Pairs Trading Based on Costationarity

by

Alvin Au

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Alvin Au

Abstract

Arbitrage is a widely sought after phenomenon in financial markets: profit without any risk is very desirable. Statistical arbitrage is a related concept: the idea is to take advantage of market inefficiencies using statistical techniques and mathematical models. It is by no means risk-free however. We focus on the statistical arbitrage technique "pairs trading" utilizing both cointegration and minimum distance pairs. We discuss the algorithms involved and simulate these based on data from the NASDAQ 100.

There have been recent forages into financial applications and time series with wavelets. However, ideas surrounding pairs trading through the use of wavelets have been little to non-existent. Our contribution is the application of wavelets and costationarity as an approach to pairs trading. We applied the concept of estimating the evolutionary wavelet spectrum, which is analogous to the spectrum for time series but for wavelets. Following the estimation of the evolutionary wavelet spectrum, we find variance stationary linear combinations of the differenced stock prices. This is essentially the concept of costationarity: finding variance stationary linear combinations from non-stationary processes using time-varying coefficients. We then compare the results of the application of the costationarity method to the minimum distance method and to the cointegration method. We find that there are significant improvements on the minimum distance method, but that it does not have a large improvement over the cointegration method.

Acknowledgements

I would like to thank my supervisor, Professor Shoja'eddin Chenouri for being there to support and lead me through the various stages of this entire process. Without him, this would not be possible. I would also like to thank Professor Bin Li and Professor Tony Wirjanto for being on my thesis committtee.

Thank you to all my classmates and friends for being with me through both the good times and the bad (and I'm sorry if you had to listen to me rant in the bad), and a huge thanks to my parents for supporting me throughout all my endeavours.

Table of Contents

List of Tables vii

List of Figures xi

1 Introduction 1

1.1 Time Series 1

1.1.1 Wiener Processes 4

1.2 Pairs Trading 5

1.2.1 Minimum-Distance Method 7

1.2.2 Stochastic Spread Method 8

1.2.3 Cointegration Method 13

1.3 Application of Pairs Trading on Data 22

1.3.1 Minimum Distance Method 23

1.3.2 Cointegration Method 28

1.3.3 Application of the Cointegration Method to Stock Data 30

1.3.4 Upper and Lower Bound of Two Standard Deviations from the Mean 33

1.3.5 Upper and Lower Bound of One Standard Deviation from the Mean 45

2 Wavelet Analysis of Time Series **59**

2.1 Introduction 59

2.2 Fourier Series and Fourier Transforms 60

2.3 Wavelets 64

2.4 Non-decimated Wavelet Transform 71

2.5 Locally Stationary Processes 72

2.5.1 Estimation of the EWS 76

2.6 Costationarity 77

2.7 Pairs Trading based on Costationarity on Stock Data 80

2.8 Comparison of the Costationarity Method with the Minimum Distance Method 83

2.9 Comparison of the Costationarity Method with the Cointegration Method . 103

3 Conclusion **117**

3.1 Future Work 118

Appendices **120**

A Table of Stocks Used **121**

B R Code **123**

References **147**

List of Tables

1.1 The training and testing periods for the 91 elligible stocks in the NASDAQ 100 34

1.2 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pairs 1 to 5) 41

1.3 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pairs 6 to 10) 42

1.4 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pair 11) 42

1.5 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from November 21 2012 - May 28 2013 (pairs 3,8,13) 43

1.6 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 1 to 5) 43

1.7 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 6 to 10) 44

1.8 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 11 to 13) 44

1.9 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 1 to 5) 53

1.10 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 6 to 10) 54

1.11 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 11 to 13) 54

1.12 Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from November 21 2012 - May 28 2013 (pairs 3,8,13) 55

1.13 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 1 to 5) 55

1.14 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 6 to 10) 56

1.15 Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 11 to 13) 56

1.16 A comparison of the trading results on the (out of sample) data of 6 months using training data of 3 years from May 21 2012 - November 21 2012 (pairs 3,8,13) for trading bounds of 1 and 2 standard deviations from the mean. Only the pairs that remain cointegrated after the trading period have been selected for the comparison. 58

2.1 The training and testing periods for the MDM and CM elligible stocks in the NASDAQ 100 86

2.2 The averaged returns across the useable solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%) 87

2.3 The averaged returns across 10 solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%) 88

2.4 The averaged returns across the useable solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%) 89

2.5 The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column. 90

2.6 The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column. 91

2.7 The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column. 92

2.8 The difference between the averaged returns of each method (CM and MDM) for each test and each pair. The rows indicate which test number the return is representing. Each value is a percent return (%), with positive values representing the CM performing better than the MDM, and negative values representing the MDM performing better than the CM. The pairs in order from 1 to 10 are SYMC & YHOO, CMCSA & MXIM, INTC & MDLZ, AMAT & FOXA, DISCA & MAT, FOXA & SBUX, FOXA & QVCA, FISV & LLTC, MAT & VOD, and MAT & MYL. 93

2.9 The training and testing periods for the CIM and CM elligible stocks in the NASDAQ 100 106

2.10 The averaged returns across the solutions used for each test and for each method (CM and CIM). The rows indicate which test number the return is representing. Each value is a percent return (%). 106

2.11 The total number of trades executed on each of the 5 tests for CM and CIM. The stock pairs that are relevant are labelled at the top of each column. . . 107

2.12 The difference between the averaged returns of each method (CM and CIM) for each test and each pair. Each value is a percent return (%), with positive values representing the CM performing better than the CIM, and negative values representing the CIM performing better than the CM. 107

A.1 The 91 stocks used from the NASDAQ 100 that had data points from May 20th, 2009 to May 12th, 2015 122

List of Figures

1.1 The price paths for ten stocks of 1000 days each. The first four pairs are simulated from a simplified ECM. 24

1.2 The spreads for the first three pairs in the minimum distance simulation and the days that the trade positions are open and closed. The black portion of the spread represents the training set and the green portion of the spread represents the test set. The upper bounds and lower bounds of the spreads (the mean +/- 2 standard deviations) are represented by the blue horizontal lines, and the red line represents the historical mean of the training set. . . 26

1.3 The spreads for the fourth and fifth pairs in the minimum distance simulation and the days that the trade positions are open and closed. The black portion of the spread represents the training set and the green portion of the spread represents the test set. The upper bounds and lower bounds of the spreads (the mean +/- 2 standard deviations) are represented by the blue horizontal lines, and the red line represents the historical mean of the training set. 27

1.4 The simulated price paths for ten stocks of 1000 days each. The first five pairs are simulated from the simplified ECM as in the minimum distance method example. The next five pairs have varying intercepts and coefficient terms. 29

1.5 The spreads for the six cointegrated pairs in the cointegration simulation and the days that the trade positions are open and closed. The first 600 days comprise the training set and is indicated in black. The test spread is for the next 400 days and are labeled in green. The red line represents the historical mean of the training set. The blue lines represent the upper and lower bounds of the trades, given by the mean +/- 2 standard deviations of the training set. 31

1.6 The training spread (in black) and the 6 month test spread (in green) for the first six cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012. 35

1.7 The training spread (in black) and the 6 month test spread (in green) for the last four pairs cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012. 36

1.8 The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period November 21 2012 - May 28 2013. 37

1.9 The training spread (in black) and the 12 month test spread (in green) for the first 6 cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. 38

1.10 The training spread (in black) and the 12 month test spread (in green) for the 7th to 12th cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. 39

1.11 The training spread (in black) and the 12 month test spread (in green) for the 13th cointegrated pair using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. . . . 40

1.12 The training spread (in black) and the 6 month test spread (in green) cointegrated pairs (1 to 6) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012. 46

1.13 The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs (6 to 12) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012. 47

1.14 The training spread (in black) and the 6 month test spread (in green) for the cointegrated pair (13) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012. 48

1.15 The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs 3,8,13 using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period November 21 2012 - May 28 2013. 49

1.16 The training spread (in black) and the 12 month test spread (in green) for the first 6 cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. 50

1.17 The training spread (in black) and the 12 month test spread (in green) for the 7th to 12th cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. 51

1.18 The training spread (in black) and the 12 month test spread (in green) for the 13th cointegrated pair using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013. . . . 52

1.19 The training spread (in black) and the 6 month test spread (in green) cointegrated pairs (3,8, 13) using data from the stocks of the NASDAQ 100. The top row of spreads shows the trades with bounds of 2 standard deviations from the mean, while the second row shows the trades with bounds 1 standard deviation from the mean. These pairs are traded on the period May 21 2012 - November 21 2012, but are used mainly as a comparison for using different standard deviations on the bounds. The pairs have been selected retrospectively after the trades have happened and have been determined to remain cointegrated. 57

2.1 A father wavelet on the left plot. The right plot shows that the relationship described in Equation 2.31: the Haar father wavelet can be written as a sum of dilated and translated father wavelets. 67

2.2 The Doppler function in the top left plot (1). The other plots (2),(3), and (4) are projections of the Doppler function into father wavelet spaces $J = 2, 4$ and 6. Notice that each plot has the doppler function being projected onto 2^J different coefficients (4, 16, 64). 68

2.3 A Haar mother wavelet (left) and a mother wavelet child $\psi_{2,2}$ (right) 69

2.4 A spectrum $S_j(z)$ from Equation 2.52 on the left. The resulting function that is simulated from the spectrum is plotted on the right. 75

2.5 The plots of the prices of the stock pairs. The black and blue lines represent the stock prices of the first and second of the stocks in the title of each plot respectively. The green line is where the stocks start to diverge in some cases, and this corresponds with the 5th test set. It is for this reason why we consider comparing the returns only from tests 1-4 with the tests from 1-10, and there is a noticeable difference albeit mainly from one outlier. . . 94

2.6 The plots of the 1st test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 95

2.7 The plots of the 2nd test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 96

2.8 The plots of the 3rd test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 97

2.9 The plots of the 4th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 98

2.10 The plots of the 5th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 98

2.11 The plots of the 6th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 99

2.12 The plots of the 7th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 99

2.13 The plots of the 8th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 100

2.14 The plots of the 9th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 101

2.15 The plots of the 10th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO. 102

2.16 The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair CMCSA,GILD. 108

2.17 The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair CMCSA,GILD. 109

2.18 The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair CSCO,WYNN. 109

2.19 The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair CSCO,WYNN. 110

2.20 The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA. 111

2.21 The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA. 112

2.22 The plots of the 3rd test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA. 112

2.23 The plots of the 4th test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA. 113

2.24 The plots of the 5th test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA. 113

2.25 The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL. 114

2.26 The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL. 114

2.27 The plots of the 3rd test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL. 115

2.28 The plots of the 4th test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL. 115

2.29 The plots of the 5th test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL. 116

2.30 The plots of the trajectories of the stock pairs (P_t^B) and their cointegration relationship counterpart ($\alpha + \beta P_A^t$) over the 5 test periods. The pairs CMCSA,GILD and CSCO,WYNN are false positives for cointegration, while the pairs HSIC,LBTYA and QVCA,SIAL have much longer lasting cointegrating relationships. The red lines represent the training set and the green lines represent the test set. The blue lines represent P_t^B (the second stock in the titles), while the black lines represent $\alpha + \beta P_A^t$, where P_A^t is the first stock in the titles. 116

Chapter 1

Introduction

In this chapter, we will introduce basic concepts regarding time series and the idea of pairs trading, including the three main approaches used in this particular type of statistical arbitrage.

1.1 Time Series

In the analysis of time series, we wish to discover temporal relationships in our data. For this reason, we study stochastic processes.

Definition 1. A stochastic process X_t is described as *weakly stationary* if its mean and variance are constant, and if its autocovariance only varies with the length of the time interval. That is, a stochastic process X_t is weakly stationary if for all t and any s,

$$\begin{aligned} \mathrm{E}\left[X_t\right] &= \mathrm{E}\left[X_{t-s}\right] = \mu \\ \mathrm{E}\left[(X_t-\mu)^2\right] &= \mathrm{E}\left[(X_{t-s}-\mu)^2\right] = \sigma^2 \\ \mathrm{E}\left[(X_t-\mu)(X_{t-s}-\mu)\right] &= \gamma_s\,, \end{aligned} \tag{1.1}$$

where μ, σ^2, and γ_s are constants.

Definition 2. A *stochastic process* $\{X_t\}_{t=-\infty}^{\infty}$ is a sequence of random variables that is indexed by time. In contrast to sampling data from a population where the random variables are independent, the ordering of the random variables is very important here because we wish to capture the dependence between observations.

Definition 3. Let $X_t \sim i.i.d.(0, \sigma^2)$. Then $\{X_t\}_{t=-\infty}^{\infty}$ is known as a *white noise process* with $\mathrm{E}[X_t] = 0$, $\mathrm{Var}[X_t] = \sigma^2$, and $\mathrm{Cov}(X_t, X_{t-s}) = 0$ for all $t \neq s$ and is denoted by $\mathrm{WN}(0, \sigma^2)$.

Definition 4. A stochastic process X_t is an autoregressive process of order p, or an *AR(p) process* if it can be written in the form

$$X_t - \mu = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \epsilon_t,$$

where μ is a constant and ϵ_t is a i.i.d. $\mathrm{WN}(0, \sigma^2)$ process.

The lag operator L can be defined as the following:

$$L^k X_t = X_{t-k}.$$

This AR(p) process X_t can be written as:

$$\Phi(L) X_t = \mu + \epsilon_t,$$

where $\Phi(L) = 1 - \phi_1 L^1 - \phi_2 L^2 - ... - \phi_p L^p$.

For this AR(p) process to be stationary, the roots of the equation

$$1 - \phi_1 L^1 - \phi_2 L^2 - ... - \phi_p L^p = 0$$

must not lie on the unit circle.

A stochastic process X_t is a moving average process of order q, or an *MA(q) process* if it can be written in the form

$$X_t - \mu = \epsilon_t \ + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q},$$

where μ is a constant and ϵ_t is a i.i.d. $\mathrm{WN}(0, \sigma^2)$ process.

An AR(p) process uses past data to model the current data. This results in correlation between the past and present at each point in time, and as a result, the autocorrelation function decays to zero gradually. However, the MA(q) process is advantageous when correlation is only required for very few lags. When both AR(p) and MA(q) processes are used together to model a time series, the result is an ARMA(p, q) process.

Definition 5. A stochastic process X_t is an autoregressive moving average process with paramaters p, q, or an *ARMA*(p, q) *process* if it can be written in the form

$$X_t - \mu = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q},$$

where μ is a constant and ϵ_t is a i.i.d. WN$(0, \sigma^2)$ process.

Often, time series are not stationary. As the AR(p), MA(q), and ARMA(p, q) processes are used to model stationary series, it is useful to make data stationary before modelling. This can be done through the process of differencing the process.

Definition 6. The *first difference* of a stochastic process X_t is defined as:

$$\Delta X_t = X_t - X_{t-1},$$

and for any $d \geq 1$, the d^{th} order difference is defined as:

$$\Delta^d X_t = \Delta \left(\Delta^{d-1} X_t \right)$$

A stochastic process is *integrated of order d* if the d^{th} order difference of X_t is a weakly stationary process.

Definition 7. A stochastic process X_t is said to be an *autoregressive integrated moving average* model with parameters p, d, q, or an ARIMA(p, d, q) process, if $\Delta^d X_t$ is an ARMA(p, q) process.

1.1.1 Wiener Processes

We will briefly discuss Wiener processes, as the stocastic spread method of pairs trading uses stochastic calculus.

The weak form of the efficient market hypothesis states that future prices cannot be predicted from the past (Ross et al. (2013)). Whether or not this is true in the markets today, it is one of the assumptions that Markov processes model well.

A Markov process is a type of stochastic process where only the present value of a variable is relevant for predicting the future (Hull (2009)). A Wiener process $\{Z_t\}$ has the following properties:

Property 1. *(Increments are independent)*
For all $0 = t_0 < t_1 < ... < t_m$, the increments

$$Z_{t_1} - Z_{t_0}, Z_{t_2} - Z_{t_1}, ..., Z_{t_m} - Z_{t_{m-1}}$$

are independent.

Property 2. *(Increments are normal)*
The increments $Z_t - Z_s$ are independent normally distributed with

$$Z_t - Z_s \sim N(0, t - s)\,,$$

for any $0 < s < t$.

Property 3. Z_t is continuous in t.

The continuous case generalized Wiener process X_t which can be defined by the following stochastic differential equation:

$$dX_t = a\,dt + b\,dZ_t\,,$$

where a and b are constants, and where Z_t is a Wiener process on some defined probability space.

The constants a and b describe the mean change per unit of time and variance per unit time respectively. These are known as the *drift* of the process and the *diffusion* of the process respectively.

A process Y_t is called an Ito process if it can be represented in the following form:

$$dY_t = a_t\,dt + b_t\,dZ_t\,.$$

Note that now a_t and b_t are functions of t, and hence, can be also functions of Y_t. Y_t is also known as an Ito diffusion.

A particular type of Ito processes has the very useful property of being able to model mean reversion, which is exactly what is desired in pairs trading. These processes, known as Ornstein-Uhlenbeck processes, have the following form:

$$dY_t = \theta(\mu - Y_t)\,dt + \sigma\,dZ_t\,. \tag{1.2}$$

Mean reversion occurs in the state variable Y. This can be seen in the drift term $\theta(\mu - Y_t)$. If $Y_t < \mu$, the drift is positive. If $Y_t > \mu$, the drift is negative. In both cases, Y_t moves towards μ at a speed of θ.

1.2 Pairs Trading

The idea of arbitrage has been identified and researched heavily by hedge funds for many years. The possibility of positive returns on investments without any risk has garnered huge amounts of interest in both the industry and the academic world. Statistical arbitrage is a related concept, although it is not risk-free by any means. The idea is to take advantage of market inefficiencies using statistical techniques and mathematical models.

One of the most basic investment practices is to buy a stock long: taking an ownership of a unit of stock and hoping that the stock appreciates in value. Another form of investment

is to short sell a stock. Short selling involves the borrowing of a stock, selling it at the current time, and a promise to return the stock back to its original owner at a later time. The stocks to be returned are purchased at a later date, with a possibly different price. Of course, the profit potential here is that the short-seller expects the price of the stock to drop, resulting in a positive difference between the sold stocks at the beginning and the stocks repurchased at a later date.

Pairs trading, one of the many techniques in statistical arbitrage, involves choosing two stocks which have very similar historical price movements. If at any point the two stock price movements diverge significantly from each other, there is an opportunity for profit if the prices are expected to converge back to a long-run equilibrium eventually. At a point of divergence, the overvalued stock is sold short and the undervalued stock is bought in a long position. When they converge back to their equilibrium, the positions are closed and the profit is realized.

Jacobs and Levy (1993) state that long/short equity strategies can be split into three categories: market neutral, equitized, and hedge strategies. Market neutral strategies attempt to eliminate market exposure to systematic risks, while profiting from the excess returns from both the long position and the short position versus a benchmark index. These excess returns are referred to as *alphas*. The systematic risks can be quantified through *betas*. The other two long/short strategies attempt to earn returns on not only the two alphas, but also a return on the beta. Market neutral strategies maintain a portfolio beta of zero. There is less risk, but also less return involved. Fung and Hsieh (1999) state that market neutral funds actively seek to avoid major risk factors, but take bets on relative price movements. Pairs trading is attributed to being a market neutral strategy by Nath (2003) and Vidyamurthy (2004). Alexander and Dimitriu (2002) demonstrate it is possible to create a market-neutral strategy using cointegrated pairs of stock not with each other, but with the index.

Pairs trading is not a new concept. Since the mid-1980s, when Nunzio Tartaglia and her group of academics started researching arbitrage opportunities in the market, this technique

has been used in hedge funds ever since (Vidyamurthy (2004)). To this day, three main approaches towards pairs trading have been consistently referenced: the minimum-distance method, the stochastic spread method, and the cointegration method.

1.2.1 Minimum-Distance Method

Gatev et al. (2006) introduced a method of selecting the pair of stocks based on two steps; first constructing an index of cumulative total returns for a number of liquid stocks, and then finding a second stock by minimizing the sum of squared differences between the two normalized price series. A *normalized price series* is obtained as follows: having each price series start at 1, and each following value of the series is generated from the returns of the stock. This is the first stage of their pairs trading implementation; they call this stage the *pairs formation* stage which takes place over a period of 12 months.

The second stage of the implementation is called the *trading period*, where the pairs with the smallest distances are used to trade over a period of 6 months. The trading rule Gatev et al. (2006) propose is to open a position in the pair when the prices diverge by more than two historical standard deviations. When the prices meet again, they will close the position. If the prices do not meet, the positions are closed at the end of the trading period. One dollar worth of the higher priced stock is sold short, and one dollar worth of the lower priced stock is bought long.

Nath (2003) also administered an alternative version of the minimum-distance method. For each stock, the sum of squared differences of the normalized prices is recorded between every other stock. When the price difference is greater than the 15th percentile of all the other differences, the long/short positions are opened. When the price difference hits the median or the trading period is over, the positions are closed. Nath (2003) also considers risk management in the form of a stop-loss trigger. When the price difference hits the 5th percentile, the positions are automatically closed to prevent any further loss.

As it is mentioned by Do et al. (2006), the issue with the minimum-distance method is

that there is an assumption that the price level difference is level through time. However, this is only the case in short periods of time with pairs of securities in which the risk and returns are very similar. Do and Faff (2010) also mention that the profits of this strategy have been declining, for the reason that many pairs do not converge together within their specified trading period. Do and Faff (2012) also measure whether pairs trading is viable after considering transaction costs. The results are not very positive; pairs trading is unprofitable on average. However, better matched pairs that are formed within refined industry groups are mildly profitable.

1.2.2 Stochastic Spread Method

The stochastic spread method is an attempt by Elliott et al. (2005) to introduce a parametric model for pairs trading. The observed spread Y_k, which is defined as the difference between two prices, is modelled by the discrete process

$$Y_k = X_k + D\,\omega_k\,, \tag{1.3}$$

for $k = 0, 1, 2...$, where the ω_k are i.i.d. $\mathcal{N}(0,1)$ and $D > 0$.

The state variable X_k follows a discretized Ornstein-Uhlenbeck process at time $t_k = k\tau$ for $k = 0, 1, 2, \ldots$:

$$X_{k+1} - X_k = \tau b\left(\frac{a}{b} - X_k\right) + \sigma\sqrt{\tau}\,\epsilon_{k+1}\,, \tag{1.4}$$

where the ϵ_k are i.i.d. $\mathcal{N}(0,1)$ and independent of the ω_k in 1.3 , $a \in \mathbb{R}$, $b > 0$, $\sigma \geq 0$, and $\tau > 0$ is the time step.

Then $X_k \sim N(\mu_k, \sigma_k)$, where

$$\begin{aligned} \mu_k &= \frac{a}{b} - \frac{a}{b}\,(1 - b\tau)^k + (1 - b\tau)^k \mu_0 \\ \sigma_k^2 &= \sigma^2\tau\left[\frac{1 - (1 - b\tau)^{2k}}{1 - (1 - b\tau)^2}\right] + (1 - b\tau)^{2k}\,\sigma_0^2\,. \end{aligned} \tag{1.5}$$

As $k \to \infty$,

$$\begin{aligned} \mu_k &= \frac{a}{b} \\ \sigma_k^2 &= \frac{\sigma^2 \tau}{1-(1-b\tau)^2} \,. \end{aligned} \tag{1.6}$$

as long as $\tau > 0$ and $|1 - b\tau| < 1$.

Equation 1.4 can also be written in the form:

$$X_{k+1} = A + BX_k + C\epsilon_{k+1} \,. \tag{1.7}$$

with $A = a\tau$, $0 < B = 1 - b\tau < 1$, and $C = \sigma\sqrt{\tau}$.

Equations 1.3 and 1.7 are transition and measurement equations that are linear and Gaussian, which means that they can be used with the Kalman filter procedure. The Kalman Filter procedure is used by Elliott et al. (2005) to calculate the linear least square forecasts of the state vector X_k with the observed data through k:

$$\widehat{X}_{k+1|k} = \widehat{\mathrm{E}}\left[X_{k+1}|\gamma_k\right] . \tag{1.8}$$

where $\gamma_k = (y_k, y_{k-1}...y_1, x_k, x_{k-1}, ..., x_1)$.

The forecasts are calculated recursively, with $\widehat{X}_{1|0}$ being generated first, followed by $\widehat{X}_{2|1}, \widehat{X}_{3|2}, ...\widehat{X}_{k|k-1}$. The values of A, B, C, and D are estimated with the E-M Algorithm, as detailed by Shumway and Stoffer (1982).

Instead of using the discretized Ornstein-Uhlenbeck process as in 1.7, it is also possible to start with the continuous version $X_{k\tau}$ where $\{X_t | t \geq 0\}$ satisfies the stochastic differential equation

$$dX_t = \theta(\mu - X_t)\, dt + \sigma\, dZ_t \,. \tag{1.9}$$

Do et al. (2006) build on this method in their paper, and then discretize the transition equation to facilitate econometric estimation in a state space setting. One of the advantages of doing this is that there are explicit results for the first passage times for the standardized Ornstein-Uhlenbeck process.

The observation process is given by:

$$Y_t = X_t + D\omega_t \,. \tag{1.10}$$

As before, the mean-reversion in the spread is modelled by X_t and noise is modelled through ω_t. The model suggested by Elliott et al. (2005) is known as the Vasicek model for modelling interest rates. One of the primary concerns when modelling interest rates with the Vasicek model is that the model produces negative results, which is not seen in reality. However, this is not of concern here as the spread can definitely take on negative values while trading.

The model is very useful because there are closed form solutions for the conditional expected time and variance in a Vasicek model, given the current spread. This is shown below. For a function $f(X_t, t) = X_t e^{\theta t}$, applying Ito's lemma results in

$$df(X_t, t) = e^{\theta t}\, dX_t + \theta x_t e^{\theta t}\, dt,$$

and from Equation 1.2,

$$df(X_t, t) = \mu\theta e^{\theta t}\, dt + \sigma e^{\theta t}\, dZ_t \,.$$

Taking the integral from 0 to t on both sides,

$$X_t e^{\theta t} - X_0 = \int_0^t \mu\theta e^{\theta s}\, ds + \int_0^t \sigma e^{\theta s}\, dZ_s \,.$$

So

$$X_t = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \int_0^t \sigma e^{\theta s}\, dZ_s \,. \tag{1.11}$$

Furthermore, taking the conditional expectation of X_t given X_0 results in has

$$\mathrm{E}\left[X_t | X_0\right] = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}).$$

The conditional variance is derived as follows:

$$\mathrm{Var}\left(X_t | X_0\right) = \mathrm{Var}\left(\int_0^t \sigma e^{\theta(s-t)} dZ_s\right)$$

$$= \mathrm{E}\left[(\sigma \int_0^t e^{\theta(s-t)} dZ_s)^2\right] \tag{1.12}$$

$$= \mathrm{E}\left[\sigma^2 \int_0^t e^{2\theta(s-t)} ds\right] \tag{1.13}$$

$$= \frac{\sigma^2}{2\theta}[1 - e^{-2\theta t}]\,,$$

where 1.12 to 1.13 is a result of Ito's isometry.

The discretized version of Equation 1.11 is

$$X_k = X_{k-1} e^{-\theta\Delta} + \mu(1 - e^{-\theta\Delta}) + \epsilon_k\,, \tag{1.14}$$

where Δ is the time interval in years between two observations.

The discretized time measurement equation is

$$Y_k = X_k + D\,\omega_k\,.$$

These two discretized transition and measurement equations are linear and Gaussian, which means that they can be used with the Kalman filter procedure to get optimal estimates of the parameters θ, μ, σ, D.

Do et al. (2006) mention that this model is too restrictive as it assumes that the stocks will always return to equilibrium in the long run. This greatly restricts the number of plausible stocks available for the statistical arbitrage. Do et al. (2006) propose a model that generalizes the current stochastic spread model, the stochastic residual spread model.

In the stochastic residual spread model, there is an assumption that there is an equilibrium in the relative valuation of the two stocks measured by some spread. Any mispricing can be quantified by a residual spread function $G(R_t^A, R_t^B, U_t)$, where R_t^A and R_t^B are the returns of the two stocks, and U_t denotes an exogenous vector that may be needed to create the equilibrium.

The state space representation is very similar to the previous model 1.9, with the state being driven by X_t, the state of mispricing:

$$dX_t = \theta(\mu - X_t)\, dt + \sigma\, dZ_t\,, \tag{1.15}$$

and the observed mispricing:

$$Y_t = G_t = X_t + D\,\omega_t\,. \tag{1.16}$$

The difference here is that the observed mispricing G is driven by the Arbitrage Pricing Theory from Ross (1976). The APT model describes the return of a risky asset as the sum of the risk premiums times the exposure to each factor and the risk free rate. Do et al. (2006) assert that the relative APT on two stocks can be written as

$$R_t^A = R_t^B + \Gamma r_t^m + e_t\,, \tag{1.17}$$

where $r^m = R_m - r_f$ denotes the excess of market return over the risk free rate, $\Gamma = \beta_A - \beta_B$, where β_A and β_B describe the movement of A and B to the market, and e_t is a residual noise term.

The residual spread function is then defined as:

$$G_t = G\left(R_t^A, R_t^B, U_t\right) = R_t^A - R_t^B - \Gamma r_t^m\,. \tag{1.18}$$

The discrete state space model is then constructed with the transition equation being:

$$X_k = X_{k-1}\, e^{-\theta\Delta} + \mu(1 - e^{-\theta\Delta}) + \epsilon_k\,, \tag{1.19}$$

and the measurement equation:

$$Y_k = X_k + \Gamma r_k^m + D\omega_k\,. \tag{1.20}$$

Again being a linear and Gaussian state space model, the Kalman Filter and the E-M Algorithm can be used to estimate the linear least square forecasts of the state vector and the parameters of the state-space system.

1.2.3 Cointegration Method

The following section is focussed on the main topic of this thesis: the cointegration method. First is a discussion on the problems of using correlation in pairs trading, followed by a historical review of the cointegration approach and its limitations.

Discussion on Correlation

Using the concept of correlation has been a staple in investment analysis, being used extensively in both portfolio and risk management. However, the theory for correlation only works for stationary processes. Alexander and Dimitriu (2002) mention that the use of correlation analysis in many financial applications means that valuable information is lost in the process of making financial time-series stationary. This might occur in the process of taking the first differences of log prices so that all analysis is done on returns of assets instead of on the prices themselves. One advantage of using cointegration instead of correlation, is that cointegration allows the usage of all the information from the financial variables. Furthermore, a cointegration relationship characterizes the long run relationship of the time series' involved, whereas correlation is usually only a short run measure.

As such, pairs trading, which is predicated on the hypothesis that the stocks chosen have similar price movements in the long run, is clearly more suited to cointegration rather than correlation analysis.

Cointegration

Similar to the method proposed by Elliott et al. (2005), the cointegration method attempts to use statistics to show that a pair of stocks can have a mean-reverting return with the concept of cointegration, introduced by Engle and Granger (1987). For two time series that are integrated of order d, if there is a linear combination of the two that results in a time series of order $(d - b)$, $b > 0$, then the two time series are cointegrated of order (d,b),

which can be written as $CI(d, b)$. The most relevant case to pairs trading occurs when $d = b = 1$, which results in a stationary time series as the linear combination is integrated of order 0 (i.e. a stationary time series).

Because a weakly stationary process has a constant mean and a constant variance across time, when the process departs from the mean, it is expected to revert back eventually. The constant variance also restricts the process from departing too far from the mean. We refer to this property as *mean-reversion*. Hence, for a pair of cointegrated stocks, it is expected that the spread generated by the linear combination of the two stocks will be mean-reverting. A trading position of shorting the spread will then be taken when the spread is above its historical mean, and closed when it reverts back to the mean. Similarly, a long position in the spread will be taken when the spread is below its historical mean, and closed when it reaches the mean.

Engle and Granger (1987) also introduced the idea of capturing the dynamics of cointegration with an error correction model (ECM). In this model, there is an assumption that the two time series have a long-run equilibrium. If either of the time series move away from this equilibrium, the error correcting term will force a return towards the equilibrium. The error correction model representation is represented by:

$$\begin{aligned} \Delta Y_t &= \lambda_0 + \gamma_Y \left(Y_{t-1} - \alpha - \beta X_{t-1}\right) + \sum_{i=1}^{t-1} \lambda_{Y,i} \, \Delta Y_{t-i} + \sum_{i=1}^{t-1} \lambda_{X,i} \, \Delta X_{t-i} + \epsilon_{Y,t} \\ \Delta X_t &= \psi_0 + \gamma_X \left(Y_{t-1} - \alpha - \beta X_{t-1}\right) + \sum_{i=1}^{t-1} \psi_{Y,i} \, \Delta Y_{t-i} + \sum_{i=1}^{t-1} \psi_{X,i} \, \Delta X_{t-i} + \epsilon_{X,t} \,, \end{aligned} \tag{1.21}$$

where λ_0 and ψ_0 represent the deterministic trends in the time series, $Y_{t-1} - \alpha - \beta X_{t-1}$ represents the long-run equilibrium, the γ term represents the speed at which the time series reverts to the long-run equilibrium, the sums represent short-run lag dynamics, and the ϵ terms are white-noise. The γ terms must be opposite in sign to facilitate the return to the long-run equilibrium.

Given that the two time series are cointegrated, this model allows for simple forecasts given the past data. The essential step is then to ensure that the two time series are coin-

tegrated. This is done using the Engle and Granger 2-step approach (Engle and Granger (1987)). A regression is first performed with the two time series integrated of order 1:

$$Y_t = \alpha + \beta X_t + \epsilon_t \quad \text{for } t = 1...T. \tag{1.22}$$

The β term is known as the cointegration coefficient. The second step is that the estimated residuals $\widehat{\epsilon}_t$ from the regression are tested for stationarity using the Augmented Dickey-Fuller test.

It is worth noting that if the variables Y_t and X_t are cointegrated, it has been shown by Stock (1987) that the OLS estimates of α and β converge to their true values faster than the OLS estimates in the case where Y_t and X_t are stationary variables. Hence Stock (1987) has described this phenomenon as the regression yielding "superconsistent" estimators for α and β.

Augmented Dickey-Fuller Test

The Dickey-Fuller test was developed as a stationarity test. More specifically, Dickey and Fuller (1979) consider three different regression equations that can be used to test for the presence of a unit root. These regression equations represent a first-order autoregressive process, as follows:

$$\Delta Y_t = \gamma Y_{t-1} + \epsilon_t \tag{1.23}$$

$$\Delta Y_t = a_0 + \gamma Y_{t-1} + \epsilon_t \tag{1.24}$$

$$\Delta Y_t = a_0 + \gamma Y_{t-1} + a_2 t + \epsilon_t \,. \tag{1.25}$$

These equations represent a random walk, a random walk with drift, and a random walk with drift and a linear time trend respectively. The null hypothesis of $\gamma = 0$ is tested through the estimation of these regression equations by ordinary least squares. The estimates, γ and its standard error, are used towards a t-statistic that is used in conjunction with tables developed by Dickey and Fuller in the testing of this hypothesis.

The regression equations 1.23, 1.24, and 1.25, are extended in the Augmented Dickey-Fuller (ADF) test. They incorporate multiple lags of the time series in the regression:

$$\Delta Y_t = \gamma Y_{t-1} + \sum_{i=2}^{p} \beta_i \, \Delta Y_{t-i+1} + \epsilon_t \tag{1.26}$$

$$\Delta Y_t = a_0 + \gamma Y_{t-1} + \sum_{i=2}^{p} \beta_i \, \Delta Y_{t-i+1} + \epsilon_t \tag{1.27}$$

$$\Delta Y_t = a_0 + \gamma Y_{t-1} + a_2 t + \sum_{i=2}^{p} \beta_i \, \Delta Y_{t-i+1} + \epsilon_t \, . \tag{1.28}$$

The additional lags incorporated here are intended to render the residuals approximately independent. This is to facilitate the critical values tabulated by Dickey and Fuller, as those tables are simulated under the assumption of i.i.d. residuals. The issue of incorporating moving average components is not a problem. An invertible MA model can be represented by an AR model, and Ross (1984) showed that an unknown ARIMA$(p, 1, q)$ process can be well approximated by an ARIMA$(n, 1, 0)$ autoregression of order n. The selection of lag length can be determined by an information criterion such as the AIC.

A disadvantage of using the Engle-Granger two step method is that it is often not known which series should be chosen as the independent variable in the regression and which should be the dependent variable. Depending on the choices in this regard, the cointegration coefficient changes and testing takes much longer in overall procedure as testing for valid pairs is typically done on a vast number of assets. Phillips and Ouliaris (1990) and Johansen (1988) have proposed different tests which are independent of this choice. We will discuss the Johansen test in more depth later.

For a test of cointegration, recall that we have generated the estimated residuals from Equation 1.22, $\widehat{\epsilon}_t$. Then the typical test for stationarity is of the autoregressive form:

$$\Delta \, \widehat{\epsilon}_t = a_1 \, \widehat{\epsilon}_{t-1} + e_t \, . \tag{1.29}$$

As the residuals are being generated from a regression, there is no need for an intercept term. The null hypothesis for the test is then $a_1 = 0$. If this can be rejected, then the residuals can be concluded to be stationary, and hence, the two time series are cointegrated of order (1,1).

As mentioned previously, the additional lags in Equations (1.26), (1.27), and (1.28) are incorporate to render the residuals approximately independent so that the tables simulated by Dickey and Fuller can be used. However, another problem arises from the testing of stationarity of the residuals from a cointegrating regression.

Often in practice, when testing the residual time series obtained from the cointegrating regression for stationarity, it is not possible to use the Dickey-Fuller tables. This is because the residuals are being estimated, and as the values of α and β are minimizing the sum of squared residuals, the residuals are biased towards stationarity. This is a major problem when the number of variables used in the regression varies and when the sample size is small. MacKinnon (1990) developed critical values towards this issue using response surface analysis for any finite sample size.

Johansen Cointegration Test

As noted before, the Engle-Granger two step method has a disadvantage: it is not certain which of the variables should be picked as the the cointegrating regressor and which should be picked as the regressand. Additionally, the residuals are being estimated from the regressions, which has required the use of different critical values than the standard t-tables offer. Johansen (1988) developed a procedure that relies on maximum likelihood estimators and allows the testing for multiple cointegrating vectors. The procedure is a multivariate generalization of the Dickey-Fuller test that utilizes the relationship between the rank of a matrix and its characteristic roots.

Consider the following:

$$\begin{aligned} \Delta X_t &= A_1\, X_{t-1} - X_{t-1} + \epsilon_t \\ &= (A_1 - I)\, X_{t-1} + \epsilon_t \\ &= \pi\, X_{t-1} + \epsilon_t \end{aligned} \tag{1.30}$$

where $\pi = A_1 - I$, X_t and ϵ_t are $n \times 1$ vectors, A_1 is an $n \times n$ matrix of coefficients, and I is an $n \times n$ identity matrix.

Johansen (1988) showed that the rank of π is then the number of cointegrating vectors. If rank(π)=0, then there exists no linear combination of the processes in X_t that are stationary.

This can then also be generalized to allow for higher order autoregressive terms:

$$\Delta X_t = \pi\, X_{t-1} + \sum_{i=1}^{p-1} \Delta X_{t-i} + \epsilon_t\,, \tag{1.31}$$

where $\pi = (\sum_{i=1}^{p} A_i - I)$ and $\pi_i = -\sum_{j=i+1}^{p} A_j$.

The matrix π can be decomposed into the form of $\pi = \alpha\beta'$ of size $p \times r$. β is the matrix of cointegrating vectors and α represents the rate at which the variables return to the long-run equilibrium in the form of error-correcting coefficients. The parameter p can be selected again using maximum likelihood criterion such as the AIC.

The number of distinct cointegrating vectors is determined by checking the significance of the characteristic roots of π. Johansen's method involves finding the residuals e_{1t} and e_{2t} from the following two regressions:

$$\begin{aligned} \Delta X_t &= B_1\, \Delta X_{t-1} + ... + B_{p-1}\, \Delta X_{t-p+1} + e_{1t} \\ \Delta X_{t-1} &= C_1\, \Delta X_{t-1} + ... + C_{p-1}\, \Delta X_{t-p+1} + e_{2t}\,. \end{aligned} \tag{1.32}$$

Then the product moment matrices are calculated from these residuals: $S_{ij} = T^{-1} \sum_{t=1}^{T} e_{it} e'_{jt}$. The eigenvalues λ_i are obtained as the solutions to

$$|\lambda_i S_{22} - S_{12} S_{11}^{-1} S_{12}'| = 0\,. \tag{1.33}$$

These $\widehat{\lambda}_i$ are ordered such that $\widehat{\lambda}_1 > \widehat{\lambda}_2 > ...\widehat{\lambda}_n$, which are then used towards two test statistics Johansen derived. The first one, the trace statistic, tests the null hypothesis under a restricted model that the number of distinct cointegrating vectors is less than or equal to r against the alternative of the unrestricted model.

$$\lambda_{trace} = -T \sum_{i=r+1}^{n} \ln(1 - \widehat{\lambda}_i)\,, \tag{1.34}$$

for $r = 0, 1, 2, ...n - 1$.

The second test statistic, the maximal eigenvalue statistic, is used to test the null hypothesis of r cointegrating vectors against the alternative of $r + 1$ cointegrating vectors:

$$\lambda_{max} = -T \ln(1 - \widehat{\lambda}_{r+1})\,. \tag{1.35}$$

The Johansen methodology is very useful for cases when there is a possibility of multiple cointegrating vectors. However, for pairs trading, often the Engle-Granger two-step methodology is preferred for its simplicity. Note also that given the Johansen cointegration test is a procedure using maximum likelihood estimation, it also assumes Gaussianity on the distribution of the data. The Johansen cointegration test is therefore not an ideal test to use on stock data.

Use of Cointegration in Pairs Trading

Vidyamurthy (2004) takes a variation on the Engle Granger 2 step approach in terms of finding pairs of stocks suitable for pairs trading. Instead of requiring the residuals to be stationary, Vidyamurthy (2004) only requires that they be mean-reverting. This is done in two ways: modelling the residuals parametrically with a mean-reverting process such as the ARMA process, or by measuring the number of times that the time series transitions across its long time mean. This measurement of transitions is known as the number of zero crossings of a time series. Vidyamurthy (2004) chooses not to define pairs for trading using the strict rules of cointegration, because it limits the number of actual pairs in the stock market that this strict definition can be applied to.

This is a very interesting and valid point to make, but the criteria that Vidyamurthy (2004) selects pairs with has a lot of room for error. In the worst case scenario, the estimated mean-reverting spread series could be very different from the true series, making any attempts at statistical arbitrage highly risky and potentially very unprofitable.

Lin et al. (2006) performed another study on pairs trading using the cointegration methodology. Three assumptions were made about the pairs to simplify the arbitrage strategy:

Assumption 1. The two-share price series are always cointegrated over the pairs trading period;

Assumption 2. The long and short positions always apply to the same shares in the share pair. For any trade, S_1 always represents the short position while S_2 represents the long position;

Assumption 3. At the opening of any trade, the price for the shorted share S_1 is always higher than the price of the share in long position S_2.

Define

$N_{S_k}(t_j)$ is the number of shares of S_k at time t_j

$P_{S_k}(t_j)$ is the price of S_k at time t_j

for $k = 1, 2$ and $j = 0, c$ (where c is the time at the close of the positions).

At the opening of the trade, $N_{S_2}(t_0)$ shares of S_2 are bought for $N_{S_2}(t_0)\, P_{S_2}(t_0)$. $N_{S_1}(t_0)$ shares of S_1 are sold short to fund this purchase for a gain of $N_{S_1}(t_0)\, P_{S_1}(t_0)$. At the close, the shares of S_2 are sold for $N_{S_2}(t_c)\, P_{S_2}(t_c)$ and the shares of S_1 are returned at a price of $N_{S_1}(t_c)\, P_{S_1}(t_c)$. It should be noted that the stocks being considered are non-dividend paying stocks.

Thus the profit equation is given as

$$TP_t = N_{S_2}(t_0)\left[P_{S_2}(t_c) - P_{S_2}(t_0)\right] + N_{S_1}(t_0)\left[P_{S_1}(t_0) - P_{S_1}(t_c)\right]. \tag{1.36}$$

Assuming that the trader wants a positive profit, there is a starting condition of $TP_t > K > 0$ where K is determined by the trader. Also the opening trades must be covered entirely by the short-sell, so we need

$$N_{S_1}(t_0)\, P_{S_1}(t_0) \geq N_{S_2}(t_0)\, P_{S_2}(t_0). \tag{1.37}$$

Lin et al. (2006) define two conditions for opening and closing trades. These *opening trade conditions* and *closing trade conditions* are denoted OTC and CTC respectively. The OTC states that a trade can be opened if for a positive integer a,

$$P_{S_1}(t_0) - \beta\, P_{S_2}(t_0) = \epsilon_{t_0} > a > 0\,. \tag{1.38}$$

This strategy requires $\beta > 0$, as we are selling S_1 and using the funds from that to buy βS_2. In practice, this condition occurs in many cointegrated share price series, so it is not very restrictive.

For both 1.37 and 1.38 to be true, a condition on the number of shares bought and sold is needed. For a buyer to purchase β shares of S_2, n shares of S_1 must be sold short. For $n = 1$, the initial outlay is then:

$$P_{S_1}(t_0) - \beta\, P_{S_2}(t_0) = \epsilon_{t_0} > 0 \tag{1.39}$$

The profit at time t_c can also be calculated:

$$\begin{aligned} &N_{S_2}(t_0)\,[P_{S_2}(t_c) - P_{S_2}(t_0)] + N_{S_1}(t_0)\,[P_{S_1}(t_0) - P_{S_1}(t_c)] \\ &= \beta\,[P_{S_2}(t_c) - P_{S_2}(t_0)] + [\epsilon_{t_0} + \beta\, P_{S_2}(t_0) - \epsilon_{t_c} - \beta\, P_{S_2}(t_c)] \\ &= [\epsilon_{t_0} - \epsilon_{t_c}]\,. \end{aligned} \tag{1.40}$$

The trading strategy can be summarized in several steps. They name this strategy the cointegrating coefficient weighting (CCW) strategy as the dollar amounts of investment in each pair depends on the cointegrating coefficients.

Step 1. Select a, b such that $a > b$. Lin et al. (2006) set b to be the mean of ϵ_t and a to be $b + k\sigma$ for varying values of k.

Step 2. Open a trade at time t_0 when $P_{S_1}(t_0) > P_{S_2}(t_0)$ and when 1.38 is true.

Step 3. Buy β shares of S_2 and sell 1 share of S_1 at time t_0

Step 4. Close the trading positions when $\epsilon_{t_c} < b$.

Then, the profit from the trade will be

$$\begin{aligned} &(\epsilon_{t_0} - \epsilon_{t_c}) \\ &\geq a - b \\ &\geq b + k\sigma - b \\ &\geq k\sigma \end{aligned} \tag{1.41}$$

since $\epsilon_{t_0} > a$ and $\epsilon_{t_c} < b$.

The strategy outlined here is opened when $\epsilon_{t0} > a$. This is a condition that means that the price of S_1 is overvalued compared to the price of S_2, which is undervalued. Hence S_1 is sold short and β shares of S_2 are bought long. This is in fact the same as shorting the spread created by the difference of $P_{S_1} - \beta\, P_{S_2}$ should be shorted until the spread reaches the equilibrium value (the mean of the historical spread).

The strategy in reverse can be applied when $\epsilon_{t0} < -a$. Here, the price of S_1 is undervalued compared to the price of S_2, which is overvalued. Then S_1 is bought long and β shares of S_2 are sold short, which is the same as going on on the spread. The position is closed when the spread hits the historical mean.

1.3 Application of Pairs Trading on Data

Some examples will be provided below to provide a better understanding of how some of these pairs trading methods work. By generating data from simulations we can see how the arbitrage works with known outcomes before applying it further to real data.

1.3.1 Minimum Distance Method

The data used in the following example are mainly generated from a simplified form of equation 1.21:

$$\begin{aligned} \Delta Y_t &= \gamma_Y \left(Y_{t-1} - X_{t-1}\right) + \epsilon_{Y,t} \\ \Delta X_t &= \gamma_X \left(Y_{t-1} - X_{t-1}\right) + \epsilon_{X,t} \,. \end{aligned} \tag{1.42}$$

Four pairs of cointegrated prices are generated, and one extra set is generated from two different ARIMA(p,d,q) models. This is done to show that the cointegrated pairs tend to be the ones matched up by both the minimum distance method and the cointegration method with varying values of γ_Y and γ_X, and different means and standard deviations for $\epsilon_{Y,t}$ and $\epsilon_{X,t}$.

As in the minimum distance method, the sum of squared deviations are calculated for each possible pair out of the $\binom{10}{2}$ total pairs in the 10 generated stock prices. The 5 pairs with the lowest sum of squared deviations are chosen to be traded together. The spread is calculated here simply as the difference between the two prices. Of the 1000 stock price values generated by the models, the first 600 data points are used as a training set to determine the minimum distance pairs to be used for trading. The last 400 data are used as the test set on which the trades are made. A simple trading rule is established: if the spread price exceeds two standard deviations above or below the mean spread price, a position is opened. The mean and standard deviations for the spread are calculated using the entire history out of the training set. When the spread converges back to the mean, the position is closed. The positions are also closed at the end of the trading period (at $t = 1000$ days), regardless of whether or not the spread is near the mean price or not. Transaction costs are not considered here for simplicity. Figure 1.1 shows the price paths of the generated asset pairs.

Trades that result from the pairs generated from the ECM move together as expected and as a result, the trading rule results in a profit. The last pair that was chosen with the minimum distance method was, also as expected, not a particularly well behaving pair. This is because one stock is generated from one of the ECM, and the other follows an

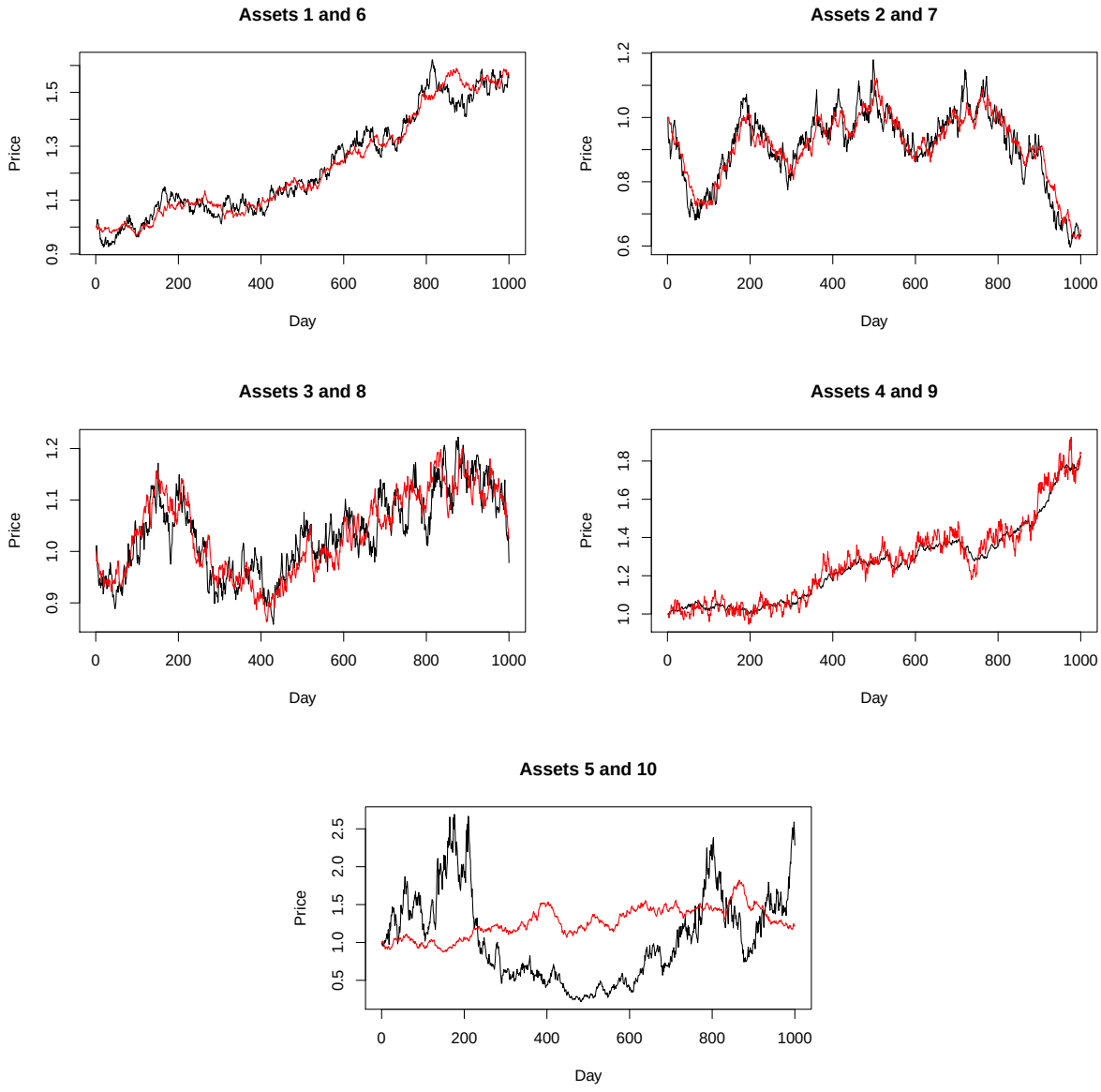

Figure 1.1: The price paths for ten stocks of 1000 days each. The first four pairs are simulated from a simplified ECM.

ARIMA process. The spread here is not stationary and thus it is dangerous to trade on such a spread. The example here demonstrates a negative profit value when the spread is not generated necessarily mean-reverting. Figure 1.2 and the first pair in Figure 1.3 show the spreads and their trade positions for the pairs generated by the ECM. The last pair in figure 1.3 shows the spread and the trade positions of the non-stationary spread pair. This demonstrates the importance of finding pairs which have mean-reverting spreads. With real data, finding pairs that have small sum of squared deviations might not necessarily translate to a profitable pair in the future because the spread may not be stationary. Previous papers regarding the minimum distance method often have tried to minimize this risk by only choosing the pairs with the lowest minimum distance. However, as the cointegration method relies on the concept of finding stationary spreads, there is a much stronger case for mean-reversion in cointegrated pairs than when compared to the minimum distance method. Hence, it is useful to also examine an application of the cointegration method to simulated data.

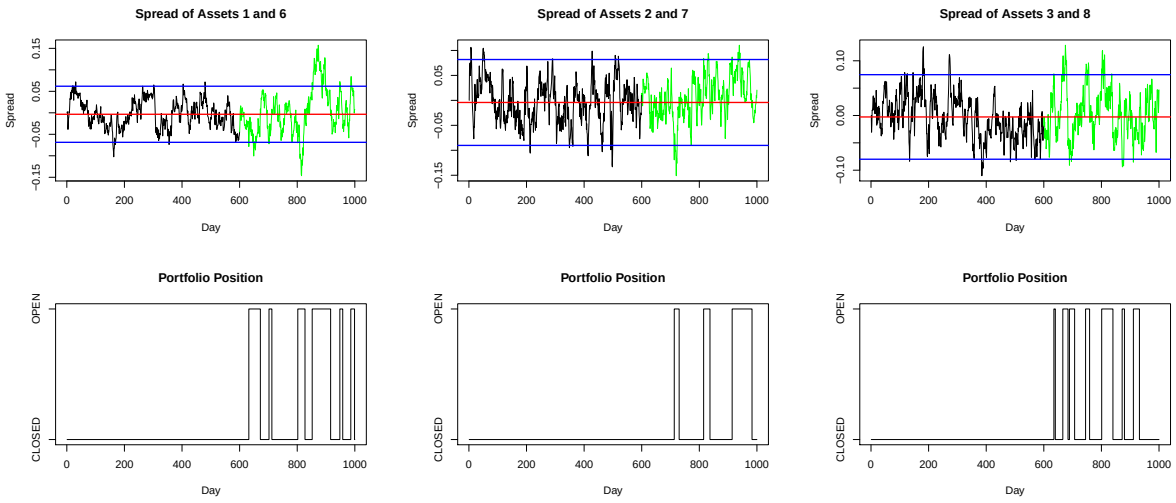

Figure 1.2: The spreads for the first three pairs in the minimum distance simulation and the days that the trade positions are open and closed. The black portion of the spread represents the training set and the green portion of the spread represents the test set. The upper bounds and lower bounds of the spreads (the mean +/- 2 standard deviations) are represented by the blue horizontal lines, and the red line represents the historical mean of the training set.

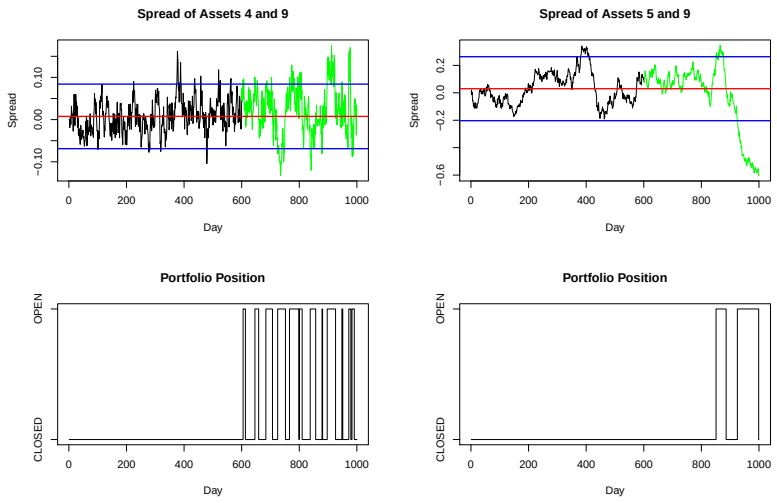

Figure 1.3: The spreads for the fourth and fifth pairs in the minimum distance simulation and the days that the trade positions are open and closed. The black portion of the spread represents the training set and the green portion of the spread represents the test set. The upper bounds and lower bounds of the spreads (the mean +/- 2 standard deviations) are represented by the blue horizontal lines, and the red line represents the historical mean of the training set.

1.3.2 Cointegration Method

The data used in this following example are again generated from a simplified form of equation 1.21, but the relationship between Y_{t-1} and X_{t-1} is slightly more generalized with an intercept and a coefficent term for X_{t-1}:

$$\begin{aligned} \Delta Y_t &= \gamma_Y \left(Y_{t-1} - \alpha - \beta X_{t-1}\right) + \epsilon_{Y,t} \\ \Delta X_t &= \gamma_X \left(Y_{t-1} - \alpha - \beta X_{t-1}\right) + \epsilon_{X,t} \,. \end{aligned} \tag{1.43}$$

Ten pairs of cointegrated prices are generated, with the first five pairs using the simple relationship from the minimum distance method ($\alpha = 0$, $\beta = 1$), and the next five pairs with varying α and β. For each pair, 1000 data points are generated again, with varying values of γ_Y and γ_X, with different means and standard deviations for $\epsilon_{Y,t}$ and$\epsilon_{X,t}$. The price paths for the 20 assets can be seen in Figure 1.4.

Here the Engle Granger (EG) two step methodology is used in determining whether each pair of stock prices is cointegrated. As mentioned before, there are several weaknesses to using this methodology. The decision on which variable to take as the regressor and which as the regressand is a problem. As such, we will require both the EG and the Johansen methodologies to pass for cointegration before labelling a pair as such. Small sample sizes and the number of variables being considered in the cointegration inhibit the use of the ADF test for stationarity. This is not as much of a problem as the previous one because only two stocks are considered for cointegration at each time. As well, the number of sample values we are using for the training period is at minimum a year of trading days. Thus, we consider the ADF test a suitable choice for our simulations and tests.

However, a caveat to note here is that because the time series are generated by the ECM, the prices are not necessarily integrated of order 1. Some of the time series are generated as stationary processes to begin with. As such, even though we have generated mean reverting pairs, the cointegration method does not necessarily recognize these as suitable options to trade on. Following the test for integration, they are fitted to a linear model for estimation of α and β. The spread is calculated from the residuals of the model,

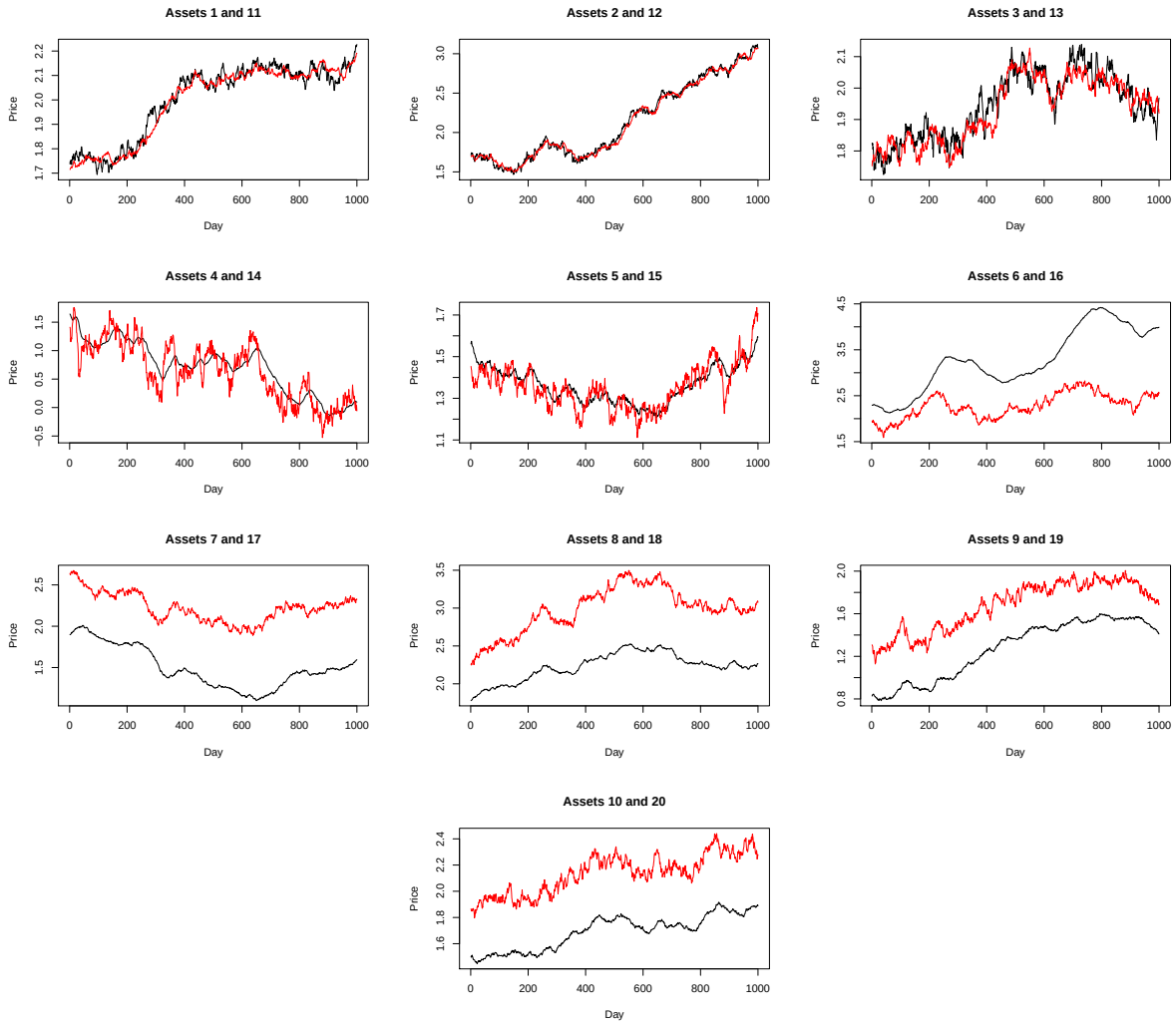

Figure 1.4: The simulated price paths for ten stocks of 1000 days each. The first five pairs are simulated from the simplified ECM as in the minimum distance method example. The next five pairs have varying intercepts and coefficient terms.

and tested for stationarity through the ADF test. Here, the assumption is that the errors follow an AR(1) process. If this test of stationarity is passed, then the same procedure is done on the same two pairs but with the regressor and regressand switched. If these

tests are passed, then we consider the pair of prices to be cointegrated and the trading rule established by Lin et al. (2006) is used.

It is possible that pairs that were not generated together using the ECM can be cointegrated, and this is certainly the case in the example. It is still possible to trade on these pairs, but a trade must proceed with caution even after finding the tests for cointegration are passed, as the possibility of false positive is a definite problem. The possibility of a series being cointegrated over the training period and then diverging is also a problem. Thus the training period cannot be too long to avoid including data where there are structural breaks, but also cannot be too short so that the trading rule can be established well. For this simulation, an arbitrary value of 600 time points was used for the training period. The rest of the 400 time points generated were used for the trading period. The results can be seen in Figure 1.5.

1.3.3 Application of the Cointegration Method to Stock Data

In this section, we will apply the same methodology used in the simulation in the above example to real data. As it has been mentioned before, pairs trading is a statistical arbitrage strategy. Ideally, the strategy would be risk-free and result in profits based on the assumption that the spreads are mean-reverting. Unfortunately, there are other items of importance to consider. Again, transaction costs that would cut returns significantly are not considered here for the strategy. As well, it was mentioned that the price of the short stock sold should cover the price of the long stock, and hence, there should not need to be any capital invested at the beginning of the trades. However, in reality, brokers require a margin account on the side. Since shorting a stock is the act of selling a borrowed stock, this margin account is used as a guarantee that the short seller will be able to pay up, as well as accounting for the fact that the shorted stock may rise in price. Regulation T, stipulated by the Federal Reserve, states that 50% of the value of the shorted stocks must be in the margin account at the beginning of the sale. Following the initial sale, the maintenance margin is 25%, meaning that the account must have 25% or more of the value

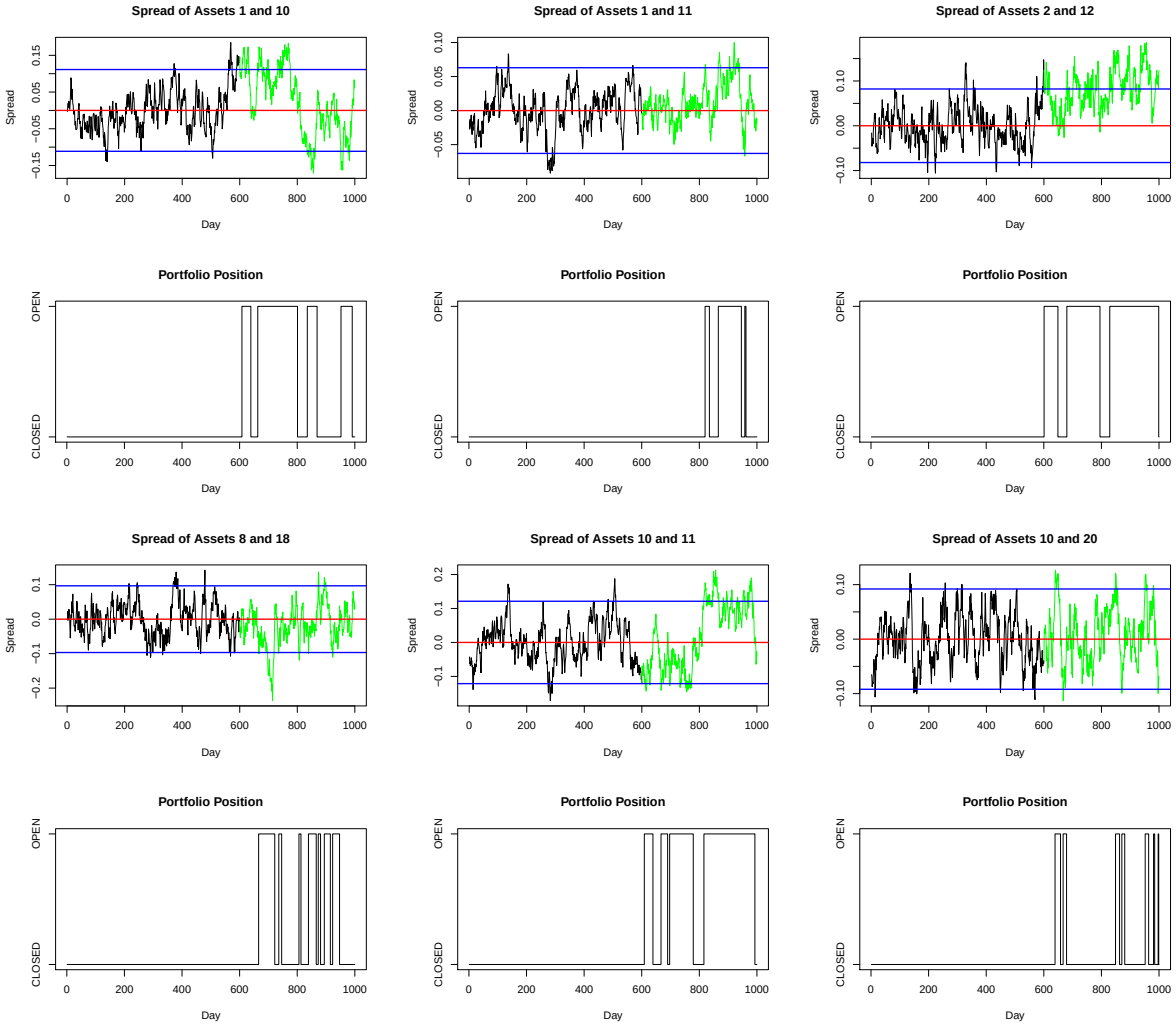

Figure 1.5: The spreads for the six cointegrated pairs in the cointegration simulation and the days that the trade positions are open and closed. The first 600 days comprise the training set and is indicated in black. The test spread is for the next 400 days and are labeled in green. The red line represents the historical mean of the training set. The blue lines represent the upper and lower bounds of the trades, given by the mean +/- 2 standard deviations of the training set.

of the shorted stocks at any point, but as this may differ from broker to broker, we will not consider that in our return calculations. Hence, we will only consider the simple return r for the trades as follows:

$$r = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{k_i}{S_{li} + 0.5 S_{si}} \right). \tag{1.44}$$

where N is the total number of trades, k_i is the profit or loss from trade i, S_{li} is the value of the long stock at the open of trade i, and S_{si} is the value of the short stock at the open of trade i.

The total profit TP and average profit per trade AP will also be calculated as follows:

$$TP = \sum_{i=1}^{N} k_i. \tag{1.45}$$

$$AP = \frac{TP}{N} \tag{1.46}$$

The stock prices are gathered from the constituents of the NASDAQ 100. There are 107 constituents in this Index, made up the largest non-financial companies listed on the NASDAQ. We only consider the companies with historical stock data as far back as May 2009. Although historical data was available for many companies before this, we wanted to avoid the stock crash of 2008. As a result, only 91 constituents were actually considered in our tests.

A time frame for the training and the trading periods needed to be selected: 1, 2 and 3 years were considered for the training period. The number of cointegrated pairs found for each training period was 17, 21 and 13 respectively. We chose to focus on the 3 year training period rather than the 1 and 2 year training periods. This was because as the training spreads generated by the 1 and 2 year spreads did not revert back to the mean often, although having passed the ADF test and the Johansen test. Two different trading periods were used: 6 months and 12 months.

1.3.4 Upper and Lower Bound of Two Standard Deviations from the Mean

Recall that in the trading rule established by Lin et al. (2006), we choose a value k for which the upper and lower bounds ($b + k\sigma$ and $b - k\sigma$) are established, where b is the historical mean and σ is the historical standard deviation of our training spread. In our first simulation we test $k = 2$. Note that because of this, although 13 pairs of stocks were found to be cointegrated in the training set, only 11 were actually traded on in the first test period because the threshold of two standard deviations was exceeded at some point. The training periods and the test periods can be seen in Table 1.1. Two sets of 6 month trading periods are tested, one immediately following the other. This emulates a cointegrating pair being traded in practice. However, for the second testing period, the current cointegrating pairs are updated with the most recent 3 yeras of training data, and are tested for cointegration again. This ensures that the pairs are still cointegrated before trading once again. In the 12 month trading period, this is not done and as such, pairs that are cointegrated at the beginning but not so after 6 months are still traded on. It is important to note that only 3 of the 13 cointegrated pairs at the end of the first training period are still cointegrated at the end of the second training period. This is evidence that cointegration relationships may undergo structural breaks and change or disappear altogether.

The training and test spreads for the 6 month test spread for the testing period May 21 2012 - November 21 2012 can be seen in Figures 1.6 and 1.7. The corresponding results are summarized in Tables 1.2, 1.3, and 1.4.

For the testing period November 21 2012 - May 28 2013, the spreads can be seen in Figure 1.8. The corresponding results are summarized in Table 1.5.

The 12 month test spread from the period May 21 2012 - May 28 2013 can be seen in Figures 1.9, 1.10, and 1.11. The corresponding results are summarized in Tables 1.6, 1.7, and 1.8.

The 3 year training period and the 6 month trading periods	
Training Period	**Testing Period**
May 20 2009 - May 21 2012	May 21 2012 - November 21 2012
November 18 2009 - November 21 2012	November 21 2012 - May 28 2013
The 3 year training period and the 12 month trading period	
Training Period	**Testing Period**
May 20 2009 - May 21 2012	May 21 2012 - May 28 2013

Table 1.1: The training and testing periods for the 91 elligible stocks in the NASDAQ 100

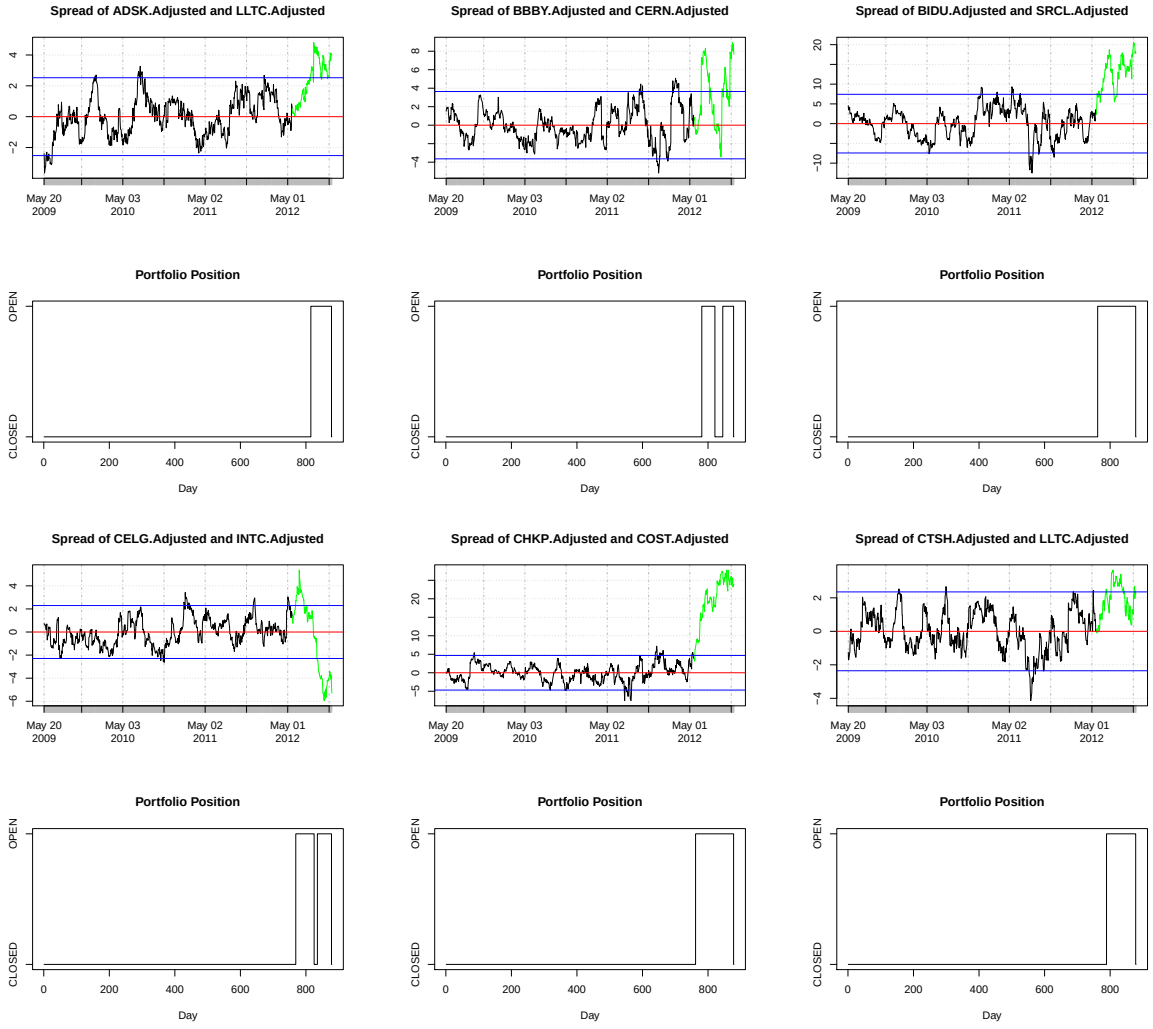

Figure 1.6: The training spread (in black) and the 6 month test spread (in green) for the first six cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012.

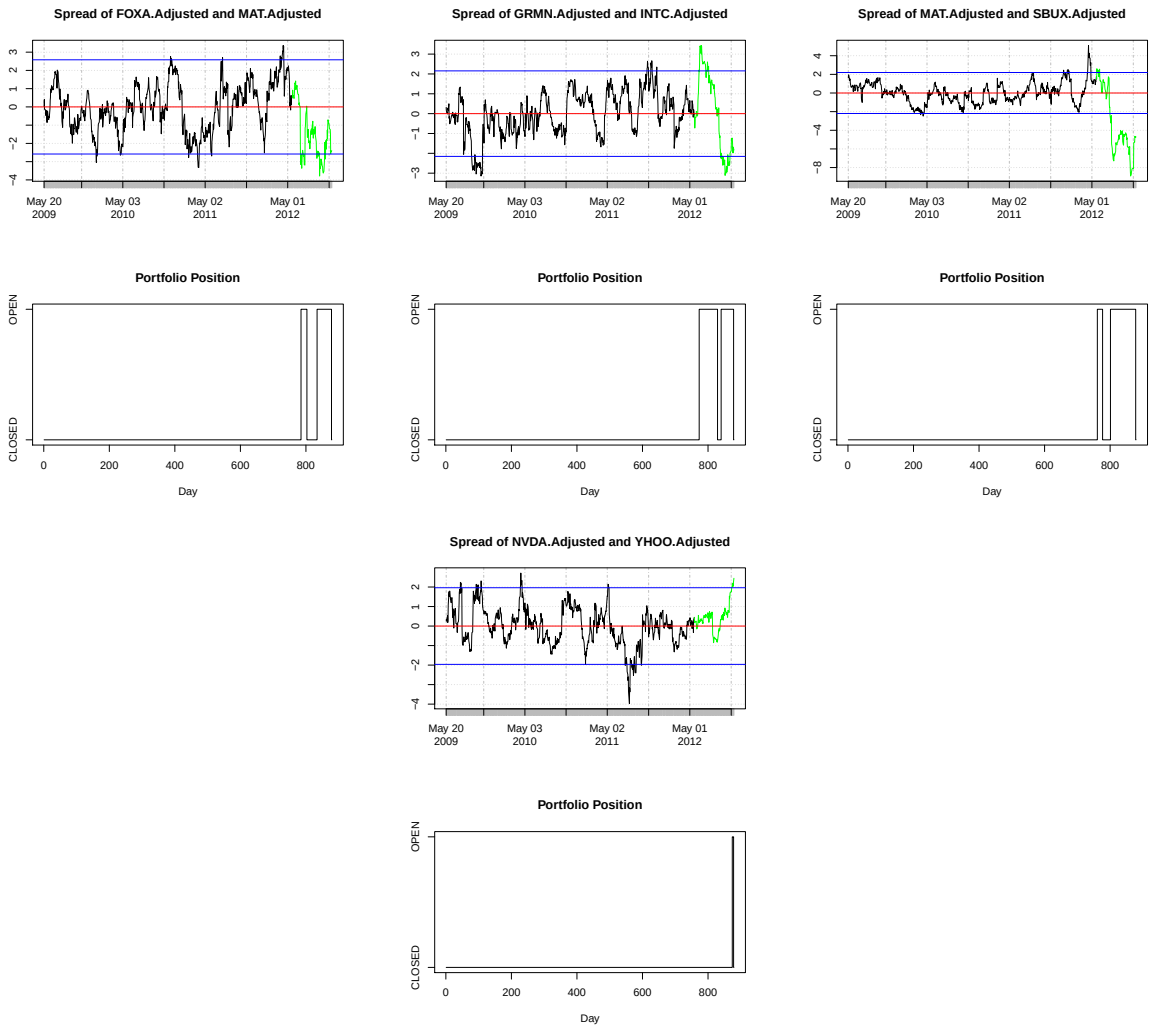

Figure 1.7: The training spread (in black) and the 6 month test spread (in green) for the last four pairs cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012.

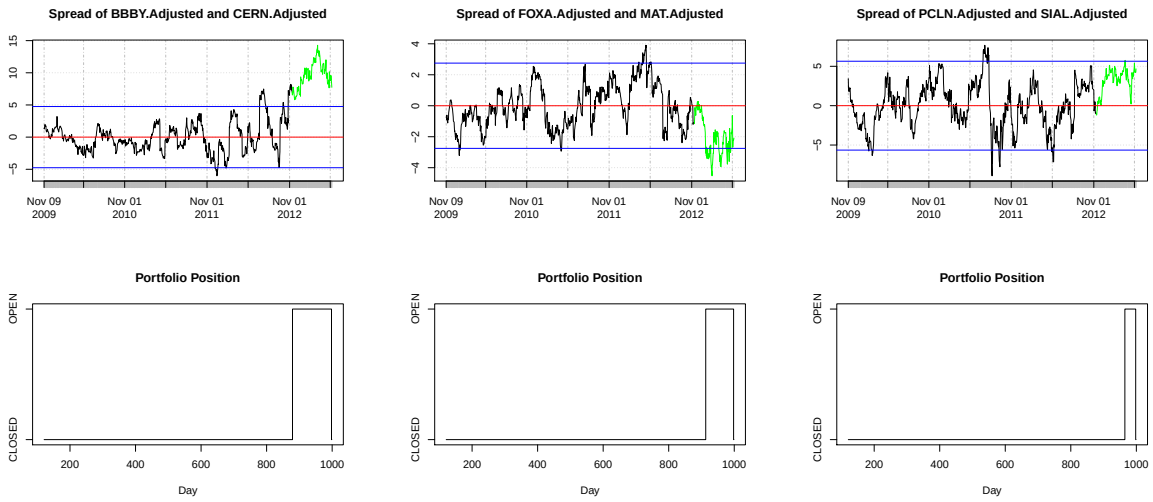

Figure 1.8: The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period November 21 2012 - May 28 2013.

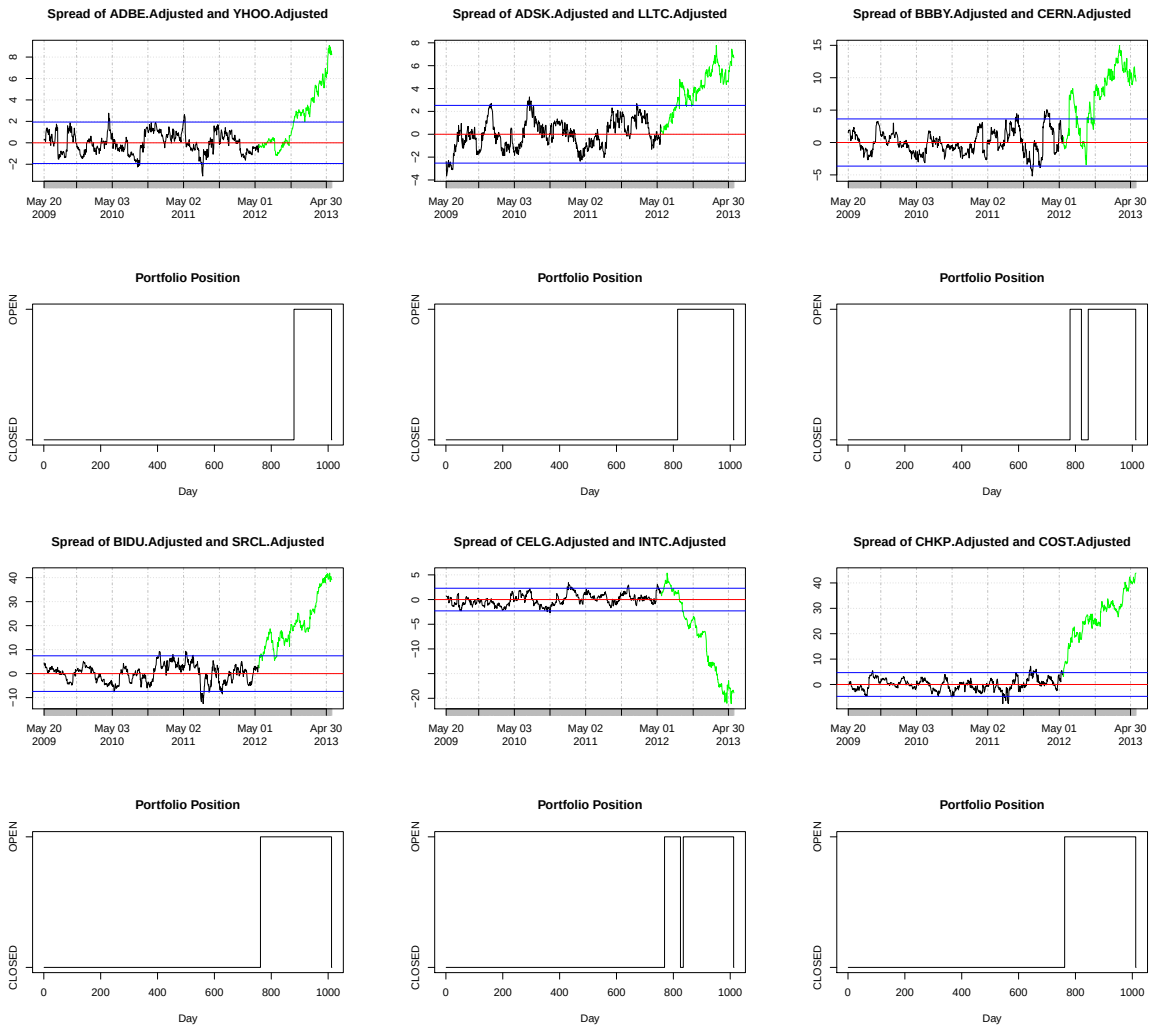

Figure 1.9: The training spread (in black) and the 12 month test spread (in green) for the first 6 cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

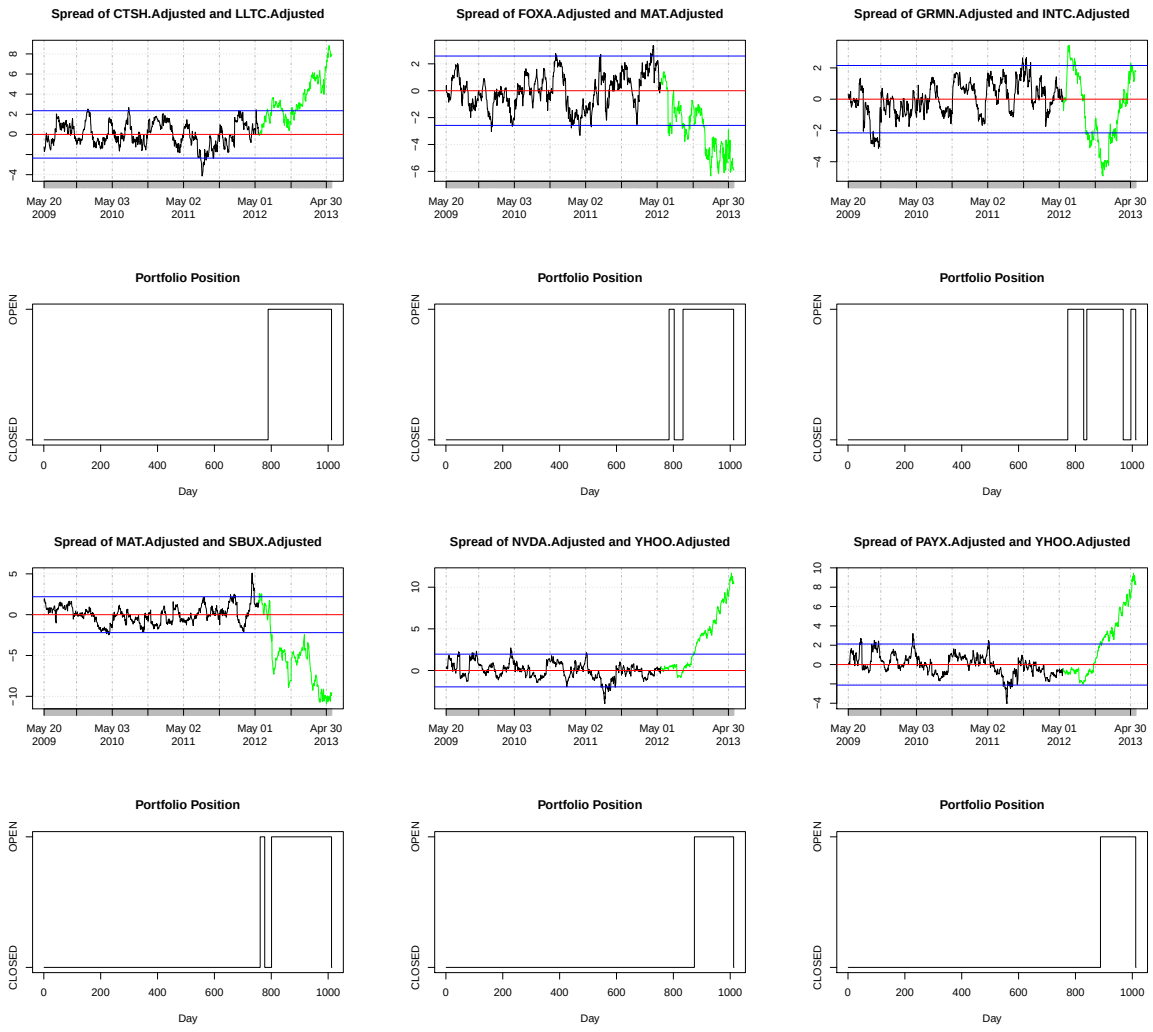

Figure 1.10: The training spread (in black) and the 12 month test spread (in green) for the 7th to 12th cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

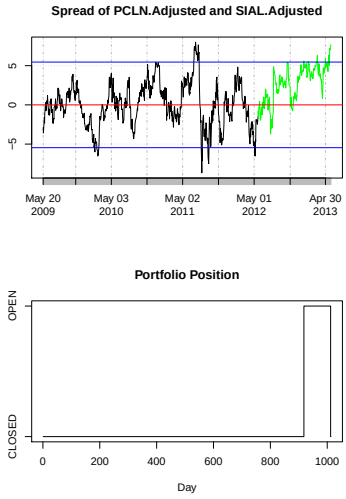

Figure 1.11: The training spread (in black) and the 12 month test spread (in green) for the 13th cointegrated pair using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

Asset Pair	ADBE,YHOO	ADSK,LLTC	BBBY,CERN	BIDU,SRCL	CELG,INTC
Total Profit	-0.492	-1.049	3.473	-13.513	-2.45
Total # of Trades	1	1	2	1	2
Avg Profit per Trade	-0.492	-1.049	1.736	-13.513	-1.225
Return	-1.99%	-2.41%	6.42%	-11.02%	-6.64%
Annualized Return	-3.97%	-4.82%	12.85%	-22.05%	-13.28%

Table 1.2: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pairs 1 to 5)

Asset Pair	CHKP,COST	CTSH,LLTC	FOXA,MAT	GRMN,INTC	MAT,SBUX
Total Profit	-17.582	0.452	4.653	0.746	-0.2
Total # of Trades	1	1	2	2	2
Avg Profit per Trade	-17.582	0.452	2.326	0.373	-0.1
Return	-16.82%	1.09%	10.11%	2.27%	-0.26%
Annualized Return	-33.63%	2.17%	20.22%	4.55%	-0.53%

Table 1.3: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pairs 6 to 10)

Asset Pair	NVDA,YHOO
Total Profit	-1.25
Total # of Trades	1
Avg Profit per Trade	-1.25
Return	-5.22%
Annualized Return	-10.43%

Table 1.4: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - November 21 2012 (pair 11)

Asset Pair	BBBY,CERN	FOXA,MAT	PCLN,SIAL
Total Profit	-1.582	-0.244	-1.218
Total # of Trades	1	1	1
Avg Profit per Trade	-1.582	-0.244	-1.218
Return	-3.04%	-0.47%	-1.11%
Annualized Return	-6.08%	-0.94%	-2.22%

Table 1.5: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 2 standard deviations from November 21 2012 - May 28 2013 (pairs 3,8,13)

Asset Pair	ADBE,YHOO	ADSK,LLTC	BBBY,CERN	BIDU,SRCL	CELG,INTC
Total Profit	-6.226	-4.155	1.337	-31.511	-13.997
Total # of Trades	1	1	2	1	2
Avg Profit per Trade	-6.226	-4.155	0.669	-31.511	-6.998
Return	-25.13%	-9.55%	2.31%	-25.7%	-40.45%
Annualized Return	-25.13%	-9.55%	2.31%	-25.7%	-40.45%

Table 1.6: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 1 to 5)

Asset Pair	CHKP,COST	CTSH,LLTC	FOXA,MAT	GRMN,INTC	MAT,SBUX
Total Profit	-37.583	-5.445	0.005	5.076	-4.606
Total # of Trades	1	1	2	3	2
Avg Profit per Trade	-37.583	-5.445	0.002	1.692	-2.303
Return	-35.95%	-13.09%	0.78%	15.49%	-11.7%
Annualized Return	-35.95%	-13.09%	0.78%	15.49%	-11.7%

Table 1.7: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 6 to 10)

Asset Pair	NVDA,YHOO	PAYX,YHOO	PCLN,SIAL
Total Profit	-8.354	-6.065	-2.126
Total # of Trades	1	1	1
Avg Profit per Trade	-8.354	-6.065	-2.126
Return	-34.86%	-23.38%	-1.99%
Annualized Return	-34.86%	-23.38%	-1.99%

Table 1.8: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 2 standard deviations from May 21 2012 - May 28 2013 (pairs 11 to 13)

1.3.5 Upper and Lower Bound of One Standard Deviation from the Mean

The same tests on the stock data from the last section are repeated with a threshold of one standard deviation from the mean. The trades here find the same cointegrated pairs as in the previous test, as we are using the same training data. The only difference is that with a lower threshold, the number of cointegrated pairs stays the same at 13, but all of them are traded on in the trading period. The possible profits are not as large for each trade as the quantity $k\sigma$ is now smaller, but there are more possible trades that open and close. This is both an advantageous property and a disadvantageous one at the same time. For the pairs that diverge significantly from the mean, the higher trading upper and lower bounds from using 2 standard deviations provides more of a safety net against loss as the trades are not opened as close. However, this can be protected against by using stop-loss triggers in practice. The advantage of using 1 standard deviation can be highlighted by comparing the three pairs that have been found to still be cointegrated at the 6 month mark. That is, comparing pairs 3,8 and 13, we can see that the profits are higher as more trades are executed since the threshold for profit is not as extreme. This can be seen in the Figures 1.19 and 1.16.

The training and test spreads for the 6 month test spread for the testing period May 21 2012 - November 21 2012 can be seen in Figures 1.12 and 1.13. The corresponding results are summarized in Tables 1.9, 1.10, and 1.11.

For the testing period November 21 2012 - May 28 2013, the spreads can be seen in Figure 1.15. The corresponding results are summarized in Table 1.12.

The 12 month test spread from the period May 21 2012 - May 28 2013 can be seen in Figures 1.16, 1.17, and 1.18. The corresponding results are summarized in Tables 1.13, 1.14, and 1.15.

The most important thing to note in these tests for cointegration on stock data is that there are many false positives from the cointegration test. Traders should be very cautious

of the results from the ADF test and the Johansen test when trying to find cointegrated pairs to trade on, as the cointegration relationship either changes over time, breaks down, or doesn't exist at all.

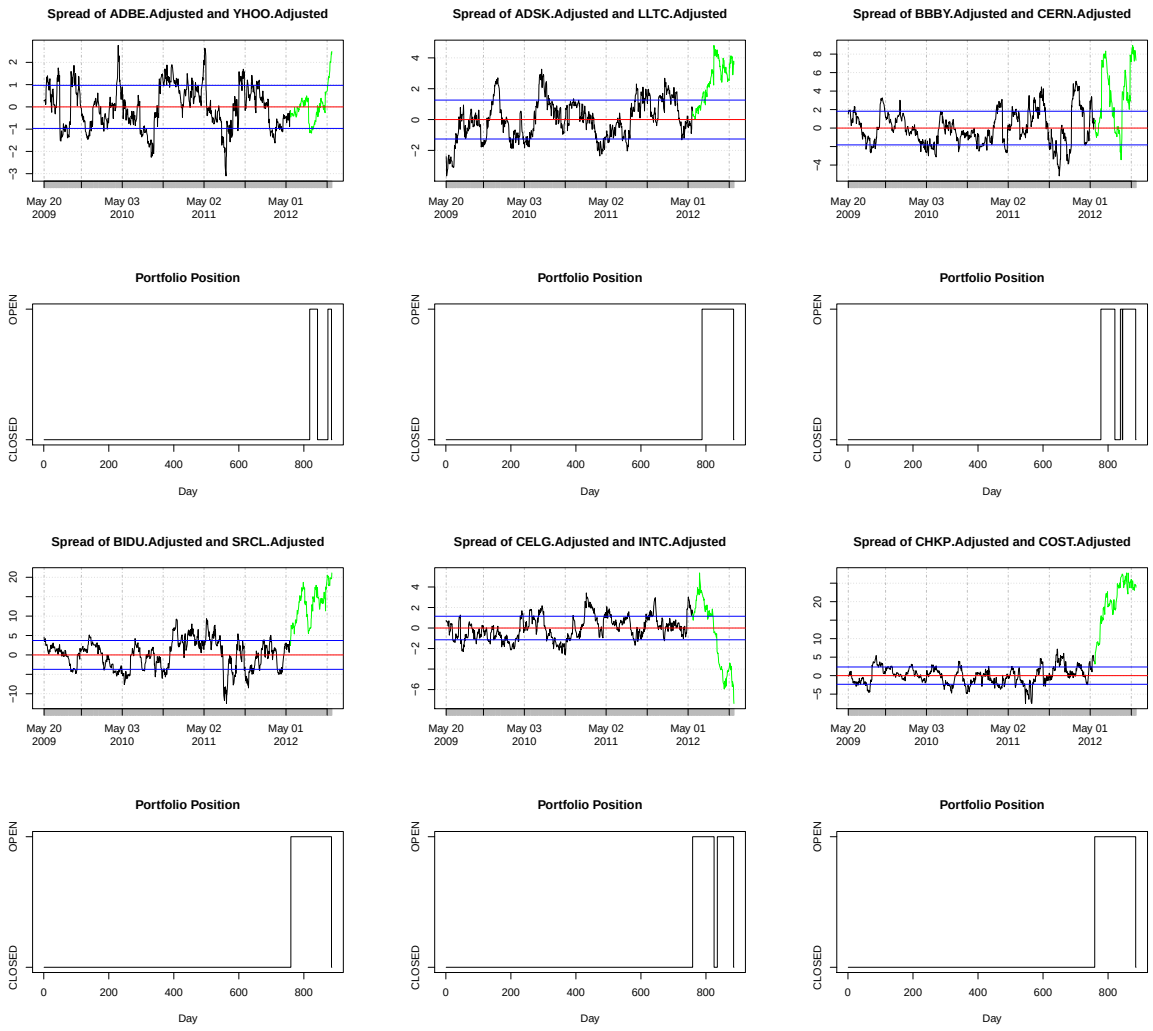

Figure 1.12: The training spread (in black) and the 6 month test spread (in green) cointegrated pairs (1 to 6) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012.

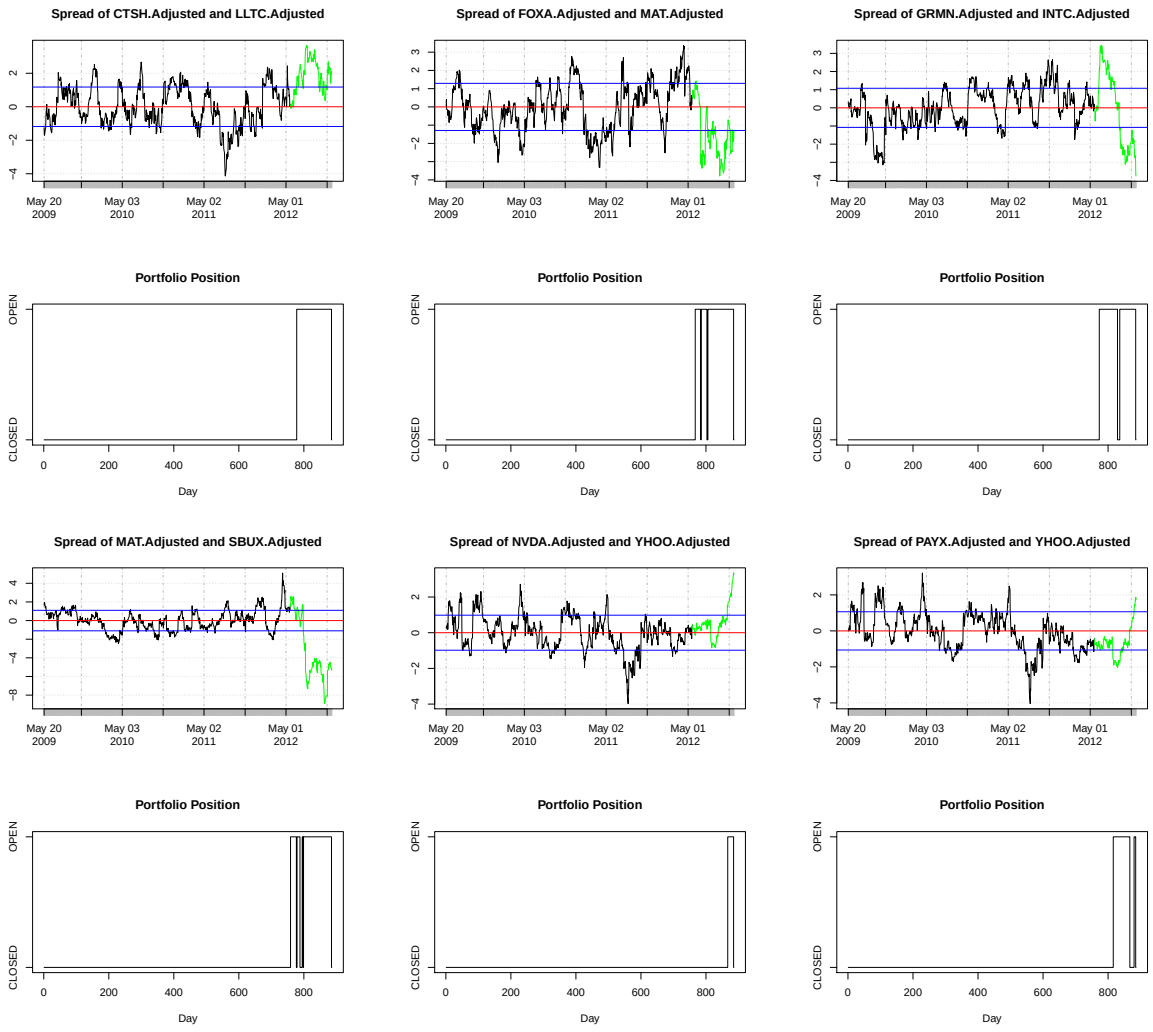

Figure 1.13: The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs (6 to 12) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012.

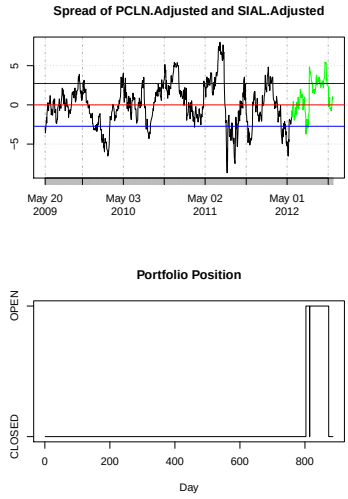

Figure 1.14: The training spread (in black) and the 6 month test spread (in green) for the cointegrated pair (13) using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - November 21 2012.

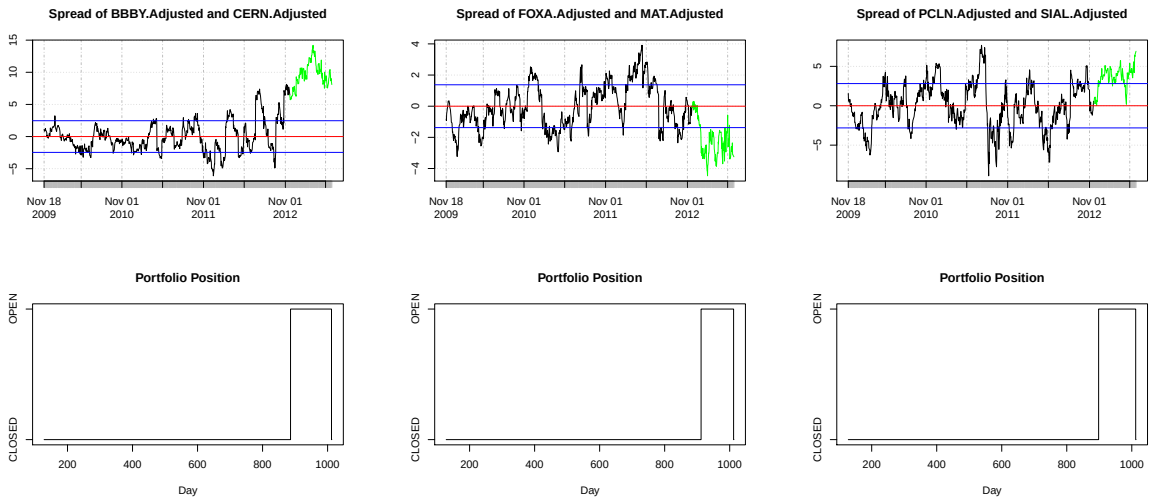

Figure 1.15: The training spread (in black) and the 6 month test spread (in green) for the cointegrated pairs 3,8,13 using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period November 21 2012 - May 28 2013.

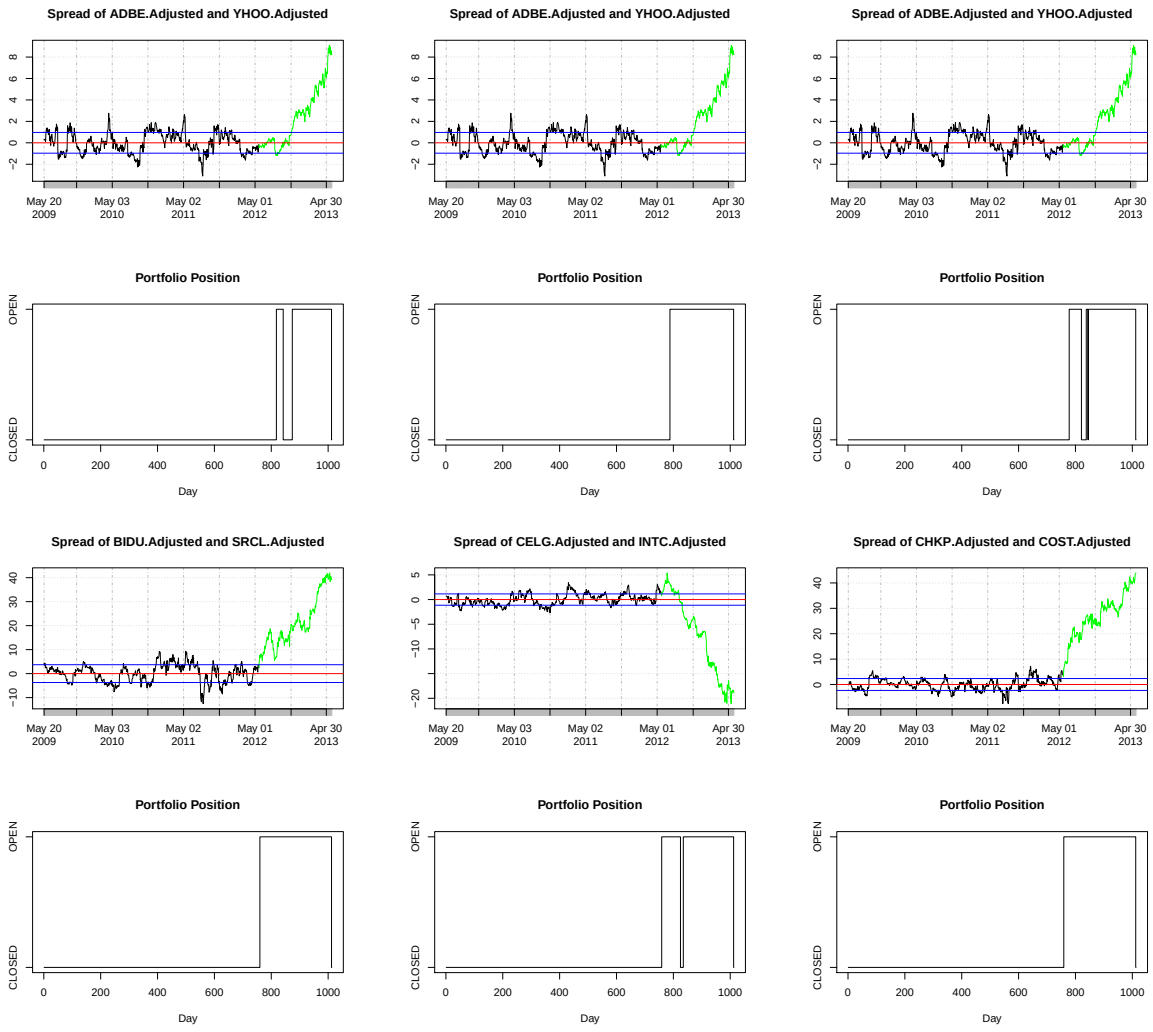

Figure 1.16: The training spread (in black) and the 12 month test spread (in green) for the first 6 cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

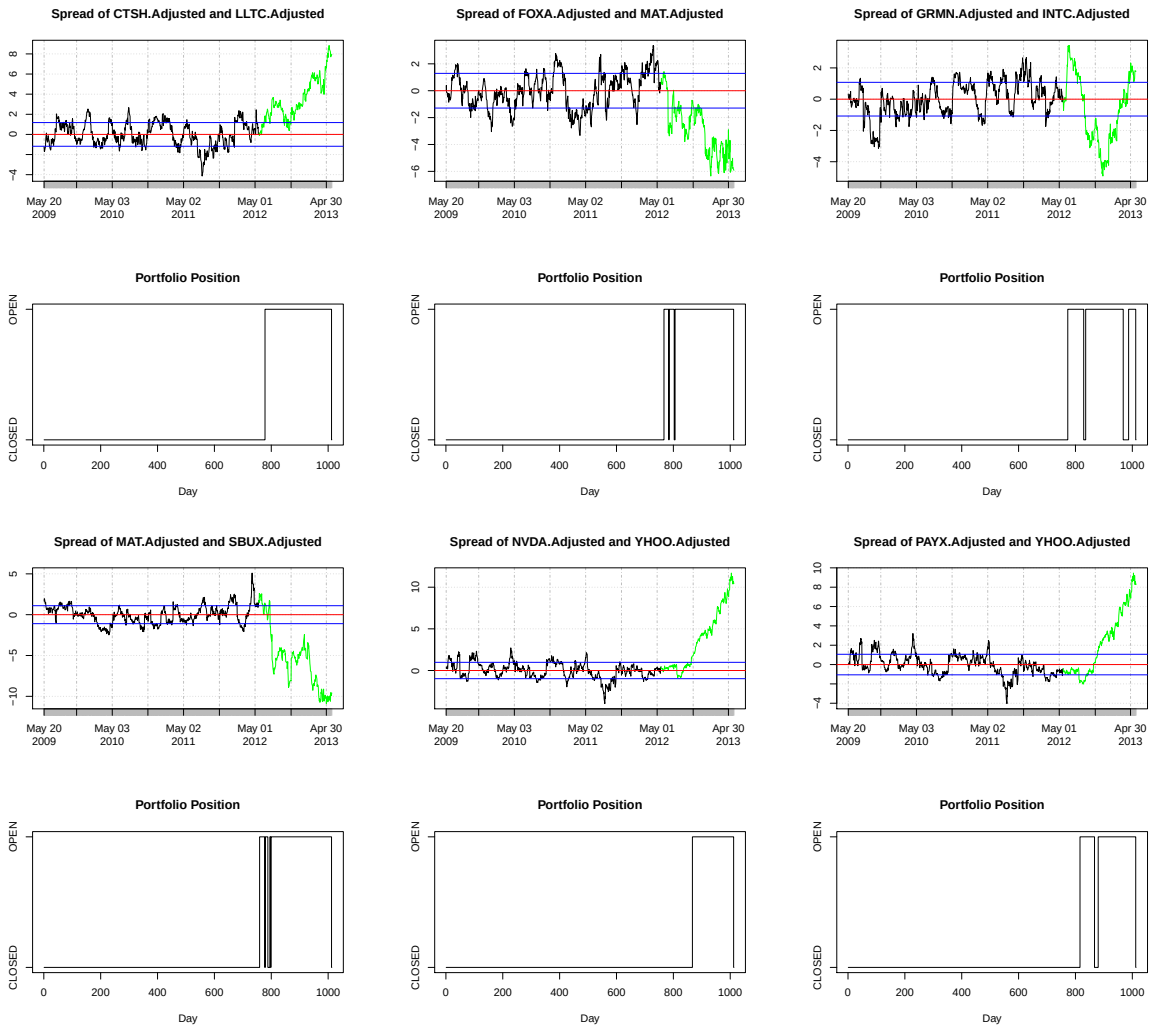

Figure 1.17: The training spread (in black) and the 12 month test spread (in green) for the 7th to 12th cointegrated pairs using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

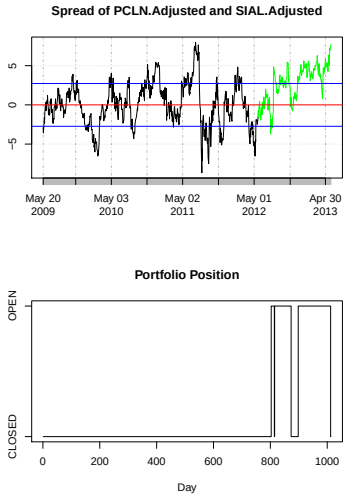

Figure 1.18: The training spread (in black) and the 12 month test spread (in green) for the 13th cointegrated pair using data from the stocks of the NASDAQ 100. The trades are done on an upper and lower bound of two standard deviations from the mean, and traded on the period May 21 2012 - May 28 2013.

Asset Pair	ADBE,YHOO	ADSK,LLTC	BBBY,CERN	BIDU,SRCL	CELG,INTC
Total Profit	-0.208	-2.32	4.408	-16.735	-3.537
Total # of Trades	2	1	3	1	2
Avg Profit per Trade	-0.104	-2.32	1.469	-16.735	-1.769
Return	-0.4%	-5.47%	7.24%	-13.77%	-10.34%
Annual Return	-0.8%	-10.93%	14.48%	-27.54%	-20.67%

Table 1.9: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 1 to 5)

Asset Pair	CHKP,COST	CTSH,LLTC	FOXA,MAT	GRMN,INTC	MAT,SBUX
Total Profit	-20.156	-0.624	7.141	-0.283	1.954
Total # of Trades	1	1	3	2	4
Avg Profit per Trade	-20.156	-0.624	2.38	-0.142	0.488
Return	-19.06%	-1.52%	16.91%	-1.01%	5.82%
Annual Return	-38.11%	-3.04%	33.83%	-2.01%	11.64%

Table 1.10: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 6 to 10)

Asset Pair	NVDA,YHOO	PAYX,YHOO	PCLN,SIAL
Total Profit	-1.815	1.048	13.511
Total # of Trades	1	2	2
Avg Profit per Trade	-1.815	0.524	6.756
Return	-7.73%	4.64%	13.35%
Annual Return	-15.46%	9.29%	26.71%

Table 1.11: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - November 21 2012 (pairs 11 to 13)

Asset Pair	BBBY,CERN	FOXA,MAT	PCLN,SIAL
Total Profit	-1.582	-1.569	-3.873
Total # of Trades	1	1	1
Avg Profit per Trade	-1.582	-1.569	-3.873
Return	-3.04%	-3.07%	-3.7%
Annual Return	-6.08%	-6.15%	-7.4%

Table 1.12: Trading results on the (out of sample) data of 6 months using training data of 3 years with a threshold of 1 standard deviation from November 21 2012 - May 28 2013 (pairs 3,8,13)

Asset Pair	ADBE,YHOO	ADSK,LLTC	BBBY,CERN	BIDU,SRCL	CELG,INTC
Total Profit	-5.941	-5.425	2.272	-34.733	-15.085
Total # of Trades	2	1	3	1	2
Avg Profit per Trade	-2.971	-5.425	0.757	-34.733	-7.542
Return	-23.46%	-12.78%	3.12%	-28.58%	-44.15%
Annual Return	-23.46%	-12.78%	3.12%	-28.58%	-44.15%

Table 1.13: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 1 to 5)

Asset Pair	CHKP,COST	CTSH,LLTC	FOXA,MAT	GRMN,INTC	MAT,SBUX
Total Profit	-40.158	-6.521	2.493	2.929	-2.452
Total # of Trades	1	1	3	3	4
Avg Profit per Trade	-40.158	-6.521	0.831	0.976	-0.613
Return	-37.97%	-15.86%	7.3%	8.79%	-5.45%
Annual Return	-37.97%	-15.86%	7.3%	8.79%	-5.45%

Table 1.14: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 6 to 10)

Asset Pair	NVDA,YHOO	PAYX,YHOO	PCLN,SIAL
Total Profit	-8.919	-5.357	9.321
Total # of Trades	1	2	3
Avg Profit per Trade	-8.919	-2.678	3.107
Return	-37.99%	-20.51%	9.34%
Annual Return	-37.99%	-20.51%	9.34%

Table 1.15: Trading results on the (out of sample) data of 12 months using training data of 3 years with a threshold of 1 standard deviation from May 21 2012 - May 28 2013 (pairs 11 to 13)

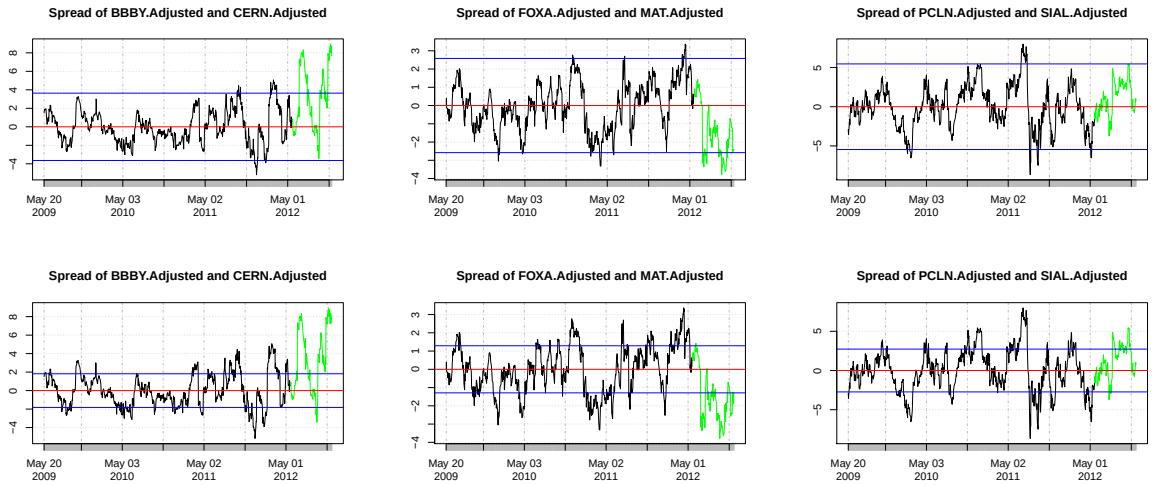

Figure 1.19: The training spread (in black) and the 6 month test spread (in green) cointegrated pairs (3,8, 13) using data from the stocks of the NASDAQ 100. The top row of spreads shows the trades with bounds of 2 standard deviations from the mean, while the second row shows the trades with bounds 1 standard deviation from the mean. These pairs are traded on the period May 21 2012 - November 21 2012, but are used mainly as a comparison for using different standard deviations on the bounds. The pairs have been selected retrospectively after the trades have happened and have been determined to remain cointegrated.

Asset Pair	BBBY,CERN		FOXA,MAT		PCLN,SIAL	
k = # of Std Dev	2	1	2	1	2	1
Total Profit	3.473	4.408	4.653	7.141	N/A	13.511
Total # of Trades	2	3	2	3	N/A	2
Avg Profit per Trade	1.736	1.469	2.326	2.38	N/A	6.756
Return	6.42%	7.24%	10.11%	16.91%	N/A	13.35%
Annual Return	12.85%	14.48%	20.22%	33.83%	N/A	26.71%

Table 1.16: A comparison of the trading results on the (out of sample) data of 6 months using training data of 3 years from May 21 2012 - November 21 2012 (pairs 3,8,13) for trading bounds of 1 and 2 standard deviations from the mean. Only the pairs that remain cointegrated after the trading period have been selected for the comparison.

Chapter 2

Wavelet Analysis of Time Series

2.1 Introduction

As we have seen, the concept of cointegration can be quite important to pairs trading as our trading strategy depends on the mean reversion and stationarity of our residual spread series. In practice, there are issues that lie even after finding suitable cointegrated pairs. For one, it is often not clear how long the trading period can be because it is not known how long the cointegration relationship exists for. This can be seen in the earlier application to real data. The data is stationary for the training data but several pairs have spreads that depart significantly from the mean and do not seem to be reverting to the training data mean.

In a related direction, the idea of global stationarity for the spread may be too restrictive to find many viable pairs for trading. This is where the idea of costationarity can conceivably make a big difference. The concept, introduced by Cardinali and Nason (2011), has been a recent attempt to adapt the concept of cointegration for locally stationary processes. Recall that a weakly stationary process has a autocovariance and mean that does not change with time. The alternative is a non-stationary process that does depend on time. The idea for locally stationary processes lies in between these two extremes: if the

function's statistical properties change very slowly over time, then for localized sections of the time series, the process will be stationary. The approach taken by Cardinali and Nason (2011) revolved around the usage of wavelets. The topics relevant to local stationary processes regarding wavelets will be discussed in the following sections.

2.2 Fourier Series and Fourier Transforms

Fourier analysis is one of the most predominant methods for the analysis of stationary processes. We start with Fourier series: any periodic, absolutely integrable function $g_p(t)$ can be written as a linear combination of sine and cosine terms with varying amplitude, phase and frequency. This can be represented in the form:

$$g_p(t) = \sum_{n=-\infty}^{\infty} c_n \left(\frac{2\pi i n t}{T_0} \right), \tag{2.1}$$

where

$$c_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t) \exp\left(\frac{-2\pi i n t}{T_0} \right) dt\,. \tag{2.2}$$

The sine and cosine functions are embedded in the complex exponential function in Equation 2.2 by Euler's formula:

$$e^{it} = \cos(t) + i \sin(t)\,. \tag{2.3}$$

The functions $w_t = e^{it}$ form an orthonormal basis. For a function $g(t)$ that is aperiodic, the idea is to represent it with Equation 2.1 and let the period become infinitely large:

$$g(t) = \lim_{T_0 \to \infty} g_p(t) \tag{2.4}$$

By defining

$$\begin{aligned} \Delta\omega &= \frac{1}{T_0}, \\ \omega_n &= \frac{n}{T_0}, \end{aligned} \tag{2.5}$$

and

$$G(\omega_n) = c_n T_0, \tag{2.6}$$

where $\Delta\omega$ is the frequency, we can rewrite Equation2.1 as:

$$g_p(t) = \sum_{n=-\infty}^{\infty} G(\omega_n) \exp(2\pi i \omega_n t) \Delta\omega \,, \tag{2.7}$$

where

$$G(\omega_n) = \int_{-T_0/2}^{T_0/2} g_p(t) \exp(-2\pi i \omega_n t)\, dt \,. \tag{2.8}$$

Now, taking the limit as T_0 approaches infinity, we get:

$$g(t) = \int_{-\infty}^{\infty} G(\omega) \exp(2\pi i \omega t)\, d\omega, \tag{2.9}$$

where

$$G(\omega) = \int_{-\infty}^{\infty} g(t) \exp(-2\pi i \omega t)\, dt, \tag{2.10}$$

which are known as the inverse Fourier transform of $G(t)$ and the Fourier transform of $g(t)$ respectively. Through this, it is possible to present any aperiodic, square integral process in terms of exponentials and to transform a function of time (t) into a function of frequency (ω).

Define the total energy over the interval $[-\pi, \pi]$, as

$$\int_{-\pi}^{\pi} g_p^2(t) dt = 2\pi \sum_{n=0}^{\infty} c_n^2. \tag{2.11}$$

This is known as *Parseval's relation*. Calculating the energy over all time for a periodic function is not relevant, as it would be infinite. However, the concept of energy per unit time, also known as *power*, is quite useful:

$$\text{Total power} = \frac{\text{Total energy over } [-\pi, \pi]}{2\pi} = \sum_{n=0}^{\infty} c_n^2. \tag{2.12}$$

For a non-periodic function $g(t)$, the analogous form for Parseval's relation is:

$$\begin{aligned} \text{Total energy over } (-\infty, \infty) &= \int_{-\infty}^{\infty} g^2(t)dt \\ &= \int_{-\infty}^{\infty} |G(\omega)|^2 \, d\omega \, . \end{aligned} \tag{2.13}$$

Here, $|G(\omega)|^2 \, d\omega$ represents the contribution to the total energy from the components in $g(t)$ whose frequencies lie between ω and $\omega + d\omega$. As such, $|G(\omega)^2|$ can be considered a density function of the energy contribution by the components in $g(t)$. The total energy of a non-periodic function that is square integrable is finite, in comparison to that of a periodic function, in which the energy over the interval $(-\infty, \infty)$ is infinite.

However, for a zero-mean stationary series X_t, there is no guarantee that we may have a Fourier series representation, as there is no reason for it to be periodic. Similarly, there is no reason for it to possible for X_t to be represented by a Fourier integral as it does not necessarily have to be absolutely integrable. By defining a new function:

$$X_{t,T} = \begin{cases} X_t, & \text{if } -T \leq t \leq T, \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

where T is some arbitrary defined chop off point for the realization of X_t. $X_{t,T}$ can now be represented by a Fourier integral, as it is aperiodic and absolutely integrable only on the finite interval $(-T, T)$. Then $|G_T(\omega)|^2 \, d\omega$ would be analogous to that in Equation 2.13, but for $G_T(\omega)$. Unfortunately, in this case, we cannot just let $T \to \infty$, as this would just be the same as trying to represent X_t with a Fourier integral. The energy that would be represented in this interval would be infinite, as in the periodic process case. The power, on the other hand:

$$\lim_{T \to \infty} \frac{|G_T(\omega)|^2}{2T}. \tag{2.15}$$

may be finite.

However, this is just the contribution to the power for one realization. Thus we define

the *power spectral density function* of X_t, or the *spectrum* of X_t as:

$$S(\omega) = \lim_{T\to\infty}\left[E\left\{\frac{|G_T(\omega)|^2}{2T}\right\}\right], \tag{2.16}$$

which is just the average over all realizations of the contribution to the total power from the components in X_t with frequencies between ω and $\omega + d\omega$.

It can be shown that the Fourier transform of the autocovariance function is exactly this: the spectrum of X_t. Then, $S(\omega)\, d\omega$ is again just the contribution to the total variance of X_t for frequencies in the range $(\omega, \omega + d\omega)$.

Returning to the representation for the zero-mean stationary stochastic process, X_t, we reiterate that a more general Fourier expansion is needed than a straightforward Fourier series or a Fourier integral. Priestley (1983) showed that it can be written in the form:

$$X_t = \int_{-\pi}^{\pi} e^{i\omega t}\, d\xi(\omega) = \int_{-\pi}^{\pi} e^{i\omega t}\, |d\xi(\omega)| e^{i\arg\{d\xi(\omega)\}}\,. \tag{2.17}$$

This is known as the *spectral representation theorem*, where $d\xi(\omega)$ is a process known as an *orthonormal increments process*, $\arg\{d\xi(\omega)\}$ represents random phases, and $|d\xi(\omega)|$ represents random amplitudes of the process. The process $\xi(\omega)$ has the following properties for all $|\omega| \leq \pi$.:

Property 1. $E\{d\xi(\omega)\} = 0$

Property 2. $E\{|d\xi(\omega)|^2\} = dS^I(\omega)$, where $S^I(\omega)$ is the integrated spectrum of $\{X(t)\}$.

Property 3. For any two distinct frequencies ω and ω',

$$\mathrm{cov}\{d\xi(\omega), d\xi(\omega')\} = E\{d\xi(\omega) d\xi(\omega')\} = 0.$$

Then, the autocovariance γ_s can be written as:

$$\begin{aligned}
\gamma_s = E\{X_t X_{t+s}\} &= E\{X_t^* X_{t+s}\} \\
&= E\left\{\int_{-\pi}^{\pi} e^{-i\omega' t}\, d\xi^*(\omega') \int_{-\pi}^{\pi} e^{i\omega(t+s)}\, d\xi(\omega)\right\} \\
&= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} e^{it(\omega-\omega')} e^{is\omega}\, E\left\{d\xi^*(\omega')\, d\xi(\omega)\right\}.
\end{aligned} \tag{2.18}$$

By Property 3 of the orthogonal increments process, $E\Big\{ d\xi^*(\omega')\, d\xi(\omega) \Big\} = dS^I(\omega)$ iff $\omega' = \omega$. Then,

$$\gamma_s = \int_{-\pi}^{\pi} e^{is\omega}\, dS^I(\omega)\,. \tag{2.19}$$

If the the integrated spectrum $S^I(\omega)$ is differentiable everywhere, we have:

$$d\, S^I(\omega) = S(\omega) d\omega, \tag{2.20}$$

and so:

$$\gamma_s = \int_{-\pi}^{\pi} S(\omega) e^{is\omega}\, d\omega. \tag{2.21}$$

Thus the autocovariance function of X_t, γ_s, is the inverse Fourier transform of the spectrum of X_t, $S(\omega)$. If the spectrum is square integrable, then $S(\omega)$ is the Fourier transform of γ_s, and we have a Fourier transform pair.

One major flaw of using Fourier transforms for analysis is that Fourier coefficients are not localized in time. For example, a discontinuity or a change in $g(t)$ will cause all of the coefficients to be affected. Wavelets are however, localized in time, and as such are very suited to the problem of finding and coping with concepts such as local stationarity.

2.3 Wavelets

In Fourier analysis, a function can be represented as a linear combination of coefficients and complex exponential basis functions. Wavelets analysis takes a very similar path, but the basis functions must decay to 0 rapidly. With a mother wavelet ψ, one can then compute the basis generated by dilation and translation:

$$\psi_{j,k}(x) = 2^{j/2}\, \psi(2^j x - k)\,, \tag{2.22}$$

for $j, k \in \mathbb{Z}$.

For an orthogonal basis, this requires

$$< \psi_{j,k}, \psi_{j',k'} >= \int_{-\infty}^{\infty} \psi_{j,k}(x)\, \psi_{j',k'}(x)\, dx \;=\; \delta_{j,j'}\, \delta_{k,k'}\,, \tag{2.23}$$

where $\delta_{a,b}$ is the Kronecker delta. That is,

$$\delta_{a,b} = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.24}$$

With an orthogonal basis, a function $f(x)$ can be written as a linear combination of coefficients of these wavelets and coefficients as follows:

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k}\, \psi_{j,k}(x)\,, \tag{2.25}$$

where

$$d_{j,k} = \int_{-\infty}^{\infty} f(x)\, \psi_{j,k}(x) dx \; = < f, \psi_{j,k} > . \tag{2.26}$$

These $d_{j,k}$ are known as the wavelet coefficients of f. We will return to this approximation of $f(x)$. What do the j in the coefficients actually represent? This is the concept of 'scale' in the breakdown of our function. With wavelets, we will always be working with dyadic data. That is, data that is of the length 2^J. At the finest scale, the wavelet coefficients capture the most detail in the data. As we move to 'lower' or coarser scales, the detail that is captured by the various coefficents becomes more spread out, giving a rougher estimate of the function. As it can be seen in Equation 2.25, the function can be estimated using only the coefficients $d_{j,k}$ and the wavelets $\psi_{j,k}$. However, it is useful to introduce new coefficients and a new wavelet. Define the *Haar father wavelet* as:

$$\phi(x) = \begin{cases} 1, & \text{if } x \in [0, 1], \\ 0, & \text{otherwise.} \end{cases} \tag{2.27}$$

and the finest-level father wavelet coefficients to be

$$c_{J,k} = \int_0^1 f(x)\, 2^{J/2}\, \phi(2^J x - k)\, dx, \tag{2.28}$$

for $k = 0, ...2^J - 1$. By Equation 2.22, we can write the father wavelets with the same notation:

$$c_{J,k} = \int_0^1 f(x)\, \phi_{J,k}(x)\, dx. \tag{2.29}$$

This is just the integral of $f(x)$ over the interval $[2^{-J}k, 2^{-J}(k+1)]$, which is proportional to the local average of $f(x)$ over that interval. Through these coefficients and the dilated and translated father wavelets, $f(x)$ can be approximated by the following:

$$f_J(x) = \sum_{k=0}^{2^J-1} c_{J,k}\, \phi_{J,k}(x). \tag{2.30}$$

However, note that this is approximation that changes by scale. The level of detail of the approximation decreases with J. It can be seen that the father wavelet coefficients can be calculated from the integral in Equation 2.29. In practice, this is not needed as there is a way of deriving coarser scale coefficients from the finer scale coefficients. It is important to note for the Haar father wavelet,

$$\phi(x) = \phi(2x) + \phi(2x-1)\,. \tag{2.31}$$

This relationship can be seen in Figure 2.1.

With this, we can write the father wavelet coefficient $c_{j-1,k}$ as follows:

$$\begin{aligned}
c_{j-1,k} &= \int_{2^{-(j-1)}k}^{2^{-(j-1)}(k+1)} f(x)\, \phi_{j-1,k}(x)\, dx \\
&= 2^{-1/2} \int_{2^{-j}2k}^{2^{-j}(2k+2)} f(x)\, 2^{j/2} \phi(2^{j-1}x - k)\, dx \\
&= 2^{-1/2} \left\{ \int_{2^{-j}(2k}^{2^{-j}(2k+1)} f(x)\, 2^{j/2} \phi(2^j x - 2k)\, dx + \int_{2^{-j}(2k+1}^{2^{-j}(2k+2)} f(x)\, 2^{j/2} \phi(2^j x - 2k - 1))\, dx \right\} \\
&= 2^{-1/2} \left\{ \int_{2^{-j}(2k}^{2^{-j}(2k+1)} f(x)\, \phi_{j,2k}(x)\, dx + \int_{2^{-j}(2k+1}^{2^{-j}(2k+2)} f(x)\, \phi_{j,2k+1}(x)\, dx \right\} \\
&= \frac{1}{\sqrt{2}} (c_{j,2k} + c_{j,2k+1})\,.
\end{aligned} \tag{2.32}$$

As such, we only require the finest level father wavelet coefficients to obtain all the father wavelet coefficients. With discrete dyadic data, the finest level father wavelet coefficients

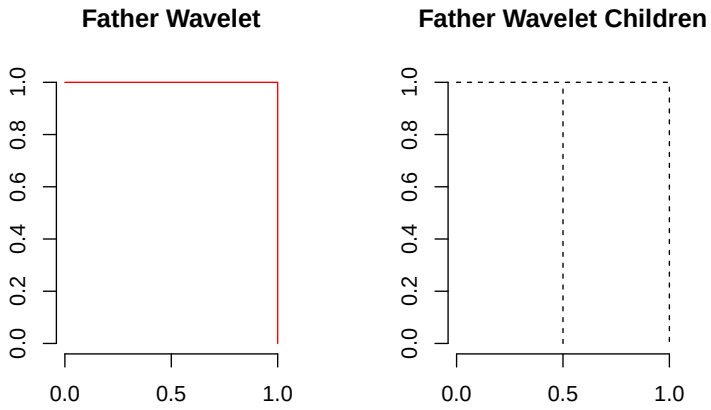

Figure 2.1: A father wavelet on the left plot. The right plot shows that the relationship described in Equation 2.31: the Haar father wavelet can be written as a sum of dilated and translated father wavelets.

are just the data points themselves. We have now obtained the algorithm to determine the father wavelet coefficients for our data. However, the function cannot be approximated well by linear combinations of father wavelets and father wavelet coefficients only. The difference between levels of approximations is the detail that was initially discussed with the mother wavelets and the mother wavelet coefficients $d_{j,k}$. Considering the two coarsest level approximations $f_0(x)$ and $f_1(x)$, we have:

$$f_0(x) = c_{0,0}\,\phi_{0,0}(x) \tag{2.33}$$

and

$$f_0(x) = c_{1,0}\,\phi_{1,0}(x) + c_{1,1}\,\phi_{1,1}(x). \tag{2.34}$$

The difference between the two is:

$$
\begin{aligned}
f_1(x) - f_0(x) &= c_{1,0}\,\phi_{1,0}(x) + c_{1,1}\,\phi_{1,1}(x) - c_{0,0}\,\phi_{1,0}(x) \\
&= c_{1,0}\,2^{1/2}\phi(2x) + c_{1,1}\,2^{1/2}\phi(2x-1) - c_{0,0}\,\phi(x) \\
&= c_{1,0}\,2^{1/2}\phi(2x) + c_{1,1}\,2^{1/2}\phi(2x-1) - 2^{-1/2}(c_{1,0}+c_{1,1})\,\phi(x) \\
&= c_{1,0}\,2^{1/2}\phi(2x) + c_{1,1}\,2^{1/2}\phi(2x-1) - 2^{-1/2}(c_{1,0}+c_{1,1})\,[\phi(2x)+\phi(2x-1)] \\
&= 2^{-1/2}\Big[(2c_{1,0}-c_{1,1}-c_{1,0})\phi(2x) + (2c_{1,1}-c_{1,0}-c_{1,1})\phi(2x-1)\Big] \\
&= 2^{-1/2}\Big[(c_{1,0}-c_{1,1})\phi(2x) - (c_{1,0}-c_{1,1})\phi(2x-1)\Big] \\
&= 2^{-1/2}(c_{1,0}-c_{1,1})\Big[\phi(2x) - \phi(2x-1)\Big]\,.
\end{aligned}
\tag{2.35}
$$

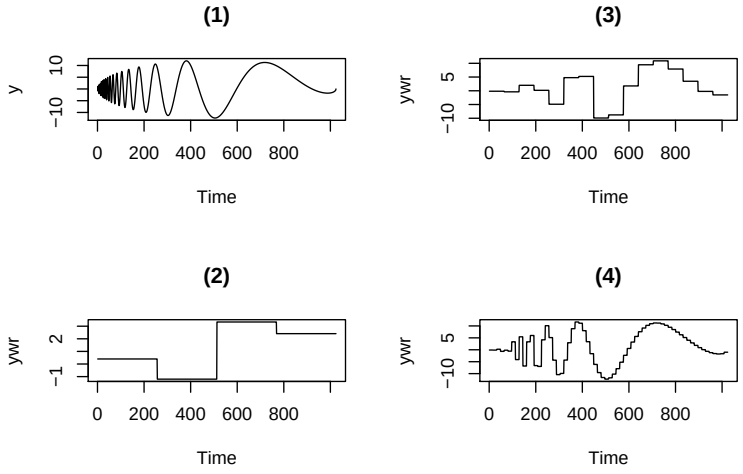

Figure 2.2: The Doppler function in the top left plot (1). The other plots (2),(3), and (4) are projections of the Doppler function into father wavelet spaces $J = 2, 4$ and 6. Notice that each plot has the doppler function being projected onto 2^J different coefficients (4, 16, 64).

Defining the Haar mother wavelet as:

$$\begin{aligned}\psi(x) &= \phi(2x) - \phi(2x-1) \\ &= \begin{cases} 1, & \text{if } x \in [0, \frac{1}{2}), \\ -1, & \text{if } x \in [\frac{1}{2}, 1), \\ 0, & \text{otherwise.} \end{cases}\end{aligned} \tag{2.36}$$

and the Haar mother wavelet coefficients $d_{j,k}$ as

$$d_{j,k} = 2^{-1/2}(c_{j+1,2k} - c_{j+1,2k+1}), \tag{2.37}$$

we have

$$\begin{aligned} f_1(x) - f_0(x) &= d_{0,0}\,\psi(x) \\ f_1(x) &= f_0(x) + d_{0,0}\,\psi(x) \\ &= c_{0,0}\,\phi(x) + d_{0,0}\,\psi(x). \end{aligned} \tag{2.38}$$

With this, the approximation for a higher scale approximation to f can be presented as the

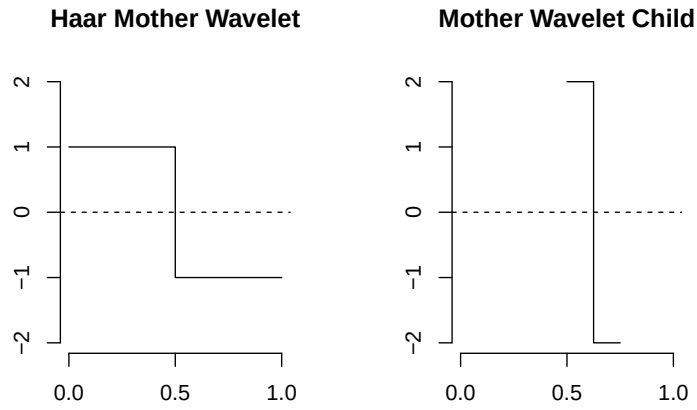

Figure 2.3: A Haar mother wavelet (left) and a mother wavelet child $\psi_{2,2}$ (right)

next coarser scale approximation to f and the detail that is obtained from the difference

between scales. For any scale, we can write:

$$\begin{aligned} f_{j+1}(x) &= f_j(x) + \sum_{k=0}^{2^j-1} d_{j,k}\,\psi_{j,k}(x) \\ &= \sum_{k=0}^{2^j-1} c_{j,k}\,\phi_{j,k}(x) + \sum_{k=0}^{2^j-1} d_{j,k}\,\psi_{j,k}(x). \end{aligned} \tag{2.39}$$

Note that these are the j-th scale approximations for f. If we wanted the finest scale approximation for f, Equation 2.39 can be telescoped to arrive at a final approximation:

$$f_J(x) = c_{0,0}(x)\phi(x) + \sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1} d_{j,k}\,\psi_{j,k}(x)\,. \tag{2.40}$$

This can be seen as a smooth, averaging approximation from the coarsest father wavelet coefficient, and all the detail for the function coming at different scales from the mother wavelet coefficients. This, in its entirety, is the basis behind the discrete wavelet transform. From the data, we can find our finest level mother wavelet coefficients with simple differences from the finest level father wavelet coefficients. Again, these finest level father wavelet coefficients are just the data points for discrete dyadic data. Then the next finest father wavelet coefficients can be formed from the data as well. From there, it is just a matter of applying the same method on the father wavelet coefficients until all the wavelet coefficients are found.

It is important to note here that we have been only discussing the formulation for Haar wavelets, the simplest mother and father wavelets possible. Daubechies (1988) developed families of orthogonal wavelets that were much smoother than the Haar wavelets, and yet still compactly supported. Other wavelets such as the Shannon wavelet and Meyer wavelets also exist and are mentioned in detail in Daubechies (1992) and a summary exists in Nason (2008). These other wavelets do not have a structure as simple as the Haar, and as such, their coefficients are also more complicated. This begs for a generalization to the formulas we have derived so far. Recall that for the Haar father wavelets, they can be written as:

$$\phi(x) = \phi(2x) + \phi(2x-1)\,. \tag{2.41}$$

This can be generalized for other wavelets as:

$$\phi(x) = \sum_{n\in\mathbb{Z}} h_n \, \phi_{1,n}(x) \, . \tag{2.42}$$

For the Haar case, $h_1 = h_0 = \frac{1}{\sqrt{2}}$, as:

$$\begin{aligned} \phi(x) &= h_0 \, \phi_{1,0}(x) + h_1 \, \phi_{1,1}(x) \\ &= h_0 \, 2^{1/2} \, \phi(2x) + h_1 \, 2^{1/2} \phi(2x-1) \, . \end{aligned} \tag{2.43}$$

As the mother wavelet can also be written as a linear combination of coefficients and father wavelets, we have a similar generalization here:

$$\psi_(x) = \sum_{n\in\mathbb{Z}} g_n \, \phi_{1,n}(x) \, , \tag{2.44}$$

where

$$g_n = (-1)^{n-1} \, h_{1-n}. \tag{2.45}$$

For the Haar mother wavelet, $g_0 = \frac{-1}{\sqrt{2}}$ and $g_1 = \frac{1}{\sqrt{2}}$.

This representation becomes much more useful for wavelets more complicated than the Haar wavelet family, but we will only focus on the Haar family for the sake of simplicity.

2.4 Non-decimated Wavelet Transform

Notice that in Equations 2.32 and 2.37 that we have the coarser mother and father coefficients coming from a sum of the finer coefficients, but mainly that they are coming from a sum or difference of $k = 2k$ or $k = 2k + 1$. This is known as *dyadic decimation* by a factor of 2. By obtaining the coefficients in this manner, we obtain an orthogonal transformation. However, we also lose some information between data points. For example, for a data set of 4 points $\{y_1, y_2, y_3, y_4\}$, $c_{1,0} = 2^{-1/2} \, (y_1 + y_2)$ and $c_{1,1} = 2^{-1/2} \, (y_3 + y_4)$. The coefficients $d_{1,0}$ and $d_{1,1}$ are also obtained using differences between the same data values. We do not

have any information on the sums and differences between y_2 and y_3 however. By shifting the decimation 1 data point, we can obtain this information. This is the idea behind the non-decimated wavelet transform. Both versions of the decimated transform are computed and then combined together. These two versions can be used separately, but it is useful to put them together in one time-ordered package of coefficients in the analysis of time series. We can refer to these decimated shifts as the even and odd decimations. Returning to our data set $\{y_1, y_2, y_3, y_4\}$, the finest scale father wavelet coefficients would be the two sets: one as our original decimated version, and the other as $c_{1,0} = 2^{-1/2}(y_2 + y_3)$, $c_{1,1} = 2^{-1/2}(y_1 + y_4)$, $d_{1,0} = 2^{-1/2}(y_2 - y_3)$, $d_{1,1} = 2^{-1/2}(y_1 - y_4)$. The even and odd decimations are applied to the coefficients at each and every scale $J - j$, resulting in 2^j sets of coefficients of length $2^{-j}n$ for $j = 1, 2, ...J$. This results in $2^{-j}n\,2^j = n$ wavelet coefficients at each scale, and with J scales, there are a total of Jn coefficients produced by the non-decimated wavelet transform. With regards to locally stationary processes, a different notation for the discrete wavelets is taken by Nason et al. (2000) :

$$\psi_{-1,n} = \sum_k g_{n-2k}\delta_{0,k} = g_n, \text{ for } n = 0, ...N_{-1} - 1, \tag{2.46}$$

$$\psi_{j-1,n} = \sum_k h_{n-2k}\psi_{j,k}, \text{ for } n = 0, ...N_{j-1} - 1, \tag{2.47}$$

$$N_j = (2^j - 1)(N_h - 1) + 1, \tag{2.48}$$

where $\delta_{0,k}$ is the Kronecker delta, and N_h is the number of non-zero elements of $\{h_k\}$.

2.5 Locally Stationary Processes

Recall that with the spectral representation theorem, we can write a stationary process X_t as:

$$X_t = \int_{-\pi}^{\pi} e^{i\omega t}\, d\xi(\omega) = \int_{-\pi}^{\pi} e^{i\omega t}\, |d\xi(\omega)| e^{i \arg\{d\xi(\omega)\}}. \tag{2.49}$$

In this representation, $|d\xi(\omega)|$ represents the random amplitudes of X_t, but it does not depend on time. Nason et al. (2000) introduced the *locally stationary wavelet (LSW)*

process $\{X_{t,T}\}_{t=0,1...T-1}$, for $T = 2^J - 1$:

$$X_{t,T} = \sum_{j=-J}^{-1} \sum_{k} w_{j,k,T}\, \psi_{j,k}(t)\, \xi_{j,k}, \tag{2.50}$$

where $\{\xi_{j,k}\}$ is an orthonormal increment sequence, and where $\{\psi_{j,k}(t)\}$ is a discrete non-decimated family of wavelets for $j = -1, -2, ... - J, k = 0, 1...T - 1$, and $\{w_{j,k,T}\}$ is a set of amplitudes. The parallel with the spectral representation of a stochastic stationary process should be very apparent now; we are just replacing the complex exponential basis functions with the mother wavelets. There are random amplitudes here as well, in the form of $w_{j,k,T}\}$, and the orthonormal increment sequence is the same. The difference is that this is a discrete approximation, as we are using the DWT to form our wavelet coefficients.

There are an extra three conditions set by Nason et al. (2000) on the representation in Equation 2.50 for them to be LSW processes:

1. $E\{\xi_{j,k}\} = 0\,.$
2. $\mathrm{Cov}\{\xi_{j,k}, \xi_{l,m}\} = \delta_{j,l}\, \delta_{k,m}\,.$
3. $\sup_k |w_{j,k;T} - W_j(\frac{k}{T})| \leq \frac{C_j}{T}$, where $\{C_j\}$ is a set of constants that have a finite sum: $\sum_{j=-\infty}^{-1} C_j \leq \infty\,.$

The first two conditions are analogous to Properties 1 and 3 that were introduced for the orthonormal increments previously. The first condition also means that LSW processes have zero-mean; and any data that we wish to model as a LSW process will have to be de-trended before doing so. This problem will be addressed later. The second property states that the orthnormal increments must be uncorrelated, as before. The third property controls the rate the $w_{j,k;T}$ are allowed to change over time, by limiting the difference between it and a function $W_j(z)$, for $z \in (0, 1)$. This is needed for estimation purposes, as the slower $w_{j,k;T}$ changes, the more data can be used for the estimation of $W_j(z)$.

The concept for LSW processes that is analogous to the spectrum for Fourier representations of stationary processes is the *evolutionary wavelet spectrum (EWS)*:

$$S_j(z) = |W_j(z)|^2 \tag{2.51}$$

for $j = -1, -2, ... - J$, and $z \in (0, 1)$. The EWS is, like the spectrum, a way to determine how the variance is distributed. Instead of a measure of how it is distributed across frequencies, it is a measure of scale (j) and location (z). By using the rescaled time $z = \frac{k}{T}$, this allows increasing amounts of data to contribute to the estimation of the local structure of $W_j(z)$.

Figure 2.4 shows the spectrum $S_j(z)$:

$$\Psi_H(u) = \begin{cases} \cos^2(4\pi z), & \text{for } j = -6, z \in (0, 1), \\ 1, & \text{for } j = -3, z \in (300/1024, 400/1024), \\ 1, & \text{for } j = -1, z \in (800/1024, 900/1024), \end{cases} \tag{2.52}$$

At the level $J = -6$, we have the coefficients that form the coarse level structure of the right plot in Figure 2.4. From $z = 300$ to $z = 400$, there is a burst that increases the variance of the function at a finer level $J = -3$, and again from $z = 800$ to $z = 900$ at an the finest level $J = -1$. We can see that these bursts add "noise" as we approach the finest levels to the function.

The introduction of rescaled time also allows us to understand the reasoning behind the switch of notation in equations 2.46 and 2.47. In this notation, the data lie on scale 0, and starting with scale -1 the wavelet coefficients start from the finest and gradually become coarser as the scale moves toward $-J$. By using this numbering scheme, the support of the wavelets on the finest scale is fixed and constant with respect to the length of the observed time series, T. The addition of extra data means that coarser wavelets can be included, which means that $-J$ should approach to $-\infty$ as T gets larger.

Nason et al. (2000) also define the *autocorrelation wavelets*, $\Psi_j(s)$, of the discrete

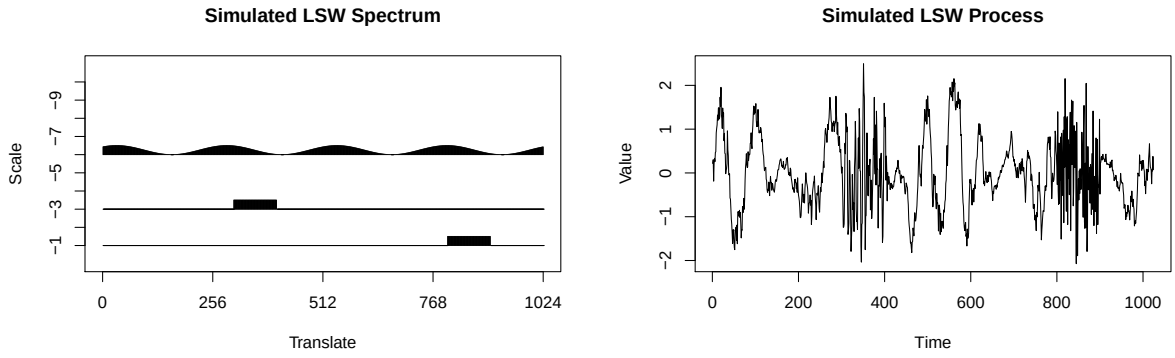

Figure 2.4: A spectrum $S_j(z)$ from Equation 2.52 on the left. The resulting function that is simulated from the spectrum is plotted on the right.

wavelets as:

$$\Psi_j(s) = \sum_k \psi_{j,k}(0)\,\psi_{j,k}(s). \tag{2.53}$$

for all $j < 0$ and $s \in \mathbb{Z}$.

The Haar continuous autocorrelation wavelets are:

$$\Psi_H(u) = \int_{-\infty}^{\infty} \psi_H(x)\,\psi_H(x-u)\,dx = \begin{cases} 1 - 3\,|u|, & \text{for } |u| \in [0, 1/2], \\ |u| - 1, & \text{for } |u| \in (1/2, 1], \end{cases} \tag{2.54}$$

where $\psi_H(x)$ is the continuous Haar mother wavelet in equation 2.36. With the formula $\Psi_j(s) = \Psi_H(2^j|s|)$, we can obtain the discrete autocorrelation wavelets from the continuous version. This can be extended to other families of wavelets such as the Daubechies' compactly supported wavelets, but unlike the Haar, they do not have a simple closed form solution.

2.5.1 Estimation of the EWS

With the non-decimated wavelet transform, we can obtain the non-decimated wavelet coefficients $d_{j,k;T}$ using the realizations of the process $x_1, x_2, ...x_T$:

$$d_{j,k;T} = \sum_{t=1}^{T} x_t \, \psi_{j,k}(t). \tag{2.55}$$

The *raw wavelet periodogram* is constructed using these wavelet coefficients through:

$$I^j_{k,T} = |d_{j,k;T}|^2. \tag{2.56}$$

This is a result that comes from the fact that $X_{t,T}$ can be represented as a linear combination of the wavelets and the coefficients $w_{j,k;T}\xi_{j,k}$, or the inverse wavelet transform of these coefficients. As such, taking the wavelet transform of the realizations $\{x_t\}$ will result in an estimate for $w_{j,k;T}$. Using this, and the fact that $w_{j,k;T}$ is close to $W_j(z)$ by the third condition of LSW processes, we can take the square of the wavelet coefficients to obtain an estimate of $S_j(z)$. However, the estimate is biased. Nason et al. (2000) also demonstrate that the vector $\mathbf{I}(z)$ of raw wavelet periodograms for $j = -1.. - J$ have an expectation of

$$E\{\mathbf{I}(z)\} = A\mathbf{S}(z) + O(T^{-1}), \tag{2.57}$$

for all $z \in (0, 1)$, $\mathbf{S}(z) = \{S_j(z)\}_{j=-1,...-J}$, where

$$\mathbf{I}(z) \; = \left\{ I^j_{\lfloor zT \rfloor, T} \right\}_{j=-1,...,-J}, \tag{2.58}$$

and A is the inner product matrix of the autocorrelation wavelets:

$$A_{jl} =< \Psi_j, \Psi_l >= \sum_s \Psi_j(s) \, \Psi_l(s). \tag{2.59}$$

Then, the *corrected wavelet periodogram* can be constructed by:

$$\mathbf{L}(z) = A^{-1}\mathbf{I}(z), \tag{2.60}$$

which has an expected value of

$$E\{\mathbf{L}(z)\} = \mathbf{S}(z) + O(T^{-1}). \tag{2.61}$$

In addition to this, the variance of the raw wavelet periodogram is also biased:

$$\text{var}\{\mathbf{I}(z)\} = 2\left\{ \sum_{l} A_{jl} S_l(z) \right\}^2 + O(2^{-j}/T). \tag{2.62}$$

Fortunately, there is a straightforward solution for this, and that is to smooth the corrected wavelet periodogram in 2.61. However, Nason et al. (2000) suggest that it is often easier to smooth the raw wavelet periodogram before correcting it, as the distributional properties of $\mathbf{I}(z)$ are easier to examine compared to $\mathbf{L}(z)$. Hence the smoothing parameters are easier to find for the raw wavelet periodogram.

Smoothing is a vast topic that we will not discuss in depth here, but the general idea of smoothing begins with the assumption that our estimates of the wavelet coefficients are part noise, part signal. For large coefficients, they are assumed to be representative of the true signal and noise, but for small coefficents, they are assumed to only be contributions from noise. Thus, by removing all the wavelet coefficients below a designated threshold, the noise can be effectively removed from our estimates. For reference, Nason (2008) covers this topic in much more detail.

2.6 Costationarity

With the understanding of the estimation of the EWS in the last chapter, we can now direct our attention to the concept of costationarity. Cardinali and Nason (2011) derived this concept by combining the concept of cointegration with locally stationary wavelet processes. Recall that two processes X_t and Y_t that are integrated of order 1 are cointegrated if there is a linear combination of the two that is stationary. That is, if we can form Z_t:

$$Z_t = \alpha X_t + \beta Y_t \,, \tag{2.63}$$

such that Z_t is stationary, then X_t and Y_t are cointegrated. This notion is extended by allowing α and β to vary with time. This is a less restrictive model as we wish to find

$$Z_t = \alpha_t X_t + \beta_t Y_t, \tag{2.64}$$

where α_t and β_t are complexity constrained sequences with constraint C, and X_t and Y_t are locally stationary processes. Cardinali and Nason (2011) mention that locally stationary processes are not limited to just the *locally stationary wavelet (LSW) processes* ; the ones defined by Dahlhaus (1997), the locally stationary Fourier processes are also applicable in their costationarity framework. However, we will focus purely on the LSW processes here. The constraint C is needed is because without it, the α_t and β_t may then be set to follow the data perfectly. These solutions would not be useful under an out-of-sample test.

We believe a piecewise-constant function with C being a constraint on the number of breaks that are allowed is suitable for the context of pairs trading. This is because we hope that our pairs are, in informal terms, "cointegrated locally". We hope to find that the pairs of stocks are cointegrated, but we admit the possibility of a shift in the cointegration coefficients as time passes, and we hope to identify this with costationarity.

As LSW processes are required to be zero mean processes, we only have to worrry about whether or not the covariance varies with time. This can be done by applying the covariance operator to Z_t. However, in practice, this is not feasible as it is too computationally complicated. Cardinali and Nason (2011) turn to the spectrum instead. In the case of LSW processes, we use the metric of the EWS. If this can be found to be a constant measure with respect to time for a given set of vectors (α_t, β_t), then X_t and Y_t are recognized to be costationary. The solutions obtained by finding (α_t, β_t) are not necessarily unique. The algorithm that is used in Cardinali and Nason (2011) finds many costationary solutions and then determines at the end which differ the most from each other.

The algorithm for finding costationary solutions is computed in the following steps using realizations $\{X_t, Y_t\}$ for $t = 1...T$:

1. Randomly compute input vectors (α_t, β_t) for $t = 1...T$.

2. Form the combination $Z_t = \alpha_t X_t + \beta_t Y_t$.

3. Compute the spectral estimate (the EWS) $\hat{p}_Z(z, j) = S_j(z)$ for $\{Z_t\}$, where, recall that $z = t/T$ for the notion of rescaled time.

4. Compute the constancy of the spectral estimate using the test statistic $\tau(\hat{p}_Z)$. The constancy is tested in a hypothesis that is described below.

Regarding the first step of the algorithm, we first numerically optimize estimates of the test statistic $\tau(\hat{p}_Z)$ over the vectors (α_t, β_t). With the numerically optimized estimates (α_t^*, β_t^*), a statistical test of stationarity is then applied to $Z_t = \alpha_t^* X_t + \beta_t^* Y_t$ through the test statistic $\tau(\hat{p}_Z)$.

Here we are testing the null hypothesis of:

$$H_0 : S_j(z) \text{ is a constant function of } z \in (0, 1) \text{ for all } j \tag{2.65}$$

versus the alternative:

$$H_A : S_j(z) \text{ is not constant for some } j. \tag{2.66}$$

The test statistic that is being used in Cardinali and Nason (2011) is the following:

$$\tau_p = J^{-1} \sum_{j=1}^{J} \int_0^1 \{S_j(z) - \bar{S}_j\}^2 \, dz, \tag{2.67}$$

where $\bar{S}_j = \int_0^1 S_j(z) \, dz$. If the spectrum does not depend on time, then this statistic τ_p should equal zero for all j.

The test is carried out using a parametric bootstrap based on the assumption of Gaussianity of the innovations in the LSW processes. Then the bootstrap test for stationarity is as follows:

1. Evaluate τ_p on the data set; this is referred to as $\tau_p^{(1)}$.

2. From the sample, compute $\bar{p}(j)$, the spectral estimate that assumes the data are stationary.

3. Repeat for $i = 2$ to B ($[B-1]$ number of repetitions):

   (a) Simulate Z_t from the stationary model using the squared amplitudes from $\bar{p}(j)$ using Gaussian innovations.

   (b) Compute the same test statistic as beforeon the simulated data. This test statistic will be referred to as $\tau_p^(i)$.

4. The p-value of the entire test will be given by $p = \{\text{Number of } \tau_p^{(i)} > \tau_p^{(1)}\}/B$.

As per the usual hypothesis tests, if p is very small, we will reject the null hypothesis that the spectrum of Z_t is constant, and hence X_t and Y_t are not costationary for the given numerically minimized α_t and β_t.

The estimator for the EWS Cardinali and Nason (2011) use and prove to be consistent is the time average of the corrected wavelet periodogram:

$$\frac{1}{T}\sum_{k=1}^{T}\mathbf{L}_k, \tag{2.68}$$

where $\mathbf{L}_k$ is the same as in 2.60.

2.7 Pairs Trading based on Costationarity on Stock Data

In this chapter we apply the concept of costationarity to stock data. However, first we must address an issue that prevents us from using stock prices in our pairs trading strategy.

As mentioned before, because the LSW processes require our data to be zero-mean, we cannot use stock prices anymore in the estimation of the evolutionary wavelet spectrum. There have been examples of log returns being used in the context of pairs trading:

Chen et al. (2014) trade using a strategy of modelling the volatility through a three-regime threshold nonlinear GARCH model. However, note that they trade using thresholds computed in their model based on the log returns, and not simply based on the historical mean +/- k standard deviations. A hedge ratio is not considered in their model, as they only short and long each respective stock when they meet the thresholds. Because costationarity does not allow us to find thresholds for the trades as a function of the log returns, we cannot emulate Chen et al. (2014)'s strategy entirely.

Let P_t^A represent the first stock price in our pairs trade and P_t^B represent the second stock price. Let X_t and Y_t represent the log returns of the stocks respectively, that is: $X_t = \log\left(\frac{P_t^A}{P_{t-1}^A}\right)$ and $Y_t = \log\left(\frac{P_t^B}{P_{t-1}^B}\right)$.

Through costationarity we are able to find α_t and β_t such that our linear combination $Z_t = \alpha_t X_t + \beta_t Y_t$ is stationary. Unfortunately, having a stationary Z_t which is a linear combination of log returns is not useful in terms of trading the stocks themselves, as the α_t and β_t cannot be used directly to calculate the quantity of P_t^A and P_t^B to purchase long and sell short. We can very easily get the price level back from the log returns of one stock, but to do so for a linear combination is a different matter altogether.

That is, for

$$X_t = \log\left(\frac{P_t^A}{P_{t-1}^A}\right) \tag{2.69}$$

and through the following:

$$\begin{aligned} & P_1^A \exp\left\{\sum_{t=2}^{j} \log\left(\frac{P_t^A}{P_{t-1}^A}\right)\right\} \text{ for } j = 2, \ldots, n \\ &= P_1^A \exp\left\{\log\left(\frac{P_j^A}{P_1^A}\right)\right\} \text{ for } j = 2, \ldots, n \\ &= P_j^A \text{ for } j = 2, \ldots, n \end{aligned} \tag{2.70}$$

since log returns can be telescoped in a sum. However, for a linear combination,

$$\begin{aligned} Z_t &= \alpha\, X_t + \beta\, Y_t \\ Z_t &= \alpha \log\left(\frac{P_t^A}{P_{t-1}^A}\right) + \beta \log\left(\frac{P_t^B}{P_{t-1}^B}\right) \end{aligned} \tag{2.71}$$

we do not have the same application of being able to convert this back to a linear combination in terms of the price level of the stocks P_t^A and P_t^B. That is, if we take the exponential of the cumulative sum of the linear combination, we arrive with

$$\begin{aligned}
&\exp\left\{\sum_{t=2}^{j}\left(\alpha\log\left(\frac{P_t^A}{P_{t-1}^A}\right)+\beta\log\left(\frac{P_t^B}{P_{t-1}^B}\right)\right)\right\} \text{ for } j=2,\ldots,n \\
&=\exp\left\{\alpha\log\left(\frac{P_j^A}{P_1^A}\right)+\beta\log\left(\frac{P_j^B}{P_1^B}\right)\right\} \text{ for } j=2,\ldots,n \\
&=\left(\frac{P_j^A}{P_1^A}\right)^{\alpha}\left(\frac{P_j^B}{P_1^B}\right)^{\beta} \text{ for } j=2,\ldots,n\,.
\end{aligned} \tag{2.72}$$

Thus, we cannot get a useful interpretation out of α and β from the costationary solution from log returns as our X_t and Y_t. This demonstration is obviously not time-varying as in our costationary solution, which makes using the log-return measure even more complicated to trade from. Hence a measure we used for X_t and Y_t, whilst fulfilling the zero-mean requirement, is the difference of the stock prices. That is,

$$\begin{aligned}
X_t &= \Delta(P_t^A) = P_t^A - P_{t-1}^A \\
Y_t &= \Delta(P_t^B) = P_t^B - P_{t-1}^B
\end{aligned} \tag{2.73}$$

Using this measure, we can obtain the estimates α_t and β_t that form the costationary solution $Z_t = \alpha_t X_t + \beta_t Y_t$ and the α's and β's can used directly towards the amount of each of stock A and B we want to purchase or short. Since we have a costationary solution Z_t, we can add this to the differenced values $\alpha_t P_{t-1}^A + \beta_t P_{t-1}^B$ to get $P_t^S = \alpha_t P_t^A + \beta_t P_t^B$. Ideally, since Z_t constitutes the change in the spread of $\alpha_t P_t^A + \beta_t P_t^B$, and the price differences follow a relatively constant variance in contrast to before (when they were not stationary), the spread P_t^S should also be more stable. Of course, this relies on the relationship between P_t^A and P_t^B to still lie close to each other. If that relationship breaks, ultimately there will be no profitable opportunity anymore. We use Cardinali and Nason (2011)'s R package "costat" and Nason's R pacakge "wavethresh" to obtain our solutions.

2.8 Comparison of the Costationarity Method with the Minimum Distance Method

We saw in the cointegration examples previously that for a training set of 3 years of historical data, 13 cointegrated pairs were found. However, after 6 months, only 3 remained cointegrated. For our test, we prefer to test pairs whose relationships remain relatively stable for longer periods of time. The minimum distance method was much better at finding such pairs. Using training data of 512 trading days, or roughly 2 years, the 10 pairs of stocks in the NASDAQ 100 (91 total stocks) with the smallest least squared distance were used to find costationary solutions using our spread metric in Equation 2.73. For the algorithm we used, the training data length had to be a power of 2, so we used the most relevant set to our previous simulation. Using 512 data points instead of 1024 also shortened the algorithm running time by a huge margin.

Taking the differenced stock prices of each stock, and finding a costationary solution meant that we found α_t and β_t. We restricted the changing values of α_t and β_t to 4, as it would be difficult to arrive at a suitable test set if we allowed α and β to vary any further. That is, each α and β was allowed to last for 128 trading days in the training set. The test set then utilized the last known α and β to trade on the stocks for a period of 50 trading days (i.e. for each solution we had computed values for $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2, \beta_3, \beta_4$, and the α_4 and β_4 were the coefficients used to trade on the test set). We wanted to be able to trade the stocks for a short period of time before another change in the α or β, but also a long enough time for the spread to be able to diverge and revert for profit. For each training set, the algorithm from Cardinali and Nason (2011) generates a number of solutions that all can be considered stationary, but with different α and β for each solution. The reason this is so is because the α's and β's are formed from a randomized starting point, so many different solutions are possible. We generated 10 solutions per training set, but ideally much more should be generated and averaged to get a good idea whether this method can work with many different coefficients that all try to achieve costationarity.

In the solutions, not all tests have 10 usable solutions because the algorithm does not necessarily find convergent solutions. In addition, we only accept the solutions with our test set α and β being of opposite sign, as we wish to still use a long-short strategy.

For each stock pair, the training and testing was repeated 10 times, with each consecutive training period covering the last 512 trading days, moving 50 trading days forward after each test. A list of the training and testing periods can be seen in Table 2.1.

For a comparison, the minimum-distance simple spread of the difference between the pairs of stocks was used to contrast with the profits from the costationarity method. Ultimately, because the α_t's and β_t's were computed numerically, there are some very drastically different values in many of the simulations. As a result, the total profit, and the average profit for each test cannot be compared very well as the investment value also varies very greatly. However, the number of trades and return on income is still very relevant and is the metric we compare on. As before, with the cointegration application, the return on income is calculated as in Equation 1.44.

Again, as we have generated 10 solutions from each training set, there are a significant amount of returns to compare. For simplicity, we have averaged all the returns with the arithmetic mean from each possible solution and have arrived at one return value for each test set. The average returns of each pair can be seen in Tables 2.2, 2.3, and 2.4. If we examine the values in these figures, we can see that the costationarity method (CM) outperforms the minimum distance method (MDM) quite often. There are cases when the MDM does perform slightly better, but these are not altogether that common. There are also some cases when the MDM does drastically better than the CM; but by and large the CM is relatively stronger while the stocks are still quite close together. We can examine this in Figure 2.5. Looking at the stock pairs' movement, as several stocks reach around the 5th test set they are apart from each other much more than in the general training set and in the first 4 test sets. Thus if we only compare the profits from the first 4 tests, there are no significant outliers where the MDM vastly outperforms the CM.

As a general rule, as in the cointegration method, the stocks can be re-evaluated at the

end of 4 tests for new minimum-distance pairs, and the algorithm can be reset from there. The difference in the returns between the CM and the MDM can be seen in Table 2.8. If we sum these differences up from tests 1 to 4, we can see that the CM as a whole has a 59.18% higher return than the MDM. This is in stark contrast to the 2.80% overall higher return from the CM to the MDM if we include all 10 tests.

One final note about the CM is that there are also more trades in general being executed in the test period. This is a good thing typically because for every solution that has more than 1 trade means that there is at least one positive profit even if the second trade is not completed by the end of the test period (and possibly results in an overall loss). However, the best metric to compare results by is still the averaged returns. The average number of trades executed per test and pair can be seen in Tables 2.5, 2.6, and 2.7.

For brevity, we have not included all 1000 plots for each solution, test, and stock (there are 10 solutions per test, 10 test periods per pair, and 10 stock pairs in total). We have only included plots from the pair with the lowest minimum distance. The 10 solutions for the first test of this pair can be seen in Figures 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, and 2.15.

For these figures, it is useful to explain what each plot represents in detail. In Figure 2.6, in the first plots on the top left, "SYMC.Adjusted" and "YAHOO.Adjusted" displays the difference metric in Equation 2.73. The plot titled "SYMC.Adjusted YAHOO.Adjusted" with the blue and black lines represent the training data of the two stocks, with SYMC in black, and YAHOO in blue. For each next set of four plots, the different solutions are represented. The plot with the title "Mean-Removed Spread" displays the trajectory of the spread following the trades using α_t and β_t with the respective means of each section with different coefficients removed. This is done because the coefficents from section to section can vary drastically. The test set for the costationary solution is in green. The plot below displays the full training set and test spread of the MDM, and has the title "Min-Dist Spread". The test spread is highlighted in green. Beside these two, the plots titled "CM Test Spread" and "MDM Test Spread" are zoomed-in versions of the previous plots

starting from the start of the time when coefficients α_4, and β_4 were in effect. The blue horizontal lines representing the trade boundaries and the red horizontal line representing the historical mean. The test spreads in each are again highlighted in green. This plot arrangement is repeated for each test set in the other figures.

The 2 year training period and the 50 day trading periods	
Training Period	**Testing Period**
May 20 2009 - May 31 2011	June 1 2011 - August 10 2011
July 31 2009 - August 10 2011	August 11 2011 - October 20 2011
October 12 2009 - October 20 2011	October 21 2011 - January 3 2012
December 22 2009 - January 3 2011	January 4 2011 - March 15 2012
March 8 2010 - March 15 2012	March 16 2012 - May 25 2012
May 18 2010 - May 25 2012	May 29 2012 - August 7 2012
July 29 2010 - August 7 2012	August 8 2012 - October 17 2012
October 8 2010 - October 17 2012	October 18 2012 - January 2 2013
December 20 2010 - January 2 2013	January 3 2013 - March 15 2013
March 3 2011 - March 15 2013	March 18 2013 - May 28 2013

Table 2.1: The training and testing periods for the MDM and CM elligible stocks in the NASDAQ 100

	SYMC,YHOO		CMCSA,MXIM		INTC,MDLZ		AMAT,FOXA	
	Costat	Mindist	Costat	Mindist	Costat	Mindist	Costat	Mindist
Test 1	-4.171	-5.431	1.290	11.863	4.562	-3.205	7.777	12.765
Test 2	10.194	14.824	5.572	7.424	4.653	10.189	-3.609	-5.684
Test 3	-7.433	1.217	3.167	5.483	3.687	2.601	-5.507	-10.737
Test 4	10.983	-2.028	2.045	-0.480	5.178	-0.651	9.953	0.979
Test 5	8.066	11.003	-4.070	-8.524	0.895	6.124	-2.822	-9.023
Test 6	3.750	6.219	-18.683	-12.655	-3.123	8.627	-12.108	-14.578
Test 7	5.460	7.031	-6.871	-12.689	-8.529	-12.317	-9.352	-12.337
Test 8	-6.769	-1.581	4.170	7.909	1.598	2.018	1.506	-1.933
Test 9	1.122	10.227	-0.480	0.034	-4.616	-5.174	-12.819	-11.521
Test 10	-0.883	-11.474	2.283	-10.246	-7.212	-0.979	4.754	-3.686

Table 2.2: The averaged returns across the useable solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%)

	DISCA,MAT		FOXA,SBUX		FOXA,QVCA		FISV,LLTC	
	Costat	Mindist	Costat	Mindist	Costat	Mindist	Costat	Mindist
Test 1	3.382	2.186	-1.455	-19.248	15.689	12.830	3.656	-0.753
Test 2	-0.714	-2.776	0.464	-9.322	12.132	8.858	6.737	9.391
Test 3	-0.099	-3.771	-0.742	-4.110	3.975	15.899	6.764	0.419
Test 4	-3.327	-5.448	-6.644	-9.449	2.393	3.157	-0.011	-3.040
Test 5	9.944	15.093	0.821	-7.011	11.686	-0.436	-4.115	-8.448
Test 6	3.294	1.660	-27.657	42.530	-9.097	-10.401	2.328	6.502
Test 7	-1.632	10.004	-8.619	0.795	-2.622	-3.165	-1.171	-7.692
Test 8	-2.654	0.000	2.561	21.256	0.704	-5.160	2.984	4.695
Test 9	-4.489	-2.247	-16.581	1.228	-6.509	-13.394	1.761	-1.260
Test 10	0.324	10.781	-3.379	4.245	0.010	-5.023	-4.111	-4.162

Table 2.3: The averaged returns across 10 solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%)

	MAT,VOD		MAT,MYL	
	Costat	Mindist	Costat	Mindist
Test 1	5.345	10.068	-0.099	-1.678
Test 2	7.581	12.341	-16.274	-21.711
Test 3	6.784	6.771	9.502	15.958
Test 4	-9.094	-17.271	-8.186	-11.508
Test 5	4.709	7.547	6.403	0.198
Test 6	2.260	-0.614	5.030	-7.249
Test 7	9.087	-13.020	0.037	-3.123
Test 8	-4.993	-4.509	-1.879	14.705
Test 9	0.788	-9.765	2.919	-0.037
Test 10	-5.242	-6.988	1.034	-2.427

Table 2.4: The averaged returns across the useable solutions for each test and for each method (CM and MDM). The rows indicate which test number the return is representing. Each value is a percent return (%)

	SYMC,YHOO		CMCSA,MXIM		INTC,MDLZ		AMAT,FOXA	
	Costat	Mindist	Costat	Mindist	Costat	Mindist	Costat	Mindist
Test 1	1.167	1.000	1.625	1.000	1.000	1.000	1.625	1.000
Test 2	1.429	1.000	1.286	1.000	1.000	1.000	1.000	1.000
Test 3	1.000	1.000	1.000	1.000	1.571	1.000	1.571	1.000
Test 4	1.167	1.000	1.333	1.000	1.000	1.000	1.750	1.000
Test 5	2.000	2.000	1.111	1.000	2.000	1.000	1.000	1.000
Test 6	1.000	1.000	1.000	1.000	1.200	2.000	1.000	1.000
Test 7	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Test 8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Test 9	1.333	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Test 10	1.250	1.000	1.375	1.000	1.000	1.000	1.000	1.000

Table 2.5: The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column.

	DISCA,MAT		FOXA,SBUX		FOXA,QVCA		FISV,LLTC	
	Costat	Mindist	Costat	Mindist	Costat	Mindist	Costat	Mindist
Test 1	1.333	1.000	1.625	1.000	2.667	2.000	1.429	1.000
Test 2	1.000	1.000	1.000	1.000	2.000	2.000	1.429	2.000
Test 3	1.000	1.000	1.000	1.000	1.000	3.000	1.556	1.000
Test 4	1.000	1.000	1.000	1.000	1.000	1.000	1.571	1.000
Test 5	1.200	1.000	1.200	1.000	1.000	1.000	1.000	1.000
Test 6	1.429	1.000	1.500	2.000	1.333	1.000	1.000	1.000
Test 7	1.000	1.000	1.000	1.000	1.000	1.000	1.200	1.000
Test 8	1.556	1.000	1.000	1.000	1.200	1.000	1.167	1.000
Test 9	1.000	1.000	1.000	1.000	1.000	1.000	1.667	1.000
Test 10	1.167	1.000	1.000	1.000	1.000	1.000	1.111	1.000

Table 2.6: The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column.

	MAT,VOD		MAT,MYL	
	Costat	Mindist	Costat	Mindist
Test 1	1.000	1.000	1.625	2.000
Test 2	2.000	2.000	1.000	1.000
Test 3	1.000	1.000	1.000	1.000
Test 4	1.000	1.000	1.000	1.000
Test 5	1.000	1.000	1.286	1.000
Test 6	1.333	1.000	1.667	1.000
Test 7	1.000	1.000	1.000	1.000
Test 8	1.000	1.000	1.429	1.000
Test 9	1.286	1.000	1.444	1.000
Test 10	1.200	1.000	2.000	1.000

Table 2.7: The total number of trades executed on each of the 10 tests for CM and MDM. The stock pairs that are relevant are labelled at the top of each column.

	Pair									
	1	2	3	4	5	6	7	8	9	10
Test 1	1.26	-10.57	7.77	-4.99	1.20	17.79	2.86	4.41	-4.72	1.58
Test 2	-4.63	-1.85	-5.54	2.08	2.06	9.79	3.27	-2.65	-4.76	5.44
Test 3	-8.65	-2.32	1.09	5.23	3.67	3.37	-11.92	6.35	0.01	-6.46
Test 4	13.01	2.52	5.83	8.97	2.12	2.80	-0.76	3.03	8.18	3.32
Test 5	-2.94	4.45	-5.23	6.20	-5.15	7.83	12.12	4.33	-2.84	6.20
Test 6	-2.47	-6.03	-11.75	2.47	1.63	-70.19	1.30	-4.17	2.87	12.28
Test 7	-1.57	5.82	3.79	2.98	-11.64	-9.41	0.54	6.52	22.11	3.16
Test 8	-5.19	-3.74	-0.42	3.44	-2.65	-18.70	5.86	-1.71	-0.48	-16.58
Test 9	-9.11	-0.51	0.56	-1.30	-2.24	-17.81	6.88	3.02	10.55	2.96
Test 10	10.59	12.53	-6.23	8.44	-10.46	-7.62	5.03	0.05	1.75	3.46

Table 2.8: The difference between the averaged returns of each method (CM and MDM) for each test and each pair. The rows indicate which test number the return is representing. Each value is a percent return (%), with positive values representing the CM performing better than the MDM, and negative values representing the MDM performing better than the CM. The pairs in order from 1 to 10 are SYMC & YHOO, CMCSA & MXIM, INTC & MDLZ, AMAT & FOXA, DISCA & MAT, FOXA & SBUX, FOXA & QVCA, FISV & LLTC, MAT & VOD, and MAT & MYL.

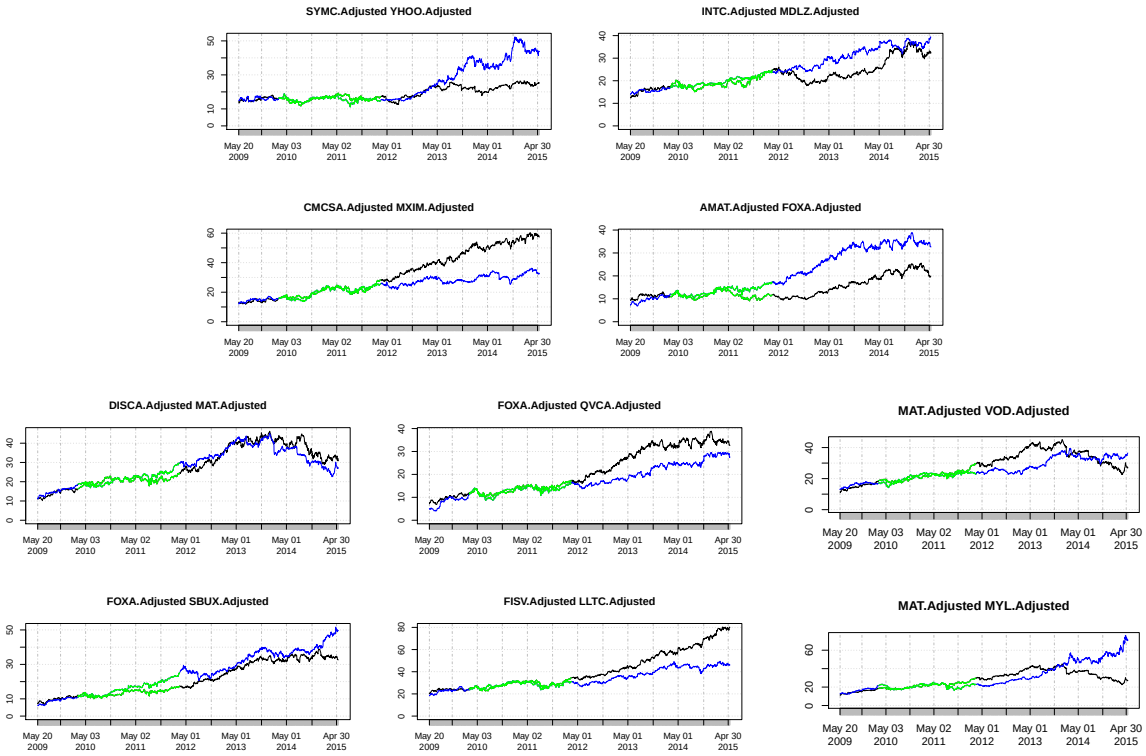

Figure 2.5: The plots of the prices of the stock pairs. The black and blue lines represent the stock prices of the first and second of the stocks in the title of each plot respectively. The green line is where the stocks start to diverge in some cases, and this corresponds with the 5th test set. It is for this reason why we consider comparing the returns only from tests 1-4 with the tests from 1-10, and there is a noticeable difference albeit mainly from one outlier.

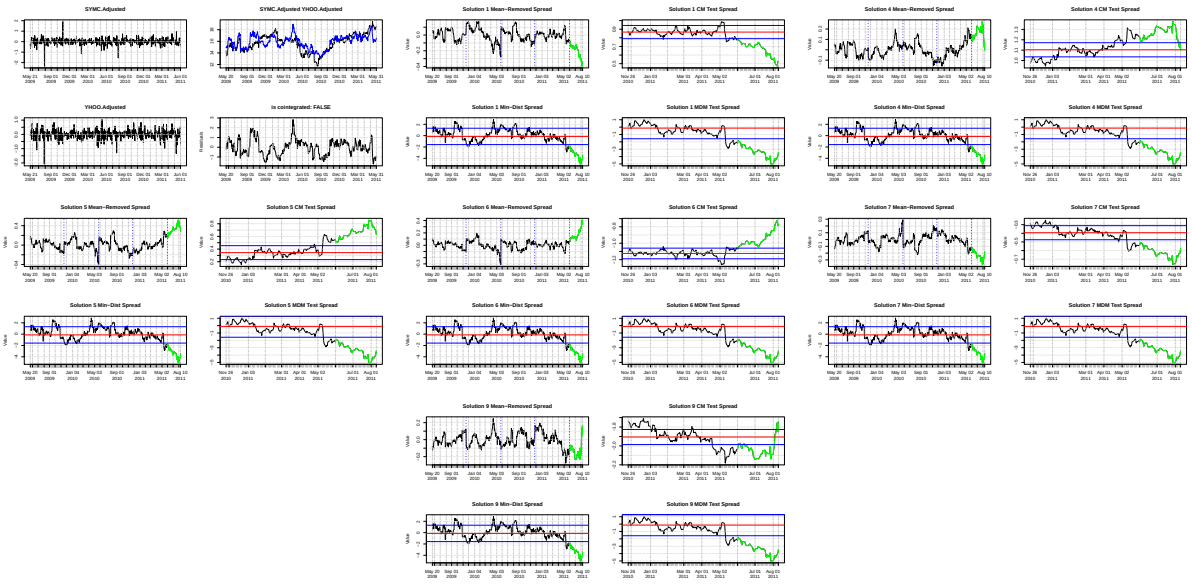

Figure 2.6: The plots of the 1st test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

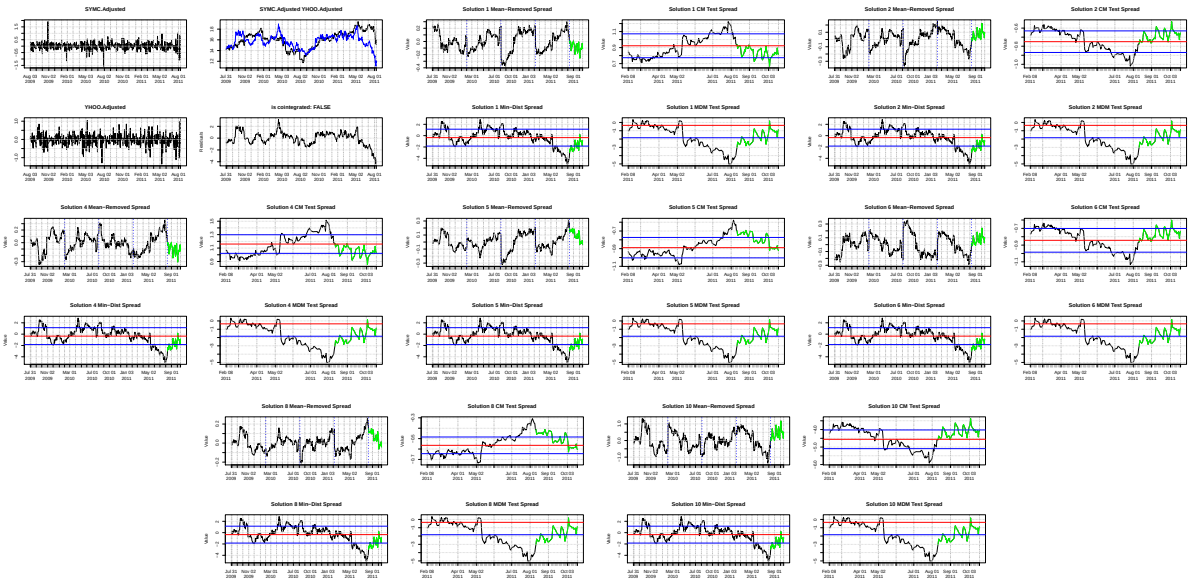

Figure 2.7: The plots of the 2nd test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

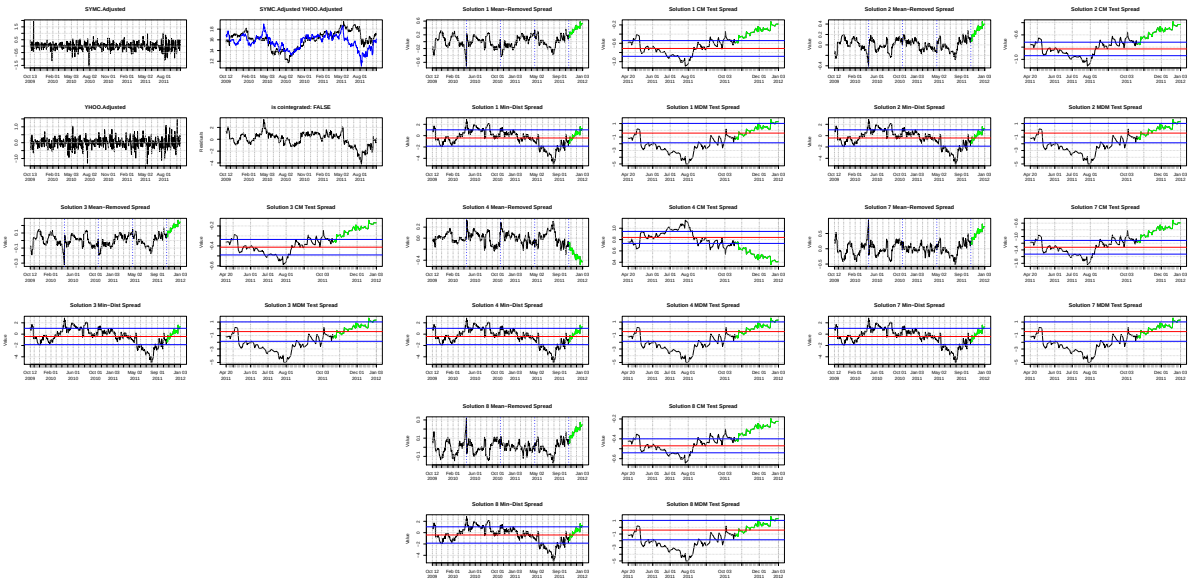

Figure 2.8: The plots of the 3rd test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

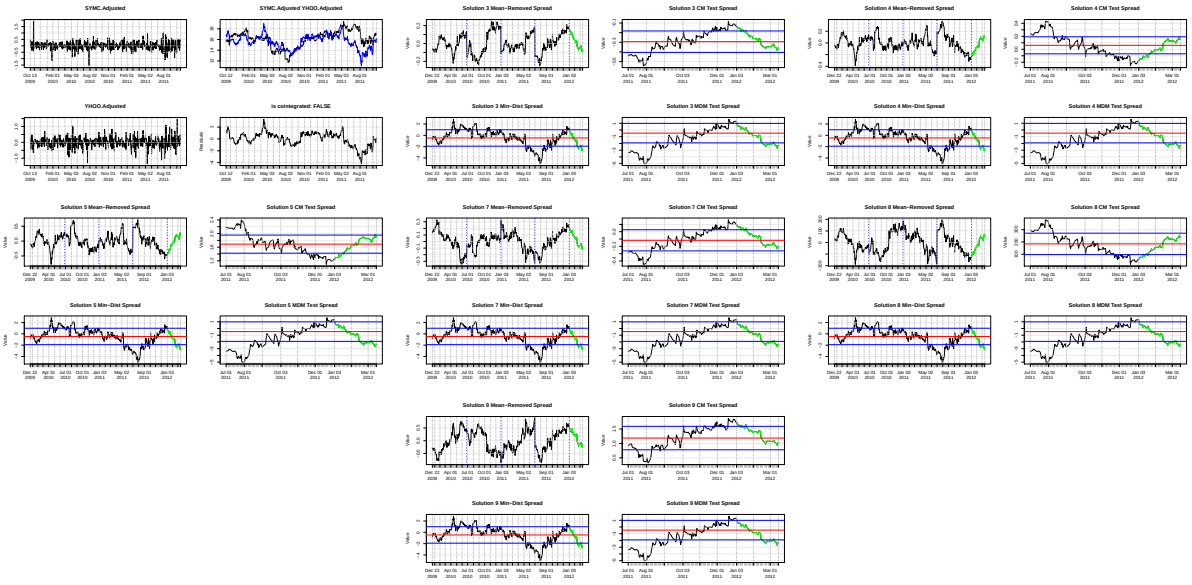

Figure 2.9: The plots of the 4th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

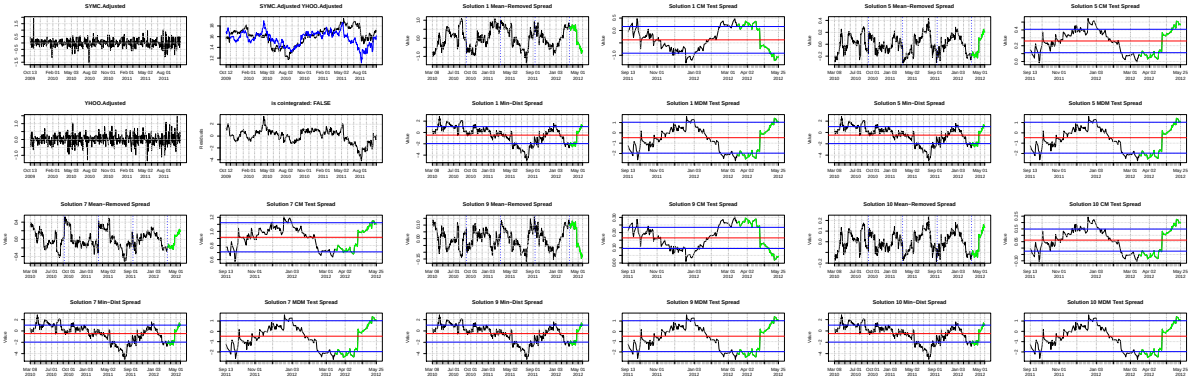

Figure 2.10: The plots of the 5th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

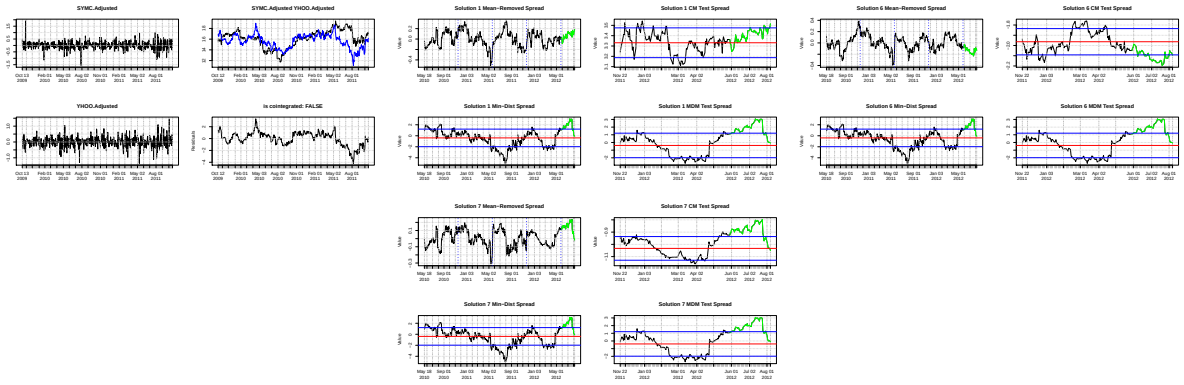

Figure 2.11: The plots of the 6th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

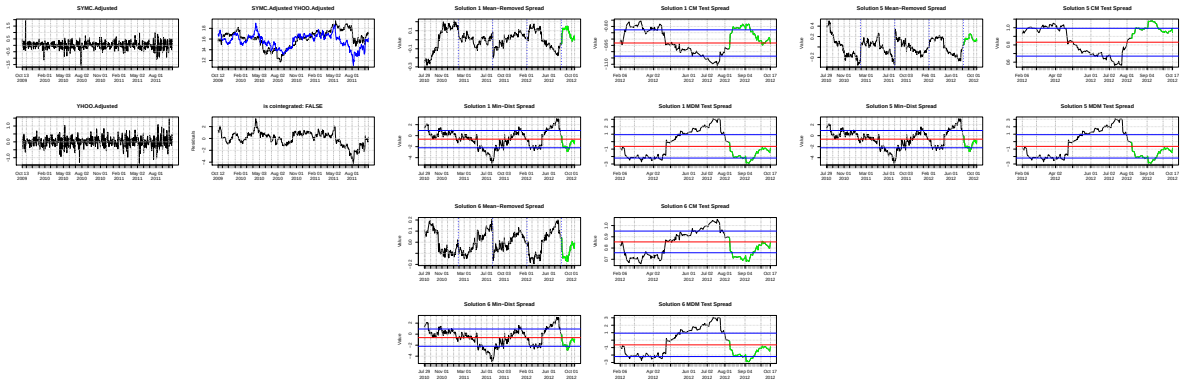

Figure 2.12: The plots of the 7th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

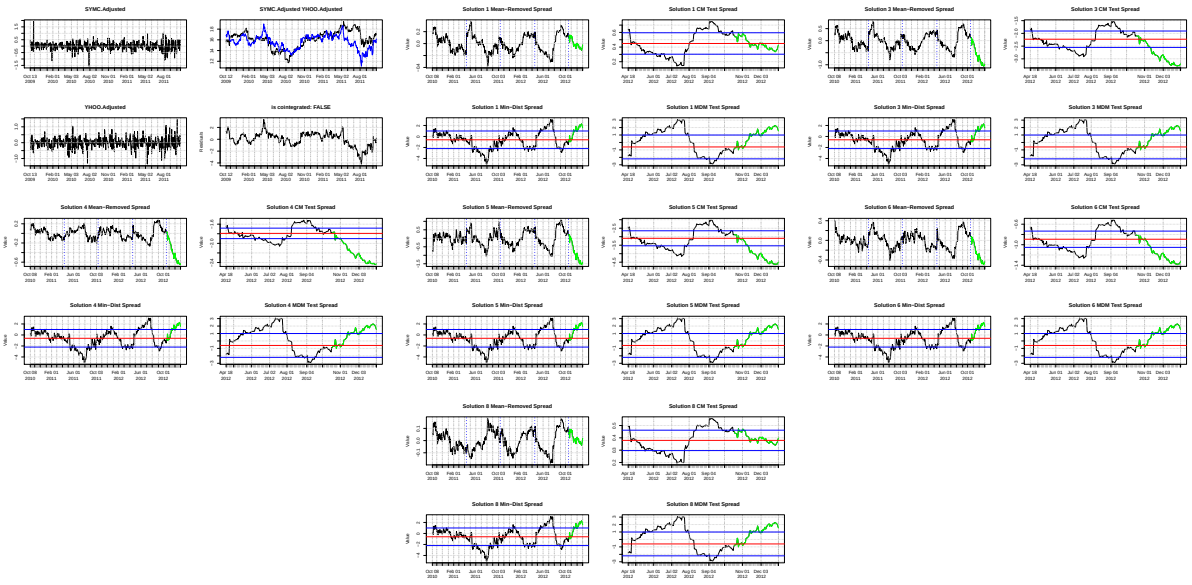

Figure 2.13: The plots of the 8th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

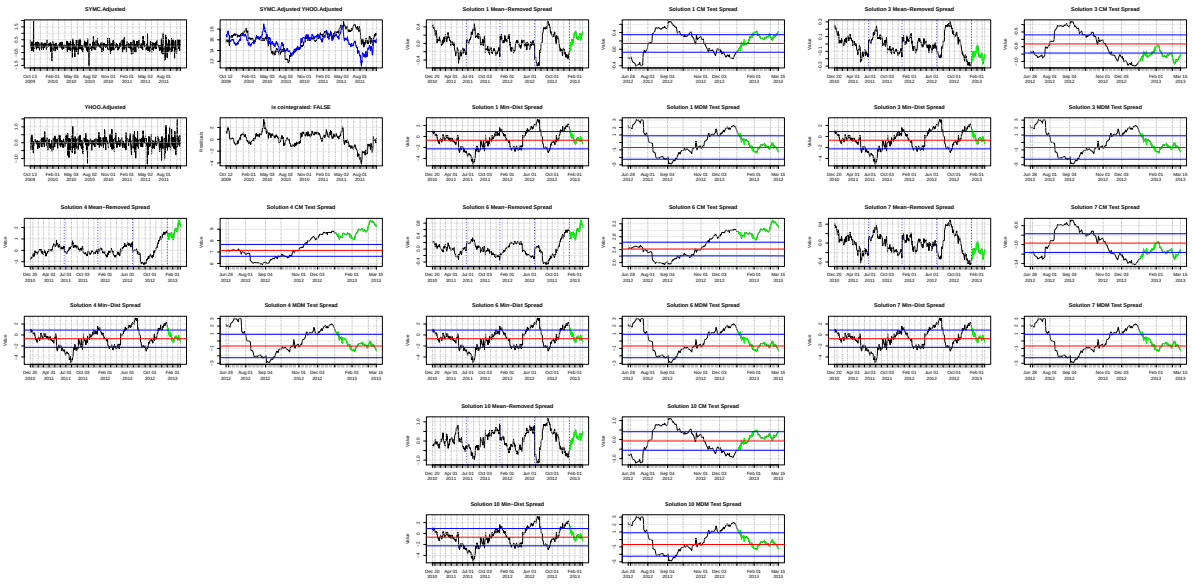

Figure 2.14: The plots of the 9th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

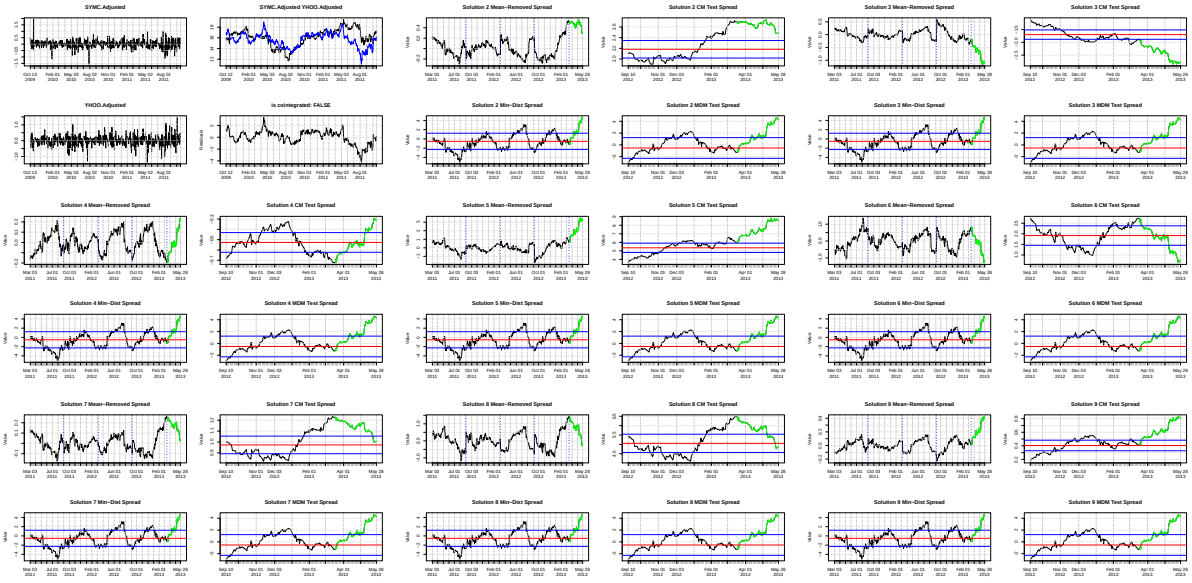

Figure 2.15: The plots of the 10th test in the costationary solutions versus the minimum-distance solutions of the stock pair SYMC,YAHOO.

2.9 Comparison of the Costationarity Method with the Cointegration Method

The costationarity method (CM) is more suited to compare with the minimum distance method (MDM) as the requirement of the stock pairs being cointegrated severely limits the number of available pairs, as well as the possible problem of having false positive pairs and the possibility of the cointegration relationship dissapearing. However, there is still a comparison to be made with the cointegration method (CIM). Thus, we attempt to compare the costationarity method (CM) with the cointegration method (CIM) between pairs which seem to have a lasting cointegrating relationship, and contrast this with pairs that do not.

For this comparison we do not use the cointegrated pairs in the previous application of just the cointegration method. This is because the costationarity method must use data sets with lengths that are powers of 2, and again we use a training set of 512 trading days. However, seeing as the stocks may not remain cointegrated for that long, we limit the number of tests to 5 consecutive periods, or 250 trading days. That is roughly around the same as 12 months of trading. The first two pairs, (CMCSA,GILD) and (CSCO,WYNN), have been deliberately selected for comparison after noticing that the cointegration relationship breaking down very quickly. The last two pairs, (HSIC,LBTYA) and (QVCA,SIAL), have strongly persisting cointegration relationships. These trajectories of the stock prices and the corresponding cointegrating relationships can be seen in Figure 2.30. The training and testing periods for the stocks can be seen in Figure 2.9.

As the stocks pairs here are found using the cointegration method, it is not as simple as the costationarity application on the pairs found using the minimum distance method. We can no longer just apply the algorithm on the differenced stock prices of each pair. Rather, we do so on the differenced regression relationship and the corresponding differenced stock price. That is, for P_t^A and P_t^B, the prices of the stocks of a cointegrated pair, they have

the cointegration relationship:

$$P_t^B = \alpha + \beta P_t^A + \epsilon_t \,. \tag{2.74}$$

Then, as before, we can take the differences of P_t^B and $\alpha + \beta P_t^A$ for our spread metric in Equation 2.73:

$$\begin{aligned} \Delta P_t^B &= P_t^B - P_{t-1}^B \quad \text{and} \\ \Delta(\alpha + \beta P_t^A) &= \alpha + \beta P_t^A - (\alpha + \beta P_{t-1}^A) \\ &= \beta(P_t^A - P_{t-1}^A) \\ &= \beta \Delta(P_t^A) \,. \end{aligned} \tag{2.75}$$

Stationary linear combinations are then found using the costationarity algorithm with this spread metric. That is, for the original solution

$$Z_t = \alpha_t X_t + \beta_t Y_t \,, \tag{2.76}$$

we will let

$$\begin{aligned} X_t &= \Delta P_t^B \quad \text{and} \\ Y_t &= \beta \Delta(P_t^A) \end{aligned} \tag{2.77}$$

Beware that the α_t and β_t are the time-varying coefficients found from the costationarity algorithm and the α and β are the coefficients from the regression of the stock prices of the stock pair.

Again, for this comparison we look at the averaged returns across the useable solutions generated from the costationarity algorithm. The solutions that don't converge and the solutions without opposite signed α_4's and β_4's are not used. It turns out that the returns for the cointegrated stocks that stay cointegrated (HSIC,LBTYA) and (QVCA,SIAL), actually perform much better with the cointegration method than the costationarity method. Where the cointegration relationship actually is a false positive, or just disappears relatively quickly, with the pairs (CMCSA,GILD) and (CSCO,WYNN), the costationarity method performs much better across the five tests. The relative return table with the difference between the two methods can be seen in Table 2.12.

However, that is just a relative measure: if we take a look at the absolute returns for the pairs which have the faltering cointegration relationship, we notice that the returns altogether are not very good. The costationarity method (CMCSA,GILD) gives quite a negative return on Test 1, and performs much better on Tests 2,4, and 5, but that is just because the cointegration method performs spectacularly poorly. The costationarity method returns are quite small with returns of 0.452% 0.877%, and 4.772% in Tests 2, 4 and 5. Similarly with the (CSCO,WYNN) pairing, the CM only outperforms CIM when the CIM performs very poorly. In general, both these stock pairs are not tradeable with either method, although the cointegration mitigates the loss quite a bit better.

With one of the pairs that do remain cointegrated however, (HSIC,LBTYA), the cointegration method performs much better than the costationarity method. There is a stark contrast between the returns of the two methods in Tests 1 and Test 4. However a closer look at the spread diagrams shows why this is the case. This can be seen in Figures 2.20 and 2.23. For the cointegration method, there is a profit when the stock rises from the lower threshold to the mean, then again when it jumps up to the upper threshold and immediately back down and up again. It can be seen that the costationarity solutions are quite similar; however they do not spike like the cointegration spread and as such there is only one profitable trade. For Test 4, the spreads are quite similar but there is one solution for the costationarity method that results in quite a large loss so the returns average out to a lot less. In the stock pair (QVCA,SIAL), there is not much of a difference in the returns of the two methods. All of the solutions of each of the stock pairs can be seen in Figures 2.16, 2.17, 2.18, 2.19, 2.20, 2.21, 2.22, 2.23, 2.24, 2.25, 2.26, 2.27, 2.28, and 2.29. We have chosen only to include the first two tests for the solutions to the first two stock pairs (CMCSA,GILD) and (CSCO,WYNN), because we do not believe that they should be traded beyond those two test periods. The cointegration relationship clearly disintegrates and as such, should not be traded once this is clear.

The number of trades here is quite similar for both methods: sometimes the costationarity method has more trades executed than the cointegration method; but sometimes it is the other way around. In this regard, we can reasonably conclude that the costationarity

method is not a huge improvement over the cointegration method for these tests.

The 2 year training period and the 50 day trading periods	
Training Period	**Testing Period**
May 12 2010 - May 21 2012	May 22 2012 - August 01 2012
July 23 2010 - August 01 2012	August 02 2012 - October 11 2012
October 04 2010 - October 11 2012	October 12 2012 - December 26 2012
December 14 2010 - December 26 2012	December 27 2012 - March 11 2013
February 25 2011 - March 11 2013	March 12 2013 - May 21 2013

Table 2.9: The training and testing periods for the CIM and CM elligible stocks in the NASDAQ 100

	CMCSA,GILD		CSCO,WYNN		HSIC,LBTYA		QVCA,SIAL	
	Costat	CIM	Costat	CIM	Costat	CIM	Costat	CIM
Test 1	-14.777	4.005	-7.509	-11.639	1.305	16.524	4.032	6.681
Test 2	0.452	-17.555	21.435	29.149	-11.996	-12.919	9.927	9.444
Test 3	-1.780	-4.526	-2.832	1.356	1.373	5.011	3.137	2.830
Test 4	0.877	-19.942	-9.410	-2.931	-0.980	10.494	0.803	N/A
Test 5	4.772	-27.192	-3.156	-13.384	-3.590	5.665	3.801	4.259

Table 2.10: The averaged returns across the solutions used for each test and for each method (CM and CIM). The rows indicate which test number the return is representing. Each value is a percent return (%).

	CMCSA,GILD		CSCO,WYNN		HSIC,LBTYA		QVCA,SIAL	
	Costat	CIM	Costat	CIM	Costat	CIM	Costat	CIM
Test 1	1.000	1.000	1.000	1.000	1.571	3.000	1.667	2.000
Test 2	1.000	1.000	1.000	1.000	1.000	1.000	3.000	3.000
Test 3	1.000	1.000	1.250	1.000	1.000	1.000	1.000	1.000
Test 4	1.125	1.000	1.000	1.000	1.667	2.000	1.125	0
Test 5	1.714	1.000	1.000	1.000	1.000	1.000	2.500	2.000

Table 2.11: The total number of trades executed on each of the 5 tests for CM and CIM. The stock pairs that are relevant are labelled at the top of each column.

	CMCSA,GILD	CSCO,WYNN	HSIC,LBTYA	QVCA,SIAL
Test 1	-18.78	4.13	-15.22	-2.65
Test 2	18.01	-7.71	0.92	0.48
Test 3	2.75	-4.19	-3.64	0.31
Test 4	20.82	-6.48	-11.47	0.80
Test 5	31.96	10.23	-9.25	-0.46

Table 2.12: The difference between the averaged returns of each method (CM and CIM) for each test and each pair. Each value is a percent return (%), with positive values representing the CM performing better than the CIM, and negative values representing the CIM performing better than the CM.

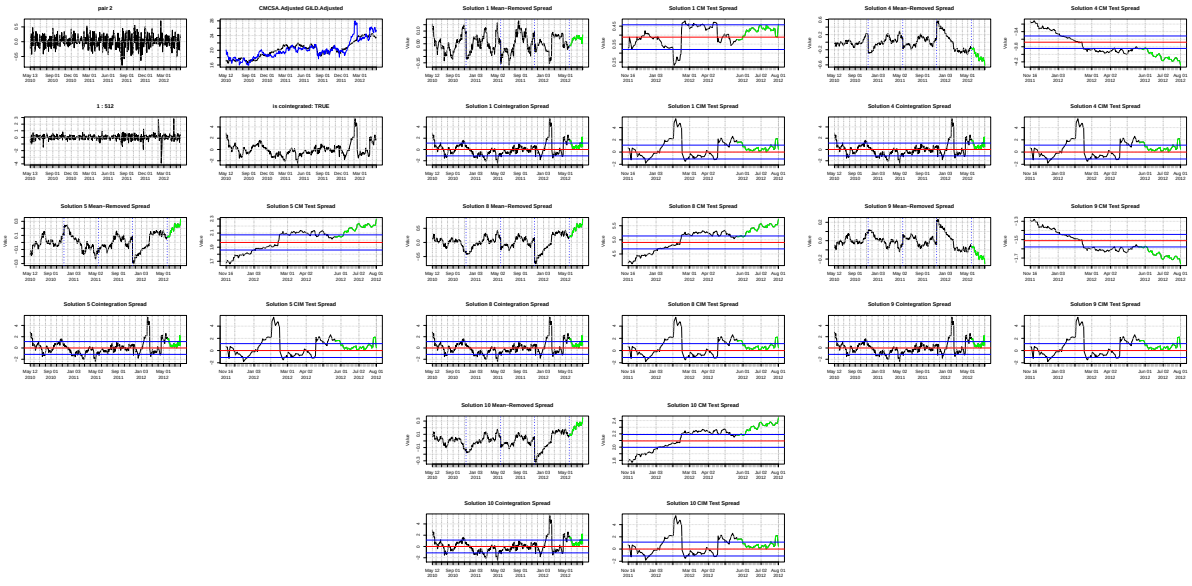

Figure 2.16: The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair CMCSA,GILD.

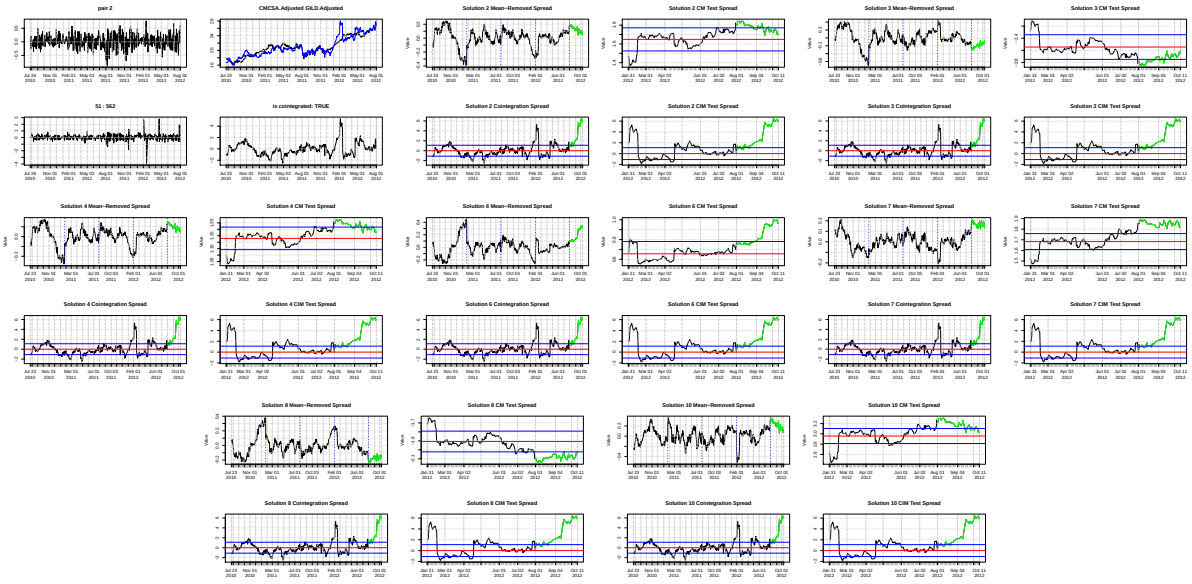

Figure 2.17: The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair CMCSA,GILD.

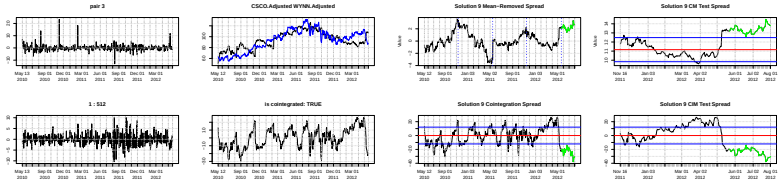

Figure 2.18: The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair CSCO,WYNN.

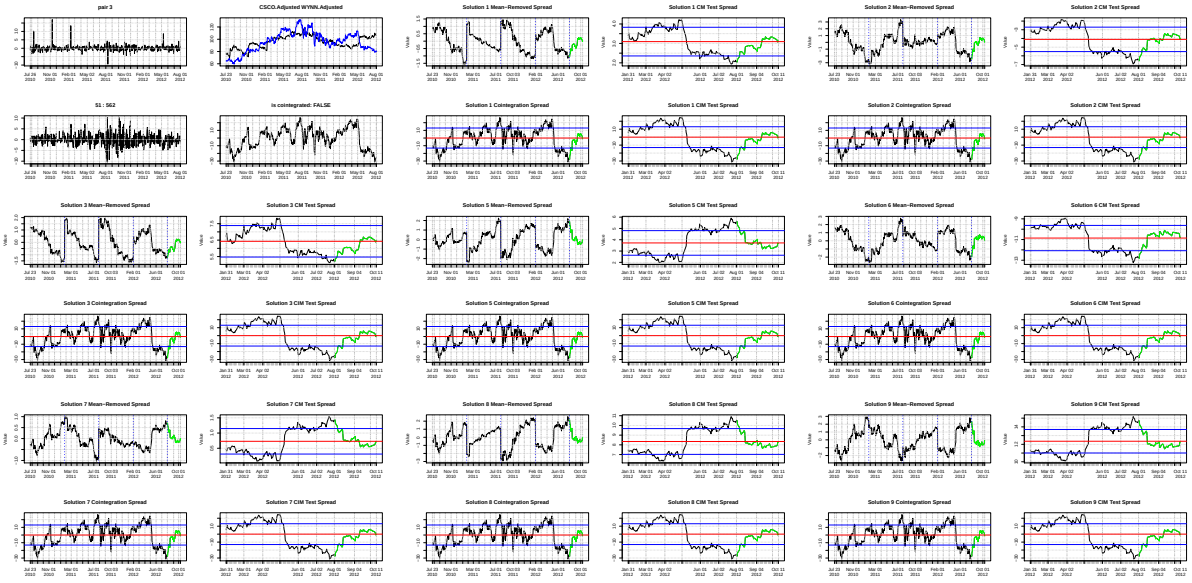

Figure 2.19: The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair CSCO,WYNN.

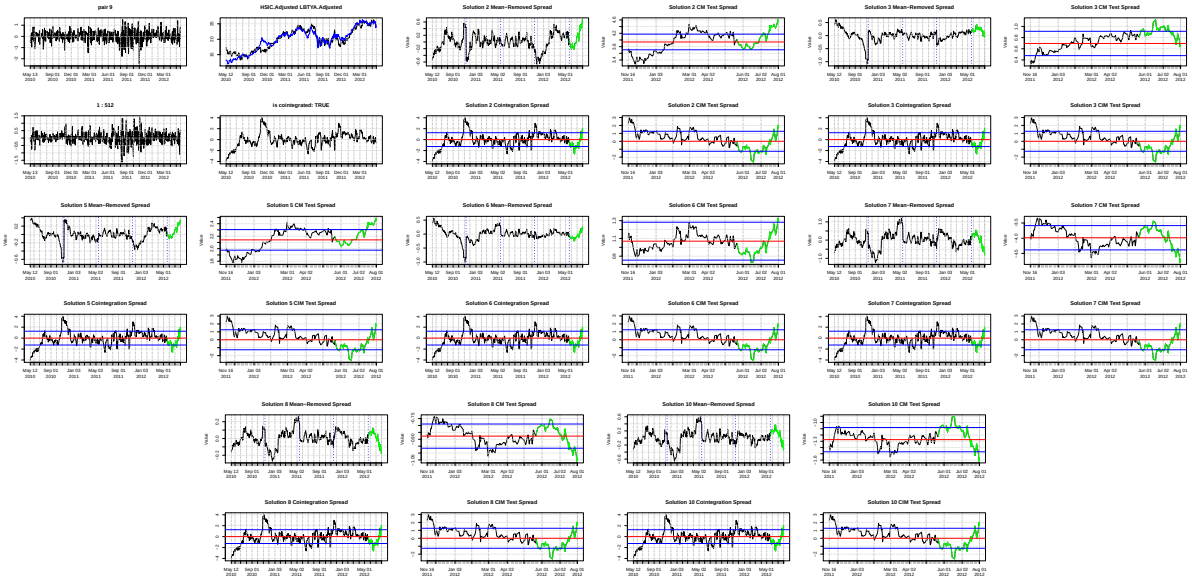

Figure 2.20: The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA.

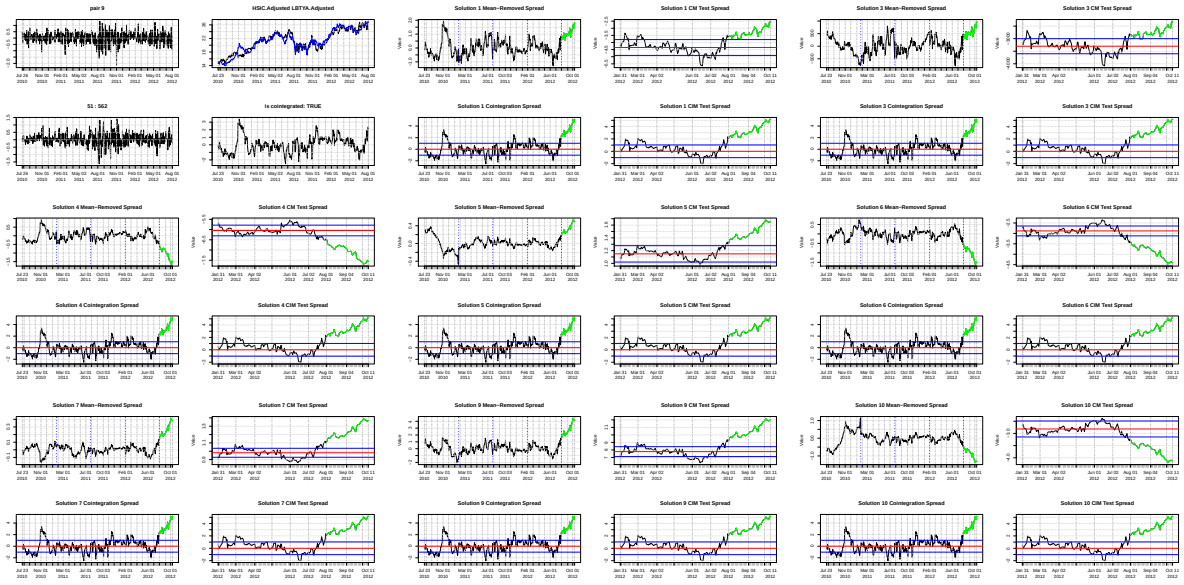

Figure 2.21: The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA.

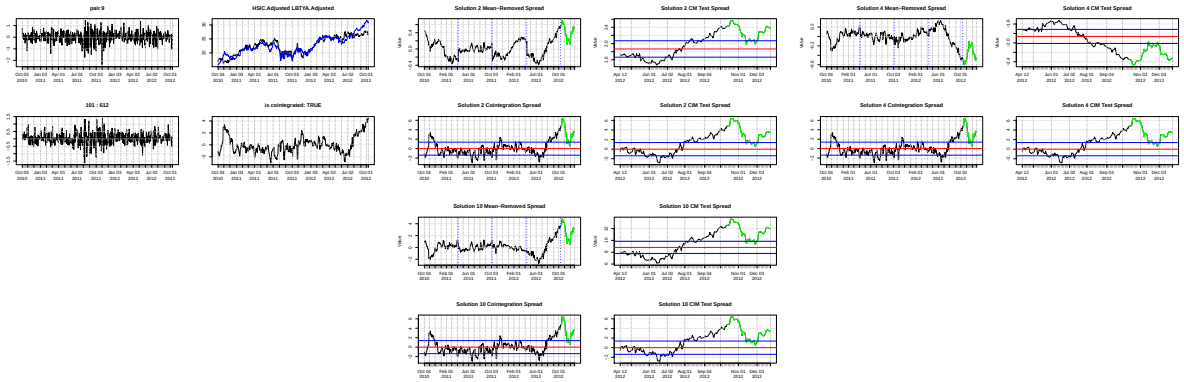

Figure 2.22: The plots of the 3rd test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA.

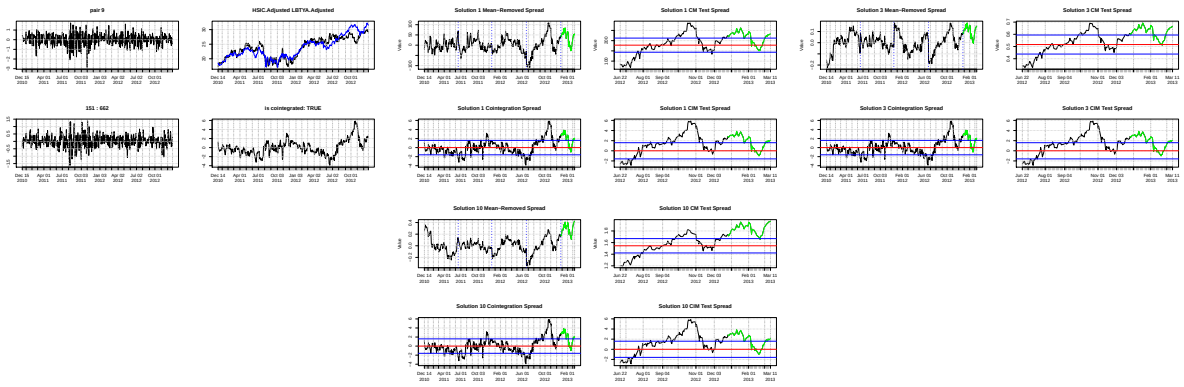

Figure 2.23: The plots of the 4th test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA.

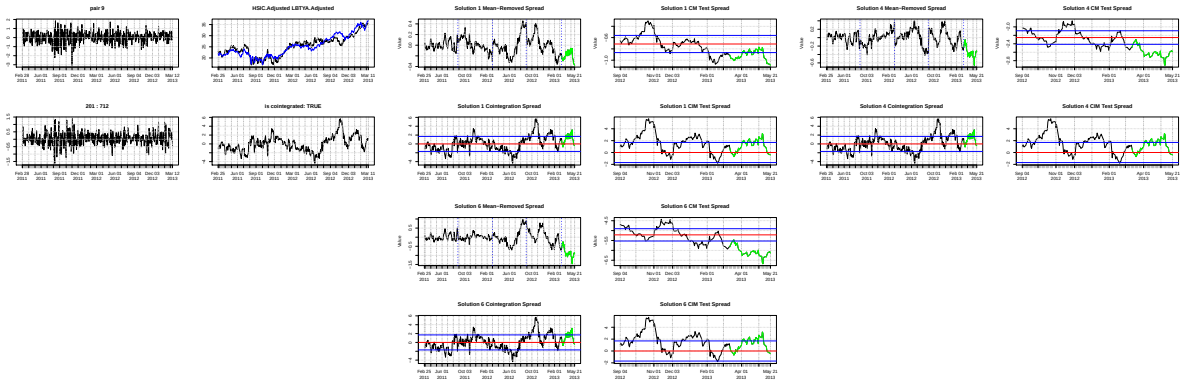

Figure 2.24: The plots of the 5th test in the costationary solutions versus the cointegration solutions of the stock pair HSIC,LBTYA.

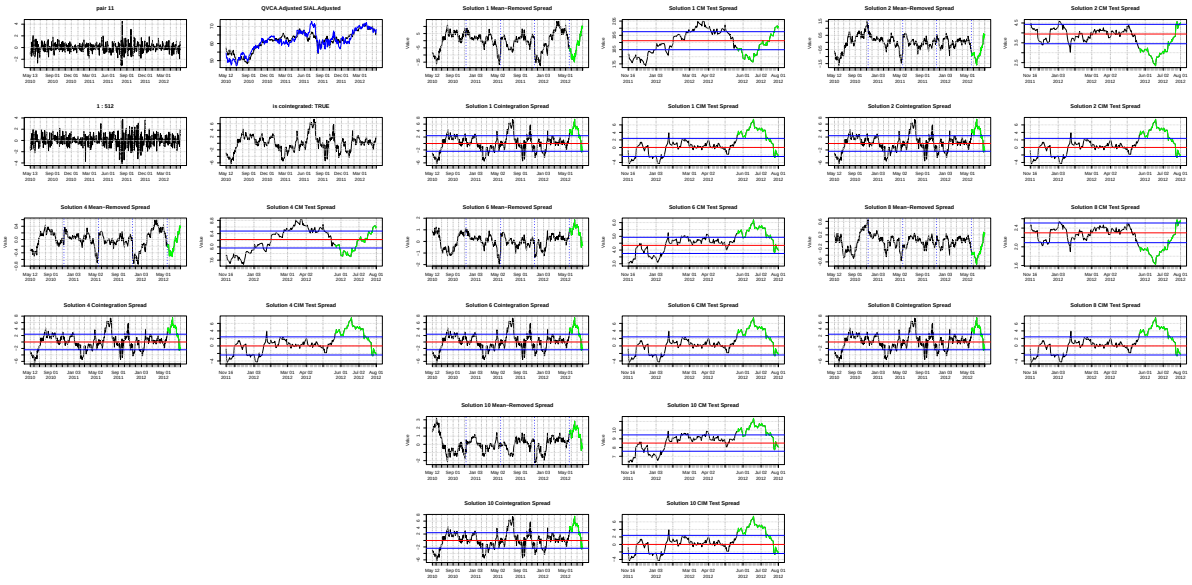

Figure 2.25: The plots of the 1st test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL.

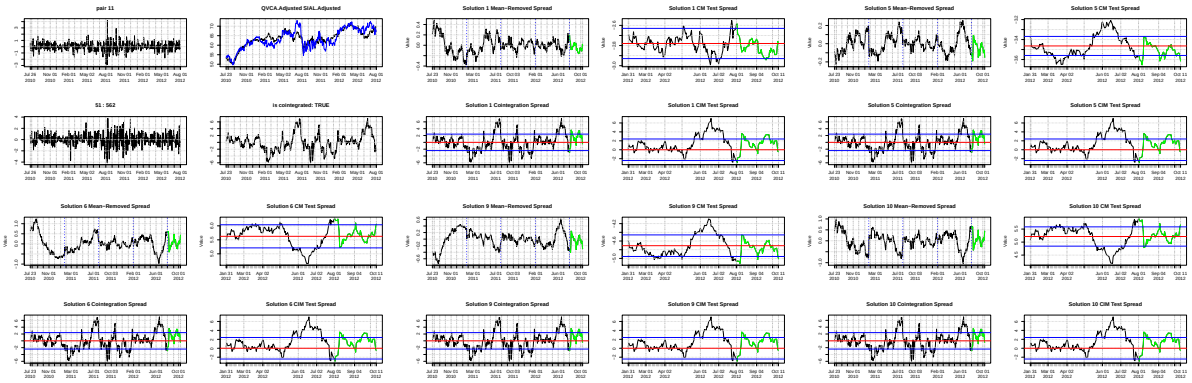

Figure 2.26: The plots of the 2nd test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL.

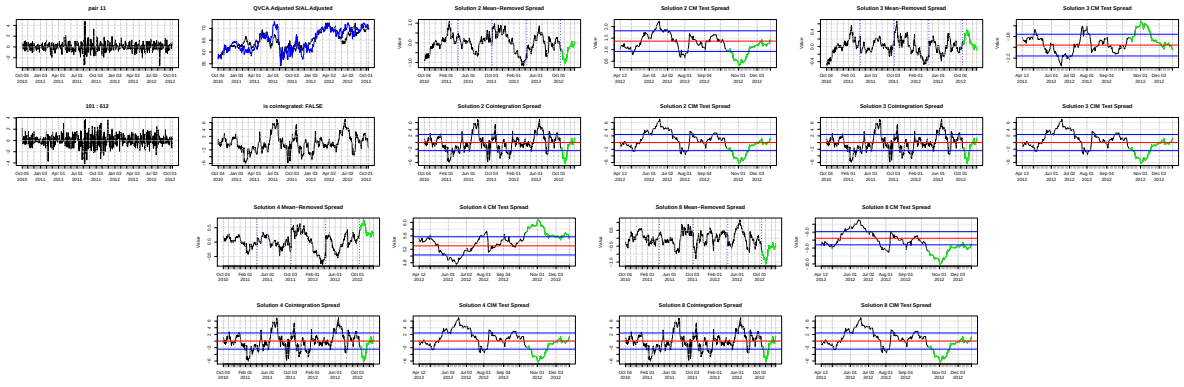

Figure 2.27: The plots of the 3rd test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL.

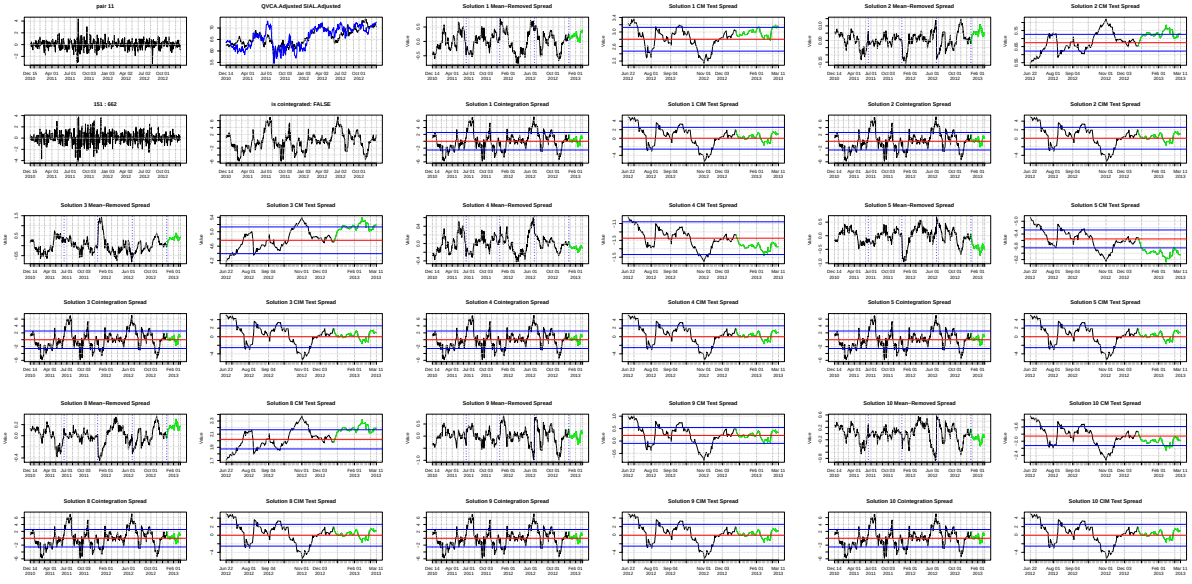

Figure 2.28: The plots of the 4th test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL.

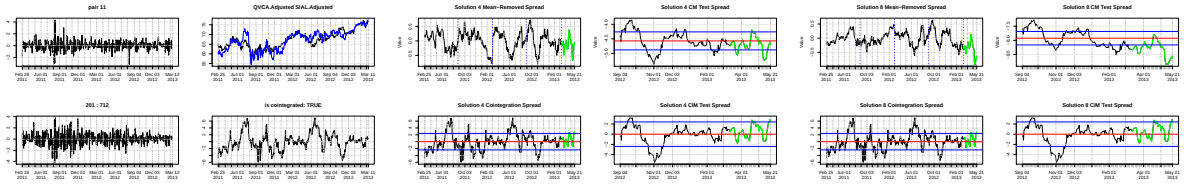

Figure 2.29: The plots of the 5th test in the costationary solutions versus the cointegration solutions of the stock pair QVCA,SIAL.

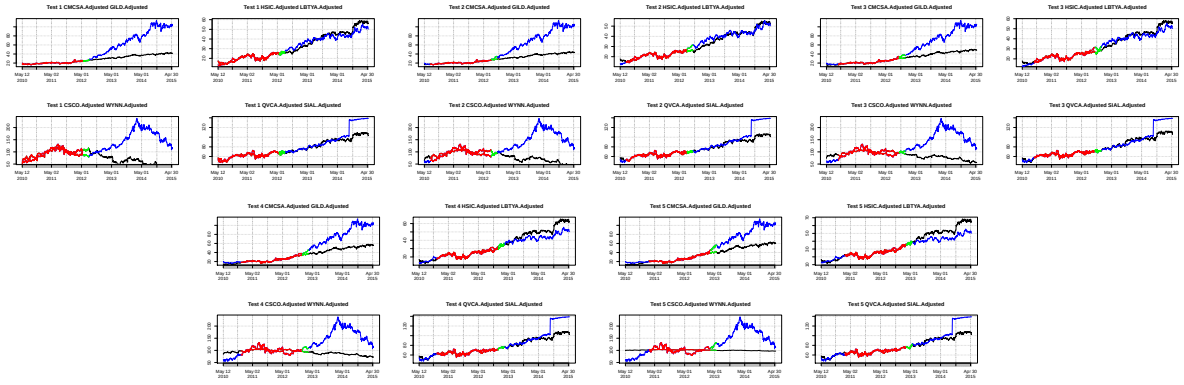

Figure 2.30: The plots of the trajectories of the stock pairs (P_t^B) and their cointegration relationship counterpart ($\alpha + \beta P_A^t$) over the 5 test periods. The pairs CMCSA,GILD and CSCO,WYNN are false positives for cointegration, while the pairs HSIC,LBTYA and QVCA,SIAL have much longer lasting cointegrating relationships. The red lines represent the training set and the green lines represent the test set. The blue lines represent P_t^B (the second stock in the titles), while the black lines represent $\alpha + \beta P_A^t$, where P_A^t is the first stock in the titles.

Chapter 3

Conclusion

There has been much research done on the topic of pairs trading in the past, with many focussing on cointegration, stochastic spread and minimum-distance methods. However, most of these stay confined to application onto new data sets and refining the methods used. We focussed our work on the application of wavelets to pairs trading, and its effects on the current methods. By allowing the parameters of α and β to change over time, we compensate for the temporal changes in the relationship between the stock pairs. This has resulted in a generally improved minimum-distance method through the use of costationary solutions, but has not been particularly fruitful in the application to cointegration. This has been the result of, in part, the fact that the algorithm for finding cointegrated pairs is very stringent, allowing very few pairs. On top of this, after finding the pairs, the false positives highly outnumber the amount of truly cointegrated pairs.

It is for that reason, that we advocate for the use of the costationarity method towards improving the minimum distance method, but we also strongly want to iterate that the entire method of costationarity is just another potential tool to add to the trading techniques available. We feel that with extra work in this topic, there may be some significant breakthroughs in the advancement of pairs trading.

3.1 Future Work

1. **Spread Metric**

   Related to the zero-mean limitation above, our spread metric of the differenced stock prices was used simply because it was the most convenient for converting the coefficients back into relatable terms for purchasing and shorting the respective stocks. This is by no means a perfect metric; it may be interesting to compare how other metrics perform in comparison if a way to use the coefficients with log returns is possible.

2. **Training and Testing Period**

   With the training and testing periods we used, they were not rigorously tested to be the most optimal periods for finding cointegrating relationships. However, the issue with this is that finding an "optimal" training period for a set of data will result in the problematic phenomenon known as data snooping. We will be biased towards this set of data, but perhaps there can be something done about the optimal training and testing periods for the costationarity algorithm, with the interest between comparing sample sizes with different powers of two, and different numbers of time varying coefficients allowed per set.

3. **Stock Choices**

   In our work, we tried using the NASDAQ 100 to find minimum-distance and cointegrated pairs. However, there have been studies comparing the stocks from different industries and different countries, which have resulted in cointegrated pairs of stocks or indexes. This may be interesting to look at as the application of the costationarity method to more truly cointegrated pairs may be very interesting to look at. The small sample size of working cointegrated pairs our study has been done on considerably limits the conclusions we can make. The complications begin when comparing returns between different countries or different commodities as the regulations can be vastly different from country to country. Nevertheless, as a pilot study it may be

of interest to do so in the future.

4. **Gaussianity of Innovations in LSW Processes**
   The assumption that the innovations in locally stationary wavelet processes are Gaussian is what drove the test of spectral constancy to be carried out using a parametric bootstrap. This assumption may be invalid in the case of pairs trading, when we inherently believe that the stocks have very similar price paths. Future work could look into expanding this assumption to a less restrictive distribution for the innovations.

5. **Stop-Loss Triggers** The returns obtained from all three methods are not particularly high. In fact, returns are quite poor in several of our tests. Given this fact, it may be beneficial to see if we can limit this in the future through stop-loss triggers at a given point and compare those results to the ones obtained in this thesis.

Appendices

Appendix A

Table of Stocks Used

Stocks Used in Training and Testing											
1	AAPL	2	ADBE	3	ADI	4	ADP	5	ADSK	6	AKAM
7	ALTR	8	ALXN	9	AMAT	10	AMGN	11	AMZN	12	ATVI
13	BBBY	14	BIDU	15	BIIB	16	BRCM	17	CA	18	CELG
19	CERN	20	CHKP	21	CHRW	22	CMCSA	23	COST	24	CSCO
25	CTRX	26	CTSH	27	CTXS	28	DISCA	29	DISH	30	DLTR
31	DTV	32	EBAY	33	EQIX	34	ESRX	35	EXPD	36	EXPE
37	FAST	38	FFIV	39	FISV	40	FOSL	41	FOXA	42	GILD
43	GMCR	44	GOOGL	45	GRMN	46	HSIC	47	INTC	48	INTU
49	ILMN	50	ISRG	51	KLAC	52	LBTYA	53	QVCA	54	LLTC
55	MAR	56	MAT	57	MDLZ	58	MNST	59	MSFT	60	MU
61	MXIM	62	MYL	63	NFLX	64	NTAP	65	NVDA	66	ORLY
67	PAYX	68	PCAR	69	PCLN	70	QCOM	71	REGN	72	ROST
73	SBAC	74	SBUX	75	SIAL	76	SNDK	77	SPLS	78	SRCL
79	STX	80	SYMC	81	TSCO	82	TXN	83	VIAB	84	VIP
85	VOD	86	VRTX	87	WDC	88	WFM	89	WYNN	90	XLNX
91	YHOO										

Table A.1: The 91 stocks used from the NASDAQ 100 that had data points from May 20th, 2009 to May 12th, 2015

Appendix B

R Code

```
##########################################################################
###################-------------------------------#####################
#################          mindist vs costat          ####################
###################-------------------------------#####################
##########################################################################
library(egcm)
library(tseries)
library(quantmod)
library(costat)

##getting stock data
# stocks<-nasdaq_100[,1]
# stocks2<-as.character(stocks)
# #getSymbols(stocks2)
#
# close.price.names<-paste(stocks,"[,6]",sep='')
#
# newstring<-""
# for(i in close.price.names){
#     newstring<-paste(newstring,i,",",sep="")
# }

##########################################################################
###################-------------------------------#####################
#################    getting and cleaning data     ####################
###################-------------------------------#####################
##########################################################################

# close.price<-cbind(AAPL[,6],ADBE[,6],ADI[,6],ADP[,6],ADSK[,6],AKAM[,6],
ALTR[,6],ALXN[,6],AMAT[,6],AMGN[,6],AMZN[,6],ATVI[,6],AVGO[,6],BBBY[,6],
BIDU[,6],BIIB[,6],BRCM[,6],CA[,6],CELG[,6],CERN[,6],CHKP[,6],CHRW[,6],
CHTR[,6],CMCSA[,6],COST[,6],CSCO[,6],CTRX[,6],CTSH[,6],CTXS[,6],DISCA[,6],
DISH[,6],DLTR[,6],DTV[,6],EBAY[,6],EQIX[,6],ESRX[,6],EXPD[,6],EXPE[,6],FAST[,6],
FB[,6],FFIV[,6],FISV[,6],FOSL[,6],FOXA[,6],GILD[,6],GMCR[,6],GOOG[,6],
GOOGL[,6],GRMN[,6],HSIC[,6],INTC[,6],INTU[,6],ILMN[,6],
ISRG[,6],KLAC[,6],KRFT[,6],LBTYA[,6],QVCA[,6],LLTC[,6],LMCA[,6],
```

```
MAR[,6],MAT[,6],MDLZ[,6],MNST[,6],MSFT[,6],MU[,6],MXIM[,6],MYL[,6],
NFLX[,6],NTAP[,6],NVDA[,6],NXPI[,6],ORLY[,6],PAYX[,6],PCAR[,6],PCLN[,6],
QCOM[,6],REGN[,6],ROST[,6],SBAC[,6],SBUX[,6],SIAL[,6],SNDK[,6],SPLS[,6],
SRCL[,6],STX[,6],SYMC[,6],TSCO[,6],TSLA[,6],TRIP[,6],TXN[,6],VIAB[,6],
VIP[,6],VOD[,6],VRSK[,6],VRTX[,6],WDC[,6],WFM[,6],WYNN[,6],XLNX[,6],YHOO[,6])

##cleaning prices by removing NA's
# missing.prices<-rep(0,length(close.price[1,]))
# for(i in 1:length(missing.prices)){
#       temp<-FALSE
#       for(j in close.price[,i]){
#           temp<-temp|is.na(j)
#       }
#       missing.prices[i]<-temp
# }
#
#which(ls()=="test.close.price")
#save(list=ls()[c(42,213)], file = "stockprices.RData")
clean.close.price<-close.price[600:1111,-which(missing.prices==1)]
test.close.price<-close.price[600:2104,-which(missing.prices==1)]

################################################################################
################################################################################

library(egcm)
library(tseries)
library(quantmod)
library(costat)
library(xts)

origdirectory<-#set original directory here
setwd(origdirectory)

##### min dist calculation
clean.close.price<-close.price[600:1111,-which(missing.prices==1)]
test.close.price<-close.price[600:2104,-which(missing.prices==1)]

min.dist.pairs<-function(){
    totalstocks<-length(clean.close.price[1,])
    MSD<-c()
    counter<-0
    for(stock1 in 1:totalstocks){
        if(stock1+1<=totalstocks){
            counter<-stock1+1

            for(stock2 in counter:totalstocks){
                #print(paste(stock1,stock2,sep=" "))
                MSD<-rbind(MSD,c(stock1,stock2,sum((coredata(
                clean.close.price[,stock1])-coredata(clean.close.price[,stock2]))^2)))
            }
        }
    }
    colnames(MSD)<-c("stock1","stock2","distance")
    return(MSD)
}
MSD<-min.dist.pairs()
sorted.dist.pairs<-MSD[order(MSD[,3]),]
minimum.dist.pairs<-head(sorted.dist.pairs,20)
minimum.dist.pairs
```

```
############################
ROImat=matrix(rep(NA,20),nrow=2,ncol=10)
rownames(ROImat)=c("costat_ROI","mindist_ROI")

numtradesmat=matrix(rep(0,20),nrow=2,ncol=10)
rownames(numtradesmat)=c("costat_avg_trades","mindist_Avg_trades")
##########
##########list of ROI/number of trades/average profit matrices
ROIlist=list(stock1=ROImat,stock2=ROImat,stock3=ROImat,stock4=ROImat,
             stock5=ROImat,stock6=ROImat,stock7=ROImat,stock8=ROImat,stock9=ROImat,
             stock10=ROImat,stock11=ROImat,stock12=ROImat)
numtradeslist=list(stock1=numtradesmat,stock2=numtradesmat,stock3=numtradesmat,
stock4=numtradesmat,
                   stock5=numtradesmat,stock6=numtradesmat,stock7=numtradesmat,
                   stock8=numtradesmat,stock9=numtradesmat,stock10=numtradesmat,
                   stock11=numtradesmat,stock12=numtradesmat)

for(q in 1:10){
    directorystock<-paste(origdirectory,"/stock",as.character(q),sep="")
    if(!file.exists(directorystock)){
        dir.create(directorystock)
    }
    setwd(directorystock)

    pairtA<-test.close.price[,sorted.dist.pairs[q,1]]
    pairtB<-test.close.price[,sorted.dist.pairs[q,2]]

    #new type of returns
    sizedatlag<-512
    sizedat<-sizedatlag+1

    ##time.frame for shifting of tests
    time.frame<-matrix(nrow=10,ncol=2)
    start<-1
    trading.period<-50
    time.frame[1,]<-c(start,start+sizedatlag-1)

    for(i in 2:length(time.frame[,1])){
        time.frame[i,]<-time.frame[i-1,]+trading.period
    }
    one.step.time.frame<-time.frame+1
    fullreturnsA<-pairtA-lag(pairtA)
    fullreturnsB<-pairtB-lag(pairtB)

    ######for each test time frame
    for(test.no in 1:length(time.frame[,1])){
        #set to directory for test number
        directoryjk<-paste(directorystock,"/",as.character(test.no),sep="")
        if(!file.exists(directoryjk)){
            dir.create(directoryjk)
        }
        setwd(directoryjk)

        #set data for training
        r0A<-pairtA[time.frame[test.no,1]:time.frame[test.no,2]]
        r0B<-pairtB[time.frame[test.no,1]:time.frame[test.no,2]]

        retA<-fullreturnsA[one.step.time.frame[test.no,1]:one.step.time.frame[test.no,2]]
        retB<-fullreturnsB[one.step.time.frame[test.no,1]:one.step.time.frame[test.no,2]]
```

```
pdf(paste("returns_prices",q,".pdf",sep=""), width=12, height=6)
par(mfcol=c(2,2))
plot(retA,main=paste(colnames(r0A)))
plot(retB,main=colnames(r0B))
plot(r0A,type='l',ylim=c(min(r0A,r0B),max(r0A,r0B)),
main=paste(colnames(r0A),colnames(r0B)))
lines(r0B,col='blue')

model1<-egcm(X=r0A,Y=r0B)
plot(r0B-model1$beta*r0A-model1$alpha,
main=paste("is_cointegrated:",is.cointegrated(model1)),ylab="Residuals")
dev.off()


#BootTOS(retA)
#BootTOS(retB)
#not stationary


#test<-findstysols(Nsims=10,Ncoefs=3,retA,retB)
#saveRDS(test,paste("test",test.no,".rds",sep=""))
test<-readRDS(paste("test",test.no,".rds",sep=""))
par(mfcol=c(2,2))
test$convergence
#plot(test,solno=1)

testlist<-list(test)

n.tests=10
Ncoefs=3
TT = length(test$tsx)
Zmat = matrix(nrow = TT, ncol = n.tests)
alphas<-matrix(nrow=Ncoefs+1,ncol=n.tests)
betas<-matrix(nrow=Ncoefs+1,ncol=n.tests)

for(soln in testlist[]){
    N = length(soln$convergence)
    nosol = soln$convergence == 1 | soln$convergence == 10 | soln$pvals < 0.05
    N2 = N - sum(nosol, na.rm = TRUE)


    if(N2 == 0){
        stop('there_are_no_converging_costationary_solutions')
        #completesol<-seq(1:N)
    }else if (N2==N){
        completesol<-seq(1:N)
    }else{
        completesol<-seq(1:N)[-which(nosol)]
    }
    for(i in completesol){
        alpha<-soln$endpar[i,1:Ncoefs]
        betaseq.start<-Ncoefs+1
        betaseq.end<-Ncoefs*2
        beta<-soln$endpar[i,betaseq.start:betaseq.end]
        coefs<-coeftofn(alpha,beta,TT)

        Zmat[,i]=coefs$alpha*soln$tsx+coefs$beta*soln$tsy
        alphas[,i]<-c(coefs$alpha[1],coefs$alpha[129],coefs$alpha[257],coefs$alpha[385])
```

```
        betas[,i]<-c(coefs$beta[1],coefs$beta[129],coefs$beta[257],coefs$beta[385])

    }

}


alltotalprofit<-c()
allaverageprofit<-c()
alltotalROI<-c()
allnumtrades<-c()
##plotting the test spreads for costat soln, min dist soln, cointegration soln.
for(A in completesol){
    #solnum<-which(test$pvals==max(test$pvals))
    solnum<-A

    alpha<-soln$endpar[solnum,1:Ncoefs]
    betaseq.start<-Ncoefs+1
    betaseq.end<-Ncoefs*2
    beta<-soln$endpar[solnum,betaseq.start:betaseq.end]
    coefs<-coeftofn(alpha,beta,TT)

    newP<-Zmat[,solnum]+coefs$alpha*r0A+coefs$beta*r0B

    section1<-newP[1:128]-mean(newP[1:128])
    section2<-newP[129:256]-mean(newP[129:256])
    section3<-newP[257:384]-mean(newP[257:384])
    section4<-newP[385:512]-mean(newP[385:512])
    totalsect<-c(section1,section2,section3,section4)
    #plot.ts(newP)

    testlength<-50
    teststart<-time.frame[test.no,2]+1
    testend<-teststart+testlength-1
    testp<-coefs$alpha[sizedatlag]
        *pairtA[teststart:testend]+coefs$beta[sizedatlag]*pairtB[teststart:testend]
    sectionlength<-length(test$tsx)/(Ncoefs+1)
    #for calculating the mean of the last alpha coefficient set
    last.set.start<-sizedatlag-sectionlength+1
    last.set<-newP[last.set.start:sizedatlag]
    #print(paste("SOLUTION",A))
    #print(adf.test(last.set))
    ######
    pdf(paste("solution",A,".pdf",sep=""), width=12, height=6)
    par(mfcol=c(2,2))
    plot(c(totalsect,testp-mean(last.set)),
        main=paste("Solution",A,"Mean-Removed Spread"),ylab="Value")
    lines(testp-mean(last.set),col='green')
    abline(v = as.POSIXct(index(newP[128])), col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[256])), col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[384])), col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[512])), col = 'blue', lty = 3, lwd = 2)
    ########################################
    simple.spread<-pairtB[time.frame[test.no,1]:testend]-pairtA[time.frame[test.no,1]:testend]
    mean.ss<-mean(simple.spread)
    sd.ss<-sd(simple.spread)
    ub.ss<-mean.ss+sd.ss
    lb.ss<-mean.ss-sd.ss
    plot(simple.spread,main=paste("Solution",A,'Min-Dist Spread'), ylab="Value")
```

```
            lines(simple.spread[513:562],col='green')

            abline(mean.ss,0,col='red')
            abline(ub.ss,0,col='blue')
            abline(lb.ss,0,col='blue')
            ############################
            #####test set for costat

            plot(c(last.set,testp),main=paste("Solution",A,"CM_Test_Spread"),ylab='Value')
            lines(testp,col='green')

            mean.di<-mean(last.set)
            sd.di<-sd(totalsect)
            ub.di<-mean.di+sd.di
            lb.di<-mean.di-sd.di
            abline(mean.di,0,col='red')
            abline(ub.di,0,col='blue')
            abline(lb.di,0,col='blue')
            #abline(v=513,col='blue')
            ##############################
            ####test set for min dist
            last.set.simple.spread<-simple.spread[last.set.start:length(simple.spread)]
            plot(last.set.simple.spread,main=paste("Solution",A,"MDM_Test_Spread"))
            lines(simple.spread[513:562],col='green')
            abline(mean.ss,0,col='red')
            abline(ub.ss,0,col='blue')
            abline(lb.ss,0,col='blue')

            dev.off()
            profits<-profit_calc()
            alltotalprofit<-c(alltotalprofit,profits$totalprofit)
            allaverageprofit<-c(allaverageprofit,profits$avgprofit)
            alltotalROI<-c(alltotalROI,profits$totalROI)
            allnumtrades<-c(allnumtrades,profits$numtrades)

        }
        #print(paste("stock",q,"test.no",test.no))
        avgROI<-round(100*mean(alltotalROI,na.rm=TRUE),3)
        ROIlist[[q]][1,test.no]<-avgROI
        numtradeslist[[q]][1,test.no]<-mean(allnumtrades,na.rm=TRUE)

        mindist_profits<-mindist_profit_calc()
        md_ROI<-round(100*mindist_profits$totalROI,3)
        md_numtrades<-mindist_profits$numtrades


        ROIlist[[q]][2,test.no]<-md_ROI
        numtradeslist[[q]][2,test.no]<-md_numtrades

        setwd(directorystock)
    }
    setwd(origdirectory)
}

##############################################################################################
##############################################################################################

### profit calculations
profit_calc<-function(){
```

```
spreadL<-length(testp)
tradesopen<-c()
tradesclosed<-c()
trade.position<-"closed"
investedval<-c()
last.profit<-0
#if alpha is -ve, shorting alpha to long the spread, long beta
#if alpha is +ve, shorting beta to long the spread, long alpha
short_alpha<-FALSE
if(coefs$alpha[sizedatlag]>0 & coefs$beta[sizedatlag]<0){
    short_alpha<-FALSE
    dontcount<-FALSE
}else if(coefs$alpha[sizedatlag]<0 & coefs$beta[sizedatlag]>0){
    short_alpha<-TRUE
    dontcount<-FALSE
}else{
    print("ERROR, alpha and betas are not opposite")
    print(paste("q=",q,"test.no=",test.no,"A=",A))
    if(file.exists(paste("solution",A,".pdf",sep=""))){
        file.remove(paste("solution",A,".pdf",sep=""))
    }
    dontcount<-TRUE
}

###function to get the value invested at the time of opening the stock
return_invested_val<-function(long_short_spread){
    testpairA<-pairtA[teststart:testend]
    testpairB<-pairtB[teststart:testend]
    if(long_short_spread=="LONG"){
        #long spread
        if(short_alpha==TRUE){#short alpha, long beta,
            ##alpha is negative, beta is positive
            invest_val<-0.5*abs(coefs$alpha[sizedatlag])
                    *testpairA[k]+coefs$beta[sizedatlag]*testpairB[k]
        }else{#short beta, long alpha, alpha is positive, beta is negative
            invest_val<-coefs$alpha[sizedatlag]
                    *testpairA[k]+0.5*abs(coefs$beta[sizedatlag])*testpairB[k]
        }

    }else if(long_short_spread=="SHORT"){
        #short spread
        if(short_alpha==TRUE){#short beta,
            # long alpha, alpha is negative, beta is positive
            invest_val<-abs(coefs$alpha[sizedatlag])*
                    testpairA[k]+0.5*coefs$beta[sizedatlag]*testpairB[k]
        }else{#short alpha, long beta, alpha is positive, beta is negative
            invest_val<-0.5*coefs$alpha[sizedatlag]*
                    testpairA[k]+abs(coefs$beta[sizedatlag])*testpairB[k]
        }
    }
    return(invest_val)
}

invest_val<-0
for(k in 1:spreadL){
    #print(paste(trade.position,testp[k], "mean.di=",mean.di, "k=",k))
    #trade is closed
    if(trade.position=="closed"){
        if(testp[k]>ub.di){#if spread is greater than the upper bound,short spread
```

```
            #open trade
            #record price
            #set status to be above mean
            spread.position<-"SHORT"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"above_mean"
            #store value invested
            invest_val<-return_invested_val(spread.position)
            investedval<-rbind(investedval,invest_val)
        }else if(testp[k]<lb.di){# if spread is less than the lower bound,long spread
            spread.position<-"LONG"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"below_mean"
            #store value invested
            invest_val<-return_invested_val(spread.position)
            investedval<-rbind(investedval,invest_val)
        }
    }else{#trade is open
        if(original.position=="above_mean"){#position above mean
            if(testp[k]<=mean.di){#price is below mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }

        }else{#boolean is below mean
            if(testp[k]>=mean.di){#price is above mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }
        }
    }

}
if(trade.position=="open"){
    tradesclosed<-rbind(tradesclosed,testp[k])
}

if(length(tradesopen)>=1){
    if(trade.position=="open"){
        if(original.position=="below_mean"){
            last.profit<-coredata(tradesclosed)-coredata(tradesopen)
            last.profit<-last.profit[length(last.profit)]
        }else{
            last.profit<-coredata(tradesopen)-coredata(tradesclosed)
            last.profit<-last.profit[length(last.profit)]
        }
    }

    if(length(tradesopen)!=1){
```

```
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
            profitlength<-length(profit)-1
            profit<-c(profit[1:profitlength],last.profit)
        }else if(length(tradesopen==1)&trade.position=="closed"){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
        }else{
            profit<-last.profit
        }

        #number of trades
        numtrades<-length(profit)

        #profit stats for one solution
        totalprofit<-sum(profit)
        averageprofit<-totalprofit/numtrades
        totalROI<-sum(profit/investedval)

        if(dontcount==TRUE){
            return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,numtrades=NA))
        }else{
            return(list(totalprofit=totalprofit,
                avgprofit=averageprofit,totalROI=totalROI,numtrades=numtrades,
                tradesopen=tradesopen,tradesclosed=tradesclosed))
        }
    }else{
        #print("no trades executed")
        #print(paste("test.no =",test.no,"A =",A))
        return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,numtrades=NA))
    }

}

#min_dist profit_calc
mindist_profit_calc<-function(){
    simple.spread<-pairtB[time.frame[test.no,1]:testend]-pairtA[time.frame[test.no,1]:testend]
    Bval<-pairtB[time.frame[test.no,1]:testend]
    Aval<-pairtA[time.frame[test.no,1]:testend]
    #spread is long B, short A
    mean.ss<-mean(simple.spread)
    sd.ss<-sd(simple.spread)
    ub.ss<-mean.ss+sd.ss
    lb.ss<-mean.ss-sd.ss
    testp<-simple.spread[513:562]


    spreadL<-length(testp)
    tradesopen<-c()
    tradesclosed<-c()
    trade.position<-"closed"
    investedval<-c()

    invest_val<-0
    last.profit<-0
    for(k in 1:spreadL){
        #print(paste(trade.position,testp[k], "mean.di=",mean.di, "k=",k))
        #trade is closed
        if(trade.position=="closed"){
            if(testp[k]>ub.ss){#if spread is greater than the upper bound,short spread
                #open trade
```

```
            #record price
            #set status to be above mean
            spread.position<-"SHORT"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"above_mean"
            #store value invested, short spread: long A, short B
            invest_val<-0.5*Bval[k]+Aval[k]
            investedval<-rbind(investedval,invest_val)
        }else if(testp[k]<lb.ss){# if spread is less than the lower bound,long spread
            spread.position<-"LONG"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"below_mean"
            #store value invested
            invest_val<-0.5*Aval[k]+Bval[k]
            investedval<-rbind(investedval,invest_val)
        }
    }else{#trade is open
        if(original.position=="above_mean"){#position above mean
            if(testp[k]<=mean.ss){#price is below mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }

        }else{#boolean is below mean
            if(testp[k]>=mean.ss){#price is above mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }
        }
    }

}
if(trade.position=="open"){
    tradesclosed<-rbind(tradesclosed,testp[k])
}

if(length(tradesopen)>=1){

    if(trade.position=="open"){
        if(original.position=="below_mean"){
            last.profit<-coredata(tradesclosed)-coredata(tradesopen)
            last.profit<-last.profit[length(last.profit)]
        }else{
            last.profit<-coredata(tradesopen)-coredata(tradesclosed)
            last.profit<-last.profit[length(last.profit)]
        }
    }

    if(length(tradesopen)!=1){
```

```
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
            profitlength<-length(profit)-1
            profit<-c(profit[1:profitlength],last.profit)
        }else if(length(tradesopen==1)&trade.position=="closed"){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
        }else{
            profit<-last.profit
        }

        #number of trades
        numtrades<-length(profit)

        #profit stats for one solution
        totalprofit<-sum(profit)
        averageprofit<-totalprofit/numtrades
        totalROI<-sum(profit/investedval)
        return(list(totalprofit=totalprofit,avgprofit=averageprofit,totalROI=totalROI,
                numtrades=numtrades,tradesopen=tradesopen,tradesclosed=tradesclosed))

    }else{
        #("no trades executed")
        return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,
                numtrades=NA,tradesopen=NA,tradesclosed=NA))
    }

}

#########################################################################
###################-----------------------------#######################
##################    cointegration vs costat     ######################
###################-----------------------------#######################
#########################################################################
library(egcm)
library(tseries)
library(quantmod)
library(costat)

cointegrated.pairs<-function(){
    total.stocks<-length(clean.close.price[1,])
    counter<-0
    cointegrated<-c()

    for(stock1 in 1:totalstocks){
        if(stock1+1<=totalstocks){
            counter<-stock1+1

            for(stock2 in counter:totalstocks){

                x<-coredata(clean.close.price[,stock1])
                y<-coredata(clean.close.price[,stock2])
                model<-egcm(x,y,i1test="adf",urtest="adf",
                        p.value=0.03)
                model2<-egcm(x,y,i1test="adf",urtest="jo-e",
                        p.value=0.03)
                cointegrated.bool<-
                is.cointegrated(model)&is.cointegrated(model2)
                cointegrated<-
                        rbind(cointegrated,c(stock1,stock2,cointegrated.bool))
                print(paste(stock1,stock2))
```

```
            }
        }
    }
    cointegrated.pairs<-cointegrated[which(cointegrated[,3]==1),]
    return(cointegrated.pairs)
}

clean.close.price<-close.price[846:1357,-which(missing.prices==1)]
test.close.price<-close.price[846:2104,-which(missing.prices==1)]
costat_coint<-cointegrated.pairs()

clean.close.price<-close.price[600:1357,-which(missing.prices==1)]
test.close.price<-close.price[600:2104,-which(missing.prices==1)]
#threeyearmat<-cointegrated.pairs()

threeyearmat
costat_coint

##########################################################################
##########################################################################

library(egcm)
library(tseries)
library(quantmod)
library(costat)

origdirectory<-#set top level directory here
setwd(origdirectory)

clean.close.price<-close.price[846:1357,-which(missing.prices==1)]
test.close.price<-close.price[846:2104,-which(missing.prices==1)]

############################
ROImat=matrix(rep(NA,20),nrow=2,ncol=10)
rownames(ROImat)=c("costat_ROI","mindist_ROI")

numtradesmat=matrix(rep(0,20),nrow=2,ncol=10)
rownames(numtradesmat)=c("costat_avg_trades","mindist_Avg_trades")
##########
##########list of ROI/number of trades/average profit matrices
ROIlist=list(stock1=ROImat,stock2=ROImat,stock3=ROImat,stock4=ROImat,
             stock5=ROImat,stock6=ROImat,stock7=ROImat,stock8=ROImat,
             stock9=ROImat,stock10=ROImat,stock11=ROImat,
             stock12=ROImat,stock13=ROImat)
numtradeslist=list(stock1=numtradesmat,stock2=numtradesmat,
stock3=numtradesmat,stock4=numtradesmat,
                   stock5=numtradesmat,stock6=numtradesmat,
                   stock7=numtradesmat,stock8=numtradesmat,
                   stock9=numtradesmat,stock10=numtradesmat,
                   stock11=numtradesmat,stock12=numtradesmat,
                   stock13=numtradesmat)

for(q in c(2,3,9,11)){
    directorystock<-paste(origdirectory,"/stock",
        as.character(q),sep="")
    if(!file.exists(directorystock)){
        dir.create(directorystock)
    }
    setwd(directorystock)
```

```
pairtA<-test.close.price[,costat_coint[q,1]]
pairtB<-test.close.price[,costat_coint[q,2]]

#new type of returns
sizedatlag<-512
sizedat<-sizedatlag+1

##time.frame for shifting of tests
time.frame<-matrix(nrow=10,ncol=2)
start<-1
trading.period<-50
time.frame[1,]<-c(start,start+sizedatlag-1)

for(i in 2:length(time.frame[,1])){
    time.frame[i,]<-time.frame[i-1,]+trading.period
}
one.step.time.frame<-time.frame+1

#fullreturnsA<-pairtA-lag(pairtA)
fullreturnsB<-pairtB-lag(pairtB)

######for each test time frame
for(test.no in 1:5){
    #set to directory for test number
    directoryjk<-paste(directorystock,"/",
            as.character(test.no),sep="")
    if(!file.exists(directoryjk)){
        dir.create(directoryjk)
    }
    setwd(directoryjk)

    #set data for training
    r0A<-pairtA[time.frame[test.no,1]:time.frame[test.no,2]]
    r0B<-pairtB[time.frame[test.no,1]:time.frame[test.no,2]]

    #recalibrate model
    model1<-egcm(X=r0A,Y=r0B)
    pairA_trajectory<-model1$alpha+model1$beta*pairtA
    fullreturnsA<-pairA_trajectory-lag(pairA_trajectory)

    #set data for training
    r0A<-pairA_trajectory[time.frame[test.no,1]:time.frame[test.no,2]]
    r0B<-pairtB[time.frame[test.no,1]:time.frame[test.no,2]]

    retA<-fullreturnsA[one.step.time.frame[test.no,1]:one.step.time.frame[test.no,2]]
    retB<-fullreturnsB[one.step.time.frame[test.no,1]:one.step.time.frame[test.no,2]]

    pdf(paste("returns_prices",q,".pdf",sep=""),width=12, height=6)
    par(mfcol=c(2,2))
    plot(retA,main=paste("pair",q))
    plot(retB,main=paste(time.frame[test.no,1],":",time.frame[test.no,2]))
    plot(r0A,type='l',ylim=c(min(r0A,r0B),max(r0A,r0B)),
    main=paste(colnames(r0A),colnames(r0B)))
    lines(r0B,col='blue')

    plot(r0B-r0A,main=paste("is_cointegrated:",is.cointegrated(model1)) )
    dev.off()
```

```
#BootTOS(retA)
#BootTOS(retB)
#not stationary


#test<-findstysols(Nsims=10,Ncoefs=3,retA,retB)
#saveRDS(test,paste("test",test.no,".rds",sep=""))
test<-readRDS(paste("test",test.no,".rds",sep=""))
par(mfcol=c(2,2))
test$convergence
#plot(test,solno=1)

testlist<-list(test)

n.tests=10
Ncoefs=3
TT = length(test$tsx)
Zmat = matrix(nrow = TT, ncol = n.tests)
alphas<-matrix(nrow=Ncoefs+1,ncol=n.tests)
betas<-matrix(nrow=Ncoefs+1,ncol=n.tests)

for(soln in testlist[]){
    N = length(soln$convergence)
    nosol = soln$convergence == 1 |
    soln$convergence == 10 | soln$pvals < 0.05
    N2 = N - sum(nosol, na.rm = TRUE)


    if(N2 == 0){
        stop('there are no converging costationary solutions')
        #completesol<-seq(1:N)
    }else if (N2==N){
        completesol<-seq(1:N)
    }else{
        completesol<-seq(1:N)[-which(nosol)]
    }
    for(i in completesol){
        alpha<-soln$endpar[i,1:Ncoefs]
        betaseq.start<-Ncoefs+1
        betaseq.end<-Ncoefs*2
        beta<-soln$endpar[i,betaseq.start:betaseq.end]
        coefs<-coeftofn(alpha,beta,TT)

        Zmat[,i]=coefs$alpha*soln$tsx+coefs$beta*soln$tsy
        alphas[,i]<-c(coefs$alpha[1],
                coefs$alpha[129],coefs$alpha[257],coefs$alpha[385])
        betas[,i]<-c(coefs$beta[1],coefs$beta[129],
                coefs$beta[257],coefs$beta[385])

    }

}


alltotalprofit<-c()
allaverageprofit<-c()
alltotalROI<-c()
allnumtrades<-c()
```

```
##plotting the test spreads for costat soln,
##min dist soln, cointegration soln.
for(A in completesol){
    #solnum<-which(test$pvals==max(test$pvals))
    solnum<-A

    alpha<-soln$endpar[solnum,1:Ncoefs]
    betaseq.start<-Ncoefs+1
    betaseq.end<-Ncoefs*2
    beta<-soln$endpar[solnum,betaseq.start:betaseq.end]
    coefs<-coeftofn(alpha,beta,TT)

    newP<-Zmat[,solnum]+coefs$alpha*r0A+coefs$beta*r0B
    section1<-newP[1:128]-mean(newP[1:128])
    section2<-newP[129:256]-mean(newP[129:256])
    section3<-newP[257:384]-mean(newP[257:384])
    section4<-newP[385:512]-mean(newP[385:512])
    totalsect<-c(section1,section2,section3,section4)
    #plot.ts(newP)


    testlength<-50
    teststart<-time.frame[test.no,2]+1
    testend<-teststart+testlength-1
    testp<-coefs$alpha[sizedatlag]*
            (model1$beta*pairtA[teststart:testend]+model1$alpha)
        +coefs$beta[sizedatlag]*pairtB[teststart:testend]
    sectionlength<-length(test$tsx)/(Ncoefs+1)

    #for calculating the mean of the last alpha coefficient set
    last.set.start<-sizedatlag-sectionlength+1
    last.set<-newP[last.set.start:sizedatlag]
    #print(paste("SOLUTION",A))
    #print(adf.test(last.set))
    ######
    pdf(paste("solution",A,".pdf",sep=""), width=12, height=6)
    par(mfcol=c(2,2))
    plot(c(totalsect,testp-mean(last.set)),
        main=paste("Solution",A,"Mean-Removed Spread"),ylab="Value")
    lines(testp-mean(last.set),col='green')
    abline(v = as.POSIXct(index(newP[128])),
        col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[256])),
        col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[384])),
         col = 'blue', lty = 3, lwd = 2)
    abline(v = as.POSIXct(index(newP[512])),
         col = 'blue', lty = 3, lwd = 2)

    #cointegration.spread<-pairtB[teststart:testend]
        -model1$beta*pairtA[teststart:testend]-model1$alpha
    cointegration.spread<-pairtB[teststart:testend]
        -pairA_trajectory[teststart:testend]
    mean.cs<-mean(model1$residuals)
    sd.cs<-sd(model1$residuals)
    ub.cs<-mean.cs+sd.cs
    lb.cs<-mean.cs-sd.cs
    coint.train.spread<-r0B-r0A
    total.cointegration.spread<-rbind(coint.train.spread,
```

```
            cointegration.spread)
        plot(total.cointegration.spread,main=paste("Solution",
            A,'Cointegration Spread'))
        lines(cointegration.spread,col='green')
        #abline(v=513,col='blue')
        abline(mean.cs,0,col='red')
        abline(ub.cs,0,col='blue')
        abline(lb.cs,0,col='blue')
        #########
        plot(c(last.set,testp),main=paste("Solution",
            A,"CM Test Spread"),ylab='Value')
        lines(testp,col='green')

        mean.di<-mean(last.set)
        sd.di<-sd(totalsect)
        ub.di<-mean.di+sd.di
        lb.di<-mean.di-sd.di
        abline(mean.di,0,col='red')
        abline(ub.di,0,col='blue')
        abline(lb.di,0,col='blue')
        ####test set for cointegration
        simple.spread<-pairtB[time.frame[test.no,1]:testend]
            -pairtA[time.frame[test.no,1]:testend]
        last.set.simple.spread<-total.cointegration.spread
            [last.set.start:length(simple.spread)]
        plot(last.set.simple.spread,
            main=paste("Solution",A,"CIM Test Spread"))
        lines(total.cointegration.spread[513:562],col='green')
        abline(mean.cs,0,col='red')
        abline(ub.cs,0,col='blue')
        abline(lb.cs,0,col='blue')

        dev.off()
        profits<-profit_calc()
        #print(profits)
        alltotalprofit<-c(alltotalprofit,profits$totalprofit)
        allaverageprofit<-c(allaverageprofit,profits$avgprofit)
        alltotalROI<-c(alltotalROI,profits$totalROI)
        allnumtrades<-c(allnumtrades,profits$numtrades)

    }
    #print(paste("stock",q,"test.no",test.no))
    avgROI<-round(100*mean(alltotalROI,na.rm=TRUE),3)
    ROIlist[[q]][1,test.no]<-avgROI
    numtradeslist[[q]][1,test.no]<-mean(allnumtrades,na.rm=TRUE)

    coint_profits<-coint_profit_calc()
    cd_ROI<-round(100*coint_profits$totalROI,3)
    cd_numtrades<-coint_profits$numtrades


    ROIlist[[q]][2,test.no]<-cd_ROI
    numtradeslist[[q]][2,test.no]<-cd_numtrades

    setwd(directorystock)
  }
  setwd(origdirectory)
}
```

```
#####################################################################
#####################################################################
### profit calculations
profit_calc<-function(){
    spreadL<-length(testp)
    tradesopen<-c()
    tradesclosed<-c()
    trade.position<-"closed"
    investedval<-c()
    last.profit<-0
    #if alpha is -ve, shorting alpha to long the spread, long beta
    #if alpha is +ve, shorting beta to long the spread, long alpha
    short_alpha<-FALSE
    if(coefs$alpha[sizedatlag]>0 & coefs$beta[sizedatlag]<0){
        short_alpha<-FALSE
        dontcount<-FALSE
    }else if(coefs$alpha[sizedatlag]<0 & coefs$beta[sizedatlag]>0){
        short_alpha<-TRUE
        dontcount<-FALSE
    }else{
        print("ERROR, alpha and betas are not opposite")
        print(paste("q=",q,"test.no=",test.no,"A=",A))
        if(file.exists(paste("solution",A,".pdf",sep=""))){
            file.remove(paste("solution",A,".pdf",sep=""))
        }
        dontcount<-TRUE
    }

    ###function to get the value invested at the time of
    ##opening the stock
    return_invested_val<-function(long_short_spread){
        testpairA<-pairA_trajectory[teststart:testend]
        testpairB<-pairtB[teststart:testend]
        if(long_short_spread=="LONG"){
            #long spread
            if(short_alpha==TRUE){
            #short alpha, long beta, alpha is negative, beta is positive
                invest_val<-0.5*abs(coefs$alpha[sizedatlag])
                        *testpairA[k]+coefs$beta[sizedatlag]*testpairB[k]
            }else{#short beta, long alpha, alpha is positive, beta is negative
                invest_val<-coefs$alpha[sizedatlag]*testpairA[k]
                        +0.5*abs(coefs$beta[sizedatlag])*testpairB[k]
            }

        }else if(long_short_spread=="SHORT"){
            #short spread
            if(short_alpha==TRUE){
            #short beta, long alpha, alpha is negative, beta is positive
                invest_val<-abs(coefs$alpha[sizedatlag])*testpairA[k]
                        +0.5*coefs$beta[sizedatlag]*testpairB[k]
            }else{#short alpha, long beta, alpha is positive,beta is negative
                invest_val<-0.5*coefs$alpha[sizedatlag]*testpairA[k]
                        +abs(coefs$beta[sizedatlag])*testpairB[k]
            }
        }
        return(invest_val)
    }

    invest_val<-0
```

```
for(k in 1:spreadL){
    #print(paste(trade.position,testp[k], "mean.di=",mean.di, "k=",k))
    #trade is closed
    if(trade.position=="closed"){
        if(testp[k]>ub.di){#if spread is greater than the upper bound,short spread
            #open trade
            #record price
            #set status to be above mean
            spread.position<-"SHORT"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"above_mean"
            #store value invested
            invest_val<-return_invested_val(spread.position)
            investedval<-rbind(investedval,invest_val)
        }else if(testp[k]<lb.di){# if spread is less than the lower bound,long spread
            spread.position<-"LONG"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"below_mean"
            #store value invested
            invest_val<-return_invested_val(spread.position)
            investedval<-rbind(investedval,invest_val)
        }
    }else{#trade is open
        if(original.position=="above_mean"){#position above mean
            if(testp[k]<=mean.di){#price is below mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }

        }else{#boolean is below mean
            if(testp[k]>=mean.di){#price is above mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }
        }
    }

}
if(trade.position=="open"){
    tradesclosed<-rbind(tradesclosed,testp[k])
}

if(length(tradesopen)>=1){
    if(trade.position=="open"){
        if(original.position=="below_mean"){
            last.profit<-coredata(tradesclosed)-coredata(tradesopen)
            last.profit<-last.profit[length(last.profit)]
        }else{
            last.profit<-coredata(tradesopen)-coredata(tradesclosed)
```

```
                last.profit<-last.profit[length(last.profit)]
            }
        }

        if(length(tradesopen)!=1){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
            profitlength<-length(profit)-1
            profit<-c(profit[1:profitlength],last.profit)
        }else if(length(tradesopen==1)&trade.position=="closed"){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
        }else{
            profit<-last.profit
        }

        #number of trades
        numtrades<-length(profit)

        #profit stats for one solution
        totalprofit<-sum(profit)
        averageprofit<-totalprofit/numtrades
        totalROI<-sum(profit/investedval)

        if(dontcount==TRUE){
            return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,numtrades=NA))
        }else{
            return(list(totalprofit=totalprofit,avgprofit=averageprofit,
            totalROI=totalROI,numtrades=numtrades,tradesopen=tradesopen,
            tradesclosed=tradesclosed))
        }
    }else{
        #print("no trades executed")
        #print(paste("test.no =",test.no,"A =",A))
        return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,numtrades=NA))
    }

}

#min_dist profit_calc
mindist_profit_calc<-function(){
    simple.spread<-pairtB[time.frame[test.no,1]:testend]-pairtA[time.frame[test.no,1]:testend]
    Bval<-pairtB[time.frame[test.no,1]:testend]
    Aval<-pairtA[time.frame[test.no,1]:testend]
    #spread is long B, short A
    mean.ss<-mean(simple.spread)
    sd.ss<-sd(simple.spread)
    ub.ss<-mean.ss+sd.ss
    lb.ss<-mean.ss-sd.ss
    testp<-simple.spread[513:562]


    spreadL<-length(testp)
    tradesopen<-c()
    tradesclosed<-c()
    trade.position<-"closed"
    investedval<-c()

    invest_val<-0
    last.profit<-0
    for(k in 1:spreadL){
```

```
    #print(paste(trade.position,testp[k], "mean.di=",mean.di, "k=",k))
    #trade is closed
    if(trade.position=="closed"){
        if(testp[k]>ub.ss){#if spread is greater than the upper bound,short spread
            #open trade
            #record price
            #set status to be above mean
            spread.position<-"SHORT"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"above_mean"
            #store value invested, short spread: long A, short B
            invest_val<-0.5*Bval[k]+Aval[k]
            investedval<-rbind(investedval,invest_val)
        }else if(testp[k]<lb.ss){# if spread is less than the lower bound,long spread
            spread.position<-"LONG"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"below_mean"
            #store value invested
            invest_val<-0.5*Aval[k]+Bval[k]
            investedval<-rbind(investedval,invest_val)
        }
    }else{#trade is open
        if(original.position=="above_mean"){#position above mean
            if(testp[k]<=mean.ss){#price is below mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }

        }else{#boolean is below mean
            if(testp[k]>=mean.ss){#price is above mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }
        }
    }

}
if(trade.position=="open"){
    tradesclosed<-rbind(tradesclosed,testp[k])
}

if(length(tradesopen)>=1){

    if(trade.position=="open"){
        if(original.position=="below_mean"){
            last.profit<-coredata(tradesclosed)-coredata(tradesopen)
            last.profit<-last.profit[length(last.profit)]
        }else{
            last.profit<-coredata(tradesopen)-coredata(tradesclosed)
```

```
                last.profit<-last.profit[length(last.profit)]
            }
        }

        if(length(tradesopen)!=1){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
            profitlength<-length(profit)-1
            profit<-c(profit[1:profitlength],last.profit)
        }else if(length(tradesopen==1)&trade.position=="closed"){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
        }else{
            profit<-last.profit
        }

        #number of trades
        numtrades<-length(profit)

        #profit stats for one solution
        totalprofit<-sum(profit)
        averageprofit<-totalprofit/numtrades
        totalROI<-sum(profit/investedval)
        return(list(totalprofit=totalprofit,avgprofit=averageprofit,
                totalROI=totalROI,numtrades=numtrades,tradesopen=tradesopen,
                tradesclosed=tradesclosed))

    }else{
        #("no trades executed")
        return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,numtrades=NA,
                tradesopen=NA,tradesclosed=NA))
    }

}

#cointegration profit_calc
coint_profit_calc<-function(){
    r0A<-pairtA[time.frame[test.no,1]:time.frame[test.no,2]]
    r0B<-pairtB[time.frame[test.no,1]:time.frame[test.no,2]]
    model1<-egcm(X=r0A,Y=r0B)
    coint.train.spread<-r0B-model1$alpha-model1$beta*r0A
    cointegration.spread<-pairtB[teststart:testend]-model1$beta*pairtA[teststart:testend]
        -model1$alpha

#     head(r0B-pairA_trajectory[time.frame[test.no,1]:time.frame[test.no,2]])

    Bval<-pairtB[time.frame[test.no,1]:testend]
    Aval<-model1$beta*pairtA[time.frame[test.no,1]:testend]+model1$alpha
    #spread is long B, short A
    mean.ss<-mean(coint.train.spread)
    sd.ss<-sd(coint.train.spread)
    ub.ss<-mean.ss+sd.ss
    lb.ss<-mean.ss-sd.ss
    testp<-cointegration.spread


    spreadL<-length(testp)
    tradesopen<-c()
    tradesclosed<-c()
    trade.position<-"closed"
    investedval<-c()
```

```
invest_val<-0
last.profit<-0
for(k in 1:spreadL){
    #print(paste(trade.position,testp[k], "mean.di=",mean.di, "k=",k))
    #trade is closed
    if(trade.position=="closed"){
        if(testp[k]>ub.ss){#if spread is greater than the upper bound,short spread
            #open trade
            #record price
            #set status to be above mean
            spread.position<-"SHORT"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"above_mean"
            #store value invested, short spread: long A, short B
            invest_val<-0.5*Bval[k]+Aval[k]
            investedval<-rbind(investedval,invest_val)
        }else if(testp[k]<lb.ss){# if spread is less than the lower bound,long spread
            spread.position<-"LONG"
            trade.position<-"open"
            tradesopen<-rbind(tradesopen,testp[k])
            original.position<-"below_mean"
            #store value invested
            invest_val<-0.5*Aval[k]+Bval[k]
            investedval<-rbind(investedval,invest_val)
        }
    }else{#trade is open
        if(original.position=="above_mean"){#position above mean
            if(testp[k]<=mean.ss){#price is below mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }

        }else{#boolean is below mean
            if(testp[k]>=mean.ss){#price is above mean
                trade.position="closed"
                tradesclosed<-rbind(tradesclosed,testp[k])
                original.position<-"NA"
                #close trade
                #record price
                #set status to be neutral
            }
        }
    }

}
if(trade.position=="open"){
    tradesclosed<-rbind(tradesclosed,testp[k])
}

if(length(tradesopen)>=1){

    if(trade.position=="open"){
        if(original.position=="below_mean"){
```

```
                last.profit<-coredata(tradesclosed)-coredata(tradesopen)
                last.profit<-last.profit[length(last.profit)]
            }else{
                last.profit<-coredata(tradesopen)-coredata(tradesclosed)
                last.profit<-last.profit[length(last.profit)]
            }
        }

        if(length(tradesopen)!=1){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
            profitlength<-length(profit)-1
            profit<-c(profit[1:profitlength],last.profit)
        }else  if(length(tradesopen==1)&trade.position=="closed"){
            profit<-abs(coredata(tradesclosed)-coredata(tradesopen))
        }else{
            profit<-last.profit
        }

        #number of trades
        numtrades<-length(profit)

        #profit stats for one solution
        totalprofit<-sum(profit)
        averageprofit<-totalprofit/numtrades
        totalROI<-sum(profit/investedval)
        return(list(totalprofit=totalprofit,avgprofit=averageprofit,
        totalROI=totalROI,numtrades=numtrades,tradesopen=tradesopen,tradesclosed=tradesclosed))

    }else{
        #("no trades executed")
        return(list(totalprofit=NA,avgprofit=NA,totalROI=NA,
                numtrades=NA,tradesopen=NA,tradesclosed=NA))
    }

}



#####plot of profit for our strategy
plot(c(last.set,testp),main=paste("solution",A,"pval=",test$pvals[solnum]))
lines(testp,col="green")

mean.di<-mean(last.set)
sd.di<-sd(totalsect)
ub.di<-mean.di+sd.di
lb.di<-mean.di-sd.di
abline(mean.di,0,col='red')
abline(ub.di,0,col='blue')
abline(lb.di,0,col='blue')
points(tradesopen,pch=18,col='black')
points(tradesclosed,pch=18,col='red')
##########
#plot of min dist profit
simple.spread<-pairtB[time.frame[test.no,1]:testend]-pairtA[time.frame[test.no,1]:testend]
mean.ss<-mean(simple.spread)
sd.ss<-sd(simple.spread)
ub.ss<-mean.ss+sd.ss
lb.ss<-mean.ss-sd.ss
plot(simple.spread,main='simple spread')
```

```
lines(simple.spread[513:562],col='red')

abline(mean.ss,0)
abline(ub.ss,0,col='red')
abline(lb.ss,0,col='red')
points(tradesopen,pch=18,col='black')
points(tradesclosed,pch=18,col='red')
##########


profits<-profit_calc()

#average profit stats for all solutions
alltotalprofit<-c(alltotalprofit,profits$totalprofit)
allaverageprofit<-c(allaverageprofit,profits$avgprofit)
alltotalROI<-c(alltotalROI,profits$totalROI)

###
mean(alltotalprofit)
mean(allaverageprofit)
mean(alltotalROI)

alltotalprofit<-c()
allaverageprofit<-c()
alltotalROI<-c()
```

References

Alexander, C. and Dimitriu, A. (2002). The cointegartion alpha: Enhanced index tracking and long-short equity market neutral strategies. Discussion paper, Finance ISMA Center, University of Reading, UK.

Cardinali, A. and Nason, G. P. (2011). Costationarity of locally stationary time series. *Journal of Time Series Econometrics*, 2:1–35.

Cardinali, A. and Nason, G. P. (2013). Costationarity of locally stationary time series using costat. *Journal of Statistical Software*, 55.

Chen, C. W. S., Chen, M., and Chen, S.-Y. (2014). Pairs trading via three-regime threshold autoregressive garch models. In *Modelling Dependence in Econometrics*, pages 127–140. Springer International Publishing.

Dahlhaus, R. (1997). Fitting time series models to nonstationary processes. *The Annals of Statistics*, 25:1–37.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996.

Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM.

Dickey, D. and Fuller, W. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74:427–431.

Dickey, D. and Fuller, W. (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, 49:1057–1072.

Do, B. and Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal*, 66:83–95.

Do, B. and Faff, R. (2012). Are pairs trading profits robust to trading costs? *Journal of Financial Research*, 35(2):261–287.

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. P. (2005). Pairs trading. *Quantitative Finance*, 5:271–276.

Enders, W. (2003). *Applied Econometric Time Series*. John Wiley and Sons, Incorporated.

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

Fung, W. and Hsieh, D. A. (1999). A primer on hedge funds. *Journal of Empirical Finance*, 6:309–331.

Gatev, E., Goetzmann, W. N., and Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827.

Haykin, S. S. (1983). *Communication Systems*. John Wiley and Sons Inc.

Hull, J. (2009). *Options, futures and other derivatives*. Pearson Prentice Hall.

Jacobs, B. I. and Levy, K. (1993). Long-short equity investing. *Journal of Portfolio Management*, 1:52–64.

Johansen, S. (1988). Statistical analysis of cointegrating vectors. *Journal of Economic Dynamics and Control*, 12:231–254.

Lin, Y.-X., McCrae, M., and Culati, C. (2006). Loss protection in pairs trading through minimum profit bounds: A cointegration approach. *Journal of Applied Mathematics and Decision Sciences*, pages 1–14.

MacKinnon, J. (1990). Critical values for cointegration tests. Working paper, Queen's University.

Nason, G. P. (2008). *Wavelet Methods in Statistics with R*. Springer-Verlag New York.

Nason, G. P., Von Sachs, R., and Kroisandt, G. (2000). Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62:271–292.

Nath, P. (2003). High frequency pairs trading with us treasury securities: Risks and rewards for hedge funds. Working paper, London Business School.

Phillips, P. and Ouliaris, S. (1990). Asymptotic properties of residual based tests for cointegration. *Econometrica*, 58:165–193.

Priestley, M. B. (1983). *Spectral Analysis and Time Series*. Academic Press Inc.

Ross, S., Hillier, D., Westerfield, R., Jaffe, J., and Jordan, B. (2013). *Corporate Finance*. Mcgraw-Hill.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13:341–360.

Ross, S. A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71:599–607.

Shumway, R. and Stoffer, D. (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3(4):253–264.

Stock, J. H. (1987). Asymptotic properties of least squares estimators of cointegrating vectors. *Econometrica*, 55:277–302.

Vidyamurthy, G. (2004). *Pairs Trading: Quantitative Methods and Analysis*. New York: John Wiley.