FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTEMENT OF MATHEMATICS AND STATISTICS

Empirical analysis of time-lagged cross-correlations in the Norwegian Stock Market:

a discussion of the Efficient Market Hypothesis —

Marte Jespersen

Master thesis in Statistics STA-3900 December 2016

Abstract

In this thesis we challenge the existence of weak efficiency in the Norwegian Stock Marked, by analysing time-lagged cross correlations between log-return series from 811 stocks listed on the Oslo stock exchange and by creating prediction strategies based on the discovered patterns. We limit the strategies to predicting the direction of the movements of the time series only, i.e. either generating a “buy” or a “sell” signal. We use separate time periods, mainly focusing on two-year periods within the timeframe 2006-2015, and do approximately 80 000 Monte Carlo simulation-tests in each of them. The strategy is tested on real, unexamined data. Our results strongly indicate that information based trading strategies give higher returns and entail lower risk, than random, but similarly constructed strategies.

Most of our analysis is conducted on data from the period January 1st 2006 to July 1st 2015, and we use the period July 1st 2015 to July 1st 2016 as our testing period.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Martin Rypdal for his guidance and good advice throughout the process of working with this thesis. Also, a big thanks to Espen Sirnes for providing the data.

I also want to express my appreciation to my family for all their support and care. You are the best.

Table of Contents

List of Figures

List of Tables

1 Introduction

According to the Efficient Market Hypothesis (EMH) no one should be able to earn abnormal returns on the stock market by analysing historical price data. If the EMH is valid in the Norwegian Stock Market and the marked is indeed efficient, we should not be able to find a strategy, using only historical data on close-prices, that will consistently beat the market.

To beat the marked means having a rate of return that is consistently higher than the average return of the marked while keeping the same level of risk. We measure the average rate of return using Monte Carlo[1] simulations, testing if we do better using information from historical data than we would have using no information. That is, we compare information-based strategies with random, but similarly constructed, strategies used on the same shares, and in the same time periods.

1.1 The efficient market hypothesis

The efficient market hypothesis (EMH) is based on the assumption that, at any point in time, all available information relevant to the price of a share is reflected in the price set by the rational market. Once new information indicates that the price should shift, investors respond and the prices quickly adjust. Due to this immediate reflection of all available information in the stock-price, there are no over- or under-valued stocks in the marked, and every stock is priced to its intrinsic value. This means that no investor can beat the market by generating abnormal returns (Kendall, 1953). The price is said to be random due to the fact that new information is random and unpredictable, and attempts to outperform the market is considered a game of chance rather than one of skill.

The EMH was developed by Eugene Fama, who defined the efficient market to be a market with a large amount of profit maximizing rational players that were actively competing with each other in trying to predict future stock values. Furthermore, all information that is relevant in determining the value of the security should be almost freely available to all (Fama E. , 1965 a). A market can be considered efficient with respect to some set of information, if this information is of no use when it comes to earning profits on account of it (Jensen, 1978). Fama (1970) identified three distinct levels of market

[1] Used to simulate statistics based on some assumption of the properties of the underlying distribution.

efficiency based on what type of information that was to be included; the weak, the semi strong, and the strong form of efficiency. A weak form of market efficiency entails that all historical information, including past returns, is reflected in the current price of the stock (Bodie, Kane, & Marcus, 2007). This assumption is consistent with the random walk hypothesis that states that stock prices have random movements, and that price changes are independent of each other (Kendall, 1953; Roberts, 1959; Fama E. , 1965 b). If the weak form of the efficient market hypothesis holds, no one can beat the marked by consistently earning abnormal returns using technical analysis of historical price movements. One can, however, beat the market using fundamental analysis or private information.

If a market is semi-strong efficient, all publicly available information, including historical data, will be reflected in the price of a security. No type of fundamental analysis including the systematic study of companies, sectors or the economy at large, will consistently beat the market (Bodie, Kane, & Marcus, 2007). If this form of EMH holds, one can only consistently earn abnormal returns using private information.

The strong form of market efficiency entails that all relevant information, whether public or not, is reflected in the stock prices. This means that excess returns cannot consistently be achieved using any information, not even doing insider trading (Brealey, Myers, & Marcus, 1999).

In practice, it is considered almost impossible to have a perfectly efficient market. This is due to the fact that people are not always rational and anomalies do occur. The “true” values of the stocks are not always reflected correctly in the prices, and empirical studies show that predictable price patterns can emerge and create opportunity for abnormal profit making.

Different markets around the world exhibit various degrees of efficiency. The types of anomalies discovered vary in nature but they all seem to have one thing in common, namely that they are short lived once discovered. It is believed that once anomalies that are predictable in nature are reported, they become part of the information that is reflected in the stock prices, i.e. competed away. One example of this is the disappearance of the weekend effect in the UK stock market (Steeley, 2001).

The existence of anomalies in stock markets seems to be well accepted, but the question of whether one can take advantage of them to gain superior returns is under debate. Even if anomalies persist over time, there are no guarantees they will keep doing so in the future, and if they do, there might be hidden cost associated with the strategy used when trading on them.

It is common to differentiate between three main types of anomalies: Technical, calendar, and fundamental. The two former relates to the weak form of market efficiency, while the latter relates to the semi-strong form. Technical anomalies are patterns found when analyzing historical information such as prices, returns or volume that one can use to predict future price changes. These types of anomalies include *the Momentum Effect,* which involves investors outperforming by buying past winners and selling past losers. Several studies have analyzed the Norwegian market with respect to momentum effects. Korneliussen and Rasmussen (2014) found evidence of short-term momentum using data from OSE in the period 1991 to 2010 and Nygaard (2011) found momentum in small cap stocks in the period 2002 to 2007. Another example of a technical anomaly is the *Moving averages* anomaly, which involve earning higher returns buying stocks when short period averages raises over long period averages, and vice versa (Brock, Lakonishok, & LeBaron, 1992).

Fundamental anomalies create predictability in stock price changes that can be found through the examination of the underlying forces that affects the economy, using analysis of publicly available information. Examples of fundamental anomalies include *Growth versus Value investing*, where investors can take advantage of the fact that growth companies[2] often appear to be overestimated and value companies[3] underestimated. What type of stocks that do better, value stocks or growth stocks, has alternated throughout history, but during the 21st century value stocks have generally done better (TD, 2009). Value investing includes strategies like buying stocks with low price-to-book ratios or low price-to-earnings ratios. There is however, no indication of higher returns related to firm size in the Norwegian stock market (Korneliussen & Rasmussen, 2014).

Calendar anomalies involve seasonality in stock returns, e.g. daily, weekly, monthly etc. that appear to be systematic and creates opportunity to predict future price movements. A 2015 study on the calendar effects in the Norwegian Stock market suggests that there were no calendar anomalies in returns on the OBX and OSEAX indices, but the returns on

[2] A company with high returns on equity (ROE) whose revenues and earnings are expected to grow faster than the average companies in the same sector.

[3] A company with stocks trading at a lower value than should be expected.

the small cap index were observed to be significantly higher on Fridays (weekend effect) and on the last trading day before Christmas (Holiday effect) (Yavrumyan, 2015).

Traditionally, the question of whether one has beaten the marked or not is determined through the use of the CAPM[4] (or similar models with more explanatory factors[5]). One problem with this method is the Joint Hypothesis Problem, which states that it is not possible to sufficiently test market efficiency using a model based approach. If a test of this kind is rejected, it could be because the market is truly inefficient, or it can be because an incorrect model has been assumed. This entails that market efficiency as such cannot be rejected (Campbell, Lo, & MacKinlay, 1997).

Another factor to address is the risk involved with trading on a specific stock, e.g. if there is any causal relationship between volatility and expected return. A common belief is that there is a trade-off between risk and return i.e. that one can earn greater returns if one except a higher risk. However, the empirical findings are contradicting; some research concludes that the relationship between expected return and volatility is negative (Black, 1976), (Daouk & Ng, 2007), (Glosten, Jagannathan, & Runke, 1993), (Christie, 1982), while others come to the opposite conclusion (French, Schwert, & Stambaugh, 1987), (Campbell & Hentschel, 1992), (Jiang & Lee, 2004). In this thesis we define risky assets to have highly volatile stock prices and we will measure this volatility using the standard deviation of the returns.

In the course of our analysis, we will not rely on a factor model to determine whether our returns are abnormal or not, but rather use Monte Carlo simulation and Bootstrap methods[6]. We will assess the risk in a similar manner, and thus not compare it with marked risk.

---

[4] Capital Asset Pricing Model: A model for determining the expected returns for financial assets. The model was built from Harry Markowitz’s (1952) work on portfolio optimization, and was later developed by Sharpe (1964), Lintner (1965) and Mossin (1966).

[5] A 2009 study found a three- factor model using the factors; liquidity, market and size, suitable for the cross-section of Norwegian stock returns (Næs, Skjeltorp, & Ødegaard, 2009).

[6] Re-sampling Method for generating new samples by drawing with replacement from the original one. The method is used to assign measures of accuracy to sample estimates (Efron & Tibshirani, 1993).

1.2 Overview of content

In chapter 2 we will give a brief introduction of the data, followed by a description of the financial series we analyse, including arguments for using log-return series. Section 2.3, explains how we build, and implement, the test used to assess the significance level of the correlations. 2.4 shows that trends in the log-returns will not influence the correlation between series. Section 2.5 will deal with how we chose what stocks to analyse, and what trading strategies to use on them. Section 2.6 explains how we conduct our experiments, while chapter 3 summarizes the results that were found in 2.6. We finalize the thesis with a brief discussion of the process done.

2 Data and methods

2.1 The data

The financial series we analyse in this thesis are the adjusted closing prices found in TITLON[7]. TITLON is a database that contains daily financial data from Oslo Stock Exchange. The data dates back to 1980, and is updated every 6 months.

The Oslo Stock Exchange offers regulated markets for trading securities in Norway. Trading is done between 09:00 and 16:20 all days except weekends and holidays when the exchange is closed. Founded in 1819, the Oslo Stock Exchange has grown to be the world's leading exchange for fishery and aquaculture, managing the trades of companies worth a total of 2 billion NOK (Oct. 2016). The energy sector, which includes businesses related to oil and gas, accounts for around one third of the total market (oslobors.no, 2016).

Unlike most other databases that deals with Norwegian data, TITLON offers prices adjusted not only for corporate events such as stock splits, but also for cash or stock dividends. This means that the price is unbiased to these kinds of actions and will not influence the estimated correlation values used in this thesis.

All data used in our analysis, was imported in one long string, which we reassembled into tables. The price series were then made into series of logarithmic prices. Together with the price series we downloaded, we also got the names of the shares, their identification code, and the corresponding dates to every price in the series. The task of handling such a huge amount of data proved challenging in several ways, thus we found ourselves in need of removing some problematic shares from the data set. (E.g. shares containing only one price or only zeros). After the inspection and assembling of the raw data, we ended up with 811 log-price series to use in our analysis. The complete text files with the code used in this thesis is written in Mathematica[8] and is available in the appendix.

[7] https://titlon.uit.no/

[8] A global computation system developed by Wolfram Research.

2.2 Financial Time Series

Our financial time series data are sequences of prices $P(t)$ of stocks over a specific period of time. The percentage change in these prices at discrete time t, often referred to as the simple return, is defined as:

$$r'(t) = \frac{P(t+\Delta t) - P(t)}{P(t)} = \frac{P(t+\Delta t)}{P(t)} - 1 \tag{1}$$

The returns of a prediction system are one of the most important ways of measuring its performance.

Often, especially when looking at longer time periods, one can observe that the price changes depend on the price level (Mitchell, 1915). If we assume that the conditional standard deviation of the price can be expressed as

$$sd(P(t)|P(t) = m) \propto m$$

then a logarithmic transformation will give a Brownian process[9] and remove the problem of the level effect. The commonly used Black-Scholes (1973) model for the price of a stock is

$$dP(t) = \mu P(t)dt + \sigma P(t)dB(t) \tag{2}$$

which is a stochastic differential equation with a level effect. The last term is a normal random variable with a standard deviation of $\sigma P(t)$, i.e. proportional with the price, and $B(t)$ is a Brownian process. The solution to equation (2) is

$$P(t) = e^{\mu t + \sigma B(t)} \tag{3}$$

[9] A stochastic process with stationary independent increments.

Thus, we can express the returns as

$$r(t) = r'(t) + 1 = \frac{P(t+\Delta t)}{P(t)} = e^{\mu\Delta t + \sigma[B(t+\Delta t) - B(t)]} = e^{\mu\Delta t + \sigma\omega(t)}$$

where $\omega(t) = B(t+\Delta t) - B(t)$ is a stationary white noise process. This makes the log-returns stationary and without level- effect:

$$x(t) = \mu\Delta t + \sigma\omega(t) \tag{4}$$

When the returns are small, we can use the following approximation of the log-returns to the simple returns.

$$x(t) = \log\frac{P(t+\Delta t)}{P(t)} = \log P(t+\Delta t) - \log P(t) \approx r(t), \qquad r \ll 1.$$

The error of the log-return vs. the simple return is in $\boldsymbol{o}(|x|)$ and thus, for our daily returns, it is negligible.

A general way of modelling the log-return series, that is consistent with the EMH, is to consider it as a combination of a volatility process and a process that determines the direction of the movements

$$x(t) = \sigma(t)s(t) \tag{5}$$

Where $\sigma(t) > 0$ has persistent temporal dependence and describes the volatility of the process $x(t)$, and

$$s(t) = \begin{cases} +1 & r(t) > 0 \\ -1 & r(t) < 0 \end{cases} \tag{6}$$

describes the direction of the movements. The process (6) has independent (and uncorrelated) increments, as is required by the efficient marked hypothesis, and we assume that $s(t)$ is independent of $\sigma(t)$.

We have that when $E\big(x(t)\big) = 0$, the time-lagged co-variance of (5) is:

$$E(x(t)x(t+\Delta t)) = E(\sigma(t)\sigma(t+\Delta t))E(s(t)s(t+\Delta t)) = 0$$

If $E(s(t)s(t+\Delta t)) = 0$.

On the other hand, if we now look at the simple returns

$$r(t) = e^{x(t)} = 1 + x(t) + \frac{1}{2}x(t)^2 + \cdots \tag{7}$$

We can approximate (7) with

$$r(t) = 1 + x(t) + \frac{1}{2}x(t)^2 \tag{8}$$

and see that we do not get uncorrelated increments using (8) because

$$E(x(t)^2x(t+\Delta t)^2) = E(\sigma(t)^2\sigma(t+\Delta t)^2)E(s(t)^2s(t+\Delta t)^2) = E(\sigma(t)^2\sigma(t+\Delta t)^2)$$

The correlation in returns gets a contribution from the volatility persistence (the correlations of $\sigma(t)$), while the log-returns does not. We would have gotten the same results for any series we defined this way (5). Our reason for choosing to use log-returns (and not returns) is that we consider returns not to be stationary due to the previously mentioned level effect. We take the logarithm of all prices in our table and create log-return-series in the following way: assume share i has a series of prices P_i of length n. We create a corresponding series of log-returns of length $n-1$

$$x_i(t) = \log P_i(t+1) - \log P_i(t), \qquad t = 1, 2, \ldots, n-1.$$

We assume that the log returns $x_i(t)$ are independent, but that they do not necessarily follow the same distribution for every share i. In section 2.3.1, we will assume that the Black-Scholes model is correct and treat every series as they were normally distributed in

Method 1, while in Method 2 we will use estimations of the empirical distribution functions $\widehat{f_i}$ for $i = 1, \dots, 811$.

When we examined the log return series in our dataset, we found that all the series did not exhibit the same type of behaviour. In particular, we found that some series showed normal properties, while others did not. Rather than looking at each and every share, we considered them as a group and examined some of their aggregated statistical properties. We will use these results when we implement Method 1, that is when we simulate log-return series based on a known distribution:

$$r_i \sim Normal(\hat{\mu}, \hat{\sigma}) \tag{9}$$

Where expected value of the mean values of all the shares is $\hat{\mu} \approx 0$, and the expected value of the standard deviations of all the shares is $\hat{\sigma} \approx 0.06$.

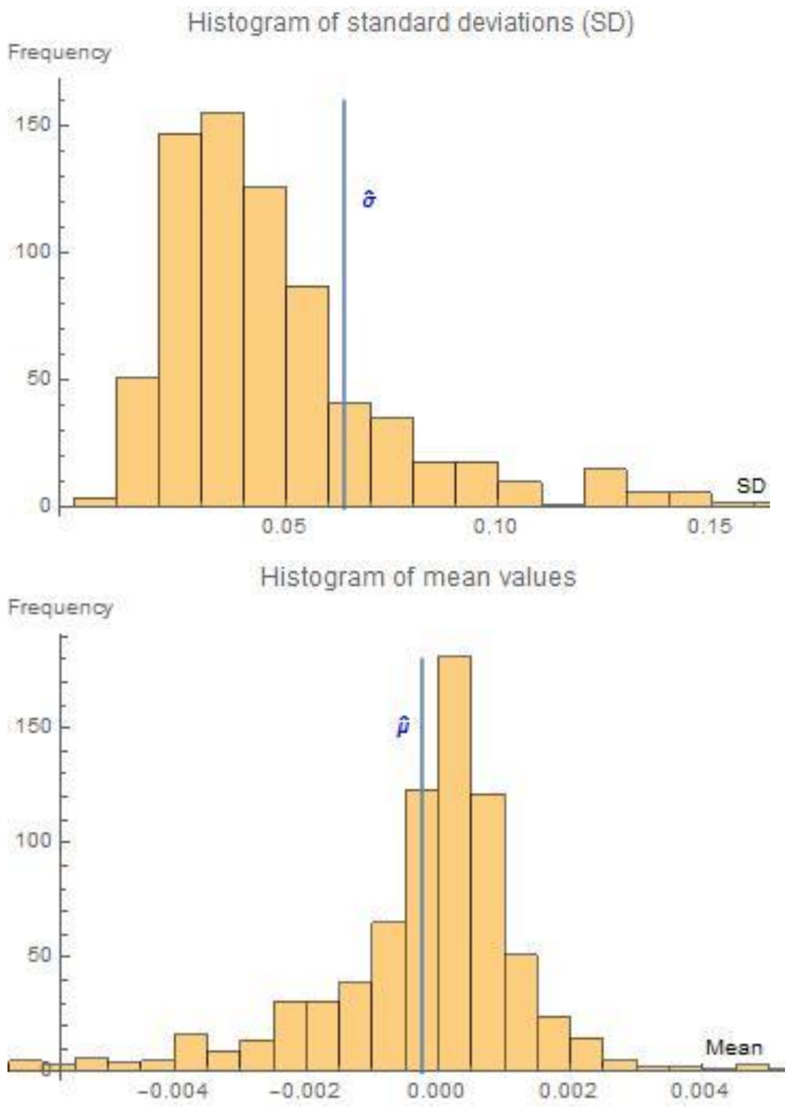


Figure 1. *Histograms of the mean values and standard deviation of the shares. Top: Histogram of the standard deviations of all 811 stocks. The blue vertical line indicates the location of $\hat{\sigma}$. Bottom: Histogram of the mean values of the stocks. The blue vertical line indicates the location of $\hat{\mu}$.*

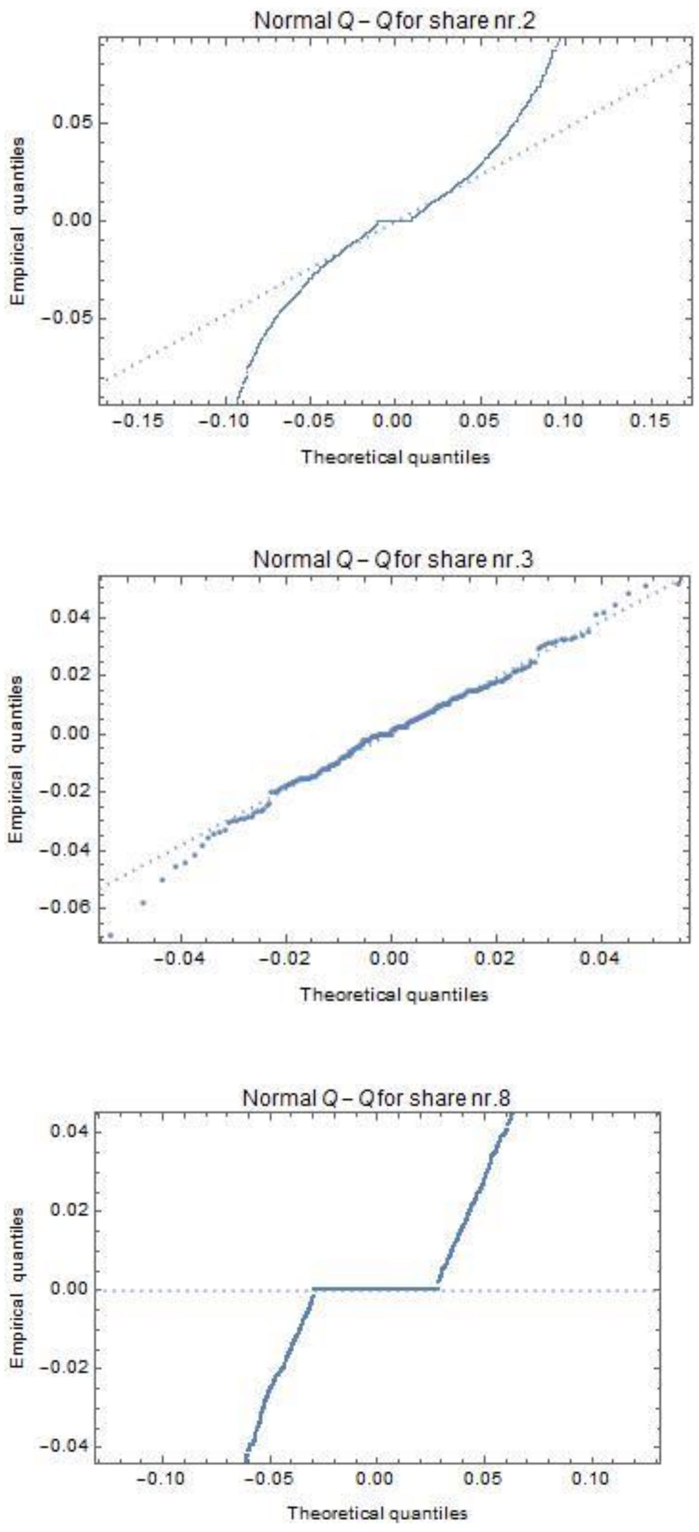


Figure 2. *Quantile-quantile plots of log-returns. Example of quantile plots of three different stocks. The first has thicker tails than that of the normal, the second looks approximately normal, the third has extremely heavy tails and a large number of values centred at zero.*

The problems with assuming normality is that the actual data seems not to possess the normal property, at least not consistently. Secondly, we have the issue of volatility persistence. To atone for these issues, we use a second and more preferable method using the empirical distribution function instead of the normal. We keep the results from Method 1 for comparative reasons.

Consider the log-return series $\underline{x} = x_{(1)}, \ldots x_{(n)}$ to be a sorted sample of independent identically distributed real random variables that follow the same underlying distribution F. The empirical distribution function is a step function that estimates F by jumping $\frac{1}{n}$ at each of the n observations in our sample:

$$\widehat{F_n}(x) = \frac{number\ of\ x_i \leq x}{n} = \sum_{i=1}^{n} \mathbf{1}_{x_i \leq x} \tag{10}$$

where $\mathbf{1}_{x_i \leq x}$ indicates the event $x_i \leq x$.

We know that[10]

$$\operatorname*{Sup}_{x} \left| \widehat{F_n}(x) - F(x) \right| \xrightarrow{n \to \infty} 0$$

and $\widehat{F_n}(x)$ is a non-parametric maximum likelihood estimator of $F(x)$.

Sampling from (10) is done by drawing with replacement from our original sample x.

Another way we could have improved our first method would have been to alter the Black and Scholes equation (2) such that it included a more complicated process:

$$dP(t) = \mu P(t)dt + \sigma P(t)dY(t)$$

where

$$\sigma(Y(t + \Delta t) - Y(t)) = \sigma(t)s(t).$$

[10] The Glivenko–Cantelli theorem (Glivenko, 1933) (Cantelli, 1933)

This is a combination of a volatility process and a direction process as defined in (6). These type of models include GARCH[11], simple volatility models, multi-fractal random walks, and more, which are used to model financial price series.

[11] Generalized Autoregressive Conditional Heteroscedasticity model.

2.3 The correlation matrices and how to test their significance

In this thesis we will create and analyse correlation matrices $C_{i,j}$ that holds correlation values between log-return series with one-day lag, for every pair of shares i,j in our data set.

To be able to examine different time periods separately, we need one matrix for every unique time period we wish to examine. This means that we have to alter the length of the log-return-series every time we create a new correlation matrix to fit that specific time period. As an example, say we are looking at the period 2012 -13, then we alter all return series in such a way that they only hold values corresponding to the year 2012-13, and then we precede with taking the correlations. We end up with a matrix that contains the correlations of the shares during that specific period only. When calculating the correlation between two shares, we can only compare the parts with equal dates in time. If, for example, one share has values dating from February to June 2012 and another has values from March to October the same year, then we cut both into series containing values from March to June 2012. We do the same for all combinations of shares.

The correlation $C_{i,j}$ between each pair of log-return series is calculated in the following way:

$$C_{i,j} = \frac{1}{T}\sum_{t=1}^{T} \frac{\left\{E\left(x_i(t)x_j(t+1)\right) - E\big(x_i(t)\big)E\left(x_j(t+1)\right)\right\}}{sd(x_i)sd\big(x_j\big)},$$

$$T = \min\big(length(x_i), length(x_j)\big)$$

Where $E(x_i(t))$ is the expectation of x_i at time t, and $sd(x_i)$ is the standard deviation of x_i.

These correlations make a square matrix with elements: $\left|C_{i,j}\right| \leq 1$, and unlike the correlation matrix without lag, these matrices will not be symmetric. Some shares will not have any values in the period we wish to examine, or the union between two series are empty sets, leaving the corresponding entries in the correlation matrix blank. The amount of blank spaces will wary with the different time periods we choose to analyse.

$$C_{i,j} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,811} \\ c_{2,1} & \ddots & [blank] & \vdots \\ \vdots & [blank] & \ddots & \vdots \\ c_{811,1} & \cdots & \cdots & c_{811,811} \end{bmatrix} \tag{11}$$

Where $c_{i,j}$ is the cross correlation between the log-return series of share i at time t and share j at time t

To get a sense of the behaviour of the correlation values between the shares in our dataset, we look at a histogram of all correlation estimates created using the 811 shares in the period 1980-2015, together with a Normal QQ-plot.

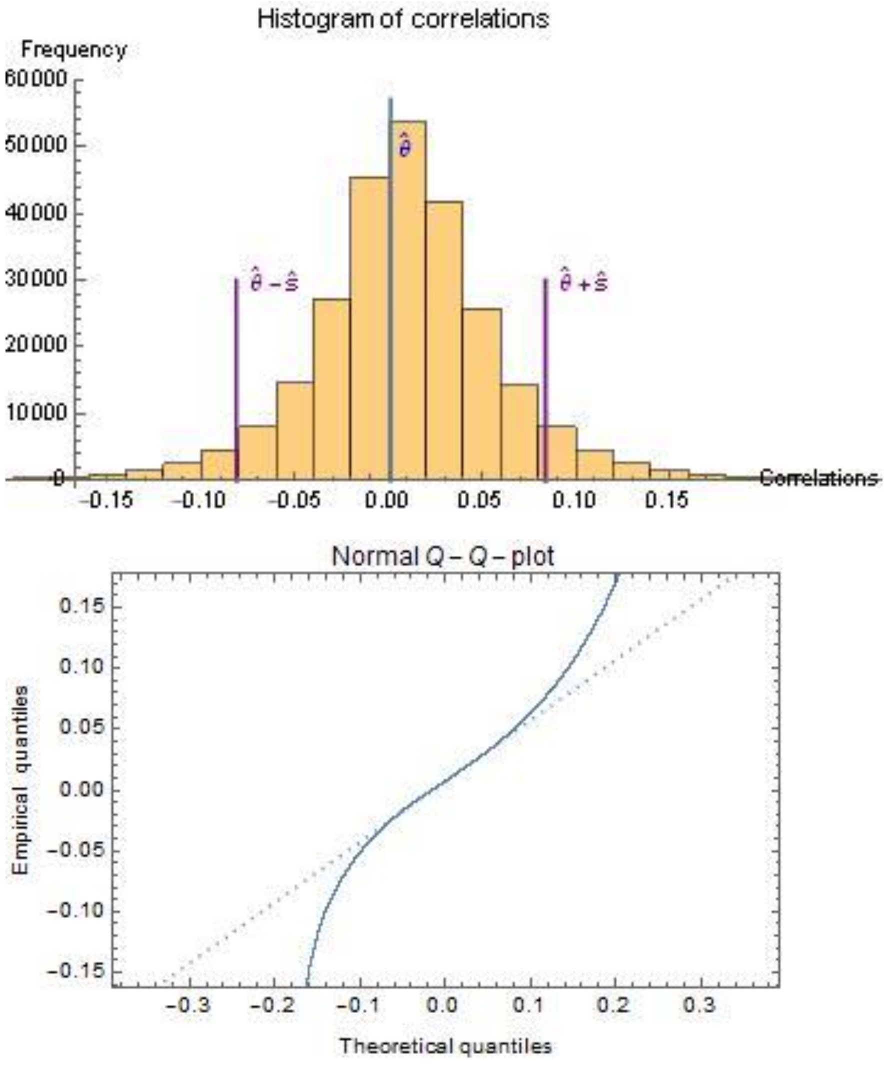


Figure 3. *Histogram and Q-Q-plot of correlation values between the shares. Top: Histogram of the estimated correlation values. Also showing the correlation mean (blue vertical line) and one standard deviation from the mean (purple vertical line). Bottom: Normal quantile-quantile plot of the correlation values.*

We can see that the estimated correlation values follow a symmetric distribution with zero mean, which means that on average we can expect the correlations to be close to zero, and since the distribution has thicker tails than that of the normal distribution, we can expect to get a higher frequency of large values.

We need a method to determine whether the correlations in our matrices are statistically significant or not, and how much correlation we should expect to get by chance if the log price series were indeed uncorrelated.

There are several ways we can build such tests, depending on the assumptions we make. We will use the following two methods with emphasis on the latter. Both methods involve re-sampling the test statistic $\hat{C}_{i,j}$ under some assumption about the distribution of the underlying data. The process results in B replications of the statistic $\theta_{i,j}^{*1}, \theta_{i,j}^{*2}, \dots, \theta_{i,j}^{*B}$. We will use these to assess the probability that our original estimates $\hat{C}_{i,j}$ come from that specific underlying distribution. If $\hat{C}_{i,j}$ deviates too much from our re-sampled correlation estimates, we will conclude that it did not come from that specific distribution, and thus is significant.

In the parametric method (Method 1) we assume normality in our returns series, and use simulation. With this method we can use the estimated value of the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the returns, and use Monte Carlo simulation. We sample the new log return series from the distribution (9) using our estimated values:

$$x_i^* \sim Normal(0\,, 0.06), \qquad i = 1,2, \dots, 811.$$

As mentioned in section 2.2 this is not necessarily a suitable simplification, and we therefore add a better (but computationally much slower) approach: Method 2.

In the non-parametric method (Method 2) we assume independency only, and build the tests using the estimated empirical distributions (10), for $i = 1, 2, \dots, 811$. With this method we allow the series to have varying forms of distributions, normal or otherwise. The idea is that our original series give the best population estimate, and we therefore repeat our experiment by drawing new samples from the original one.

For each of the correlation matrices described in 4.2, we make corresponding matrices created using simulated data. Hence, we need the simulated log-returns series to be of equal length of those used in the creation of the original matrix.

We repeat this process B times, stacking each of the simulated matrices unto one another, resulting in a matrix of dimension $[811 \times 811 \times B]$.

$$\Theta_{i,j}^{b} = \begin{bmatrix} \theta_{1,1}^{*1:B} & \theta_{1,2}^{*1:B} & \cdots & \theta_{1,811}^{*1:B} \\ \theta_{2,1}^{*1:B} & \ddots & [blank] & \vdots \\ \vdots & [blank] & \ddots & \vdots \\ \theta_{811,1}^{*1:B} & \cdots & \cdots & \theta_{811,811}^{*1:B} \end{bmatrix} \tag{12}$$

Each entry $\theta_{i,j}^{*1:B}$ holds B unique correlation values corresponding to pairs of uncorrelated simulated return series of length equal to that of the log-return series of share *i* and *j* in the original matrix.

The set of B values: $\theta_{i,j}^{*1:B}$, will be used to build a test and determine limits of significance for entry $C_{i,j}$ in the original matrix. The tests are created under the assumption that our null- hypothesis $H_0^{1,2}$ is true, that is:

For method 1,
H_0^1: That our return series are i.i.d. $Normal(0, 0.06)$.

For method 2,
H_0^2: That our series have independent increments that, within each series, follow the same unknown distribution f.

That is, under $H_0^{1,2}$ our correlation values, $C_{i,j}$ should follow the same distribution as the simulated estimates $\theta_{i,j}^{*1:B}$

$$H_0^{1,2}: C_{i,j} \sim \hat{g}\left(\theta_{i,j}^{*1:B}\right)$$

where $\hat{g}$ is the empirical distribution function of the correlations.

When doing a high number of simultaneous tests, we increase the rate of error for every additional test we do. In order to atone for this, we will use the Bonferroni correction (Bonferroni, 1936) to ensure that the significance level for the whole family of tests is $\alpha = 0.05$. That is, we test each individual hypothesis at a level of significance of $\frac{\alpha}{m}$. Where m is the number of tests performed.

To illustrate why we do this, consider a situation where we only do $m = 2$ tests with a level of significant of each at $\alpha = 0.05$, and we assume both null-hypothesis are true. Let $P(1)$ and $P(2)$ be the event of making the mistake of wrongly rejecting null-hypothesis number 1 and 2 respectively. Then the probability of rejecting a hypothesis wrongly is:

$$P(1) + P(2) - P(1 \cap 2) \leq 0.05 + 0.05$$

where $P(1 \cap 2)$ is unknown and depend on the relationship between the two tests.

We see that as the number of tests increase, we end up with an increasing probability of wrongfully rejecting a H_0 (type I error). This probability is however, always less than or equal to the number of tests performed times the level of significant for each test. This means that we can alter the level of significance for each test to $\alpha' = \frac{\alpha}{m}$ and receive a new bound for the whole family of test:

$$P(type\ I\ error) \leq m\alpha' = m\frac{\alpha}{m} = \alpha$$

If the number of tests performed, or the correlation between the tests is high, the Bonferroni method becomes conservative and we risk excepting too many hypotheses wrongfully (type II error). This means that some significant correlations might slip through our fingers.

2.3.1 Implementing the tests

We create the test matrices (12) using method 1 and method 2. In method 1, for each non-blank entry in the matrix, we do the following.

1) For $i = 1, \ldots, 811$, simulate $x_i^* = x_1, \ldots, x_k$ from the normal distribution (9) where k is the length of the original log-return series x_i
2) Estimate the correlation between x_i^* and x_j^* with one-day lag $\hat{\theta}_{i,j}^{*b}$ for every $i, j = 1, \ldots, 811$.
3) Repeat step 1 and 2 until we have B replications of the estimates: $\hat{\theta}_{i,j}^{*1}, \hat{\theta}_{i,j}^{*2}, \ldots, \hat{\theta}_{i,j}^{*B}$.

In method 2, we use the same procedure as we did in method 1, only now we sample x_i^* from the empirical distribution (10).

Under H_0 the values $\hat{\theta}_{i,j}^{*b}$ are equally likely values of $C_{i,j}$ and therefore we reject H_0 if $C_{i,j}$ falls too far away from the mean value of the $\hat{\theta}_{i,j}^{*b}{'s}$. In particular, we reject H_0 if $C_{i,j}$ falls outside the interval determined by $\hat{\theta}_{i,j}^{High}$ and $\hat{\theta}_{i,j}^{Low}$in such a way that:

$$P\left(C_{i,j} \in \left(\hat{\theta}_{i,j}^{Low}, \hat{\theta}_{i,j}^{High}\right)\right) = 1 - \frac{2\alpha}{m},$$

$$where\ \alpha = 0.05, \qquad m = number\ of\ non - blank\ enrties\ in\ matrix$$

To determine the limits $\hat{\theta}_{i,j}^{Low}, \hat{\theta}_{i,j}^{High}$, we use a percentile method based on sorting the estimated correlations $\hat{\theta}_{i,j}^{*b}$ generated by our sampling procedure (Efron & Tibshirani, 1993, p. 171) This is appropriate because our distribution is symmetric, and because it is the fastest computationally wise[12].

We sort the $\hat{\theta}_{i,j}^{*b}$ and find the quantiles for the $\frac{\alpha}{2}$ percentiles of the highest and lowest values.

$$\hat{\theta}_{i,j}^{Low} = \hat{\theta}_{i,j}^{(*)}[N_L]$$
$$\hat{\theta}_{i,j}^{High} = \hat{\theta}_{i,j}^{(*)}[N_H]$$

[12] Despite its simplicity, these matrices took 80-90 hours each to compute using this method.

Where N_L denotes the $B\left(\frac{\alpha}{2}\right)'th$ place of the sorted correlation estimates $\hat{\theta}_{i,j}^{(*)}$, and N_H denotes the $B\left(1-\frac{\alpha}{2}\right)'th.$

We are interested in knowing whether a particular null-hypothesis was rejected or not, and which log-returns series that were involved. Thus, we save the information as a “1” if the null-hypothesis was rejected, and as a “0” if it was not. This results in matrices consisting of either zeros or ones in the entries corresponding to the appropriate log- return series.

$$R_{i,j} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,811} \\ r_{2,1} & \ddots & [blank] & \vdots \\ \vdots & [blank] & \ddots & \vdots \\ r_{811,1} & \cdots & \cdots & r_{811,811} \end{bmatrix}, \qquad (13)$$
$$r_{i,j} = \begin{cases} 1\ if\ H_0\ is\ rejected \\ 0\ if\ H_0\ is\ not\ rejected \end{cases}$$

The matrices tell us which shares correlates significantly with which shares with one-day lag in a particular pre-determined time interval. We illustrate the relationships in **Figure 4**, where we show 30 shares and how they significantly relate to one another in the period 01.01.14-01.0715.

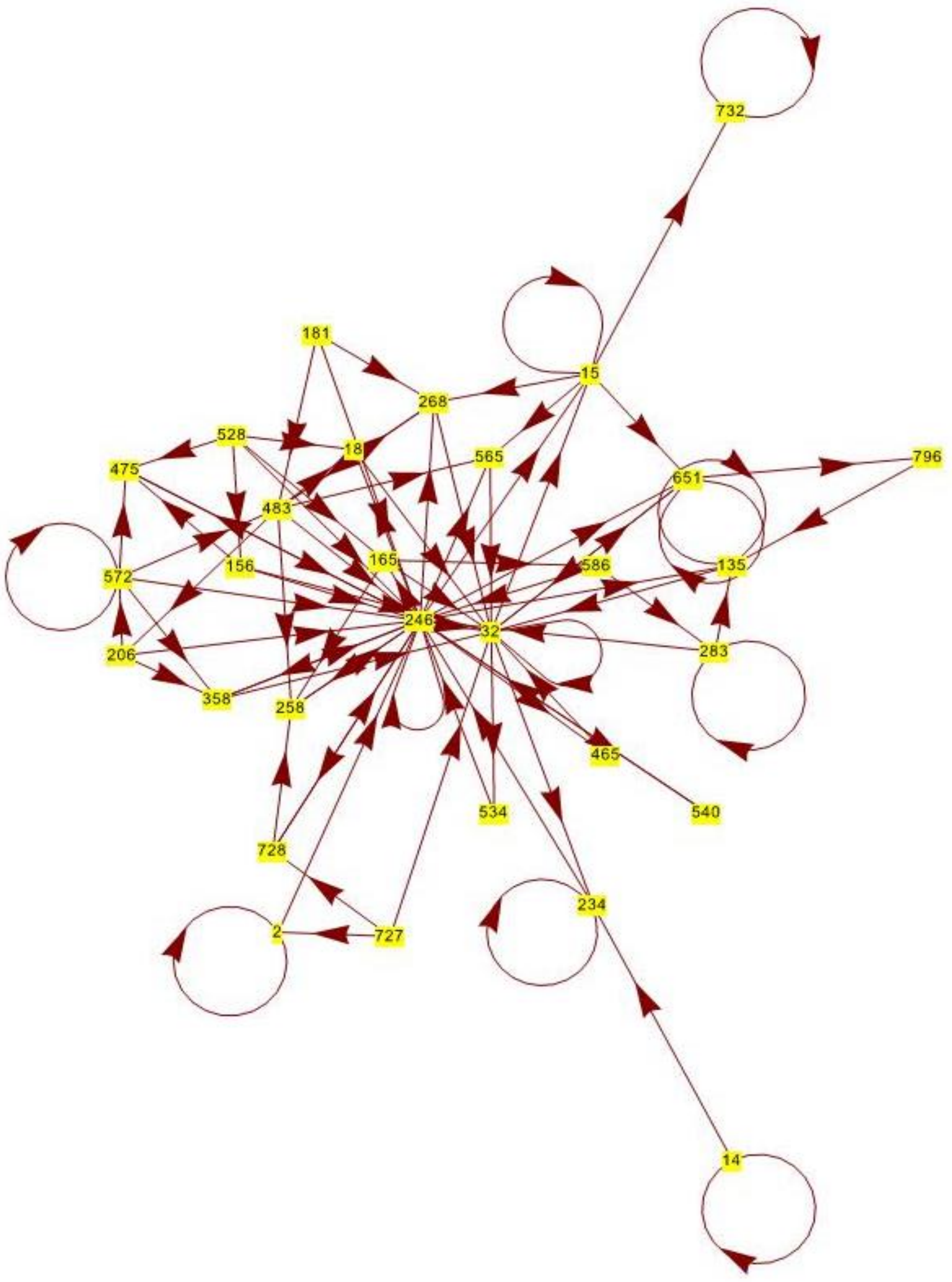


Figure 4. *Predictions.*

The illustration shows shares that have significant correlation between them. Arrows from share i to share j indicates that share i predicts share j the next day. The plot shows 30 shares in the period January 1st 2014 to August 1st 2015.

2.4 Removing the trend

One interesting question is whether the return series have a linear trend caused by the inflation rate, and whether this creates correlation between our series. If it does, we should see fewer significant correlations in our correlation matrix when we remove the trend, as well as a decrease in correlation as a whole. We will look at this theoretically and show that the trend component does not influence the correlation.

Using equation (3) we derived an expression for the log-return series (4) that had a linear trend. We consider two such series:

$$x_1(t) = \mu_1 \Delta t + \omega_1(t), \qquad \sigma_{x_1} = \sigma_{\omega_1}$$
$$x_2(t) = \mu_2 \Delta t + \omega_2(t), \qquad \sigma_{x_2} = \sigma_{\omega_2}$$

Where the variance of the process only depends on the variance of the term $\omega_1(t)$.

We look at how the removal of the trend should affect our results. The correlation between the log-returns of one share at time t and another share time $t + \Delta t$ is:

$$cor\left(\left(x_1(t), x_2(t+\Delta t)\right)\right) = \frac{E\big(x_1(t)\, x_2(t+\Delta t)\big) - E\big(x_1(t)\big)E\big(x_2(t+\Delta t)\big)}{\sigma_{x_1}\sigma_{x_2}}$$

$$= \frac{E\left(\big(\mu_1\Delta t + \omega_1(t)\big)\,\big(\mu_2\Delta t + \omega_2(t+\Delta t)\big)\right) - E\big(\mu_1\Delta t + \omega_1(t)\big)E\big(\mu_2\Delta t + \omega_2(t+\Delta t)\big)}{\sigma_{x_1}\sigma_{x_2}}$$

$$= \frac{\mu_1\Delta t\mu_2\Delta t + 0 + 0 + E\left(\left(\omega_1(t)\omega_2(t+\Delta t)\right)\right) - \mu_1\Delta t\mu_2\Delta t}{\sigma_{x_1}\sigma_{x_2}} = \frac{E\left(\left(\omega_1(t)\omega_2(t+\Delta t)\right)\right)}{\sigma_{x_1}\sigma_{x_2}}$$

$$= cor\left(\left(\omega_1(t)\omega_2(t+\Delta t)\right)\right)$$

We see that the correlation does not depend on the trend, and thus we should not see any change in correlation.

To do a quick test of the theoretical result, we find the trends, μ_i and remove them from their respective log-price series. We compare the matrix $C_{i,j}$ with a new matrix created in the same way, only now using the log-price series without trend.

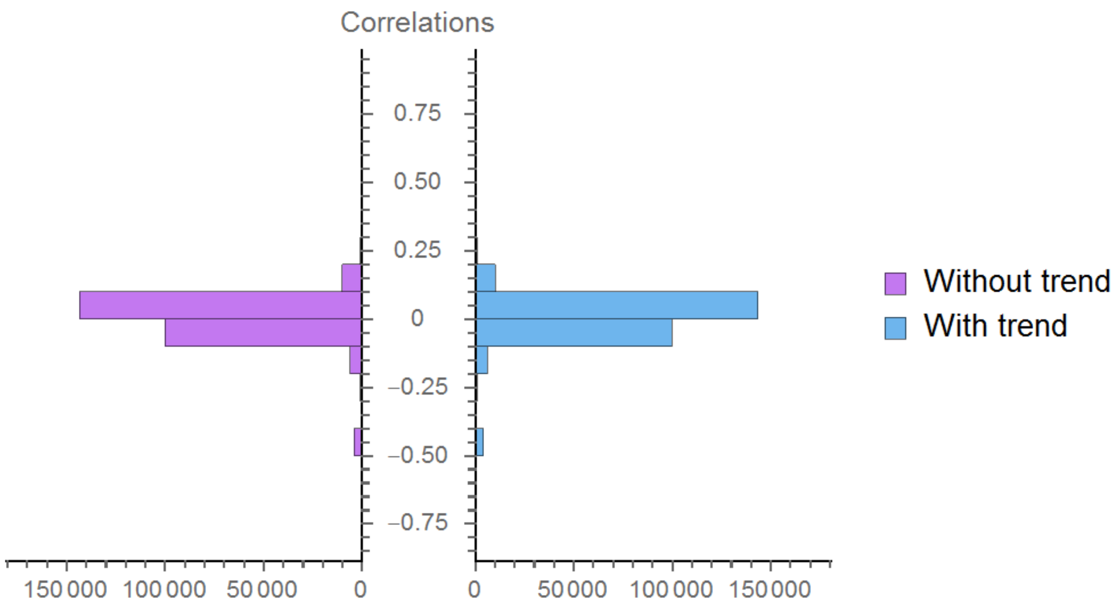


Figure 5. *Paired Histogram of correlations before and after removing the trends*

The conclusion is that the change in correlations before and after removal of the trend is non-existing, which is consistent with the theory. We will precede our analysis using the unchanged original log-price series.

2.5 The prediction system

2.5.1 What shares to examine further

In developing a system for predicting movements in the log-return series, our goal is to find a pattern in the cross correlation matrices (11) that enable us to earn abnormal returns by trading on one stock j on the basis of monitoring the movements of another share i.

Rather than looking at the entire dataset as a whole, we want to examine shorter intervals to better understand the potential continuity of correlations throughout history. To start with, we look at two-year periods from year 2006 to year 2015. If we find significant correlations between shares that repeat in all, or in many of these intervals, we will look further back in time on these particular shares.

For each time period we create a new $R_{i,j}^{T}$ matrix as described in equation (13) with periods

$$T = 06, 08, 10, 12, 14,$$

corresponding to the years: $\{\{2006-2008\},\{2008-2010\},\{2010-2012\},\{2012-2014\},\{2014-2015\}\}$ respectively.

We want to find out if the correlation between shares is similar in the consecutive periods, specifically, if it is the same shares that are associated with each other in every period. When analysing the $R_{i,j}^{T}$ matrices, we find what shares have significant correlations with what shares in each of the periods T, and for every series $x_i(t)$, that has significant correlations with $x_j(t+1)$, we separate between the following:

1) Share i correlates significantly with other shares (including itself) in both periods, T_1 and T_2.
2) Share i auto correlates significantly in both periods, T_1 and T_2.
3) Share i cross-correlates with share j in both periods, T_1 and T_2.

We count the number of shares fitting each of the categories above. The following table describes the results found.

		Method 1	Method 2	Method 1	Method 2	Method 1	Method 2
$Period: T_1$	$Period: T_2$	**1**)	**1**)	**2**)	**2**)	**3**)	**3**)
12	14	61	159	34	66	1	12
08	10	53	120	35	66	3	19
06	10	62	111	31	49	1	3
10	14	64	133	42	62	1	10
08	14	40	127	18	47	0	22
06	14	52	116	17	38	0	3
10	12	71	129	55	81	1	5
08	12	32	113	19	53	0	5
06	12	51	102	22	45	0	1
06	08	55	106	24	46	0	4

Table 1. *Overview of the amount of significant correlations.*
The two first columns describes the periods we compare, followed by two columns describing the amount of shares that correlates significantly with other shares in both these periods. The next two columns show the amount of shares that has significant auto correlations in both periods. The last two columns describe the amount of shares that have cross correlation with share j in both periods. The header of the table tells us what method is used when deriving the numbers in the table. Method 1 refers to the normality assumption used when setting limits of significance, while Method 2 refers to the assumption of independency

We see a greater number of significant correlations in all categories using Method 2. Since we are solely interested in the cross-correlations, we take a closer look at the shares corresponding to category 3). That is, the pairs of shares that kept a significant cross correlation throughout two (or more) periods. We fetch the identification number of these shares, and summarize the results in Table 2.

T_1	T_2	$Method\ 1$	$Method\ 2$
12	14	$587 \rightarrow 798$	$025 \rightarrow 483$ $045 \rightarrow 507$ $563 \rightarrow 135$ $025 \rightarrow 508$ $206 \rightarrow 284$ $587 \rightarrow 431$ $032 \rightarrow 179$ $289 \rightarrow 328$ $621 \rightarrow 061$ $032 \rightarrow 686$ $311 \rightarrow 798$ $795 \rightarrow 538$
08	10	$174 \rightarrow 788$ $796 \rightarrow 014$ $796 \rightarrow 357$	$139 \rightarrow 333$ $206 \rightarrow 174$ $796 \rightarrow 014$ $174 \rightarrow 246$ $528 \rightarrow 305$ $796 \rightarrow 135$ $174 \rightarrow 319$ $528 \rightarrow 486$ $796 \rightarrow 179$ $174 \rightarrow 597$ $528 \rightarrow 677$ $796 \rightarrow 246$ $174 \rightarrow 602$ $540 \rightarrow 037$ $796 \rightarrow 319$ $174 \rightarrow 788$ $579 \rightarrow 540$ $796 \rightarrow 357$ $796 \rightarrow 788$
06	10	$621 \rightarrow 658$	$558 \rightarrow 503$ $621 \rightarrow 358$ $621 \rightarrow 658$
10	14	$030 \rightarrow 639$	$037 \rightarrow 795$ $402 \rightarrow 483$ $738 \rightarrow 015$ $037 \rightarrow 801$ $501 \rightarrow 508$ $738 \rightarrow 311$ $315 \rightarrow 415$ $660 \rightarrow 800$ $796 \rightarrow 135$ $795 \rightarrow 789$
08	14	$-$	$014 \rightarrow 234$ $165 \rightarrow 586$ $528 \rightarrow 018$ $015 \rightarrow 651$ $181 \rightarrow 268$ $528 \rightarrow 156$ $015 \rightarrow 732$ $246 \rightarrow 465$ $528 \rightarrow 246$ $032 \rightarrow 234$ $246 \rightarrow 540$ $528 \rightarrow 475$ $032 \rightarrow 534$ $246 \rightarrow 565$ $540 \rightarrow 246$ $165 \rightarrow 032$ $246 \rightarrow 728$ $572 \rightarrow 358$ $165 \rightarrow 258$ $483 \rightarrow 206$ $727 \rightarrow 002$ $796 \rightarrow 135$
06	14	$-$	$283 \rightarrow 246$ $686 \rightarrow 602$ $749 \rightarrow 658$
10	12	$443 \rightarrow 596$	$443 \rightarrow 596$ $563 \rightarrow 636$ $622 \rightarrow 174$ $607 \rightarrow 808$ $636 \rightarrow 679$
08	12	$-$	$032 \rightarrow 199$ $501 \rightarrow 165$ $761 \rightarrow 165$ $540 \rightarrow 287$ $796 \rightarrow 200$
06	12	$-$	$686 \rightarrow 727$
06	08	$-$	$17 \rightarrow 253$ $020 \rightarrow 092$ $020 \rightarrow 699$ $658 \rightarrow 253$

Table 2. *Correlations between shares that remain significant throughout different periods. The first two columns describe what periods in time we compare, the third shows what pairs has significant cross correlations in both these periods using Method 1, and the last shows the same only when using Method 2.*

One problem related to the correlation described with the $C_{i,j}$ matrices is whether the correlation is caused by extremely high correlation some days, rather than consistent every-day correlation. If the correlation comes from only a few days, it will most likely be difficult for us to create reliable predictions on account of them. To eliminate this problem, we look at the absolute value of the log-returns series and remove the days that cause high values relative to the other elements in the series. If we only need to remove a few days to get non-significant correlation, i.e. $C_{i,j} \in \left(\hat{\theta}_{i,j}^{Low}, \hat{\theta}_{i,j}^{High}\right)$, we have an indication that the correlation came from high correlation on a few days. If we, on the other hand, need to remove many days to see a decline in correlation, we might have correlation on a high number of days, increasing our chances of finding a reliable pattern. We summarize the results in Table 3. below.

T	$i \to j$	$\hat{\theta}_{i,j}^{Low}, \hat{\theta}_{i,j}^{High}$	$C_{i,j}$	$\#days\ removed$	$C_{i,j}^{*}$
14	$025 \to 483$	$-0.113, 0.139$	-0.149	1	0.019
14	$025 \to 508$	$-0.094, 0.168$	-0.136	1	0.012
14	$032 \to 179$	$-0.069, 0.093$	-0.108	1	-0.026
14	$032 \to 686$	$-0.088, 0.119$	-0.363	All	$High$
14	$045 \to 507$	$-0.079, 0.118$	-0.092	$1-76$	-0.064
14	$206 \to 284$	$-0.095, 0.106$	-0.147	$1-10$	-0.083
14	$289 \to 328$	$-0.114, 0.137$	-0.195	1	-0.036
14	$311 \to 798$	$-0.118, 0.133$	-1	$-$	$-$
14	$563 \to 135$	$-0.083, 0.151$	-0.091	$1-12$	-0.075
14	$587 \to 431$	$-0.132, 0.153$	-0.134	$1-7$	-0.058
14	$621 \to 061$	$-0.074, 0.110$	-0.130	$1-11$	-0.038
14	$795 \to 538$	$-0.117, 0.108$	-0.818	$-$	$-$
14	$037 \to 795$	$-0.161, 0.172$	-0.663	$-$	$-$
14	$037 \to 801$	$-0.121, 0.124$	-0.243	$1-2$	-0.114
14	$315 \to 415$	$-0.080, 0.152$	-0.111	$1-35$	-0.079
14	$402 \to 483$	$-0.149, 0.096$	-0.266	$1-5$	-0.063
14	$501 \to 508$	$-0.097, 0.104$	-0.133	1	0.045
14	$660 \to 800$	$-$	$-$	$-$	$-$
14	$738 \to 015$	$-0.168, 0.116$	-0.183	$1-7$	-0.129
14	$738 \to 311$	$-0.110, 0.122$	-0.113	$1-11$	0.092
14	$795 \to 789$	$-$	$-$	$-$	$-$

14	$014 \rightarrow 234$	$-0.123, 0.120$	-0.126	1	-0.122
14	$015 \rightarrow 651$	$-0.115, 0.102$	-0.128	$1-2$	-0.047
14	$015 \rightarrow 732$	$-0.121, 0.108$	-0.151	$1-11$	-0.114
14	$032 \rightarrow 234$	$-0.122, 0.112$	-0.428	All	$High$
14	$032 \rightarrow 534$	$-0.107, 0.103$	-0.164	$1-2$	-0.063
14	$165 \rightarrow 032$	$-0.147, 0.166$	-0.205	1	0.007
14	$165 \rightarrow 258$	$-0.091, 0.166$	-0.103	$1-22$	-0.072
14	$165 \rightarrow 586$	$-0.097, 0.140$	-0.112	$1-2$	-0.094
14	$181 \rightarrow 268$	$-0.085, 0.116$	-0.113	1	-0.055
14	$246 \rightarrow 465$	$-$	$-$	$-$	$-$
14	$246 \rightarrow 540$	$-$	$-$	$-$	$-$
14	$246 \rightarrow 565$	$-$	$-$	$-$	$-$
14	$246 \rightarrow 728$	$-$	$-$	$-$	$-$
14	$483 \rightarrow 206$	$-0.108, 0.127$	-0.141	All	$High$
14	$528 \rightarrow 018$	$-0.104, 0.124$	-0.122	$1-103$	-0.102
14	$528 \rightarrow 156$	$-0.122, 0.105$	-0.130	$1-3$	-0.110
14	$528 \rightarrow 246$	$-$	$-$	$-$	$-$
14	$528 \rightarrow 475$	$-0.085, 0.127$	-0.156	$1-28$	-0.074
14	$540 \rightarrow 246$	$-$	$-$	$-$	$-$
14	$572 \rightarrow 358$	$-0.127, 0.108$	-0.143	$1-6$	-0.058
14	$727 \rightarrow 002$	$-0.094, 0.114$	-0.107	1	-0.027
14	$796 \rightarrow 135$	$-0.087, 0.141$	-0.100	$1-27$	0.008
14	$587 \rightarrow 798$	$-$	$-$	$-$	$-$
14	$30 \rightarrow 639$	$-0.135, 0.146$	-0.150	$1-2$	0.050
14	$283 \rightarrow 246$	$-$	$-$	$-$	$-$
14	$686 \rightarrow 602$	$-0.104, 0.104$	-0.149	$1-2$	-0.073
14	$749 \rightarrow 658$	$-0.111, 0.111$	-0.115	$1-2$	-0.105
12	$025 \rightarrow 483$	$-0.113, 0.139$	-0.115	$1-6$	-0.112
12	$025 \rightarrow 508$	$-0.094, 0.168$	-0.108	$1-13$	-0.070
12	$032 \rightarrow 179$	$-0.069, 0.093$	-0.117	$1-2$	-0.080
12	$032 \rightarrow 686$	$-0.088, 0.119$	-0.125	$1-16$	-0.085
12	$045 \rightarrow 507$	$-0.079, 0.118$	-0.095	$1-4$	-0.076

12	$206 \rightarrow 284$	$-0.095, 0.106$	-0.114	$1-8$	-0.090
12	$289 \rightarrow 328$	$-0.114, 0.137$	-0.126	$1-2$	-0.032
12	$311 \rightarrow 798$	$-0.118, 0.133$	-0.146	$1-5$	0.038
12	$563 \rightarrow 135$	$-0.083, 0.151$	-0.088	$1-2$	-0.082
12	$587 \rightarrow 431$	$-0.132, 0.153$	-0.152	$1-7$	-0.102
12	$621 \rightarrow 061$	$-0.074, 0.110$	-0.110	$1-7$	-0.059
12	$795 \rightarrow 538$	$-0.117, 0.108$	-0.137	$1-5$	-0.106
12	$443 \rightarrow 596$	$-0.114, 0.224$	-0.151	1	-0.029
12	$563 \rightarrow 636$	$-$	$-$	$-$	$-$
12	$607 \rightarrow 808$	$-0.095, 0.111$	-0.099	$1-3$	-0.088
12	$622 \rightarrow 174$	$-0.114, 0.102$	-0.135	$1-15$	-0.106
12	$636 \rightarrow 679$	$-$	$-$	$-$	$-$
12	$032 \rightarrow 199$	$-0.092, 0.130$	-0.134	$1-7$	-0.092
12	$501 \rightarrow 165$	$-0.095, 0.109$	-0.109	$1-42$	-0.092
12	$540 \rightarrow 287$	$-0.104, 0.102$	-0.117	$1-4$	0.101
12	$761 \rightarrow 165$	$-0.091, 0.111$	-0.129	$1-10$	-0.070
12	$796 \rightarrow 200$	$-0.113, 0.102$	-0.130	$1-8$	-0.112
12	$587 \rightarrow 798$	$-0.178, 0.180$	-0.166	1	-0.170
12	$686 \rightarrow 727$	$-0.115, 0.110$	-0.169	$1-299$	-0.105
10	$037 \rightarrow 795$	$-0.163, 0.185$	-0.214	$1-24$	-0.143
10	$037 \rightarrow 801$	$-0.154, 0.215$	-0.194	$1-15$	-0.152
10	$315 \rightarrow 415$	$-0.116, 0.106$	-0.125	$1-7$	-0.102
10	$402 \rightarrow 483$	$-0.151, 0.142$	-0.152	$1-4$	-0.075
10	$501 \rightarrow 508$	$-0.095, 0.116$	-0.153	1	-0.089
10	$660 \rightarrow 800$	$-0.144, 0.203$	-0.181	$1-4$	-0.141
10	$738 \rightarrow 015$	$-0.083, 0.200$	-0.139	$1-4$	-0.026
10	$738 \rightarrow 311$	$-0.089, 0.157$	-0.131	$1-91$	-0.037
10	$795 \rightarrow 789$	$-$	$-$	$-$	$-$
10	$796 \rightarrow 135$	$-0.114, 0.154$	-0.117	1	-0.104
10	$139 \rightarrow 333$	$-0.097, 0.150$	-0.118	$1-4$	-0.073
10	$174 \rightarrow 246$	$-0.092, 0.099$	-0.115	$1-2$	-0.086
10	$174 \rightarrow 319$	$-0.094, 0.136$	-0.110	$1-5$	-0.086

10	$174 \rightarrow 597$	$-0.103, 0.137$	-0.162	$1-23$	-0.090
10	$174 \rightarrow 602$	$-0.078, 0.109$	-0.083	1	-0.077
10	$174 \rightarrow 788$	$-0.105, 0.120$	-0.120	$1-13$	-0.101
10	$206 \rightarrow 174$	$-0.110, 0.099$	-0.111	1	-0.110
10	$528 \rightarrow 305$	$-0.083, 0.147$	-0.102	1	-0.083
10	$528 \rightarrow 486$	$-0.090, 0.136$	-0.091	1	-0.089
10	$528 \rightarrow 677$	$-0.127, 0.119$	-0.141	$1-24$	-0.085
10	$540 \rightarrow 037$	$-0.100, 0.125$	-0.109	$1-2$	-0.092
10	$579 \rightarrow 540$	$-0.128, 0.116$	-0.152	$1-3$	-0.065
10	$796 \rightarrow 014$	$-0.124, 0.138$	-0.169	$1-7$	-0.121
10	$796 \rightarrow 179$	$-0.098, 0.090$	-0.104	1	-0.081
10	$796 \rightarrow 246$	$-0.125, 0.109$	-0.145	$1-2$	-0.114
10	$796 \rightarrow 319$	$-0.118, 0.121$	-0.118	1	-0.106
10	$796 \rightarrow 357$	$-0.096, 0.096$	-0.149	$1-41$	-0.094
10	$796 \rightarrow 788$	$-0.107, 0.132$	-0.110	$1-2$	-0.106
10	$443 \rightarrow 596$	$-0.393, 0.378$	-0.545	$1-5$	-0.387
10	$563 \rightarrow 636$	$-too\ few$	$-$	$-$	$-$
10	$607 \rightarrow 808$	$-0.103, 0.121$	-0.117	$1-8$	-0.100
10	$622 \rightarrow 174$	$-0.118, 0.119$	-0.118	1	-0.101
10	$636 \rightarrow 679$	$-too\ few$	$-$	$-$	$-$
10	$558 \rightarrow 503$	$-0.305, 0.265$	-0.308	1	-0.276
10	$621 \rightarrow 358$	$-0.102, 0.126$	-0.136	$1-7$	-0.101
10	$621 \rightarrow 658$	$-0.123, 0.133$	-0.169	$1-9$	-0.066
08	$139 \rightarrow 333$	$-0.097, 0.150$	-0.113	1	-0.082
08	$174 \rightarrow 246$	$-0.092, 0.099$	-0.161	$1-2$	-0.091
08	$174 \rightarrow 319$	$-0.094, 0.136$	-0.094	1	-0.080
08	$174 \rightarrow 597$	$-0.103, 0.137$	-0.154	$1-11$	-0.103
08	$174 \rightarrow 602$	$-0.078, 0.109$	-0.126	$1-20$	-0.041
08	$174 \rightarrow 788$	$-0.105, 0.120$	-0.149	$1-4$	-0.104
08	$206 \rightarrow 174$	$-0.110, 0.09$	-0.113	$1-14$	-0.019
08	$528 \rightarrow 305$	$-0.083, 0.147$	-0.123	$1-3$	0.055
08	$528 \rightarrow 486$	$-0.090, 0.136$	-0.139	$1-10$	-0.088

08	$528 \rightarrow 677$	$-0.127, 0.119$	-0.141	1	-0.014
08	$540 \rightarrow 037$	$-0.100, 0.125$	-0.148	$1-3$	-0.100
08	$579 \rightarrow 540$	$-0.128, 0.116$	-0.218	$1-2$	-0.120
08	$796 \rightarrow 014$	$-0.124, 0.138$	-0.144	1	-0.122
08	$796 \rightarrow 135$	$-0.114, 0.154$	-0.152	$1-18$	-0.111
08	$796 \rightarrow 179$	$-0.098, 0.090$	-0.173	$1-10$	-0.096
08	$796 \rightarrow 246$	$-0.125, 0.109$	-0.216	$1-13$	-0.115
08	$796 \rightarrow 319$	$-0.118, 0.121$	-0.118	1	-0.100
08	$796 \rightarrow 357$	$-0.096, 0.096$	-0.140	$1-3$	-0.039
08	$796 \rightarrow 788$	$-0.163, 0.118$	-0.202	$1-7$	-0.092
08	$014 \rightarrow 234$	$-0.168, 0.116$	-0.183	$1-7$	-0.129
08	$015 \rightarrow 651$	$-0.096, 0.131$	-0.096	1	-0.033
08	$015 \rightarrow 732$	$-0.110, 0.110$	-0.112	$1-4$	-0.026
08	$032 \rightarrow 234$	$-0.099, 0.090$	-0.436	$1-12$	0.070
08	$032 \rightarrow 534$	$-0.094, 0.111$	-0.100	$1-143$	-0.041
08	$165 \rightarrow 032$	$-$	$-$	$-$	$-$
08	$165 \rightarrow 258$	$-$	$-$	$-$	$-$
08	$165 \rightarrow 586$	$-$	$-$	$-$	$-$
08	$181 \rightarrow 268$	$-0.104, 0.077$	-0.135	1	0.015
08	$246 \rightarrow 465$	$-0.110, 0.103$	-0.139	1	-0.102
08	$246 \rightarrow 540$	$-0.105, 0.121$	-0.128	All	$High$
08	$246 \rightarrow 565$	$-0.128, 0.138$	-0.177	1	-0.111
08	$246 \rightarrow 728$	$-0.110, 0.087$	-0.134	$1-3$	-0.083
08	$483 \rightarrow 206$	$-0.098, 0.086$	-0.103	1	-0.089
08	$528 \rightarrow 018$	$-0.120, 0.162$	-0.153	1	-0.030
08	$528 \rightarrow 156$	$-0.150, 0.156$	-0.241	1	-0.010
08	$528 \rightarrow 246$	$-0.124, 0.113$	-0.303	1	-0.031
08	$528 \rightarrow 475$	$-0.118, 0.107$	-0.224	1	-0.078
08	$540 \rightarrow 246$	$-0.103, 0.106$	-0.103	$1-2$	-0.082
08	$572 \rightarrow 358$	$-0.083, 0.087$	-0.087	$1-2$	-0.068
08	$727 \rightarrow 002$	$-0.098, 0.130$	-0.117	1	-0.068
08	$032 \rightarrow 199$	$-0.099, 0.135$	-0.101	$1-14$	-0.094

08	$501 \to 165$	$-$	$-$	$-$	$-$
08	$540 \to 287$	$-0.124, 0.111$	-0.141	$1-5$	-0.117
08	$761 \to 165$	$-$	$-$	$-$	$-$
08	$796 \to 200$	$-0.128, 0.099$	-0.136	1	-0.123
08	$17 \to 253$	$-0.107, 0.111$	-0.115	1	-0.099
08	$20 \to 92$	$-0.255, 0.235$	-0.388	1	-0.204
08	$20 \to 699$	$-0.247, 0.253$	-0.273	1	-0.192
08	$658 \to 253$	$-0.106, 0.125$	-0.127	$1-8$	-0.086
06	$017 \to 253$	$-0.114, 0.149$	-0.127	$1-4$	-0.080
06	$020 \to 092$	$-0.110, 0.100$	-0.110	$1-2$	-0.101
06	$020 \to 699$	$-0.111, 0.126$	-0.181	$1-3$	-0.101
06	$658 \to 253$	$-0.109, 0.162$	-0.131	$1-4$	-0.091
06	$283 \to 246$	$-0.218, 0.255$	-0.245	1	-0.180
06	$686 \to 602$	$-0.118, 0.108$	-0.146	$1-30$	-0.104
06	$749 \to 658$	$-0.126, 0.125$	-0.127	1	-0.114
06	$558 \to 503$	$-0.119, 0.144$	-0.143	$1-15$	0.117
06	$621 \to 358$	$-0.119, 0.111$	-0.123	1	-0.118
06	$621 \to 658$	$-0.155, 0.116$	-0.157	1	-0.122
06	$686 \to 727$	$-0.106, 0.092$	-0.109	$1-50$	-0.095

Table 3. *Correlation before and after removing outliers from the log- return series.*
The tables describe the period we examine, which shares that were involved (the notation $i \to j$ indicates that share i predict share j the next day), the interval of non-significant correlation values, the original correlation, the number of days we removed to get a non-significant correlation, and finally, what correlation value we ended up with after the days were removed. Some entries are left blank due to the length of the sets being too short.

The highlighted pairs are shares where we needed to remove five days or more to get a correlation value within the limits determined by $\hat{\theta}_{i,j}^{Low}$ and $\hat{\theta}_{i,j}^{High}$, and we will only focus on these in our further analysis.

We notice that all correlations are negative, and will keep this observation in mind when trying to derive appropriate trading strategies. We are especially interested in whether the sign of the correlation is consistent when looking at other aspects of the series e.g. relative to the size of the volatility.

2.5.2 The selected pairs

In the previous section we found that a large number of correlations survived the process of eliminating high and low increments. This does however not ensure any predictability. Some patterns in the correlation values, even if they seem to be predictable at certain times, may appear and disappear randomly, or even change sign.

To determine which shares we want to examine further, we consider the following:

- what pairs of shares kept a significant correlation after days were removed?
- how many days did we need to remove before reaching a non-significant level?
- what pairs had a significant correlation in more than one period after days were removed?

There are no pairs that repeat themselves throughout every period in time, but some kept their significant correlation in two periods, all of them are related to Method 2. These are:

T_1	T_2	$i \rightarrow j$
12	14	$032 \rightarrow 686$, $206 \rightarrow 284$, $587 \rightarrow 431$, $621 \rightarrow 061$
08	10	$174 \rightarrow 597$
10	14	$315 \rightarrow 415$, $738 \rightarrow 311$
08	14	$032 \rightarrow 234$
08	12	$032 \rightarrow 199$
06	12	$686 \rightarrow 727$

Table 4. *Shares with high every-day correlation that repeats throughout time periods.*

We put more weight on consecutive periods and periods that are close to the present time, we therefore only examine the highlighted rows further. To be able to test our resulting strategies we need both the companies in a correlation pair $i \rightarrow j$ to still exist in the testing period; July 2015 to July 2016, and thus have to check this. Out of our 5 pairs, three did not fit these criteria:

$32 \rightarrow 686,\ 206 \rightarrow 284,\ 174 \rightarrow 597.$

We want to take a closer look at the shares that kept a significant correlation after more than 20 days were removed in resent periods, e.g. in period 12 and 14. These are: $045 \rightarrow 507,\ 165 \rightarrow 258, 528 \rightarrow 18, 528 \rightarrow 475, 796 - 135,\ 501 \rightarrow 165, 315 \rightarrow 415, 032 \rightarrow 234, 686 \rightarrow 727.$

But out of these, only 5 pairs exist today:

$045 \rightarrow 507,\ 165 \rightarrow 258,\ 501 \rightarrow 165,\ 315 \rightarrow 415,\ 686 \rightarrow 727.$

All in all, this leaves us with 7 pairs to examine further:

$587 \rightarrow 431,\ \ 621 \rightarrow 061,\ \ 045 \rightarrow 507,\ \ 165 \rightarrow 258,\ \ 501 \rightarrow 165,\ \ 315 \rightarrow 415,$

$\text{and}\ \ 686 \rightarrow 727.$

Rather than removing days that cause high correlation, we now wish to differentiate between the very large / very small movements in the whole return series.

Motivated by the direction function (6) we change the entire returns series in such a way that they only contain 0's, 1's or -1's, representing no change, positive change and negative change in the return series respectively. This means that we do not necessarily care about how much our return series change, only if they change more or less then some pre-determined boundaries. In doing this we seek to reveal if some type of change in the return- series of share i often appear together with some type of shift in the return series of share j the next day.

We will set the limits for whether we generate a $0, 1$ or -1 in three different ways, and analyse them separately.

For method 1 $(k = 1)$ we have that

$$s_i^k(t+1) = \begin{cases} 1, & if\ x_i(t) < x_i(t+1) \\ -1, & if\ x_i(t) > x_i(t+1) \\ 0, & otherwise \end{cases} \tag{14}$$

For method 2 $(k = 2)$ we have that

$$s_i^k(t+1) = \begin{cases} 1, & if\ x_i(t) - x_i(t+1) > \ sd\big(x_i(t)\big) \\ -1, & if\ x_i(t) - x_i(t+1) < \ sd\big(x_i(t)\big) \\ 0, & otherwise \end{cases} \tag{15}$$

For method 3 $(k = 3)$ we have that

$$s_i^k(t+1) = \begin{cases} 1, & if\ x_i(t) - x_i(t+1) > 2\ sd\big(x_i(t)\big) \\ -1, & if\ x_i(t) - x_i(t+1) < 2\ sd\big(x_i(t)\big) \\ 0, & otherwise \end{cases} \tag{16}$$

Where $s_i^k(t)$ is the new series for share i containing 0, 1 or -1, depending on method used; $k = 1, 2 \; or \; 3$. I.e. the volatility of share i.

To do a check for the continuity in size and sign, we take the correlation $C_{i,j}^k$ between $s_i^k(t)$ and $s_j^k(t+1)$ for $k = 1, 2, 3$.

T	$i \rightarrow j$	$C_{i,j}$	$C_{i,j}^1$	$C_{i,j}^2$	$C_{i,j}^3$
14 12	$587 \rightarrow 431$	-0.134 -0.152	-0.000 -0.083	-0.157 -0.051	-0.161 -0.204
14 12	$621 \rightarrow 061$	-0.130 -0.110	-0.079 -0.082	-0.133 -0.057	-0.068 -0.035
14	$045 \rightarrow 507$	-0.092	-0.082	-0.102	-0.126
14	$165 \rightarrow 258$	-0.103	-0.023	-0.008	0.000
14	$315 \rightarrow 415$	-0.111	$+0.002$	-0.025	-0.061
12	$501 \rightarrow 165$	-0.109	-0.061	-0.113	-0.087
12	$686 \rightarrow 727$	-0.169	-0.222	-0.269	-0.215

Table 5. *Correlations after remaking the log-return series*

The table shows what periods we examine, together with the shares involved and their correlation values. The $C_{i,j}$ is the correlation we got using our original data, while the three last columns shows the correlation values using method 1, 2 and 3 respectively.

We eliminate the highlighted pair from our further analysis due to the inconsistency in the signs of the correlations. We want to look for a significant difference between the mean value of the returns of share j following 1's or -1's in the s_i^k- series, and compare it with the mean value of the entire series, $x_j(t)$. That is, we want to test if we would have done better if we had invested in share j every day after the decline of share i, and compare it to the case where we had invested in share j every day regardless of the movements in share i. We do a similar comparison for increases in share i.

To do this we need to create new series for the j shares containing only the increments corresponding to the days after series i had increases (or decreases).

That is

$x_j^{k,1}$ $(t+1)$ contains the values of $x_j(t+1)$ corresponding to when $s_i^k(t) = 1$ and, $x_j^{k,-1}$ $(t+1)$ contains the values of $x_j(t+1)$ corresponding to when $s_i^k(t) = -1$, for $t = 1, \dots, T-1$ and $k = 1,2,3$.

For the mean value and the standard deviation of the new series we use the following notation:

$$\mu_k^1 = E(x_j^{k,(1)}(t))$$
$$sd_k^1 = sd\left(x_j^{k,(1)}(t)\right)$$
$$\mu_k^{-1} = E(x_j^{k,(-1)}(t))$$
$$sd_k^{-1} = sd\left(x_j^{k,(-1)}(t)\right)$$

We summarize the results in Table 6.

T	$i \rightarrow j$	μ_0	sd_0
14	$587 \rightarrow 431$	-0.002201	0.055676
14	$621 \rightarrow 061$	0.000862	0.015256
14	$045 \rightarrow 507$	0.000901	0.016509
14	$165 \rightarrow 258$	-0.000284	0.066115
14	$501 \rightarrow 165$	-0.000127	0.030001
14	$686 \rightarrow 727$	0.000682	0.013550
12	$587 \rightarrow 431$	-0.002811	0.052258
12	$621 \rightarrow 061$	0.001159	0.018701
12	$045 \rightarrow 507$	0.000661	0.012073
12	$165 \rightarrow 258$	0.000330	0.057684
12	$501 \rightarrow 165$	-0.000595	0.020269
12	$686 \rightarrow 727$	0.000960	0.011763

T	$i \rightarrow j$	μ_1^1	sd_1^1	μ_1^{-1}	sd_1^{-1}
14	$587 \rightarrow 431$	-0.007171	0.047140	0.002645	0.063254
14	$621 \rightarrow 061$	-0.001078	0.016611	0.002512	0.013843

14	$045 \rightarrow 507$	0.000354	0.015005	0.001499	0.017711
14	$165 \rightarrow 258$	−0.004885	0.067137	0.004885	0.065500
14	$501 \rightarrow 165$	0.002262	0.029879	−0.002209	0.030731
14	$686 \rightarrow 727$	0.001082	0.013700	0.000323	0.013531
12	$587 \rightarrow 431$	−0.005421	0.049606	−0.000247	0.054884
12	$621 \rightarrow 061$	0.000738	0.019088	0.001493	0.018394
12	$045 \rightarrow 507$	−0.000549	0.011836	0.001770	0.012216
12	$165 \rightarrow 258$	−0.000183	0.058048	0.000352	0.057696
12	$501 \rightarrow 165$	−0.001650	0.019121	0.000191	0.021514
12	$686 \rightarrow 727$	0.000282	0.011502	0.001686	0.012033
T	$i \rightarrow j$	μ_2^1	sd_2^1	μ_2^{-1}	sd_2^{-1}
14	$587 \rightarrow 431$	−0.010548	0.049939	0.018287	0.056616
14	$621 \rightarrow 061$	−0.004684	0.019744	0.002784	0.015220
14	$045 \rightarrow 507$	−0.002397	0.014223	0.002872	0.013629
14	$165 \rightarrow 258$	−0.001307	0.051136	0.010673	0.070128
14	$501 \rightarrow 165$	0.006857	0.031434	−0.001470	0.040234
14	$686 \rightarrow 727$	0.000869	0.016768	0.001333	0.015157
12	$587 \rightarrow 431$	−0.021532	0.049209	0.001167	0.043588
12	$621 \rightarrow 061$	0.000570	0.021527	0.003120	0.022409
12	$045 \rightarrow 507$	−0.000948	0.010055	0.001081	0.011549
12	$165 \rightarrow 258$	0.011600	0.069771	−0.000254	0.056318
12	$501 \rightarrow 165$	−0.001191	0.017970	0.002786	0.020794
12	$686 \rightarrow 727$	0.000516	0.010112	0.003722	0.013770
T	$i \rightarrow j$	μ_3^1	sd_3^1	μ_3^{-1}	sd_3^{-1}
14	$587 \rightarrow 431$	−0.025130	0.046784	0.030405	0.058119
14	$621 \rightarrow 061$	0.003009	0.005783	0.008266	0.016773
14	$045 \rightarrow 507$	−0.005378	0.012616	0.004006	0.015923
14	$165 \rightarrow 258$	−0.002081	0.051502	0.022036	0.054263
14	$501 \rightarrow 165$	−0.002657	0.031895	0.001687	0.043947
14	$686 \rightarrow 727$	0.002060	0.014520	0.001369	0.020922
12	$587 \rightarrow 431$	−0.017630	0.048406	0.052839	0.087029
12	$621 \rightarrow 061$	−0.004568	0.020126	0.000546	0.018281

12	$045 \rightarrow 507$	-0.001039	0.009610	0.001360	0.012180
12	$165 \rightarrow 258$	0.023520	0.067182	-0.008559	0.051821
12	$501 \rightarrow 165$	-0.002280	0.019718	0.001835	0.021846
12	$686 \rightarrow 727$	0.002282	0.008798	0.007311	0.016747

Table 6 *Comparison of means and standard deviation after non-random sampling* The table shows what period we examine together with what share i that predicts share j in that period. The μ_0 and sd_0 describes the mean value and standard deviation of the original log- return series of share j. The μ_k^1, μ_k^{-1}, sd_k^1 and sd_k^{-1} describes the mean values and standard deviations of the same log-return series, only now using solely the data following "ups" or "downs" in the corresponding i-series respectively. The subscript k indicates what method was used in creating the "up" and "down" categories.

We can see that the means differ, but we need a method to determine whether the differences are significant or not. In the next section we will build tests for estimating the significance of these deviances, and use these tests to determine the structure of our trading strategies.

2.5.3 Finding the trading strategies

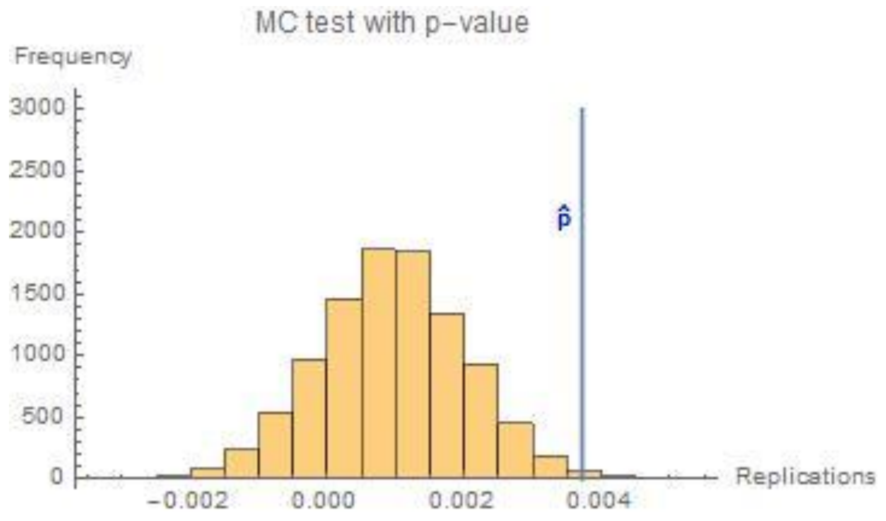
From the previous section, we have the mean values corresponding to no systematic method of partitioning compared with the mean values after partitioning according to the three methods, $k = 1, 2,3$.

We assume that since the increments of the log-return series $x_1, \dots, x_n$ are independent and random, there should be no difference in the mean value between a sample drawn on the basis of buy or sell signals, and samples of the same size drawn randomly.

We want to test this, i.e. if the mean values μ_k^s for $s = 1, -1$, differ significantly from the mean values we would get by drawing random samples from x_j, each of the same sizes as $x_j^{k,1}$ and $x_j^{k,-1}$.

That is, can we expect higher returns by following a strategy that tells us to buy or sell share j after we encounter specific movements in the i- series, than we would have gotten if we placed an equal amount of buy's and sell's randomly?

To answer this, we simulate $B = 10\,000$ independent replications of the sample mean, $(\mu_k^s)^{*1:B}$, corresponding to the random strategies and compare the results with what we got following our derived strategies. The $(\mu_k^s)^{*1:B}$ are found by sampling from $\hat{F}$ under H_0, i.e. drawing B random samples from our original log-return series, and calculating the sample mean for each of them. We illustrate two of the resulting distribution estimates in **Figure 6**, together with the corresponding sample mean of our original sample.



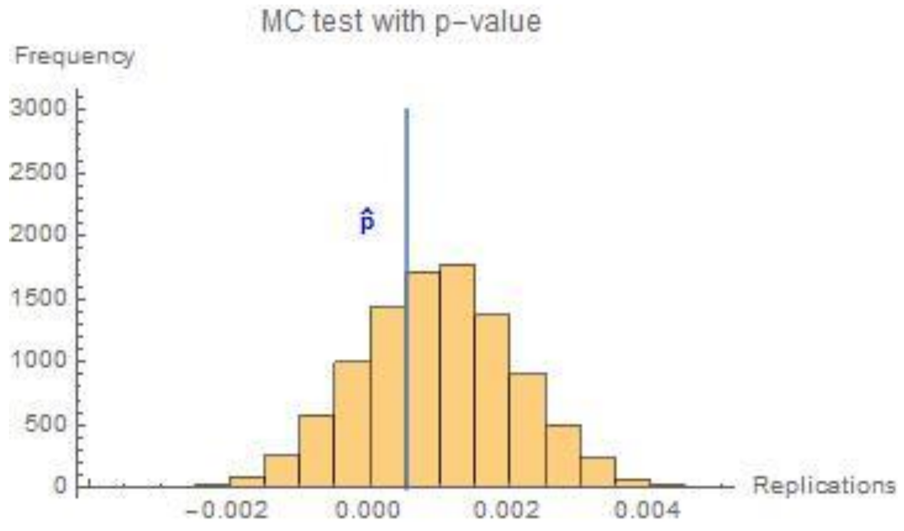


Figure 6. *MC tests for* $686 \rightarrow 727$.

Shows sample means obtained using 10 000 random strategies on shares $686 \rightarrow 727$ *together with the mean we got using our strategy on the same shares (blue vertical line). In both examples we used method 2 (*$k = 2$*). In the top figure we are looking at* μ_k^s *when* $s = -1$ *and in the bottom when* $s = 1$.

For every pair of shares we need 6 tests: μ_k^s for $s = 1, -1$ and for each using the three methods $k = 1,2,3$.

We are interested in both very high, and very low allocations of our test statistic μ_k^s. A high location indicates negative correlation while a low indicates the opposite. It is important to differentiate between the two if we are to build our strategy accordingly. We use a one sided test for the p-value and test the upper limit if the correlation was negative, and the lower limit if it was positive. The p-value tells us that, if the null hypothesis is true and there is no difference between our test statistic and the simulated replicas, what is the probability that we would observe a more extreme test statistic in the direction of the alternative hypothesis than we did. That is

$$P = P(\mu_k^s \geq (\mu_k^s)^{*b}\ ), \qquad C_{i,j} < 0$$
$$P = P(\mu_k^s \leq (\mu_k^s)^{*b}\ ), \qquad C_{i,j} > 0$$

We estimate this probability by counting the number of simulated replicas that is allocated above (or below) our original statistic.

$$\hat{P}^n{}_{MC} = \frac{1 + \{\#\ (\mu_k^s)^{*b} \geq \mu_k^s\}}{1 + B}, \qquad C_{i,j} < 0$$
$$\hat{P}^p{}_{MC} = 1 - \hat{P}^n{}_{MC}, \qquad C_{i,j} < 0$$

Where $\#\ (\mu_k^s)^{*b}$ is the number of simulated replicas with values more extreme than our original estimate.

The following table summarizes the results from the testing procedure.

		μ_k^s	$p - values$	$p - values$	$p - values$
T	$i(t) \rightarrow j(t+1)$	s	$k = 1$	$k = 2$	$k = 3$
14	$587 \rightarrow 431$	1 −1	0.111 0.129	0.129 0.009	0.024 0.010
14	$621 \rightarrow 061$	1 −1	0.071 0.092	0.020 0.228	0.326 0.056
14	$045 \rightarrow 507$	1 −1	0.332 0.315	0.040 0.172	0.035 0.226
14	$165 \rightarrow 258$	1 −1	0.181 0.181	0.459 0.134	0.470 0.091
14	$501 \rightarrow 165$	1 −1	0.144 0.179	0.029 0.306	0.328 0.368
14	$686 \rightarrow 727$	1 −1	0.347 0.348	0.457 0.344	0.296 0.404
12	$587 \rightarrow 431$	1 −1	0.221 0.227	0.037 0.319	0.275 0.030
12	$621 \rightarrow 061$	1 −1	0.367 0.391	0.371 0.176	0.071 0.439

12	$501 \rightarrow 165$	1 -1	0.213 0.282	0.379 0.038	0.291 0.211
12	$045 \rightarrow 507$	1 -1	0.058 0.080	0.115 0.372	0.241 0.359
12	$165 \rightarrow 258$	1 -1	0.453 0.484	0.044 0.478	0.014 0.194
12	$686 \rightarrow 727$	1 -1	0.191 0.167	0.345 0.005	0.244 0.001

Table 7. *P-values*

The table describes what period we do our testing in and what pairs of shares are involved. The third column indicates whether we are testing μ_k^s for $s = 1$ or -1. The three last columns contain p-values corresponding to $k = 1, 2$ and 3 respectively.

We can see that on a there are some strong indications of significance on a 5% level. To find a suitable strategy to use, we need both low p-values and a consistent pattern in the sign of the correlations. **Table 8** will depict a more elaborate summery of the result showed in **Table 7**, and we will use these to determine what strategy to use in our experiments.

The shares that we analyse have price series that span further back in time than what we have looked at this far. We will nevertheless, not look at the entire lifespan of the stocks but rather on the last couple of two-year time periods $T = 12, 14$.

In the following table, the p-values are subscripted with an *n* if the correlation was negative and a *p* if it was positive. We are interested in whether we can recognise any predictable pattern that we are able to take advantage of.

$\boxed{587 \to 431}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$-.00220$	1	$-.00717$	0.111_n	$-.01055$	0.129_n	$-.02513$	0.024_n
		-1	$.00265$	0.129_n	$.01829$	0.009_n	$.03041$	0.010_n
12	$-.00281$	1	$-.00542$	0.221_n	$-.02153$	0.037_n	$-.01763$	0.275_n
		-1	$-.00025$	0.227_n	$.00117$	0.319_n	$.05284$	0.030_n
				$\underline{0.172}$		$\underline{0.124}$		$\underline{0.085}$

$\boxed{621 \to 061}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$.00086$	1	$-.00108$	0.071_n	$-.00468$	0.020_n	$.00301$	$\mathbf{0.326_p}$
		-1	$.00251$	0.092_n	$.00278$	0.228_n	$.00827$	0.056_n
12	$.00116$	1	$.00074$	0.367_n	$.00057$	0.371_n	$-.00457$	0.071_n
		-1	$.00149$	0.391_n	$.00312$	0.176_n	$.00055$	$\mathbf{0.439_p}$
				$\underline{0.230}$		$\underline{0.199}$		

$\boxed{045 \to 507}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$.00090$	1	$.00035$	0.332_n	$-.00240$	0.040_n	$-.00538$	0.035_n
		-1	$.00150$	0.315_n	$.00287$	0.172_n	$.00401$	0.226_n
12	$.00066$	1	$-.00055$	0.058_n	$-.00095$	0.115_n	$-.00104$	0.241_n
		-1	$.00177$	0.080_n	$.00108$	0.372_n	$.00136$	0.359_n
				$\underline{0.172}$		$\underline{0.124}$		$\underline{0.085}$

$\boxed{165 \to 258}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$-.00028$	1	$-.00489$	0.181_n	$-.00131$	0.459_n	$-.00208$	0.470_n
		-1	$.00489$	0.181_n	$.01067$	0.134_n	$.02204$	0.091_n
12	$.00033$	1	$-.00018$	0.453_n	$.01160$	$\mathbf{0.044_p}$	$.02352$	$\mathbf{0.014_p}$
		-1	$.00035$	0.484_n	$-.00025$	$\mathbf{0.478_p}$	$-.00856$	$\mathbf{0.194_p}$
				$\underline{0.325}$				

$\boxed{501 \to 165}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$-.00013$	1	$.00226$	$\mathbf{0.144_p}$	$.00686$	$\mathbf{0.029_p}$	$-.00266$	0.328_n
		-1	$-.00221$	$\mathbf{0.179_p}$	$-.00147$	$\mathbf{0.306_p}$	$.00169$	0.368_n

12	$-.00060$	1 -1	$-.00165$ $.00019$	0.213_n $\mathbf{0.282_p}$	$-.00119$ $.00279$	0.379_n 0.038_n	$-.00228$ $.00184$	0.291_n 0.211_n
								$\underline{0.300}$

$\boxed{686 \rightarrow 727}$

T	$\mu_0 \approx$	$r_j^{(*)}$	$\mu_1 \approx$	p_1	$\mu_2 \approx$	p_2	$\mu_3 \approx$	p_3
14	$.00068$	1 -1	$.00108$ $.00032$	$\mathbf{0.347_p}$ $\mathbf{0.348_p}$	$.00087$ $.00133$	$\mathbf{0.457_p}$ 0.344_n	$.00206$ $.00137$	$\mathbf{0.296_p}$ 0.404_n
12	$.00096$	1 -1	$.00028$ $.00169$	0.191_n 0.167_n	$.00052$ $.00372$	0.345_n 0.005_n	$.00228$ $.00731$	$\mathbf{0.244_p}$ 0.001_n

***Table 8** P-values Extended. The table headings describe the pairs of shares that are tested. The first column tells us what time period we examine, while the third column indicates whether we are testing μ_k^s for $s = 1$ or -1. The rest of the columns shows the mean values and their corresponding p-values, associated with method $k = 1, 2$ and 3 respectively. Under the tables you find the mean value of the p-values listed above.*

When evaluating the results displayed in **Table 8**, we first want to eliminate the trading strategies that would include inconsistent trading patterns, these are:
Shares $621 \rightarrow 061$ when $k = 3$, shares $165 \rightarrow 258$ when $k = 2$ and 3, shares $501 \rightarrow 165$ when $k = 1$ and 2, and lastly we eliminated shares $686 \rightarrow 727$ entirely.

The next step is to assess the magnitude of the p-values. We do a simple study of the mean values of the p-values corresponding to each of the categories. The results are shown underneath the listed p-values in **Table 8**. The highlighted columns depict the strategies we will use on the different pairs of stocks. We summarize the results in the following table.

$587 \rightarrow 431$	$k = 3$
$621 \rightarrow 061$	$k = 2$
$045 \rightarrow 507$	$k = 2$
$165 \rightarrow 258$	$k = 1$
$501 \rightarrow 165$	$k = 3$

***Table 9** Strategies The pairs of shares that are tested together with the best strategy k.*

In the next section we will test these strategies on previously un-examined data from the period July 1st 2015 to July 1st 2016 and see if we can use them to earn abnormal returns.

2.6 The trading strategies

In this section we test the predicting strengths associated with our newly developed trading strategies. We find that not only do the methods result in generating abnormal returns, but they also lead to considerable less risk compared to that of an average random investment strategy.

In all our strategies we had negative correlation between our shares, that means that we will buy share j when share i declines and vice versa. Thus, the differences between the strategies we use on the shares depend only on how we set the limit for generating a "buy" or "sell" signal i.e. if $k = 1{,}2$ or 3. (**Table 9**)

We conduct our experiment on an unexamined time period, namely the year: July 1st 2015 to July 1st 2016.

Starting with an arbitrary capital of 100 000 NOK and no shares, we wait until we get our first "buy" signal from share i. The signal is generated by monitoring the movements of share i, specifically if the movements are greater than the limits set by k in equations (14) (15) and (16). If they are, this indicates a prediction that share j will increase in value the next day and thus it generates a "buy" signal. When the signal comes, we invest all our money in share j, wanting to buy the day before the price increases so that we hold the share while it increases in value. We keep our money invested until we reach a "sell" signal. The "sell" signal is crated in the same manner as a "buy", but indicates a prediction of share j decreasing in value. When we receive this signal we sell all the shares we hold at that moment in time. We keep following this strategy until we reach the end of the test-period.

When we conduct the experiment in this manner we miss the opportunity of acting on some signals, that is, either the times we encounter "sell" but don't have any shares to sell, or the times we encounter "buy" but don't have any money. We consider this unproblematic since it does not contradict our intentions, namely not to hold stocks we believe will decrease in value, and hold stocks we believe will increase. Furthermore, a strategy involving lending money and shorting stocks would prove unnecessary complicated for our purpose.

At the end of our investment period we counted the number of days our money was invested and calculated the daily rate of return for each of the five shares in our experiment.

The ultimate question to answer was; did we beat the market using our strategies?

First, it depends on how we interpret the phrase “beating the market”, secondly it depends on the two factors: are our returns significantly better using an information-based strategy, then what we could expect to achieve without using the information, and is the level of risk the same?

To answer the question regarding the abnormal returns, we consider what would have happened if we were to repeat the same experiment 10 000 times only with random generations of an equal amount of “buy” and “sell” signals, each time estimating the daily returns from that specific experiment. From these estimates we create a test to determine the level of significance of our result, using the null- hypothesis that the returns are equal using a strategy with information or without. We repeat the process for every share in our experiment (**Table 9**). The null-hypothesis is

$$H_0 : r_{info} = r_{rand}, \tag{17}$$

where r_{info} is the estimated return using an information-strategy, and r_{rand} is estimated returns with random strategies.

The algorithm is as follows: generate a sample $\underline{x}^{*b}$ by drawing from the original log-return series $x_j = x_1, \dots, x_n$ using the same procedure as in our experiment only now using random buy and sell signals. We do this $b = 1, \dots, 10\,000$ times, and for each we calculate the daily returns r^{*b} from the sample $\underline{x}^{*b}$. We calculate the p-value by finding the amount of r^{*b} greater than the daily returns we got from our information based experiment, i.e.

$$p_b = \frac{1 + \#\{r^{*b} \geq r_{info}\}}{1 + 10\,000}$$

Where p_b is the estimated probability that we would observe a more extreme test statistic in the direction of the alternative hypothesis than our original r_{info}.

The p-value describes how well we did, trading at one share using our strategy compared to results using random sampling, but we also want a measure on how well we did in total, using all five pairs of stocks

If the null hypothesis (17) is true, the p-values from the five experiments (**Table 10**), $p_1, \dots, p_5$ should be uniformly distributed between 0 and 1.

$$p_b \sim U[0,1], \qquad b = 1, \dots, 5$$

We can use this to test the probability of our results being drawn from a uniform distribution by assuming that $p_1, \dots, p_5$ are independent identically distributed uniform variables. We compare our test statistic:

$$\overline{p}_b = \frac{1}{5}\sum_{b=1}^{5} p_b = 0.304$$

with the mean of 5 random draws from the uniform distribution generated 10 000 times. This results in the estimates $\overline{p}^{*1}, \overline{p}^{*2}, \dots, \overline{p}^{*10\,000}$which computes the p-value:

$$\hat{p}_{\overline{p}_b} = \frac{1 + \{\#\overline{p}^* \leq 0.304\}}{1} = 0.069 \tag{18}$$

We get a significant result, but we do not know whether these excess returns come with greater risk or not, i.e. do we have a risk-return trade-off? To find out, we test the standard deviations of our return series against bootstrapped replications, using the same procedure as we did when testing the mean. The results are shown in the last column of **Table 10**.

We repeat the p-value estimation (18), only this time using the mean values of the p-values we got when testing the standard deviations,

$$\overline{p}_{SD} = \frac{1}{5}(0.635 + 0.023 + 0.033 + 0.005 + 0.002) = 0.140,$$

and get that

$$\hat{p}_{\overline{p}_{SD}} = \frac{1 + \{\#\overline{p}^* \leq 0.140\}}{1} = 0.0017.$$

3 Main results

When testing if our returns generated by an information based trading strategy exceeded returns generated by random trading strategies, we found that they mostly did. In one of the five experiments done, we did less well than random strategies, while in the four others we did better. When testing the results as a group, we got a p-value of 0.069. This strongly indicates that our returns from the group as whole exceed what is expected by the null hypothesis (17), and we conclude that we have done significantly better using strategies based on information, then we would have done using random strategies.

The result regarding the risk involved, gives an even lower p-value when calculated in the same manner as we did with the returns. The likelihood of achieving our aggregated level of risk under the assumption of there being no difference is a staggering 0.0017. This contradicts the assumption that higher reward comes with higher risk, at least when risk is measured in the terms of a stocks standard deviation.

The following table summarize the results.

$i \to j$	k	$p - value\ for\ r_{info}$	$p - value\ for\ \widehat{sd}(r)$
$587 \to 431$	3	0.16	0.635
$621 \to 061$	2	0.24	0.023
$045 \to 507$	2	0.37	0.033
$165 \to 258$	1	0.69	0.005
$501 \to 165$	3	0.06	0.002
		$\hat{p}_{\overline{p}_b} = 0.069$	$\hat{p}_{\overline{p}_{SD}} = 0.0017$

***Table 10** Main results*
The pairs of shares that are tested together with the strategy performed k. The third column describes the p-values for the returns tested, while the last column describes the p-values associated with the amount of risk taken. The highlighted row is the p-value associated with the whole group of shares.

4 Discussion

When starting this analysis, our initial assumption was that there would be few significant correlations between the log- return series of the shares, this due to the fact that the EMH predicts stock prices to be random and uncorrelated with one another. This assumption was not supported by our findings. Even with setting a strict significance boundary using the Bonferroni correction, we found a large number of significant cross-correlations, and even more auto-correlations. The question became: could we find a pattern within these correlations that could be used to predict future movements in stock prices? We analysed the cross-correlations only, and left the auto-correlations unexamined for this time.

One interesting question was whether pairs of stocks that we found to be significantly correlated in some years, would keep this correlation other years, and if they did, could we take advantage of this? We reassembled the time series into periods that contained prices over two-year periods only, and analysed the periods separately. For each time period, we created between 50 000 and 100 000 tests using Monte Carlo simulation techniques, and even with using the fastest method we could think of, the calculation time was still an impressive 70 -100 hours per test set. What we found was that a large number of the significant correlations were repeatable throughout the different time-periods. Nevertheless, we eliminated many of them from our further analysis. This was due to the fact that for these pairs of shares, the correlation seemed to come from high correlation on a particular few days, rather than a consistent correlation over the whole set of days, thus making it difficult to make good predictions on account of them.

After eliminating shares from further investigation, we turned the remaining log-price series of the predictor shares i into series containing $1's, -1's$ and $0's$, representing positive, negative or no change from one day to the next in the log-price series, respectively. The purpose was to categorize specific types of movements in the i- series, and see if any of them corresponded consistently with certain movements in the j- series. We created limits for whether we generated a "buy" or a "sell" signal in three different ways, and tested whether there was any difference between the returns of share j following $1's$ or $-1's$ in the i series, and the same number of returns originating from drawing randomly from share j. We found such differences and thus assigned p-values to each case, in order to explain the degree of displacement.

On the basis of the p-values we found, we determined investment-strategies to test on an unexamined data set.

To limit the interference of personal beliefs in the course of our analysis, we numbered the stocks 1 to 811 rather than using their true names.

The results found were that we did beat random strategies of the same structure, both in terms of generating higher returns and in terms of the risk involved. However, whether these results contradict the EMH or not, is a non-trivial question due to the different ways to interpret the hypothesis.

If we define the statement "beating the market" to mean beating the market as a whole, our finding does neither support nor contradict this. However, we did not set out to find shares that, in general, did better than the market, but rather to find strategies detailing how to invest in one particular stock, that outperformed other trading strategies on that same stock. Even though it would have been interesting to know, our analysis did not differentiate between what type of share we were looking at or the risk level associated with it, e.g. the sector involved, the trading volume, whether they were "value companies" etc. Future research might profit from investigating these issues further

If one, however, interprets the meaning of the EMH to be that one cannot use publicly available information to earn abnormal, and risk adjusted returns, then we may have a contradiction. If the return of a stock can be defined as abnormal when it exceeds the expected return of that particular stock, we have a contradiction. If not, and the return has to be "abnormal" relative to the return of the market to fit the criteria of the EMH, then we can only conclude that we have found good trading strategies for these particular stocks.

When it comes to the risk involved, we run into the same problem. We could have chosen to compare the risk of a trading strategy on one particular stock, against the market risk, but we only focused on the risk relative to trading on other days on that specific stock. The volatility of the stocks may reflect the market, or it may not. Our results only show that the risk is considerably less compared to the expected risk of the stock itself.

Even though we found patterns to take advantage of, there might have been many more patterns that we missed, or patterns that was too intricate to use. One reason for this may be that we had such a large amount of data that we systematically searched, that spurious relationship patterns (Yule, 1926) appeared due to chance alone. Another reason we were not able to take advantage of patterns, may have been because they appear and disappear in a seemingly random fashion. There may be an underlying mechanism causing two shares to correlate in some periods, but then this relation cease to exist for some unknown reason (Sugihara, et al., 2012). Patterns can also, as the EMH predicts, disappear because someone published information about the existence of their presence.

References

Black, F. (1976). “Studies of stock price volatility changes”. *Proceedings of the 1976 meetings of the American Statistical Association*. Business and Economics Statistics Section (American Statistical Association, Washington, DC).

Black, F., & Scholes, M. (1973). “The Pricing of Options and Corporate Liabilities,”. *Journal of Political Economy, 81*.

Bodie, Z., Kane, A., & Marcus, A. (2007). *Essentials of investments, 6th Edition.* McGraw-Hill.

Bonferroni, C. (1936). "Teoria statistica delle classi e calcolo delle probabilità". *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*.

Brealey, R., Myers, S., & Marcus, A. (1999). *Fundamentals of Corporate Finance, 2nd Edition.* McGraw-Hill.

Brock, W., Lakonishok, J., & LeBaron, B. (1992). "Simple Technical Trading Rules and the Stochastic Properties of Stock Return". *Journal of Finance, 47*.

Campbell, J., & Hentschel, L. (1992). “No News is Good News: An Asymmetric Model of Changing Volatility in Stock Returns”. *Journal of Financial Economics, 31*.

Campbell, J., Lo, A., & MacKinlay, C. (1997). *"The Econometrics of Financial Markets"*. Princeton University Press.

Cantelli, F. (1933). "Sulla determinazione empirica delle leggi di probabilita" Giorn. Ist. Ital. Attuari 4. *Giorn. Ist. Ital. Attuari 4,221-424*.

Christie, A. (1982). “The Stochastic Behavior of Common Stock Variances”. *Journal of Financial Economics, 10*.

Daouk, H., & Ng, D. (2007). “Is Unlevered Firm Volatility Asymmetric?”. *Working paper*.

Efron, B., & Tibshirani, R. (1993). "An Introduction to the Bootstrap". *Boca Raton, FL: Chapman & Hall/CRC*.

Fama, E. (1965 a). "Random Walks in Stock Market Prices". *Selected Papers*.

Fama, E. (1965 b). "The behavior of stock-market prices". *The Journal of Business,* pp. 34-105.

Fama, E. (1970). "Efficient capital markets: A review of theory and empirical work". *Journal of Finance,* pp. 383-417.

French, K., Schwert, W., & Stambaugh, R. (1987). “Expected Stock Returns and Volatility”. *Journal of Financial Economics, 19.*

Glivenko, V. (1933). "Sulla determinazione empirica della legge di probabilita". *Giorn. Ist. Ital. Attuari 4, 92-99.*

Glosten, L., Jagannathan, R., & Runke, D. (1993). ”On the Relation Between Expected Value and the Volatility of the Nominal Excess Return on Stocks”. *Journal of Finance, 47.*

Jensen, M. (1978). "Some anomalous evidence regarding market efficiency". *Journal of Financial Economics*, p. 98.

Jiang, X., & Lee, B. (2004). “On the Dynamic Relation Between Returns and Idiosyncratic Volatility”. *Working Paper.*

Kendall, M. (1953). "The analysis of economic time series, part I: Prices". *Journal of the Royal Statistical Society.*

Korneliussen, F., & Rasmussen, C. (2014). ”Systematic Risk Factors at Oslo Stock Exchange". *Master’s thesis.* Oslo and Akershus University College of Applied Sciences.

Lintner, J. (1965). "The valuation of risk assets on the selection of risky investments in stock portfolios and capital budgets.". *Review of Economics and Statistics, 47.*

Markowitz, H. (1952). "Portfolio Selection". *The Journal of Finance.*

Mitchell, W. C. (1915). "The making and using of index numers". *Bulletin of the United States Bureau of Labor Statistics 173.*

Mossin, J. (1966). "Equilibrium in a Capital Asset Market". *The Econometric Society.*

Nygaard, K. (2011). ”The Disposition Effect and Momentum: Evidence from Norwegian Household Investors.”. *Part of PhD dissertation, Norwegian School of Economics and Business Administration.*

Næs, R., Skjeltorp, J., & Ødegaard, B. (2009). "What factors affect the Oslo Stock Exchange?". *Working Paper 2009/24, Norges Bank.*

oslobors.no. (2016, November 5). Retrieved from 05.11.16.https://www.oslobors.no/ob_eng/Oslo-Boers/Listing/Shares-equity-certificates-and-rights-to-shares/Oslo-Boers-and-Oslo-Axess

Roberts, H. (1959). "Stock market ‘Patterns’ and financial analysis: Methodological suggestions". *Journal of Finance*, pp. 1-10.

Sharpe, W. (1964). "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk". *Blackwell Publishing for the American Finance Association*.

Steeley, J. (2001). "A note on information seasonality and the disappearance of the weekend effect in the UK stock market". *Journal of Banking and Finance, Volume 25*.

Sugihara, G., May, R., Ye, H., Hsieh, C., Deyle, E., Fogarty, M., & Munch, S. (2012). "Detecting Causality in Complex Ecosystems". *Science Vol. 338*.

TD, A. (2009, March 2). "Growth vs. Value: Two Approaches to Stock Investing". *TDAmeritrade. Archived from the original*.

Yavrumyan, E. (2015). "Efficient Market Hypothesis and Calendar Effects: Evidence from the Oslo Stock Exchange". *Reprosentralen, Universitetet i Oslo*.

Yule, G. (1926). "Why do we sometimes get nonsense correlations between time series? A study in sampling and the nature of time series". *Journal of the Royal Statistical Society 89*, pp. 1-64.

APPENDIX

The data.

```
(*Read in the data from TITLON*)
date1 = Map[StringSplit[#] &,
    Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_1.txt", String], 1]][[All, 1]];
codes1 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_1.txt", String], 1]][[
   All, 2]];
names1 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_1.txt", String], 1]][[
   All, 3]];
AdjPrice1 = ToExpression[Map[StringSplit[#] &,
     Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_1.txt", String], 1]][[All, 4]]];
date2 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_2.txt", String], 1]][[
   All, 1]];
codes2 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_2.txt", String], 1]][[
   All, 2]];
names2 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_2.txt", String], 1]][[
   All, 3]];
AdjPrice2 = ToExpression[Map[StringSplit[#] &,
     Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_2.txt", String], 1]][[All, 4]]];
date3 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_3.txt", String], 1]][[
   All, 1]];
codes3 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_3.txt", String], 1]][[
   All, 2]];
names3 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_3.txt", String], 1]][[
   All, 3]];
AdjPrice3 = ToExpression[Map[StringSplit[#] &,
     Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_3.txt", String], 1]][[All, 4]]];
date4 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_4.txt", String], 1]][[
   All, 1]];
codes4 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_4.txt", String], 1]][[
   All, 2]];
names4 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_4.txt", String], 1]][[
   All, 3]];
AdjPrice4 = ToExpression[Map[StringSplit[#] &,
     Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\1_4.txt", String], 1]][[All, 4]]];
(*join them*)
date = Join[date1, date2, date3, date4];
codes = Join[codes1, codes2, codes3, codes4];
names = Join[names1, names2, names3, names4];
AdjPrice = Join[AdjPrice1, AdjPrice2, AdjPrice3, AdjPrice4];

lengder = Map[Length[#] &, Split[codes]] ;
 (*del opp de som har identiske elementer, ta så lengden av sublisten*);
Length[lengder]; (*811*)
L = FoldList[Plus, 0, lengder];
t = 1;
(*assemble into table take logarithm*)
pristab = Table[
   AdjPrice[[L[[t]] + 1 ;; L[[t + 1]]]], {t, 1, Length[L] - 1}];
pristab = ToExpression[pristab];
datotab = Table[
   date[[L[[t]] + 1 ;; L[[t + 1]]]], {t, 1, Length[L] - 1}];

logpristab = N[Log[pristab]];

ListPlot[logpristab[[74]], Joined → True, AspectRatio → 1 / 5, ImageSize → 1000];
```

Removing the trend.

```
mu = Table[0, {i, 1, 811}];
notrendprice = Table[0, {i, 1, 811}];
Do[
 notrendprice[[i]] = logpristab[[i]];
 , {i, 1, 811}]
Do[
  mu[[i]] = Normal[LinearModelFit[logpristab[[i]], x, x]] (*find trend*)
  , {i, 1, 811}];
Do[
  notrendprice[[i]] = logpristab[[i]] - Table[mu[[i]], {x, 1, Length[logpristab[[i]]]}]
  , {i, 1, 811}];
notrendprice = ToExpression[notrendprice];


ListPlot[notrendprice[[74]], Joined → True, AspectRatio → 1 / 5, ImageSize → 1000];
```

Behavior of the return series.

```
(*Examine sd of return series*)

n = 811;

distSD = {};

Do[
 utvalgi = logpristab[[i]]; (*extract the values in these position for stock i*)
 If[Length[utvalgi] > 30, (*if the set is greater than some number, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  sd = StandardDeviation[returnsi];
  distSD = Append[distSD, sd]]
 , {i, 1, n}]

meanSD = Mean[distSD];

sdplot = ContourPlot[x == meanSD, {x, -1, 1}, {y, 0, 160}];

histSD = Histogram[distSD, AxesLabel → "Frequency", PlotLabel → "Histogram of standard deviations (SD)",
   Epilog → {Text[Style[OverHat["σ"], Blue, Bold], {0.070, 120}], Text["SD", {0.16, 8}]}];

Show[histSD, sdplot]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\figur1.jpg", %117, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\figur1.jpg"

distM = {};

Do[
 utvalgi = logpristab[[i]]; (*extract the values in these position for stock i*)
 If[Length[utvalgi] > 30, (*if the set is greater than 30, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  m = Mean[returnsi];
  distM = Append[distM, m]]
 , {i, 1, n}]


histMU = Histogram[distM, AxesLabel → "Frequency", PlotLabel → "Histogram of mean values",
   Epilog → {Text[Style[OverHat["μ"], Blue, Bold], {-0.0005, 150}], Text["Mean", {0.0045, 10}]}];

meanM = Mean[distM];

mplot = ContourPlot[x == meanM, {x, -1, 1}, {y, 0, 180}];

Show[histMU, mplot]
```

```
Export["C:\\Users\\marte\\Dropbox\\Thesis\\Figure2.jpg", %192, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\Figure2.jpg"

utvalgi = logpristab[[2]];(*extract the values in these position for stock i*)
returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];

QuantilePlot[returnsi] (*look at one series at the time*);

Show[%197, FrameLabel → {{HoldForm[Empirical quantiles], None}, {HoldForm[Theoretical quantiles], None}},
 PlotLabel → HoldForm[Normal Q - Q for share nr.2], LabelStyle → {GrayLevel[0]}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\QQ1.jpg", %204, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\QQ1.jpg"

utvalgi = logpristab[[8]];(*extract the values in these position for stock i*)
returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];

QuantilePlot[returnsi] (*look at one series at the time*);

Show[%201, FrameLabel → {{HoldForm[Empirical quantiles], None}, {HoldForm[Theoretical quantiles], None}},
 PlotLabel → HoldForm[Normal Q - Q for share nr.8], LabelStyle → {GrayLevel[0]}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\QQ3.jpg", %202, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\QQ3.jpg"

utvalgi = logpristab[[3]];(*extract the values in these position for stock i*)
returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];

QuantilePlot[returnsi] (*look at one series at the time*);

Show[%209, FrameLabel → {{HoldForm[Empirical quantiles], None}, {HoldForm[Theoretical quantiles], None}},
 PlotLabel → HoldForm[Normal Q - Q for share nr.3], LabelStyle → {GrayLevel[0]}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\QQ2.jpg", %210, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\QQ2.jpg"
```

Removing trend and examine the nature of the correlation matrices.

```
(*compare correlation matrices before (matrix2) and after(matrix3) removing trend*)

n = 811;
matrix2 = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab[[i]], datotab[[j]]]; (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 70,
    (*if the set is greater than some number, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix2[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix2.txt", matrix2]


matrix3 = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab[[i]], datotab[[j]]]; (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[notrendprice[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[notrendprice[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 70,
    (*if the set is greater than some number, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[returnsi] > 0 && StandardDeviation[returnsj] > 0, (*if the returns series
      are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix3[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix3.txt", matrix3]


"C:\\Users\\marte\\Dropbox\\Thesis\\matrix3.txt"

matrix2 = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix2.txt"];

matrix3 = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix3.txt"];

Total[Total[matrix2]];
```

```
Total[Total[matrix3]];

(*lag matrix that describe correlation*)

(*separate positive og negative values*)
Negative2 = Table["empty", {n}, {n}];
Do[
  If[matrix2[[i, j]] <= 0, Negative2[[i, j]] = matrix2[[i, j]]]
  , {i, 1, n}, {j, 1, n}];
Positive2 = Table["empty", {n}, {n}];
Do[
  If[matrix2[[i, j]] > 0, Positive2[[i, j]] = matrix2[[i, j]]]
  , {i, 1, n}, {j, 1, n}];
Negative3 = Table["empty", {n}, {n}];
Do[
  If[matrix3[[i, j]] <= 0, Negative3[[i, j]] = matrix3[[i, j]]]
  , {i, 1, n}, {j, 1, n}];
Positive3 = Table["empty", {n}, {n}];
Do[
  If[matrix3[[i, j]] > 0, Positive3[[i, j]] = matrix3[[i, j]]]
  , {i, 1, n}, {j, 1, n}];

(*matrise beskriver type endringer i korrelasjonsmatrisene*)
KindOfDiff = Table["empty", {n}, {n}];

Do[
  If[Negative3[[i, j]] ≤ Negative2[[i, j]], KindOfDiff[[i, j]] = 1, KindOfDiff[[i, j]] = 0]
   (*if they are more(!) negative now, return 1*)
   If[Positive3[[i, j]] > Positive2[[i, j]], KindOfDiff[[i, j]] = 2, KindOfDiff[[i, j]] = 0]
   (*if they are more(!) positive now, return 2*)
   If[matrix2[[i, j]] > 0 && matrix3[[i, j]] ≤ 0, KindOfDiff[[i, j]] = 3] (*if they change sign, return 3*)
   If[matrix3[[i, j]] > 0 && matrix2[[i, j]] ≤ 0, KindOfDiff[[i, j]] = 3] (*if they change sign, return 3*)
  , {i, 1, n}, {j, 1, n}];

KindOfDiff[[1 ;; 15, 1 ;; 15]] // MatrixForm

Length[Position[KindOfDiff, 1]];

Length[Position[KindOfDiff, 0]];
Length[Position[KindOfDiff, "empty"]];

c = matrix2 - matrix3;

Total[Total[c]]

Dimensions[Flatten[c]]

p1 = Flatten[c];

Histogram[p1[[1 ;; 3000]]];

ListPlot[p1[[1 ;; 500]], Joined → True, AspectRatio → 1 / 5, ImageSize → 1000];

(*We look at the original matrix*)

p2 = Flatten[matrix2];
bort = Position[p2, _ ? (# == "empty" &)];
pp2 = Delete[p2, bort];

mean2 = Mean[pp2];

sd2 = StandardDeviation[pp2];

hist2 = Histogram[pp2, Epilog → {Text[Style[OverHat["θ"], Blue, Bold], {0.01, 50 000}],
     Text[Style[OverHat["θ"] - OverHat["s"], Purple, Bold], {-0.060, 30 000}],
     Text[Style[OverHat["θ"] + OverHat["s"], Purple, Bold], {0.105, 30 000}]}];

pl2 = ContourPlot[x == mean2, {x, -0.5, 0.5}, {y, 0, 57 000}];
```

```
plsd2pos = ContourPlot[x == mean2 + sd2, {x, -0.5, 0.5}, {y, 0, 30000}, ColorFunction → "Rainbow"];
plsd2neg = ContourPlot[x == mean2 - sd2, {x, -0.5, 0.5}, {y, 0, 30000}, ColorFunction → "Rainbow"];

Show[hist2, pl2, plsd2pos, plsd2neg];

Show[%306, AxesLabel → {HoldForm[Correlations], HoldForm[Frequency]},
  PlotLabel → HoldForm[Histogram of correlations], LabelStyle → {9, GrayLevel[0]}];

Export["C:\\Users\\marte\\Dropbox\\Thesis\\Figure3.jpg", %307, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\Figure3.jpg"

QuantilePlot[pp2];

Show[%310, FrameLabel → {{HoldForm[Empirical quantiles], None}, {HoldForm[Theoretical quantiles], None}},
 PlotLabel → HoldForm[Normal Q - Q - plot], LabelStyle → {GrayLevel[0]}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\QQ4.jpg", %311, "JPEG"]

"C:\\Users\\marte\\Dropbox\\Thesis\\QQ4.jpg"

(*We look at the matrix with trend removd*)

p3 = Flatten[matrix3];

borte = Position[p3, _ ? (# == "empty" &)];
pp3 = Delete[p3, borte];

mean3 = Mean[pp3];

sd3 = StandardDeviation[pp3]

hist3 = Histogram[pp3];
pl3 = ContourPlot[x == mean3, {x, -0.5, 0.5}, {y, 0, 57000}];

Show[hist3, pl3];

Histogram[{p2, p3}]

paired = PairedHistogram[pp3, pp2, ChartLegends → {{"Without trend", "With trend"}},
   ChartStyle → {"Pastel", None, None}, PlotLabel → "Correlations"];


Export["C:\\Users\\marte\\Dropbox\\Thesis\\.jpg", %311, "JPEG"]

mean3 - mean2;

sd3 - sd2;
```

Look at two-year periods.
Period 14.

```
startdato = AbsoluteTime["01.01.2014"]

pristab14tit = Table[0, {i, 1, n}];
datotab14tit = Table[0, {i, 1, n}];
Monitor[Do[abstid = Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[
      {StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &, Map[StringSplit[#, "."] &, datotab[[i]]]]]];
  (*split strengene der det er punktum, bytt om på dag mnd (europeisk-amerikansk måte),
  sett tilbake punktum, ta absolutt tid*)
  pos = Position[abstid, _ ? (# ≥ startdato &)];
  datotab14tit[[i]] = Extract [datotab[[i]], pos];
  pristab14tit[[i]] = Extract[pristab[[i]], pos];
  , {i, 1, n}], {i}]

(*Export["C:\\Users\\marte\\Dropbox\\Thesis\\datotab14tit.txt",datotab14tit]
 Export["C:\\Users\\marte\\Dropbox\\Thesis\\pristab14tit.txt",pristab14tit]*)
```

```
n = 811;

datotab14tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab14tit.txt"];
pristab14tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab14tit.txt"];

lengder14 = Table["empty", {i, 1, n}];
Do[
 lengder14[[i]] = Length[pristab14tit[[i]]]
 , {i, 1, n}]
logpristab14 = N[Log[pristab14tit]];

lengder14;

logpristab14 = N[Log[pristab14tit]];

(*Create the correlation matrix for period 14-15*)

matrix14tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab14tit[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab14[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab14tit[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab14[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix14tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_14_tit.txt", matrix14tit]

matrix14tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_14_tit.txt"];

(*look at the empirical distribution of the corr*)

mat14 = Flatten[matrix14tit];
hist14 = Histogram[mat14] (*make histogram of all values in the matrix*)

(*Check mean and sd. To do this I need to remove entries with "empty"*)

b14 = Position[mat14, _? (# == "empty" &)];
dist14 = Delete[mat14, b14];

mean14 = Mean[dist14];

StandardDeviation[dist14];

pl14 = ContourPlot[x == mean14, {x, -1, 1}, {y, 0, 8000}];
a = Show[hist14, pl14];
Labeled[a, {Rotate["Frequency", 90 Degree], "Correlations during the 14 period."}, {Left, Bottom}]

QuantilePlot[dist14]
```

Simulate uncorrelated series using normality assumtion.

```
n = Length[lengder14];

matriseliste14tit = {}; (*want to save every matrix k in a list*)
Monitor[
 Do[If[lengder14[[i]] == 0, lengder14[[i]] = 1], {i, 1, n}]
  Do[
   SomeTable14 = Table[RandomReal[NormalDistribution[0, 0.06], lengder14[[i]]], {i, 1, Length[lengder14]}];
   (*the simulated data*)
   n = Length[lengder14]; (*ant aksjer*)
   mat = Table[0, {n}, {n}];
   Do[
    If[Length[SomeTable14[[i]]] ≤ Length[SomeTable14[[j]]],
     len = Length[SomeTable14[[i]]], len = Length[SomeTable14[[j]]]];
    rekkei = Drop[SomeTable14[[i]], Length[SomeTable14[[i]]] - len];
    rekkej = Drop[SomeTable14[[j]], Length[SomeTable14[[j]]] - len];
    If[Min[Length[rekkei], Length[rekkej]] > 3 && Length[rekkei] == Length[rekkej] &&
      StandardDeviation[Drop[rekkei, 1]] > 0 && StandardDeviation[Drop[rekkej, -1]] > 0,
     korrelasjon = Correlation[Drop[rekkei, 1], Drop[rekkej, -1]] ;
     (*take the correlation of the prices*)
     mat[[i, j]] = korrelasjon;] (*put the correlation in the appropriate matrix coordinate*)

    , {i, 1, n}, {j, 1, n}];
   matriseliste14tit = Append[matriseliste14tit, mat];


   , {k, 1, 500}], {i, j, k}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14tit.txt", matriseliste14tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14tit.txt"

matriseliste14tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14tit.txt"];

matriseliste14tit[[1 ;; 2, 1 ;; 10, 1 ;; 10]] // MatrixForm (*take a look at the first entries*)
```

Bootstrap log return series to estimate the EDF.

```
n = Length[lengder14];

(*We use the following code with every period. So you need to replace 14 with 12,
10 etc. every time you want to look at a different period*)
```

```
matriseliste14boot2 = {};
Monitor[
 Do[
  matboot = Table[0, {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)

  Do[
   snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab14tit[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab14[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab14tit[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab14[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = RandomChoice[Drop[utvalgi, 1] - Drop[utvalgi, -1], Length[utvalgi]];
    returnsj = RandomChoice[Drop[utvalgj, 1] - Drop[utvalgj, -1], Length[utvalgj]];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matboot[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];

  matriseliste14boot2 = Append[matriseliste14boot2, matboot];
  , {k, 1, 50}], {i, j, k}]
(*Export some at the time to avoid memory issues*)
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14boot2.txt", matriseliste14boot2]

"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14boot2.txt"

matriseliste14boot = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14boot.txt"];

matriseliste14boot2 = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14boot2.txt"];

matriseliste14boot3 = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste14boot3.txt"];

Dimensions[matriseliste14boot]

matriseliste14BOOT = {};

matriseliste14BOOT =
  Append[Append[Append[matriseliste14BOOT, matriseliste14boot], matriseliste14boot2], matriseliste14boot3];
```

We need to change the zero's to 1's in qlistHigh and the zero's to -1 in qlisteLow when we use True/False matrix.

```
n = 811;

beta14 = Length[Position[matrix14tit, _? (# ≠ "empty" &)]]
 (*Set as Bonferoni- limit: Nr of non-empty entries. e.i. numer of tests performed*)

beta14 = 64 419;

qlisteHigh14 = Table[Quantile[matriseliste14tit[[All, i, j]], 1 - 0.05 / beta14], {i, 1, n}, {j, 1, n}];
(*find the quantiles of alpha signif.*)

qlisteHigh14boot = Table[Quantile[matriseliste14BOOT[[All, i, j]], 1 - 0.05 / beta14], {i, 1, n}, {j, 1, n}];
(*find the quantiles of alpha signif.*)

Do[
 If[qlisteHigh14[[i, j]] == 0, qlisteHigh14[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
```

```
Do[
 If[qlisteHigh14boot[[i, j]] == 0, qlisteHigh14boot[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]

MatrixForm[qlisteHigh14[[1 ;; 10, 1 ;; 10]]]

qlisteLow14 = Table[Quantile[matriseliste14tit[[All, i, j]], 0.05 / beta14], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow14[[i, j]] == 0, qlisteLow14[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]

qlisteLow14boot = Table[Quantile[matriseliste14BOOT[[All, i, j]], 0.05 / beta14], {i, 1, n}, {j, 1, n}];

Do[
 If[qlisteLow14boot[[i, j]] == 0, qlisteLow14boot[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]

MatrixForm[qlisteLow14[[1 ;; 10, 1 ;; 10]]]

(*Matrix tells which shares has signif. corr with which shares*)

TrueFalseMat14tit = Table[0, {n}, {n}];
Do[
  If[matrix14tit[[i, j]] >= qlisteHigh14[[i, j]], TrueFalseMat14tit[[i, j]] = 1, TrueFalseMat14tit[[i, j]] = 0]
   If[matrix14tit[[i, j]] <= qlisteLow14[[i, j]], TrueFalseMat14tit[[i, j]] = 1, TrueFalseMat14tit[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];

Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat14tit.txt", TrueFalseMat14tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat14tit.txt"

TrueFalseMat14tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat14tit.txt"];

TrueFalseMat14boot = Table[0, {n}, {n}];
Do[
  If[matrix14tit[[i, j]] >= qlisteHigh14boot[[i, j]],
    TrueFalseMat14boot[[i, j]] = 1, TrueFalseMat14boot[[i, j]] = 0]
   If[matrix14tit[[i, j]] <= qlisteLow14boot[[i, j]], TrueFalseMat14boot[[i, j]] = 1,
    TrueFalseMat14boot[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat14boot.txt", TrueFalseMat14boot]

TrueFalseMat14boot = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat14boot.txt"];

namelist = Split[names][[All, 1]][[1 ;; 811]];

(*count sign. corr. and find placement and name of shares*)

Count[Flatten[Flatten[TrueFalseMat14tit]], 1]

(*conut the number of signif. corr*)

trans14 = Transpose[TrueFalseMat14tit];

counting14tit = Table[0, {n}, {2}];
trans14tit = Transpose[TrueFalseMat14tit];
Do[
 counting14tit [[i, 1]] = Count[TrueFalseMat14tit[[i]], 1] + Count[trans14tit[[i]], 1];
 counting14tit [[i, 2]] = namelist[[i]];
 If[TrueFalseMat14tit[[i, i]] == 1, counting14tit[[i, 1]] = counting14tit[[i, 1]] - 1]
 (*remove the one that is counted twice*)
 , {i, 1, n}]

counting14tit; (*number of sign.corr. together with the name of the share at time t. This also show zeros*)

(*take only the ones that has sign. corr. 1 or more*)
```

```
mange14tit = Table[0, {n}];
Do[
 If[counting14tit[[i, 1]] ≥ 1, mange14tit[[i]] = 1 ]
 , {i, 1, n}]
tel14tit = Position[mange14tit, 1];
High14tit = Extract[counting14tit, tel14tit];

High14tit; (*this is BOTH ways together -- being predicted / predict others*)

Length[High14tit]

(*        ************The shares that predict others the most******************     *)
inf14others = Table[0, {n}, {2}];
Do[
 inf14others [[i, 1]] = Count[trans14tit[[i]], 1];
 inf14others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf14others;

(*dont look at those with zero*)

inf14 = Table[0, {n}];
Do[
 If[inf14others[[i, 1]] ≥ 1, inf14[[i]] = 1 ]
 , {i, 1, n}]
count14 = Position[inf14, 1];
HighInf14others = Extract[inf14others, count14];

HighInf14others;

Length[HighInf14others]
```

Period: 2012-2014.

```
(*cut into appropriate time periods*)

startdato12 = AbsoluteTime["01.01.2012"]
sluttdato12 = AbsoluteTime["01.01.2014"]

pristab12tit = Table[0, {i, 1, n}];
datotab12tit = Table[0, {i, 1, n}];
Monitor[Do[abstid = Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[
      {StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &, Map[StringSplit[#, "."] &, datotab[[i]]]]]];
  (*(europeisk-amerikansk måte)*)
  pos1 = Position[abstid, _? (# ≥ startdato12 &) ];
  pos2 = Position[abstid, _? (# < sluttdato12 &) ];
  pos = Intersection[pos1, pos2];
  datotab12tit[[i]] = Extract [datotab[[i]], pos];
  pristab12tit[[i]] = Extract[pristab[[i]], pos];
  , {i, 1, n}], {i}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\datotab12tit.txt", datotab12tit]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\pristab12tit.txt", pristab12tit]

datotab12tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab12tit.txt"];
pristab12tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab12tit.txt"];

logpristab12 = N[Log[pristab12tit]];

lengder12 = Table["empty", {i, 1, n}]; (*get length of the series*)
Do[
 lengder12[[i]] = Length[pristab12tit[[i]]]
 , {i, 1, n}]
```

“Matrix12tit” (period 2012-2014).

```
(*create correlation matrix for this period*)
```

```
matrix12tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab12tit[[i]], datotab12tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab12tit[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab12[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab12tit[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab12[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix12tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_12_tit.txt", matrix12tit]

matrix12tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_12_tit.txt"];

(*take a look at the standard deviation of the series in this period*)

distSD12 = {};
Do[
 utvalgi = logpristab12[[i]]; (*extract the values in these position for stock i*)
 If[Length[utvalgi] > 30, (*if the set is greater than 30, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  sd = StandardDeviation[returnsi];
  distSD12 = Append[distSD12, sd]]
 , {i, 1, n}]

Mean[distSD12] (*we well use the mean over the entire period*)
```

Test assuming idependence and normality.

```
matriseliste12tit = {}; (*want to save every matrix k in a list*)
Do[If[lengder12[[i]] == 0, lengder12[[i]] = 1], {i, 1, n}]
Do[
 SomeTable12 = Table[RandomReal[NormalDistribution[0, 0.06], lengder12[[i]]], {i, 1, Length[lengder12]}];
 (*cannot be neg*)
 n = Length[lengder12]; (*ant aksjer*)
 mat = Table[0, {n}, {n}];
 Do[
  If[Length[SomeTable12[[i]]] ≤ Length[SomeTable12[[j]]],
   len = Length[SomeTable12[[i]]], len = Length[SomeTable12[[j]]]];
  rekkei = Drop[SomeTable12[[i]], Length[SomeTable12[[i]]] - len];
  rekkej = Drop[SomeTable12[[j]], Length[SomeTable12[[j]]] - len];
  If[Min[Length[rekkei], Length[rekkej]] > 3 && Length[rekkei] == Length[rekkej] &&
    StandardDeviation[Drop[rekkei, 1]] > 0 && StandardDeviation[Drop[rekkej, -1]] > 0,
   korrelasjon = Correlation[Drop[rekkei, 1], Drop[rekkej, -1]] ;
   (*take the correlation of the prices*)
   mat[[i, j]] = korrelasjon;] (*put the correlation in the appropriate matrix coordinate*)

  , {i, 1, n}, {j, 1, n}];
 matriseliste12tit = Append[matriseliste12tit, mat];

 , {k, 1, 500}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste12tit.txt", matriseliste12tit]
```

```
"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste12tit.txt"

matriseliste12tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste12tit.txt"];
```

Limits of sign. and creating the True/false- matrix.

```
beta12 = Length[Position[matrix12tit, _ ? (# ≠ "empty" &)]] (*Set as Bonferoni-limit →number of tests performed*)

(*find the quantiles of alpha- signif. *)
qlisteHigh12 = Table[Quantile[matriseliste12tit[[All, i, j]], 1 - 0.05 / beta12], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteHigh12[[i, j]] == 0, qlisteHigh12[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow12 = Table[Quantile[matriseliste12tit[[All, i, j]], 0.05 / beta12], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow12[[i, j]] == 0, qlisteLow12[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]
(*create true-false- matrix*)
TrueFalseMat12tit = Table[0, {n}, {n}];
Do[
  If[matrix12tit[[i, j]] >= qlisteHigh12[[i, j]], TrueFalseMat12tit[[i, j]] = 1, TrueFalseMat12tit[[i, j]] = 0]
   If[matrix12tit[[i, j]] <= qlisteLow12[[i, j]], TrueFalseMat12tit[[i, j]] = 1, TrueFalseMat12tit[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat12tit.txt", TrueFalseMat12tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat12tit.txt"

TrueFalseMat12tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat12tit.txt"];
```

Do the same only with bootstrap method.

```
(*the following matrix is created using the same procedure/code as in period 14*)
mat12BOOT = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\mat12BOOT.txt"];
(*find the quantiles of alpha- signif. *)
qlisteHigh12boot = Table[Quantile[mat12BOOT[[All, i, j]], 1 - 0.05 / beta12], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteHigh12boot[[i, j]] == 0, qlisteHigh12boot[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow12boot = Table[Quantile[mat12BOOT[[All, i, j]], 0.05 / beta12], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow12boot[[i, j]] == 0, qlisteLow12boot[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]
(*create true-false- matrix*)
TrueFalseMat12boot = Table[0, {n}, {n}];
Do[
  If[matrix12tit[[i, j]] >= qlisteHigh12boot[[i, j]],
    TrueFalseMat12boot[[i, j]] = 1, TrueFalseMat12boot[[i, j]] = 0]
   If[matrix12tit[[i, j]] <= qlisteLow12boot[[i, j]], TrueFalseMat12boot[[i, j]] = 1,
    TrueFalseMat12boot[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat12boot.txt", TrueFalseMat12boot]
"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat12boot.txt"
```

Period: 2010-2012.

```
(*cut into appropriate time periods*)
```

```
startdato10 = AbsoluteTime["01.01.2010"]
 sluttdato10 = AbsoluteTime["01.01.2012"]
 pristab10tit = Table[0, {i, 1, n}];
 datotab10tit = Table[0, {i, 1, n}];
 Monitor[Do[abstid =
    Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[{StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &,
       Map[StringSplit[#, "."] &, datotab[[i]]]]]];
   pos1 = Position[abstid, _? (# ≥ startdato10 &)];
   pos2 = Position[abstid, _? (# < sluttdato10 &)];
   pos = Intersection[pos1, pos2];
   datotab10tit[[i]] = Extract [datotab[[i]], pos];
   pristab10tit[[i]] = Extract[pristab[[i]], pos];
   , {i, 1, n}], {i}]

 Export["C:\\Users\\marte\\Dropbox\\Thesis\\pristab10tit.txt", pristab10tit] *)

datotab10tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab10tit.txt"];
pristab10tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab10tit.txt"];

lengder10 = Table["empty", {i, 1, n}];
Do[
 lengder10[[i]] = Length[pristab10tit[[i]]]
 , {i, 1, n}]

logpristab10 = N[Log[pristab10tit]];
```

“Matrix10tit” (period 2010-2012).

```
(*create correlation matrix for this period*)

matrix10tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab10tit[[i]], datotab10tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab10tit[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab10[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab10tit[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab10[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix10tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_10_tit.txt", matrix10tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_10_tit.txt"

matrix10tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_10_tit.txt"];
```

Test assuming idependence and normality.

```
Monitor[
  matriseliste10tit = {}; (*want to save every matrix k in a list*)
  Do[If[lengder10[[i]] == 0, lengder10[[i]] = 1], {i, 1, n}]
   Do[
    SomeTable10 = Table[RandomReal[NormalDistribution[0, 0.06], lengder10[[i]]], {i, 1, Length[lengder10]}];
    (*cannot be neg*)
    n = Length[lengder10]; (*ant aksjer*)
    mat = Table[0, {n}, {n}];
    Do[
     If[Length[SomeTable10[[i]]] ≤ Length[SomeTable10[[j]]],
      len = Length[SomeTable10[[i]]], len = Length[SomeTable10[[j]]]];
     rekkei = Drop[SomeTable10[[i]], Length[SomeTable10[[i]]] - len];
     rekkej = Drop[SomeTable10[[j]], Length[SomeTable10[[j]]] - len];
     If[Min[Length[rekkei], Length[rekkej]] > 3 && Length[rekkei] == Length[rekkej] &&
        StandardDeviation[Drop[rekkei, 1]] > 0 && StandardDeviation[Drop[rekkej, -1]] > 0,
      korrelasjon = Correlation[Drop[rekkei, 1], Drop[rekkej, -1]] ;
      (*take the correlation of the prices*)
      mat[[i, j]] = korrelasjon;] (*put the correlation in the appropriate matrix coordinate*)

     , {i, 1, n}, {j, 1, n}];
    matriseliste10tit = Append[matriseliste10tit, mat];

    , {k, 1, 500}], {i, j, k}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste10tit.txt", matriseliste10tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste10tit.txt"

matriseliste10tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste10tit.txt"];

Dimensions[matriseliste10tit]
```

Limits of sign. and creating the True/false- matrix.

```
beta10 = Length[Position[matrix10tit, _?(# ≠ "empty" &)]] (*Set as Bonferoni-limit*)

qlisteHigh10 = Table[Quantile[matriseliste10tit[[All, i, j]], 1 - 0.05 / beta10], {i, 1, n}, {j, 1, n}];
(*find the quantiles of alpha signif. *)
Do[
 If[qlisteHigh10[[i, j]] == 0, qlisteHigh10[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]

qlisteLow10 = Table[Quantile[matriseliste10tit[[All, i, j]], 0.05 / beta10], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow10[[i, j]] == 0, qlisteLow10[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]

(*create true-false matrix for this period*)

TrueFalseMat10tit = Table[0, {n}, {n}];
Do[
  If[matrix10tit[[i, j]] >= qlisteHigh10[[i, j]], TrueFalseMat10tit[[i, j]] = 1, TrueFalseMat10tit[[i, j]] = 0]
   If[matrix10tit[[i, j]] <= qlisteLow10[[i, j]], TrueFalseMat10tit[[i, j]] = 1, TrueFalseMat10tit[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat10tit.txt", TrueFalseMat10tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat10tit.txt"

TrueFalseMat10tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat10tit.txt"];
```

Do the same only with bootstrap method.

```
(*the following matrix is created using the same procedure/code as in period 14*)
mat10BOOT = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\mat10BOOT.txt"];
(*find the quantiles of alpha- signif. *)
qlisteHigh10boot = Table[Quantile[mat10BOOT[[All, i, j]], 1 - 0.05 / beta10], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteHigh10boot[[i, j]] == 0, qlisteHigh10boot[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow10boot = Table[Quantile[mat10BOOT[[All, i, j]], 0.05 / beta10], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow10boot[[i, j]] == 0, qlisteLow10boot[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]
(*create true-false- matrix*)
TrueFalseMat10boot = Table[0, {n}, {n}];
Do[
  If[matrix10tit[[i, j]] >= qlisteHigh10boot[[i, j]],
    TrueFalseMat10boot[[i, j]] = 1, TrueFalseMat10boot[[i, j]] = 0]
   If[matrix10tit[[i, j]] <= qlisteLow10boot[[i, j]], TrueFalseMat10boot[[i, j]] = 1,
    TrueFalseMat10boot[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat10boot.txt", TrueFalseMat10boot]
"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat10boot.txt"
```

Period:2008-2010.

```
(*cut into appropriate time periods*)

startdato08 = AbsoluteTime["01.01.2008"]
sluttdato08 = AbsoluteTime["01.01.2010"]
pristab08tit = Table[0, {i, 1, n}];
datotab08tit = Table[0, {i, 1, n}];
Monitor[Do[abstid =
   Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[{StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &,
      Map[StringSplit[#, "."] &, datotab[[i]]]]]];
  pos1 = Position[abstid, _ ? (# ≥ startdato08 &)];
  pos2 = Position[abstid, _ ? (# < sluttdato08 &)];
  pos = Intersection[pos1, pos2];
  datotab08tit[[i]] = Extract [datotab[[i]], pos];
  pristab08tit[[i]] = Extract[pristab[[i]], pos];
  , {i, 1, n}], {i}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\pristab08tit.txt", pristab08tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\pristab08tit.txt"

datotab08tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab08tit.txt"];
pristab08tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab08tit.txt"];

lengder08 = Table["empty", {i, 1, n}];
Do[
 lengder08[[i]] = Length[pristab08tit[[i]]]
 , {i, 1, n}]


logpristab08 = N[Log[pristab08tit]];
```

“Matrix08tit” (period 2008-2010).

```
(*create correlation matrix for this period*)
```

```
matrix08tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab08tit[[i]], datotab08tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab08tit[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab08[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab08tit[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab08[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix08tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_08_tit.txt", matrix08tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_08_tit.txt"

matrix08tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_08_tit.txt"];
```

Test assuming idependence and normality.

```
Monitor[
 matriseliste08tit = {}; (*want to save every matrix k in a list*)
 Do[If[lengder08[[i]] == 0, lengder08[[i]] = 1], {i, 1, n}]
  Do[
   SomeTable08 = Table[RandomReal[NormalDistribution[0, 0.06], lengder08[[i]]], {i, 1, Length[lengder08]}];
   (*cannot be neg*)
   n = Length[lengder08]; (*ant aksjer*)
   mat = Table[0, {n}, {n}];
   Do[
    If[Length[SomeTable08[[i]]] ≤ Length[SomeTable08[[j]]],
     len = Length[SomeTable08[[i]]], len = Length[SomeTable08[[j]]]];
    rekkei = Drop[SomeTable08[[i]], Length[SomeTable08[[i]]] - len];
    rekkej = Drop[SomeTable08[[j]], Length[SomeTable08[[j]]] - len];
    If[Min[Length[rekkei], Length[rekkej]] > 3 && Length[rekkei] == Length[rekkej] &&
      StandardDeviation[Drop[rekkei, 1]] > 0 && StandardDeviation[Drop[rekkej, -1]] > 0,
     korrelasjon = Correlation[Drop[rekkei, 1], Drop[rekkej, -1]] ;
     (*take the correlation of the prices*)
     mat[[i, j]] = korrelasjon;] (*put the correlation in the appropriate matrix coordinate*)

    , {i, 1, n}, {j, 1, n}];
   matriseliste08tit = Append[matriseliste08tit, mat];

   , {k, 1, 500}], {i, j, k}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste08tit.txt", matriseliste08tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste08tit.txt"

matriseliste08tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste08tit.txt"];

Dimensions[matriseliste08tit]
```

Limits of sign. and creating the True/false- matrix.

```
beta08 = Length[Position[matrix08tit, _? (# ≠ "empty" &)]] (*Set as Bonferoni-limit*)
```

```
qlisteHigh08 = Table[Quantile[matriseliste08tit[[All, i, j]], 1 - 0.05 / beta08], {i, 1, n}, {j, 1, n}];
(*find the quantiles of alpha signif.*)
Do[
 If[qlisteHigh08[[i, j]] == 0, qlisteHigh08[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]

qlisteLow08 = Table[Quantile[matriseliste08tit[[All, i, j]], 0.05 / beta08], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow08[[i, j]] == 0, qlisteLow08[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]

(*create trur-false matrix*)

TrueFalseMat08tit = Table[0, {n}, {n}];
Do[
  If[matrix08tit[[i, j]] >= qlisteHigh08[[i, j]], TrueFalseMat08tit[[i, j]] = 1, TrueFalseMat08tit[[i, j]] = 0]
   If[matrix08tit[[i, j]] <= qlisteLow08[[i, j]], TrueFalseMat08tit[[i, j]] = 1, TrueFalseMat08tit[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat08tit.txt", TrueFalseMat08tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat08tit.txt"

TrueFalseMat08tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat08tit.txt"];
```

Do the same only with bootstrap method.

```
(*the following matrix is created using the same procedure/code as in period 14*)
mat08BOOT = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\mat08BOOT.txt"];
(*find the quantiles of alpha- signif. *)
qlisteHigh08boot = Table[Quantile[mat08BOOT[[All, i, j]], 1 - 0.05 / beta08], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteHigh08boot[[i, j]] == 0, qlisteHigh08boot[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow08boot = Table[Quantile[mat08BOOT[[All, i, j]], 0.05 / beta08], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow08boot[[i, j]] == 0, qlisteLow08boot[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]
(*create true-false- matrix*)
TrueFalseMat08boot = Table[0, {n}, {n}];
Do[
  If[matrix08tit[[i, j]] >= qlisteHigh08boot[[i, j]],
    TrueFalseMat08boot[[i, j]] = 1, TrueFalseMat08boot[[i, j]] = 0]
   If[matrix08tit[[i, j]] <= qlisteLow08boot[[i, j]], TrueFalseMat08boot[[i, j]] = 1,
    TrueFalseMat08boot[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat08boot.txt", TrueFalseMat08boot]
"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat08boot.txt"
```

Period:2006-2008.

```
(*cut into appropriate time periods*)
```

```
startdato06 = AbsoluteTime["01.01.2006"]
sluttdato06 = AbsoluteTime["01.01.2008"]
pristab06tit = Table[0, {i, 1, n}];
datotab06tit = Table[0, {i, 1, n}];
Monitor[Do[abstid = Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[
      {StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &, Map[StringSplit[#, "."] &, datotab[[i]]]]]];
  (*split strengene der det er punktum, bytt om på dag mnd (europeisk-amerikansk måte),
  sett tilbake punktum, ta absolutt tid*)
  pos1 = Position[abstid, _ ? (# ≥ startdato06 &)];
  pos2 = Position[abstid, _ ? (# < sluttdato06 &)];
  pos = Intersection[pos1, pos2];
  datotab06tit[[i]] = Extract [datotab[[i]], pos];
  pristab06tit[[i]] = Extract[pristab[[i]], pos];
  , {i, 1, n}], {i}]

Export["C:\\Users\\marte\\Dropbox\\Thesis\\pristab06tit.txt", pristab06tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\pristab06tit.txt"

datotab06tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab06tit.txt"];
pristab06tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab06tit.txt"];

lengder06 = Table["empty", {i, 1, n}];
Do[
 lengder06[[i]] = Length[pristab06tit[[i]]]
 , {i, 1, n}]

logpristab06 = N[Log[pristab06tit]];
```

“Matrix06tit” (period 2006-2008).

```
(*create correlation matrix for this period*)

matrix06tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datotab06tit[[i]], datotab06tit[[j]]];
   (*for each i and j, find the intersecting dates*)
   Posi = Position[datotab06tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab06[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datotab06tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab06[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix06tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_06_tit.txt", matrix06tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_06_tit.txt"

matrix06tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_06_tit.txt"];
```

Test assuming idependence and normality.

```
Monitor[
 matriseliste06tit = {}; (*want to save every matrix k in a list*)
 Do[If[lengder06[[i]] == 0, lengder06[[i]] = 1], {i, 1, n}]
  Do[
   SomeTable06 = Table[RandomReal[NormalDistribution[0, 0.06], lengder06[[i]]], {i, 1, Length[lengder06]}];
   (*cannot be neg*)
   n = Length[lengder06]; (*ant aksjer*)
   mat = Table[0, {n}, {n}];
   Do[
    If[Length[SomeTable06[[i]]] ≤ Length[SomeTable06[[j]]],
     len = Length[SomeTable06[[i]]], len = Length[SomeTable06[[j]]]];
    rekkei = Drop[SomeTable06[[i]], Length[SomeTable06[[i]]] - len];
    rekkej = Drop[SomeTable06[[j]], Length[SomeTable06[[j]]] - len];
    If[Min[Length[rekkei], Length[rekkej]] > 3 && Length[rekkei] == Length[rekkej] &&
      StandardDeviation[Drop[rekkei, 1]] > 0 && StandardDeviation[Drop[rekkej, -1]] > 0,
     korrelasjon = Correlation[Drop[rekkei, 1], Drop[rekkej, -1]] ;
     (*take the correlation of the prices*)
     mat[[i, j]] = korrelasjon;] (*put the correlation in the appropriate matrix coordinate*)

    , {i, 1, n}, {j, 1, n}];
   matriseliste06tit = Append[matriseliste06tit, mat];
   , {k, 1, 500}], {i, j, k}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste06tit.txt", matriseliste06tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste06tit.txt"

matriseliste06tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\matriseliste06tit.txt"];
```

Limits of sign. and creating the True/false- matrix.

```
beta06 = Length[Position[matrix06tit, _ ? (# ≠ "empty" &)]] (*Set as B-limit*)
qlisteHigh06 = Table[Quantile[matriseliste06tit[[All, i, j]], 1 - 0.05 / beta06], {i, 1, n}, {j, 1, n}];
(*find the quantiles of alpha signif.*)
Do[
 If[qlisteHigh06[[i, j]] == 0, qlisteHigh06[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow06 = Table[Quantile[matriseliste06tit[[All, i, j]], 0.05 / beta06], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow06[[i, j]] == 0, qlisteLow06[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]

(*create true-false matrix*)

TrueFalseMat06tit = Table[0, {n}, {n}];
Do[
  If[matrix06tit[[i, j]] >= qlisteHigh06[[i, j]], TrueFalseMat06tit[[i, j]] = 1, TrueFalseMat06tit[[i, j]] = 0]
   If[matrix06tit[[i, j]] <= qlisteLow06[[i, j]], TrueFalseMat06tit[[i, j]] = 1, TrueFalseMat06tit[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat06tit.txt", TrueFalseMat06tit]

"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat06tit.txt"

TrueFalseMat06tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat06tit.txt"];
```

Do the same only with bootstrap method.

```
(*the following matrix is created using the same procedure/code as in period 14*)
mat06BOOT = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\mat06BOOT.txt"];
(*find the quantiles of alpha- signif. *)
qlisteHigh06boot = Table[Quantile[mat06BOOT[[All, i, j]], 1 - 0.05 / beta06], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteHigh06boot[[i, j]] == 0, qlisteHigh06boot[[i, j]] = 1]
 , {i, 1, n}, {j, 1, n}]
qlisteLow06boot = Table[Quantile[mat06BOOT[[All, i, j]], 0.05 / beta06], {i, 1, n}, {j, 1, n}];
Do[
 If[qlisteLow06boot[[i, j]] == 0, qlisteLow06boot[[i, j]] = -1]
 , {i, 1, n}, {j, 1, n}]
(*create true-false- matrix*)
TrueFalseMat06boot = Table[0, {n}, {n}];
Do[
  If[matrix06tit[[i, j]] >= qlisteHigh06boot[[i, j]],
    TrueFalseMat06boot[[i, j]] = 1, TrueFalseMat06boot[[i, j]] = 0]
   If[matrix06tit[[i, j]] <= qlisteLow06boot[[i, j]], TrueFalseMat06boot[[i, j]] = 1,
    TrueFalseMat06boot[[i, j]] = 0]
  , {i, 1, n}, {j, 1, n}];
Export["C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat06boot.txt", TrueFalseMat06boot]
"C:\\Users\\marte\\Dropbox\\Thesis\\TrueFalseMat06boot.txt"
```

Method 1.
Find the shares that predict others the most in each period.

```
(*        ************The shares that predict others the most******************     *)
trans14 = Transpose[TrueFalseMat14tit];
inf14others = Table[0, {n}, {2}];
Do[
 inf14others [[i, 1]] = Count[trans14[[i]], 1];
 inf14others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf14others;

trans12 = Transpose[TrueFalseMat12tit];

inf12others = Table[0, {n}, {2}];
Do[
 inf12others [[i, 1]] = Count[trans12[[i]], 1];
 inf12others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf12others;

trans10 = Transpose[TrueFalseMat10tit];

inf10others = Table[0, {n}, {2}];
Do[
 inf10others [[i, 1]] = Count[trans10[[i]], 1];
 inf10others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf10others;

trans08 = Transpose[TrueFalseMat08tit];

inf08others = Table[0, {n}, {2}];
Do[
 inf08others [[i, 1]] = Count[trans08[[i]], 1];
 inf08others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf08others;
```

```
trans06 = Transpose[TrueFalseMat06tit];

inf06others = Table[0, {n}, {2}];
Do[
 inf06others [[i, 1]] = Count[trans06[[i]], 1];
 inf06others [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf06others;
```

Method 2.
Find the shares that predict others most in each period.

```
trans14boot = Transpose[TrueFalseMat14boot];

inf14othersboot = Table[0, {n}, {2}];
Do[
 inf14othersboot [[i, 1]] = Count[trans14boot[[i]], 1];
 inf14othersboot [[i, 2]] = namelist[[i]];
 , {i, 1, n}]
inf14othersboot;

trans12boot = Transpose[TrueFalseMat12boot];

inf12othersboot = Table[0, {n}, {2}];
Do[
 inf12othersboot [[i, 1]] = Count[trans12boot[[i]], 1];
 inf12othersboot [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf12othersboot;

trans10boot = Transpose[TrueFalseMat10boot];

inf10othersboot = Table[0, {n}, {2}];
Do[
 inf10othersboot [[i, 1]] = Count[trans10boot[[i]], 1];
 inf10othersboot [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf10othersboot;

trans08boot = Transpose[TrueFalseMat08boot];

inf08othersboot = Table[0, {n}, {2}];
Do[
 inf08othersboot [[i, 1]] = Count[trans08boot[[i]], 1];
 inf08othersboot [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf08othersboot;

trans06boot = Transpose[TrueFalseMat06boot];

inf06othersboot = Table[0, {n}, {2}];
Do[
 inf06othersboot [[i, 1]] = Count[trans06boot[[i]], 1];
 inf06othersboot [[i, 2]] = namelist[[i]];
 , {i, 1, n}]

inf06othersboot;
```

Are the shares that predict others the same in any two periods?
Method 1.
2012-2014 <--> 2014-2015.

```
pos12inf = Position[inf12others[[All, 1]], _? (# > 1 &)];
pos14inf = Position[inf14others[[All, 1]], _? (# > 1 &)];
bothInf1214 = Intersection[Extract[inf12others[[All, 2]], pos12inf], Extract[inf14others[[All, 2]], pos14inf]];

bothInf1214 (*these shares "predict" other shares in both periods*)

(*find the positions of these shares*)

pos1214 = Table["empty", {i, 1, Length[bothInf1214]}];
Do[
 pos1214[[i]] = Position[namelist, bothInf1214[[i]]]
 , {i, 1, Length[bothInf1214]}]

pos1214;

pos1214 = Flatten[Sort[pos1214]]
```

Find what shares that is predicted by the shares found above.

```
(*in period 14*)

affected14 = Table[0, {Length[pos1214]}, {2}];
Do[
 affected14 [[i, 1]] = Position[trans14[[pos1214[[i]]]], 1];
 affected14 [[i, 2]] = pos1214[[i]];

 , {i, 1, Length[pos1214]}]

affected14

(*in period 12*)

affected12 = Table[0, {Length[pos1214]}, {2}];
Do[
 affected12 [[i, 1]] = Position[trans12[[pos1214[[i]]]], 1];
 affected12 [[i, 2]] = pos1214[[i]];

 , {i, 1, Length[pos1214]}]

affected12

(*are these the same shares?*)

theSame = Table[0, {Length[affected12]}, {2}];
Do[
 theSame[[i, 1]] = Intersection[affected14 [[i, 1]], affected12[[i, 1]]];
 theSame[[i, 2]] = pos1214[[i]]
 , {i, 1, Length[affected12]}]

theSame
```

```
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame[[i]][[1]]], theSame[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSame]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSame[[i]][[1]]], theSame[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSame[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame]}];
teller
liste
```

Are the shares that predict others the same in any two periods?
Method 2.
2012-2014 <--> 2014-2015.

```
pos12infboot = Position[inf12othersboot[[All, 1]], _? (# > 1 &)];
pos14infboot = Position[inf14othersboot[[All, 1]], _? (# > 1 &)];
bothInf1214boot = Intersection[
   Extract[inf12othersboot[[All, 2]], pos12infboot], Extract[inf14othersboot[[All, 2]], pos14infboot]];

Length[pos14infboot = Position[inf14othersboot[[All, 1]], _? (# > 1 &)]]

bothInf1214boot (*these shares "predict" other shares in both periods*)

Length[bothInf1214boot]

(*find the positions of these shares*)

pos1214boot = Table["empty", {i, 1, Length[bothInf1214boot]}];
Do[
 pos1214boot[[i]] = Position[namelist, bothInf1214boot[[i]]]
 , {i, 1, Length[bothInf1214boot]}]

pos1214boot;

pos1214boot = Flatten[Sort[pos1214boot]]
```

Find what shares that is predicted by the shares found above.

```
(*in period 14*)

affected14boot = Table[0, {Length[pos1214boot]}, {2}];
Do[
 affected14boot [[i, 1]] = Position[trans14boot[[pos1214boot[[i]]]], 1];
 affected14boot [[i, 2]] = pos1214boot[[i]];

 , {i, 1, Length[pos1214boot]}]

affected14boot;

(*in period 12*)
```

```
affected12boot = Table[0, {Length[pos1214boot]}, {2}];
Do[
 affected12boot [[i, 1]] = Position[trans12boot[[pos1214boot[[i]]]], 1];
 affected12boot [[i, 2]] = pos1214boot[[i]];

 , {i, 1, Length[pos1214boot]}]

affected12boot;

(*are these the same shares?*)

theSameboot = Table[0, {Length[affected12boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected14boot [[i, 1]], affected12boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1214boot[[i]]
 , {i, 1, Length[affected12boot]}]

theSameboot

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2008-2010 vs 2010-2012.

```
pos08inf = Position[inf08others[[All, 1]], _?(# > 1 &)];
pos10inf = Position[inf10others[[All, 1]], _?(# > 1 &)];
bothInf0810 = Intersection[Extract[inf08others[[All, 2]], pos08inf], Extract[inf10others[[All, 2]], pos10inf]];

pos0810 = Table["empty", {i, 1, Length[bothInf0810]}];
Do[
 pos0810[[i]] = Position[namelist, bothInf0810[[i]]]
 , {i, 1, Length[bothInf0810]}]

pos0810 = Flatten[Sort[pos0810]]
```

Find what shares that is predicted by the shares found above.

```
(*in period 10*)

affected10 = Table[0, {Length[pos0810]}, {2}];
Do[
 affected10[[i, 1]] = Position[trans10[[pos0810[[i]]]], 1];
 affected10 [[i, 2]] = pos0810[[i]];

 , {i, 1, Length[pos0810]}]

affected10;
```

```
(*in period 08*)

affected08 = Table[0, {Length[pos0810]}, {2}];
Do[
 affected08[[i, 1]] = Position[trans08[[pos0810[[i]]]], 1];
 affected08 [[i, 2]] = pos0810[[i]];

 , {i, 1, Length[pos0810]}]

affected08;

(*are these the same in both periods?*)

theSame0810 = Table[0, {Length[affected08]}, {2}];
Do[
 theSame0810[[i, 1]] = Intersection[affected10 [[i, 1]], affected08[[i, 1]]];
 theSame0810[[i, 2]] = pos0810[[i]]
 , {i, 1, Length[affected08]}]

theSame0810

(*the shares*)

Length[bothInf0810]

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0810[[i]] [[1]]], theSame0810[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSame0810]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0810[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0810[[i]] [[1]]], theSame0810[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSame0810[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0810]}];
teller
liste
```

Method 2.
2008-2010 vs 2010-2012.

```
pos08infboot = Position[inf08othersboot[[All, 1]], _ ? (# > 1 &)];
pos10infboot = Position[inf10othersboot[[All, 1]], _ ? (# > 1 &)];
bothInf0810boot = Intersection[
   Extract[inf08othersboot[[All, 2]], pos08infboot], Extract[inf10othersboot[[All, 2]], pos10infboot]];

bothInf0810boot (*these shares "predict" other shares in both periods*)

(*find the positions of these shares*)

pos0810boot  = Table["empty", {i, 1, Length[bothInf0810boot]}];
Do[
 pos0810boot[[i]] = Position[namelist, bothInf0810boot[[i]]]
 , {i, 1, Length[bothInf0810boot]}]

pos0810boot;

pos0810boot = Flatten[Sort[pos0810boot]]
```

Find what shares that is predicted by the shares found above.

```
(*in period 10*)

affected10boot = Table[0, {Length[pos0810boot]}, {2}];
Do[
 affected10boot [[i, 1]] = Position[trans10boot[[pos0810boot[[i]]]], 1];
 affected10boot [[i, 2]] = pos0810boot[[i]];

 , {i, 1, Length[pos0810boot]}]

affected10boot;

(*in period 08*)

affected08boot = Table[0, {Length[pos0810boot]}, {2}];
Do[
 affected08boot [[i, 1]] = Position[trans08boot[[pos0810boot[[i]]]], 1];
 affected08boot [[i, 2]] = pos0810boot[[i]];

 , {i, 1, Length[pos0810boot]}]

affected08boot;

(*are these the same shares?*)

theSameboot = Table[0, {Length[affected08boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected10boot [[i, 1]], affected08boot[[i, 1]]];
 theSameboot[[i, 2]] = pos0810boot[[i]]
 , {i, 1, Length[affected08boot]}]

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1
2006-2008 vs 2010-2012.

```
pos06inf = Position[inf06others[[All, 1]], _? (# > 1 &)];
pos10inf = Position[inf10others[[All, 1]], _? (# > 1 &)];
bothInf0610 = Intersection[Extract[inf06others[[All, 2]], pos06inf], Extract[inf10others[[All, 2]], pos10inf]];

Length[bothInf0610]

pos0610 = Table["empty", {i, 1, Length[bothInf0610]}];
Do[
 pos0610[[i]] = Position[namelist, bothInf0610[[i]]]
 , {i, 1, Length[bothInf0610]}]
```

```
pos0610 = Flatten[Sort[pos0610]]
```

Find what shares that is predicted by the shares found above.

```
(*in period 10*)

affected10 = Table[0, {Length[pos0610]}, {2}];
Do[
 affected10[[i, 1]] = Position[trans10[[pos0610[[i]]]], 1];
 affected10 [[i, 2]] = pos0610[[i]];
 , {i, 1, Length[pos0610]}]

(*in period 06*)

affected06 = Table[0, {Length[pos0610]}, {2}];
Do[
 affected06[[i, 1]] = Position[trans06[[pos0610[[i]]]], 1];
 affected06 [[i, 2]] = pos0610[[i]];
 , {i, 1, Length[pos0610]}]

(*are these the same?*)

theSame0610 = Table[0, {Length[pos0610]}, {2}];
Do[
 theSame0610[[i, 1]] = Intersection[affected10 [[i, 1]], affected06[[i, 1]]];
 theSame0610[[i, 2]] = pos0610[[i]]
 , {i, 1, Length[pos0610]}]

theSame0610

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0610[[i]] [[1]]], theSame0610[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSame0610]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0610[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0610[[i]] [[1]]], theSame0610[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSame0610[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0610]}];
teller
liste
```

Method 2.
2006-2010 vs 2010-2012.

```
pos06infboot = Position[inf06othersboot[[All, 1]], _? (# > 1 &)];
pos10infboot = Position[inf10othersboot[[All, 1]], _? (# > 1 &)];
bothInf0610boot = Intersection[
   Extract[inf06othersboot[[All, 2]], pos06infboot], Extract[inf10othersboot[[All, 2]], pos10infboot]];

bothInf0610boot; (*these shares "predict" other shares in both periods*)

Length[bothInf0610boot]

(*find the positions of these shares*)
```

```
pos0610boot = Table["empty", {i, 1, Length[bothInf0610boot]}];
Do[
 pos0610boot[[i]] = Position[namelist, bothInf0610boot[[i]]]
 , {i, 1, Length[bothInf0610boot]}]

pos0610boot;

pos0610boot = Flatten[Sort[pos0610boot]];
```

Find what shares that is “predicted” by the shares found above.

```
(*in period 10*)

affected10boot = Table[0, {Length[pos0610boot]}, {2}];
Do[
 affected10boot [[i, 1]] = Position[trans10boot[[pos0610boot[[i]]]], 1];
 affected10boot [[i, 2]] = pos0610boot[[i]];

 , {i, 1, Length[pos0610boot]}]

affected10boot;

(*in period 06*)

affected06boot = Table[0, {Length[pos0610boot]}, {2}];
Do[
 affected06boot [[i, 1]] = Position[trans06boot[[pos0610boot[[i]]]], 1];
 affected06boot [[i, 2]] = pos0610boot[[i]];

 , {i, 1, Length[pos0610boot]}]

affected06boot;

(*are these the same shares?*)

theSameboot = Table[0, {Length[affected06boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected10boot [[i, 1]], affected06boot[[i, 1]]];
 theSameboot[[i, 2]] = pos0610boot[[i]]
 , {i, 1, Length[affected06boot]}]

theSameboot;

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2010-2012 vs 2014-2015.

```
pos14inf = Position[inf14others[[All, 1]], _? (# > 1 &)];
pos10inf = Position[inf10others[[All, 1]], _? (# > 1 &)];
bothInf1014 = Intersection[Extract[inf10others[[All, 2]], pos10inf], Extract[inf14others[[All, 2]], pos14inf]];
pos1014 = Table["empty", {i, 1, Length[bothInf1014]}];
Do[
 pos1014[[i]] = Position[namelist, bothInf1014[[i]]]
 , {i, 1, Length[bothInf1014]}]
pos1014 = Flatten[Sort[pos1014]]


affected10 = Table[0, {Length[pos1014]}, {2}];
Do[
 affected10[[i, 1]] = Position[trans10[[pos1014[[i]]]], 1];
 affected10 [[i, 2]] = pos1014[[i]];
 , {i, 1, Length[pos1014]}]

affected14 = Table[0, {Length[pos1014]}, {2}];
Do[
 affected14[[i, 1]] = Position[trans14[[pos1014[[i]]]], 1];
 affected14 [[i, 2]] = pos1014[[i]];
 , {i, 1, Length[pos1014]}]

theSame1014 = Table[0, {Length[affected10]}, {2}];
Do[
 theSame1014[[i, 1]] = Intersection[affected10 [[i, 1]], affected14[[i, 1]]];
 theSame1014[[i, 2]] = pos1014[[i]]
 , {i, 1, Length[affected10]}]

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame1014[[i]] [[1]]], theSame1014[[i]] [[2]]],
    teller ++;
   ];
  , {i, 1, Length [theSame1014]}];
teller

42


teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame1014[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame1014[[i]] [[1]]], theSame1014[[i]] [[2]]] == False,
    teller ++;
    liste = Append[liste, theSame1014[[i]]];
   ];
  , {i, 1, Length [theSame1014]}];
teller
liste
```

Method 2.
2010-2012 vs 2014-2015.

```
pos14infboot = Position[inf14othersboot[[All, 1]], _? (# > 1 &)];
pos10infboot = Position[inf10othersboot[[All, 1]], _? (# > 1 &)];
bothInf1410boot = Intersection[
   Extract[inf14othersboot[[All, 2]], pos14infboot], Extract[inf10othersboot[[All, 2]], pos10infboot]];

bothInf1410boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1410boot]

(*find the positions of these shares*)
```

```
pos1410boot = Table["empty", {i, 1, Length[bothInf1410boot]}];
Do[
 pos1410boot[[i]] = Position[namelist, bothInf1410boot[[i]]]
 , {i, 1, Length[bothInf1410boot]}]

pos1410boot;

pos1410boot = Flatten[Sort[pos1410boot]];

(*Find what shares that is “predicted” by the shares found above.*)
(*in period 10*)
affected10boot = Table[0, {Length[pos1410boot]}, {2}];
Do[
 affected10boot [[i, 1]] = Position[trans10boot[[pos1410boot[[i]]]], 1];
 affected10boot [[i, 2]] = pos1410boot[[i]];

 , {i, 1, Length[pos1410boot]}]
affected10boot;
(*in period 14*)
affected14boot = Table[0, {Length[pos1410boot]}, {2}];
Do[
 affected14boot [[i, 1]] = Position[trans14boot[[pos1410boot[[i]]]], 1];
 affected14boot [[i, 2]] = pos1410boot[[i]];

 , {i, 1, Length[pos1410boot]}]
affected14boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected14boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected10boot [[i, 1]], affected14boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1410boot[[i]]
 , {i, 1, Length[affected14boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2008-2010 vs 2014-2015.

```
pos14inf = Position[inf14others[[All, 1]], _?(# > 1 &)];
pos08inf = Position[inf08others[[All, 1]], _?(# > 1 &)];
bothInf0814 = Intersection[Extract[inf08others[[All, 2]], pos08inf], Extract[inf14others[[All, 2]], pos14inf]];
pos0814 = Table["empty", {i, 1, Length[bothInf0814]}];
Do[
 pos0814[[i]] = Position[namelist, bothInf0814[[i]]]
 , {i, 1, Length[bothInf0814]}]
test0814 = Flatten[Sort[pos0814]]

Length[bothInf0814]
```

```
affected08 = Table[0, {Length[test0814]}, {2}];
Do[
 affected08[[i, 1]] = Position[trans08[[test0814[[i]]]], 1];
 affected08 [[i, 2]] = test0814[[i]];
 , {i, 1, Length[pos0814]}]

affected14 = Table[0, {Length[test0814]}, {2}];
Do[
 affected14[[i, 1]] = Position[trans14[[test0814[[i]]]], 1];
 affected14 [[i, 2]] = test0814[[i]];
 , {i, 1, Length[pos0814]}]

theSame0814 = Table[0, {Length[affected08]}, {2}];
Do[
 theSame0814[[i, 1]] = Intersection[affected08 [[i, 1]], affected14[[i, 1]]];
 theSame0814[[i, 2]] = test0814[[i]]
 , {i, 1, Length[affected08]}]
theSame0814

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0814[[i]] [[1]]], theSame0814[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSame0814]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0814[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0814[[i]] [[1]]], theSame0814[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSame0814[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0814]}];
teller
liste
```

Method 2.
2008-2012 vs 2014-2015.

```
pos14infboot = Position[inf14othersboot[[All, 1]], _?(# > 1 &)];
pos08infboot = Position[inf08othersboot[[All, 1]], _?(# > 1 &)];
bothInf1408boot = Intersection[
   Extract[inf14othersboot[[All, 2]], pos14infboot], Extract[inf08othersboot[[All, 2]], pos08infboot]];

bothInf1408boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1408boot]

 (*find the positions of these shares*)

pos1408boot  = Table["empty", {i, 1, Length[bothInf1408boot]}];
Do[
 pos1408boot[[i]] = Position[namelist, bothInf1408boot[[i]]]
 , {i, 1, Length[bothInf1408boot]}]

pos1408boot;

pos1408boot = Flatten[Sort[pos1408boot]];
```

```
(*Find what shares that is "predictd" by the shares found above.*)
(*in period 08*)
affected08boot = Table[0, {Length[pos1408boot]}, {2}];
Do[
 affected08boot [[i, 1]] = Position[trans08boot[[pos1408boot[[i]]]], 1];
 affected08boot [[i, 2]] = pos1408boot[[i]];

 , {i, 1, Length[pos1408boot]}]
affected08boot;
(*in period 14*)
affected14boot = Table[0, {Length[pos1408boot]}, {2}];
Do[
 affected14boot [[i, 1]] = Position[trans14boot[[pos1408boot[[i]]]], 1];
 affected14boot [[i, 2]] = pos1408boot[[i]];

 , {i, 1, Length[pos1408boot]}]
affected14boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected14boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected08boot [[i, 1]], affected14boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1408boot[[i]]
 , {i, 1, Length[affected14boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]] [[1]]], theSameboot[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]] [[1]]], theSameboot[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2006-2008 vs. 2014-2015.

```
pos14inf = Position[inf14others[[All, 1]], _ ? (# > 1 &)];
pos06inf = Position[inf06others[[All, 1]], _ ? (# > 1 &)];
bothInf0614 = Intersection[Extract[inf06others[[All, 2]], pos06inf], Extract[inf14others[[All, 2]], pos14inf]];
pos0614 = Table["empty", {i, 1, Length[bothInf0614]}];
Do[
 pos0614[[i]] = Position[namelist, bothInf0614[[i]]]
 , {i, 1, Length[bothInf0614]}]
test0614 = Flatten[Sort[pos0614]]

Length[bothInf0614]
```

```
affected06 = Table[0, {Length[test0614]}, {2}];
Do[
 affected06[[i, 1]] = Position[trans06[[test0614[[i]]]], 1];
 affected06 [[i, 2]] = test0614[[i]];
 , {i, 1, Length[pos0614]}]

affected14 = Table[0, {Length[test0614]}, {2}];
Do[
 affected14[[i, 1]] = Position[trans14[[test0614[[i]]]], 1];
 affected14 [[i, 2]] = test0614[[i]];
 , {i, 1, Length[pos0614]}]

theSame0614 = Table[0, {Length[affected06]}, {2}];
Do[
 theSame0614[[i, 1]] = Intersection[affected06 [[i, 1]], affected14[[i, 1]]];
 theSame0614[[i, 2]] = test0614[[i]]
 , {i, 1, Length[affected06]}]
theSame0614

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0614[[i]] [[1]]], theSame0614[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSame0614]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0614[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0614[[i]] [[1]]], theSame0614[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSame0614[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0614]}];
teller
liste
```

Method 2.
2006-2012 vs 2014-2015.

```
pos14infboot = Position[inf14othersboot[[All, 1]], _? (# > 1 &)];
pos06infboot = Position[inf06othersboot[[All, 1]], _? (# > 1 &)];
bothInf1406boot = Intersection[
   Extract[inf14othersboot[[All, 2]], pos14infboot], Extract[inf06othersboot[[All, 2]], pos06infboot]];

bothInf1406boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1406boot]

(*find the positions of these shares*)

pos1406boot  = Table["empty", {i, 1, Length[bothInf1406boot]}];
Do[
 pos1406boot[[i]] = Position[namelist, bothInf1406boot[[i]]]
 , {i, 1, Length[bothInf1406boot]}]

pos1406boot;

pos1406boot = Flatten[Sort[pos1406boot]];
```

```
(*Find what shares that is “predictd” by the shares found above.*)
(*in period 06*)
affected06boot = Table[0, {Length[pos1406boot]}, {2}];
Do[
 affected06boot [[i, 1]] = Position[trans06boot[[pos1406boot[[i]]]], 1];
 affected06boot [[i, 2]] = pos1406boot[[i]];

 , {i, 1, Length[pos1406boot]}]
affected06boot;
(*in period 14*)
affected14boot = Table[0, {Length[pos1406boot]}, {2}];
Do[
 affected14boot [[i, 1]] = Position[trans14boot[[pos1406boot[[i]]]], 1];
 affected14boot [[i, 2]] = pos1406boot[[i]];

 , {i, 1, Length[pos1406boot]}]
affected14boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected14boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected06boot [[i, 1]], affected14boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1406boot[[i]]
 , {i, 1, Length[affected14boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]] [[1]]], theSameboot[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]] [[1]]], theSameboot[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2010-2012 vs. 2012-2014.

```
pos12inf = Position[inf12others[[All, 1]], _? (# > 1 &)];
pos10inf = Position[inf10others[[All, 1]], _? (# > 1 &)];
bothInf1012 = Intersection[Extract[inf10others[[All, 2]], pos10inf], Extract[inf12others[[All, 2]], pos12inf]];
pos1012 = Table["empty", {i, 1, Length[bothInf1012]}];
Do[
 pos1012[[i]] = Position[namelist, bothInf1012[[i]]]
 , {i, 1, Length[bothInf1012]}]
test1012 = Flatten[Sort[pos1012]]

Length[bothInf1012]
```

```
affected10 = Table[0, {Length[test1012]}, {2}];
Do[
 affected10[[i, 1]] = Position[trans10[[test1012[[i]]]], 1];
 affected10 [[i, 2]] = test1012[[i]];
 , {i, 1, Length[pos1012]}]

affected12 = Table[0, {Length[test1012]}, {2}];
Do[
 affected12[[i, 1]] = Position[trans12[[test1012[[i]]]], 1];
 affected12 [[i, 2]] = test1012[[i]];
 , {i, 1, Length[pos1012]}]

theSame1012 = Table[0, {Length[affected10]}, {2}];
Do[
 theSame1012[[i, 1]] = Intersection[affected10 [[i, 1]], affected12[[i, 1]]];
 theSame1012[[i, 2]] = test1012[[i]]
 , {i, 1, Length[affected10]}]
theSame1012

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame1012[[i]][[1]]], theSame1012[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSame1012]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame1012[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSame1012[[i]][[1]]], theSame1012[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSame1012[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame1012]}];
teller
liste
```

Method 2.
2010-2012 vs 2012-2014.

```
pos12infboot = Position[inf12othersboot[[All, 1]], _?(# > 1 &)];
pos10infboot = Position[inf10othersboot[[All, 1]], _?(# > 1 &)];
bothInf1210boot = Intersection[
   Extract[inf12othersboot[[All, 2]], pos12infboot], Extract[inf10othersboot[[All, 2]], pos10infboot]];

bothInf1210boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1210boot]

(*find the positions of these shares*)

pos1210boot  = Table["empty", {i, 1, Length[bothInf1210boot]}];
Do[
 pos1210boot[[i]] = Position[namelist, bothInf1210boot[[i]]]
 , {i, 1, Length[bothInf1210boot]}]

pos1210boot;

pos1210boot = Flatten[Sort[pos1210boot]];
```

```
(*Find what shares that is “predicted” by the shares found above.*)
(*in period 10*)
affected10boot = Table[0, {Length[pos1210boot]}, {2}];
Do[
 affected10boot [[i, 1]] = Position[trans10boot[[pos1210boot[[i]]]], 1];
 affected10boot [[i, 2]] = pos1210boot[[i]];

 , {i, 1, Length[pos1210boot]}]
affected10boot;
(*in period 12*)
affected12boot = Table[0, {Length[pos1210boot]}, {2}];
Do[
 affected12boot [[i, 1]] = Position[trans12boot[[pos1210boot[[i]]]], 1];
 affected12boot [[i, 2]] = pos1210boot[[i]];

 , {i, 1, Length[pos1210boot]}]
affected12boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected12boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected10boot [[i, 1]], affected12boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1210boot[[i]]
 , {i, 1, Length[affected12boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2008-2010 vs. 2012-2014.

```
pos12inf = Position[inf12others[[All, 1]], _?(# > 1 &)];
pos08inf = Position[inf08others[[All, 1]], _?(# > 1 &)];
bothInf0812 = Intersection[Extract[inf08others[[All, 2]], pos08inf], Extract[inf12others[[All, 2]], pos12inf]];
pos0812 = Table["empty", {i, 1, Length[bothInf0812]}];
Do[
 pos0812[[i]] = Position[namelist, bothInf0812[[i]]]
 , {i, 1, Length[bothInf0812]}]
test0812 = Flatten[Sort[pos0812]]

Length[bothInf0812]
```

```
affected08 = Table[0, {Length[test0812]}, {2}];
Do[
 affected08[[i, 1]] = Position[trans08[[test0812[[i]]]], 1];
 affected08 [[i, 2]] = test0812[[i]];
 , {i, 1, Length[pos0812]}]

affected12 = Table[0, {Length[test0812]}, {2}];
Do[
 affected12[[i, 1]] = Position[trans12[[test0812[[i]]]], 1];
 affected12 [[i, 2]] = test0812[[i]];
 , {i, 1, Length[pos0812]}]

theSame0812 = Table[0, {Length[affected08]}, {2}];
Do[
 theSame0812[[i, 1]] = Intersection[affected08 [[i, 1]], affected12[[i, 1]]];
 theSame0812[[i, 2]] = test0812[[i]]
 , {i, 1, Length[affected08]}]
theSame0812

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0812[[i]][[1]]], theSame0812[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSame0812]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0812[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0812[[i]][[1]]], theSame0812[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSame0812[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0812]}];
teller
liste
```

Method 2.
2008-2012 vs 2012-2014.

```
pos12infboot = Position[inf12othersboot[[All, 1]], _?(# > 1 &)];
pos08infboot = Position[inf08othersboot[[All, 1]], _?(# > 1 &)];
bothInf1208boot = Intersection[
   Extract[inf12othersboot[[All, 2]], pos12infboot], Extract[inf08othersboot[[All, 2]], pos08infboot]];

bothInf1208boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1208boot]

 (*find the positions of these shares*)

pos1208boot  = Table["empty", {i, 1, Length[bothInf1208boot]}];
Do[
 pos1208boot[[i]] = Position[namelist, bothInf1208boot[[i]]]
 , {i, 1, Length[bothInf1208boot]}]

pos1208boot;

pos1208boot = Flatten[Sort[pos1208boot]];
```

```
(*Find what shares that is “predictd” by the shares found above.*)
(*in period 08*)
affected08boot = Table[0, {Length[pos1208boot]}, {2}];
Do[
 affected08boot [[i, 1]] = Position[trans08boot[[pos1208boot[[i]]]], 1];
 affected08boot [[i, 2]] = pos1208boot[[i]];

 , {i, 1, Length[pos1208boot]}]
affected08boot;
(*in period 12*)
affected12boot = Table[0, {Length[pos1208boot]}, {2}];
Do[
 affected12boot [[i, 1]] = Position[trans12boot[[pos1208boot[[i]]]], 1];
 affected12boot [[i, 2]] = pos1208boot[[i]];

 , {i, 1, Length[pos1208boot]}]
affected12boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected12boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected08boot [[i, 1]], affected12boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1208boot[[i]]
 , {i, 1, Length[affected12boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2006-2008 vs. 2012-2014.

```
pos12inf = Position[inf12others[[All, 1]], _? (# > 1 &)];
pos06inf = Position[inf06others[[All, 1]], _? (# > 1 &)];
bothInf0612 = Intersection[Extract[inf06others[[All, 2]], pos06inf], Extract[inf12others[[All, 2]], pos12inf]];
pos0612 = Table["empty", {i, 1, Length[bothInf0612]}];
Do[
 pos0612[[i]] = Position[namelist, bothInf0612[[i]]]
 , {i, 1, Length[bothInf0612]}]
test0612 = Flatten[Sort[pos0612]]

Length[bothInf0612]
```

```
affected06 = Table[0, {Length[test0612]}, {2}];
Do[
 affected06[[i, 1]] = Position[trans06[[test0612[[i]]]], 1];
 affected06 [[i, 2]] = test0612[[i]];
 , {i, 1, Length[pos0612]}]

affected12 = Table[0, {Length[test0612]}, {2}];
Do[
 affected12[[i, 1]] = Position[trans12[[test0612[[i]]]], 1];
 affected12 [[i, 2]] = test0612[[i]];
 , {i, 1, Length[pos0612]}]

theSame0612 = Table[0, {Length[affected06]}, {2}];
Do[
 theSame0612[[i, 1]] = Intersection[affected06 [[i, 1]], affected12[[i, 1]]];
 theSame0612[[i, 2]] = test0612[[i]]
 , {i, 1, Length[affected06]}]
theSame0612

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0612[[i]][[1]]], theSame0612[[i]][[2]]], (*find how many are AC*)
    teller++;
   ];
  , {i, 1, Length [theSame0612]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0612[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0612[[i]][[1]]], theSame0612[[i]][[2]]] == False,
    teller++; (*find how many are CC*)
    liste = Append[liste, theSame0612[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0612]}];
teller
liste
```

Method 2.
2006-2012 vs 2012-2014.

```
pos12infboot = Position[inf12othersboot[[All, 1]], _?(# > 1 &)];
pos06infboot = Position[inf06othersboot[[All, 1]], _?(# > 1 &)];
bothInf1206boot = Intersection[
   Extract[inf12othersboot[[All, 2]], pos12infboot], Extract[inf06othersboot[[All, 2]], pos06infboot]];

bothInf1206boot; (*these shares "predict" other shares in both periods*)

Length[bothInf1206boot]

(*find the positions of these shares*)

pos1206boot  = Table["empty", {i, 1, Length[bothInf1206boot]}];
Do[
 pos1206boot[[i]] = Position[namelist, bothInf1206boot[[i]]]
 , {i, 1, Length[bothInf1206boot]}]

pos1206boot;

pos1206boot = Flatten[Sort[pos1206boot]];
```

```
(*Find what shares that is “inf by the shares found above.*)
(*in period 06*)
affected06boot = Table[0, {Length[pos1206boot]}, {2}];
Do[
 affected06boot [[i, 1]] = Position[trans06boot[[pos1206boot[[i]]]], 1];
 affected06boot [[i, 2]] = pos1206boot[[i]];

 , {i, 1, Length[pos1206boot]}]
affected06boot;
(*in period 12*)
affected12boot = Table[0, {Length[pos1206boot]}, {2}];
Do[
 affected12boot [[i, 1]] = Position[trans12boot[[pos1206boot[[i]]]], 1];
 affected12boot [[i, 2]] = pos1206boot[[i]];

 , {i, 1, Length[pos1206boot]}]
affected12boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected12boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected06boot [[i, 1]], affected12boot[[i, 1]]];
 theSameboot[[i, 2]] = pos1206boot[[i]]
 , {i, 1, Length[affected12boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Method 1.
2006-2008 vs. 2008-2010.

```
pos08inf = Position[inf08others[[All, 1]], _? (# > 1 &)];
pos06inf = Position[inf06others[[All, 1]], _? (# > 1 &)];
bothInf0608 = Intersection[Extract[inf06others[[All, 2]], pos06inf], Extract[inf08others[[All, 2]], pos08inf]];
pos0608 = Table["empty", {i, 1, Length[bothInf0608]}];
Do[
 pos0608[[i]] = Position[namelist, bothInf0608[[i]]]
 , {i, 1, Length[bothInf0608]}]
test0608 = Flatten[Sort[pos0608]]

Length[bothInf0608]
```

```
affected06 = Table[0, {Length[test0608]}, {2}];
Do[
 affected06[[i, 1]] = Position[trans06[[test0608[[i]]]], 1];
 affected06 [[i, 2]] = test0608[[i]];
 , {i, 1, Length[pos0608]}]

affected08 = Table[0, {Length[test0608]}, {2}];
Do[
 affected08[[i, 1]] = Position[trans08[[test0608[[i]]]], 1];
 affected08 [[i, 2]] = test0608[[i]];
 , {i, 1, Length[pos0608]}]

theSame0608 = Table[0, {Length[affected06]}, {2}];
Do[
 theSame0608[[i, 1]] = Intersection[affected06 [[i, 1]], affected08[[i, 1]]];
 theSame0608[[i, 2]] = test0608[[i]]
 , {i, 1, Length[affected06]}]
theSame0608

teller = 0;
Do[
  If[
    MemberQ[Flatten [theSame0608[[i]] [[1]]], theSame0608[[i]] [[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSame0608]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSame0608[[i]] [[1]]] ≠ {} &&
     MemberQ[Flatten [theSame0608[[i]] [[1]]], theSame0608[[i]] [[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSame0608[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSame0608]}];
teller
liste
```

Method 2.
2006-2012 vs 2008-2010.

```
pos08infboot = Position[inf08othersboot[[All, 1]], _?(# > 1 &)];
pos06infboot = Position[inf06othersboot[[All, 1]], _?(# > 1 &)];
bothInf0806boot = Intersection[
   Extract[inf08othersboot[[All, 2]], pos08infboot], Extract[inf06othersboot[[All, 2]], pos06infboot]];

bothInf0806boot; (*these shares "predict" other shares in both periods*)

Length[bothInf0806boot]

 (*find the positions of these shares*)

pos0806boot  = Table["empty", {i, 1, Length[bothInf0806boot]}];
Do[
 pos0806boot[[i]] = Position[namelist, bothInf0806boot[[i]]]
 , {i, 1, Length[bothInf0806boot]}]

pos0806boot;

pos0806boot = Flatten[Sort[pos0806boot]];
```

```
(*Find what shares that is “predicted” by the shares found above.*)
(*in period 06*)
affected06boot = Table[0, {Length[pos0806boot]}, {2}];
Do[
 affected06boot [[i, 1]] = Position[trans06boot[[pos0806boot[[i]]]], 1];
 affected06boot [[i, 2]] = pos0806boot[[i]];

 , {i, 1, Length[pos0806boot]}]
affected06boot;
(*in period 08*)
affected08boot = Table[0, {Length[pos0806boot]}, {2}];
Do[
 affected08boot [[i, 1]] = Position[trans08boot[[pos0806boot[[i]]]], 1];
 affected08boot [[i, 2]] = pos0806boot[[i]];

 , {i, 1, Length[pos0806boot]}]
affected08boot;
(*are these the same shares?*)
theSameboot = Table[0, {Length[affected08boot]}, {2}];
Do[
 theSameboot[[i, 1]] = Intersection[affected06boot [[i, 1]], affected08boot[[i, 1]]];
 theSameboot[[i, 2]] = pos0806boot[[i]]
 , {i, 1, Length[affected08boot]}]
theSameboot;
teller = 0;
Do[
  If[
    MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]], (*find how many are AC*)
    teller ++;
   ];
  , {i, 1, Length [theSameboot]}];
teller

teller = 0;
liste = {};
Do[
  If[
    Flatten [theSameboot[[i]][[1]]] ≠ {} &&
     MemberQ[Flatten [theSameboot[[i]][[1]]], theSameboot[[i]][[2]]] == False,
    teller ++; (*find how many are CC*)
    liste = Append[liste, theSameboot[[i]]]; (*what shares are these*)
   ];
  , {i, 1, Length [theSameboot]}];
teller
liste
```

Relation-Plots

```
shares1 = {14, 174, 200, 465, 597, 621, 658, 727, 796, 806}

namelist = Split[names][[All, 1]][[1 ;; 811]];

n = Length[shares1];

matrix10only10 = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
names10only10 = Table["empty", {i, 1, n}];

Do[
  matrix10only10[[i, j]] = TrueFalseMat10tit[[shares1[[i]], shares1[[j]]]];
  names10only10[[j]] = namelist[[shares1[[j]]]]
  , {i, 1, n}, {j, 1, n}];

matrix10only10 // MatrixForm;

trans10only10 = Transpose[matrix10only10]; (*transpose becase i=j in code*)

GraphPlot[trans10only10, VertexLabeling → True, DirectedEdges → True, SelfLoopStyle → All,
 VertexRenderingFunction → (Text[shares1[[#2]], #1, Background → Green] &), ImageSize → 500]
```

```
shares = {587, 174, 796, 415, 579, 621, 30, 501, 32, 686, 798, 597, 14, 319, 172, 508, 658, 639, 234, 602};
shares2 = Sort[shares]
n = Length[shares2]

matrix10only20 = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
names10only20 = Table["empty", {i, 1, n}];
Do[
  matrix10only20[[i, j]] = TrueFalseMat10tit[[shares2[[i]], shares2[[j]]]];
  names10only20[[j]] = namelist[[shares2[[j]]]]
  , {i, 1, n}, {j, 1, n}];
trans10only20 = Transpose[matrix10only20];(*transpose becase i=j in code*)



GraphPlot[trans10only20, VertexLabeling → True, DirectedEdges → True, SelfLoopStyle → All,
 VertexRenderingFunction → (Text[shares2[[#2]], #1, Background → Yellow] &), ImageSize → 1000]

shares1 = {14, 234, 15, 651, 732, 32, 534, 165, 258, 586, 181, 268,
  246, 465, 565, 728, 483, 206, 528, 18, 156, 475, 540, 572, 358, 727, 2, 796, 135, 283}

shares1 = Sort[shares1]

n = Length[shares1]

n = 15

matrix14only30 = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Do[
  matrix14only30[[i, j]] = TrueFalseMat14boot[[shares1[[i]], shares1[[j]]]];
  , {i, 1, n}, {j, 1, n}];
trans14only30 = Transpose[matrix14only30];(*transpose becase i=j in code*)

GraphPlot[trans14only30, VertexLabeling → True, DirectedEdges → True, SelfLoopStyle → All,
 VertexRenderingFunction → (Text[shares1[[#2]], #1, Background → Yellow] &), ImageSize → 1000]

(*look at some periods with simulated returns*)

flat12 = Flatten[matriseliste12tit[[1]]];
flat10 = Flatten[matriseliste10tit[[1]]];
flat08 = Flatten[matriseliste08tit[[1]]];

Dimensions[flat12];

delete12 = Position[flat12, _? (# == "empty" &) ];
dist12 = Delete[flat12, delete12];
delete10 = Position[flat10, _? (# == "empty" &) ];
dist10 = Delete[flat10, delete10];
delete08 = Position[flat08, _? (# == "empty" &) ];
dist08 = Delete[flat08, delete08];

mean12 = Mean[dist12]

mean10 = Mean[dist10]
mean08 = Mean[dist08]

StandardDeviation[dist12]
StandardDeviation[dist10]

hist12 = Histogram[dist12, 5];
p12 = ContourPlot[x == mean12, {x, -0.5, 0.5}, {y, 0, 570000}];

Show[hist12, p12]

hist10 = Histogram[dist10, 5];
p10 = ContourPlot[x == mean10, {x, -0.5, 0.5}, {y, 0, 570000}];
Show[hist10, p10]

all = dist12 + dist10 + dist08;

histAll = Histogram[all, 100]
```

```
Mean[all]

Variance[all]
```

Look at what happens when we remove the returns with the greates abs. values.
Method 1.

```
(*limits of sign.*)

i =; (*add appropriate share*)
j =; (*add appropriate share*)
qlisteHigh08[[i, j]](*change period and shares depending on interest*)
qlisteLow08[[i, j]]

(*corr. before we remove days*)

matrix10tit[[357, 796]] (*change period and shares depending on interest*)

(*take the correlation again, only this time without the greatest valued returns*)

i = 639;(*change period and shares depending on interest*)
j = 30;
snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab14tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab14[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab14tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab14[[j]], Posj];(*extract the values in these position for stock j*)
If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  bort = Map[{#} &, Reverse[Ordering[Abs[returnsi]]][[1 ;; 2]]];
  (*remove the greatest valued returns*)
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
  returnsi = Delete[returnsi, bort];
  returnsj = Delete[returnsj, bort];
  If[Min[Length[returnsi], Length[returnsj]] > 3 && Length[returnsi] == Length[returnsj] &&
    StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
    returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
   korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ,
   korrelasjon = {}(*take the correlation of the series*)
  ]];
korrelasjon
```

Look at what happens when we remove the returns with the greates values.
Method 2.

```
(*limits of sign.*)

i = 253;
j = 658;
qlisteHigh08boot[[i, j]](*change period and shares depending on interest*)
qlisteLow08boot[[i, j]]

(*corr. before we remove days*)

matrix08tit[[i, j]] (*change period and shares depending on interest*)

(*take the correlation again, only this time without the greatest valued returns*)
```

```
snitt = Intersection[datotab08tit[[i]], datotab08tit[[j]]]; (*for each i and j, find the intersecting dates*)
Posi = Position[datotab08tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab08[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab08tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab08[[j]], Posj];(*extract the values in these position for stock j*)
If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  bort = Map[{#} &, Reverse[Ordering[Abs[returnsi]]][[1 ;; 8]]];
  (*remove the greatest valued returns*)
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
  returnsi = Delete[returnsi, bort];
  returnsj = Delete[returnsj, bort];
  If[Min[Length[returnsi], Length[returnsj]] > 3 && Length[returnsi] == Length[returnsj] &&
    StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
    returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
   korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ,
   korrelasjon = {}(*take the correlation of the series*)
  ]];
korrelasjon

Length[returnsi]

datotab04tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab04tit.txt"];
pristab04tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab04tit.txt"];
logpristab04 = N[Log[pristab04tit]];

datotab02tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab02tit.txt"];
pristab02tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab02tit.txt"];
logpristab02 = N[Log[pristab02tit]];

datotab00tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab00tit.txt"];
pristab00tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab00tit.txt"];
logpristab00 = N[Log[pristab00tit]];

datotab98tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab98tit.txt"];
pristab98tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab98tit.txt"];
logpristab98 = N[Log[pristab98tit]];

datotab96tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab96tit.txt"];
pristab96tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab96tit.txt"];
logpristab96 = N[Log[pristab96tit]];

datotab94tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab94tit.txt"];
pristab94tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab94tit.txt"];
logpristab94 = N[Log[pristab94tit]];

datotab92tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab92tit.txt"];
pristab92tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab92tit.txt"];
logpristab92 = N[Log[pristab92tit]];

datotab90tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\datotab90tit.txt"];
pristab90tit = ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\pristab90tit.txt"];
logpristab90 = N[Log[pristab90tit]];
```

S-series
2014

Alternative A.

```
returnsi = returnsj = 0;
```

```
i = 796;
j = 806;
snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab14tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab14[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab14tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab14[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi14 = returnsi;
newseriesi14 = newseriesi;
newseriesj14 = newseriesj;

Length[newseriesi]

Pospos1j14 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j14 = Position[newseriesj, -1];
Posneg1i14 = Drop[Posneg1j14 + 1, -1] (*the next day, i-series*)
Pospos1i14 = Pospos1j14 + 1 ;

nextdaypos1i14 = Extract[returnsi14, Pospos1i14];
nextdayneg1i14 = Extract[returnsi14, Posneg1i14];
meanA114 = Mean[nextdaypos1i14]
meanA214 = Mean[nextdayneg1i14]

boot14 = {};
Do[
 sample14 = RandomChoice[returnsi14, Length[nextdaypos1i14]] ;
 mean = Mean[sample14];
 boot14 = Append[boot14, mean];
 , {k, 1, 5000}]
hist14 = Histogram[boot14];
plA114 = ContourPlot[x == Mean[nextdaypos1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14, plA114]
pvalueA114 = CDF[SmoothKernelDistribution[boot14]][Mean[nextdaypos1i14]]

boot14neg = {};
Do[
 sample14 = RandomChoice[returnsi14, Length[nextdayneg1i14]] ;
 mean = Mean[sample14];
 boot14neg = Append[boot14neg, mean];
 , {k, 1, 5000}]
hist14neg = Histogram[boot14neg];
plA214 = ContourPlot[x == Mean[nextdayneg1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14neg, plA214]
pvalueA214 = 1 - CDF[SmoothKernelDistribution[boot14neg]][Mean[nextdayneg1i14]]
```

Alternative B.

```
returnsi = returnsj = 0;
```

```
i = 658;
j = 621;
snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab14tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab14[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab14tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab14[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 0.5 * sdi, newseriesi[[i + 1]] = 1]; (*test different fractions*)
 If[returnsi[[i + 1]] - returnsi[[i]] < -0.5 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 0.5 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -0.5 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi14 = returnsi;
newseriesi14 = newseriesi;
newseriesj14 = newseriesj;

Pospos1j14 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j14 = Position[newseriesj, -1];
Posneg1i14 = Drop[Posneg1j14 + 1, -1] (*the next day, i-series*)
Pospos1i14 = Pospos1j14 + 1

nextdaypos1i14 = Extract[returnsi14, Pospos1i14];
nextdayneg1i14 = Extract[returnsi14, Posneg1i14];
meanB114 = Mean[nextdaypos1i14]
meanB214 = Mean[nextdayneg1i14]

boot14 = {};
Do[
 sample14 = RandomChoice[returnsi14, Length[nextdaypos1i14]] ;
 mean = Mean[sample14];
 boot14 = Append[boot14, mean];
 , {k, 1, 5000}]
hist14 = Histogram[boot14];
plB114 = ContourPlot[x == Mean[nextdaypos1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14, plB114]
pvalueB114 = 1 - CDF[SmoothKernelDistribution[boot14]][Mean[nextdaypos1i14]]

boot14neg = {};
Do[
 sample14 = RandomChoice[returnsi14, Length[nextdayneg1i14]] ;
 mean = Mean[sample14];
 boot14neg = Append[boot14neg, mean];
 , {k, 1, 5000}]
hist14neg = Histogram[boot14neg];
plB214 = ContourPlot[x == Mean[nextdayneg1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14neg, plB214]
pvalueB214 = 1 - CDF[SmoothKernelDistribution[boot14neg]][Mean[nextdayneg1i14]]
```

Alternative C.

```
i = 465;
j = 727;
snitt = Intersection[datotab14tit[[i]], datotab14tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab14tit[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab14[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab14tit[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab14[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi14 = returnsi;
newseriesi14 = newseriesi;
newseriesj14 = newseriesj;

Pospos1j14 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j14 = Position[newseriesj, -1];
Posneg1i14 = Posneg1j14 + 1 (*the next day, i-series*)
Pospos1i14 = Pospos1j14 + 1

nextdaypos1i14 = Extract[returnsi14, Pospos1i14];
nextdayneg1i14 = Extract[returnsi14, Posneg1i14];
meanC114 = Mean[nextdaypos1i14]
meanC214 =  Mean[nextdayneg1i14]

boot14 = {};
Do[
 sample14  = RandomChoice[returnsi14, Length[nextdaypos1i14]] ;
 mean  = Mean[sample14];
 boot14 = Append[boot14, mean];
 , {k, 1, 5000}]
hist14 = Histogram[boot14];
plC114 = ContourPlot[x == Mean[nextdaypos1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14, plC114]
pvalueC114 = 1 - CDF[SmoothKernelDistribution[boot14]][Mean[nextdaypos1i14]]

boot14neg = {};
Do[
 sample14  = RandomChoice[returnsi14, Length[nextdayneg1i14]] ;
 mean  = Mean[sample14];
 boot14neg = Append[boot14neg, mean];
 , {k, 1, 5000}]
hist14neg = Histogram[boot14neg];
plC214 = ContourPlot[x == Mean[nextdayneg1i14], {x, -1, 1}, {y, 0, 1000}];
Show[hist14neg, plC214]
pvalueC214 = 1 - CDF[SmoothKernelDistribution[boot14neg]][Mean[nextdayneg1i14]]
```

2012.
Alternative A.

```
returnsi = returnsj = 0;

i = 796;
j = 806;
snitt = Intersection[datotab12tit[[i]], datotab12tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab12tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab12[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab12tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab12[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi12 = returnsi;
newseriesi12 = newseriesi;
newseriesj12 = newseriesj;

Pospos1j12 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j12 = Position[newseriesj, -1];
Posneg1i12 = Posneg1j12 + 1 (*the next day, i-series*)
Pospos1i12 = Drop[Pospos1j12 + 1, -1]

nextdaypos1i12 = Extract[returnsi12, Pospos1i12];
nextdayneg1i12 = Extract[returnsi12, Posneg1i12];
meanA112 = Mean[nextdaypos1i12]
meanA212 = Mean[nextdayneg1i12]

boot12 = {};
Do[
 sample12 = RandomChoice[returnsi12, Length[nextdaypos1i12]] ;
 mean = Mean[sample12];
 boot12 = Append[boot12, mean];
 , {k, 1, 5000}]
hist12 = Histogram[boot12];
plA112 = ContourPlot[x == Mean[nextdaypos1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12, plA112]
pvalueA112 = CDF[SmoothKernelDistribution[boot12]][Mean[nextdaypos1i12]]

boot12neg = {};
Do[
 sample12 = RandomChoice[returnsi12, Length[nextdayneg1i12]] ;
 mean = Mean[sample12];
 boot12neg = Append[boot12neg, mean];
 , {k, 1, 5000}]
hist12neg = Histogram[boot12neg];
plA212 = ContourPlot[x == Mean[nextdayneg1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12neg, plA212]
pvalueA212 = 1 - CDF[SmoothKernelDistribution[boot12neg]][Mean[nextdayneg1i12]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 796;
j = 806;
snitt = Intersection[datotab12tit[[i]], datotab12tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab12tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab12[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab12tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab12[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 0.5 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -0.5 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 0.5 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -0.5 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi12 = returnsi;
newseriesi12 = newseriesi;
newseriesj12 = newseriesj;

Pospos1j12 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j12 = Position[newseriesj, -1];
Posneg1i12 = Posneg1j12 + 1 (*the next day, i-series*)
Pospos1i12 = Drop[Pospos1j12 + 1 , -1]

nextdaypos1i12 = Extract[returnsi12, Pospos1i12];
nextdayneg1i12 = Extract[returnsi12, Posneg1i12];
meanB112 = Mean[nextdaypos1i12]
meanB212 =  Mean[nextdayneg1i12]

boot12 = {};
Do[
 sample12  = RandomChoice[returnsi12, Length[nextdaypos1i12]] ;
 mean  =  Mean[sample12];
 boot12 = Append[boot12, mean];
 , {k, 1, 5000}]
hist12 = Histogram[boot12];
plB112 = ContourPlot[x == Mean[nextdaypos1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12, plB112]
pvalueB112 = CDF[SmoothKernelDistribution[boot12]][Mean[nextdaypos1i12]]

boot12neg = {};
Do[
 sample12  = RandomChoice[returnsi12, Length[nextdayneg1i12]] ;
 mean  =  Mean[sample12];
 boot12neg = Append[boot12neg, mean];
 , {k, 1, 5000}]
hist12neg = Histogram[boot12neg];
plB212 = ContourPlot[x == Mean[nextdayneg1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12neg, plB212]
pvalueB212 = 1 - CDF[SmoothKernelDistribution[boot12neg]][Mean[nextdayneg1i12]]
```

Alternative C.

```
i = 796;
j = 806;
snitt = Intersection[datotab12tit[[i]], datotab12tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab12tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab12[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab12tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab12[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi12 = returnsi;
newseriesi12 = newseriesi;
newseriesj12 = newseriesj;

Pospos1j12 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j12 = Position[newseriesj, -1];
Posneg1i12 = Posneg1j12 + 1 (*the next day, i-series*)
Pospos1i12 = Drop[Pospos1j12 + 1 , -1]

nextdaypos1i12 = Extract[returnsi12, Pospos1i12];
nextdayneg1i12 = Extract[returnsi12, Posneg1i12];
meanC112 = Mean[nextdaypos1i12]
meanC212 = Mean[nextdayneg1i12]

boot12 = {};
Do[
 sample12 = RandomChoice[returnsi12, Length[nextdaypos1i12]] ;
 mean = Mean[sample12];
 boot12 = Append[boot12, mean];
 , {k, 1, 5000}]
hist12 = Histogram[boot12];
plC112 = ContourPlot[x == Mean[nextdaypos1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12, plC112]
pvalueC112 = CDF[SmoothKernelDistribution[boot12]][Mean[nextdaypos1i12]]

boot12neg = {};
Do[
 sample12 = RandomChoice[returnsi12, Length[nextdayneg1i12]] ;
 mean = Mean[sample12];
 boot12neg = Append[boot12neg, mean];
 , {k, 1, 5000}]
hist12neg = Histogram[boot12neg];
plC212 = ContourPlot[x == Mean[nextdayneg1i12], {x, -1, 1}, {y, 0, 1000}];
Show[hist12neg, plC212]
pvalueC212 = 1 - CDF[SmoothKernelDistribution[boot12neg]][Mean[nextdayneg1i12]]
```

2010-2012
Alternative A.

```
returnsi = returnsj = 0;

i = 796;
j = 806;
snitt = Intersection[datotab10tit[[i]], datotab10tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab10tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab10[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab10tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab10[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi10 = returnsi;
newseriesi10 = newseriesi;
newseriesj10 = newseriesj;

Pospos1j10 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j10 = Position[newseriesj, -1];
Posneg1i10 = Posneg1j10 + 1(*the next day, i-series*)
Pospos1i10 = Drop[Pospos1j10 + 1 , -1]

nextdaypos1i10 = Extract[returnsi10, Pospos1i10];
nextdayneg1i10 = Extract[returnsi10, Posneg1i10];
meanA110 = Mean[nextdaypos1i10]
meanA210 =  Mean[nextdayneg1i10]

boot10 = {};
Do[
 sample10 = RandomChoice[returnsi10, Length[nextdaypos1i10]] ;
 mean = Mean[sample10];
 boot10 = Append[boot10, mean];
 , {k, 1, 5000}]
hist10 = Histogram[boot10];
plA110 = ContourPlot[x == Mean[nextdaypos1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10, plA110]
pvalueA110 = CDF[SmoothKernelDistribution[boot10]][Mean[nextdaypos1i10]]

boot10neg = {};
Do[
 sample10 = RandomChoice[returnsi10, Length[nextdayneg1i10]] ;
 mean = Mean[sample10];
 boot10neg = Append[boot10neg, mean];
 , {k, 1, 5000}]
hist10neg = Histogram[boot10neg];
plA210 = ContourPlot[x == Mean[nextdayneg1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10neg, plA210]
pvalueA210 = 1 - CDF[SmoothKernelDistribution[boot10neg]][Mean[nextdayneg1i10]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 796;
j = 806;
snitt = Intersection[datotab10tit[[i]], datotab10tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab10tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab10[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab10tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab10[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 0.5 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -0.5 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 0.5 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -0.5 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi10 = returnsi;
newseriesi10 = newseriesi;
newseriesj10 = newseriesj;

Pospos1j10 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j10 = Position[newseriesj, -1];
Posneg1i10 = Posneg1j10 + 1(*the next day, i-series*)
Pospos1i10 = Pospos1j10 + 1

nextdaypos1i10 = Extract[returnsi10, Pospos1i10];
nextdayneg1i10 = Extract[returnsi10, Posneg1i10];
meanB110 = Mean[nextdaypos1i10]
meanB210 =  Mean[nextdayneg1i10]

boot10 = {};
Do[
 sample10  = RandomChoice[returnsi10, Length[nextdaypos1i10]] ;
 mean  =  Mean[sample10];
 boot10 = Append[boot10, mean];
 , {k, 1, 5000}]
hist10 = Histogram[boot10];
plB110 = ContourPlot[x == Mean[nextdaypos1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10, plB110]
pvalueB110 = CDF[SmoothKernelDistribution[boot10]][Mean[nextdaypos1i10]]

boot10neg = {};
Do[
 sample10  = RandomChoice[returnsi10, Length[nextdayneg1i10]] ;
 mean  =  Mean[sample10];
 boot10neg = Append[boot10neg, mean];
 , {k, 1, 5000}]
hist10neg = Histogram[boot10neg];
plB210 = ContourPlot[x == Mean[nextdayneg1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10neg, plB210]
pvalueB210 = CDF[SmoothKernelDistribution[boot10neg]][Mean[nextdayneg1i10]]
```

Alternative C.

```
i = 727;
j = 465;
snitt = Intersection[datotab10tit[[i]], datotab10tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab10tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab10[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab10tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab10[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi10 = returnsi;
newseriesi10 = newseriesi;
newseriesj10 = newseriesj;

Pospos1j10 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j10 = Position[newseriesj, -1];
Posneg1i10 = Posneg1j10 + 1 (*the next day, i-series*)
Pospos1i10 = Pospos1j10 + 1

nextdaypos1i10 = Extract[returnsi10, Pospos1i10];
nextdayneg1i10 = Extract[returnsi10, Posneg1i10];
meanC110 = Mean[nextdaypos1i10]
meanC210 =  Mean[nextdayneg1i10]

boot10 = {};
Do[
 sample10 = RandomChoice[returnsi10, Length[nextdaypos1i10]] ;
 mean = Mean[sample10];
 boot10 = Append[boot10, mean];
 , {k, 1, 5000}]
hist10 = Histogram[boot10];
plC110 = ContourPlot[x == Mean[nextdaypos1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10, plC110]
pvalueC110 = CDF[SmoothKernelDistribution[boot10]][Mean[nextdaypos1i10]]

boot10neg = {};
Do[
 sample10 = RandomChoice[returnsi10, Length[nextdayneg1i10]] ;
 mean = Mean[sample10];
 boot10neg = Append[boot10neg, mean];
 , {k, 1, 5000}]
hist10neg = Histogram[boot10neg];
plC210 = ContourPlot[x == Mean[nextdayneg1i10], {x, -1, 1}, {y, 0, 1000}];
Show[hist10neg, plC210]
pvalueC210 = CDF[SmoothKernelDistribution[boot10neg]][Mean[nextdayneg1i10]]
```

2008-2010
Alternative A.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab08tit[[i]], datotab08tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab08tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab08[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab08tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab08[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi08 = returnsi;
newseriesi08 = newseriesi;
newseriesj08 = newseriesj;

Length[newseriesi]

Mean[returnsi08]

Pospos1j08 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j08 = Position[newseriesj, -1];
Posneg1i08 = Drop[Posneg1j08 + 1 , -1](*the next day, i-series*)
Pospos1i08 = Pospos1j08 + 1

nextdaypos1i08 = Extract[returnsi08, Pospos1i08];
nextdayneg1i08 = Extract[returnsi08, Posneg1i08];
meanA108 = Mean[nextdaypos1i08]
meanA208 =  Mean[nextdayneg1i08]

boot08 = {};
Do[
 sample08  = RandomChoice[returnsi08, Length[nextdaypos1i08]] ;
 mean  = Mean[sample08];
 boot08 = Append[boot08, mean];
 , {k, 1, 5000}]
hist08 = Histogram[boot08];
plA108 = ContourPlot[x == Mean[nextdaypos1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08, plA108]
pvalueA108 = 1 - CDF[SmoothKernelDistribution[boot08]][Mean[nextdaypos1i08]]

boot08neg = {};
Do[
 sample08  = RandomChoice[returnsi08, Length[nextdayneg1i08]] ;
 mean  = Mean[sample08];
 boot08neg = Append[boot08neg, mean];
 , {k, 1, 5000}]
hist08neg = Histogram[boot08neg];
plA208 = ContourPlot[x == Mean[nextdayneg1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08neg, plA208]
pvalueA208 = CDF[SmoothKernelDistribution[boot08neg]][Mean[nextdayneg1i08]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab08tit[[i]], datotab08tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab08tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab08[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab08tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab08[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi08 = returnsi;
newseriesi08 = newseriesi;
newseriesj08 = newseriesj;

Pospos1j08 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j08 = Position[newseriesj, -1];
Posneg1i08 = Posneg1j08 + 1(*the next day, i-series*)
Pospos1i08 = Pospos1j08 + 1

nextdaypos1i08 = Extract[returnsi08, Pospos1i08];
nextdayneg1i08 = Extract[returnsi08, Posneg1i08];
meanB108 = Mean[nextdaypos1i08]
meanB208 =  Mean[nextdayneg1i08]

boot08 = {};
Do[
 sample08  = RandomChoice[returnsi08, Length[nextdaypos1i08]] ;
 mean  = Mean[sample08];
 boot08 = Append[boot08, mean];
 , {k, 1, 5000}]
hist08 = Histogram[boot08];
plB108 = ContourPlot[x == Mean[nextdaypos1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08, plB108]
pvalueB108 = CDF[SmoothKernelDistribution[boot08]][Mean[nextdaypos1i08]]

boot08neg = {};
Do[
 sample08  = RandomChoice[returnsi08, Length[nextdayneg1i08]] ;
 mean  = Mean[sample08];
 boot08neg = Append[boot08neg, mean];
 , {k, 1, 5000}]
hist08neg = Histogram[boot08neg];
plB208 = ContourPlot[x == Mean[nextdayneg1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08neg, plB208]
pvalueB208 = CDF[SmoothKernelDistribution[boot08neg]][Mean[nextdayneg1i08]]
```

Alternative C.

```
i = 14;
j = 796;
snitt = Intersection[datotab08tit[[i]], datotab08tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab08tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab08[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab08tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab08[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi08 = returnsi;
newseriesi08 = newseriesi;
newseriesj08 = newseriesj;

Pospos1j08 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j08 = Position[newseriesj, -1];
Posneg1i08 = Posneg1j08 + 1 (*the next day, i-series*)
Pospos1i08 = Pospos1j08 + 1

nextdaypos1i08 = Extract[returnsi08, Pospos1i08];
nextdayneg1i08 = Extract[returnsi08, Posneg1i08];
meanC108 = Mean[nextdaypos1i08]
meanC208 = Mean[nextdayneg1i08]

boot08 = {};
Do[
 sample08 = RandomChoice[returnsi08, Length[nextdaypos1i08]] ;
 mean = Mean[sample08];
 boot08 = Append[boot08, mean];
 , {k, 1, 5000}]
hist08 = Histogram[boot08];
plC108 = ContourPlot[x == Mean[nextdaypos1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08, plC108]
pvalueC108 = CDF[SmoothKernelDistribution[boot08]][Mean[nextdaypos1i08]]

boot08neg = {};
Do[
 sample08 = RandomChoice[returnsi08, Length[nextdayneg1i08]] ;
 mean = Mean[sample08];
 boot08neg = Append[boot08neg, mean];
 , {k, 1, 5000}]
hist08neg = Histogram[boot08neg];
plC208 = ContourPlot[x == Mean[nextdayneg1i08], {x, -1, 1}, {y, 0, 1000}];
Show[hist08neg, plC208]
pvalueC208 = 1 - CDF[SmoothKernelDistribution[boot08neg]][Mean[nextdayneg1i08]]
```

2006-2008
Alternative A.

```
returnsi = returnsj = 0;

i = 658;
j = 621;
snitt = Intersection[datotab06tit[[i]], datotab06tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab06tit[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab06[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab06tit[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab06[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi06 = returnsi;
newseriesi06 = newseriesi;
newseriesj06 = newseriesj;

Mean[returnsi06]

Pospos1j06 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j06 = Position[newseriesj, -1];
Posneg1i06 = Posneg1j06 + 1 (*the next day, i-series*)
Pospos1i06 = Drop[Pospos1j06 + 1 , -1]

nextdaypos1i06 = Extract[returnsi06, Pospos1i06];
nextdayneg1i06 = Extract[returnsi06, Posneg1i06];
meanA106 = Mean[nextdaypos1i06]
meanA206 =  Mean[nextdayneg1i06]

boot06 = {};
Do[
 sample06  = RandomChoice[returnsi06, Length[nextdaypos1i06]] ;
 mean  = Mean[sample06];
 boot06 = Append[boot06, mean];
 , {k, 1, 5000}]
hist06 = Histogram[boot06];
plA106 = ContourPlot[x == Mean[nextdaypos1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06, plA106]
pvalueA106 = CDF[SmoothKernelDistribution[boot06]][Mean[nextdaypos1i06]]

boot06neg = {};
Do[
 sample06  = RandomChoice[returnsi06, Length[nextdayneg1i06]] ;
 mean  = Mean[sample06];
 boot06neg = Append[boot06neg, mean];
 , {k, 1, 5000}]
hist06neg = Histogram[boot06neg];
plA206 = ContourPlot[x == Mean[nextdayneg1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06neg, plA206]
pvalueA206 = 1 - CDF[SmoothKernelDistribution[boot06neg]][Mean[nextdayneg1i06]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 658;
j = 621;
snitt = Intersection[datotab06tit[[i]], datotab06tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab06tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab06[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab06tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab06[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi06 = returnsi;
newseriesi06 = newseriesi;
newseriesj06 = newseriesj;

Pospos1j06 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j06 = Position[newseriesj, -1];
Posneg1i06 = Posneg1j06 + 1(*the next day, i-series*)
Pospos1i06 = Drop[Pospos1j06 + 1 , -1]

nextdaypos1i06 = Extract[returnsi06, Pospos1i06];
nextdayneg1i06 = Extract[returnsi06, Posneg1i06];
meanB106 = Mean[nextdaypos1i06]
meanB206 =  Mean[nextdayneg1i06]

boot06 = {};
Do[
 sample06  = RandomChoice[returnsi06, Length[nextdaypos1i06]] ;
 mean  = Mean[sample06];
 boot06 = Append[boot06, mean];
 , {k, 1, 5000}]
hist06 = Histogram[boot06];
plB106 = ContourPlot[x == Mean[nextdaypos1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06, plB106]
pvalueB106 = CDF[SmoothKernelDistribution[boot06]][Mean[nextdaypos1i06]]

boot06neg = {};
Do[
 sample06  = RandomChoice[returnsi06, Length[nextdayneg1i06]] ;
 mean  = Mean[sample06];
 boot06neg = Append[boot06neg, mean];
 , {k, 1, 5000}]
hist06neg = Histogram[boot06neg];
plB206 = ContourPlot[x == Mean[nextdayneg1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06neg, plB206]
pvalueB206 = 1 - CDF[SmoothKernelDistribution[boot06neg]][Mean[nextdayneg1i06]]
```

Alternative C.

```
i = 658;
j = 621;
snitt = Intersection[datotab06tit[[i]], datotab06tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab06tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab06[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab06tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab06[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi06 = returnsi;
newseriesi06 = newseriesi;
newseriesj06 = newseriesj;

Pospos1j06 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j06 = Position[newseriesj, -1];
Posneg1i06 = Posneg1j06 + 1 (*the next day, i-series*)
Pospos1i06 = Pospos1j06 + 1

nextdaypos1i06 = Extract[returnsi06, Pospos1i06];
nextdayneg1i06 = Extract[returnsi06, Posneg1i06];
meanC106 = Mean[nextdaypos1i06]
meanC206 = Mean[nextdayneg1i06]

boot06 = {};
Do[
 sample06 = RandomChoice[returnsi06, Length[nextdaypos1i06]] ;
 mean = Mean[sample06];
 boot06 = Append[boot06, mean];
 , {k, 1, 5000}]
hist06 = Histogram[boot06];
plC106 = ContourPlot[x == Mean[nextdaypos1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06, plC106]
pvalueC106 = CDF[SmoothKernelDistribution[boot06]][Mean[nextdaypos1i06]]

boot06neg = {};
Do[
 sample06 = RandomChoice[returnsi06, Length[nextdayneg1i06]] ;
 mean = Mean[sample06];
 boot06neg = Append[boot06neg, mean];
 , {k, 1, 5000}]
hist06neg = Histogram[boot06neg];
plC206 = ContourPlot[x == Mean[nextdayneg1i06], {x, -1, 1}, {y, 0, 1000}];
Show[hist06neg, plC206]
pvalueC206 = CDF[SmoothKernelDistribution[boot06neg]][Mean[nextdayneg1i06]]
```

2004-06
Alternative A.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab04tit[[i]], datotab04tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab04tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab04[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab04tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab04[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi04 = returnsi;
newseriesi04 = newseriesi;
newseriesj04 = newseriesj;

Mean[returnsi04]

Pospos1j04 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j04 = Position[newseriesj, -1];
Posneg1i04 = Drop[Posneg1j04 + 1 , -1] (*the next day, i-series*)
Pospos1i04 = Pospos1j04 + 1

nextdaypos1i04 = Extract[returnsi04, Pospos1i04];
nextdayneg1i04 = Extract[returnsi04, Posneg1i04];
meanA104 = Mean[nextdaypos1i04]
meanA204 =  Mean[nextdayneg1i04]

boot04 = {};
Do[
 sample04  = RandomChoice[returnsi04, Length[nextdaypos1i04]] ;
 mean  = Mean[sample04];
 boot04 = Append[boot04, mean];
 , {k, 1, 5000}]
hist04 = Histogram[boot04];
plA104 = ContourPlot[x == Mean[nextdaypos1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04, plA104]
pvalueA104 = 1 - CDF[SmoothKernelDistribution[boot04]][Mean[nextdaypos1i04]]

boot04neg = {};
Do[
 sample04  = RandomChoice[returnsi04, Length[nextdayneg1i04]] ;
 mean  = Mean[sample04];
 boot04neg = Append[boot04neg, mean];
 , {k, 1, 5000}]
hist04neg = Histogram[boot04neg];
plA204 = ContourPlot[x == Mean[nextdayneg1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04neg, plA204]
pvalueA204 = CDF[SmoothKernelDistribution[boot04neg]][Mean[nextdayneg1i04]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab04tit[[i]], datotab04tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab04tit[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab04[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab04tit[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab04[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi04 = returnsi;
newseriesi04 = newseriesi;
newseriesj04 = newseriesj;

Pospos1j04 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j04 = Position[newseriesj, -1];
Posneg1i04 = Posneg1j04 + 1 (*the next day, i-series*)
Pospos1i04 = Pospos1j04 + 1

nextdaypos1i04 = Extract[returnsi04, Pospos1i04];
nextdayneg1i04 = Extract[returnsi04, Posneg1i04];
meanB104 = Mean[nextdaypos1i04]
meanB204 =  Mean[nextdayneg1i04]

boot04 = {};
Do[
 sample04 = RandomChoice[returnsi04, Length[nextdaypos1i04]] ;
 mean = Mean[sample04];
 boot04 = Append[boot04, mean];
 , {k, 1, 5000}]
hist04 = Histogram[boot04];
plB104 = ContourPlot[x == Mean[nextdaypos1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04, plB104]
pvalueB104 = CDF[SmoothKernelDistribution[boot04]][Mean[nextdaypos1i04]]

boot04neg = {};
Do[
 sample04 = RandomChoice[returnsi04, Length[nextdayneg1i04]] ;
 mean = Mean[sample04];
 boot04neg = Append[boot04neg, mean];
 , {k, 1, 5000}]
hist04neg = Histogram[boot04neg];
plB204 = ContourPlot[x == Mean[nextdayneg1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04neg, plB204]
pvalueB204 = 1 - CDF[SmoothKernelDistribution[boot04neg]][Mean[nextdayneg1i04]]
```

Alternative C.

```
i = 727;
j = 465;
snitt = Intersection[datotab04tit[[i]], datotab04tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab04tit[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab04[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab04tit[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab04[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi04 = returnsi;
newseriesi04 = newseriesi;
newseriesj04 = newseriesj;

Pospos1j04 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j04 = Position[newseriesj, -1];
Posneg1i04 = Posneg1j04 + 1 (*the next day, i-series*)
Pospos1i04 = Pospos1j04 + 1

nextdaypos1i04 = Extract[returnsi04, Pospos1i04];
nextdayneg1i04 = Extract[returnsi04, Posneg1i04];
meanC104 = Mean[nextdaypos1i04]
meanC204 = Mean[nextdayneg1i04]

boot04 = {};
Do[
 sample04 = RandomChoice[returnsi04, Length[nextdaypos1i04]] ;
 mean = Mean[sample04];
 boot04 = Append[boot04, mean];
 , {k, 1, 5000}]
hist04 = Histogram[boot04];
plC104 = ContourPlot[x == Mean[nextdaypos1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04, plC104]
pvalueC104 = CDF[SmoothKernelDistribution[boot04]][Mean[nextdaypos1i04]]

boot04neg = {};
Do[
 sample04 = RandomChoice[returnsi04, Length[nextdayneg1i04]] ;
 mean = Mean[sample04];
 boot04neg = Append[boot04neg, mean];
 , {k, 1, 5000}]
hist04neg = Histogram[boot04neg];
plC204 = ContourPlot[x == Mean[nextdayneg1i04], {x, -1, 1}, {y, 0, 1000}];
Show[hist04neg, plC204]
pvalueC204 = CDF[SmoothKernelDistribution[boot04neg]][Mean[nextdayneg1i04]]
```

02-04
Alternative A.

```
returnsi = returnsj = 0; *

i = 727;
j = 465;
snitt = Intersection[datotab02tit[[i]], datotab02tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab02tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab02[[i]], Posi]; (*extract the values in these position for stock i*)
Posj = Position[datotab02tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab02[[j]], Posj]; (*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi02 = returnsi;
newseriesi02 = newseriesi;
newseriesj02 = newseriesj;

Mean[returnsi02]

Pospos1j02 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j02 = Position[newseriesj, -1];
Posneg1i02 = Drop[Posneg1j02 + 1 , -1] (*the next day, i-series*)
Pospos1i02 = Pospos1j02 + 1

nextdaypos1i02 = Extract[returnsi02, Pospos1i02];
nextdayneg1i02 = Extract[returnsi02, Posneg1i02];
meanA102 = Mean[nextdaypos1i02]
meanA202 =  Mean[nextdayneg1i02]

boot02 = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdaypos1i02]] ;
 mean  = Mean[sample02];
 boot02 = Append[boot02, mean];
 , {k, 1, 5000}]
hist02 = Histogram[boot02];
plA102 = ContourPlot[x == Mean[nextdaypos1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02, plA102]
pvalueA102 = CDF[SmoothKernelDistribution[boot02]][Mean[nextdaypos1i02]]

boot02neg = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdayneg1i02]] ;
 mean  = Mean[sample02];
 boot02neg = Append[boot02neg, mean];
 , {k, 1, 5000}]
hist02neg = Histogram[boot02neg];
plA202 = ContourPlot[x == Mean[nextdayneg1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02neg, plA202]
pvalueA202 = 1 - CDF[SmoothKernelDistribution[boot02neg]][Mean[nextdayneg1i02]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab02tit[[i]], datotab02tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab02tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab02[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab02tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab02[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi02 = returnsi;
newseriesi02 = newseriesi;
newseriesj02 = newseriesj;

Pospos1j02 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j02 = Position[newseriesj, -1];
Posneg1i02 = Posneg1j02 + 1 (*the next day, i-series*)
Pospos1i02 = Pospos1j02 + 1

nextdaypos1i02 = Extract[returnsi02, Pospos1i02];
nextdayneg1i02 = Extract[returnsi02, Posneg1i02];
meanB102 = Mean[nextdaypos1i02]
meanB202 =  Mean[nextdayneg1i02]

boot02 = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdaypos1i02]] ;
 mean  = Mean[sample02];
 boot02 = Append[boot02, mean];
 , {k, 1, 5000}]
hist02 = Histogram[boot02];
plB102 = ContourPlot[x == Mean[nextdaypos1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02, plB102]
pvalueB102 = CDF[SmoothKernelDistribution[boot02]][Mean[nextdaypos1i02]]

boot02neg = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdayneg1i02]] ;
 mean  = Mean[sample02];
 boot02neg = Append[boot02neg, mean];
 , {k, 1, 5000}]
hist02neg = Histogram[boot02neg];
plB202 = ContourPlot[x == Mean[nextdayneg1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02neg, plB202]
pvalueB202 = CDF[SmoothKernelDistribution[boot02neg]][Mean[nextdayneg1i02]]
```

Alternative C.

```
i = 727;
j = 465;
snitt = Intersection[datotab02tit[[i]], datotab02tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab02tit[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab02[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab02tit[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab02[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi02 = returnsi;
newseriesi02 = newseriesi;
newseriesj02 = newseriesj;

Pospos1j02 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j02 = Position[newseriesj, -1];
Posneg1i02 = Posneg1j02 + 1 (*the next day, i-series*)
Pospos1i02 = Pospos1j02 + 1

nextdaypos1i02 = Extract[returnsi02, Pospos1i02];
nextdayneg1i02 = Extract[returnsi02, Posneg1i02];
meanC102 = Mean[nextdaypos1i02]
meanC202 =  Mean[nextdayneg1i02]

boot02 = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdaypos1i02]] ;
 mean  =  Mean[sample02];
 boot02 = Append[boot02, mean];
 , {k, 1, 5000}]
hist02 = Histogram[boot02];
plC102 = ContourPlot[x == Mean[nextdaypos1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02, plC102]
pvalueC102 = 1 - CDF[SmoothKernelDistribution[boot02]][Mean[nextdaypos1i02]]

boot02neg = {};
Do[
 sample02  = RandomChoice[returnsi02, Length[nextdayneg1i02]] ;
 mean  =  Mean[sample02];
 boot02neg = Append[boot02neg, mean];
 , {k, 1, 5000}]
hist02neg = Histogram[boot02neg];
plC202 = ContourPlot[x == Mean[nextdayneg1i02], {x, -1, 1}, {y, 0, 1000}];
Show[hist02neg, plC202]
pvalueC202 = CDF[SmoothKernelDistribution[boot02neg]][Mean[nextdayneg1i02]]
```

00-02
Alternative A.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab00tit[[i]], datotab00tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab00tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab00[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab00tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab00[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi00 = returnsi;
newseriesi00 = newseriesi;
newseriesj00 = newseriesj;

Mean[returnsi00]

Pospos1j00 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j00 = Position[newseriesj, -1];
Posneg1i00 = Drop[Posneg1j00 + 1 , -1](*the next day, i-series*)
Pospos1i00 = Pospos1j00 + 1

nextdaypos1i00 = Extract[returnsi00, Pospos1i00];
nextdayneg1i00 = Extract[returnsi00, Posneg1i00];
meanA100 = Mean[nextdaypos1i00]
meanA200 =  Mean[nextdayneg1i00]

boot00 = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdaypos1i00]] ;
 mean  = Mean[sample00];
 boot00 = Append[boot00, mean];
 , {k, 1, 5000}]
hist00 = Histogram[boot00];
plA100 = ContourPlot[x == Mean[nextdaypos1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00, plA100]
pvalueA100 = CDF[SmoothKernelDistribution[boot00]][Mean[nextdaypos1i00]]

boot00neg = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdayneg1i00]] ;
 mean  = Mean[sample00];
 boot00neg = Append[boot00neg, mean];
 , {k, 1, 5000}]
hist00neg = Histogram[boot00neg];
plA200 = ContourPlot[x == Mean[nextdayneg1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00neg, plA200]
pvalueA200 = 1 - CDF[SmoothKernelDistribution[boot00neg]][Mean[nextdayneg1i00]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab00tit[[i]], datotab00tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab00tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab00[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab00tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab00[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi00 = returnsi;
newseriesi00 = newseriesi;
newseriesj00 = newseriesj;

Pospos1j00 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j00 = Position[newseriesj, -1];
Posneg1i00 = Drop[Posneg1j00 + 1 , -1] (*the next day, i-series*)
Pospos1i00 = Pospos1j00 + 1

nextdaypos1i00 = Extract[returnsi00, Pospos1i00];
nextdayneg1i00 = Extract[returnsi00, Posneg1i00];
meanB100 = Mean[nextdaypos1i00]
meanB200 =  Mean[nextdayneg1i00]

boot00 = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdaypos1i00]] ;
 mean  = Mean[sample00];
 boot00 = Append[boot00, mean];
 , {k, 1, 5000}]
hist00 = Histogram[boot00];
plB100 = ContourPlot[x == Mean[nextdaypos1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00, plB100]
pvalueB100 = 1 - CDF[SmoothKernelDistribution[boot00]][Mean[nextdaypos1i00]]

boot00neg = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdayneg1i00]] ;
 mean  = Mean[sample00];
 boot00neg = Append[boot00neg, mean];
 , {k, 1, 5000}]
hist00neg = Histogram[boot00neg];
plB200 = ContourPlot[x == Mean[nextdayneg1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00neg, plB200]
pvalueB200 = CDF[SmoothKernelDistribution[boot00neg]][Mean[nextdayneg1i00]]
```

Alternative C.

```
i = 727;
j = 465;
snitt = Intersection[datotab00tit[[i]], datotab00tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab00tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab00[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab00tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab00[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi00 = returnsi;
newseriesi00 = newseriesi;
newseriesj00 = newseriesj;

Pospos1j00 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j00 = Position[newseriesj, -1];
Posneg1i00 = Posneg1j00 + 1 (*the next day, i-series*)
Pospos1i00 = Pospos1j00 + 1

nextdaypos1i00 = Extract[returnsi00, Pospos1i00];
nextdayneg1i00 = Extract[returnsi00, Posneg1i00];
meanC100 = Mean[nextdaypos1i00]
meanC200 =  Mean[nextdayneg1i00]

boot00 = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdaypos1i00]] ;
 mean  =  Mean[sample00];
 boot00 = Append[boot00, mean];
 , {k, 1, 5000}]
hist00 = Histogram[boot00];
plC100 = ContourPlot[x == Mean[nextdaypos1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00, plC100]
pvalueC100 = CDF[SmoothKernelDistribution[boot00]][Mean[nextdaypos1i00]]

boot00neg = {};
Do[
 sample00  = RandomChoice[returnsi00, Length[nextdayneg1i00]] ;
 mean  =  Mean[sample00];
 boot00neg = Append[boot00neg, mean];
 , {k, 1, 5000}]
hist00neg = Histogram[boot00neg];
plC200 = ContourPlot[x == Mean[nextdayneg1i00], {x, -1, 1}, {y, 0, 1000}];
Show[hist00neg, plC200]
pvalueC200 = CDF[SmoothKernelDistribution[boot00neg]][Mean[nextdayneg1i00]]
```

98-00
Alternative A.

```
returnsi = returnsj = 0;

i = 727;
j = 465;
snitt = Intersection[datotab98tit[[i]], datotab98tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab98tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab98[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab98tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab98[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi98 = returnsi;
newseriesi98 = newseriesi;
newseriesj98 = newseriesj;

Pospos1j98 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j98 = Position[newseriesj, -1];
Posneg1i98 = Posneg1j98 + 1 (*the next day, i-series*)
Pospos1i98 = Pospos1j98 + 1

nextdaypos1i98 = Extract[returnsi98, Pospos1i98];
nextdayneg1i98 = Extract[returnsi98, Posneg1i98];
meanA198 = Mean[nextdaypos1i98]
meanA298 =  Mean[nextdayneg1i98]

boot98 = {};
Do[
 sample98 = RandomChoice[returnsi98, Length[nextdaypos1i98]] ;
 mean = Mean[sample98];
 boot98 = Append[boot98, mean];
 , {k, 1, 5000}]
hist98 = Histogram[boot98];
plA198 = ContourPlot[x == Mean[nextdaypos1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98, plA198]
pvalueA198 = 1 - CDF[SmoothKernelDistribution[boot98]][Mean[nextdaypos1i98]]

boot98neg = {};
Do[
 sample98 = RandomChoice[returnsi98, Length[nextdayneg1i98]] ;
 mean = Mean[sample98];
 boot98neg = Append[boot98neg, mean];
 , {k, 1, 5000}]
hist98neg = Histogram[boot98neg];
plA298 = ContourPlot[x == Mean[nextdayneg1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98neg, plA298]
pvalueA298 = CDF[SmoothKernelDistribution[boot98neg]][Mean[nextdayneg1i98]]
```

Alternative B.

```
returnsi = returnsj = 0;

i = 658;
j = 621;
snitt = Intersection[datotab98tit[[i]], datotab98tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab98tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab98[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab98tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab98[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi98 = returnsi;
newseriesi98 = newseriesi;
newseriesj98 = newseriesj;

Pospos1j98 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j98 = Position[newseriesj, -1];
Posneg1i98 = Drop[Posneg1j98 + 1, -1] (*the next day, i-series*)
Pospos1i98 = Pospos1j98 + 1

nextdaypos1i98 = Extract[returnsi98, Pospos1i98];
nextdayneg1i98 = Extract[returnsi98, Posneg1i98];
meanB198 = Mean[nextdaypos1i98]
meanB298 = Mean[nextdayneg1i98]

boot98 = {};
Do[
 sample98 = RandomChoice[returnsi98, Length[nextdaypos1i98]] ;
 mean = Mean[sample98];
 boot98 = Append[boot98, mean];
 , {k, 1, 5000}]
hist98 = Histogram[boot98];
plB198 = ContourPlot[x == Mean[nextdaypos1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98, plB198]
pvalueB198 = 1 - CDF[SmoothKernelDistribution[boot98]][Mean[nextdaypos1i98]]

boot98neg = {};
Do[
 sample98 = RandomChoice[returnsi98, Length[nextdayneg1i98]] ;
 mean = Mean[sample98];
 boot98neg = Append[boot98neg, mean];
 , {k, 1, 5000}]
hist98neg = Histogram[boot98neg];
plB298 = ContourPlot[x == Mean[nextdayneg1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98neg, plB298]
pvalueB298 = 1 - CDF[SmoothKernelDistribution[boot98neg]][Mean[nextdayneg1i98]]
```

Alternative C.

```
i = 658;
j = 621;
snitt = Intersection[datotab98tit[[i]], datotab98tit[[j]]];
(*for each i and j, find the intersecting dates*)
Posi = Position[datotab98tit[[i]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab98[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datotab98tit[[j]], _ ? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab98[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi]
sdj = StandardDeviation[returnsj]
newseriesi = newseriesj = Table[0, Length[returnsi]];
Do[
 If[returnsi[[i + 1]] - returnsi[[i]] > 2 * sdi, newseriesi[[i + 1]] = 1];
 If[returnsi[[i + 1]] - returnsi[[i]] < -2 * sdi, newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i + 1]] - returnsj[[i]] > 2 * sdj, newseriesj[[i + 1]] = 1];
 If[returnsj[[i + 1]] - returnsj[[i]] < -2 * sdj, newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]

returnsi98 = returnsi;
newseriesi98 = newseriesi;
newseriesj98 = newseriesj;

Pospos1j98 = Position[newseriesj, 1]; (*find the positions of the 1 and -1 in the j-series*)
Posneg1j98 = Position[newseriesj, -1];
Posneg1i98 = Drop[Posneg1j98 + 1 , -1] (*the next day, i-series*)
Pospos1i98 = Pospos1j98 + 1

nextdaypos1i98 = Extract[returnsi98, Pospos1i98];
nextdayneg1i98 = Extract[returnsi98, Posneg1i98];
meanC198 = Mean[nextdaypos1i98]
meanC298 =  Mean[nextdayneg1i98]

boot98 = {};
Do[
 sample98  = RandomChoice[returnsi98, Length[nextdaypos1i98]] ;
 mean  = Mean[sample98];
 boot98 = Append[boot98, mean];
 , {k, 1, 5000}]
hist98 = Histogram[boot98];
plC198 = ContourPlot[x == Mean[nextdaypos1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98, plC198]
pvalueC198 = 1 - CDF[SmoothKernelDistribution[boot98]][Mean[nextdaypos1i98]]

boot98neg = {};
Do[
 sample98  = RandomChoice[returnsi98, Length[nextdayneg1i98]] ;
 mean  = Mean[sample98];
 boot98neg = Append[boot98neg, mean];
 , {k, 1, 5000}]
hist98neg = Histogram[boot98neg];
plC298 = ContourPlot[x == Mean[nextdayneg1i98], {x, -1, 1}, {y, 0, 1000}];
Show[hist98neg, plC298]
pvalueC298 = CDF[SmoothKernelDistribution[boot98neg]][Mean[nextdayneg1i98]]
```

Testing period.

```
(*Read in the data from TITLON*)
date15 = Map[StringSplit[#] &,
    Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\selected.txt", String], 1]][[All, 1]];
codes15 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\selected.txt", String],
     1]][[All, 2]];
names15 = Map[StringSplit[#] &, Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\selected.txt", String],
     1]][[All, 3]];
AdjPrice15 = ToExpression[Map[StringSplit[#] &,
     Drop[ReadList["C:\\Users\\marte\\Dropbox\\Thesis\\Titlon\\selected.txt", String], 1]][[All, 4]]];


lengder15 = Map[Length[#] &, Split[codes15]] ;
(*del opp de som har identiske elementer, ta så lengden av sublisten*);
Length[lengder15]; (*16*)
L = FoldList[Plus, 0, lengder15]; (**)
t = 1;
(*sett dem sammen i table, og ta log av prisene*)
pristab15 = Table[
   AdjPrice15[[L[[t]] + 1 ;; L[[t + 1]]]], {t, 1, Length[L] - 1}];
pristab15 = ToExpression[pristab15];
datotab15 = Table[
   date15[[L[[t]] + 1 ;; L[[t + 1]]]], {t, 1, Length[L] - 1}];

lengder15

n = Length[lengder15] ;

startdato = AbsoluteTime["01.07.2015"]
3 597 523 200
pricetab15 = Table[0, {i, 1, n}];
datetab15 = Table[0, {i, 1, n}];
Monitor[Do[abstid =
   Map[AbsoluteTime[#] &, Map[StringJoin[#] &, Map[{StringJoin[#[[2]], "."], StringJoin[#[[1]], "."], #[[3]]} &,
      Map[StringSplit[#, "."] &, datotab15[[i]]]]]];
  (*split strengene der det er punktum, bytt om på dag mnd (europeisk-amerikansk måte),
  sett tilbake punktum, ta absolutt tid*)
  pos = Position[abstid, _ ? (# ≥ startdato &) ];
  datetab15[[i]] = Extract [datotab15[[i]], pos];
  pricetab15[[i]] = Extract[pristab15[[i]], pos];
  , {i, 1, n}], {i}]
logpristab15 = N[Log[pricetab15]];

lengder15

lengder15 = Table["empty", {i, 1, n}];
Do[
 lengder15[[i]] = Length[pricetab15[[i]]]
 , {i, 1, n}]
logpristab15 = N[Log[pricetab15]];

lengder15
```

```
(*Create the correlation matrix for period 15*)
matrix15tit = Table["empty", {i, 1, n}, {j, 1, n}]; (*lag tom matrise*)
Monitor[
 Do[
   snitt = Intersection[datetab15[[i]], datetab15[[j]]]; (*for each i and j, find the intersecting dates*)
   Posi = Position[datetab15[[i]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock i*)
   utvalgi = Extract[logpristab15[[i]], Posi];
   (*extract the values in these position for stock i*)
   Posj = Position[datetab15[[j]], _? (MemberQ[snitt, #] == True &)];
   (*find the corresponding possitions for stock j*)
   utvalgj = Extract[logpristab15[[j]], Posj];
   (*extract the values in these position for stock j*)
   If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
    returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
    returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
    If[Min[Length[returnsi], Length[returnsj]] > 1 && Length[returnsi] == Length[returnsj] &&
      StandardDeviation[Drop[returnsi, 1]] > 0 && StandardDeviation[Drop[returnsj, -1]] > 0, (*if the
      returns series are greater than one, and they have equal length, and their sd is not zero, then..*)
     korrelasjon = Correlation[Drop[returnsi, 1], Drop[returnsj, -1]] ;
     (*take the correlation of the series*)
     matrix15tit[[i, j]] = korrelasjon; (*put the correlation in the appropriate matrix coordinate*)
    ]];
   , {i, 1, n}, {j, 1, n}];
 , {i, j}]
Export["C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_15_tit.txt", matrix15tit]


"C:\\Users\\marte\\Dropbox\\Thesis\\matrix_year_15_tit.txt"

matrix15tit[[i, j]]

namelist15 = Split[names15][[All, 1]][[1 ;; 16]];
```

The Experiment.

Strategy 1 used on shares []

```
(*In this section you need to un-comment parts depending on whether k=1,2,3*)
```

```
i = 6;
j = 5;
snitt = Intersection[datetab15[[i]], datetab15[[j]]]; (*for each i and j, find the intersecting dates*)
Posi = Position[datetab15[[i]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock i*)
utvalgi = Extract[logpristab15[[i]], Posi];(*extract the values in these position for stock i*)
Posj = Position[datetab15[[j]], _? (MemberQ[snitt, #] == True &)];
(*find the corresponding possitions for stock j*)
utvalgj = Extract[logpristab15[[j]], Posj];(*extract the values in these position for stock j*)

If[Min[Length[utvalgi], Length[utvalgj]] > 3, (*if the set is greater than 3, then find the returns*)
  returnsi = Drop[utvalgi, 1] - Drop[utvalgi, -1];
  returnsj = Drop[utvalgj, 1] - Drop[utvalgj, -1];
 ];
sdi = StandardDeviation[returnsi];
sdj = StandardDeviation[returnsj];
newseriesi = newseriesj = Table[0, Length[returnsi]];
(*Un-comment when k=2 and add "2*sdi" when k=3*)
(*Do[ (*create a new series with ones/zeros*)
  If[returnsi[[k+1]]-returnsi[[k]]>sdi,newseriesi[[k+1]]=1];
  If[returnsi[[k+1]]-returnsi[[k]]<-sdi,newseriesi[[k+1]]=-1];
  ,{k,1,Length[returnsi]-1}]
 Do[
  If[returnsj[[k+1]]-returnsj[[k]]>sdj,newseriesj[[k+1]]=1];
  If[returnsj[[k+1]]-returnsj[[k]]<-sdj,newseriesj[[k+1]]=-1];
  ,{k,1,Length[returnsj]-1}]*)
Do[
 If[returnsi[[i]] < returnsi[[i + 1]], newseriesi[[i + 1]] = 1];
 If[returnsi[[i]] > returnsi[[i + 1]], newseriesi[[i + 1]] = -1];
 , {i, 1, Length[returnsi] - 1}]
Do[
 If[returnsj[[i]] < returnsj[[i + 1]], newseriesj[[i + 1]] = 1];
 If[returnsj[[i]] > returnsj[[i + 1]], newseriesj[[i + 1]] = -1];
 , {i, 1, Length[returnsj] - 1}]


Pospos1j = Position[newseriesj, 1];
Posneg1j = Position[newseriesj, -1];
Length[snitt]

posbuyi = Posneg1j (*1 day lag → buy/sell i the same day j has a 1 or -1*)
posselli = Pospos1j
```

The experiment/method

```
a = 100000;
b = 0;

buysell = Table[0, Length[newseriesi] + 1];

Do[
 buysell[[i]] = newseriesj[[i]]
 , {i, 1, Length[newseriesj]}]

buysell;(*-1 is buy i and 1 is sell i, when correlation is negative*)

alist = {};
blist = {};
clist = {};
```

```
Monitor[
 Do[
  c = Floor[a/Exp[utvalgi[[i]]]]; (*nr of shares we can afford. utvalgi is logprice*)
  If[buysell[[i]] == -1 && a > 0, (*hvis flere kjøp kommer etter hverandre kjøper vi ikke på nytt*)
    a = a - (1 + 0.0000) * c * Exp[utvalgi[[i]]]; (*kurtasje på 0.05% av handelsummen*)
    b = b + c;
   ]
   If[buysell[[i]] == 1 && b > 0,
    a = a + b * Exp[utvalgi[[i]]];
    b = b - b (*selg alt vi har*)
   ];
  alist = Append[alist, a];
  blist = Append[blist, b];
  clist = Append[clist, c];
  , {i, 1, Length[buysell]}], {a}]

alist

blist

restest = ((a) + b * Exp[utvalgi[[Length[utvalgi]]]]) / 100 000

1.69904

ListPlot[(alist + blist * Exp[utvalgi]) / 100 000, Joined → True, PlotRange → All]

Last[(alist + blist * Exp[utvalgi]) / 100 000]

1.69904

risk = StandardDeviation[(alist + blist * Exp[utvalgi]) / 100 000]

0.367348

ListPlot[Exp[utvalgi] / Exp[utvalgi[[1]]], Joined → True, PlotRange → All]

Last[Exp[utvalgi] / Exp[utvalgi[[1]]]]

risk2 = StandardDeviation[Exp[utvalgi] / Exp[utvalgi[[1]]]] (*holding the share all the time*)

Length[buysell]

Split[buysell];

A = Flatten [Position[buysell, _ ? (# == -1 &)]]

Length[A]

B = Flatten [Drop[Position[buysell, _ ? (# == 1 &)], 0]]

Length[B]
```

```
t = A[[1]];
tliste = {t};
sliste = {s};
bryt = 0;
While[bryt == 0,
  posB = Position[B, _? (# > t &)];
  If[Length [posB] > 0,
   s = B[[First[posB][[1]]]];
   ,
   bryt = 1;
  ];
  posA = Position[A, _? (# > s &)];
  If[Length[posA] > 0,
   t = A[[First[posA][[1]]]];
   ,
   bryt = 1;
  ];
  tliste = Append[tliste, t];
  sliste = Append[sliste, s];
 ];


sliste = Drop[sliste, 1]; (*pos. salg*)
sliste = Drop[sliste, -1];

sliste = Append[sliste, Length[buysell]]

tliste (*pos. kjøp*)

tliste = Drop[tliste, -1]

daysInvest = sliste - tliste (*number of days holding the share*)

Total[daysInvest]

res = ((a) + b * Exp[utvalgi[[Length[utvalgi]]]]) / 100 000

Years = N[Total[daysInvest] / 365] (*how many years is this*)

daily = res ^ (1 / Total[daysInvest]) (*daglig rente*)
```

Bootstrap test for sample mean

Need there to by the same number of actions taken as in my experiment, but the type of action together with the time the action took place, should be random.
"buysell" describes the actions taken in my experiment.

```
actions = Length[buysell] - Count[buysell, 0] (*number of actions taken*)
```

I need a new list of length Length[buysell] with 1's and -1's placed randomly (actions) times. The list should start out as a list of only zeros.

```
buysellrand = Table[0, Length[buysell]];

randposition = Partition[Sort[RandomSample[Range[Length[buysell]], actions]], 1]
(* (action) random numbers between 1 and Length[buysell] --no replacement*)

buysellrand = ReplacePart[buysellrand, randposition → 1]
```

```
bootinterest = {};
test = {}; test2 = {}; test3 = {};

bootinterest = {};
test = {}; test2 = {}; test3 = {};
Do[
 buysellrand = Table[0, Length[buysell]];
 randposition = Partition[Sort[RandomSample[Range[Length[buysell]], actions]], 1];
 buysellrand = ReplacePart[buysellrand, randposition → 1];
 Do[
  If[buysellrand[[i]] == 1 && RandomReal[] > 0.5, buysellrand[[i]] = -1]; (*50% chanse of changing a 1 to a -1*)
  , {i, 1, Length[buysell]}];
 Arand = Flatten [Position[buysellrand, _? (# == -1 &)]];
 Brand = Flatten [Drop[Position[buysellrand, _? (# == 1 &)], 0]];
 t = Arand[[1]];
 tliste = {t};
 sliste = {s};
 bryt = 0;
 While[bryt == 0,
  posB = Position[Brand, _? (# > t &)];
  If[Length [posB] > 0,
   s = Brand[[First[posB][[1]]]];
   ,
   bryt = 1;
  ];
  posA = Position[Arand, _? (# > s &)];
  If[Length[posA] > 0,
   t = Arand[[First[posA][[1]]]];
   ,
   bryt = 1;
  ];
  tliste = Append[tliste, t];
  sliste = Append[sliste, s];
 ];
 investmentdays = {};
 tliste = Drop[tliste, -1];
 sliste = Drop[sliste, 1];
 Do[
  If[Last[sliste] == sliste[[i - 1]], Drop[sliste, -1]];(*av og til skjer dette*)
  , {i, 1, Length[sliste]}];

 daysInvestrand = sliste - tliste;
 investmentdays = Total[daysInvestrand];
 Yearsrand = N[investmentdays/365];
 a = 100000;
 b = 0;

 Do[
  c = Floor[a/Exp[utvalgi[[i]]]] ; (*nr of shares we can afford*)
  If[buysellrand[[i]] == -1 &&  a > 0,
    a = a - (1 + 0.0000) * c * Exp[utvalgi[[i]]]; (*remember to add kurtasje*)
    b = b + c
   ]
   If[buysellrand[[i]] == 1 &&  b > 0,
    a = a + b * Exp[utvalgi[[i]]];
    b = b - b
   ]
  , {i, 1, Length[buysellrand]}];
 money = a + b * Exp[utvalgj[[Length[buysell]]]];
 interest1 = money / 100000;
 interest = interest1^ (1/investmentdays);

 bootinterest = Append[bootinterest, interest1];
 test = Append[test, Yearsrand];
 test2 = Append[test2, money];
 test3 = Append[test3, interest]
 , {k, 1, 10000}]
```

```
hist = Histogram[test3];
pl = ContourPlot[x == daily, {x, 0, 1.7}, {y, 0, 4000}];
Show[hist, pl]
```

```
pvalue = 1 - CDF[SmoothKernelDistribution[test3]][daily]
```

Bootstrap test for SD

```
actions = Length[buysell] - Count[buysell, 0] (*number of actions taken*)
```

I need a new list of length Length[buysell] with 1’s and -1’s placed randomly (actions) times. The list should start out as a list of only zeros.

```
buysellrand = Table[0, Length[buysell]];
```

```
randposition = Partition[Sort[RandomSample[Range[Length[buysell]], actions]], 1]
(* (action) random numbers between 1 and Length[buysell] --no replacement*)
```

```
buysellrand = ReplacePart[buysellrand, randposition → 1]
```

```
bootinterest = {};
sdn = {}; new1 = {}; new = {};

bootinterest = {};
sdn = {}; new1 = {}; new = {};
Do[
 buysellrand = Table[0, Length[buysell]];
 randposition = Partition[Sort[RandomSample[Range[Length[buysell]], actions]], 1];
 buysellrand = ReplacePart[buysellrand, randposition → 1];
 Do[
  If[buysellrand[[i]] == 1 && RandomReal[] > 0.5, buysellrand[[i]] = -1]; (*50% chanse of changing a 1 to a -1*)
  , {i, 1, Length[buysell]}];
 Arand = Flatten [Position[buysellrand, _? (# == -1 &)]];
 Brand = Flatten [Drop[Position[buysellrand, _? (# == 1 &)], 0]];
 t = Arand[[1]];
 tliste = {t};
 sliste = {s};
 bryt = 0;
 While[bryt == 0,
  posB = Position[Brand, _? (# > t &)];
  If[Length [posB] > 0,
   s = Brand[[First[posB][[1]]]];
   ,
   bryt = 1;
  ];
  posA = Position[Arand, _? (# > s &)];
  If[Length[posA] > 0,
   t = Arand[[First[posA][[1]]]];
   ,
   bryt = 1;
  ];
  tliste = Append[tliste, t];
  sliste = Append[sliste, s];
 ];
 investmentdays = {};
 tliste = Drop[tliste, -1];
 sliste = Drop[sliste, 1];
 Do[
  If[Last[sliste] == sliste[[i - 1]], Drop[sliste, -1]];(*av og til skjer dette*)
  , {i, 1, Length[sliste]}];

 daysInvestrand = sliste - tliste;
 investmentdays = Total[daysInvestrand];
 Yearsrand = N[investmentdays / 365];
 a = 100 000;
 b = 0;

 Do[
  c = Floor[a / Exp[utvalgi[[i]]]] ; (*nr of shares we can afford*)
  If[buysellrand[[i]] == -1 &&  a > 0,
    a = a - (1 + 0.0000) * c * Exp[utvalgi[[i]]]; (*remember to add kurtasje*)
    b = b + c
   ]
   If[buysellrand[[i]] == 1 &&  b > 0,
    a = a + b * Exp[utvalgi[[i]]];
    b = b - b
   ];

  new1 = (a + b * Exp[utvalgi[[i]]]) / 100 000;
  new = Append[new, new1]
  , {i, 1, Length[buysellrand]}];

 sdn = Append[sdn, StandardDeviation[new]];

 , {k, 1, 10 000}]


buysellrand;
```

```
pos = Position[buysellrand, _ ? (# != 0 &)];

histSD = Histogram[sdn];

pl = ContourPlot[x == risk, {x, 0, 2}, {y, 0, 150}];
Show[histSD, pl]

pvalue = CDF[SmoothKernelDistribution[sdn]][risk]
```

How good are our results?

```
p = Mean[{0.16, 0.24, 0.37, 0.69, 0.06}]

s = Table[Plus @@ {RandomReal[], RandomReal[], RandomReal[], RandomReal[], RandomReal[]}, {10000}] / 5

CDF[EmpiricalDistribution[s], p]

p = Mean[{0.635, 0.023, 0.033, 0.005, 0.002}]
CDF[EmpiricalDistribution[s], p]
```

UNIVERSITY OF TROMSØ