Pairs Trading with Robust Correlation

by

Jieren Wang

AN ESSAY SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

May 2009

Abstract

This essay compares the performance of two types of correlation measures in triggering trades in a pairs trading application in the presence of high-frequency stock prices. One correlation measure is the commonly-used Pearson correlation and the other is a robust correlation measure called Maronna correlation. These correlation measures are used to define three methods of initiating trades – called trigger mechanisms – and the characteristics of these mechanisms. We test the relative performance of trading strategies using three types of triggering mechanisms on historical data and perform statistical tests based on these results. We find that trading strategies based on trigger mechanisms which employ robust measures of correlation yield consistently lower returns but more favorable risk characteristics.

Table of Contents

Appendices

List of Tables

List of Figures

Acknowledgements

To get any distance in life and career one needs the help and support of others. This is particularly true for me, as I have so many people to thank that helped make this research possible.

First, I want to thank my advisor Dr. Rachel Kuske for great patience and support in the process of writing this essay. She helped me to make connections with industry, and encouraged me to pursue my research interests. Through her support and the financial assistance of the MITACS Accelerate BC program, I conducted the research found in this essay during an internship with Scalable Analytics, Inc. Dr. Kuske is always very thoughtful and considerate, and consequently made my work more enjoyable.

My heartfelt appreciation go to Camilo Rostoker and Alan Wagner at Scalable Analytics, for suggesting this research direction and guiding me through the process of discovery. Some of the work found in this essay is based on a paper co-authored by Mr. Rostoker, Dr. Wagner and myself which will be published at an upcoming conference on parallel computation. I also want to express my thankfulness for the efforts of Dr. Holger Hoos in helping push forward the scope of my work and providing valuable feedback.

During the course of the project we received guidance and advice on the details of trading practice from Dan Clifford, Calvin Winter. Their input added a great deal to our results. Some of their specific contributions are mentioned in footnotes in the essay, but I want to take this opportunity to thank them personally for their expert advice. I also want to thank Huakun Ding, my colleague, who was always willing to share lots of his insights about trading on the stock market.

Another source of great technical advice was Professor Harry Joe, who was always so willing to discuss statistical methods and ideas with me. He is also the one who connected with me Scalable Analytics, Inc. I relied on his kindness and expertise to get me out of a few dead ends, and I am grateful for his efforts.

One more word of thanks goes to the "co-author" of my life and this essay during the time of master's degree, Chris Ryan. He is like a lighthouse in a cold night, guiding me when I am lost, giving me hope when I am down, and encouraging me to move forward all the time. No words can really express the gratitude I have for him. This poem is from him, but it sums up my feelings as well:

> The world is hope -
> Long walks in old shoes,

With garlic-stained fingers
And shared laughs
And shared tears.

The world is love,
Made alive in the spirit.
What an amazing thing to share:
Thanks, thanks, thanks.

Above all, this work is only possible through the loving encouragement of my parents and sister. Without them I would have never been able to pursue the dream of coming to Canada to pursue my higher education and certainly would not have been able to face the challenges I have endured from living in a new country and pursuing studies in a new language and culture. My confidence in their love helps me re-double my efforts every time I face adversity. This essay is a testament to their love.

Dedication

To my family.

Chapter 1

Introduction

Pairs trading is a popular quantitative method of statistical arbitrage that has been widely used in the financial industry for over twenty years [5]. The essence of pairs trading is to exploit pairs of stocks whose movements are related to each other. When the co-movement deteriorates, the strategy is to long the under-performer and short the over-performer, anticipating that the co-movement will recover and gains can be made. If the co-movement does recover, the positions are reversed yielding arbitrage profits from the spread of the two stock prices. Pairs trading algorithms identify which pairs from all stocks in the market to trade, when to enter a position on those stocks and in what proportion, and when to reverse the position to (hopefully) realize profits. Many pairs trading algorithms, and in particular the one discussed in this essay, use correlation as a fundamental ingredient. One challenge that has recently presented itself in modern trading applications is effective computation and use of correlation in the presence of high-frequency data. The trading industry has seen an explosive growth in the use of high performance computation and real-time tracking of transactions in the market. Only in the last few years have researchers taken a keen interest in high-frequency analysis [2, 8]. One reason for this new trend is the availability of such data, for example in the form of the Trade and Quote (TAQ) database product offered by the New York Stock Exchange (NYSE). A major challenge in working with high-frequency data is its sheer volume - a single day's worth of TAQ data typically consume over 50 Gigabytes of disk space! While research using high-frequency data appears to be gaining momentum, studying market-wide intra-day correlation has yet to be explored in depth. The reason is that generating massive amounts of correlation data is computationally expensive and would take an unreasonable amount of time using traditional statistical software. The company I conducted this research with, Scalable Analytics Inc., has addressed this problem by developing a computational engine based on high-performance parallel computation for rapidly generating correlation matrices for tens of thousands of variables. The technology underlying this engine was developed at the Computer Science department at UBC [10], and the inventors, both Associate Professors in the department, are key technical advisors to the company. An overview of the engine, called `MarketMiner`, can be found in an appendix.

Computing correlation in the high-frequency domain can be a difficult undertaking not just because of the volume of data. In addition, raw price data are filled with numerous bad data points. The traditional definition of correlation by Pearson is very sensitive to these outliers [1], and thus in theory it should not be directly applied to raw high-frequency data. Several papers have

shown that *robust correlation measures* can be used to improve performance in popular financial applications when compared to using Pearson's traditional measure, including portfolio allocation and Value-at-Risk [12, 11].

One class of robust measures of correlation is called M-estimators of correlation [9]. One of the core products of Scalable Analytics Inc. is a computational engine which produces a robust M-estimator of correlation which we call Maronna correlation. It can do so in real time. The main work of this essay is to test the relative effectiveness of using alternate measures of traditional and robust correlation in pairs trading applications. In particular, we ask whether strategies that use Maronna correlation to enter a trade are significantly more profitable and/or less risky than trades using Pearson correlation. The added dimension of interest here is that this numerical analysis and comparison occurs in a high-frequency data environment.

The precise definition of the correlation measures we consider can be found in Chapter 2 which contains some basic background material. Details of how to generate correlation data from TAQ data, as well as some basic comparisons between different measures, are contained in Chapter 3. In that chapter we describe the data used in our study: historical TAQ data on 61 commonly traded stocks during March, 2008. The correlation time series that we generate are the input for a canonical pairs trading strategy which we define in Chapter 4. Based on correlations we define alternate *trigger mechanisms* which indicate when to initiate a trade between two pairs of stocks. One trigger mechanism is based solely on Pearson correlation, another on Maronna correlation. A third trigger mechanism called "Combined" is based on a combination of Pearson and Maronna.

To test the effectiveness of different measures of correlation and trigger mechanisms in pairs trading we test our trading algorithms on our set of historical data. The procedure of testing a trading strategy on historical data is called "backtesting" and is a standard activity in the trading industry. Since we are backtesting a strategy based on correlation, we want to compare various correlation measures to determine which one performs better and under what circumstances. To eliminate the potential bias of selecting specific pairs, we take a brute-force approach by backtesting over as many pairs as possible, in this case all pairs from our 61 stocks, to determine the relative performance of the strategy under different correlation measures. Chapter 5 elaborates on how we conducted backtesting in a MATLAB implementation and the challenges we encountered.

Our pairs trading algorithm produces a multitude of "returns data" – data points for each pair trade transaction. To statistically test which pairs trading algorithm performs better we process these raw "returns data" into overall performance measures of profitability and risk characteristics. These performance measures include cumulative monthly returns and more sophisticated measures like maximum drawdown. Precise definitions of these measures as well as details of how to generate them can be found at the end of Chapter 5.

In Chapter 6 we statistically analyze the performance measure data which was generated through backtesting. We perform statistical tests on the means of returns and the risk characteristics of strategies. The results of this study are discussed at the end of Chapter 6. We found

that Pearson-based trading strategies (with a data filter described below) lead to high returns, while Maronna-based trading strategies yielded lower returns and also slightly higher risk comparing with Pearson. Pairs trading algorithms using the Combined trigger mechanism had more attractive risk characteristics than both pure Maronna-based and Pearson-based trading strategies. Implications for pairs trading practice are further discussed in Chapter 7.

In the course of conducting this study, we made some simplifying assumptions that take away somewhat from the realism of our conclusions and lead to some outstanding issues. These are discussed, along with some interesting research questions that might improve our results, in Chapter 8. Finally, we make some concluding remarks in Chapter 9.

Chapter 2

Background on Robust Correlation

"Correlation" measures the degree of linear relationship between two random variables. If the correlation of two random variables is close to 1, then their realizations are tightly positively linearly related. That is, if one variable has a "high" realization the other variable will, with high probability, also have a "high" realization. If the correlation between two random variables is close to 0, then it means their realizations have no linear relationship. A change of one variable has little effect on the other variable. Finally, a correlation close to -1 indicates a tight negative linear relationship. The concept of correlation is the major point of intest driving the results of our study. We give some background on correlation for completeness.

Let $\mathbf{X} = (X_1, \ldots, X_m)$ be an m-vector of random variables, μ the m-vector of their expectations, and $\mathbf{\Sigma}$ its m by m covariance matrix. The covariance matrix is symmetric and has entries $\sigma_{i,j}$, the covariance of random variables X_i and X_j defined as follows:

$$\sigma_{i,j} = cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)].$$

From the information in the covariance matrix we can compute an m by m *correlation matrix*. Each entry $\rho_{i,j}$ is the correlation of two random variables X_i and X_j and is defined as

$$\rho_{i,j} = corr(X_i, X_j) = \frac{\sigma_{i,j}}{\sqrt{\sigma_{i,i}}\sqrt{\sigma_{j,j}}}.$$

The value $\sigma_{i,i}$ is simply the variance of random variable X_i.

A traditional and widely-used estimate of correlation is due to Pearson. It can be defined as follows: given n independent and random samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of the random vector $\mathbf{X}$, we can define *sample* estimates of expectation and covariance. Pearson uses the *mean* $\bar{x}_i$ as an estimate of expectation μ_i defined as follows:

$$\bar{x}_i = \frac{\sum_{k=1}^{n} x_{ik}}{n},$$

where x_{ik} is the k^{th} realization of random variable X_i in the kth random sample. The *sample covariance* $S_{i,j}$ of two random variables X_i and X_j is based on the mean and can be defined as follows:

$$S_{i,j} = \frac{\sum_{k=1}^{n}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{n-1}.$$

Pearson's measure of correlation is then simply:

$$C_{i,j}^{PSN} = \frac{S_{ij}}{\sqrt{S_{i,i}}\sqrt{S_{j,j}}},$$

where $S_{i,i}$ is the sample variance of X_i. Note that Pearson's measure is sensitive to outliers, just as the sample mean is sensitive to outliers when estimating expectation. In the case of estimating the expectation, a more *robust* measure is the *median*, which is defined as the 50th percentile of independent random sample data drawn on a random variable. This measure is not as sensitive to outliers as the mean. Indeed, the 50th percentile of a set of data will change little if there is a single very large or very small observation added to the sample, whereas the sample mean might change significantly. Rather informally, we call an estimator *robust* if its value is not significantly effected by the presence of outliers. Using this terminology we can say that Pearson's estimate of correlation is *not* robust. We now define a class of robust estimators that will be used in our study.

A common robust measure of mean and covariance is first introduced by Maronna in 1976 [9], which is defined as follows: given n independent and random samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of the random vector $\mathbf{X}$, Maronna denotes *sample* estimates of expectation and covariance as $\mathbf{t}$ and $\mathbf{V}$, respectively. Each entry in the covariance matrix $\mathbf{V}$, $v_{i,j}$, is the covariance estimate of the random variables X_i and X_j. Then Maronna's estimates of expectation and covariance are solutions $(\mathbf{t}, \mathbf{V})$ of systems of equations

$$\frac{\sum_{k=1}^{n}(u_1[\{(\mathbf{x}_k - \mathbf{t})'\mathbf{V}^{-1}(\mathbf{x}_k - \mathbf{t})\}^{\frac{1}{2}}](\mathbf{x}_k - \mathbf{t}))}{n} = \mathbf{0}, \tag{2.1}$$

$$\frac{\sum_{k=1}^{n}(u_2[\{(\mathbf{x}_k - \mathbf{t})'\mathbf{V}^{-1}(\mathbf{x}_k - \mathbf{t})\}](\mathbf{x}_k - \mathbf{t}))}{n} = \mathbf{V}, \tag{2.2}$$

where u_1 and u_2 are functions satisfying a set of general technical assumptions which are quite involved and will not be stated here. Further details can be found in [9]. Maronna's measure of correlation is defined as

$$C_{i,j}^{MRN} = \frac{v_{ij}}{\sqrt{v_{i,i}}\sqrt{v_{j,j}}}.$$

Maronna's measures of mean and covariance belong to a general class of estimates known as M-estimators. The origin of M-estimators is from the basic statistical technique of maximum likelihood estimation. In such a procedure, we are given a set of observations and then attempt to minimize a function representing the likelihood of the given data to arise from the probability distribution we are attempting to fit. The optimization is over the choice of some parameters or moments of that probability distribution.

Upon taking first-order conditions for this minimization, we define a system of equalities involving the parameters and partial derivatives of the likelihood function. General M-estimator can be defined by optimizing different functions, besides the likelihood function. Each yields alternate equality systems by which to decide estimates of parameters. An example of this is Equations 2.1

and 2.1.

In the case of Maronna's M-estimator, it can be seen as a generalization of maximum-likelihood estimation that provides both robustness and an attractive condition called *affine equivariance*, which essentially requires that the computed moments are consistent under affine transformations of the data [7]. Maronna [9] discovered one of the first affine equivariant robust estimators of correlation and his estimator is widely used in practical applications [1, 12].

Chapter 3

Description of Data and Generating Correlation Time Series

This chapter describes our approach to turning raw bid and ask quote data for prices of stocks into correlation time series describing the correlation of a pair of stocks over time. We then proceed with some basic exploration of the differences between Pearson and Maronna correlation time series, providing some insights for later sections. Quote data are much higher in frequency and volume than trade data, which makes processing and analyzing more challenging. A small sample of intra-day quote data are shown in Table 3.1.

Timestamp	Symbol	Bid Price	Ask Price	Bid Size	Ask Size
09:30:04	NVDA	16.38	20.1	3	3
09:30:04	NVDA	18.23	18.26	3	3
09:30:04	NVDA	18.24	18.26	1	4
09:30:04	ORCL	19.56	19.59	2	104
09:30:04	ORCL	19.58	19.62	1	1
09:30:04	SLB	82.81	83.11	1	1
09:30:04	TWX	14.01	14.2	18	5
09:30:04	TWX	14.01	14.65	2	6
09:30:04	BK	41.11	42.1	41	1
09:30:04	BK	41.13	41.5	1	1
09:30:04	BK	41.11	42.1	38	1
09:30:04	BK	41.13	41.5	3	1

Table 3.1: Sample data from the NYSE TAQ dataset.

In our high-frequency analysis we use the bid-ask midpoint (BAM) as an approximation to the stock prices, and then calculate the 1-period returns. The *bid price* is the highest price someone is willing to pay for a stock, and the *ask price* is the lowest price someone is willing to sell a stock. We choose to use the BAM instead of just the trade price, at which the stock is trading on the open market, as it allows for a closer approximation to the *real* price level between trades, which is generally not the trade price in inefficient markets. Also, using BAM is especially useful in the analysis for stocks which trade infrequently.

3.1 Correlation Time Series

We let $\phi_i(s)$ denote the true stock price of stock i at time $0 \leq s \leq s_{max}$, where $s = 0$ and $s = s_{max}$ are the start and end of the trading day respectively, measured in Δ-length time intervals. In our study we take $\Delta = 30$ seconds. For example, $s = 10$ corresponds to five minutes after the start of the trading day. We denote by $p_i(s)$ the BAM of stock i at time s, which is an estimate of $\phi_i(s)$. As is common practice in financial applications, we do not directly consider stock price but instead focus on log-returns. The *one-period return* of stock i at time s is the ratio $r_i(s) = \frac{p_i(s)}{p_i(s-1)}$ and the *log-return* at time s is $x_i(s) = \log r_i(s)$. Note that $x_i(s)$ is an estimate based on BAM data of the *true* log-return $X_i(s) = \log(\frac{\phi_i(s)}{\phi_i(s-1)})$ based on true prices. The reason for using log-returns instead of the raw prices is twofold: taking the difference of the returns yields a stationary process, while taking the log of the differences results in a distribution that is more approximately normal; both results are necessary in order to utilize statistical tests which assume stationarity and normality.

Our goal is to estimate the true correlation $\rho_{i,j}(s)$ between pairs of stocks i and j at each time s throughout the day. The *true correlation time series* for stock i and j, denoted by $\{\rho_{i,j}(s) : 0 \leq s \leq s_{max}\}$ is the time series of random variables $\rho_{i,j} = corr(X_i(s), X_j(s))$. It is important to note that this time series is not known and must be estimated. In particular, the log-returns will be estimated by the BAM and the true correlation estimated by sample measures of correlation.

For a pair of stocks i and j we compute the sample correlation of log-returns at each time step s to produce a *sample correlation time series* as time progresses. The sample correlation at a given time s is based on a sliding window of the M most recent time intervals. To be precise, the input to each pair-wise correlation calculation at time s are two vectors $\mathbf{x}_i(s)$ and $\mathbf{x}_j(s)$, containing the last M log-returns, taken in Δ-length second time intervals for stocks i and j respectively; that is, the vector of log-returns $\mathbf{x}_i(s) = (x_i(s - M + 1), \ldots, x_i(s - t), \ldots, x_i(s - 1), x_i(s))$.

Now, we define the Pearson correlation measure at time s, $C_{i,j}^{PSN}(s)$, as the sample correlation of data vectors $\mathbf{x}_i(s)$ and $\mathbf{x}_j(s)$ as defined in Chapter 2. Similarly we let $C_{i,j}^{MRN}$ denote the Maronna estimate of correlation at time s. These two measures of correlation define two sample correlation time series $\{C_{i,j}^{PSN}(s) : M \leq s \leq s_{max}\}$ and $\{C_{i,j}^{MRN}(s) : M \leq s \leq s_{max}\}$ henceforth referred to as Pearson and Maronna correlation time series respectively. Note the correlation time series start at time M because the first correlation calculation is based on log-returns from M previous time intervals, and thus cannot be computed before time M.

3.2 Comparing Pearson and Maronna Time Series

We now discuss some basic differences between these two time series–Pearson and Maronna – to get a sense of how these differences might impact trading algorithms using these different measures.

We begin by visually comparing Pearson and Maronna correlation time series based on our sample data of 30-second BAM stock prices using a time window of $M = 100$ time steps. The

data is collected from market-wide TAQ data from the S&P 500 for the one trading day (March 3rd, 2008). Figure 3.1 shows a comparison for raw stock prices for two stocks from the oil and gas sector, Chevron (CVX) and Exxon (XOM).

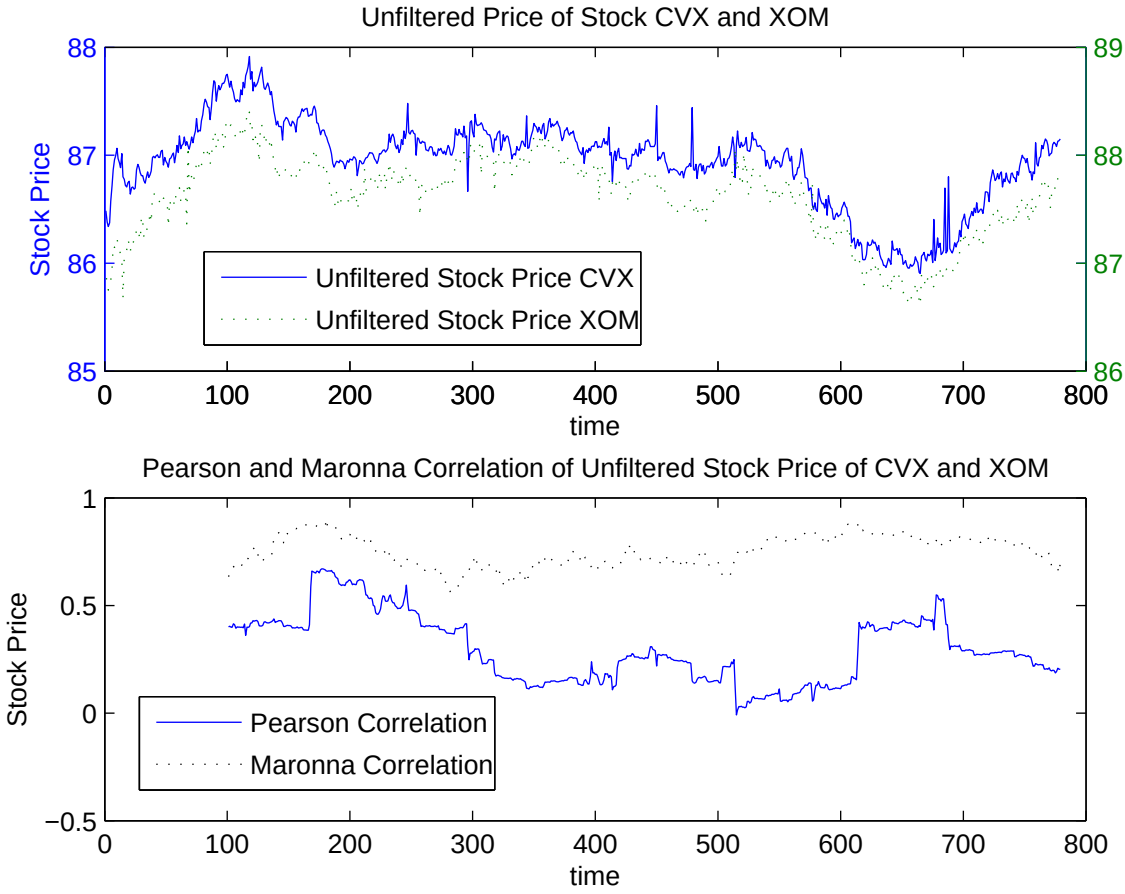

Figure 3.1: Raw Stock Price of CVX and XON and the Two Types of Correlation Measure.
From the plot of the two stock prices, one can see that despite the spikes on the CVX prices, the two stock prices are highly correlated. However, the Pearson measure of correlation is occasionally quite low, with some sudden jumps around time step 520 and 620, whereas Maronna correlation measure stays smoothly at a high level throughout the day.

From the top plot in Figure 3.1, we see that CVX and XOM seem to be highly correlated, but Pearson correlations in the lower plot are relatively low and unstable. The sudden jumps of Pearson correlation – for example, at around time step 510 and 620 – correspond with the outliers in the price time series at time step around 450 and 490. On the other hand, Maronna correlation remains smooth over the entire day. This can partially be explained by the presence of outliers, which we discuss now.

Significant amounts of outliers and noise in real-time high-frequency financial price data impose challenges to calculating accurately the 'real' correlation between prices in different stocks. On one hand, we want to eliminate the effects of outliers and noise, something which is achieved with robust and smooth correlation measures. On the other hand, an effective correlation measure cannot "over damp" the data so that the true information is lost, obscuring the true relationships between the stocks. This tension is quite relevant to our comparison of Pearson and Maronna correlation measures for high-frequency stock price data.

Raw tick TAQ data contains every raw quote, not just the best offer, so there can be many spurious ticks originating from various sources, some human typing errors but mainly from electronic trading systems generating test quotes (e.g., when testing a new feature) or far-out limit orders which have little probability of getting filled. Raw data, whether from a database or a live stream, needs to be cleaned before being analyzed and used in a financial model or strategy.

There are many techniques used in practice to clean high frequency data [6, 4], each having its own advantages and disadvantages. The exact method of cleaning will vary depending on the particular task at hand, and trade-offs between the quality of cleaning and delay need to be managed; i.e., in a real-time environment cleaning process needs to be fast and efficient. Our approach is to use a very simple but effective filter to eliminate prices that are more than a few standard deviations from their corresponding moving average and deviation.[1]

Analogous to how we defined the log-returns vectors $\mathbf{x}_i(s)$ we define a vector $\mathbf{p}_i(s)$ of prices for stock i with entries $p_i(t)$ for $0 \leq t \leq s$. In other words, $\mathbf{p}_i(s)$ is a vector of prices from the start of the trading day until time s. Let $\bar{\mathbf{p}}_i(s)$ denote the average price of stock i up to time s, which corresponds to the average of the entries in the vector $\mathbf{p}_i(s)$.

The filter works as follows: at each time $s+1$, we update the price moving average and standard deviation based on a weighted average of historical average and standard deviation and new average and standard deviation based on new price data. To update the mean at time $s+1$ we use the following rule: if the historical average is $\bar{m}_{old}$, and the new stock price $p_i(s+1)$, then the updated average is

$$\bar{m} = w_1 \bar{m}_{old} + w_2 p_i(s+1),$$

where w_1 and w_2 are weights (which sum to 1) of historical average and new updated average.

In our setting, the weights of historical data and new data depends whether the new stock price is considered to be an outlier. At each time $s+1$, if the price $p_i(s+1)$ is more than k standard deviations away from the current average $\bar{\mathbf{p}}_i(s)$, we treat $p_i(s+1)$ as an outlier, and upgrade the weight of the new average and standard deviation. If the price is within k standard deviations away from the current average $\bar{p}_i(s)$, the average and standard deviation are updated with the pre-defined weights.

The choice of weights is crucial to the performance of a filter, and involves tradeoffs between

[1]The details of this filter were contributed by personal communication with Alpha Lake Financial Analytics.

being too conservative, thus over-damping the prices, and too aggressive, which would fail to filter out outliers. The weights used in our study are not provided here due to proprietary reasons.

After implementing this basic filtering strategy we consider a comparison of Pearson correlation time series derived from filtered price data with Maronna correlation time series derived from unfiltered price data, and find that the Maronna correlation time series is still "smoother" (i.e. less jumps) than the Pearson correlation time series, as shown in Figure 3.2 for our chosen stock pair CVX and XOM. Note, however, that the filtered data does yield a Pearson series that is "closer" to the Maronna series, but the price suddenly changes at around time step 520 and 690, which the filter failed to clean, still cause sudden jumps in the Pearson correlation time series as shown in the lower plot of Figure 3.2.

Filters can result in very different Pearson correlations depending on the choice of weights, as can be seen in Figure 3.3. From Figure 3.3, we see the Pearson correlation time series is very different from the previous one in Figure 3.2 with the only difference being putting more weight on new prices. Therefore, we predict that choosing different filters potentially introduces bias on Pearson correlation calculations, a potentially unattractive feature for pairs trading strategies based on Pearson correlation.

A thorough investigation of other pairs of stocks revealed similar qualitative results. In each case we observed, Maronna correlation time series was always smoother than Pearson. We did not venture further into more quantitative investigations of this phenomenon, in part because the qualitative evidence was so clear. There are clear differences between traditional and robust correlations, something we will see clearly when designing trading strategies based on these alternate correlation measures.

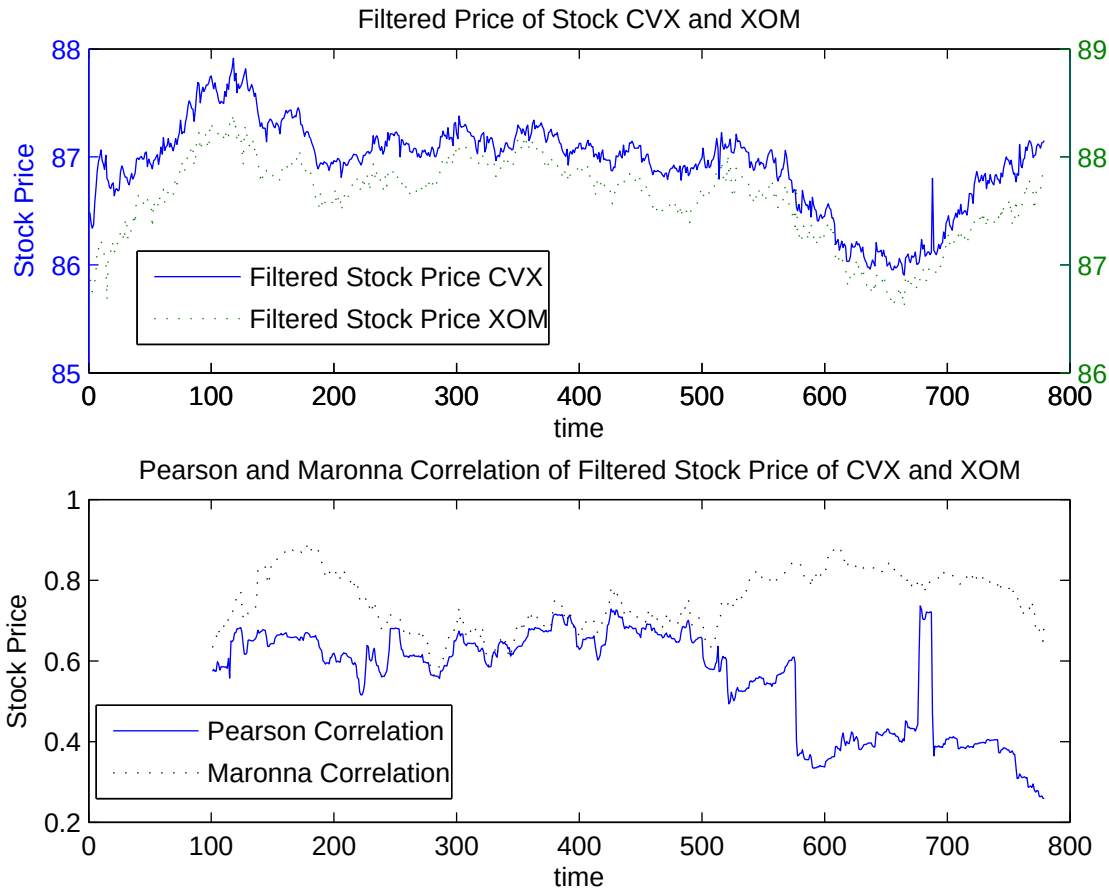

Figure 3.2: Filtered Stock Prices of CVX and XON and the Two Types of Correlation Measure. From the upper plot, one can see most of the spikes in stock prices of CVX are filtered out, except one outlier at around time step 690 due to the pitfall of the filter. From the lower plot, the Pearson correlation converges to Maronna correlation from time step 250 to 500, and the correlation time series becomes unstable due to the outliers at around time step 520 and 690 that were not filtered.

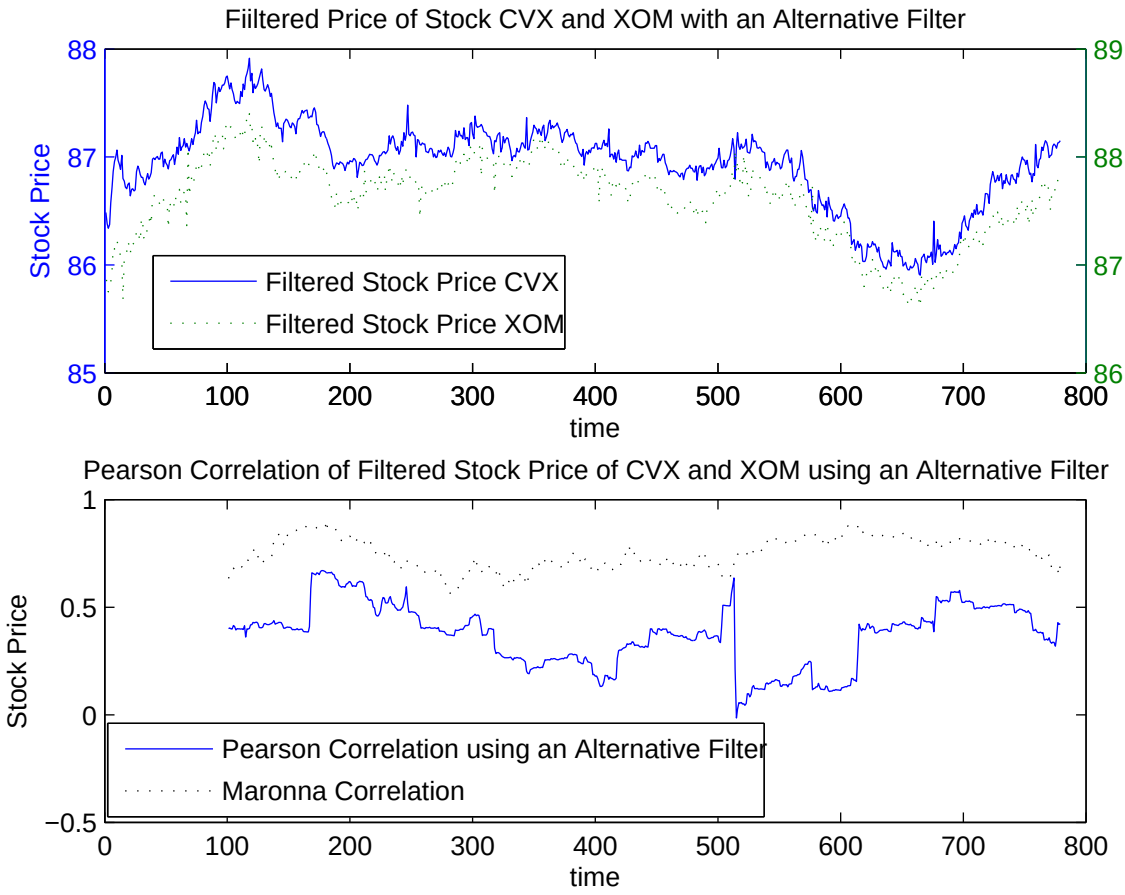

Figure 3.3: Filtered Stock Price of CVX and XON and the Pearson Correlation Measure Using an Alternative Filter.

The Pearson correlation time series shown here is quite different from the previous one in Figure 3.2. This is because of the different filtered prices using different weights in the filters. Therefore, we predict that choosing different filters potentially introduces bias on Pearson correlation calculations, a potentially unattractive feature for pairs trading strategies based on Pearson correlation.

Chapter 4

A Canonical Pairs Trading Strategy

Pairs trading can be broadly categorized into three forms: fundamental, risk, and statistical. A *fundamental pair* is a pair that has been highly correlated over a historical period, usually a few years or more, and often belong to the same industry or sector. A few well-known fundamental pairs are Chevron/Exxon, UPS/Fedex and Wal-Mart/Target. A *risk pair* occurs when a company is about to merge with or acquire another one, and thus the two securities will become highly correlated in anticipation of the adjusted price levels. A *statistical pair* refers to a pair that may or may not be fundamentally linked, but instead have been found to be highly correlated over a given (usually short) period, with a high degree of statistical certainty. Intra-day statistical pairs trading is a high-turnover strategy that uses only very recent data to determine correlations (e.g., the last few hours or days at most). This means that the strategy makes no assumptions that a pair will remain correlated next year or month, but has a certain level of confidence that the pair will remain correlated for the next few hours or days. Generally speaking, we would expect the correlations to remain stable for approximately the same amount of time used in the correlation calculation.

The usual routine for a fundamental pair trader is to first identify a number of candidate pairs. Each pair is then backtested over a given set of data and parameter sets before being promoted to a live trading environment. The exact method used to identify and backtest pairs differs from trader to trader. Some traders may employ a rigorous statistical analysis, while others simply "eye-ball" two charts to determine the degree of correlation. In live trading, the number of pairs monitored per trader can range from a few to a thousand or more; once the number of pairs exceeds what a human can watch, software for monitoring the pairs must be utilized.

A downside to fundamental pairs trading is that often the pairs are well known and widely exploited in the market, implying that mis-pricing is less common, causing arbitrage opportunities to diminish. The upside however is that because the pairs are fundamentally correlated over a long historical period there is a higher degree of certainty that their spread (difference in prices) will revert to its long-run trend.

As we shall see in the pairs trading algorithm below, we define a way to trigger a trade based on the divergence of the correlation of log-returns of stocks away from its historical average. We get alternate *trigger mechanisms* depending on which correlation estimate we use to identify a divergence. Details are found in steps 1. and 2. of the canonical pairs trading algorithm defined below. In the case of a *Pearson trigger mechanism* the algorithems takes as input the Pearson correlation time

series $\{C_{i,j}^{PSN}(s) : M \leq s \leq s_{max}\}$. A *Maronna trigger mechanism* uses the Maronna correlation time series to track divergences. For *Combined trigger mechanisms*, both correlation time series are taken as input. The motivation for designing *Combined trigger mechanisms* was to use both of the correlation measures to combine their theoretical strengths of into a single trigger. We observe from Chapter 3 that Pearson correlation is more sensitive to changes in returns, whereas Maronna is smoother and less sensitive to short term deviations. Intuitively, having a sensitive measure like Pearson is beneficial because it would hopefully indicate sooner when a pair's co-movement deteriorates. On the other hand, such a measure may be misled by outliers in the data to open a trade when it is not beneficial to do so, whereas a smoother measure like Maronna could avoid this mistake. Combined can be seen as an attempt to balance these considerations. The details of its precise definition are proprietary property of Scalable Analytics, Inc.

Table 4.1 describes the strategy parameters and typical values we use within our experimental framework, where these values are chosen based on consultation with professional pairs traders.[2] As before, all time-based parameters are in time units, defined by the time window Δ and indexed by $s = \{0, \ldots, s_{max}\}$, where s_{max} defines the end of the trading day measured in Δ-length intervals. Our algorithm has several parameters which define the specifics of it implications. We let K denote the collection of parameter sets under consideration, and use k to index a particular parameter set. Thus, for instance $\{\Delta = 30, T_{type} = Pearson, A = 0.1, M = 100, W = 60, Y = 10, d = 0.01, \ell = 2/3, RT = 60, HP = 30, ST = 20\}$ is one element of the collection K. Each unique combination of parameters yields different returns, and the difference can be quite significant. We can also backtest a trading strategy for each pair $(i, j) \in \Phi$, with Φ denoting the set of all pairs under consideration, and for each parameter vector $k \in K$ over the given time period.

The following pseudo-code outlines a canonical statistical pairs trading strategy, defined for a particular pair of stocks $(i, j) \in \Phi$ and parameter set $k \in K$ over a given trading day t. If more than one pair in Φ is being tracked this algorithm can be run in parallel for each pair.

1. At time s, calculate the average correlation over the last W time intervals as

$$\bar{C}_{i,j}(s) = \frac{\sum_{u=s-W+1}^{s} C_{i,j}(u)}{W},$$

   where $C_{i,j}(u)$ is the correlation coefficient calculated using log-returns. Note that $C_{i,j}(s)$ was defined for both Pearson and Maronna correlations in Chapter 2.

2. Check to see if $\bar{C}_{i,j}(s)$ is greater than some threshold A, and if the current correlation coefficient at time s has diverged more than $d\%$ from $\bar{C}_{i,j}(s)$ within the last Y time intervals. We refer to d as the divergence threshold. Note that typical divergence levels for pair traders with longer time horizons tend to be larger, due to the fact that the volatility of prices will

[2] We would like to thank Darren Clifford from PairCo for his valuable feedback and suggestions on our work.

Parameter	Description	Values					
Δ_s	Time window	30 sec					
T_{mech}	Trigger mechanism	Pearson		Maronna		Combined	
A	Minimum correlation for trading	0.1					
M	Time window for correlation calculation	50		100		200	
W	Time window of average correlation calculation	60			120		
Y	Time window over which divergences from the correlation average are considered	10		20		40	
d	Divergence level from correlation average required to trigger a trade	0.01%	0.02%	0.03%	0.04%	0.05%	0.10%
ℓ	Retracement level for determining when to reverse a position	1/3					
RT	Time window for measuring the spread level (used in calculating retracement level)	60					
HP	Maximum holding period for any position	30					
ST	Minimum time before market close required to open a new position	20					

Table 4.1: Strategy parameter descriptions and values

also be greater. With our intra-day strategy we use a smaller divergence level to account for lower volatility.

3. If no divergence is detected or if $\bar{C}_{i,j}(s) \leq A$, move on to the next pair. If a divergence is detected, trigger a pair trade. Go long on the stock that has "under-performed" and short the one which has "over-performed". The over-performer is simply the one which has a higher W-period return relative to the other.

4. To choose a long/short ratio, we choose a ratio that keeps us as close to cash-neutral as possible, but just slightly on the long side. For example, if we are buying MSFT at \$30 and selling IBM at \$130, a ratio of 5:1 would give us an allocation of \$150 long and \$130 short. To be more specific, suppose we have two prices $P_i > P_j$, and we want to long stock i and short j, then we want the ratio of long/short shares for stocks i and j to be 1:y, where

$$y = \lfloor \frac{P_i}{P_j} \rfloor$$

Similarly, if we short i, and long j, then

$$y = \lceil \frac{P_i}{P_j} \rceil$$

5. The next step is to decide when to reverse the positions. We reverse the position when we have reached a retracement level L, or if a given amount of time has elapsed since we entered

the position. The *retracement level* is as follows. Let S_l, S_h and $\bar{S}$ be the high, low and average of the spread during the last M time intervals, and S_s be the spread of the two stock prices at the time we opened the position. If $S_s \leq \bar{S}$, then

$$L = S_l + \ell(S_h - S_l),$$

and if $S_o \geq \bar{S}$, then

$$L = S_h - \ell(S_h - S_l)$$

where $0 < \ell < 1$ is known as the *retracement parameter*. For example, if the high of a MSFT-IBM spread is \$100, and the low \$80, and we opened the position when the spread was around \$80, and $\ell = \frac{1}{3}$, then we reverse when the spread has reached the retracement level

$$L = \$80 + \tfrac{1}{3}(\$100 - \$80) = \$80 + \tfrac{1}{3}\$20 = \$86.67.$$

Similarly, if we opened the position when the spread was around \$100, then

$$L = \$100 - \tfrac{1}{3}(\$100 - \$80) = \$100 - \tfrac{1}{3}\$20 = \$93.40$$

and we will reverse the position when the spread is lower than L. We also need to add a time-based reversal trigger in case the retracement level is never reached. Therefore, we choose not to hold a position longer than HP time periods. Thus after HP time periods the position is reversed, regardless of the situation. Finally, we should reverse all positions at the end of the trading day. We note here that the key to a good strategy is to mitigate losses and control risk. Thus, we point out, but do not implement, several other reversal conditions. The first is an absolute stop-loss: If the spread continues to drop rapidly, we want to exit and minimize our loss. The second is correlation reversion: If the correlation returns within the average range (i.e., $[\bar{C}(1-d), \bar{C})$, then we reverse the positions. The reasoning behind correlation reversion is that the prices may have adjusted to new levels and watching for spread reversion may not give us this information.

6. Once the position is reversed, we calculate the return $R_{i,j}$ for pair of stocks over both the long and short positions, with

$$R_{i,j} = \frac{\pi_{i,j}}{P_i N_i + P_j N_j}$$

where $\pi_{i,j}$ is the profit/loss of the trade (in dollars), P_i and P_j are the prices and N_i and N_j the number of shares held for stock i and j respectively. For example, suppose a trade was to long

MSFT at \$30 and short IBM at \$130 with the ratio of MSFT to IBM 5:1. If we reverse the position when MSFT is \$29 and IBM is \$120, then we profit (\$29−\$30)5+(\$120−\$130(−1)) = \$5 from this trade. The total cost, not including transaction costs, is 5(\$30) + 1(\$130) = \$280, and thus the return is \$5/\$180 = 2.8%.

As discussed above, the input to the pairs trading algorithm is the correlation time series $\{C_{i,j}(s) : M \leq s \leq s_{max}\}$ of a pair of stocks $(i, j) \in \Phi$. The output is a set of returns $R_{i,j}$ of trades that were opened and closed between times 0 and s_{max}. Thus, $R_{i,j}$ is a set of numbers, one for each trade that opened and closed during the trading day. It is not to be confused with the returns data $r_i(s)$ of a particular stock defined in the previous chapter, here the return is of a particular *trade* on a particular *pair* of stocks.

We would like to choose Φ as the full set of stocks which may potentially be chosen for backtesting, so as to optimize the strategy to perform well under that set of stocks. If there are n stocks then $|\Phi| = \frac{n(n-1)}{2}$. If our goal was to backtest over all US stocks, of which there are approximately 8000, this would require our strategy to support backtesting on over 32 million pairs! While many stocks are not liquid enough (too few trades) to be considered in our style of pairs trading, the number of potential pairs is still so large that a parallel algorithm like those which can be implemented in `MarketMiner` would be essential for real-time trading. This current study does not test on this many pairs, instead it focuses on our list of 61 commonly traded stocks and focuses on the differences in trading algorithms arising from one key parameter: trigger mechanism.

Chapter 5

Backtesting of Trading Strategies

5.1 The General Backtesting Process

A natural question is to ask which configuration of parameters results in the best performance. One way to compare them is to test on historical data and measure the performance of each. This procedure is called *backtesting*. Backtesting a pairs trading strategy on a particular pair of stocks involves choosing a suitable set of historical data H, running the strategy on H and noting wins and losses of each trade and computing some measure of performance, such as cumulative returns. For comparison, one can do backtesting on alternative configurations of a given pairs trading strategy on the same data H and compare the relative performance results. This basic procedure can be done across a variety of strategies, pairs, sets of historical data and performance measures to help identify the best overall trading strategy. In our experiments we focused on testing the performance of trading strategies where the major difference was in the type of trigger mechanism.

The raw data used in the experiments are the TAQ BAM data for the 61 highly liquid US stocks frequently traded by professional pair traders, as described in Chapter 3. Since we examine all pairs for a given set of stocks, the results presented here are based on $\binom{61}{2} = 1830$ pairs. Our strategy works on high-frequency time frames, and thus the total dataset we consider here is limited to one month (March 2008) which consists of 20 trading days. While designing our market-wide pairs trading strategy we performed some preliminary experiments using MATLAB to get a feel for the different parameters and range of values they would take. These values are given in Table 4.1. As mentioned in Chapter 2, Maronna correlation time series are computationally expensive to produce, and this causes some complexity in our experiment. We tried three different approaches, and each exhibited some trade-off among efficiency, accuracy and feasibility of implementation. In this experiment, we used MATLAB to produce correlation time series, and used them in our pair trading strategies. The caveat here is that calculating the Maronna correlation coefficients pair-wise no longer assures the resulting correlation matrix is positive semi-definite (PSD), an important theoretical property of correlation matrices. While this approach worked reasonably well for a small dataset of 61 stocks, we are also aware that this solution will not scale. Nonetheless, using MATLAB to generate our correlations directly proved to be more efficient and feasible for this small data set. How we might use an approach more integrated with `MarketMiner` is discussed in Chapter 8.

5.2 Evaluating a Trading Strategy

The approach in which an intra-day strategy is evaluated differs from strategies which make trades only occasionally (e.g., every few days or even just once a month, in the case of a pension or mutual fund). Since we have many trades each day, we not only want to evaluate how the strategy performs over multiple days through a given period of time, but also within each day. To do so, we adapt some of the trading model evaluation measurements from the high frequency finance literature [2]. In a given trading day t, for each for pair p and parameter vector k, a set $R_p^{t,k}$ of returns is generated using our trading algorithm in the previous chapter. Therefore the total set of returns for the trading periods is just the union of each days returns:

$$R_p^k = \bigcup_{t=1}^{T} R_p^{t,k}, \tag{5.1}$$

where T is the total number of trading days under consideration. The following analysis considers three key performance metrics to assess the performance of a trading strategy: cumulative returns, maximum draw-down and win-loss ratio. These performance measures can be defined either over a given pair p and parameter set k, or summarized over all pairs or over all parameter sets. Each of the three variants provides a different view of the results. For example, summarizing the results over all pairs for a given parameter set indicates which parameters are most effective, while summarizing over all parameter sets for a given pair indicates that the pair may be a particular good candidate for pairs trading and less sensitive to choice of parameters. The formulas for each of the three performance measures are given below.

1. *Cumulative Returns*: Cumulative returns measures the equity growth of a particular strategy. This measure is appropriate when we assume that the strategy always reinvests the total available capital at the start of each period. The *daily cumulative return* for pair p and parameter vector k on day t is defined as

$$r_p^{t,k} = \prod_{q=1}^{|R_p^{t,k}|} (r_{p,q}^{t,k} + 1) \tag{5.2}$$

where $r_{p,q}^{t,k}$ is the qth return on day t, and $|R_p^{t,k}|$ denotes the number of trades in the set $R_p^{t,k}$. The *total cumulative return* r_p^k over the entire trading period, again for pair p and parameter set k, is calculated as

$$r_p^k = \prod_{t=1}^{T} (r_p^{t,k}). \tag{5.3}$$

Both the daily and total cumulative returns can be further summarized by aggregating the returns over all pairs using a given parameter set, or over all parameter sets but for a particular

pair. These measures can be used to test the effects of pairs choice and parameter choice on returns of trading strategies, and thus help with refining trading strategies. For example, the total cumulative return over all pairs on day t using parameter set k is

$$r^{t,k} = \prod_{p \in \Phi} (r_p^{t,k}) \tag{5.4}$$

and similarly, the total cumulative return for pair p on day t over all parameter sets is

$$r_p^t = \prod_{k \in K} (r_p^{t,k}). \tag{5.5}$$

The same summary calculations can be applied to daily cumulative returns.

2. *Maximum Drawdown*: Maximum drawdown is a measure of the riskiness of a trading strategy. It is the maximum compounded, not annualized, loss that the strategy ever incurs at some intermediate point in time during the life of the investment strategy. It can also be seen as the "worst peak to valley drop" of the compounded return over the investment period, for the pair p:

$$MDD_p = \max_{k \in K} \left(\frac{r_{p,q_a}^k - r_{p,q_b}^k}{r_{p,q_a}^k} : q_a, q_b \in R_p^k, q_a \leq q_b \right), \tag{5.6}$$

where r_{p,q_a}^k and r_{p,q_b}^k are the cumulative returns for pair p using parameter set k from trade number 1 to q_a and q_b, respectively. Note that we could also define maximum drawdown MDD^k for a given parameter set k. Moreover, we can define the two variants of maximum drawdown on a daily basis for pair p and parameter set k, which is:

$$MDD_p^k = \max \left(\frac{r_{p,t_a}^k - r_{p,t_b}^k}{r_{p,t_a}^k} : t_a, t_b \in T, t_a \leq t_b \right). \tag{5.7}$$

3. *Win-Loss Ratio*: The win-over-loss trades ratio provides additional information on the effectiveness of strategies, essentially indicating the relative frequency of "wins" throughout a trading period. Its definition is

$$\frac{W_p^k}{L_p^k} = \frac{|\{r_{p,q}^k : r_{p,q}^k > 0, q \in R_p^k\}|}{|\{r_{p,q}^k : r_{p,q}^k < 0, q \in R_p^k\}|}, \tag{5.8}$$

where $\{r_{p,q}^k : r_{p,q}^k > 0)\}$ is a set of trades with positive returns, and $\{r_{p,q}^k : r_{p,q}^k < 0)\}$ is the set of trades with negative returns. If we are interested in the difference of the performance of

the strategies with different parameters values, we can use

$$\frac{W^k}{L^k} = \frac{|\{r^k_{p,q} : r^k_{p,q} > 0, p \in \Phi, q \in R^k_p\}|}{|\{r^k_{p,q} : r^k_{p,q} < 0, p \in \Phi, q \in R^k_p\}|}, \tag{5.9}$$

where again Φ denotes the set of all pairs under consideration.

After a series of tests, in the end we chose performance measures for pairs of stocks averaged over all parameters except for the trigger mechanism over the entire trading period of one month (March, 2008) because our goal is to compare the results over general choices of parameters except for trigger mechanism. This way we can focus on the effects of trigger mechanism and average out impacts coming from other parameter choices. The next chapter presents more details on our approach. Our evaluation procedure takes as input the returns data generated from our pairs trading algorithms and output three types of performance measures for each pair of stocks. These performance data are the raw data for the statistical analysis of the following chapter.

Chapter 6

Statistical Analysis

The results presented here focus on some preliminary performance data from trading 61 stocks generated by our MATLAB implementation. Further, some of the intuition and implications of these results, including impact on actual pairs trading practice, will be discussed in more detail in the following chapter. Here we primarily present the results of our analysis and describe the statistical tests we used to derive the results.

Performance comparisons of two different trading strategies can be done across several dimensions: the trigger mechanism used, the choices of parameters and pairs, etc. We focus attention on differences in performance arising from different choices of trigger mechanism. With a large set of returns data and their corresponding performance measures we may ask whether this information can help to shed some light on which strategies are more effective - those using Pearson, Maronna or Combined trigger mechanisms. We analyze three performance measures - cumulative monthly return (5.3), maximum daily drawdown (5.7), and the win-loss ratio (5.9). We aggregate the data by taking an average over all parameter sets considered.

Here are the specific details for our analysis. We may consider Pearson, Maronna and Combined trigger mechanisms as our *treatments*, which are applied to 1830 pairs of stocks, with other *factors* (not considered part of our treatment) consisting of the remaining elements in our parameter sets: $\{\Delta_s, A, M, W, Y, d, \ell, RT, HP, ST\}$. We run the experiments on different levels of these factors to account for bias of choosing any one level. Each pair of stocks receives each treatment at each level of the remaining factors.

The response from each treatment is one of our three performance measures - cumulative monthly return, maximum daily draw down, and win-loss ratio. We discuss in detail the case of cumulative monthly returns, but the other cases are similar. Recall our notation that r_p^k is the total cumulative return of pair p using parameter vector k over the period of one month. To highlight the fact that there are three treatments we let $r_p^{T_{mech},k'}$ denote the return with a specified trigger mechanism T_{mech} with $k' \in K'$ representing the 14 different parameter vectors of the form $\{\Delta_s, M, W, d, \ell, RT, HP, ST, Y\}$. Thus there are 14 levels of non-treatment factors, and each pair has a response $r_p^{T_{mech},k'}$ for each of these levels. Our approach is to average these responses over the different factor levels to get a single estimate of the performance of pair p using trigger mechanism T_{mech}. Thus, the sample observations from our populations are *average* cumulative returns over

the month:

$$\bar{r}_p^{T_{mech}} = \frac{\sum_{k' \in K'} r_p^{T_{mech},k'}}{|K'|}$$

where the average is over the set of alternate parameter vectors K'. We define average maximum daily drawdown and win-loss ratio for each pair of stocks and each correlation measure analogously, again where the average is over the 14 different levels of the non-treatment factors. For simplicity, we will call any trading strategies using trigger mechanism T_{mech} a T_{mech} strategy.

6.1 Descriptive Statistics

Tables 6.1, 6.2 and 6.3 contain descriptive statistics for each performance measure with respect to the different trigger mechanisms. The tables include mean, median, standard deviation, skewness and kurtosis. Mean and median give estimates of the central location of the data, and standard deviation describes the spread of the data. Also included are measures of skewness and kurtosis, which are the third and fourth moments of a distribution, respectively. Generally speaking, skewness measures the lack of symmetry of a distribution around its mean, and kurtosis measures the degree to which a distribution is more or less "peaked" than a normal distribution. A higher degree of skew to the right (large positive skewness statistic) is more attractive for a trading strategy since it means there are higher proportion of samples that are greater than the mean than those less than the mean. A greater kurtosis means there is a relatively greater probability of an observed value being either close to the mean or far from the mean.

From Table 6.1, we can see that Pearson has the highest average cumulative monthly return both in terms of mean and median. The "best" value for each measurement is shown in bold. For the mean, Pearson is 1.1521, around 3.67% more than that of Combined which has the smallest returns. For the median, Pearson has a value of 1.1278, around 2.65% more than that of Combined. The difference between Pearson and Maronna is around 0.4%, much smaller than the difference between Pearson and Combined. We also notice that although Combined has lowest returns, it also has the loweset standard deviation, which is a measure of risk. Combined has a standard deviation of 0.0747, following by Pearson 0.1085, and the highest one is Maronna with a value of 0.1235. This means the value of the monthly returns of the Combined are less spread than both Maronna and Pearson, which indicates less risk. These two observations of the means and standard deviation confirms the intuition that high returns often associate with high risk, and low returns with low risk. To balance the consideration of risk and reward, Table 6.1 also shows the Sharpe ratio, which is a measure of *risk-adjusted return* and defined as

$$S_R = \frac{\bar{r}}{\sqrt{s^2}}$$

where $\bar{r}$ is the average return and s^2 is the sample variance of the return around its mean. A higher Sharpe ratio corresponds to more attractive risk-to-reward characteristics. In particular, a risk

averse trader would tend to favor trading strategies with high Sharpe ratios. Combined has the highest Sharpe ratio (14.8568), and Maronna has the lowest (9.2899). This implies that although Combined has smallest returns, its advantage in terms of risk make it the most attractive trigger mechanism for balancing risk of reward at least according to the Sharpe ratio.

From the skewness and kurtosis of monthly returns, we find that the cumulative returns of Maronna trading strategies have a skewness of 2.8484 and kurtosis of 16.6541. Compared to the skewness and kurtosis of Pearson (1.9281 and 9.4091 respectively) and those of the Combined (1.4871 and 7.1706 respectively), Maronna is much more skewed to the right and has fatter tails than the others, which suggests more trades yield unusually high returns. Therefore, although the location estimate and standard deviation shows Maronna does not yield the highest return, and potentially have high risk, these skewness and kurtosis statistics suggest that Maronna strategies yield very high returns for select pairs.

Table 6.2 gives another aspect of risk in terms of the maximum drawdown. Notice the skewness of all the three is relatively high with values above 3, so we should look at the median as our estimate of the central location measure. Pearson has the lowest median maximum daily dawn down (1.1533%). This means if we use Pearson trigger mechanism in the pair trading strategy, on average there will be at most 1.1533% drop on the compounded returns from the last peak during March of 2008. It also shows that Maronna has the highest maximum drawdown (1.2446%) and thus Maronna mechanisms may introduce very high losses on average. Combined is close to Pearson with slightly worse performance.

Turning to our statistics for the win-loss ratio, we note that if the win loss ratio of a strategy is about 1 or lower than 1, then it may be considered ineffective since it loses more often than it wins. If the strategy has a value greatly higher than 1, it may indicate the strategy has high predictive power of the relative prices of the stocks. That is, if a strategy has a high win-loss ratio, this indicates that it is effectively anticipating the price movements of the pairs in order to make a positive return on trades.

From Table 6.3, we see that all trigger mechanisms have a win-loss ratio greater than 1 and that the mean and median of Pearson and Combined are very similar in value, while Combined has slightly higher values, 1.2787 and 1.2689. The standard deviation of Combined also is the highest overall, which means the win-loss ratio may vary a lot depending on the pair of stocks. Nonetheless, the values of all the three trigger mechanisms of win-loss ratios are relatively close, and thus the difference of these point estimates may not be significant. We test this claim later in the chapter.

In addition to these tables, box plots are included to give a qualitative appreciation of the data (see Figures 6.1 to 6.3). On each box, the central mark is the median of the distribution, the edges of the box are the 25th and 75th percentiles (or first and third quartiles), the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. From Figure 6.1, we see that Maronna has the highest spread, and some extreme high values for some pairs plotted as outliers. High spread suggests high risk, and some extreme high values may need

	Trigger mechanism: T_{mech}		
	Maronna	**Pearson**	**Combined**
Mean	1.1473	**1.1521**	1.1098
Median	1.1204	**1.1278**	1.0979
Standard Deviation	0.1235	0.1085	**0.0747**
Sharpe Ratio	9.2899	10.6184	**14.8568**
Skewness	**2.8484**	1.9281	1.4871
Kurtosis	16.6541	9.4091	**7.1706**

Table 6.1: Average cumulative monthly returns

	Trigger mechanism: T_{mech}		
	Maronna	**Pearson**	**Combined**
Mean	1.6662%	**1.5433%**	1.5666%
Median	1.2446%	**1.1533%**	1.1702%
Standard Deviation	1.5481	**1.4606**	1.4668
Skewness	**3.4443**	3.5005	3.889
Kurtosis	21.5922	**21.5295**	27.3131

Table 6.2: Average maximum daily drawdown

	Trigger mechanism: T_{mech}		
	Maronna	**Pearson**	**Combined**
Mean	1.2697	1.2724	**1.2787**
Median	**1.2652**	1.2688	1.2689
Standard Deviation	**0.1263**	0.1269	0.1356
Skewness	0.2897	0.2521	**0.3002**
Kurtosis	3.0781	**3.0665**	3.0991

Table 6.3: Average win-loss ratio

some investigation on of which pairs and why Maronna makes extreme high profit. Pearson has the highest median, and less spread than Maronna. This suggests on average Pearson is better than Maronna in term of both rewards and risks, which makes Pearson preferable to Maronna. Combined has a lower median, fewer outliers, and smaller spread. This confirms with previous observations that Combined is most preferable in terms of risk. Figure 6.2 shows the performance in term of maximum drawdown of the three trigger mechanisms are similar, but Pearson seems to have less outliers than the others. This indicates that Pearson trigger mechanism generate smoother cumulative returns over the test period, which may be attractive for some traders. From Figure 6.3, we see that the performance of win loss ratio of Maronna and Pearson are very similar. However, Combined seems to have higher spread and more outliers with extreme values on the positive side. This, again, is worthy of further investigation into which pairs and why Combined has high predictive power in these cases.

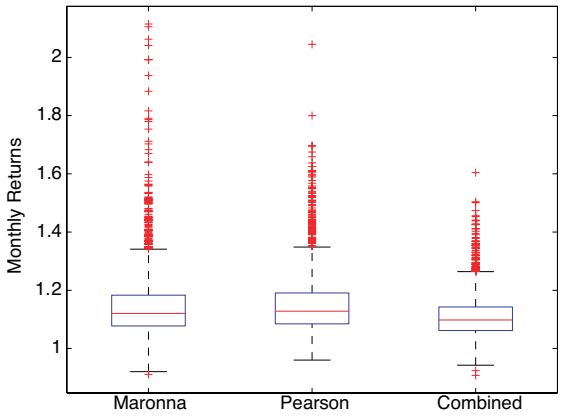

Figure 6.1: Box plot for average cumulative monthly returns
The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. From the plot, we can see that Maronna have the highest spread, and some extreme high values for some pairs as plotted as outliers. Pearson have the highest median, and less spread than Maronna. Combined has lower median, but also has a lot less outliers, and smaller spread.

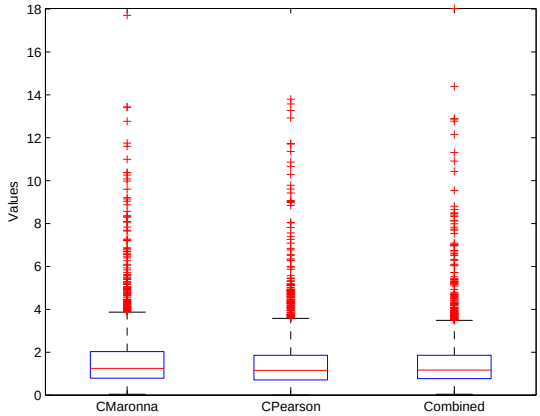

Figure 6.2: Box plot for average maximum daily drawdown
The performance of the three trigger mechanisms are similar, but Pearson seems to have less outliers than the others.

6.2 Some Tests of Significance

It is important to stress that all of these simple comparisons between values in the tables above need to be examined on a more rigorous standard of *statistical significance* in order to be truly

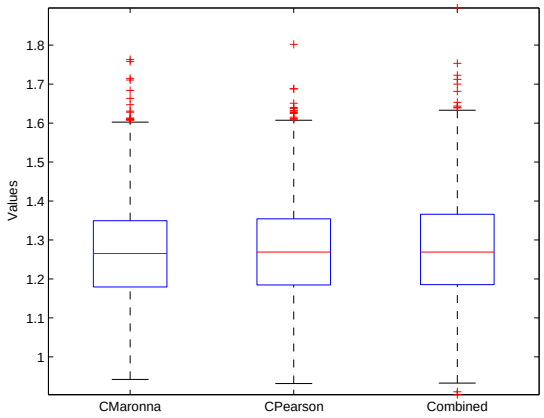

Figure 6.3: Box plot for average win-loss ratio
The performance of win loss ratio of Maronna and Pearson are very similar. However, Combined seems to have higher spread and more outliers with extreme values.

meaningful. To do so we consider a few simple statistical tests. We discuss the ideas of a basic scheme for designing tests on cumulative monthly returns as one example of the type of analysis we are interested in, and the other performance measures can be analyzed in a similar fashion. To be clear about the underlying statistical model on which we are basing our analysis we can consider our tests with respect to three populations. One population is cumulative monthly returns of pairs averaged over the 14 different parameter sets using the Pearson trigger mechanism in the trading strategy amongst *all* 'highly' correlated pairs in the market. The other populations are similarly defined, where instead we use Maronna and Combined trigger mechanisms. The averaged cumulative monthly returns of the 1830 pairs yields 1830 sample data points per population.

We conduct hypothesis tests related to whether trading under the three types of trigger mechanisms differ significantly in terms of their means. Combining information on variance, skewness and kurtosis would give a more complete measure of risk, but unfortunately there are no common statistical inference tests of skewness and kurtosis and the standard tests of variance require an assumption of normality which was not justified in most of our data sets. Thus, we focus solely on testing difference of means.

Assume the pairs arising from our 61 stocks are random draws from the three populations. It is unlikely that these populations are independent, due to the simple observation that if a pair has profitable trades in one correlation measure it is likely to have profitable trades with another correlation measure.

To deal with dependent populations, we use a paired t-test to compare the the difference in means of two populations at a time. In the paired t-test we consider differences of sample observations. In the case where we compare the Pearson and Maronna populations define the vector of

differences:

$$D_p = r_p^{PSN} - r_p^{MRN},\ p \in \Phi. \tag{6.1}$$

If we can verify the assumption that the differences D_p represent independent observations form an $N(\delta, \sigma_d^2)$, then we can test the null hypothesis H_0:

$$H_0 : \delta = 0$$

versus

$$H_a : \delta \neq 0,$$

using the paired t-test procedure, a common method in statistics analysis [13]. Note that we may safely assume normality here, since the sample size for this test $n = 1830$ is substantial and thus by the Central Limit Theorem the mean is approximately normally distributed. Besides this two-sided test, the two alternate one-sided tests can be conducted to decide which mean is larger, if significantly different. For clarity, let $\delta^{PSN,\,MRN}$ denote the population mean for the difference between trading strategies using Pearson and Maronna (as in (6.1)). Similarly define $\delta^{PSN,\,COM}$ and $\delta^{PSN,\,MRN}$. Table 6.4 is a summary of results of the paired t-tests for making inferences on the differences of means for cumulative monthly returns:

H_0	H_a	**t-statistic**	**Reject H_0 (Y/N)**
$\delta^{PSN,\,MRN} \leq 0$	$\delta^{PSN,\,MRN} > 0$	5.6352	Yes
$\delta^{PSN,\,COM} \leq 0$	$\delta^{PSN,\,COM} > 0$	47.5728	Yes
$\delta^{MRN,\,COM} \leq 0$	$\delta^{MRN,\,COM} > 0$	28.8835	Yes

Table 6.4: Paired t-test for average cumulative returns (all tests at 5% significance level)

We conclude with high statistical significance that the mean of cumulative monthly returns is highest for Pearson, next highest for Maronna and lowest for Combined.

Similar tests were conducted for average maximum drawdown and win-loss ratio for each of Pearson, Maronna, and Combined trading strategies on our 1830 pairs. The tests are set up analogously to those above and we can conclude the following with high statistical significance: the average maximum-draw-down over 14 parameter vectors is smallest for Combined, and there is no significant difference between Pearson and Maronna. As for win-loss ratio, Combined performs better than Pearson which in turn performs better than Maronna. Tables 6.5 and 6.6 give details for these conclusions.

The results in these tables are interpreted in the following chapter.

H_0	H_a	t-statistic	Reject H_0 (Y/N)
$\delta^{PSN,\ COM} = 0$	$\delta^{PSN,\ COM} \neq 0$	0.51123	No
$\delta^{MRN,\ PSN} \leq 0$	$\delta^{MRN,\ PSN} > 0$	34.9392	Yes
$\delta^{MRN,\ COM} \leq 0$	$\delta^{MRN,\ COM} > 0$	30.4685	Yes

Table 6.5: Paired t-test for average maximum-draw-down (all tests at 5% significance level)

H_0	H_a	t-statistic	Reject H_0 (Y/N)
$\delta^{PSN,\ MRN} = 0$	$\delta^{PSN,\ MRN} > 0$	1.8015	No
$\delta^{PSN,\ COM} \geq 0$	$\delta^{PSN,\ COM} < 0$	4.8359	Yes
$\delta^{MRN,\ COM} \geq 0$	$\delta^{MRN,\ COM} < 0$	6.3008	Yes

Table 6.6: Paired t-test for average win-loss ratio (all tests at 5% significance level)

Chapter 7

Discussion of Results and Implications for Traders

In this chapter we take some time to discuss some of the implications of the statistical analysis from the previous chapter for the practice of pairs trading. Although more backtesting with more pairs, more sets of historical data and different parameter sets would be needed to have a more complete picture of the effectiveness of our pairs trading strategy under alternate trading mechanisms, we can make some preliminary remarks and provide general insights based on the data that we have collected.

We begin with a summary of our findings. Combining Table 6.2 to Table 6.6, the ranks in terms of the performance measures of the three different trigger mechanisms are given in Table 7.1. A tie indicates that the statistical test comparing their performances was inconclusive.

	Trigger mechanism: T_{mech}		
	Maronna	**Pearson**	**Combined**
Cumulative Return	2	1	3
Maximum Drawdown	2	1	1
Win Loss Ratio	2	2	1

Table 7.1: Ranks in performance measure

We were actually quite surprised that on average Pearson generates significantly higher cumulative returns than Maronna. Theoretically, Maronna correlation, as a robust measure of correlation, should give a better estimate of the true correlation of highly volatile set of data. Therefore, we expect that trigger mechanisms based on Maronna correlation to detect break-downs of co-movement more accurately, and thus brings higher returns with less risk. Nonetheless, this turns out not to the case in practice. One possible explanation of this is the phenomenon in our observation that Maronna lags in identifying divergence in correlation when compared to Pearson. Careful inspection of Figure 3.2, for instance, demonstrates the phenomenon. Our intuition is that although Pearson is more sensitive to outliers, it also seems to be "quicker" to respond to price changes and thus on average enter trades at more favorable times. In comparison, Maronna enters a position at a lagged time, even though it might have a more accurate measure of the true correlation than Pearson according to theory. Thus, this observation that Maronna is less responsive than Pearson seems to outweigh the downside of calculating true correlation less accurately or over-reacting to

outlying data, which are criticisms of Pearson's measure.

These observations, however, are based on our algorithms using filtered data. It stands to reason that the less clean the data, the higher is the cost of being sensitive to outliers and thus the diminished performance of Pearson compared to Maronna. Some preliminary results we undertook on unfiltered data confirms this intuition. The trading industry's use of filters as opposed to robust correlation computational engines to deal with highly volatile and "dirty" data, seems to be partially justified by our analysis. Indeed, filters are simple and direct ways to "clean" data, whereas robust correlation measures like Maronna are more complex to implement. However, this also underscores the importance of having effective data filters in trading applications, and we propose that having a robust correlation measure like Maronna might be used as some benchmark or reference point when comparing the effectiveness of data filters.

It should also be noted that the above discussion refers to *average* cumulative returns over all sets of parameters we considered in our study. Interestingly, as can be seen from Figure 6.1 the distribution of monthly returns for Maronna trading strategy are more skewed to the right than Pearson. This is also reflected in the skewness value of Maronna trading strategies. In other words, there were more pairs that performed exceptionally well under a Maronna strategy than with Pearson. An important implication is that if a trader could identify something about the characteristics of pairs that perform well with Maronna strategies, this would provide a competitive advantage over other traders who only consider Pearson strategies.

On the other hand, we find that trading strategies with Maronna trigger mechanisms perform worse in terms of maximum drawdown, which was also surprising. This contradicts our basic intuition that Maronna correlation is a more "conservative" measure of correlation, since in theory it is less sensitive to short term trends in prices that may result in poorly-timed trades or large losses. One possible explanation of this counter-intuitive finding is the observation made earlier that Maronna lags in identifying deterioration in correlation. This lag may lead Maronna strategies triggering trades at inopportune times and thus become exposed to large losses. As intra-day traders are well aware, small lags in computation can be detrimental to a trading strategy. This phenomenon is a subject for further investigation.

Moving our focus away from direct comparisons between Maronna trigger mechanisms and Pearson trigger mechanisms, we observe that trading strategies using the Combined trading mechanism has some interesting characteristics. As mentioned previously, Combined trigger mechanisms have the largest Sharpe ratio (14.8568), which suggests that Combined strategies strike a good balance between risk and reward. We do not pursue detailed explanations of why this might be the case, as it would reveal too much information about the Combined trading mechanism which is proprietary.

Another observation that adds to the attractiveness of Combined strategies is due to the fact that up until this point we have not considered transaction costs per trade. In a practical setting, each trade made by a trader involves some cost. One indicator of how this might effect the various strategies is by looking at the average *number* of trades made per day. We found that, on average,

Combined strategies made around 30 trades per day, whereas Pearson and Maronna strategies made closer to 40 trades per day. In other words, Combined strategies made their returns on fewer trades and thus will be less adversely affected by transaction costs.

	Trigger mechanism: T_{mech}		
	Maronna	**Pearson**	**Combined**
Average Number of Trades	38.8631	38.3598	29.7760
Standard Deviation of Number of Trades	5.3500	6.4708	5.1091

Table 7.2: Average number of trades per day for each trigger mechanisms

With these numbers in mind, another observation can be made regarding the effectiveness of Combined strategies. From Table 7.1 we see that Combined outperforms Maronna and Pearson in terms of win-loss ratio. Thus, although Combined strategies trigger fewer trades, a higher ratio of these trades yield "wins". This may be attractive for a trader who bears high commission costs and prefers a strategy which enters fewer well-timed trades. Psychologically, traders may prefer this scenario, as having a larger percentage of smaller "wins" may be less stressful and give more confidence.

Chapter 8

Some Outstanding Issues

Our analysis and findings are based on some important assumptions and choices in the design, implementation and assessment of our trading strategies. Some of these assumptions and choices raise some potential issues. We will discuss some of these issues and point to future research directions that might be taken to rectify them.

8.1 Improved Integration with `MarketMiner`

As discussed in Section 5.1, we implemented the pairs trading algorithm, including the computation of robust correlation, in MATLAB. Although this turned out to be the most efficient approach of those that we tried, if larger experiments are to be undertaken, a more sophisticated approach to computation should be considered that takes advantage of the efficiencies of the `MarketMiner` platform.

Our MATLAB implementation does not use `MarketMiner` to compute correlation, but rather re-created all correlation time series in MATLAB. We were able to produce a daily return vector $R_p^{t,k}$ for a given pair p, day t and parameter vector k in approximately 2 seconds, depending on the specific pair and parameters, using an Open SUSE Linux PC with a dual core Intel Pentium 4 2.80 GHz processor. With the need to produce 1830 (number of pairs) $\cdot$ 20 (number of business days in March, 2008) $\cdot$ 42 (number of parameter sets) daily return vectors to track returns over a given month, a rough estimate for the computation time on a single computer is 854 hours. Using this same scenario but backtesting over a year would take about 445 days, and even worse, scaling up to 1000 pairs over just one month would take an estimated 19425 days, or 53 years! We were able to reduce the computation time by creating scripts which sent out independent MATLAB jobs to a Sun Grid Engine scheduler. However, even when distributing jobs the computations would be prohibitively slow for large data sets. This solution still has problems, as the matrices are still not positive semi-definite, and more importantly does not allow for a tight interaction between independent pairs throughout the course of a trading day, which can be used to optimize certain aspects of the strategy.

Given the challenging task of analyzing market-wide correlation matrices, it seems apparent that a custom implementation integrated directly with the `MarketMiner` platform is necessary to achieve the desired scale and timing objectives. The potential advantage of a tight integration with `MarketMiner` is that the outputs from each strategy (trades decision) can be gathered by a master

process to perform additional tasks such as risk management and liquidity provisioning. Also, aggregating the results into a single basket, as opposed to many individual trade orders, allows the trading system to utilize a sophisticated list-based algorithm to optimize the actual execution of the trades. This approach was not pursued in this study but is the subject or further studies at Scalable Analytics, Inc.

8.2 Implementation Shortfall

The gap between the decision price (price at which we want to buy/sell a stock) and the final trade price (price that we actually pay for a stock) is referred to as "implementation shortfall". There are a number of components that contribute to implementation shortfall, including transactions costs (commissions), "moving the market" and "lost opportunity".

Transaction costs were briefly considered in Chapter 7 where we saw that this consideration might have implications on the performance of strategies that trigger many or few trades. However, our analysis was somewhat ad hoc, looking at the impact of transaction costs only after we ran our trading algorithm. A future direction would be to consider the cost of a transaction even before triggering a trade. This would require a modification of the canonical trading strategy we used in this study.

The term "moving the market" refers to the possibility that large enough orders of stocks will significantly affect the market price. It is common to assume that traders in financial markets as "price takers", in the sense that they can purchase as many stocks as they would like at the current price being offered. This may be true if the trader is only purchasing a small number of stocks, but may not be true for large purchases.

"Lost opportunity" is the possibility that a trader's order size for a stock is sufficiently large that there are not enough stocks willing to be sold at the given price to fill the order.

Our current analysis does not address these latter two issues. To include "moving the market" and "lost opportunities" would require more sophisticated modeling of the stock market. This would likely require a rethinking of our basic pairs trading algorithm.

8.3 Improvements to the Canonical Trading Strategy

The canonical pairs trading strategy in Chapter 4 is a simplified version of a strategy that may be used by professional traders. Access to the precise details of a working strategy is hard to come by because of proprietary concerns. An effective strategy can be quite lucrative for a trading company, and giving away the details of their approach is therefore an unwise business decision. Lack of access to detailed information on specific working strategies made it difficult to know exactly how correlation is used to trigger and close trades in practice.

Many practitioners in industry identify the pairs of stocks to trade by looking for historically correlated and fundamentally related stocks, and exploit the arbitrage opportunities by only looking at the spread of the two identified stocks. Profit is made by a basic buy-low-sell-high approach on the spread. The naïve buy-low-sell-high approach can be efficient if the timing to enter and exit trades is well chosen. However, choosing the ideal timing can be extremely difficult, and technical traders use various criteria to define and detect trading opportunities, such as buy if the spread is below x, and sell when the spread is above y. This problem of timing is magnified when traders trade intra-daily because of volatility in high frequency stock prices. Highly volatile prices can cause trading strategies to become overly sensitive and thus trigger poorly timed trades.

One of the primary issues with our pairs trading strategy relates to how we choose the entrance and exit criteria of a trade. We define entrance in terms of correlation drop, and initiate a trade when the correlation of the pair drops $d\%$ from its average. However, the exit criteria of the trade has to do with retracement of the difference in *prices* of the stocks; that is, the spread. This incongruence between entry (based on correlation) and exit (based on spread) criteria warrants further investigation. The connection between correlation change and change in spread was not explored deeply and we found in practice that there are occasions where our trading strategy was effectively "waiting" to make a loss based on this exit criteria. There is scope for further investigation as to why this situation occurs. One direction is to consider pairs trading strategies that use correlation calculations in both entry and exit criteria.

8.4 Parameter Tuning

Experimentation on our trading strategies showed that performance greatly depends not only on the correlation measure but also on the other parameters in our trading algorithm, such as the percentage correlation drop d or correlation window M. We did all of our analysis in this study based on an average over *all* parameter sets we considered, which is intended to give a measure of overall performance. However, this procedure may lead to bias. If we can have an algorithm to find a set of optimal parameter sets for each strategy and compare them under their optimal parameter sets, the results will be less biased because of the parameter choice. The procedure of finding these optimal parameters is termed "parameter tuning" and is important for many financial applications. A next direction for this research would be to explore "parameter tuning" to improve our conclusions when comparing trading strategies. One computational package for tuning parameters we are considering implementing is `ParamILS` which is currently being developed at the University of British Columbia [3].

Chapter 9

Conclusions

In this essay, we first compared the traditional correlation measure, Pearson correlation, with a robust correlation measure, Maronna correlation on high frequency financial data. We then explored the use of traditional and robust correlation measures in designing pairs trading strategies. Specifically, these correlation measures were used to design alternate trigger mechanisms which was used to trigger trades in a canonical pairs trading strategy framework. The trading algorithm was backtested on a set of historical data, and performance data was collected and analyzed statistically to reveal information about the effectiveness of alternate trigger mechanisms.

In Chapter 3 we observed that preliminary plots of stock prices and correlations suggest that there are some important differences between Pearson and Maronna correlation. Pearson correlation of log returns data is unstable due to noise in raw price data, and using filters smoothed the correlation efficiently, but may cause the Pearson measure to be sensitive to the choice of filter. On the other hand, Maronna correlation is generally more stable and smooth as time evolves.

We then constructed a canonical pairs trading strategy that demonstrated how to use correlation to trigger trades and also discussed how to decide on the position and criteria to exit trades. This canonical pairs trading strategy was based on feedback and input from professionals traders, and we hope accurately represents some aspects of the practice of pairs trading. However, as we also saw in Chapter 8 there are some elements of realistic practice not faithfully represented in our algorithm, which might be incorporated for more realism in later studies.

Using our canonical pairs trading strategy, we derived performance data by backtesting on historical data on a set of 61 stocks. We discussed some of the computational issues associated with backtesting, and advocated for a parallel method of computation to speed calculations and test on a larger set of stocks. Backtesting yielded several performance measures, which we then used to test the relative performance of our trigger mechanisms. We observed significant differences between strategies which use Pearson correlation to trigger trades and those which use Maronna correlation. Each has different strengths and weaknesses in terms of their risk versus return profiles. In particular, on average Pearson yields the highest returns with the higher risk in terms of variance and win-loss ratio. Combined performs the best in terms of Sharpe ratio and win-loss ratio, which indicates that it has the lowest risk. The performance of Maronna is somewhat in between of the the other two.

We were surprised by the significantly higher cumulative returns of Pearson trigger mechanisms as compared to Maronna. This was because in theory, robust correlation should give a better

estimate of the true correlation of highly volatile set of data. Therefore, we expect that trigger mechanisms based on Maronna correlation to detect break-downs of co-movement more accurately than Pearson, and thus brings higher returns with less risk. Nonetheless, this turns out not to the case in practice. One possible explanation of this is the phenomenon we saw in practice that Maronna lags in identifying divergence in correlation when compared to Pearson. Our intuition is that although Pearson is more sensitive to outliers, it also seems to be "quicker" to respond to price changes and thus on average enter trades at more favorable times. Thus, this observation that Maronna is less responsive than Pearson seems to outweigh the downside of calculating true correlation less accurately or over-reacting to outlying data, which are criticisms of Pearson's measure.

The effect of filtering the data was also pronounced when we considered our results. Some preliminary results we undertook on unfiltered data showed that Maronna faired better when their was less aggressive filtering or smoothing of the data when compared to Pearson. Thus, the trading industry's use of filters as opposed to robust correlation computational engines to deal with highly volatile and "dirty" data, seems to be partially justified by our analysis. Indeed, filters are simple and direct ways to "clean" data, whereas robust correlation measures like Maronna are more complex to implement. However, this also underscores the importance of having effective data filters in trading applications, and we propose that having a robust correlation measure like Maronna might be used as some benchmark or reference point when comparing the effectiveness of data filters.

We should mention, however, that robust correlation did yield attractive risk characteristics in terms of win-loss ratio and variance of returns when combined with Pearson correlation in our Combined trigger mechanism. We believe this to be an interesting finding and suggests that robust correlation has attractive features for risk averse traders. Overall, we believe that there is value for traders to explore robust correlation when undertaking automated trading and we hope this study can stimulate interest and point out promising directions for future research.

Bibliography

[1] J. Chilson, R. Ng, A. Wagner, and R. Zamar, "Parallel computation of high-dimensional robust correlation and covariance matrices," *Algorithmica*, vol. 45, no. 3, pp. 403–431, 2006.

[2] M. M. Dacorogna, R. Gencay, U. Muller, R. B. Olsen, and O. V. Olsen, *An Introduction to High Frequency Finance*. Academic Press, New York, 2001.

[3] H. H. H. F. Hutter and T. Sttzle, "Automatic algorithm configuration based on local search," in *AAAI '07: Proc. of the Twenty-Second Conference on Artifical Intelligence, 2007*, 2006, pp. 1152–1157.

[4] T. N. Falkenberry, "High frequency data filtering," 2002. [Online]. Available: http://www.tickdata.com/FilteringWhitePaper.pdf

[5] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs Trading: Performance of a Relative Value Arbitrage Rule," *SSRN eLibrary*, 2006.

[6] H. Green, B. Schmidt, and K. Reher, "Algorithms for filtering of market price data," *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pp. 227–231, Mar 1997.

[7] H. Lopuhaa and P. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *The Annals of Statistics*, pp. 229–248, 1991.

[8] R. N. M. M. Tumminello, T. Di Matteo, "Correlation based networks of equity returns sampled at different time horizons," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 55, no. 2, pp. 209–217, 2007.

[9] R. Maronna, "Robust M-estimators of multivariate location and scatter," *Annals of Statistics*, vol. 4, no. 1, pp. 51–67, 1976.

[10] C. Rostoker, A. Wagner, and H. H. Hoos, "A parallel workflow for real-time correlation and clustering of high-frequency stock market data," in *IPDPS*. IEEE, 2007, pp. 1–10.

[11] V. Skintzi, G. Skiadopoulos, and A.-P. Refenes, "The effect of misestimating correlation on value-at-risk," *Journal of Alternative Investments*, vol. 7, no. 4, pp. 66–82, 2005.

[12] R. E. Welsch and X. Zhou, "Application of robust statistics to asset allocation models," *REVSTAT – Statistical Journal*, vol. 5, no. 1, pp. 97–114, 2007.

[13] D. Wichern and R. Johnson, *Applied multivariate statistical analysis.* Pearson Prentice Hall, 2007.

Appendix A

The `MarketMiner` Platform

The explosive trend toward automated trading and the availability of tick data at sub-millisecond rates introduces new demands and opportunities which require quick online analysis and decision processing. `MarketMiner` is an ongoing research project that addresses this data analysis problem by supporting the computational workload associated with performing market-wide backtesting of trading strategies.

The original design of `MarketMiner` was a basic MPI-enabled pipeline for processing quote data [10], and has since been extended to support arbitrary directed acyclic graph (DAG) stream processing workflows. One of the strengths of MPI is that it is the de-facto standard for messaging-passing parallel programming and there are a large number of high quality open-source numerical libraries available. Certain analytics platforms such as MATLAB take advantage of these MPI libraries to implement their distributed computation toolkit. Unlike MATLAB, we are able to use these libraries in a far more tightly integrated fashion beyond the simple remote procedure call which is the basis of their system.

Given the requirements of a pairs trading strategy, the enabling feature of `MarketMiner` is its ability to handle a large amount of market-wide, high frequency "tick" data from a live feed or from a historical database, and use this data to produce large correlation matrices in an online fashion. The `MarketMiner` system also has the ability to compute Maronna correlation, which in general is computationally expensive and thus not commonly implemented in statistical software packages, especially those that operate on real-time data. The `MarketMiner` system overcomes this difficulty by implementing a parallel algorithm for computing robust correlation matrices [1]. The original work investigated its scalability as an offline algorithm, and more recently in an online setting [10].