Pairs Trading using cointegration in pairs of stocks

Manda Raghava Santosh Bharadwaj

A Research Project Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Finance

Saint Mary's University

Written for MFIN 6692 under the Direction of
Dr. J.Colin Dodds

Approved by: Dr. J.Colin Dodds

Faculty Advisor

Approved by: Dr. Francis Boabang

MFIN Director

Date: 9-Sep-2014

Acknowledgements

I hereby express my sincere gratitude to my supervisor Dr.J.Colin Dodds as well as MFIN Director Dr. Francis Boabang for giving me an opportunity to work on this project and gain valuable experience and learn many new things. I would also like to thank my friends for engaging in informal discussions at various stages of the project helping me to consolidate ideas.I would also like to thank the professors from my graduate studies while at Saint Mary's University. The knowledge and experience I gained have been invaluable.

Abstract

Pairs Trading strategy using co-integration in pairs of stocks

by Manda Raghava Santosh Bharadwaj

The aim of this project is to implement pair trading strategy, which aims to generate profits in any market conditions by examining the cointegration between a pair of stocks. Pair Trading, also known as a relative spread trading, is a strategy that allows a trader to benefit from the relative price movements of two stocks. A trader can capture the anomalies, relative strength or fundamental differences in the two stocks to create profit opportunities. Pair Trading primarily involves finding correlated stocks and exploiting the volatile market conditions, which lead to a diversion in their correlation. A trader takes a short position in one stock and simultaneously takes a long position in the other. If the market goes down, the short position makes money. On the other hand, if the market goes up, the long position makes money. Creating such a portfolio enables the investor to hedge the exposure to the market. Furthermore, by taking a long-short position on this pair, when prices diverge, and then closing the position when the spread retreats to its mean or a threshold, a profit is earned.

In this project, we implement pair trading strategy using an Ornstein-Uhlenbeck (OU) process based spread model, is applied on stocks from three different sectors-Energy, HealthCare and Banking of the NYSE. Stocks were selected based on a combination of Distance Test, ADF Test and Granger-Causality Test. The paper concludes by summarizing the performance of this strategy and offers possible future enhancements and applying it to more complex scenarios.

Date: 9-Sep-2014

Table of Contents

Chapter 1: Introduction 1
1.1 Purpose of Study 1
1.2 Background 1
1.3 Need for study 2
1.4 Statement of Purpose 3
Chapter 2: Literature Review 4
Chapter 3: Methodology 6
3.1 Selection of the Trading Universe 6
3.2 Selection Strategies 9
3.3 Model and Parameter Estimation 11
3.4 Trading Algorithm 12
3.5 Back-testing 13
Chapter 4: Results 19
4.1 Risk Management: 22
Chapter 5: Conclusions & Further work 23
5.2 Further Work 24
References 25
Appendix A: Strategy Code 28
Appendix B: Stocks used in strategy 42

Chapter 1: Introduction

1.1 Purpose of Study

Market-neutral equity trading strategies exploit mispricing in a pair of similar stocks. Mispricing is more usual in a global financial crisis. Therefore, more possibilities emerge at bad times. Moreover, there are fewer market participants, which reduce competition. Therefore, it is not surprising that market-neutral trading over-performs during most severe market conditions.

The aim of this project is to implement pair trading strategy, which aims to generate profits in any market conditions by examining the cointegration between a pair of stocks.

1.2 Background

The fundamental idea of pair trading comes from the knowledge that a pair of financial instruments has historically moved together and kept a specific pattern for their spread. We could take advantage of any disturbance over this historic trend. The basic understanding of pair trading strategy is to take advantage of a perturbation, when noise is introduced to the system, and take a trading position realizing that the noise will be removed from the system rather shortly.

It involves picking a pair of stocks that typically move together, and deviate from their co-integrated behaviour during small time intervals. The pairs are selected based on our co-integration framework. Once the pairs are selected, we monitor for any deviation from the stationary behaviour of the spread. Then, once the price starts diverging we short the winner and buy the losing stock. Finally, we close the position when the price starts converging and profit from the mean-reversion of the spread.

This strategy was pioneered by Nunzio Tartaglia’s quant group at Morgan Stanley in the 1980’s. Tartaglia formed a group of mathematicians and computer scientists and developed automated trading systems to detect and take advantage of mispricing in financial markets to generate profits. One such strategy was Pairs trading and it became one of the most profitable strategies developed by this team. With the team gradually spreading to other organizations, so did the knowledge of this strategy.

1.3 Need for study

With the innovation in financial markets, new instruments and securities have made financial markets more complex and the uncertainties and risk associated with the markets have exponentially increased. It is no longer easy for an uninformed investor to create diversified portfolio as inherent risks associated with the securities are quite intricate to assess.

Statistical arbitrage techniques have become increasingly famous in their use as they are dependent on trading signals and are not driven by fundamentals, and information is easily accessible to implement a strategy. Pairs trading is one such strategy and has become famous because of the simplicity in its basic form. Moreover, it is a combination of short and long positions making it a self-financing strategy. Hence, such a strategy modified to be flexible in current market conditions would provide investors with a valuable tool in aiding toward their investment analysis and making decisions. In this project, a systematic approach to Pairs Trading is designed using the existing mathematical models and implemented on market data to examine its performance.

1.4 Statement of Purpose

Though Pairs trading is classified as a market-neutral and a statistical arbitrage strategy, it is not risk-free. Moreover, though the strategy has evolved significantly, various models in this have limitations and disadvantages along with their uses. So a more robust, uniform analytical framework is needed to be designed and implemented. This project presents a systematic approach to Pairs Trading using a combination of existing models for this strategy.

The next parts of the paper are organized as follows. Chapter 2 discusses a brief review of available literature on this topic. Chapter 3 outlines the methodology adopted, data selected and trade algorithm developed. The subsequent chapters comprise the results and conclusions.

Chapter 2: Literature Review

Though Pairs Trading strategy has been in existence for about three decades, it has not been extensively researched. This could mainly be attributed to its proprietary nature. But considerable strides have been made in the development of this strategy from being a simple trading strategy into a comprehensive quantitative model capable of being applicable to wide range of securities across complex market scenarios. Major referenced works in this area include Gatev et al(1999 and 2006), Vidyamurthy (2004), and Elliott et al(2005).

The paper by Gatev et al is an empirical piece of research which uses a simple standard deviation strategy and shows pairs trading after costs can be profitable. This is shown by testing this strategy with daily data over 1962-2002.They used the "Minimum Distance" method to select stock pairs, where distance is measured as the sum of squared differences of normalized price series. The results show an average annualized excess return up to 11 percent clearly exceeding the typical estimates of transaction costs and hence inferring that the strategy is profitable. Nath (2003) modified this method by adding a trigger that when distance crosses the 15 percentile, a trade is entered for that pair, and accounted for risk control by limiting trading period at the end of which positions have to be closed out regardless of the results. In addition, he adds a stop-loss trigger to close the position whenever the distance increases to the 5 percentile value. Though, this model is purely statistical and has its advantage in being free from mis-specification, being a static model and assuming the price level is static through time causes limitations in its use.

Vidyamurthy(2004) suggested a co-integration based approach to select the pairs of stocks in an attempt to parameterize pairs trading. He reasoned that as the logarithm of two stock prices are typically considered to be non-stationary; there is a good chance that they will be co-integrated. In that case, cointegration results can be used to determine how far the spread is from the equilibrium value thereby quantifying the mispricing and implementing the strategy based on this information.

Elliot et al presented a stochastic spread model to describe the mean reversal process and estimated a parametric model of the spread thereby overcoming the weakness of Minimum Distance method.

In the case of Do et al(2006) , they conducted a comprehensive analysis of all existing methods in detail and formulated a general approach. In their own words:

"*This paper analyzes these existing methods in detail and proposes a general approach to modeling relative mispricing for pairs trading purposes, with reference to the mainstream asset pricing theory. Several estimation techniques are discussed and tested for state space formulation, with Expectation Maximization producing stable results."*

{Page 1}

This project follows a similar approach by combining a few of the methods from the literature, thereby forming a uniform algorithm for Pairs Trading. It implements this algorithm on stock data for the three sectors: banking, healthcare and energy from the New York stock exchange market

Chapter 3: Methodology

The following steps are performed as a part of implementing the strategy:

- Selection of Asset Type
- Stocks Selection based on Distance Test, ADF Test, Granger-Causality Test
- Parameter estimation for the Ornstein-Uhlenbeck model
- Implementation of code to enter and exit the position based on stock prices
- Perform back-testing on in sample data and run final code on out of sample data

3.1 Selection of the Trading Universe

To obtain accurate results, the strategy must be implemented keeping certain things in mind. This strategy is sector neutral. A pair in our strategy always belongs to the same sector. This is done so because pairs from different sectors are highly susceptible to unpredictable sector-specific variations and the co-integration of the pair can be lost in the process. For implementation, I chose three sectors.

- Banking
- Healthcare
- Energy

From the New York stock exchange, 15 stocks* were chosen from each sector and we chose pairs among them using our selection strategies.

Table 3.1: Stocks chosen from each sector

Energy	Banking	Healthcare
ACI	BAC	ABT
AEM	BNS	JNJ
BTU	C	LLY
CAM	CM	PFE
CNQ	CS	AMGN
CNX	DB	AZX
FST	FITB	BAX
COG	HBC	BMY
GG	HDB	GSK
HAL	IBN	NVO
NOV	MS	PFE
OXY	PNC	RHHBY
PEO	RBS	SNY
SLB	RY	WCRX
TLM	WFC	MTEX

** Full description of stocks is available in Appendix B.*

The data have been selected with careful consideration of the nature of stocks in each sector and highly dynamic relationship.

There are primarily two ways to select the data:

- Constant Universe
- Dynamic Universe

Constant pairs can be selected based on the initial co-integration and we can keep trading on them throughout the trading window. However, it was observed that this may lead to selection-bias because stocks may come in and out of the universe of stocks. Also this strategy assumes that the co-integration of the stocks remains same. But during the implementation of the algorithm co-integration was found to be highly dynamic. Hence I looked at another strategy where we selected dynamic pairs based on a rolling window to check for co-integration. I took all the stocks in the universe without using any future information. For example, stocks may come in and go out of the universe that will not significantly affect the algorithm. Hence, this strategy has an advantage of being a forward-looking algorithm as well as removing the selection bias.

The in-sample data used were from 2001–2005 for testing the algorithm and optimization of parameters. For selection of the pairs for in-sample data, I started with a 5 year rolling window from 1996. The out-of-sample data were from 2006–2010. Finally, to keep the portfolio sector neutral, I looked at banking, health care, and energy.

3.2 Selection Strategies

To select pairs from our universe of stocks, three tests were applied: Distance Matrix, Augmented Dickey-Fuller (ADF) Test, and Grander Causality. Pair is selected if it passes all the three tests.

Minimum Distance Method

The distance matrix looks for the historical price movements between pairs. The stocks should have similar price movements. The main idea is to select pairs that have had similar historical price moves. According to law of one price theory (Coleman, 2009), similar securities would have similar prices. To start the process, it is assumed that all the prices are equal to 1.00 for the starting day. Then, a cumulative return index is generated for all stocks. To select pairs from this data set, the sum of squared deviations is used:

$$\boldsymbol{\gamma} = \sum_{t=1}^{N} \left(C_{x(t)} - C_{y(t)}\right)^2 \quad \text{………………..} \quad 3.1$$

where γ =distance;
$C_{x(t)}$ = Normalized cumulative return index of stock *x* over time t;
$C_{y(t)}$ = Normalized cumulative return index of stock *y* over time t.

ADF Test

In the ADF test, the ratio between two stocks must have constant mean and volatility. It also tests for unit root in the stocks returns and checks for stationarity. In order to generate a profit in a pair-trade, the ratio of the prices, R_t, needs to have both a constant mean and a constant volatility over time. For an autoregressive process AR(1) such as $\delta X_t = (\varphi - 1)X_{t-1} + \varepsilon_t$, and defining $a = \varphi_1 - 1$, the unit root test can be written as follows

Null Hypothesis: $H_0 : a = 0$

Alternate Hypothesis: $H_1 : a < 0$

The number of lagged difference terms to include is determined empirically, the idea being to include enough terms so that the error term in the tested equation is serially uncorrelated.

ADF Test Combined with Two-way Granger Causality

The Granger causality test determines if price of one stock can be used to predict another. Our top concern is the risk that one takes when entering a pair-trade, which is the possibility of a structural breakdown of the mean-reverting-price-ratio property.

Because there were too many pairs that passed the ADF test, and because some of the selected pairs did perform poorly the year after they were selected, we decided that we needed additional testing. This is where the Granger causality test in *both* directions comes in.

As mentioned above, our pairs will be selected dynamically year over year. Below is result of all three tests for healthcare sector. It shows how pairs are changing from 2006-2010.

Sector Neutrality and Beta Neutrality

Stocks from three different sectors were studied for selecting pairs. Successful pairs trading must have a portfolio that is sector and beta neutral. To avoid sector bias, pairs from different sectors were considered (banking, healthcare, and energy).

To avoid beta bias, stock pairs with similar market exposure or beta were selected. Also, stocks with a beta less than or equal to 0.1 were chosen to ensure least correlation with the market.

3.3 Model and Parameter Estimation

For spread modeling, Ornstein-Uhlenbeck (OU) was used. It is a Stochastic Spread Method to model the spread between the two stocks in a pair. This model can be viewed as the continuous time version of the discrete time AR (1) process. It satisfies the following stochastic differential equation:

$$dX(t) = (a - bX(t))dt + \sigma dW(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 3.2$$

The Process reverts to μ = a/b with strength b and the above equation can be written as

$$X_{k+1} - X_k = (a - bX_k)\tau + \sigma\sqrt{\tau}\epsilon_{k+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 3.3$$

$$X_{k+1} = A + BX_k + C\varepsilon_{k+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 3.4$$

where

$$A = a\tau \geq 0,\ 0 \leq B = 1 - B\tau < 1,\ \text{and} \quad C = \sigma\sqrt{\tau} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.5$$

There are three known methods to estimate the parameters for A, B and C:

Method of Maximum Likelihood Estimation (MLE)

Method of Moments (MOM)

Least Squares Method (LSM)

Maximum Likelihood method has been used to estimate the parameters because of its consistency even in situations when the data are not normally distributed which is not the case with Least squares method.

3.4 Trading Algorithm

The algorithm for trading enters a trade when a pair of stocks deviates significantly from its co-integrated behaviour (2 standard deviations from stationary mean).

For the time frame, a 5-year rolling window was used to check for the long-term co-integration behaviour of a pair. To check for short term changes in co-integration a 120 day rolling window was used.Pairs come in and out based on the behaviour of the universe. For example, Enron bankruptcy would result in it leaving the universe and Google IPO would represent a stock coming in the universe after it had become co-integrated to other tech stocks.

If stocks in the pair continue to diverge, and do not revert back to the mean, we stop the trading after a control-window of 40 days. The parameters of the model have been optimized by running simulations for different rolling and control windows. Optimization is performed to get the parameters for a maximum Sharpe ratio and net profit.

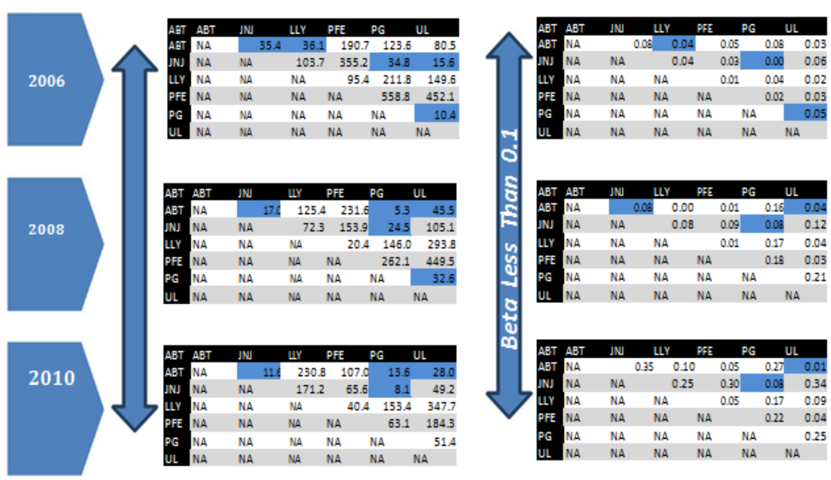
3.5 Back-testing

The back-testing prototype was built based on the trading rules and risk management strategy as discussed in the previous sections. It is built fully in MATLAB, all the pairs trading, account balance updating and parameters estimation are programmed in MATLAB.

<u>Pair selection:</u>

The pairs for in sample data were selected based on data from 1-Jan 1995 to 31-Dec 2000. Because we are assuming a dynamic universe, this work was done every year and new pairs were generated for the next year of trading. Pair selection for the out of sample was based on data from 2001 to 2005 and pairs were newly generated for every subsequent year of trading. The results for the healthcare sector are illustrated below in Figures 3.1 and 3.2.

<u>Figure 3.1: Beta and ADF Test results for Healthcare Sector</u> - 2006-2010



2006:

ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	35.4	36.1	190.7	123.6	80.5
JNJ	NA	NA	103.7	355.2	34.8	15.6
LLY	NA	NA	NA	95.4	211.8	149.6
PFE	NA	NA	NA	NA	558.8	452.1
PG	NA	NA	NA	NA	NA	10.4
UL	NA	NA	NA	NA	NA	NA

ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.08	0.04	0.05	0.08	0.03
JNJ	NA	NA	0.04	0.03	0.00	0.06
LLY	NA	NA	NA	0.01	0.04	0.02
PFE	NA	NA	NA	NA	0.02	0.03
PG	NA	NA	NA	NA	NA	0.05
UL	NA	NA	NA	NA	NA	NA

2008:

ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	17.0	125.4	231.6	5.3	45.5
JNJ	NA	NA	72.3	153.9	24.5	105.1
LLY	NA	NA	NA	20.4	146.0	293.8
PFE	NA	NA	NA	NA	262.1	449.5
PG	NA	NA	NA	NA	NA	32.6
UL	NA	NA	NA	NA	NA	NA

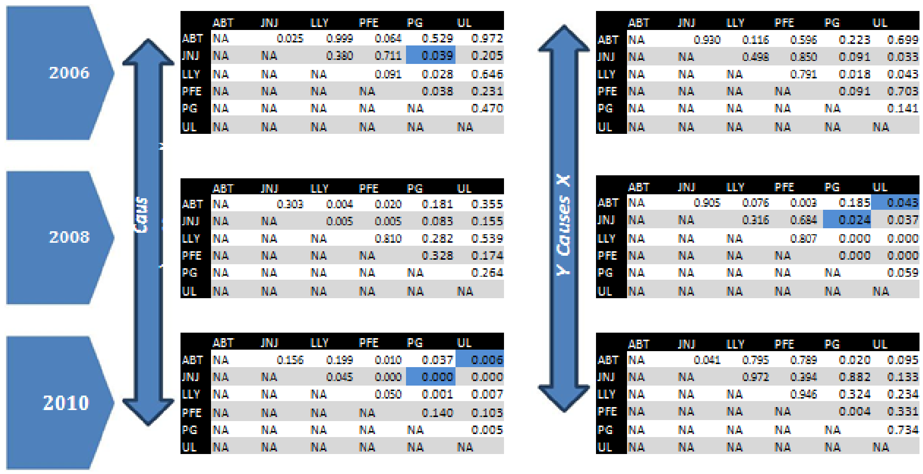
ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.08	0.00	0.01	0.16	0.04
JNJ	NA	NA	0.08	0.09	0.08	0.12
LLY	NA	NA	NA	0.01	0.17	0.04
PFE	NA	NA	NA	NA	0.18	0.03
PG	NA	NA	NA	NA	NA	0.21
UL	NA	NA	NA	NA	NA	NA

2010:

ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	11.6	230.8	107.0	13.6	28.0
JNJ	NA	NA	171.2	65.6	8.1	49.2
LLY	NA	NA	NA	40.4	153.4	347.7
PFE	NA	NA	NA	NA	63.1	184.3
PG	NA	NA	NA	NA	NA	51.4
UL	NA	NA	NA	NA	NA	NA

ABT	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.35	0.10	0.05	0.27	0.01
JNJ	NA	NA	0.25	0.30	0.08	0.34
LLY	NA	NA	NA	0.05	0.17	0.09
PFE	NA	NA	NA	NA	0.22	0.04
PG	NA	NA	NA	NA	NA	0.25
UL	NA	NA	NA	NA	NA	NA

Figure 3.2: ADF and Granger Test at 95% confidence limit for Healthcare sector – 2006-2010



	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.025	0.999	0.064	0.529	0.972
JNJ	NA	NA	0.380	0.711	0.039	0.205
LLY	NA	NA	NA	0.091	0.028	0.646
PFE	NA	NA	NA	NA	0.038	0.231
PG	NA	NA	NA	NA	NA	0.470
UL	NA	NA	NA	NA	NA	NA

	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.930	0.116	0.596	0.223	0.699
JNJ	NA	NA	0.498	0.850	0.091	0.033
LLY	NA	NA	NA	0.791	0.018	0.043
PFE	NA	NA	NA	NA	0.091	0.703
PG	NA	NA	NA	NA	NA	0.141
UL	NA	NA	NA	NA	NA	NA

	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.303	0.004	0.020	0.181	0.355
JNJ	NA	NA	0.005	0.005	0.083	0.155
LLY	NA	NA	NA	0.810	0.282	0.539
PFE	NA	NA	NA	NA	0.328	0.174
PG	NA	NA	NA	NA	NA	0.264
UL	NA	NA	NA	NA	NA	NA

	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.905	0.076	0.003	0.185	0.043
JNJ	NA	NA	0.316	0.684	0.024	0.037
LLY	NA	NA	NA	0.807	0.000	0.000
PFE	NA	NA	NA	NA	0.000	0.000
PG	NA	NA	NA	NA	NA	0.059
UL	NA	NA	NA	NA	NA	NA

	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.156	0.199	0.010	0.037	0.006
JNJ	NA	NA	0.045	0.000	0.000	0.000
LLY	NA	NA	NA	0.050	0.001	0.007
PFE	NA	NA	NA	NA	0.140	0.103
PG	NA	NA	NA	NA	NA	0.005
UL	NA	NA	NA	NA	NA	NA

	ABT	JNJ	LLY	PFE	PG	UL
ABT	NA	0.041	0.795	0.789	0.020	0.095
JNJ	NA	NA	0.972	0.394	0.882	0.133
LLY	NA	NA	NA	0.946	0.324	0.234
PFE	NA	NA	NA	NA	0.004	0.331
PG	NA	NA	NA	NA	NA	0.734
UL	NA	NA	NA	NA	NA	NA

After running all the three tests, we get below pairs in each sector for 2006, 2008 and 2010 as shown in Figure 3.3.

Figure 3.3: Pairs selected for the trading algorithm

	Banking	**Healthcare**	**Energy**
2006	DB-CS HBC-C HBC-DB PNC-C WFC-BAC	PG-J&J	NOV-CNX CNQ-BTU ACI-TLM
2008	CS-CM DB-CS PNC-HBC	ABT-UL PG-JNJ	OXY-FST PEO-FST
2010	CS-CM WFC-PNC	UL-ABT PG-JNJ	CNQ-CAM COG-CAM COG-CNQ OXY-COG

3.6 In Sample testing

The algorithm was tested on in-sample data from 2001 to 2005. We used this period to find out optimized values for several of the parameters. Pairs selected from previous 5 years were traded for the subsequent year.

Below Figure 3.4 shows a few of the pairs which were used in our trading algorithm. It can be seen that the individual stocks move together for long periods, however deviate from their co-integrated behaviour during some small time-windows.

Figure 3.4: Movement of few pairs from our Universe

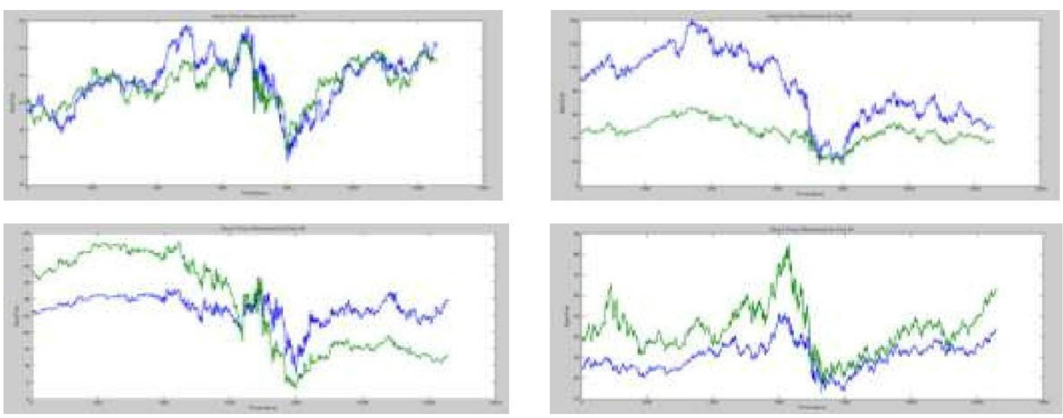
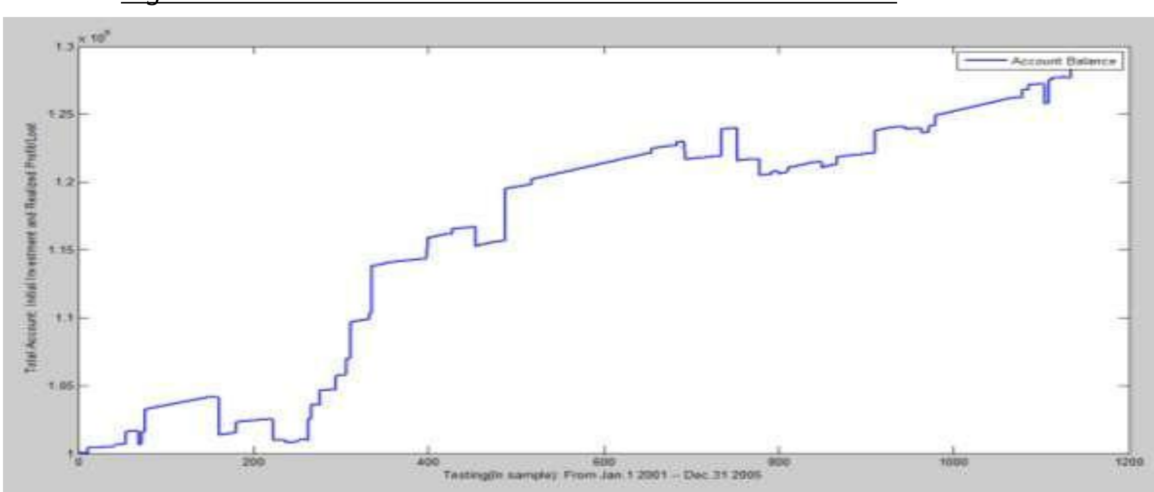


Figure 3.5: Total Account – Initial Investment + Realized Returns

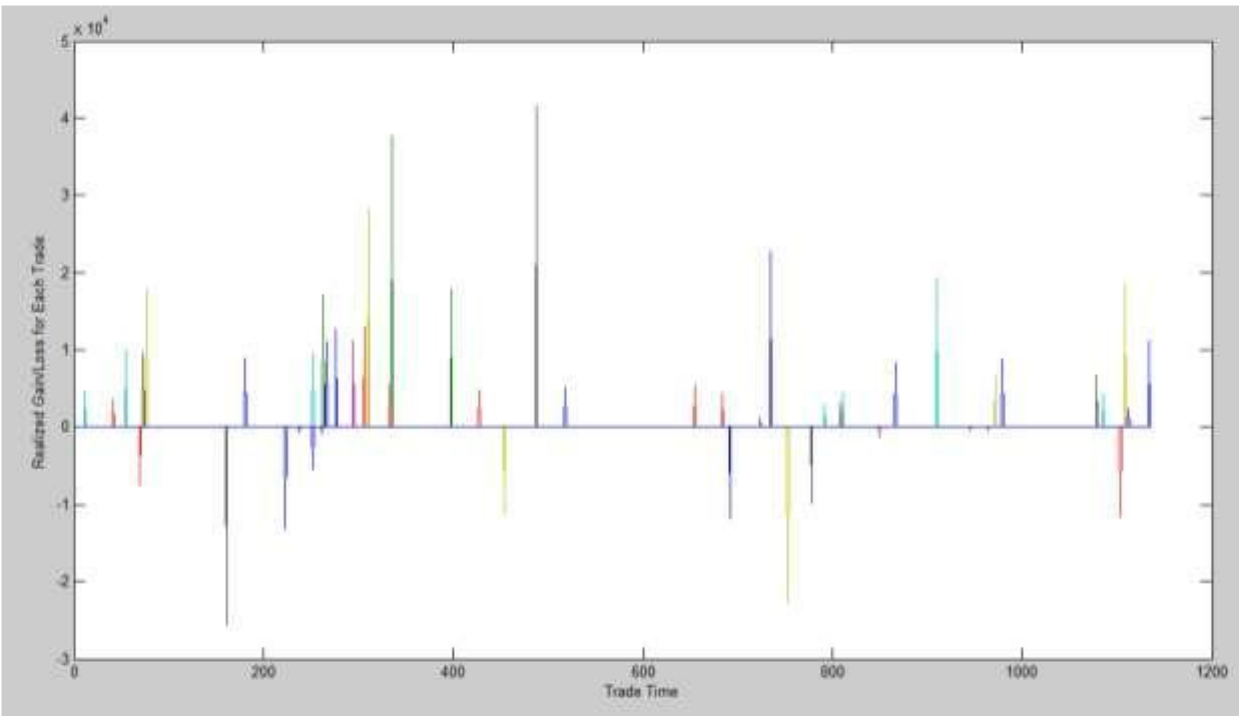


Returns from Pairs Trading Algorithm: 5.2%

Returns from S&P 500: -.55%

We obtain significantly better results than our benchmark

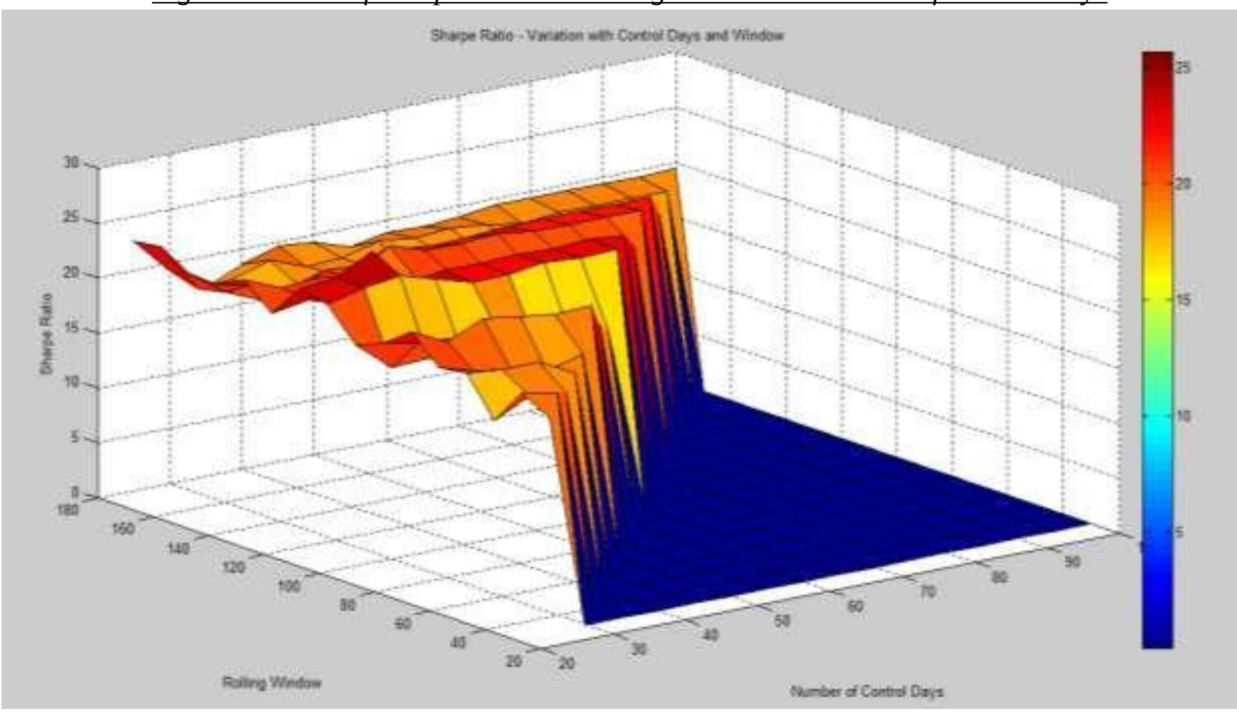
Figure 3.6: Realized gain/loss for each trade vs. trade time



Optimization of Parameters: After creating the initial algorithm for the strategy, further back-testing was performed for optimizing the parameters of the model. Two key time-windows in our model are the control-windows, which take care of the stop-loss strategy and rolling window, during which we check the short-term co-integration of the pairs. Hence, optimizing these two time-windows gave the maximum Sharpe Ratio and Risk-adjusted Return. The optimal parameters were fixed after the back-testing results were obtained. Same parameters were used to obtain final results for our out-of-sample data.

Figure 3.7 shows the plot for the variation of Sharpe Ratio with respect to rolling window and control window.

Figure 3.7: Plot of Sharpe Ratio vs. Rolling Window vs. Number of Control Days



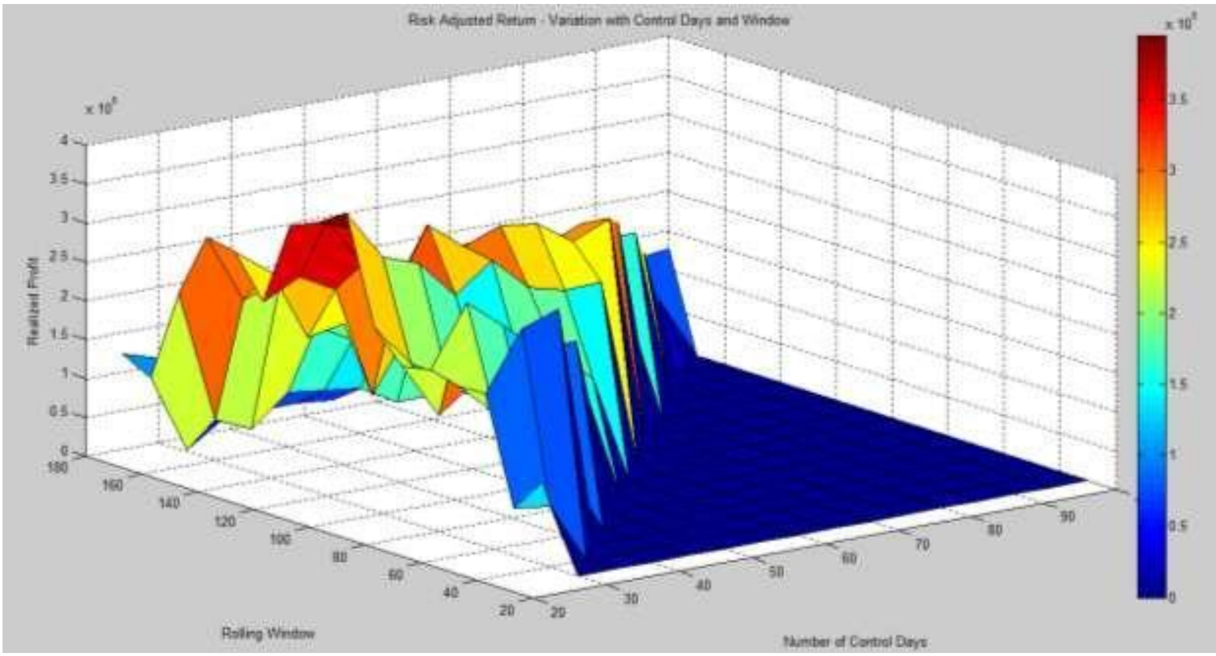
Optimal Parameters:

Control-window: 40 days

Rolling Window: 120 days

Further, Figure 3.8 shows the variation of Risk-adjusted Return on Capital with rolling window and control-window. However, the parameters were chosen based on the maximization of the Sharpe Ratio.

Figure 3.8: Plot of Risk Adjusted Return vs. Rolling Window vs. Number of Control Day

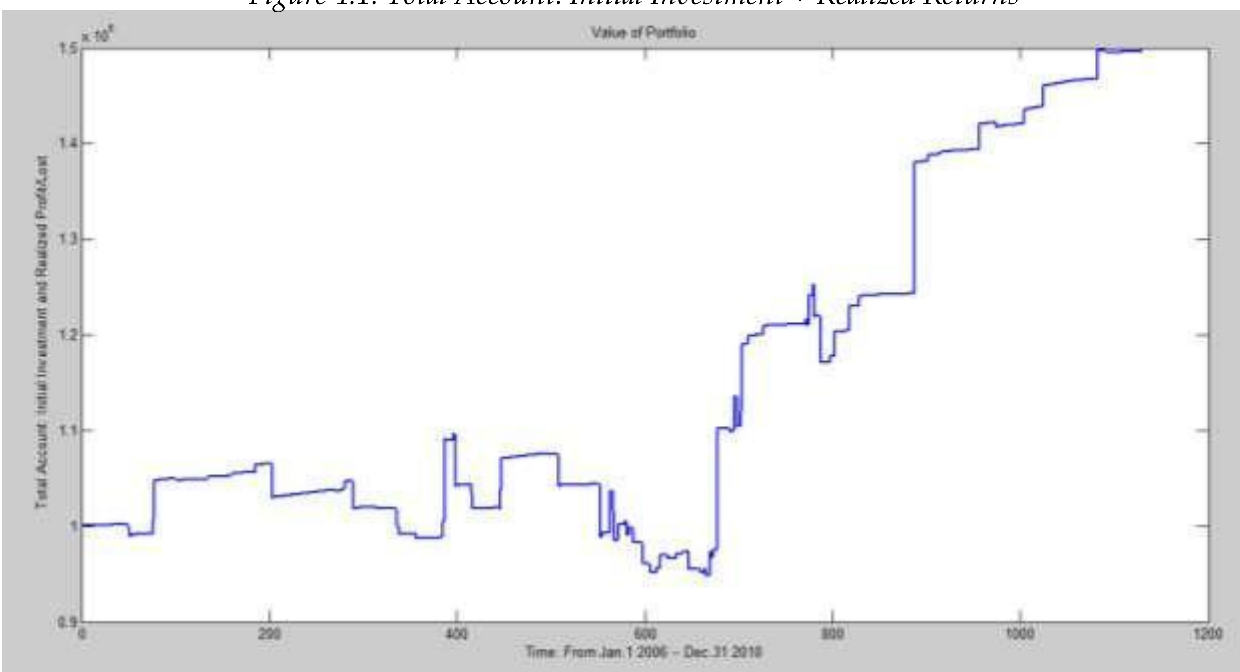


After the in-sample testing has concluded successfully, the algorithm was implemented on the out of sample data and the results and further discussions are included in the subsequent chapters.

Chapter 4: Results

Our out of sample window was from 2006-2010. We ran our algorithm in this period based on parameters optimized using in-sample data.

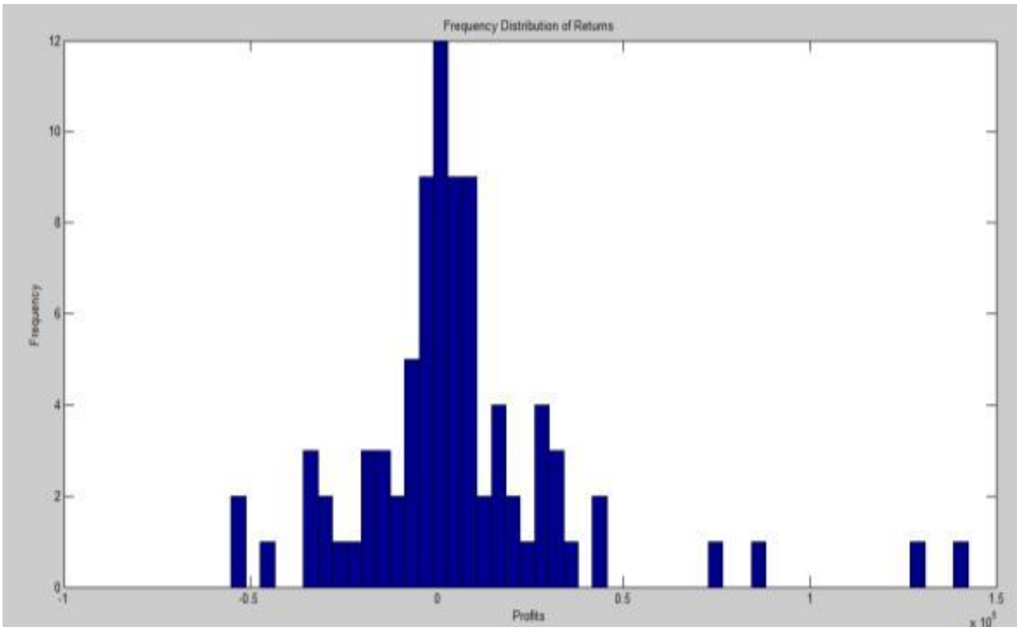
Figure 4.1: Total Account: Initial Investment + Realized Returns



Returns from Pairs Trading Algorithm: ~8.4%

Returns from S&P 500: -0.18%

Figure 4.2: Frequency Distribution of Returns



From the above figure, we can see that returns from each time period are mostly distributed around the mean return value and abnormal returns are very rare and hence do not affect the mean. Also, we can observe that there are very few cases of abnormal negative returns which indicates our strategy has been successful in hedging. The next two figures (Figures 4.3 and 4.4) show the performance of each pair and realized gain/loss at different points of time. From the first we can clearly see that almost all of the pairs yield positive returns. The second figure shows realized gain/loss during each trade.

Figure 4.3: Net Profit Loss for each Pair

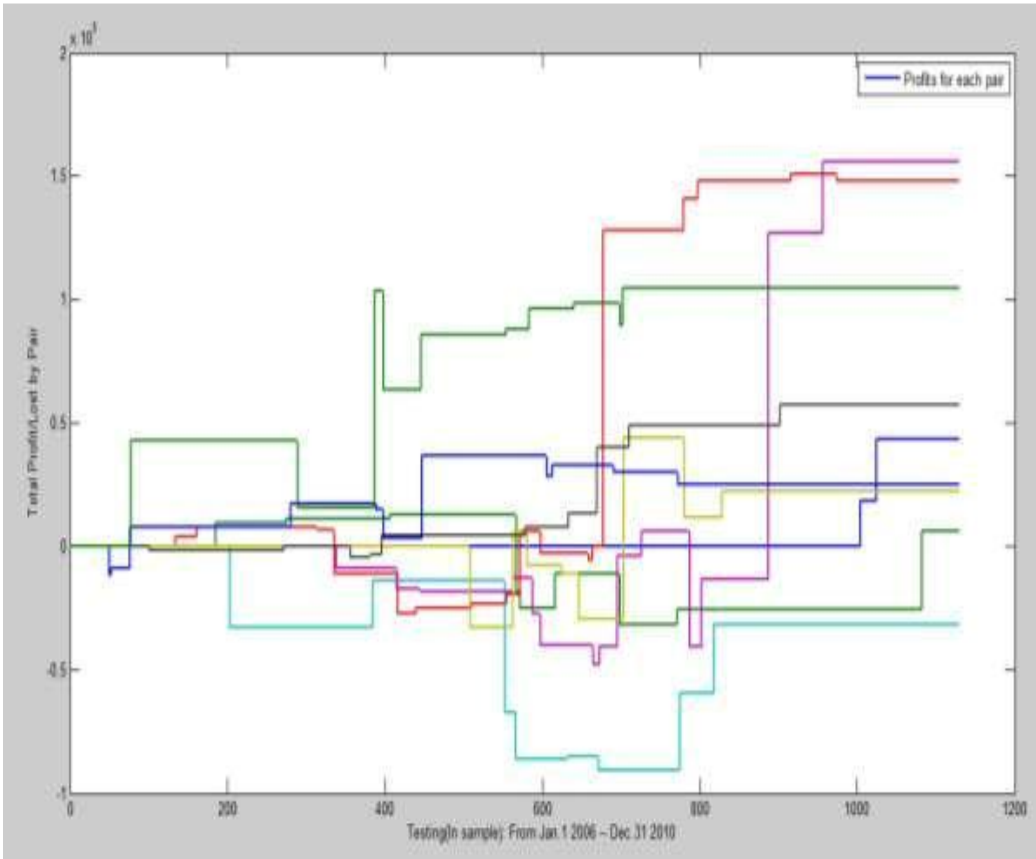
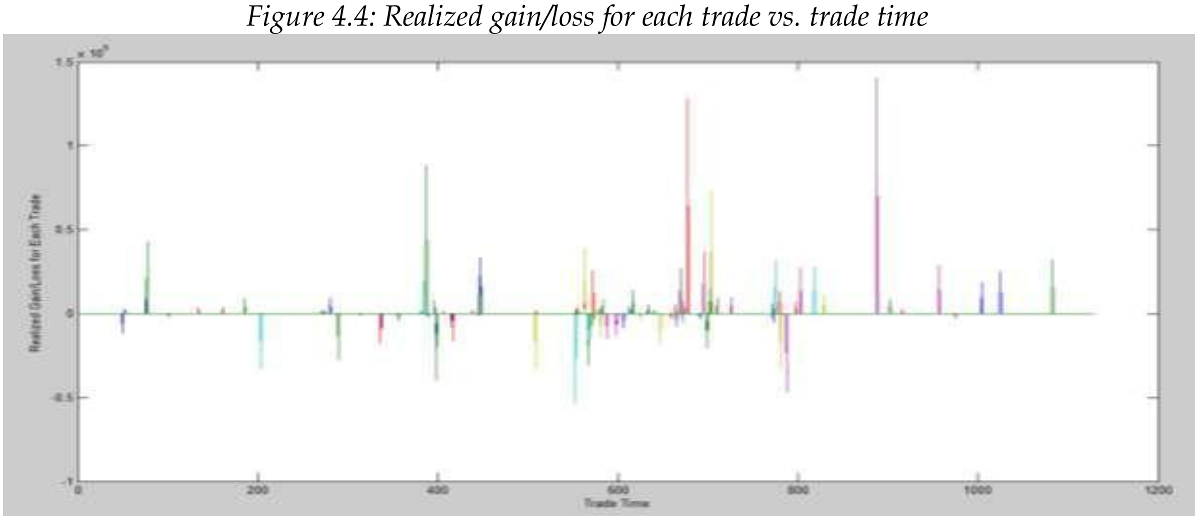


Figure 4.4: Realized gain/loss for each trade vs. trade time



4.1 Risk Management:

Risk management has become significantly important in recent years and our strategy has implemented a few methods to minimize the risk. The selection of pairs is sector-neutral and bet-neutral as stated before. Stop loss strategy has been used that will close the trade after 40 days if the trade has not yet been closed. We also will close out an open position if the spread between two stocks continues to deviate instead of converging beyond a certain threshold. This sometimes happens if there is change in the fundamental behaviour within the pair. In addition we calculate the Sharpe ratio to measure the risk adjusted performance. It is computed through the following formula:

$$S = \frac{E[R_a - R_b]}{\sigma} = \frac{E[R_a - R_b]}{\sqrt{\mathrm{var}[R_a - R_b]}}, \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad 4.1$$

Our portfolio was optimized to obtain the maximum Sharpe ratio, which was nearly 4.18. Finally, to get an understanding of how much our portfolio can lose during 10 days with 99 percent probability we calculated the value at risk (VAR) to be $15,422.00. The conditional value at risk (CVar) was also computed to get the expected loss greater than Var: $20,676.00. These metrics help us to understand how much our portfolio stands to lose.

Chapter 5: Conclusions & Further work

5.1 Conclusions

After conducting these experiments, we have concluded that changing pairs dynamically helps us in removing selection biases. The strategy makes about 8.4 percent profit per annum for the 5 year period. While the percentage of profit is not very high, the time frame includes the market crash during the subprime crisis. The strategy also out performs the S&P 500, which made -0.18 percent per annum for the 5 year period. Since we implemented a low risk strategy with small positions, profits can be increased by implementing a more risky strategy with larger positions. By using a dynamic universe, we were able to remove the selection bias, and our trading algorithm was forward looking without using any future information. Finally, the in and out of sample testing helped to create a profitable low risk strategy during one of the biggest crashes of the US Equities market. Hence, our algorithm has been successful in achieving a positive return with considerably less risk consistently for a period of four to five years.

5.2 Further Work

Pair Trading is slowly but surely evolving as a highly flexible strategy inviting almost endless possibilities for improvements and variants. Looking forward, our strategy could be implemented on higher frequency data such as tick-data or minute data. We could select pairs of stocks from different market sectors or even different markets. To make the strategy even more dynamic we could optimize the parameters based on the performance of the algorithm till date.

Presently, there are newer tests of co-integration that are more precise such as the KPSS Test and Johansen’s Test. Another idea would be to not have balanced long and short positions but to weigh them according to the current market behaviour: in a up (down) trending market the long (short) position would be larger in the expectation that the two assets will converge at a higher (lower) price. While in a stable market long and short would be roughly equal. Finally, we can model the risk and performance of a strategy using alternative ways that can incorporate skewed distributions.

References

Avellaneda,M., & Lee,J.-H.(2008). Statistical Arbitrage in the U.S. Equities Market. Social Science Research Networks.

Bakshi, G. and Z. Chen (1997), “Stock Valuation in Dynamic Economies,” working paper, Ohio State University.

Baronyan ,S and Ener,S. (2010) "Investigation Of Stochastic Pairs Trading Strategies Under Different Volatility Regimes", The Manchester School, Vol. 78, Issue s1, pp. 114-134

Do. B , Faff, R and Hamza, K (2006) A New Approach to Modeling and Estimation for Pairs Trading, Working Paper, Monash University.

Elliott, R., van der Hoek, J. and Malcolm, W. (2005) “Pairs Trading”, Quantitative Finance, Vol. 5(3), pp. 271-276.

Engle, R. and Granger, C. (1987) “Co-integration and Error Correction: Representation, Estimation, and Testing", Econometrica, Vol. 55(2), pp. 251-276.

Finch S, Ornstein-Uhlenbech Process , Unpublished Note

Gatev, E; Goetzmann, W. and Rouwenhorst, K. Rouwenhorst (2006), "Pairs Trading: Performance of a Relative-Value Arbitrage Rule", The Review of Financial Studies I, Vol. 19, No. 3, pp. 797-827.

Gatev, E; Goetzmann, W. and Rouwenhorst, K. (1999) "Pairs Trading: Performance of a Relative Value Arbitrage Rule", Unpublished Working Paper, Yale School of Management.

Lim, G and Martin, V. (1995) "Regression-based Cointegration Estimators", Journal of Economic Studies, Vol. 22(1), pp. 3-22.

Nath, P. (2003) "High Frequency Pairs Trading with U.S Treasury Securities: Risks and Rewards for Hedge Funds", Working Paper, London Business School.

Recorded Webinar: Cointegration and Pairs Trading with Econometrics Toolbox: Available on Mathworks

Vidyamurthy, G. (2004) Pairs Trading, Quantitative Methods and Analysis, John Wiley & Sons, Canada

Appendix A: Strategy Code

Code.m: File which calls Pair Trading main function.

PairsTrading.m: It calls all the other function.

SimulateOrnsteinUhlenbeck.m: Simulates OU estimates.

Spreads_Calculation.m: Calculates spread.

OU_est.m: Calculates OU estimates.

Output.m: Generates Output

SelectionStrategies.r: Checks all tests for selection process.

Code.m

```
clear all;
close all;
load Pairs.mat;

Stock_Price=Pairs_Price_Matrix;
Stock_Price = flipud(Stock_Price);
window = 120;             %Size of moving window for defining pairs
Risk_Free_Rate = 0.03;  %Risk free rate
Capital = 1000000;      %Ammount of capital traded in each position taken
(same unit as C)
Control_Days = 40;      %Control day is  the longest hold period for each
pair
[Days Pairs] = size(Stock_Price);
Pair_Number = Pairs/2;
Stop_ Loss = -0.10;     %Stop loss control
i=1;

for i = 1:Pair_Number
    figure
    plot(Stock_Price(:,[(i*2 -1) i*2]),'LineWidth',2)
    xlabel('Time(days)')
    ylabel('Stock Price')
    title(strcat('Stock Price Movement for Pair #',num2str(i)))

end
```

```
[Account  Trade_Time Cumulative UL LL] =PairsTrading(Stock_Price, Capital,
...
window, Risk_Free_Rate,Control _Days,Pair_Number,Stop_Loss); % This funciton
performs the traidng strategy


figure
plot(Cumulative,'LineWidth',2)
legend('Profits for each pair')
xlabel('Testing(In sample): From Jan.1 2001 -- Dec.31 2005')
ylabel('Total Profit/Lost by Pair')


figure
plot(Account,'LineWidth',2)
legend('Account Balance')
xlabel('Testing(In sample): From Jan.1 2001 -- Dec.31 2005')
ylabel('Total Account: Initial Investment and Realized Profit/Lost')
```

PairsTrading.m

```
function [Account Trade_Time Cumulative_Profit_Each_Pair UL LL] =
PairsTrading(Stock_Price, ...
            Capital, window, Risk_Free_Rate, Control_Days, Pair_Number,
Stop_Loss )


Risk_Free = Risk_Free_Rate/252;
Stock_Price_Matrix = Stock_Price;
Backtesting_Days = length(Stock_Price_Matrix) ; %Total Number of Test Days
Expected_ Return = 2*Risk_Free_Rate;             %Used to control trading
position


[Spreads_Matrix]= Spreads_Calculation(Stock_Price_Matrix, Pair_Number); %
Calculates the spreads matrix


% Defining all the matrix we need to use to perfrom our trading
Pairs_Status_Matrix = zeros(1,Pair_Number + 1);
Pairs_Monitor_Matrix = zeros(1,Pair_Number + 1);
Pairs_Last_Matrix = ones(1,Pair_Number + 1);
Position_Shares_Matrix = zeros(2,Pair_Number);
Stock_Update_Matrix = zeros(2,Pair_Number);
Position_Shares_Update_Matrix = zeros(2,Pair_Number);
Account_Balance_Matrix =[zeros(1,Pair_Number) Capital];


Position_Matrix= zeros(2,Pair_Number);
Original_Spread_Matrix = zeros(1,Pair_Number);
Original_Stock_Matrix =zeros(2,Pair_Number);


Cumulative_Profit =0;
Cumulative = zeros(Backtesting_Days-window+1,Pair_Number); %Store Everyday
Return
Cumulative_Profit_Each_Pair = zeros(Backtesting_Days-window+1,Pair_Number);
```

```
Trade_Time = zeros(1,Pair_Number);
Close_Time = zeros(1,Pair_Number);
Original_Sigma = zeros(1,Pair_Number); % Store the parameters for opening
new pairs
Original_Mu = zeros(1,Pair_Number); % Store the parameters for opening
new pairs

for pairs = 1: Pair_Number
    eval(['Trade_Return_' num2str(pairs) '=[];']);
    eval(['Trade_Gain_' num2str(pairs) '=[];']);
end

for day = window: Backtesting_Days % Start trading based on a rolling window

   %fprintf(1,['\nObservation #',num2str(day),'------------------------------
-']);
   %fprintf(1,['-----Cumulative Profit
= ',num2str(Cumulative_Profit)],'\n');

    Account_Balance_Matrix(1,Pair_Number + 1) =
Account_Balance_Matrix(1,Pair_Number + 1)*exp(Risk_Free); %Invest extra
money at risk free rate
    Pairs_Open_Matrix = [zeros(1,Pair_Number) 1]; %Before updating
everyday trading, the open position are assumed to be closed
    Return_Matrix = zeros(1,Pair_Number);

    for pairs = 1:Pair_Number

        Spread_Now = Spreads_Matrix(day,pairs);
        Price_A = Stock_Price_Matrix(day,2*pairs-1);
        Price_B = Stock_Price_Matrix(day,2*pairs);
        Stock_Update_Matrix(1,pairs) = Price_A;
        Stock_Update_Matrix(2,pairs) = Price_B;

        [Mu Sigma]=OU_est(Spreads_Matrix((day-
window+1):day,pairs)); %Estimate the parameters
        %Mu = mean(Spreads_Matrix((day-window+1):day,pairs));
        %Sigma = std(Spreads_Matrix((day-window+1):day,pairs));
        Return = 0.5*(abs(Spread_Now - Mu) - 0.5*Sigma);  %Return

        UL(day,pairs)= Mu + 2*Sigma;
        LL(day,pairs)= Mu - 2*Sigma;

        Check = abs(Spread_Now - Mu) - 2*Sigma ;     % Check whether the
Return satisfies the requirement
        Check_Out = abs(Spread_Now - Mu) -0.5*Sigma;  % Check the closing
position
        Check_Risk = abs(Spread_Now - Mu) - 3*Sigma;   % Risk management

        Moniter = Pairs_Monitor_Matrix(1,pairs);
        Status = Pairs_Status_Matrix(1,pairs);
        Last_Days = Pairs_Last_Matrix(1,pairs);
```

```
        RiskFree_Account = Account_Balance_Matrix(1,Pair_Number + 1);

        Shares_Pair_A = Position_Shares_Matrix(1,pairs);
        Shares_Pair_B = Position_Shares_Matrix(2,pairs);


        if Status == 0
            if Check >= 0 && Moniter >= 1 && Return >= Expected_Return

                %Pairs_Status_Matrix(1,pairs) =1;
                Pairs_Open_Matrix(1,pairs) =1;
                Return_Matrix(1,pairs) = Return;
                %fprintf(1,['\n-----------Find
Pairs','Pair#',num2str(pairs)]);

= ',num2str(Return)]);
                %fprintf(1,['\n','Price_Now_A = ',num2str(Price_A),'.
Position: ',num2str(Position_Matrix(1,pairs)),'\n', ...
                 % 'Price_Now_B = ',num2str(Price_B),'. Position:
',num2str(Position_Matrix(2,pairs)),'\n']);


                if Spread_Now -
mean(Spreads_Matrix((day-window+1):day,pairs)) >= 0
                    Position_Matrix(1,pairs)=-1;
                    Position_Matrix(2,pairs)=1;
                else
                    Position_Matrix(1,pairs)=1;
                    Position_Matrix(2,pairs)=-1;
                end

                Original_Stock_Matrix(1,pairs) = Price_A;
                Original_Stock_Matrix(2,pairs) = Price_B;

                Original_Sigma(1,pairs) = Sigma;
                Original_Mu(1,pairs) = Mu;
                Original_Spread_Matrix(1,pairs) = Spread_Now;
                Pairs_Monitor_Matrix(1,pairs)= 0;

            else
                if Check >=0

Pairs_Monitor_Matrix(1,pairs)=1+Pairs_Monitor_Matrix(1,pairs);
                end
            end
        else
            if Status == 1

                Profit = Shares_Pair_A*(Price_A -
Original_Stock_Matrix(1,pairs))*Position_Matrix(1,pairs) + ...
                                  Shares_Pair_B*(Price_B -
Original_Stock_Matrix(2,pairs))*Position_Matrix(2,pairs);
```

```
                Closing_Position = Profit +
                Account_Balance_Matrix(1,pairs); Trade_Return =
0.995*Closing_Position/Account_Balance_Matrix(1,pairs) - 1;

                if Check_Out <= 0 || Trade_Return <= Stop_Loss || Last_Days
>= Control_Days %|| Trade_Return >= abs(Stop_Loss)

                   Pairs_Monitor_Matrix(1,pairs) = 0;
                   eval(['Trade_Return_' num2str(pairs)
'=[eval([''Trade_Return_'' num2str(pairs)]) Trade_Return];']);
                   eval(['Trade_Gain_' num2str(pairs)
'=[eval([''Trade_Gain_'' num2str(pairs)]) Profit];']);
                   Pairs_Status_Matrix(1,pairs) =0;
                   Close_Time(1,pairs) = Close_Time(1,pairs) + 1;
                   Cumulative_Profit = Cumulative_Profit + Profit;
                   Cumulative(day-window+1,pairs) = Profit;


                   Account_Balance_Matrix(1,Pair_Number + 1) =
RiskFree_Account + 0.995*Closing_Position;
                   Account_Balance_Matrix(1,pairs) = 0;
                   Position_Shares_Matrix(1,pairs) = 0;
                   Position_Shares_Matrix(2,pairs) = 0;
                   Pairs_Last_Matrix(1,pairs) = 0;


                   fprintf(1,['\n-----------Close Position','Pair#
',num2str(pairs),' . ', 'Profit = ',num2str(Profit),' .']);
                   fprintf(1,['\n','Check_Out = ',num2str(Check _Out),'.
Last_Days = ',num2str(Last_Days),' Check = ',num2str(Check)]);

                   Original_A = Original_Stock_Matrix(1,pairs);
                   Original_B = Original_Stock_Matrix(2,pairs);
                   Original_Spread = Original_Spread_Matrix(1,pairs);
                   %fprintf(1,['\n','Original_Price_A =
',num2str(Original_A),'; Price_Now = ',num2str(Price_A),' ; Shares =
',num2str(Shares_Pair_A),'\n', ...
                     %              'Original_Price_B =
',num2str(Original_B),'; Price_Now = ',num2str(Price_B),' ; Shares =
',num2str(Shares_Pair_B),'\n', ...
                      %             'Original_Spread  =
',num2str(Original_Spread),'; Spread _Now = ',num2str(Spread_Now),'\n']);

                else
                    Pairs_Last_Matrix(1,pairs) = Last_Days + 1;
                end
            end
        end
    end

    %This part is for updating investment account.

    Investment = Account_Balance_Matrix(1,Pair_Number + 1); % Take the money
from risk free account
```

```
    Account_Balance_Matrix(1,Pair_Number + 1) = 0 ; % Clear the risk free
account

    Account_Update_Matrix =
Investment*Pairs_Open_Matrix*1/(sum(Pairs_Open_Matrix)); % Reinvest equally
into each pair account including risk free account
Account_Balance_ Matrix = Account_Balance_Matrix + Account_Update_Matrix;
%Update the account

    for pairs = 1:Pair_Number
        if Account_Update_Matrix(1,pairs) > 0 && Pairs_Open_Matrix(1,pairs) >
0
            Pairs_Status_Matrix(1,pairs) = 1;
            Trade_Time(1,pairs) = Trade_Time(1,pairs) +1;
        end
    end



    % Update the everyday stock share position through closing or opening
    % pairs
    for pairs = 1:Pair_Number
        Position_Shares_Update_Matrix(1,pairs) =
1/2*0.995*Account_Update_Matrix(1,pairs)/Stock_Update_Matrix(1,pairs);
        Position_Shares_Update_Matrix(2,pairs) =
1/2*0.995*Account_Update_Matrix(1,pairs)/Stock_Update_Matrix(2,pairs);
        Cumulative_Profit_Each_Pair(day-window+1,pairs)
=sum(Cumulative(:,pairs));
    end

    Position_Shares_Matrix = Position_Shares_Matrix +
Position_Shares_Update_Matrix;

    Account(1,day-window +1) = sum(Account_Balance_Matrix);

end

% Based on the trading result, calculate the sharpe ratio through risk free
% rate, anuual trading return and trading return volatility

Sharpe_Ratio = zeros(1, Pair_Number);

for pairs = 1: Pair_Number

    Sharpe_Ratio(1,pairs) = (1/5*sum(eval(['Trade_Return_' num2str(pairs)])) -
Risk_Free_Rate)/(sqrt(1/5)*std(eval(['Trade_Return_' num2str(pairs)])));

end


[r c] = size(Cumulative);
```

```
Returns_array_temp
=zeros(r,1); iter = 1;

for rows = 1:r
    sum_row = sum(Cumulative(rows,:));
    if sum_row ~= 0

        Returns_array_temp(iter,1) =
        sum_row; iter = iter +1;
    end
 end

Return_Vector = Returns_array_temp(1:iter-1);

Sharpe_Ratio_Portfolio = (((((sum(Return_Vector)+ Capital)/(Capital)^(1/5))-1)-
Risk_Free_Rate))/(sqrt(1/5)*std(Return_Vector));

display(Sharpe_Ratio_Portfolio);

VaR_Portfolio = std(Return_Vector)*1.65 - mean(Return_Vector);

display(VaR_Portfolio);

CVaR_Portfolio = std(Return_Vector)*2.063 -
mean(Return_Vector); display(CVaR_Portfolio);


%Summarizing the overall trading information for each pair
fprintf(1,['\n~~~~~The end of the Pairs Trading.','The current Balance
= ',num2str(Account(1,end)),'~~~~~~~~~~~\n'])
fprintf(1,['Pair # ; Trade Times ; Cumulative Profit ($); Average Realized
Profit/Loss ($); Maximum Gain; Maximum Loss; Sharpe Ratio'])

for pairs = 1: Pair_Number
    fprintf(1,['\n ',num2str(pairs),'; ',num2str(Trade_Time(1,pairs)),';
',num2str(sum(Cumulative(:,pairs))), ...
            '; ',num2str(sum(Cumulative(:,pairs))/Trade_Time(1,pairs)),';
',num2str(max(Cumulative(:,pairs))), ...
                '; ',num2str(min(Cumulative(:,pairs))),';
', num2str(Sharpe_Ratio(1,pairs)),'\n']);
end

plot(Cumulative);
xlabel('Trade Time')
ylabel('Realized Gain/Loss for Each Trade ')

for pairs = 1: Pair_Number
    figure
    plot(Cumulative(:,pairs));
    title(strcat('Realized Returns for Pair
    #',num2str(pairs))) xlabel('Trade Time')
```

```
    ylabel('Realized Gain/Loss for Each Trade ')
end

x = ((-(max(Return_Vector)-min(Return_Vector))/50) +
min(Return_Vector)):(max(Return_Vector)-
min(Return_Vector))/50:(((max(Return_Vector)-min(Return_Vector))/50)
+ max(Return_Vector));
hist(Return_Vector,x);
xlabel('Profits')
ylabel('Frequency')

title(strcat('Frequency Distribution of Returns'))
```

SimulateOrnsteinUhlenbeck.m

```
function [S] = SimulateOrnsteinUhlenbeck(S0, mu, sigma, lambda, deltat, t)


periods = floor(t / deltat); S
= zeros(periods, 1);
S(1) = S0;

exp_minus_lambda_deltat = exp(-lambda*deltat);

if (lambda == 0) % Handle the case of lambda = 0 i.e. no mean reversion.
    dWt = sqrt(deltat) * randn(periods,1);
else
    dWt = sqrt((1-exp(-2*lambda* deltat))/(2*lambda)) * randn(periods,1);
end

for t=2:1:periods
    S(t) = S(t -1)*exp_minus_lambda_deltat + mu*(1-exp_minus_lambda_deltat)
+ sigma*dWt(t);
end
```

Spreads_Calculation.m

```
function [Spreads_Matrix]= Spreads_Calculation(Stock_Price_Matrix,
Pair_Number)
%SPREADS_CALCULATION Summary of this function goes here
%Detailed explanation goes here

for pairs = 1 : Pair_Number
    Spreads(:,pairs)= log(Stock_Price_Matrix(:,2*pairs-1)) -
log(Stock_Price_Matrix(:,2*pairs));
    %Calculates the spreads between the log of the two stock prices in a
    %pair
end
```

```
Spreads_Matrix = Spreads;
end
```

OU_est.m

```
function [mu,sigma] = OU_est(S)

  n = length(S)-1;
  delta=1;

  Sx  = sum( S(1:end-1) );
  Sy  = sum( S(2:end) );
  Sxx = sum( S(1:end-1).^2 );

  Sxy = sum( S(1:end-1).*S(2:end) );
  Syy = sum( S(2:end).^2 );

  mu  = (Sy*Sxx - Sx*Sxy) / ( n*(Sxx - Sxy) - (Sx^2 - Sx*Sy) );
  lambda = -log( (Sxy - mu*Sx - mu*Sy + n*mu^2) / (Sxx -2*mu*Sx + n*mu^2) )
/ delta;
  a = exp(-lambda*delta);
  sigmah2 = (Syy - 2*a*Sxy + a^2*Sxx - 2*mu*(1-a)*(Sy - a*Sx) +
n*mu^2*(1-a)^2)/n;
  sigma = sqrt(sigmah2*2*lambda/(1-a^2));

  %forecast=S(n+1)*exp(-lambda)+mu*(1-exp(-lambda));
  %stdev=sigma*sqrt((1-exp(-2*lambda))/(2*lambda));
end
```

Output.m

```
[Spreads_Matrix ]= Spreads_Calculation(Stock_Price, Pair_Number);


figure
plot(Spreads_Matrix)
legend('Pair#1','Pair#2','Pair#3','Pair#4','Pair#5')
xlabel('Backtesting: From Jan.1 2006 -- Nov.29
2011') ylabel('Daily fluctuation of Pair Spread')


figure
plot(Account,'LineWidth',2)
legend('Account Balance')
xlabel('Backtesting: From Jan.1 2006 -- Nov.29 2011')
ylabel('Total Account: Initial Investment and Realized Profit or Lost ')


figure
plot(Cumulative)
legend('Pair#1','Pair#2','Pair#3','Pair#4','Pair#5')
xlabel('Backtesting: From Jan.1 2006 -- Nov.29
2011') ylabel('Cumulative Realized Gain/Loss')
```

```
figure
OU_S0= 0;
OU_mu = 0;
OU_sigma = 0.3;
OU_lambda =
0.1; OU_deltat
= 1; OU_t=500;
OU_Process =OU_Simulation( OU_S0, OU_mu, OU_sigma, OU_lambda, OU_deltat,
OU_t );
OU_UL = [ones(500,2)*0.3*diag([ -0.5
0.5])]; OU_LL = [ones(500,2)*0.3*diag([2 -2
])]; hold on
plot(OU_Process,'b')
plot(OU_UL, 'r')
plot(OU_LL, 'g') hold
off

legend('Ornstein-Uhlenbeck','2*sigma','-2*sigma','0.5*sigma','-
0.5*sigma')    xlabel('Ornstein-Uhlenbeck    Simulation:    S_0=0,    mu=0,
sigma=0.3, lambda=0.1, deltat=1, t=500')
ylabel('Ornstein-Uhlenbeck MOdel')
```

SR_Simulation.m

```
% Simulation of the windows and control days

max_SR = 0;
max_window = 0;
max_control =
0; multiplier =
1; increments =
29; gap = 150;
k = 1; %less than or equal to
1; max_inc = increments + gap;
max_w = ((max_inc -
increments)/multiplier)+1; max_c = k*max_w;
wind = zeros(max_w,1);
cont = zeros(max_c,1);
Profit_Simulation=zeros(max_w,max_c);

for window = 1:max_w
    for control_days = 1:window
    wind(window, 1) = (multiplier*window)+increments;
    cont(control_days,1) = (5*control_days)+increments;

fprintf(1,['\nSimulation_Window_',num2str(window*multiplier+increments),'_Con
trol_', num2str(control_days*multiplier+increments)]);
    Sharpe_it = SR_sim(window, control_days,
    multiplier,increments); display(Sharpe_it);
    SR_Simulation(window,control_days) = Sharpe_it;

    if SR_Simulation(window,control_days) > max_SR
```

```
        max_SR = SR_Simulation(window,control_days);
        max_window = multiplier*window + increments;
        max_control = multiplier*control_days +increments;
    end

    end
end
fprintf(1,['\n Matrix with the Simulated
Results']); display(SR_Simulation);
fprintf(1,['\nWindow with Maximum Sharpe
Ratio:',num2str(max_window),'\nControl Days for Maximum Sharpe
Ratio:', num2str(max_control)]);
fprintf(1,['\nMaximum Sharpe Ratio:',num2str(max_SR)]);
surf(cont,wind,SR_Simulation);
xlabel('Number of Control Days');
ylabel('Rolling Window');
zlabel('Sharpe Ratio');
title('Sharpe Ratio - Variation with Control Days and Window');
```

SR_Sim.m

```
function SR_sim = SR_ sim(w, cd, m, i)
SR_sim = zeros(m,i);
load Pairs.mat;

window = (m*w)+i;
Control_Days = (m*cd)+i;
Stock_Price=Pairs_Price_Matrix;
Stock_Price = flipud(Stock_Price);
Risk_Free_Rate = 0.01; %Risk free rate
Capital = 1000000;      %Ammount of capital traded in each position taken
(same unit as C)
[Days Pairs] = size(Stock Price);
Pair Number = Pairs/2;
Stop_Loss = -0.10;      %Stop loss control


[Account Sharpe_Ratio_Portfolio Net_Profit VaR_Portfolio CVaR_Portfolio]
=PairsTrading_Sim(Stock_Price, Capital, ...
window, Risk_Free_Rate,Control _Days,Pair_Number,Stop_Loss); % This funciton
performs the traidng strategy

SR_sim = Sharpe_Ratio_Portfolio;
end
```

Profit_Simulation.m

```
% Simulation of the windows and control days

max_profit = 0;
max_window = 0;
max_control = 0;
multiplier = 1;
increments = 29;
gap = 150;
k = 1; %less than or equal to 1;
max_inc = increments + gap;
max_w = ((max_inc - increments)/multiplier)+1;
max_c = k*max_w;
wind = zeros(max_w,1);
cont = zeros(max_c,1);
Profit_Simulation=zeros(max_w,max_c);

for window = 1:max_w
    for control_days = 1:window
    wind(window, 1) = (multiplier*window)+increments;
    cont(control_days,1) = (5*control_days)+increments;
```

```
fprintf(1,['\nSimulation_Window_',num2str(window*multiplier+increments),'_Con
trol_', num2str(control_days*multiplier+increments)])
    prof_it = PT_sim(window, control_days,
    multiplier,increments); display(prof_it);
    Profit_Simulation(window,control_days) = prof_it;

    if Profit_Simulation(window,control_days) > max_profit
        max_profit = Profit_Simulation(window,control _days);
        max_window = multiplier*window + increments;
        max_control = multiplier*control_days +increments;
    end

    end
end
fprintf(1,['\n Matrix with the simulated
Results']); display(Profit_Simulation);
fprintf(1,['\nWindow with Maximum Profit:',num2str(max_window),'\nControl
Days for Maximum Profit:', num2str(max_control)]);
fprintf(1,['\nMaximum Profit:',num2str(max_profit)]);
surf(cont,wind,Profit_Simulation);
xlabel('Number of Control Days');
ylabel('Rolling Window');
zlabel('Realized Profit');
title('Risk Adjusted Return - Variation with Control Days and Window');
```

PT_Sim.m

```
function Prof_sim = PT_ sim(w, cd, m,
i) Prof_sim = zeros(m,i);
load Pairs.mat;

window = (m*w)+i;
Control_Days = (m*cd)+i;
Stock_Price=Pairs_Price_Matrix;
Stock_Price = flipud(Stock_Price);
Risk_Free_Rate = 0.01; %Risk free rate
Capital = 1000000;      %Ammount of capital traded in each position taken
(same unit as C)
[Days Pairs] = size(Stock_Price);
Pair_Number = Pairs/2;
Stop_Loss = -0.10;      %Stop loss control


[Account Sharpe_Ratio_Portfolio Net_Profit VaR_Portfolio CVaR_Portfolio]
=PairsTrading_Sim(Stock_Price, Capital, ...
window, Risk_Free_Rate,Control _Days,Pair_Number,Stop_Loss); % This
funciton performs the traidng strategy

Prof _sim =
Net_Profit; end
```

Selection Strategies.R

```
distance <- function(x)
{
distance_matrix<-matrix(nrow=(NCOL(x)-1),ncol=(NCOL(x)-1)) for(i in
1:(ncol(distance_matrix)-1))
{
for(j in (i+1):(ncol(distance_matrix)))
{
price1=x[,i+1]
price2=x[,j+1]
price1=price1/price1[1]
price2=price2/price2[1]
sum=0
for (k in 1:NROW(x))
{
sum=sum+(price1[k]-price2[k])^2
}
distance_matrix[i,j]=sum
}
}
return(distance_matrix)
}

-------------------------------------------------------------------------------
--------------------
beta <- function(x)
{
beta_matrix<-matrix(nrow=(NCOL(x)-2),ncol=(NCOL(x)-2))
beta_array<-matrix(nrow=1,ncol=(NCOL(x)-2))
for(i in 1:(ncol(beta_array)))
{
y<-lm(x[,ncol(x)]~x[,i+1])
y=coef(y)
y=as.matrix(y)
beta_array[1,i]=y[2,1];
}
```

Appendix B: Stocks used in strategy

Energy Sector

ACI	Arch Coal Inc.
AEM	Agnico Eagle Mines Ltd
BTU	Peabody Energy Corporation
CAM	Cameron International Corp
CNQ	Canadian Natural Resource Ltd
CNX	CONSOL Energy Inc
FST	Forest Oil Corporation
COG	Cabot Oil & Gas Corporation
GG	Goldcorp Inc.
HAL	Halliburton Company
NOV	National-Oilwell Varco, Inc.
OXY	Occidental Petroleum Corporation
PEO	Petroleum & Resources Corporation
SLB	Schlumberger Limited
TLM	Talisman Energy Inc.

Banking sector

BAC	Bank of America Corp.
BNS	Bank of Nova Scotia
C	Citigroup Inc.
CM	Canadian Imperial Bank of Commerce
CS	Credit Suisse Group AG (ADR)
DB	Deutsche Bank AG
FITB	Fifth Third Bancorp
HBC	Home Bancorp, Inc
HDB	HDFC Bank Limited (ADR)
IBN	ICICI Bank Ltd (ADR)
MS	Morgan Stanley
PNC	PNC Financial Services Group Inc
RBS	Royal bank of Scotland
RY	Royal Bank of Canada
WFC	Wells Fargo & Co

Healthcare sector

ABT	Abbott Laboratories
JNJ	Johnson & Johnson
LLY	Eli Lilly and Co
PFE	Pfizer Inc
AMGN	Amgen, Inc
AZX	Alexandria Minerals Corporation
BAX	Baxter International Inc
BMY	Bristol-Myers Squibb Co
GSK	GlaxoSmithKline plc (ADR)
NVO	Novo Nordisk A/S (ADR)
PFE	Pfizer Inc
RHHBY	Roche Holding Ltd. (ADR)
SNY	Sanofi SA (ADR)
WCRX	Warner Chilcott Plc
MTEX	Manntech Inc.