clustered_series[clustered_series == which_clust].index # An
    index list of all stocks in  cluster
    scores, pvalues, pairs = cointegrated_stocks(
        training1[stock_ticks]
    )
    cluster_dict[which_clust] = {}
    cluster_dict[which_clust]['score_matrix'] = scores
    cluster_dict[which_clust]['pvalue_matrix'] = pvalues
    cluster_dict[which_clust]['pairs'] = pairs


pairs_discovered = [] # Create an empty list
# Run a for loop that extend the list to include all pairs.
for clust in cluster_dict.keys():
    pairs_discovered.extend(cluster_dict[clust]['pairs'])
print('The following pairs will be traded in this period:')
pairs_discovered

# Output from the print statement above: The following pairs will be traded in
    this period:
[('BAKKA','GJF'),
('GJF','MHG'),
('KOG','MHG'),
('KOG','SALM'),
('KOG','TEL'),
('MHG','TEL'),
('MHG','WWIB')]

 pairs_df = pd.DataFrame(training1[['BAKKA','GJF','KOG','MHG','TEL','SALM','WWIB
    ']])
sns.pairplot(pairs_df,size=1.5
# Output from sns.pairplot:
```

.

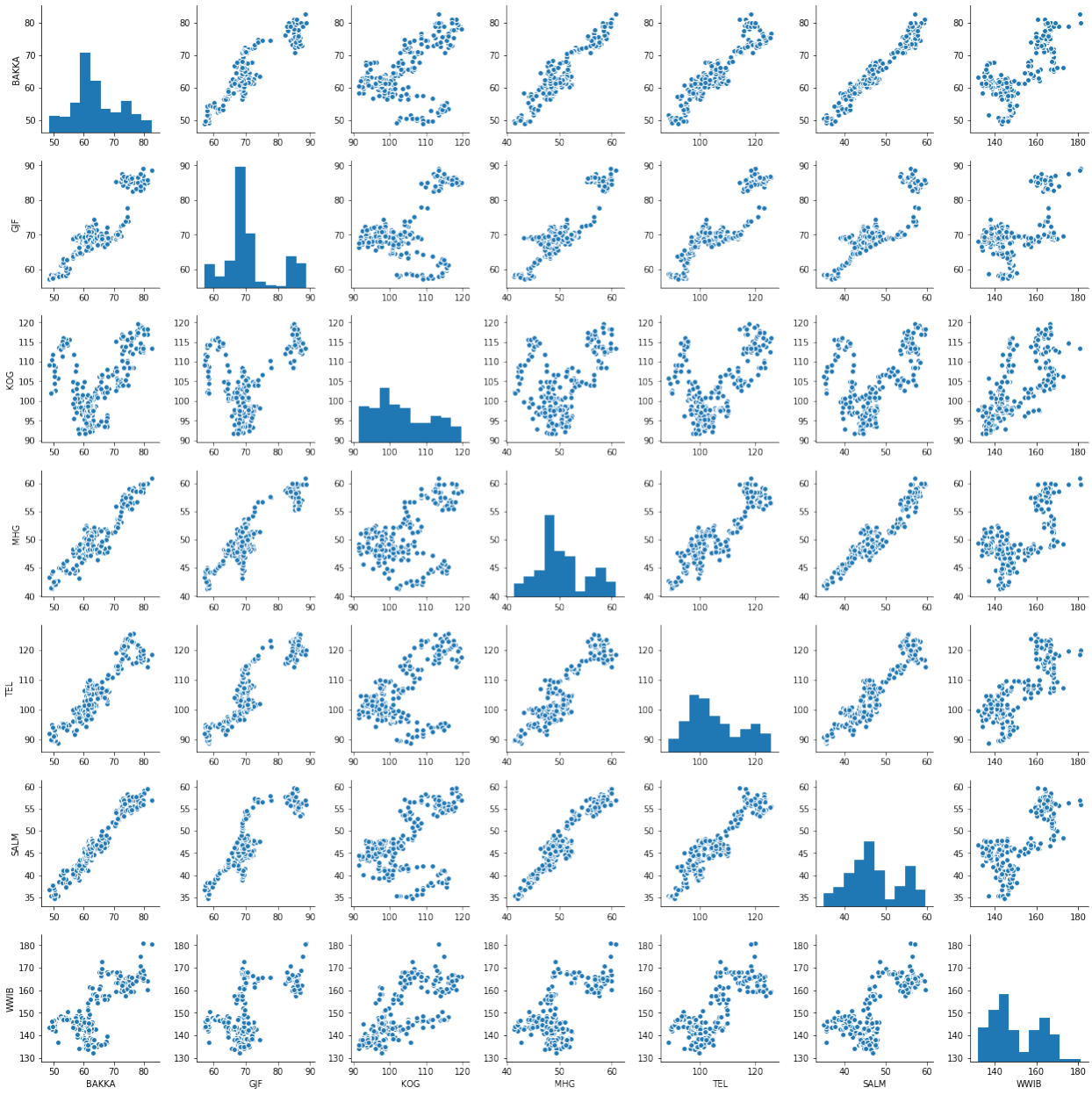



Figure 7.2: Linear relationship between filtered stocks

```
# Visualizing a stock pair and cointegration p-value
A1 = training1['MHG']
A2 = training1['WWIB']
plt.figure(figsize=(10,6))
(A1/A1.iloc[0]).plot()
(A2/A2.iloc[0]).plot(c='g')
plt.ylabel('Cumulative return',fontsize=15)
plt.title('Stock Pair',fontsize=15)
plt.legend()
score, pvalue, _ = coint(A1, A2, trend='nc')
print ('Cointegration: p-value:', "%.2f" %pvalue)

# Output from code section above:
Cointegration: p-value: 0.04
```

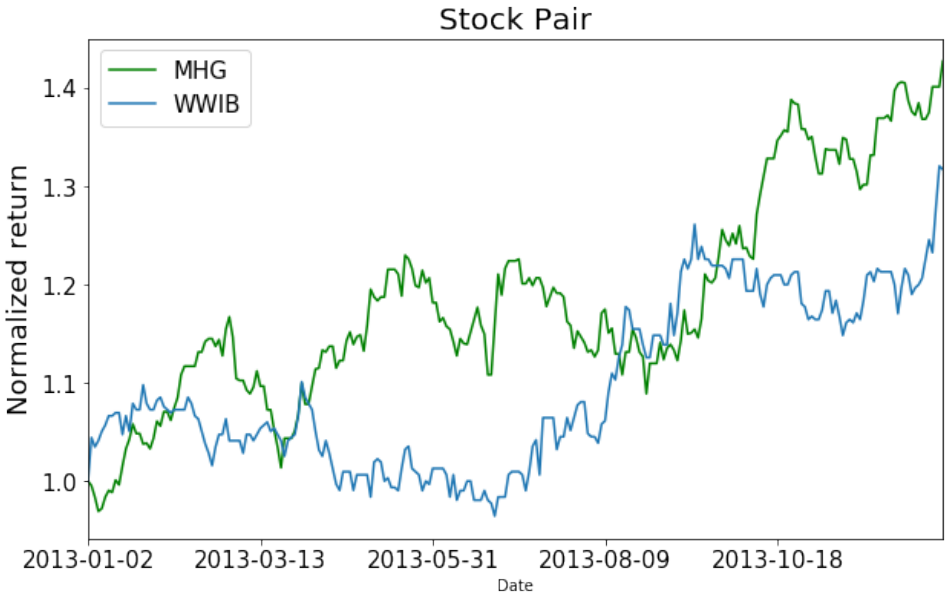



Figure 7.3: MHG-WWIB stock pair

### Check if individual price series are non-stationary

```
S1_adf = adfuller(A1)# Augmented Dickey-Fuller unit root test
S2_adf = adfuller(A2)# Augmented Dickey-Fuller unit root test

# Check if price series of stock S1 (MHG) is non-stationary
# The same code is done on S2 (WWIB).
```

```
print("S1 Prices Augmented Dickey-Fuller Test:".upper())
print("")
labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations
    Used']

for value,label in zip(S1_adf,labels):
    print(label+':'+str(value) )
if S1_adf[1] <= 0.05:
    print("Strong evidence against the null hypothesis")
    print("reject the null hypothesis. Price series has no unit root and is
    stationary")
else:
    print("Weak evidence against null hypothesis")
    print("time series has a unit root")
    print("indicating it is non-stationary")

# Checking if series are I(1)
S1_diff = A1 - A1.shift(1)
S1_diff[np.isnan(S1_diff)] = 0
S2_diff = A2 - A2.shift(1)
S2_diff[np.isnan(S2_diff)] = 0
S1_diff_adf = adfuller(S1_diff)
S2_diff_adf = adfuller(S2_diff)

# Ordinary Least Square - OLS
OLS = rg.OLS(A1, A2).fit()
beta = OLS.params[0]
spread = A1 - beta * A2
OLS.summary()
```

```
# Output from OLS regression on the spread:
```

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | MHG | R-squared: | 0.994 |
| Model: | OLS | Adj. R-squared: | 0.994 |
| Method: | Least Squares | F-statistic: | 4.365e+04 |
| Date: | Thu, 03 May 2018 | Prob (F-statistic): | 9.11e-281 |
| Time: | 14:01:24 | Log-Likelihood: | -686.40 |
| No. Observations: | 249 | AIC: | 1375. |
| Df Residuals: | 248 | BIC: | 1378. |
| Df Model: | 1 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| WWIB | 0.3354 | 0.002 | 208.915 | 0.000 | 0.332 | 0.339 |

| | | | |
|---|---|---|---|
| Omnibus: | 75.475 | Durbin-Watson: | 0.062 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 15.707 |
| Skew: | -0.276 | Prob(JB): | 0.000388 |
| Kurtosis: | 1.900 | Cond. No. | 1.00 |

Figure 7.4: Engle-Granger first step

```
spread_adf = adfuller(spread,regression='nc')
print("Augmented Dickey-Fuller Co-Integration Test on spread")
print("")
labels = ['ADF Statistic','p-value','#Lags Used','Number of Observations Used']
for value,label in zip(spread_adf,labels):
    print(label+':'+str(value) )
if spread_adf[1] <= 0.05:
    print("Strong evidence against the null hypothesis")
    print("Spread has no unit root and is stationary")
else:
    print("Weak evidence against null hypothesis")
    print("Spread has a unit root, indicating it is non-stationary
```

Augmented Dickey-Fuller Test on spread

| | |
|---|---|
| ADF Statistic: | -2.06360186303 |
| p-value: | 0.037411835533 |
| Lags Used : | 0 |
| Number of Observations Used: | 248 |

Strong evidence against the null hypothesis, spread has no unit root and is stationary

### 7.4.4 Stage 4: Trading Setup and Execution

In this section the trading algorithm is created. The steps in the trading setup are done for all stock pairs.

```
# Slicing out first testing period and a pair that are cointegrated.
pair1 = stocks.loc['2014-01-01':'2014-06-30',['MHG','WWB']].dropna()
pair1.columns = ['S1','S2']
# Creating return columns for each stock in pair1
pair1['S1ret'] = pair1['S1'].pct_change(1)
pair1.iloc[0,2] = 0
pair1['S2ret'] = pair1['S2'].pct_change(1)
pair1.iloc[0,3] = 0
```

#### Creating the rolling z-score

```
# OLS - Regression
lm_pair1 = rg.OLS(pair1['S1'], pair1['S2']).fit()
pair1_b1 = lm_pair1.params[0]
# Creating a new column calles'pair_spread'
pair1['pair_spread'] = pair1['S1'] - pair1_b1 * pair1['S2']

# Calculation the rolling 10-day covariance
rolling_pair_cov = pair1.loc[:, ['S1','S2']].\
rolling(window=10).cov(pair1.loc[:, ['S1','S2']], pairwise=True)
```

```
# Slice multi index dataframe to single index dataframe of paired stocks
    covariance
idx = pd.IndexSlice
rolling_pair_cov = rolling_pair_cov.loc[idx[:,'S1'],'S2']

# Converting Date and stock index into date index, by making stock at index
    level 1 into a new column
rolling_pair_cov = rolling_pair_cov.reset_index(level=1)

# Calculating the 10-day rolling variance
rolling_pair_var = pair1['S2'].rolling(window=10).var()
# ROLLING BETA
pair1['rolling_pair_b1'] = rolling_pair_cov['S2'] / rolling_pair_var

# CALCULATION OF THE 10-DAY ROLLING SPREAD:
pair1['rolling_pair_spread'] = pair1['S1'] - pair1['rolling_pair_b1'] * pair1['
    S2']
```

### 10-day rolling z-score

```
# CALCULATION OF THE 10-DAY ROLLING Z-SCORE:
pair1['rolling_Z_score'] = (pair1['rolling_pair_spread'] - \
                            pair1['rolling_pair_spread'].rolling(window=10).mean
    ())/ \
                            pair1['rolling_pair_spread'].rolling(window=10).std
    ()
```

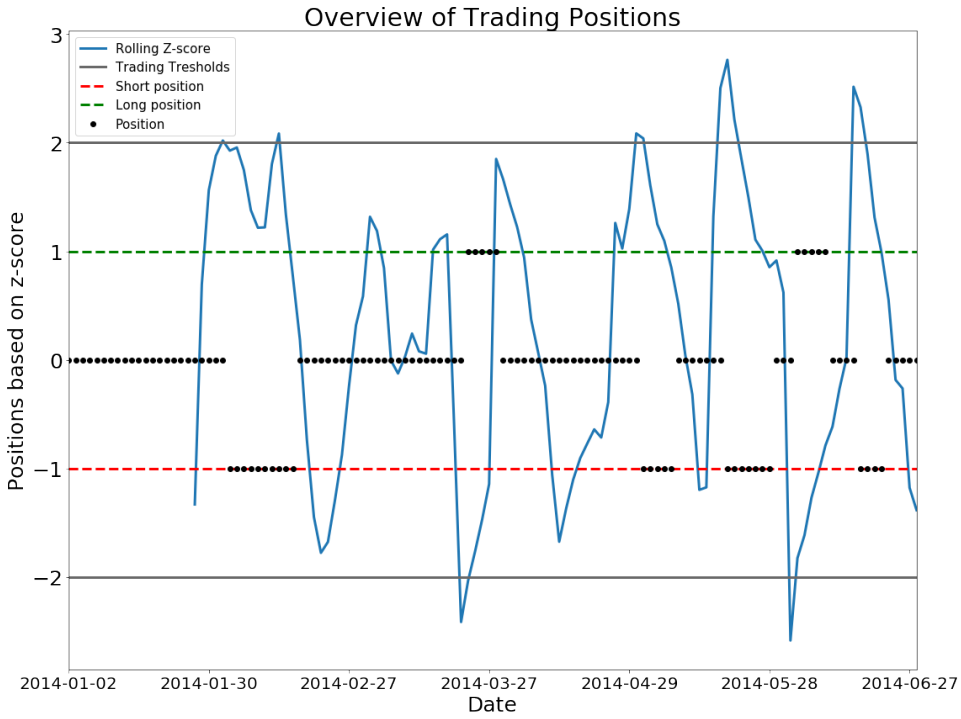
### Trading Signals Algorithm

```
# TRADING SIGNALS ALGORITHM
# Z-score the day before
pair1['rolling_Z_score(-1)'] = pair1['rolling_Z_score'].shift(1)
# Z-score two days before
pair1['rolling_Z_score(-2)'] = pair1['rolling_Z_score'].shift(2)
pair1['pair_signal'] = 0
pair_signal = 0

# The for-loop below is explained in section 3.5.1 - Trading signals and execution
for i, r in enumerate(pair1.iterrows()):
    if r[1]['rolling_Z_score(-2)'] > -2 and r[1]['rolling_Z_score(-1)'] < -2:
        pair_signal = -2
    elif r[1]['rolling_Z_score(-2)'] < -1 and r[1]['rolling_Z_score(-1)'] > -1:
        pair_signal = -1
    elif r[1]['rolling_Z_score(-2)'] < 2 and r[1]['rolling_Z_score(-1)'] > 2:
        pair_signal = 2
    elif r[1]['rolling_Z_score(-2)'] > 1 and r[1]['rolling_Z_score(-1)'] < 1:
        pair_signal = 1
    else:
        pair_signal = 0
    pair1.iloc[i, 10] = pair_signal

# TRADING POSITIONS: 1 = LONG SPREAD TRADE, -1 = SHORT SPREAD TRADE
pair1['position'] = 0
position = 0
for i, r in enumerate(pair1.iterrows()):
    if r[1]['pair_signal'] == -2:
        position = 1
    elif r[1]['pair_signal'] == -1:
        position = 0
    elif r[1]['pair_signal'] == 2:
        position = -1
    elif r[1]['pair_signal'] == 1:
        position = 0
    else:
        position = pair1.loc[:,'position'][i-1]
    pair1.iloc[i, 11] = position
```

```
# Visualising trading positions.
```




Black dots show the holding position (long, short or no position)

Figure 7.5: Trading Positions

```
# STRATEGY WITH AND WITHOUT TRANSACTION COSTS:
pair1['spread_returns'] = pair1['S1ret'] - \
pair1['rolling_pair_b1'] * pair1['S2ret']
# Strategy Without Trading Costs
pair1['return'] = pair1['spread_returns'] * pair1['position']

# Strategy With Trading Costs (0.139% Per Trade)

pair1['position(-1)'] = pair1['position'].shift(1)
pair1['cost'] = 0
cost = 0
# Adding transaction cost whenever the z-score crosses our trading signals
for i, r in enumerate(pair1.iterrows()):
    if (r[1]['pair_signal'] == -2 or r[1]['pair_signal'] == -1 or \
        r[1]['pair_signal'] == 2 or r[1]['pair_signal'] == 1) \
```

```
            and r[1]['position'] != r[1]['position(-1)']:
        cost = 0.00139
    else:
        cost = 0.000
    pair1.iloc[i, 15] = cost
pair1['ret_w_cost'] = pair1['return'] - pair1['cost']

# Visualizing the trend of the stock pair:

pair1[np.isnan(pair1)] = 0
pair1['Cumulative return'] = np.cumprod(pair1['return']+1) - 1
pair1['Cumulative return with costs'] = np.cumprod(pair1['ret_w_cost']+1) - 1
pair1['MHG return'] = np.cumprod(pair1['S1ret']+1)  - 1
pair1['WWIB return'] = np.cumprod(pair1['S2ret']+1) - 1
pair1.plot(y=['Cumulative return','Cumulative return with costs','MHG return','
   WWIB return'],figsize=(10,6))
plt.title('Pairs Strategy Cumulative Returns')
plt.legend(loc='upper left')

# Output:
```

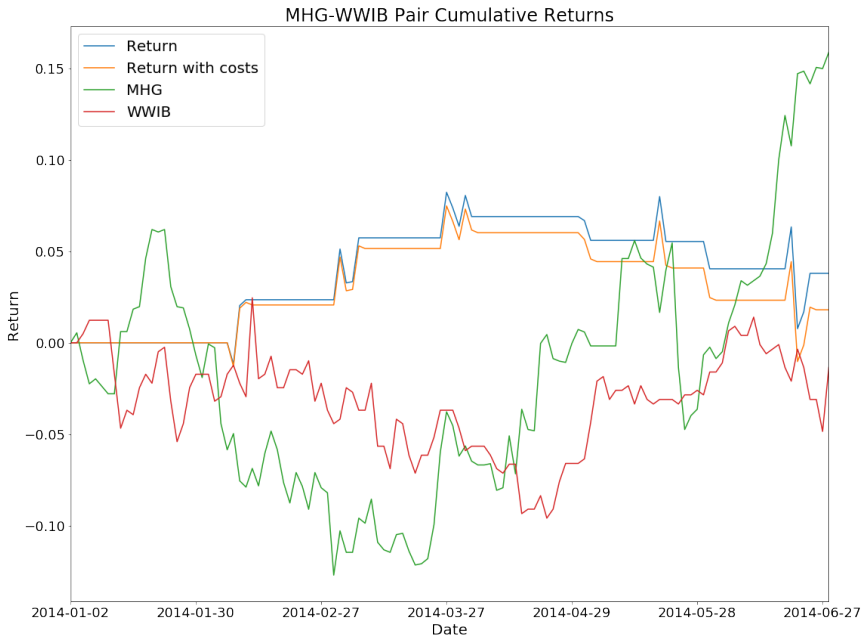



Figure 7.6: MHG-WWIB Pair Return

```
# Saving the dataframe for MHG-WWIB pair: This will be done for all stock pairs.
pair1.to_csv('MHG-WWIB_10')
```

### Creating a Portfolio of pairs

In this section we show how we have created an equally weighted portfolio of all pairs formed in period 1. In addition, performance measures are calculated. Again, the same steps are done for subsequent periods.

```
Pair1 = pd.read_csv('BAKKA-GJF_10',index_col='Date')
Pair2 = pd.read_csv('GJF-MHG_10',index_col='Date')
Pair3 = pd.read_csv('KOG-MHG_10',index_col='Date')
Pair4 = pd.read_csv('KOG-SALM_10',index_col='Date')
Pair5 = pd.read_csv('KOG-TEL_10',index_col='Date')
Pair6 = pd.read_csv('MHG-TEL_10',index_col='Date')
Pair7 = pd.read_csv('MHG-WWIB_10',index_col='Date')
Pair1_ret = Pair1[['return']]
Pair2_ret = Pair2[['return']]
Pair3_ret = Pair3[['return']]
Pair4_ret = Pair4[['return']]
Pair5_ret = Pair5[['return']]
Pair6_ret = Pair6[['return']]
Pair7_ret = Pair7[['return']]
Pair1_ret_c = Pair1[['ret_w_com']]
Pair2_ret_c = Pair2[['ret_w_com']]
Pair3_ret_c = Pair3[['ret_w_com']]
Pair4_ret_c = Pair4[['ret_w_com']]
Pair5_ret_c = Pair5[['ret_w_com']]
Pair6_ret_c = Pair6[['ret_w_com']]
Pair7_ret_c = Pair7[['ret_w_com']]

# Importing data of OSEBX for comparison
benchmark = pd.read_csv('Benchmark.csv', index_col='Date')
benchmark = benchmark['Adj Close']
benchmark = benchmark['2014-01-01':'2014-06-30']
```

```
benchmark_cum = ((benchmark/benchmark[0]))

# Creating/Concatenating a return Pandas DataFrame for all pairs (with and
    without transaction costs)
Portfolio_ret = pd.concat([Pair1_ret,Pair2_ret,Pair3_ret,Pair4_ret,Pair5_ret,\
                           Pair6_ret,Pair7_ret],axis=1)

Port_ret_com = pd.concat([Pair1_ret_c,Pair2_ret_c,Pair3_ret_c,Pair4_ret_c,
    Pair5_ret_c,\
                           Pair6_ret_c,Pair7_ret_c],axis=1)

# Changing the column names for clarity and analysis purposes:
Portfolio_ret.columns=['BAKKA-GJF','GJF-MHG','KOG-MHG','KOG-SALM','KOG-TEL','MHG
    -TEL','MHG-WWIB']
Port_ret_com.columns = ['BAKKA-GJF','GJF-MHG','KOG-MHG','KOG-SALM','KOG-TEL','
    MHG-TEL','MHG-WWIB']

# Visualising the cumulative returns for all pairs
Port_cumret = np.cumprod(Portfolio_ret +1)-1
Port_cumret_c = np.cumprod(Port_ret_com +1)-1
Port_cumret.plot(figsize=(12,8),title='Pairs Cumulative returns 2013')
Port_cumret_c.plot(figsize=(12,8),title='Pairs Cumulative returns 2013 with
    commisions')



# Period returns, volatility and sharpe ratio for each pair
num_pairs = len(Portfolio_ret.columns)
print ('Numbers of pairs in test period 1 are:',num_pairs)
print('\n')

pairs_returns = Port_cumret.iloc[-1]
print('Pairs Return Without Commisions: ')
print(pairs_returns)
print('\n')

pairs_ret_com = Port_cumret_c.iloc[-1]
print('Pairs Return With Commisions: ')
print(pairs_ret_com)
print('\n')
```

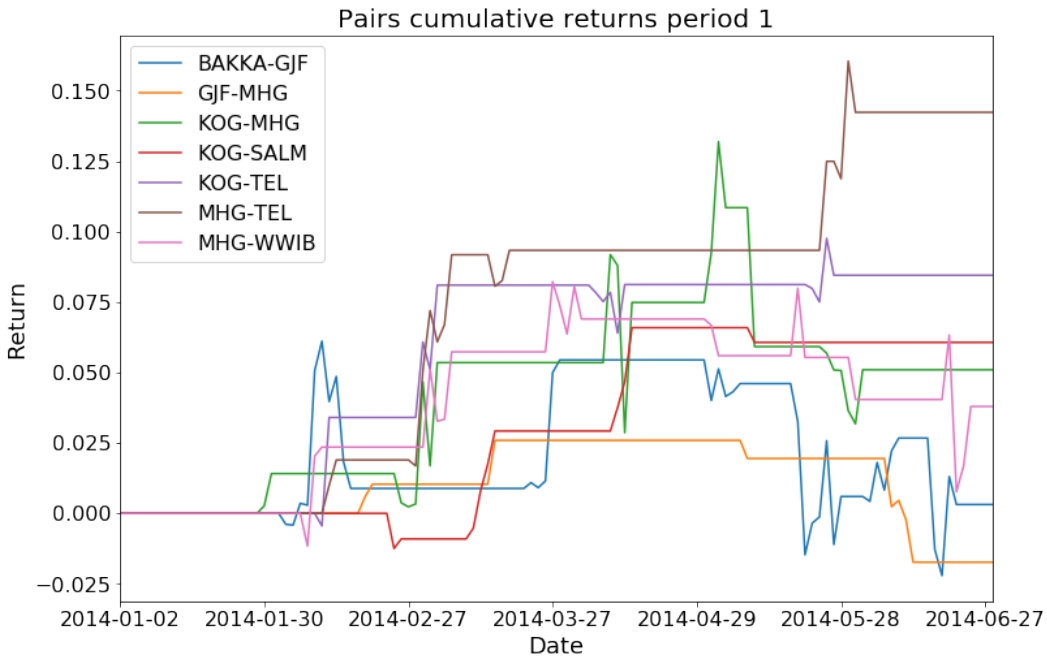



Figure 7.7: Pairs cumulative returns from period 1

```
Pairs_std = Portfolio_ret.std()*122**0.5
print('Pairs Standard Deviations: ')
print(Pairs_std)
print('\n')

Pairs_std_com = Port_ret_com.std()*122**0.5
print('Pairs Standard Deviations with Commisions: ')
print(Pairs_std_com)
print('\n')

Pairs_Sharpe = (pairs_returns -0.0097)/Pairs_std
print('Pairs Sharpe-Ratio Without Commisions')
print(Pairs_Sharpe)
print('\n')

Pairs_Sharpe = (pairs_ret_com -0.0097)/Pairs_std_com
print('Pairs Sharpe-Ratio With Commisions')
print(Pairs_Sharpe)

print('We will have equal weighting in each pair')
Weights_in_each_pair = 1/num_pairs # Allocation to each pair:
```

```
print('Weight in each pair are:', "%.4f" % Weights_in_each_pair)
# Creating a list of equal weights
weights =[Weights_in_each_pair]*num_pairs
weights = np.array(weights)

# PORTFOLIO RETURNS:
print('Portfolio Return Without Commisions:')
port_ret = np.sum(weights * pairs_returns)
print("%.4f" % port_ret)
print('\n')

print('Portfolio Return With Commisions:')
port_ret_com = np.sum(weights * pairs_ret_com)
print("%.4f" % port_ret_com)
print('\n')

# PORTFOLIO VARIANCE:
print('Portfolio Variance Without Commisions:')
port_var = np.dot(weights.T, np.dot(Portfolio_ret.cov() * 122, weights))
print(port_var)
print('\n')

print('Portfolio Variance With Commisions:')
port_var_com = np.dot(weights.T, np.dot(Port_ret_com.cov() * 122, weights))
print(port_var)
print('\n')

# PORTFOLIO VOLATILITY:
print('Portfolio Volatility Without Commisions')
port_vol = np.sqrt(np.dot(weights.T,np.dot(Portfolio_ret.cov() * 122, weights)))
print(port_vol)
print('\n')

print('Portfolio Volatility With Commisions')
port_vol_com = np.sqrt(np.dot(weights.T,np.dot(Port_ret_com.cov() * 122, weights
    )))
print(port_vol_com)
print('\n')
```

```
# SHARPE RATIO:
print('Sharpe-Ratio Without Commisions:')
SR = (port_ret -0.0097)/port_vol
print(SR)
print('\n')

print('Sharpe Ratio With Commisions:')
SR = (port_ret_com -0.0097)/port_vol_com
print(SR)

# For-loop for portfolio without commisions:
for pairs_df in (Pair1_ret,Pair2_ret,Pair3_ret,Pair4_ret,Pair5_ret,Pair6_ret,
    Pair7_ret):
    pairs_df['Normed Return'] = np.cumprod(pairs_df['return']+1)

# For-loop for portfolio with commisions:
for pairs_df2 in (Pair1_ret_c,Pair2_ret_c,Pair3_ret_c,Pair4_ret_c,Pair5_ret_c,
    Pair6_ret_c,Pair7_ret_c):
    pairs_df2['Normed Return'] = np.cumprod(pairs_df2['ret_w_com']+1)

# For-loop for portfolio without commisions:
for pairs_df , allo in zip((Pair1_ret,Pair2_ret,Pair3_ret,Pair4_ret,Pair5_ret,
    Pair6_ret,Pair7_ret),weights):
    pairs_df['Allocation'] = pairs_df['Normed Return']*allo

# For-loop for portfolio with commisions:
for pairs_df2 , allo2 in zip((Pair1_ret_c,Pair2_ret_c,Pair3_ret_c,Pair4_ret_c,
    Pair5_ret_c,Pair6_ret_c,Pair7_ret_c),weights):
    pairs_df2['Allocation'] = pairs_df2['Normed Return']*allo2
all_pairs_in_port = [Pair1_ret['Allocation'],Pair2_ret['Allocation'],Pair3_ret['
    Allocation'],Pair4_ret['Allocation'],\
                     Pair5_ret['Allocation'],Pair6_ret['Allocation'],Pair7_ret['
    Allocation']]
portfolio_cum = pd.concat(all_pairs_in_port,axis=1)

all_pairs_in_port2 = [Pair1_ret_c['Allocation'],Pair2_ret_c['Allocation'],
    Pair3_ret_c['Allocation'],Pair4_ret_c['Allocation'],\
                     Pair5_ret_c['Allocation'],Pair6_ret_c['Allocation'],
    Pair7_ret_c['Allocation']]
```

```
portfolio_cum_com = pd.concat(all_pairs_in_port2 ,axis=1)
portfolio_cum['Total Portfolio'] = portfolio_cum.sum(axis = 1)
portfolio_cum_com['Total Portfolio'] = portfolio_cum_com.sum(axis = 1)

# Visualising the portfolio performance:
portfolio_cum_com['Total Portfolio'].plot(figsize=(12,8),label='Return with
    costs')
portfolio_cum['Total Portfolio'].plot(figsize=(12,8),label = 'Return')
benchmark_cum.plot(label = 'OSEBX')
plt.title('Testperiod 1: Benchmark vs. Strategy')
plt.legend()
# Output:
```

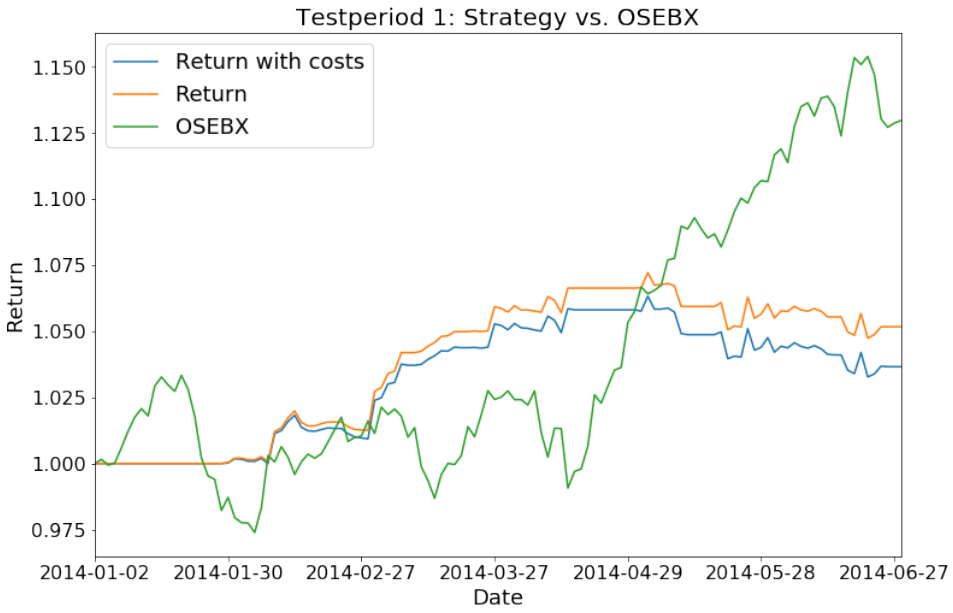



Figure 7.8: Portfolio - period 1

```
# Calculating the strategy beta
benchmark_daily_ret = benchmark_cum.pct_change(1) # cumulative return to daily
    return
df3 = pd.concat([portfolio_cum['Daily Return'],benchmark_daily_ret],axis=1)
df3.dropna(inplace=True)
cov = df3.cov() * 122
cov_with_market = cov.iloc[0,1]
market_var = benchmark_daily_ret.var() * 122
Strategy_beta = cov_with_market / market_var
```

```
print('Beta first period:')
print(Strategy_beta)
```

```
# Calculating return correlation coefficient
np.corrcoef(df3['strategy return'],df3['OSEBX return'])
```

```
array([[ 1.        , -0.06885147],
       [-0.06885147,  1.        ]])
```

## 7.5 First principal component vs. OSEBX

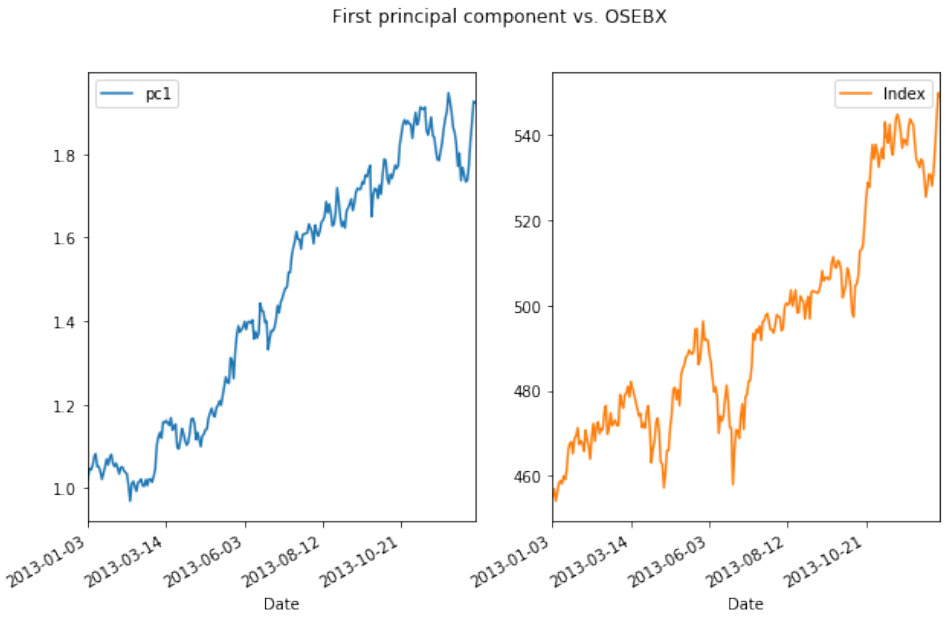



Figure 7.9: First principal components vs. OSEBX