Università degli Studi di Roma "Tor Vergata"

School of Economics

Master of Science in Finance and Banking

Thesis in Financial Econometrics

"State-Space Models for Pairs Trading: Kalman Filter Approach"

The Supervisor
Prof. Tommaso Proietti

The Student
Carlo Cascini

Academic year 2023/2024

To my family

Contents

1 Introduction

Pairs trading is a market-neutral strategy, in the category of statistical arbitrage (SA), known by financial community since 1980s. The strategy involves identifying two securities whose prices tend to move together. When their relative difference move away, the cheaper security is bought long, and the more expensive one is sold short. When the prices come back to their historical level, the positions are unwind, and profits are collected. SA strategies, as pairs trading, rely on mathematical models to identify and exploit inefficiencies in the market that may not be captured by human traders.

However, this strategy should not be intended in the context of retail traders, since it poses serious execution challenges, and involves effort in model development and maintenance. In addition, successful pairs trading requires continuous monitoring of the price relationship between assets, as well as minimizing risks associated with short selling, trading costs, and breakdowns in historical relations between pairs over time. Another important factor to take into account is the requirement for a large amount of capital. This approach typically includes handling a variety of pairs at the same time, adjusting frequently to take advantage of slight price discrepancies. A high amount of liquidity in this case is fundamental to guarantee that transaction costs will be absorbed and necessary margins will be covered, also in drawdown periods. As a result, pairs trading is best suited for institutional investors and hedge funds due to its high entrance barriers.

1.1 Traditional Methodologies

In the past few years, pairs trading had a substantial development, with many new advancements. Typically, these methods consist of three fundamental stages:

1. **Identification of pairs**. This step consists in identify security that tend to have a deep

relation.

2. **Modeling the Spread**. Once the pairs are identified, the next step is to build a model for their relative difference. The goal here is to maximize mean reversion. For instance in a "naive approach" this could be just the difference or the ratio between the prices.

3. **Establishing Trading Rules**. The final step involves creating specific trading rules based on the modeled spread, that fills automatilly orders when the signal are triggered.

Krauss (2015) recognizes five types of approaches in pairs trading:

1. **Distance Approach**: focuses on non-parametric distance metrics to identify pairs trading opportunities

2. **Cointegration Approach**: relies on formal cointegration testing to unveil stationary spread time series

3. **Time Series Approach**: in finding optimal trading rules for mean-reverting spreads

4. **Stochastic Control**: identifying optimal portfolio holdings in the legs of a pairs trade relative to other available securities

5. **Other:** Consisting of Principal Component Analysis (PCA), copula models, and machine learning approaches.

The focus of this thesis will be based on the idea from the Cointegration Approach.

1.2 Cointegration-based Approach:

The concept of cointegration is the following: while it could be difficult to predict two random walks, sometimes is easier to infer the relative movements between them. An intuitive example is the analogy of a drunk and her dog, as described by Michael P. Murray. (1994). Both the drunk and the dog are likely following a random walk, but since they are linked by the leash, their relative distance tends to revert back to the mean.

1.2.1 Definition of Cointegration:

Consider two time series x_t and y_t, that follow I(1) processes:

$$x_t = x_{t-1} + \epsilon_{y,t} \tag{1.1}$$

$$y_t = y_{t-1} + \epsilon_{y,t} \tag{1.2}$$

where $\epsilon_{x,t}$ and $\epsilon_{y,t}$ are stationary process. These time series x_t and y_t are said to be *cointegrated* if there exists a linear combination of the two, such that $z_t = x_t - \beta y_t$ follow I(0) process. A process that follow I(0) is called stationary: this condition implies that z_t has a constant mean, variance, and lag dependency over time, and exhibits mean reversion.

1.2.2 Trading strategy

Suppose the spread $z_t = y_{1t} - \beta y_{2t}$ follows an I(0) process, where y_{1t} and y_{2t} are the prices of the two assets, and β represents the "cointegration" coefficient. Suppose β in this "naive" setting is estimated using a linear regression between the two asset prices. Once we know the value of β, the trading strategy for statistical arbitrage can be outlined as follows:

When the Spread is Low (z_t < −threshold): this indicates that stock 1 is undervalued relative to stock 2. Consequently the strategy consists in buying stock 1 and short-sell stock 2, which corresponds to taking a long position in the spread. Finally, when the spread revert back, the positions taken will be closed.

Vice versa , **when the Spread is High (z_t > threshold)**: we take a short position on the spread, and we close the position when the spread reverts back.

Figure 1.1 offer a visual representation of the described strategy.

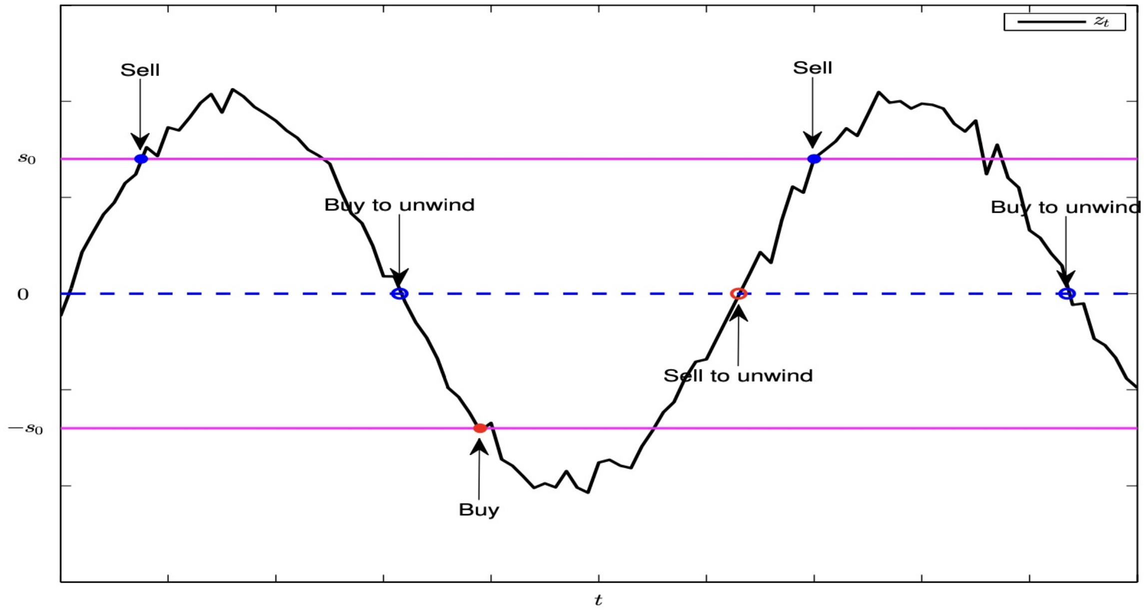

Figure 1.1: Strategy illustration

Profit Calculation: The profit from the trading strategy is determined by the difference in the spread when entering and exiting the trade. Specifically, if a trade has started at time t and exit at time $t+i$, the profit is:

$$\text{Profit} = z_{t+i} - z_t \tag{1.3}$$

The spread z_t is calculated using the portfolio weights, which satisfy the condition $\sum_i |w_i| = 1$.

$$|w_i| = \left|\frac{1}{1+\beta}\right| + \left|-\frac{\beta}{1+\beta}\right| = 1 \tag{1.4}$$

The portfolio weights are (note: the second component is negative because indicate a short position):

$$\mathbf{w} = \begin{bmatrix} \frac{1}{1+\beta} \\ -\frac{\beta}{1+\beta} \end{bmatrix} \tag{1.5}$$

Thus, the spread z_t is given by:

$$z_t = \mathbf{w}^\top \begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \frac{1}{1+\beta} y_{1,t} - \frac{\beta}{1+\beta} y_{2,t} \tag{1.6}$$

The profit when exiting the trade is:

$$z_{t+i} - z_t = \left(\frac{1}{1+\beta} y_{1,t+i} - \frac{\beta}{1+\beta} y_{2,t+i} \right) - \left(\frac{1}{1+\beta} y_{1,t} - \frac{\beta}{1+\beta} y_{2,t} \right) \tag{1.7}$$

Which simplifies, as follows:

$$z_{t+i} - z_t = \frac{1}{1+\beta} (y_{1,t+i} - y_{1,t}) - \frac{\beta}{1+\beta} (y_{2,t+i} - y_{2,t}) \tag{1.8}$$

Simple Returns: To pass from profit to simple returns, the spread is divided by its initial value:

$$\text{Simple Return} = \frac{z_{t+i} - z_t}{z_t} \tag{1.9}$$

Which can be expressed as:

$$\text{Simple Return} = \frac{1}{1+\beta} \cdot \frac{y_{1,t+i} - y_{1,t}}{y_{1,t}} - \frac{\beta}{1+\beta} \cdot \frac{y_{2,t+i} - y_{2,t}}{y_{2,t}} \tag{1.10}$$

Logarithmic Returns: Logarithmic return of a portfolio with weights $\mathbf{w}$ is given by the dot product of the weights and the vector of changes in asset prices, $\Delta \log y_t$. Specifically, if $\mathbf{w}$ is the weights vector and $\Delta \log y_t$ represents the changes in asset prices, then the portfolio return is:

$$\text{Logarithmic Return} = \mathbf{w}^\top \Delta \log y_t \tag{1.11}$$

where:

$$\mathbf{w} = \begin{bmatrix} \frac{1}{1+\beta} \\ -\frac{\beta}{1+\beta} \end{bmatrix}, \quad \Delta \log y_t = \begin{bmatrix} \log y_{1,t+i} - \log y_{1,t} \\ \log y_{2,t+i} - \log y_{2,t} \end{bmatrix}. \tag{1.12}$$

We use log-returns instead of the simple one, because they are additive over time, meaning that it is possible to obtain the total log-return over that period just taking the sum. Morever when considering log-returns the computation of the pairs trading strategy is simpler because it requires just to take the first difference. Additionally, log-returns tend to be normally distributed.

1.2.3 Limitation of this approach

The cointegration-based approach is the most used and analyzed among practitioners and in academic literature: although it has the following limitations:

Time-Varying Beta: Empirical studies show that the cointegration parameter β between two assets, do not remain fixed through time. Additionally, this quantity is subject to high levels of noise, resulting in substantial fluctuations and uncertainty. In this situation, modelling the spread in state-space, give an advantage in estimating β_t in time. This enables the model to adjust more effectively to shifts in the relation between the asset pairs, without requiring the identification of specific parameters like the window length in rolling window least squares. **Figure 1.2** displays how the relation between SPY (tracking S&P Index) and EWI (tracking MSCI Italy 25/50 Index) changed between 2010 and 2015. In the scatter plot cooler colors indicate data from previous periods, while warmer colors reflect more recent data.
As the data-points shift from blue to red, there is a relevant increase in the difference between SPY and EWI prices. This visual representation shows how the relation between these two ETFs has changed over time.
The plot compares two types of regression fits:

- OLS regression (black line): Applying a static fit, we are able to get the overall trend, but fails to capture time-clusters.

- Time-varying parameter model represented by the colored lines and based on the theoretical framework of chapter 2: this type of model dynamically adjusts the regression parameters over time, producing multiple fit-lines corresponding to different periods. As the colors shift from colder to warmer, these lines clearly show how this model can adapt its parameter to consider shifts in the fundamental relations.

While the OLS regression line remains fixed and unable to account for changes over time, the time-varying model adapts to the shifting market conditions. As a result, the colored lines better capture the shift in the relationship between SPY and EWI as time progresses.

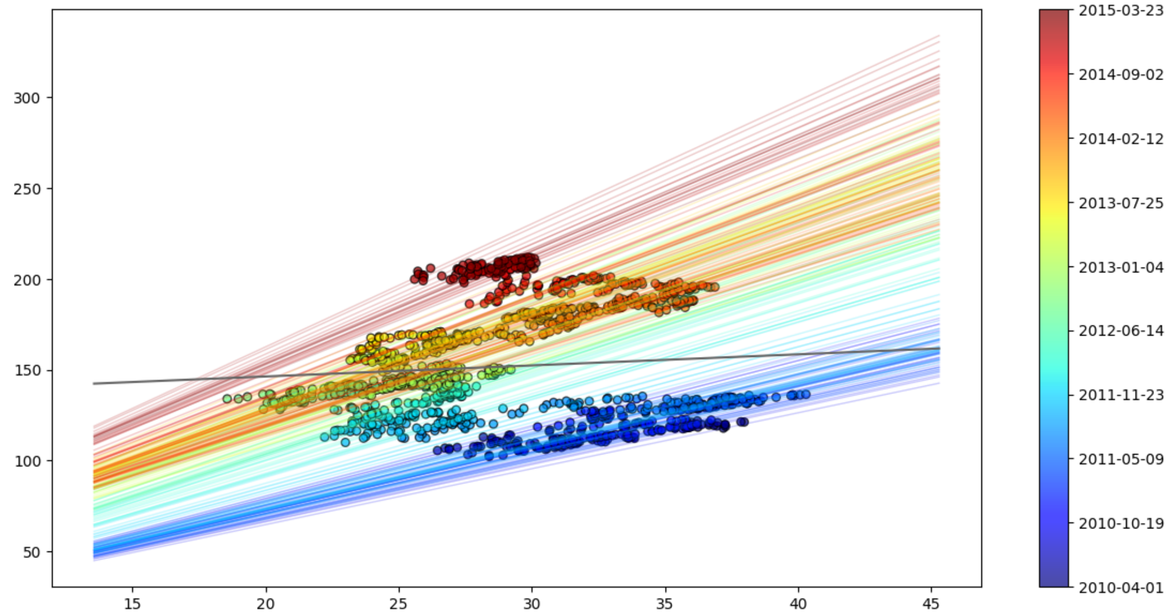

Figure 1.2: SPY (Y axis) vs EWI (X axis): OLS Regression vs. Time-Varying Model

Intermittent Cointegration: Cointegration relationships can be intermittent instead of stable in the long run. This means that the assets may exhibit cointegration in some period but not for all their life. Additionally, as found in Clegg (2014), a time series that is cointegrated in one period is not necessarily more likely to be cointegrated in the following period. This problem is considered in Clegg and Krauss (2018), as will be discussed in the chapter 3.

1.3 Research Objectives

The aim of this thesis is to face the limitations of cointegration-based methods by creating a framework that employ state space models to enhance pairs trading strategies.

The model discussed in Chapter 2 tackles the limitation of the "time-varying" β. This model, referred as KFB, addresses this limitation employing a state-space representation of the spread dynamic in order to keeping estimates of β_t up-to-date and monitoring changes in assets relations.

Conversely, chapter 3 is focused on the paper by Clegg and Krauss (2018). This paper addresses the problem of intermittent cointegration. The described partial cointegration model presents a new approach for choosing the most potential pairs and generating real-time signals.

Chapter 4 objective is to create simulated time series to evaluate the accuracy of the models in predicting the spread, under the condition that the data generating process aligns with CK assumptions.

Chapter 6 analyzes the performance of the two models by backtesting them on a specific group of ETFs from 2014 to 2023, in order to determine if they can produce market-neutral returns.

2 Modelling the Spread with Time-Varying β_t

In this chapter, we address a key limitation of the cointegration-based approach. Since it has been shown that the parameter β is not constant over time, it is necessary to model the spread with a time-varying setting $z_t = x_t - \beta_t y_t$, where β_t is treated as a time-varying coefficient

2.1 State-Space Model with Time-Varying β_t (KFB)

To address the limitations discussed earlier, we rewrite the spread in state-space. This method will be referred to as the KFB method (Kalman Filter Beta) throughout this thesis. The observation and state equations are given as follows:

$$y_t = x_t'\beta_t + w_t, \tag{2.1}$$

$$\beta_{t+1} = \beta_t + v_{t+1}, \tag{2.2}$$

where β_t, x_t, y_t are $(k \times 1)$ vectors and x_t, y_t represent the prices of the two assets. It is assumed that, conditional on x_t and the data observed through date $t-1$, denoted as $\mathcal{Y}_{t-1} = (y_{t-1}, y_{t-2}, \ldots, y_1, x_{t-1}, x_{t-2}, \ldots, x_1)'$, the vector $(v_{t+1}', w_t')'$ follows a multivariate normal distribution with the following properties:

(2.3) $$\begin{pmatrix} v_{t+1} \\ w_t \end{pmatrix} \Bigg| \mathcal{Y}_{t-1}, x_t \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_w^2 \end{pmatrix} \right)$$

Where σ_v^2 represents the variance of the disturbances in the state equation (the process noise), while σ_w^2 represents the variance of the disturbances in the observation equation (the measurement noise).

Assuming that $\beta_{t|t-1} \sim \mathcal{N}(\hat{\beta}_{t|t-1}, P_{t|t-1})$, we can write that:

(2.4) $$\begin{pmatrix} \beta_t \\ y_t \end{pmatrix} \Bigg| \mathcal{Y}_{t-1}, x_t \sim \mathcal{N}\left(\begin{pmatrix} \hat{\beta}_{t|t-1} \\ x_t'\hat{\beta}_{t|t-1} \end{pmatrix}, \begin{pmatrix} P_{t|t-1} & P_{t|t-1}x_t \\ x_t'P_{t|t-1} & x_t'P_{t|t-1}x_t + \sigma_w^2 \end{pmatrix} \right)$$

The initial state distribution is specified as follows:

$$\beta_{1|0} \sim \mathcal{N}(\hat{\beta}_{1|0}, P_{1|0})$$

The updated estimate $\hat{\beta}_t$ is

(2.5) $$\hat{\beta}_{t|t} = \hat{\beta}_{t|t-1} + K_t \left[y_t - x_t'\hat{\beta}_{t|t-1} \right],$$

Where the Kalman Gain K_t is

(2.6) $$K_t = P_{t|t-1}x_t \left[x_t'P_{t|t-1}x_t + \sigma_w^2 \right]^{-1},$$

The updated covariance matrix $P_{t|t}$ is

(2.7) $$P_{t|t} = P_{t|t-1} - K_t x_t' P_{t|t-1},$$

The prediction for the covariance matrix is

(2.8) $$P_{t+1|t} = P_{t|t} + \sigma_v^2,$$

The mean squared error (MSE) of this forecast will be

$$\mathbb{E}\left[(y_t - x_t'\beta_{t|t-1})^2 \mid x_t, \mathcal{Y}_{t-1}\right] = x_t' P_{t|t-1} x_t + \sigma_w^2 \tag{2.9}$$

The log-likelihood function is therefore:

$$L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log(x_t' P_{t|t-1} x_t + \sigma_w^2) - \frac{1}{2}\sum_{t=1}^{T}\frac{(y_t - x_t'\hat{\beta}_{t|t-1})^2}{x_t' P_{t|t-1} x_t + \sigma_w^2} \tag{2.10}$$

To find the maximum likelihood estimates the following function will be minimized with respect to the parameter σ_v^2 and σ_w^2.

$$L_R(\sigma_v^2, \sigma_w^2) = -\frac{1}{2}\sum_{t=1}^{T}\left[\log(x_t' P_{t|t-1} x_t + \sigma_w^2) + \frac{(y_t - x_t'\hat{\beta}_{t|t-1})^2}{x_t' P_{t|t-1} x_t + \sigma_w^2}\right] \tag{2.11}$$

The function `estimate_Beta_KFB` is implemented within the R functions. The estimate $\hat{\beta}_{1|0}$ is derived from least squares estimates over the training period, with $P_{1|0} = 1 \times 10^{-4}$ providing a baseline for the Kalman filter. The function `fitSSM` minimizes the negative log-likelihood using the `BFGS` algorithm (quasi-Newton) to estimate σ_v^2 and σ_w^2. It uses the `SSModel` function to characterise the state-space model representation by specifying the observation and state transition equations. The Kalman filter is then applied via `KFS` to generate state predictions, and the `rollapply` function smooths the forecast.
These parameters are then used to perform an in-sample estimation of $\hat{\beta}_{t|t}$.

2.1.1 Z Score

Once we have inferred $\hat{\beta}_{t|t}$, we can compute the dynamic spread:

$$\hat{z}_{t|t} = x_t - \hat{\beta}_{t|t} y_t \tag{2.12}$$

The Z-score is calculated as:

$$Z\text{-score}_{\mathrm{KFB}} = \frac{\hat{z}_{t|t}}{\tilde{\sigma}_{z_t}} \tag{2.13}$$

Where $\tilde{\sigma}_{z_t}$ is the standard deviation of the in-sample estimation of $\hat{z}_{t|t}$.
We use a Z-score because we want to know how many standard deviation the current spread deviates from its mean. In this way, we are able to compute when the spread is high or low compared to its historical behavior, and this measure is used to generate the trading signals.

2.2 Strength of the described methodology

The KFB model is well suited for pairs trading applications for several reasons. Firstly, it allows for the dynamic adjustment of coefficients; the KFB model estimates the time-varying coefficient $\hat{\beta}_{t|t}$ in the spread equation $\hat{z}_{t|t} = x_t - \hat{\beta}_{t|t} y_t$, enabling it to calibrate dynamically in response to shifts in the assets relation. Secondly, the model avoids overfitting as it does not require external parameters, as in the rolling window least squares, which reduces the risk of overfitting to historical data. Lastly, the KFB model performs well with non-Gaussian disturbances; although the Kalman filter assumes Gaussian disturbances, it still provides the best linear estimation even if this assumption is violated (Simon, 2006).

3 Pairs Trading in Partial Cointegration

Partial cointegration model (PCI) was proposed by Clegg and Krauss (2018) with the following motivation: cointegration implies that permanent shocks are common, while assets are subject to idiosyncratic shocks that are persistent and could blur the line between long memory, stationarity and mean reversion. Differently from cointegration tests, that have a binary result, either discarding or not-rejecting the existence of a long term relationship between assets, the PCI model is able to offer a deeper analysis of the relation by decomposing the spread into two different components: the "mean-reverting" and the "random-walk" component. Splitting the spread in this way enable us to examine the contribution of each component to the variance of the whole spread process Pairs trading strategies are profitable if the first component dominates the second. In this framework we are able to select pairs with a strong "mean-reversion" component and to use the inference on this hidden state to take trading decisions.

3.1 Strengths and Innovations

Clegg and Krauss (CK) decompose the spread between two series into two orthogonal components: one mean-reverting and the other represented by a random walk.

Let y_{1t} and y_{2t} be the prices of two assets, and consider the spread between them $s_t = y_{1t} - \beta y_{2t}$, $t = 1, \dots, n$. It is assumed that s_t is a process resulting from the sum of a permanent component τ_t, modeled with a random walk (RW), and a transitory (mean-reverting) component ψ_t, represented by a first-order autoregressive process (AR(1)), orthogonal to the former. The model specification is as follows:

$$
\begin{aligned}
y_{1t} &= \beta y_{2t} + s_t \\
s_t &= \tau_t + \psi_t, \\
\tau_t &= \tau_{t-1} + \zeta_t, \quad \zeta_t \sim \text{i.i.d.}\, N(0, \sigma_\zeta^2), \\
\psi_t &= \rho \psi_{t-1} + \kappa_t, \quad \kappa_t \sim \text{i.i.d.}\, N(0, \sigma_\kappa^2),
\end{aligned}
$$

where the autoregressive coefficient is in the stationary region, $|\rho| < 1$, and the components are independent, $E(\zeta_t, \kappa_s) = 0, \forall(t, s)$. The series y_{1t} and y_{2t} are forming a PAR (partial autoregressive) sequence linked by the parameter β. Note that $y_{1t} = \beta y_{2t} + \tau_t + \psi_t$, while $y_{2t} = y_{2t-1} + \epsilon_t$ with $\epsilon_t \sim N(0, \sigma_\epsilon^2)$ and that it is independent of κ_t and ζ_t. Note that β is constant.

By decomposing the variance of the first differences, $\text{Var}(\Delta s_t) = \text{Var}(\Delta\tau_t) + \text{Var}(\Delta\psi_t)$, is possible to measure the contribution of the mean-reverting component through the ratio

$$
R_\psi^2 = \frac{\text{Var}(\Delta\psi_t)}{\text{Var}(\Delta s_t)} = \frac{2\sigma_\kappa^2}{2\sigma_\kappa^2 + (1+\rho)\sigma_\zeta^2}
$$

This ratio is crucial to understand the proportion of the variance that is driven by the "mean reversion" component and the RW component.

3.2 PCI Estimation

Since s_t is not directly observable, the model is restated in state space. The state space representation involves two equations, an observation equation and a state equation. These equations are given as

$$
y_t = H_t s_t + V_t \quad (3)
$$

$$
S_t = F_t s_{t-1} + G_t U_t + W_t. \quad (4)
$$

The state of the system is given by s_t in (4), which may not be directly observable. It is assumed to follow a linear dynamic and it may be influenced by a control input U_t. The term W_t is a noise term, which has covariance matrix Q_t. The observable portion of the system is represented by y_t in (3). It is assumed to have a linear dependence on the hidden state

s_t, given by H_t, and to be influenced by its own noise term V_t, whose covariance matrix is R_t. The noise term V_t is assumed to be zero and that there is no control input term U_t. In addition, it is assmed that the linear dependence matrix H_t and the transition matrix F_t are time invariant. Consequently, these equations simplify to

$$Y_t = Hs_t \quad (5)$$

$$s_t = Fs_{t-1} + W_t. \quad (6)$$

The partial cointegration (PCI) system has two observable variables, y_{1t} and y_{2t}, and two hidden state variables ψ_t and τ_t. For convenience of representation, y_1 is treated as a third hidden state variable. In other words, y_{1t} is represented in both the observation equation and the state equation. The observation equation for the PCI system is therefore given as

$$Y_t = \begin{bmatrix} y_{2,t} \\ y_{1,t} \end{bmatrix} = Hs_t = \begin{bmatrix} \beta & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_{1,t} \\ \psi_t \\ \tau_t \end{bmatrix}. \quad (7)$$

And the hidden state equation for the PCI system is given as

$$s_t = \begin{bmatrix} y_{1,t} \\ \psi_t \\ \tau_t \end{bmatrix} = Fs_{t-1} + W_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ \psi_{t-1} \\ \tau_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_t \\ \kappa_t \\ \zeta_t \end{bmatrix}, \quad (8)$$

Parameter values are determined through maximum likelihood estimation of the associated Kalman filter. If the parameters of the system are known and the innovations ϵ_t, κ_t, and ζ_t are zero-mean Gaussian, uncorrelated, and white, the Kalman Filter minimizes the mean-squared error of the estimated parameters. If the innovations are zero-mean, uncorrelated, and white, but non-Gaussian, then the Kalman Filter is still the best linear estimator (Simon, 2006).
Let Θ_t denote the information that is available up to and including time t, and let Φ denote the parameter values $\beta, \rho, \sigma_\epsilon, \sigma_\kappa$, and σ_ζ. The one-step ahead prediction error given by the

Kalman filter is $e_t = y_t - E[y_t|\Theta_{t-1}, \Phi]$. In the case of the PCI model, it can be shown to be

$$e_t = \begin{bmatrix} \beta\epsilon_t + \kappa_t + \zeta_t \\ \epsilon_t \end{bmatrix}. \quad (9)$$

Since $p(\beta\epsilon_t + \kappa_t + \zeta_t, \epsilon_t) = p(\kappa_t + \zeta_t, \epsilon_t)$, the likelihood function for the Kalman filter of the PCI model can be written as

$$L(\Phi) = p(y_1|\Phi) \prod_{k=2}^{n} \phi(\kappa_t, k + \zeta_t, k; 0, \sigma_\kappa^2 + \sigma_\zeta^2) \prod_{k=2}^{n} \phi(\epsilon_t, k; 0, \sigma_\epsilon^2), \quad (10)$$

where $\phi(\cdot)$ denotes the probability density function of the normal distribution and $p(y_1|\Phi)$ is a constant term corresponding to the first observation. The interest is to optimizing for $\beta, \rho, \sigma_\kappa$, and σ_ζ, so is possible to omit the first and third term from the above product. In other words, the maximum likelihood estimates for $\beta, \rho, \sigma_\kappa$ and σ_ζ can be found by maximizing

$$L_{MR}(\beta, \rho, \sigma_\kappa, \sigma_\zeta) = \prod_{k=2}^{n} \phi(\kappa_t, k + \zeta_t, k; 0, \sigma_\kappa^2 + \sigma_\zeta^2). \quad (11)$$

The likelihood score as the objective function, and deploy Newton method to jointly optimize over $\beta, \rho, \sigma_\kappa$, and σ_ζ. The full algorithm is implemented in the R package `partialCI`. Given specific values for ρ, σ_κ, and σ_ζ, these can be substituted into the above equation to find a steady state solution, which is then used to compute the steady state Kalman gain matrix which can be computed in closed form with the following formula and implemented in R `kalman_gain` .5

$$\mathbf{K} = \begin{pmatrix} \frac{2\sigma_\kappa^2}{\sigma_\zeta\left(\sqrt{(\rho+1)^2\sigma_\zeta^2+4\sigma_\kappa^2}+\rho\sigma_\zeta+\sigma_\zeta\right)+2\sigma_\kappa^2} \\ \frac{2\sigma_\zeta}{\sqrt{(\rho+1)^2\sigma_\zeta^2+4\sigma_\kappa^2}-\rho\sigma_\zeta+\sigma_\zeta} \end{pmatrix}$$

3.2.1 Estimation Algorithm

Once we trained the parameters $\hat{\beta}$, $\hat{\rho}$, $\hat{\sigma}_\kappa$, and $\hat{\sigma}_\zeta$ the procedure for estimating the mean-reverting series $\hat{\psi}_{t|t}$ and the random walk series $\hat{\tau}_{t|t}$ in "real time" involves the following steps:

1. **Calculate the Kalman Gain:** Compute the Kalman gain $\mathbf{K}$, consisting in $\hat{k}_1$ and $\hat{k}_2$, using the parameters $\hat{\rho}$, $\hat{\sigma}_\kappa$, and $\hat{\sigma}_\zeta$.

2. **Set Initial States:** Initialize the state variables as $\kappa_{1|0} = 0$ and $\zeta_{1|0} = s_1$.

3. **Iterate Through Observations:** Process each observation in the input sequence $\mathbf{s}_t$ and apply the Kalman update equations to estimate the hidden states.

The equations for real-time states estimation:

$$s_t = y_{1,t} - \hat{\beta} y_{2,t} \tag{3.1}$$

$$\hat{E}_{t|t} = s_t - \hat{\rho}\hat{\psi}_{t-1|t-1} - \hat{\tau}_{t-1|t-1} \tag{3.2}$$

$$\hat{\psi}_{t|t} = \hat{\rho}\hat{\psi}_{t-1|t-1} + \hat{k_1}\hat{E}_{t|t} \tag{3.3}$$

$$\hat{\tau}_{t|t} = \hat{\tau}_{t-1|t-1} + \hat{k_2}\hat{E}_{t|t} \tag{3.4}$$

The algorithm is implemented in the following R function `kalman_estimate` .6
Note that these equations are applied subsequently to the data in the formation period to obtain an in-sample estimate of $\hat{\psi}_{t|t}$ and its standard deviation $\tilde{\sigma}_{\hat{\psi}_{t|t}}$

3.3 Z score

Similarly to the KFB model also for PCI a Z-score is computed with the same methodology described in 2.1.1, but in this case instead of computing the Z-score on $\hat{z}_{t|t}$ we are only interested in trading the "mean reversion" component $\hat{\psi}_{t|t}$. So the z-score becomes:

$$Z\text{-score}_{\text{PCI}} = \frac{\hat{\psi}_{t|t}}{\tilde{\sigma}_{\hat{\psi}_{t|t}}} \tag{3.5}$$

Where $\tilde{\sigma}_{\hat{\psi}_{t|t}}$ is the standard deviation of $\hat{\psi}_{t|t}$ from in-sample data.

3.4 Limitation of the Study

In Clegg and Krauss paper the back-testing of the pairs trading strategies was conducted using data from the S&P 500 index constituents over the period from January 1990 to October 2015. For the back-testing, the authors focused on forming pairs of stocks within the same sector, as defined by the Global Industry Classification Standard (GICS). The authors implemented this restriction make sure to contain the risk of having spurious correlations. This choice presents certain limitations, as also outline by the authors.

Firstly, regarding market efficiency and liquidity, the high efficiency and liquidity of S&P 500 stocks may reduce the potential for identifying significant market inefficiencies. In such a competitive environment, price anomalies are often quickly corrected, which can limit the effectiveness of mean reversion strategies.

Secondly, idiosyncratic risks are a concern, as individual stocks are subject to high specific risks, such as management changes or company-specific events. These risks are less impactful on more diversified instruments, such as ETFs.

4 Backtesting on Simulated Data

In this section, we provide a detailed overview of the setup for simulating data used to backtest the two models. The aim is to generate a pair of time series that follow the assumptions from the Clegg and Krauss paper. The primary objectives are to evaluate whether the PCI model can accurately decompose the spread and infer the hidden states, and to determine if the KFB model produces accurate inferences regarding the spread, using in both cases the same data-generating process (DGP). This setup allows us to establish a clear benchmark for assessing the models in a controlled environment.

4.1 Simulation setup

In the model, the time series Y_{2t} was defined as a random walk. The process s_t consists of the sum of ψ_t, which follows an AR(1) process, and τ_t, which follows a random walk. This configuration is consistent with the PCI model, where we set $\sigma_\kappa = 0.9$ and $\sigma_\zeta = 0.1$ and $\rho = 0.96$, resulting in a half life of mean-reversion of 17 days. The time series Y_{1t} is computed as $Y_{1t} = \beta \cdot Y_{2t} + s_t$. The code used for the simulation can be found in Section .9.

For this simulation, the proportion of variance explain by the mean reversion component R^2_ψ is approximately 0.97. This setting creates ideal conditions for the pair to exploit a mean reversion strategy. To replicate real market backtesting conditions, a 4-year dataset is used for training the parameters, followed by a 6-month period dedicated to performance testing. To ensure the reproducibility of the experiment, a fixed random seed was set.

Figure 4.1 displays a plot of the simulated time series.

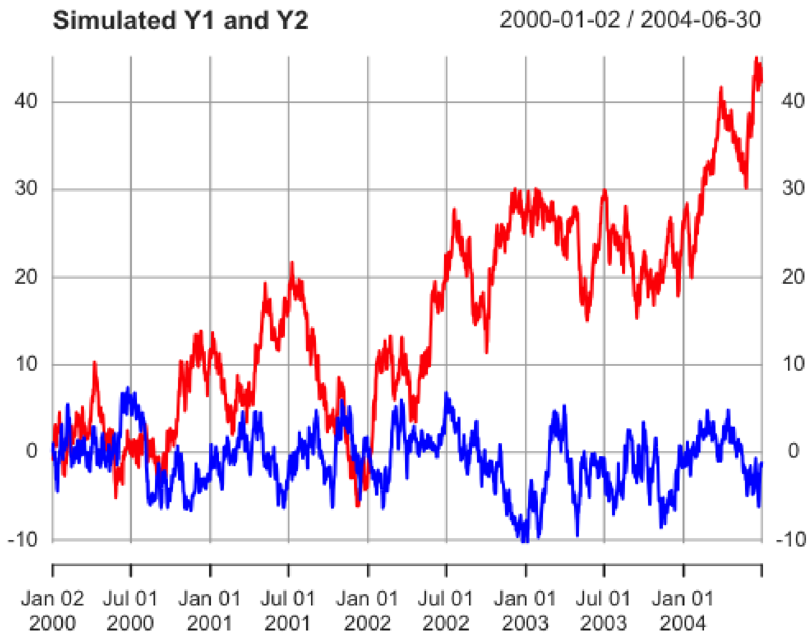

Figure 4.1: Plot of the simulated series

Since we simulate the processes ψ_t and τ_t, we know the exact values they take at every time step. This enables us to assess the PCI model's performance in accurately estimating these values. Consequently, we can evaluate the accuracy of the PCI model and its capability to break down the spread into two components by comparing estimated values with actual values.

Figure 4.2 shows the evolution of the simulated parameter ψ_t transformed with the Z-score function (black line), compared with the forecast from the PCI model; $\hat{\psi}_{t|t}$ (red line).

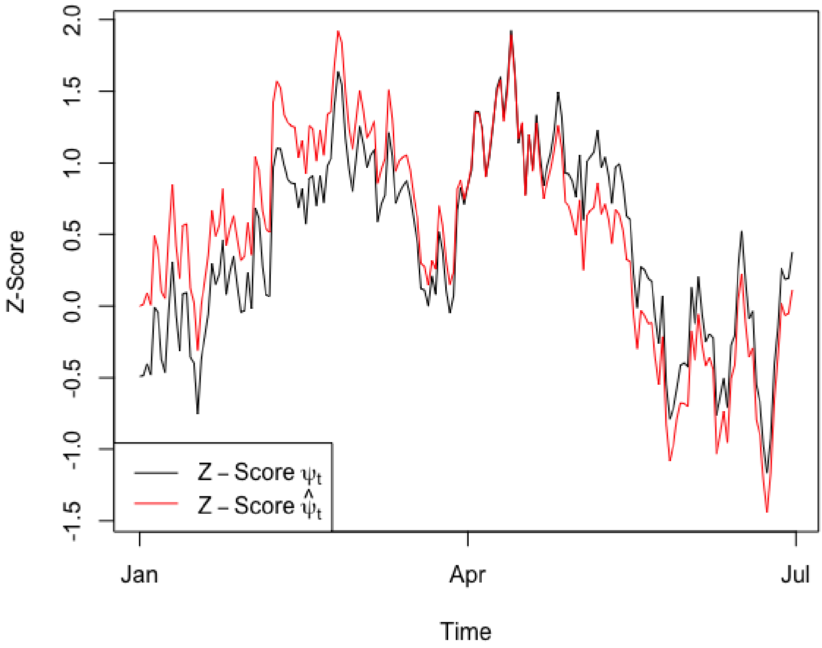

Figure 4.2: Z score $\hat{\psi}_{t|t}$ (PCI) vs Z score ψ_t (TRUE) corr=0.89

Figure 4.2 highlights the exceptional forecasting performance of the PCI in capturing the mean reversion component. Its accuracy in predicting this dynamic stands out, showcasing the method's strength in inferring this hidden state.

Figure 4.3 depict the evolution of the simulated spread z_t transformed with the Z-score function (black line), compared with forecasts from the KFB model $\hat{z}_{t|t}$ (red lines).

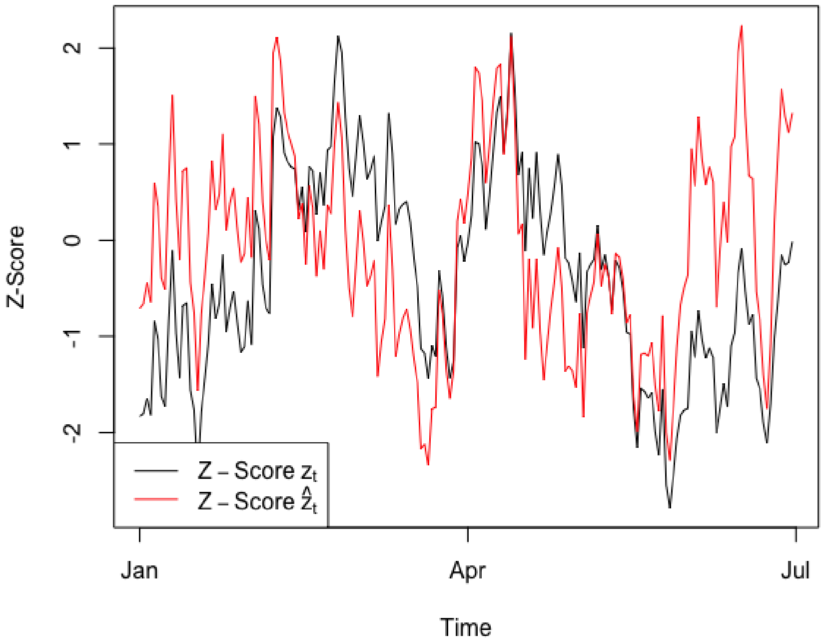

Figure 4.3: Z score $\hat{z}_{t|t}$ (KFB) vs Z score z_t (TRUE) corr=0.55

This plot highlights that the KFB model is able to capture a substantial part of the spread variability, but since it has more general assumptions, it is not able to capture all the characteristic of CK data generating process.

4.2 Results and Analysis

The simulation indicates that when the process is generated according to CK assumptions, the PCI model almost perfectly captures the "mean reversion" component, while the KFB model is considerably less precise in estimating the spread, having more general assumptions. These results are in line with our expectation, given the theoretical framework of both models. In the next chapter, in light of these findings, we will test if a pairs trading strategy, built on these two models is able to generate market-neutral returns. Additionally we will test if the PCI model is able to infer effectively the hidden states also in real market data, and if this model is superior with respect to the KFB approach.

5 Market Data Preparation

In this thesis we apply the strategy to ETFs, instead of individual stocks, differently to CK paper. Applying pairs trading strategies to ETFs offers several advantages. The first advantage is that ETFs mitigate idiosyncratic risks, as they are composed by a basket of assets that belongs to a certain index. Due to this characteristic, the relations between them have more potential to be explained by permanent component, meaning that their spread could likely be modelled by mean-reverting processes. To make a clear example: consider USO (United States Oil Fund) and XLE (Energy Select Sector SPDR Fund). These two ETFs have a deep relation, because their price is strongly related to energy prices. However, their relative movement in certain periods could move away because XLE tracks futures contracts on oil, that often anticipates the prices of stocks that produce and manage the underlying commodity, tracked by USO. This type of discrepancy in the spread could create opportunities for pairs trading strategy, where deviations from their relative behaviour can be exploited.

5.1 Extension of Time Frame

In the thesis we aim to measure the profitability of the strategy from 2014 to 2023. This period has been chosen to continue the study from CK, since the last year of the backtest in the paper was 2015. In addition this 10 year period gives us the possibility to test the strategy under period of high volatility, such as COVID-19 crisis

5.2 Selection of Data Sources

For this analysis a diversified collection of ETFs has been considered. This collection was built considering exposure to different sector, geographical, and commodity market, to make sure that our model is able to capture complex interactions between this markets, that can be exploited within this broad basket of instruments. Only ETFs that were available from 2010 were considered, to ensure that we have sufficient data to train the model correctly.

5.2.1 Dataset Construction

The dataset was created by going through multiple steps. The first step in choosing ETFs using Morningstar to filter and pinpoint ETFs that met specific criteria like asset class, historical data reliability, industry-sector specific, and geographical location. The `quantmod` package in R was utilized to collect closing price information from Yahoo Finance for each ETF. The dataset contains adjusted closing prices to reflect corporate events like dividends and stock splits. The data retrieval was scheduled from January 1, 2010, up to the current date to make sure the dataset includes a substantial history. In order to deal with missing data and prevent look-ahead bias, we used a forward-filling function, which carries the last observed value forward to maintain continuity without incorporating future information.
All ETFs were aligned to the same time-period to maintain uniformity throughout the dataset. The table below contains the 67 ETFs used in the analysis, referencing their sector, and geographical market.

5.3 ETF List with Sector and Geographical identification

ETF	Name	Geographical Market	Sector
EWA	iShares MSCI Australia	Australia	-
EWK	iShares MSCI Belgium ETF	Belgium	-
EWO	iShares MSCI Austria ETF	Austria	-
EWC	iShares MSCI Canada ETF	Canada	-
EWQ	iShares MSCI France ETF	France	-
EWG	iShares MSCI Germany ETF	Germany	-
EWH	iShares MSCI Hong Kong ETF	Hong Kong	-
EWI	iShares MSCI Italy ETF	Italy	-

ETF	**Name**	**Geographical Market**	**Sector**
EWJ	iShares MSCI Japan ETF	Japan	-
EWM	iShares MSCI Malaysia ETF	Malaysia	-
EWW	iShares MSCI Mexico ETF	Mexico	-
EWN	iShares MSCI Netherlands ETF	Netherlands	-
EWS	iShares MSCI Singapore ETF	Singapore	-
EWP	iShares MSCI Spain ETF	Spain	-
EWD	iShares MSCI Sweden ETF	Sweden	-
EWL	iShares MSCI Switzerland ETF	Switzerland	-
EWY	iShares MSCI South Korea ETF	South Korea	-
EZU	iShares MSCI Eurozone ETF	Eurozone	-
EWU	iShares MSCI United Kingdom ETF	United Kingdom	-
EWZ	iShares MSCI Brazil ETF	Brazil	-
EWT	iShares MSCI Taiwan ETF	Taiwan	-
SPY	SPDR S&P 500 ETF Trust	USA	-
EZA	iShares MSCI South Africa ETF	South Africa	-
EPI	WisdomTree India Earnings Fund	India	-
RSX	VanEck Russia ETF	Russia	-
TUR	iShares MSCI Turkey ETF	Turkey	-
EIS	iShares MSCI Israel ETF	Israel	-
THD	iShares MSCI Thailand ETF	Thailand	-
PIN	PowerShares India Portfolio	India	-
NORW	Global X MSCI Norway ETF	Norway	-
EEM	iShares MSCI Emerging Markets ETF	Emerging Markets	-
VWO	Vanguard FTSE Emerging Markets ETF	Emerging Markets	-
AAXJ	iShares Asia ex-Japan ETF	Asia (ex-Japan)	-
ILF	iShares Latin America 40 ETF	Latin America	-
AFK	VanEck Africa Index ETF	Africa	-

ETF	Name	Geographical Market	Sector
FEZ	SPDR Euro Stoxx 50 ETF	Eurozone	-
XLF	Financial Select Sector SPDR Fund	USA	Financial Sector
XLK	Technology Select Sector SPDR Fund	USA	Technology Sector
XLE	Energy Select Sector SPDR Fund	USA	Energy Sector
XLV	Health Care Select Sector SPDR Fund	USA	Health Care Sector
XLY	Consumer Discretionary Select Sector SPDR Fund	USA	Consumer Discretionary
XLI	Industrial Select Sector SPDR Fund	USA	Industrial Sector
XLB	Materials Select Sector SPDR Fund	USA	Materials Sector
XLU	Utilities Select Sector SPDR Fund	USA	Utilities Sector
IYR	iShares U.S. Real Estate ETF	USA	Real Estate
SMH	VanEck Vectors Semiconductor ETF	USA	Semiconductors
XBI	SPDR S&P Biotech ETF	USA	Biotechnology
VTI	Vanguard Total Stock Market ETF	USA	Total Market
IVV	iShares Core S&P 500 ETF	USA	S&P 500
QQQ	Invesco QQQ Trust	USA	Nasdaq 100
IWV	iShares Russell 3000 ETF	USA	Russell 3000
GLD	SPDR Gold Shares	-	Commodities (Gold)
SLV	iShares Silver Trust	-	Commodities (Silver)
USO	United States Oil Fund	-	Commodities (Oil)
UNG	United States Natural Gas Fund	-	Commodities (Natural Gas)
DBO	Invesco DB Oil Fund	-	Commodities (Oil)

ETF	Name	Geographical Market	Sector
DBC	Invesco DB Commodity Index Tracking Fund	-	Commodities
UGA	United States Gasoline Fund	-	Commodities (Gasoline)
DBA	Invesco DB Agriculture Fund	-	Commodities (Agriculture)
GSG	iShares S&P GSCI Commodity-Indexed Trust	-	Commodities
SOXX	iShares PHLX Semiconductor Sector ETF	USA	Semiconductors
FDN	First Trust Dow Jones Internet Index Fund	USA	Internet
TAN	Invesco Solar ETF	USA	Solar Energy
ICLN	iShares Global Clean Energy ETF	Global	Clean Energy
PBW	Invesco WilderHill Clean Energy ETF	USA	Clean Energy
IBB	iShares Nasdaq Biotechnology ETF	USA	Biotechnology
PNQI	Invesco NASDAQ Internet ETF	USA	Internet

6 Backtesting on Market Data

A backtest of the two methodologies analyzed is conducted on the above described market data. The two models will be tested on the same pairs and same period, to have a clear comparison.

6.1 Training of the model

To build an effective trading strategy the process of researching the most promising pairs was crucial. The objective is to select the pairs that show the strongest mean-reversion behaviour. To achieve this, the PCI (Partially Cointegrated) model has been trained using the `fit.pci` function from the `partialCI` package in R. The model was trained for all possible pairs among the 67 available ETFs over a period of 48 months, employing a 6-month rolling window, as done in the CK paper, to enhance the power of the PCI tests. Specifically, the training began with a 48-month period, such as from January 2010 to December 2013. The model trained on this window was then used for trading decisions during the subsequent 6 months, from January 2014 to June 2014. Following this, the window was advanced by 6 months, with the model retrained on the updated period from June 2010 to June 2014, and applied to trading from July 2014 to December 2014, and this logic repeated for all the analyzed period.

For each pair, several key parameters were estimated and stored to characterize the relationship between the two time series. The parameter β represents the estimated coefficient that expresses the relationship between the two series, providing insight into their relative movements. Additionally, the parameter σ_κ denotes the standard deviation of the mean-reverting component. The standard deviation of the random walk component is denoted by σ_ζ, which captures the variability in the random walk component of the spread. Furthermore, the autoregressive coefficient, represented by ρ, measures the strength of the mean-reversion process. Finally, R^2_ψ indicates the proportion of variance explained by the mean-reverting component.

For the KFB approach the training/testing follows the same logic: the training set will be use to estimate the initial distribution of $\beta_{1|0}$ via OLS and the parameter σ_v and σ_w via MLE.

6.2 Selection Algorithm

Once the model was fitted to all ETFs pairs for each 4 year period, a selection logic based on strict criteria was applied to identify the pairs that could provide the best trading opportunities in the subsequent 6 months period.

Specifically, pairs were selected if they met the following conditions:

- ρ **between 0.9 and 0.98**: to have a half-life of mean reversion between 7 and 35 days.
- R^2_ψ **Greater Than 0.8:** This criterion ensures that most of the pair's variance is explained by the mean-reverting component
- Discarded pairs when the log likelihood is positive: to ensure that the fit of the model is sufficiently good.

KFB approach is applied to the same selected stocks in the same trading period.

In section .11 (APPENDIX C), a series of tables will be presented, covering datasets from "2014 H1" through "2023 H2". These tables will illustrate the trained parameters for each selected pair in their relative test sets.

The dataset names refer to the dataframes used for testing and should be understood as follows: for example, "filtered data 2014 H1" means that the parameters were trained on data from January 1, 2010, to December 31, 2013 (previous 48 months), and these parameters will be applied to trading from January 1, 2014, to June 30, 2014 (next 6 months).

Since there are no constraint on the number of pairs that it is possible to trade in a test set, this number is variable over time, meaning that the strategy could have different capital need in time. Note that with 67 times series there are 2211 possible pairs. Table 6.1 presents the number of pairs traded in each test sets.

Year	Semester 1	Semester 2
2014	11	11
2015	14	9
2016	11	8
2017	30	19
2018	12	17
2019	25	36
2020	27	19
2021	9	12
2022	8	6
2023	2	7

Table 6.1: Number of pairs per Semester

The number of pairs varies from a minimum of 2 in 2023 H1 to a maximum of 36 in 2019 H2, that corresponds in a range going from 0.09 % to 1.6 % of all the possible pairs.

6.3 Application on Single Pair

In this section it is presented an example of trading a single pair with the two strategies. The application of the PCI model on the EWA-EWH pair on the training set from 2010-01-01 to 2013-12-31 yielded the following parameters:

$$R^2_\psi = 0.94, \quad \rho = 0.97, \quad \sigma_\psi = 0.13, \quad \sigma_\tau = 0.03$$

These values, with R^2_ψ representing a high proportion of variance explained from *mean reverting component*, and a ρ that is suggesting an half-life of mean reversion of 22 days is indicating a promising relation to be exploited from the strategy. A backtest was conducted to evaluate the performance of both the PCI and KFB strategies on the test set, spanning from January 1, 2014, to June 30, 2014.

6.3.1 Trading assumptions

- **Transaction Costs:** According to Avellaneda and Lee (2010), Clegg and Krauss (2018), transaction costs are assumed to be 0.05% per share per half-turn. This means that each time a long or short position on the spread is opened and subsequently unwound, a cost of 0.0005 * 2 is subtracted from the returns of that day.

- **Risk-Free Rate:** A risk-free rate of 2% was used for Sharpe ratio calculations, in line with 10 y treasury yield in 2014.

- **Z-Score Threshold:** A Z-score threshold of ±1 was used to generate trading signals:

6.3.2 Results

Figure 6.1 and 6.2 show a visual representation of the strategy and the generation of the trading signals respectevely from PCI and KFB strategy: In the first graph of figure 6.1 the black line represents the Z-score function applied to $\hat{\psi}_{t|t}$ from PCI model. The horizontal dotted lines at ±1 represent the threshold levels. When the Z-score exceeds these boundaries, trading signals are triggered.
The red line represents the trading signal:

- When the Z-score crosses above +1, the signal will assume value -1 indicating to open a short position on the spread.

- When the Z-score crosses below -1, the signal will assume value +1 indicating to open a long position on the spread.

- When the Z-score reverts back towards 0, the signal will take the value 0 indicating to unwind the previous long/short position, if previously the signal has been different from ±1

The second graph of figure 6.1 shows the cumulative profit and loss (P&L) from applying the pairs trading strategy based on the PCI model on the EWA-EWH spread.

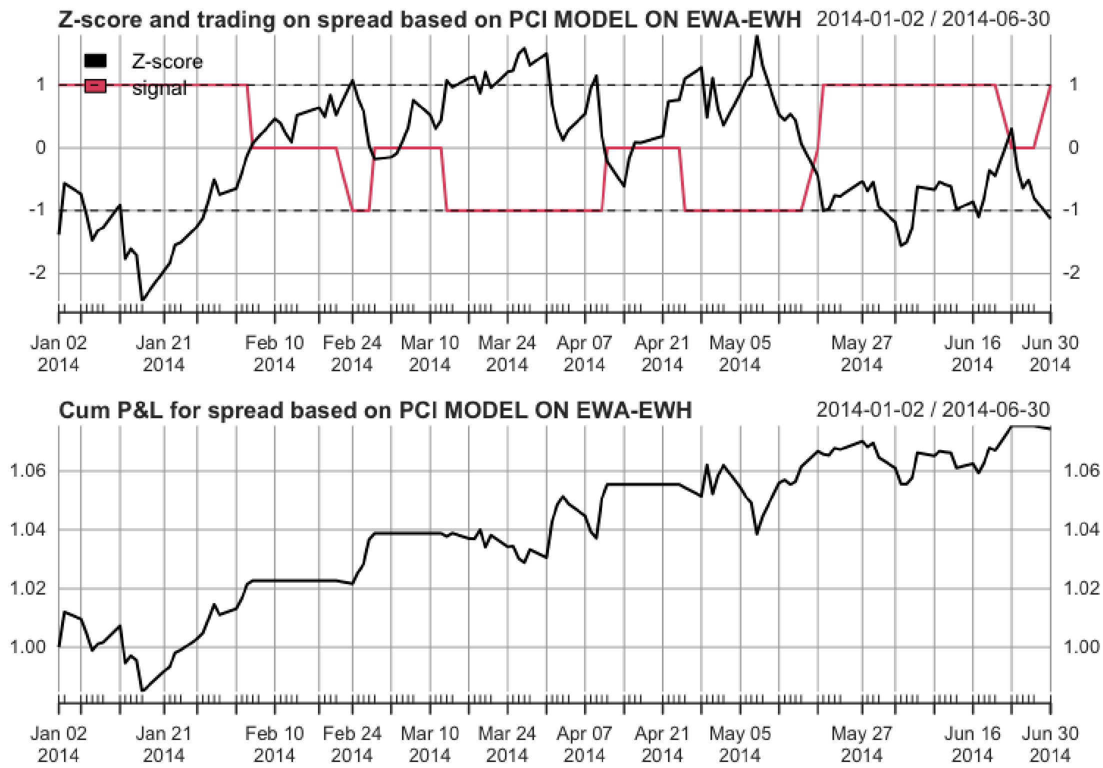

Figure 6.1: PCI strategy on EWA-EWH test set: 2014-01-01 2014-06-30

- The upward trend indicates that the strategy is profitable, as the cumulative P&L increases over time, reaching at the end of test set a profit around 7%.
- Periods of flat or downward movement reflect times when the strategy either experienced no trades or losses, corresponding to market conditions where the spread did not revert as expected.

Figure 6.2 is composed by the analogue graph with the difference that it reflects the signals generated by the KFB model. In this case the black line represent the Z-score function applied to the spread $\hat{z}_{t|t}$. The function to generate signals and to plot the cumulative P&L is the same as in figure 6.1.

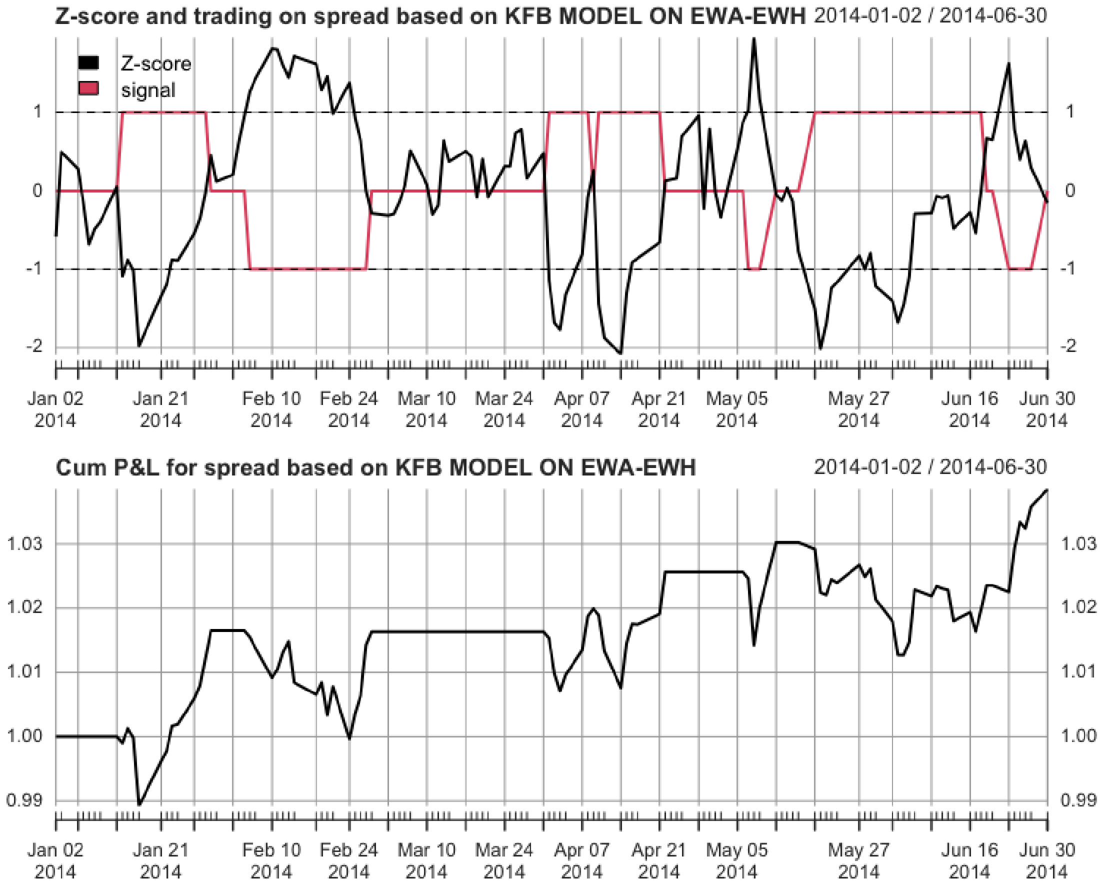

Figure 6.2: KFB strategy on EWA-EWH test set: 2014-01-01 2014-06-30

- The strategy shows an upward trend, reaching at the end of test set a profit around 4%.

Strategy	Total Return (%)	Annualized Return (%)	Sharpe Ratio (%)	Annualized St Dev (%)
PCI	7.42	15.66	1.88	6.70
KFB	3.86	8.00	1.09	5.22

Table 6.2: Performance Metrics of PCI and KFB Strategies (2014-01-02 to 2014-06-30)

Table 6.3 shows the performance metrics explained in section .4 for both model based on the signal from figures 6.1 and 6.2. In this particular pair and time frame, the PCI model demonstrates an exceptional risk/return profile, significantly outperforming the KFB model, which still maintains a commendable risk-return performance. Despite these promising results, a statistical arbitrage strategy cannot be evaluated solely based on one pair in a particular time frame. We need to extend the experiment to more pairs and over multiple years to assess whether the methodology is consistent.

The signals are generated by the algorithm described by the function `generate_signal` in section .10, which is then used in the function `pairs_trading_PCI_tr` in section .11 for calculating returns and generating figures 6.1, 6.2. The function for the PCI method takes as input the quantity $\hat{\psi}_{t|t}$ and the $\hat{\beta}$ estimated from the PCI model to generate the signals.

On the other hand, for the KFB method, a similar function is used, but the signals are generated based on the estimated spread $\hat{z}_{t|t}$ derived from the time-varying $\hat{\beta}_{t|t}$ obtained via the Kalman filter.

6.4 Portfolio Performance

This back-test described in the previous section have been extended to an equally weighted portfolio that contains all the stock pairs selected, as described in Section 6.2. In this way we are able to evaluate the performance across multiple pairs and over 10 years of trading, ensuring that the strategy is robustly evaluated.

The total returns presented in table 6.3 for the two strategies were calculated as the overall cumulative returns for each year. These strategies are then compared to the benchmark, represented by the SWDA ETF, which tracks the MSCI index for developed countries.

Year	Metric	PCI	KFB	SWDA
2014	Total Return (%)	3.41	3.31	11.92
	Sharpe Ratio	0.45	0.43	0.87
	Annualized SD (%)	3.15	3.10	10.91
2015	Total Return (%)	21.14	13.20	2.94
	Sharpe Ratio	3.44	3.57	0.06
	Annualized SD (%)	5.09	2.97	16.20
2016	Total Return (%)	13.62	5.14	27.22
	Sharpe Ratio	3.28	1.12	1.52
	Annualized SD (%)	3.35	2.78	14.78
2017	Total Return (%)	2.68	2.86	11.26
	Sharpe Ratio	0.34	0.49	0.88
	Annualized SD (%)	2.12	1.81	10.16
2018	Total Return (%)	5.40	-4.85	0.06
	Sharpe Ratio	0.69	-1.78	-0.13
	Annualized SD (%)	4.87	3.97	14.90
2019	Total Return (%)	11.59	6.78	22.21
	Sharpe Ratio	3.13	1.46	1.45
	Annualized SD (%)	2.92	3.20	12.69
2020	Total Return (%)	9.36	7.36	9.39
	Sharpe Ratio	0.36	0.48	0.29
	Annualized SD (%)	19.46	10.86	24.19
2021	Total Return (%)	4.61	5.67	20.93
	Sharpe Ratio	0.64	1.18	1.46
	Annualized SD (%)	4.11	3.08	11.92
2022	Total Return (%)	6.50	-3.52	-8.57
	Sharpe Ratio	0.66	-1.01	-0.64
	Annualized SD (%)	6.85	5.65	17.56
2023	Total Return (%)	5.73	4.71	18.37
	Sharpe Ratio	0.80	0.70	1.31
	Annualized SD (%)	4.70	3.99	11.82

Table 6.3: Performance metrics of PCI, KFB, and SWDA from 2014 to 2023.

From the table below we can argue that PCI consistently shows higher total returns compared to KFB, particularly in years like 2015 and 2019. However, SWDA, frequently outperforms both models in several year.

From the risk perspective, PCI exhibits a higher annualized standard deviation in certain years, indicating higher volatility compared to KFB. For instance, in 2020, PCI's standard deviation peaked at 19.46%, suggesting significant risk exposure during turbulent market conditions. KFB generally maintains lower volatility. However, both models display less volatility than the benchmark.

In the following figure 6.3 and table 6.4 the returns of PCI, KFB and SWDA were calculated as the cumulative returns for the overall period. Figure 6.3 provides a visual representation of the cumulated returns for the three assets over the entire period analyzed (2014-2023).

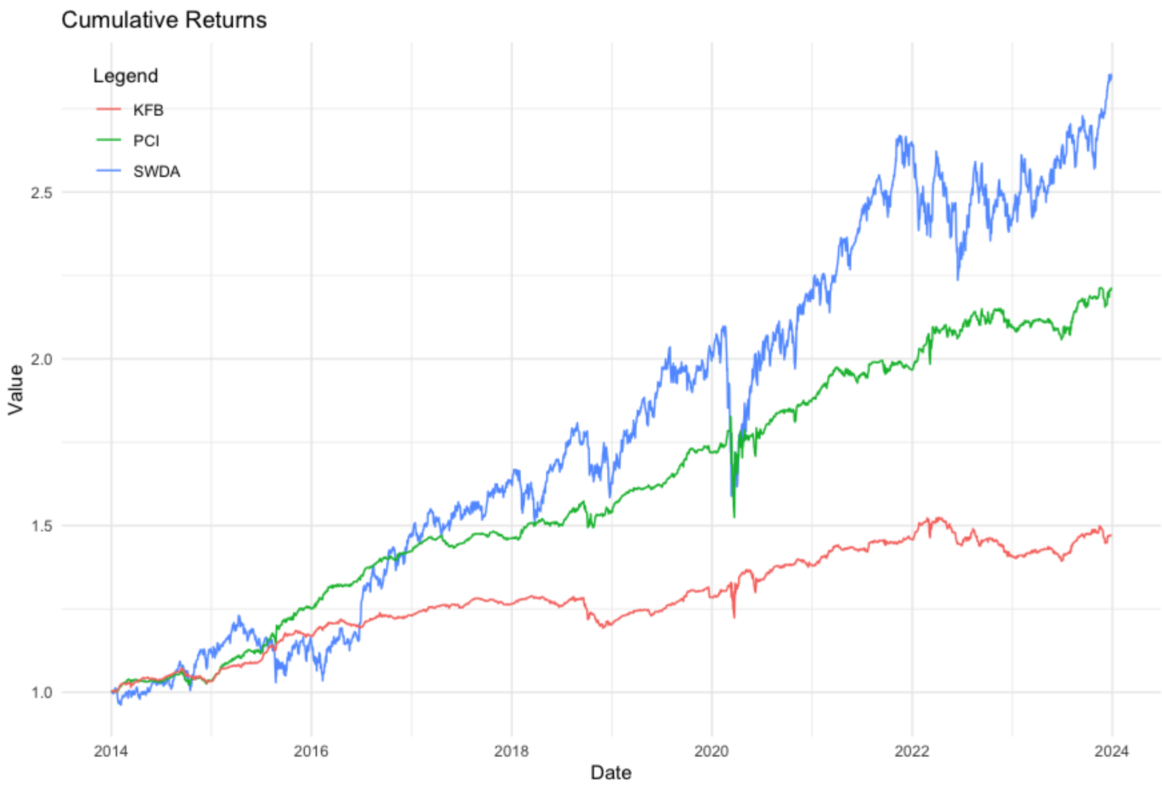

Figure 6.3: Cumulative returns

Metric	PCI	KFB	SWDA
Total Return (%)	121.42	47.27	185.48
Annualized Return (%)	8.45	4.03	11.30
Sharpe Ratio	0.82	0.41	0.58
Annualized SD (%)	7.41	4.81	15.03

Table 6.4: Overall performance metrics for PCI, KFB, and SWDA

From the overall performance, it is possible to asssert that the benchmark fund achieved the highest total returns, but PCI had a higher Sharpe ratio, indicating better risk-adjusted performance. While SWDA was more volatile, PCI delivered a solid return with lower volatility, making this strategy more efficient from risk-return perspective. KFB had the lowest return and Sharpe ratio, underperforming in both absolute and risk-adjusted terms. Figure 6.4 shows an histogram of monthly returns. The distribution of PCI and KFB is clearly leptokurtic while SWDA shows fatter tails. This means that PCI and KFB monthly returns are more concentrated around the mean, with fewer extreme deviations.

Figure 6.4: Histogram of monthly returns

6.4.1 Comments

Observing the described results, we can say that the "mean-reverting" component of the spread in the PCI model generates signals that outperform the KFB method under the analysed settings. This is consistent with the findings outlined in Chapter 4, where the performances of both models were compared on simulated data. In the simulation, the PCI model showed an outstanding ability to infer the state of $\hat{\psi}_{t|t}$, suggesting that this strategy would have performed well when the underlying process, as described by CK, was respected. In contrast, the KFB method's inference of the underlying process under CK assumptions led to less precise forecasts.

The application of the strategy across a wide range of assets and over 10 years of trading, further highlights that CK's assumptions about the underlying data generating process for the selected stocks was fairly alingned with the market data. On the other hand, the under-performance of the KFB method can likely be attributed to its less accurate inference of the current state of the spread, given the underlying data generating process. When comparing to the benchmark (SWDA), while PCI has a lower total return for the period 2014-2023, it demonstrates a superior risk-return profile, as indicated by its higher Sharpe ratio. On the other hand, KFB is unable to outperform its benchmark under risk-return profile.

It is worth noting that using a 6-month test set for the KFB method may not be optimal: the strength of KFB lies in its ability to adjust $\hat{\beta}_{t|t}$ over time, but within a short time frame like 6 months, relationships between assets typically don't change significantly. Therefore, it would be more appropriate to test the KFB method over a longer period to allow for a more meaningful evaluation of its ability to capture dynamic changes.

6.5 Exposure to Systematic Sources of Risk

This section analyses whether the exposure of the strategy to systematic risk, in order to evaluate if the strategy are indeed independent from the market movements. To pursue this scope we use Capital Asset Pricing Model (CAPM) framework, having as reference market SWDA ETF.

6.5.1 CAPM Regression Analysis

The CAPM model explains the relationship between the returns of PCI and KFB relative to a market benchmark, in this case, the SWDA index. Table 6.5 shows the results of the CAPM:

Strategy	Variable	Coefficient	Std. Error	t Value	p-Value
KFB	Intercept	0.00006241	0.00006040	1.033	0.302
	MKT	0.04558	0.00638	7.150	< 2e-16
PCI	Intercept	0.0002067	0.00009191	2.249	0.0246
	MKT	0.10580	0.00970	10.905	< 2e-16
KFB R-squared: 0.0203					
PCI R-squared: 0.04598					

Table 6.5: CAPM Regression Results for PCI and KFB Strategies

6.5.2 Findings

- **KFB Strategy:** The KFB strategy has a beta coefficient of 5%, indicating a small sensitivity to market movements. The value of the intercept, has non-significant p-value, meaning that this strategy is not able to generate returns that outperform his benchmark within its risk-return profile.

- **PCI Strategy:** The value of beta is around 11%, indicating a low exposure to market movements. On the other hand, for PCI model the value of the intercept is positive and has significant p-value, meaning that this strategy is able to outperform its relative benchmark under risk-return profile.

- **Note:** The regression was performed using **daily log-returns**. In order to compare the alpha of the PCI strategy with the results in the CK paper, we need to convert the daily alpha into a monthly equivalent. The correct transformation for log-returns, given that there are 21 trading days in a months is:

$$\alpha_{\text{monthly}} = \exp(\alpha_{\text{daily}} \times 21) - 1$$

Substituting the daily alpha value from Table 6.5 PCI results in a monthly alpha of 0.43%. This value is comparable with the 0.7% significant monthly alpha reported in the CK paper, even thought in the paper it was computed on a 3-factor model.

6.5.3 Interpretation

Both regressions have very low R-squared, respectevely 4.6% for PCI and 2.0% for KFB, indicating that only a small portion of their returns is explained by market exposure alone. These findings support the fact that this return could be considered market neutral. This characteristic is attractive for an investor who is seeking to hedge from his market exposure. The PCI strategy's significant alpha indicates its ability to generate returns beyond what would be expected based on its risk profile. In contrast, the KFB strategy's non-significant alpha shows it does not consistently outperform the market after accounting for its minimal market risk exposure.

7 Conclusions

After rigorous backtests on both simulated and market data, we found that PCI outperformed KFB, showing a better risk-return profile. Differently from the finding of CK paper, which suggested that PCI strategy had reached its saturation in S&P market, we found that this strategy is still able to generate market neutral returns if alternative assets are considered. This different finding could be due to the fact that ETFs, given their intrinsic composition have lower probability to lose their fundamental relation, differently from individual stocks, where the transient components are relevant. Additionally, this study employed more stringent criteria for selecting trading pairs, which may have further contributed to the improved performance. The dataset also included a period of heightened volatility, which can benefit mean reversion strategies. During such periods, assets are more likely to be mispriced, providing more opportunities for profitable trades.
On the other hand, the KFB's risk-return performance was lower than PCI, confirming the finding from chapter 4, where we measured a lower accuracy of this method in inferring the actual state of the spread when the d.g.p. was in line with CK assumptions.

7.1 Limitations of the study

The study presents the following limitations

- **Test Set Duration for KFB**: The choice to have a test set of 6 months for KFB may not be optimal, given the ability of this approach to adapt dynamically its parameters over time. Unlike the PCI model that have a constant cointegration parameter, that has to be recalibrated on a short time period, since the KFB has a time-varying coefficient could be tested on a longer period to leverage on its main strength.

- **Market microstructure**: A significant limitation of this study is that is not accounting for market microstructure dynamics. Since the back-test is conducted on closing price, it is not possible to assess real trading conditions, such as the bid-ask spread that we will have to pay for every transactions, and the effect of slippage that we will have if we consider to have a high amount of capital. These hidden costs could also get worst in a situation of high volatility, when liquidity could tight, and consequently the bid-ask and slippage could have an higher impact.

  This point could offer hints for future research: however, modeling slippage is challenging due to the fact that order book data are often hard to obtain. In any case enhancing the back test considering slippage effects could be crucial for simulating real-world conditions and assessing strategy's robustness during extreme market events.

7.2 Final thoughts

While we outlined the validity of PCI model, is important to acknowledge that these strategies could reach rapidly their saturation, as more market participants are aware of these methodologies. The competition in the field of quantitative finance is extremely high and the necessity of continuous developments is fundamental.
By addressing existing limitations and exploring new methodologies, researchers and practitioners can develop more resilient strategies that thrive in diverse market conditions, ensuring that quantitative hedge funds contribute significantly to price discovery and improve liquidity in the markets, reducing transaction costs and ensuring market stability.

Bibliography

[1] G. S. Maddala and I.-M. Kim, *Unit Roots, Cointegration, and Structural Change*, Cambridge University Press, 1998.

[2] J. D. Hamilton, *Time Series Analysis*, Princeton University Press, 1994.

[3] Clegg, M. and Krauss, C. (2018). *Pairs trading with partial cointegration*. *Quantitative Finance*, 18(1):121–138.

[4] Chan, E. P. (2009). *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley.

[5] Vidyamurthy, G. (2004). *Pairs Trading: Quantitative Methods and Analysis*.Wiley.

[6] Avellaneda, M., Lee, J.-H. (2010). *Statistical Arbitrage in the US Equities Market*. *Quantitative Finance*, 10(7), 761–782.

[7] Brogaard, J., Hendershott, T., Riordan, R. (2014). *High-Frequency Trading and Price Discovery*. *Review of Financial Studies*, 27(8), 2267–2306.

[8] Granger, C. W. (1986). *Developments in the study of cointegrated economic variables*. *Oxford Bulletin of Economics and Statistics*, 48(3), 213–228.

[9] Clegg, M. (2014). *On the Persistence of Cointegration in Pairs Trading*. SSRN Electronic Journal. Available at: `http://ssrn.com/abstract=2491201`

[10] Clegg, M. (2015). *Modeling Time Series with Both Permanent and Transient Components Using the Partially Autoregressive Model*. SSRN Electronic Journal. Available at: `http://ssrn.com/abstract=2556957`

[11] Dickey, D. A., Fuller, W. A. (1979). *Distribution of the Estimators for Autoregressive Time Series with a Unit Root*. *Journal of the American Statistical Association*, 74(366), 427–431.

[12] Fama, E. F., French, K. R. (2015). *A Five-Factor Asset Pricing Model*. *Journal of Financial Economics*, 116(1), 122–146.

[13] De Moura, C. E., Pizzinga, A., and Zubelli, J. (2016). *A Pairs Trading Strategy Based on Linear State Space Models and the Kalman Filter*. *Quantitative Finance*, 16(2), 115–130.

[14] Simon, D. (2006). *Optimal State Estimation: Kalman, H, and Nonlinear Approaches*. John Wiley Sons, Hoboken, N.J.

[15] Johansen, S. (1988). *Statistical Analysis of Cointegration Vectors*. *Journal of Economic Dynamics and Control*, 12(2-3), 231–254.

[16] Peterson, B. G., and Carl, P. (2014). *PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis*. R package. Available at: `http://CRAN.R-project.org/package=PerformanceAnalytics`

[17] *partialCI: Partial Cointegration Analysis*. R package. Available at: `https://cran.r-project.org/src/contrib/Archive/partialCI/` and github `https://github.com/matthewclegg/partialCI`

[18] Pástor, Ľ., & Stambaugh, R. F. (2001). *Equity Risk Premia*. *Journal of Financial Economics, 63*(3), 375–411.

[19] Pole, A. (2007). *Statistical Arbitrage: Algorithmic Trading Insights and Techniques*. John Wiley & Sons.

[20] Krauss, C. (2015). *Statistical Arbitrage Pairs Trading Strategies: Review and Outlook*. IWQW Discussion Papers, No. 09/2015. Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics.

[21] Murray, M. P. (1994). *A Drunk and Her Dog: An Illustration of Cointegration and Error Correction*. American Economic Review, 84(2), 103-108.

APPENDIX A: Definitions

.1 Definition of partial cointegration

Definition: The components of the vector $\mathbf{X}_t$ are said to be partially cointegrated of order (d, b), denoted $\mathbf{Y}_t \sim \text{PCI}(d, b)$, if

- All components of $\mathbf{Y}_t$ are $I(d)$;
- There exists a vector $\alpha \neq 0$ such that $S_t = \alpha' \mathbf{Y}_t$ and S_t can be decomposed as a sum $S_t = \tau_t + \psi_t$, where $\tau_t \sim I(d)$ and $\psi_t \sim I(d-b)$.

Given two times series $X_1 = (X_{1t})_{t=1}^T$ and $X_2 = (Y_{2t})_{t=1}^T$. We say that y_1 and y_2 are partially cointegrated if β, ρ, σ_κ, σ_ζ can be find such that the following model is satisfied:

$$
\begin{aligned}
y_{1t} &= \beta y_{2t} + s_t \\
s_t &= \tau_t + \psi_t, \\
\tau_t &= \tau_{t-1} + \zeta_t, \quad \zeta_t \sim \text{i.i.d.}\, N(0, \sigma_\zeta^2), \\
\psi_t &= \rho \psi_{t-1} + \kappa_t, \quad \kappa_t \sim \text{i.i.d.}\, N(0, \sigma_\kappa^2),
\end{aligned}
$$

Where $\beta \in \mathbb{R}$ is a parameter, $\rho \in (-1, 1)$ is the AR(1) coefficient, and ζ_t, κ_t follow mutually independent Gaussian white noise processes with expectation zero and variances σ_κ^2 and $\sigma_\zeta^2 \in \mathbb{R}^+$.

.2 Prof of R_ψ^2

The variance of the differenced series can be expressed as:

$$\text{Var}[\Delta s_t] = \text{Var}[\Delta\psi_t] + \text{Var}[\Delta\tau_t] \tag{1}$$

Given that:

$$\text{Var}[\Delta\tau_t] = \sigma_\zeta^2 \tag{2}$$

$$\text{Var}[\Delta\psi_t] = \frac{2\sigma_\kappa^2}{\rho + 1} \tag{3}$$

Substituting these into the variance of Δs_t:

$$\text{Var}[\Delta s_t] = \frac{2\sigma_\kappa^2}{\rho + 1} + \sigma_\zeta^2 \tag{4}$$

The proportion of variance attributable to mean reversion, R_ψ^2, is then given by:

$$R_\psi^2 = \frac{\text{Var}[\Delta\psi_t]}{\text{Var}[\Delta s_t]} \tag{5}$$

$$= \frac{\frac{2\sigma_\kappa^2}{\rho+1}}{\frac{2\sigma_\kappa^2}{\rho+1} + \sigma_\zeta^2} \tag{6}$$

This can be simplified further:

$$R_\psi^2 = \frac{2\sigma_\kappa^2}{2\sigma_\kappa^2 + (\rho + 1)\sigma_\zeta^2} \tag{7}$$

.3 Half-Life of the AR(1) Process

Suppose the process x_t, follows a covariance-stationary AR(1) process:

$$x_t = \rho x_{t-1} + \kappa_t, \qquad |\phi_1| < 1.$$

If we are at time t and want to make a prediction h time units ahead, denoted as $\hat{r}_t(h)$, then $\hat{r}_t(h) = \mathsf{E}[r_{t+h} \mid \mathcal{F}_t]$, where $\mathcal{F}_t$ is the σ-algebra of all information available by time t, assuming we use the mean squared error method.

We seek to determine the speed of mean reversion, quantified in literature by the half-life. The 'half life of mean reversion' is the average time it will take a process to get pulled half-way back to the mean. By setting $\hat{x}_t(h) := \hat{r}_t(h) - \mathsf{E}[r_t]$, we obtain:

$$\hat{x}_t(h) = \rho \hat{x}_{t-1}(h),$$

which implies:

$$\hat{x}_t(h) = \rho^h x_t.$$

If h represents the half-life, then:

$$\hat{x}_t(h) = \frac{1}{2} x_t.$$

This leads to the following expression for the half-life:

$$h = \frac{\log 0.5}{\log |\rho|}.$$

Figure 8.1 illustrates the relationship between the parameter ρ (on the x-axis) and the average time it takes for a process to revert halfway to its mean (on the y-axis), known as the half-life of mean reversion. As ρ ranges from 0.8 to 0.98, this time increases from approximately 5 to 35 days. As ρ approaches 1, the half-life tends to infinity, indicating that the process becomes a Random Walk, which lacks mean-reversion properties.

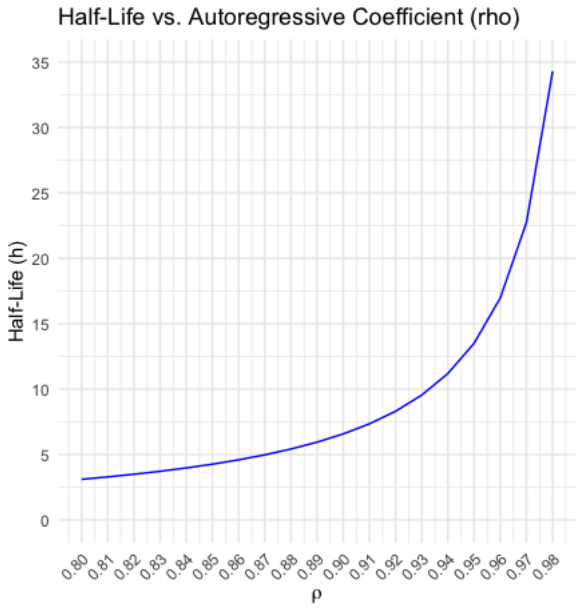

Figure 1: Half-Life as a Function of ρ. The graph illustrates how the half-life h varies with the autoregressive coefficient ρ in the range of 0.8 to 0.98.

.4 Portfolio Performance Metrics

To measure the perfomance of the strategies we used the following metrics: total return, annualized return, Sharpe ratio, daily and annualized standard deviation. These metrics are computed as follows:

- **Total Return**: For log returns, the total return over the period is calculated by the sum:

$$\text{Total Return} = \exp\left(\sum_{i=1}^{n} r_i\right) - 1$$

  Note that r_i is the log-return of the day i and n is the total number of days in the return series.

- **Average Daily Return**: The average daily return is simply the mean of the log returns:

$$\text{Average Daily Return} = \frac{1}{n}\sum_{i=1}^{n} r_i$$

- **Annualized Return**: To convert daily log-return in annual base:

$$\text{Annualized Return} = \exp(252 \times \text{Average Daily Return}) - 1$$

  Note that we assume 252 trading days in a year.

- **Sharpe Ratio**: To obtain a risk-adjusted measure of the returns:

$$\text{Sharpe Ratio} = \frac{\text{Average Daily Return} - \text{Daily Risk-Free Rate}}{\text{Daily Standard Deviation of Returns}} \times \sqrt{252}$$

  where the daily risk-free rate is converted from an annual rate using:

$$\text{Daily Risk-Free Rate} = (1 + \text{Annual Risk-Free Rate})^{1/252} - 1$$

  Note that this ratio is undefined when stdev is 0.

- **Daily Standard Deviation of Returns**: Measure the daily "volatility" of the portfolio:

$$\text{Daily Standard Deviation} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_i - \text{Average Daily Return})^2}$$

- **Annualized Standard Deviation**: To convert daily standard deviation in annual base:

$$\text{Annualized Standard Deviation} = \text{Daily Standard Deviation} \times \sqrt{252}$$

APPENDIX B: R codes

.5 Kalman Gain Function

```
kalman_gain <- function(rho, sigmaM, sigmaR) {
  denom <- (sigmaR*(sqrt((rho + 1)^2 * sigmaR^2 + 4 * sigmaM^2) +
     rho * sigmaR + sigmaR) + 2 * sigmaM^2)

  K1 <- 2 * sigmaM^2 / denom
  K2 <- 2* sigmaR / (sqrt((rho + 1)^2 * sigmaR^2 + 4 * sigmaM^2) -
     rho * sigmaR + sigmaR)

  return(c(K1, K2))
}
```

Listing 1: Kalman Gain Function

.6 Kalman Estimate Function

```
kalman_estimate <- function(X, rho, sigmaM, sigmaR) {
  # Calculate Kalman gain
  K <- kalman_gain(rho, sigmaM, sigmaR)

  # Initialize vectors for estimates
  M <- numeric(length(X))
  R <- numeric(length(X))

```

```
  # Initial values
  M[1] <- 0
  R[1] <- X[1]

  # Iterate through observations
  for (i in 2:length(X)) {
    # Predicted value
    xhat <- rho * M[i - 1] + R[i - 1]

    # Prediction error
    e <- X[i] - xhat

    # Update estimates using Kalman gain
    M[i] <- rho * M[i - 1] + e * K[1]
    R[i] <- R[i - 1] + e * K[2]
  }

  # Return the estimated states
  return(list(M = M, R = R))
}
```

Listing 2: Kalman Estimate Function

.7 PCI Model Training and Testing Function

```
PCI_train_test <- function(Y_train, Y_test) {
  # Fit the model on the training data
  train <- fit.pci(Y_train)
  h<- statehistory.pci(train)

  # Extract the trained beta parameter
  beta <- train$beta

```

```
  # Extract  parameters
  rho <- train$rho
  sigma_M <- train$sigma_M
  sigma_R <- train$sigma_R
  sd<-sd(h$M) #IN-SAMPLE sigma hat psi_t|t


  X_test <- Y_test[,1] - beta * Y_test[,2]
  X_test <- as.numeric(X_test)

  # Perform Kalman filtering on the test data
  result_test <- kalman_estimate(X_test, rho, sigma_M, sigma_R)


  M_t_xts <- xts(result_test$M, order.by = index(Y_test))
  R_t_xts <- xts(result_test$R, order.by = index(Y_test))


  return(list(
    beta = beta,
    M_t = M_t_xts,
    R_t = R_t_xts
    sd=sd
  ))
}
```

Listing 3: PCI Model Training and Testing Function

.8 KFB implementation

```
estimate_beta_KFB<- function(Y, training_period, smoothing_param) {
  estimate_beta_LS <- function(Y_train) {
    lm_fit <- lm(Y_train[, 1] ~ Y_train[, 2])
```

```
  return(list(beta = coef(lm_fit)[2]))  # Extract the slope as beta
}

T <- nrow(Y)

# Initialize empty xts for storing beta estimates
beta_Kalman_smoothing <- xts(rep(NA, T), index(Y))
colnames(beta_Kalman_smoothing) <- "beta-Kalman"

# Estimate initial beta using least squares on the specified training period
Y_train <- Y[training_period, ]  # Select data for the specified training period
init <- estimate_beta_LS(Y_train)
a1 <- matrix(init$beta, 1, 1)  # Initial beta state
P1 <- 1e-4 * diag(1)  # Variance of initial point
P1inf <- 0 * diag(1)

# Create Kalman model for training period
model_train <- SSModel(
  as.matrix(Y_train[, 1]) ~ 0 + SSMcustom(
    Z = array(1, dim = c(1, 1, length(training_period))),  # p x m x T array
    T = matrix(1, nrow = 1, ncol = 1),  # State transition matrix
    R = matrix(1, nrow = 1, ncol = 1),  # State noise covariance matrix
    Q = matrix(1, nrow = 1, ncol = 1),  # Observation noise covariance matrix
    a1 = a1
  )
)

```

```
# Fit the model to estimate the parameters
fit_model <- fitSSM(model_train, inits = c(1e-8, 1e-8))  # Initial values for the variances

# Extract estimated parameters
estimated_params <- fit_model$optim.out$par
sigma_v <- estimated_params[1]  # Observation variance
sigma_w <- estimated_params[2]  # State transition variance

# Create Kalman model for the full dataset using estimated parameters
Ht <- matrix(sigma_v)
Qt <- sigma_w * matrix(1, nrow = 1, ncol = 1)
model <- SSModel(
  as.matrix(Y[, 1]) ~ 0 + SSMcustom(
    Z = array(as.vector(Y[, 2]), dim = c(1, 1, T)),  # p x m x T array
    T = matrix(1, nrow = 1, ncol = 1),  # State transition matrix
    R = matrix(1, nrow = 1, ncol = 1),  # State noise covariance matrix
    Q = Qt,
    a1 = a1,
    P1 = P1,
    P1inf = P1inf
  ),
  H = Ht
)

# Run Kalman filtering and smoothing
out <- KFS(model)

# Extract smoothed beta estimates (i.e., the estimated actual state)
```

```
  beta_Kalman_f[] <- out$alphahat[, 1]

  # Handle missing values
  beta_Kalman_f <- na.locf(beta_Kalman_f, fromLast = TRUE)

  # Initialize beta_Kalman_filtering
  beta_Kalman_filtering <- xts(rep(NA, T), index(Y))  # Initialize
     as empty xts

  # Apply rolling mean with the given smoothing parameter
  beta_Kalman_filtering[] <- rollapply(as.numeric(beta_Kalman_f),
     width = 30, FUN = mean, fill = NA, align = "right")
  beta_Kalman_filtering <- na.locf(beta_Kalman_filtering, fromLast
     = TRUE)

  return(list(beta = beta_Kalman_filtering))
}
```

Listing 4: Estimate Beta Using Kalman Filter Function

.9 Create simulated data for chapter 4

```
set.seed(123)
# Parameters for simulation
T <- 1642  # Number of time points
sigma_zeta <- 0.1  # Standard deviation of zeta_t
sigma_kappa <- 0.9  # Standard deviation of kappa_t
rho <- 0.96  # AR(1) coefficient for psi_t
rho_beta <- 1

tau <- numeric(T)
psi <- numeric(T)
s <- numeric(T)
```

```r
y2 <- cumsum(rnorm(T))
y1 <- numeric(T)
beta_true <- numeric(T)

# Initial values for tau, psi, and beta
tau[1] <- rnorm(1, 0, sigma_zeta)
psi[1] <- rnorm(1, 0, sigma_kappa)
beta_true[1] <- 0  # Initial beta value

# Simulate the processes
for (t in 2:T) {
  zeta_t <- rnorm(1, 0, sigma_zeta)
  kappa_t <- rnorm(1, 0, sigma_kappa)

  tau[t] <- tau[t - 1] + zeta_t
  psi[t] <- rho * psi[t - 1] + kappa_t
  beta_true[t] <- rho_beta * beta_true[t - 1]
  s[t] <- tau[t] + psi[t]
  y1[t] <- beta_true[t] * y2[t] + s[t]
}
```

Listing 5: simulated data from CK model

.10 Generate Signals

```r
generate_signal <- function(Z_score, threshold_long, threshold_
    short) {
  signal <- Z_score
  colnames(signal) <- "signal"
  signal[] <- NA

  # Initial position
  signal[1] <- 0
```

```
  if (Z_score[1] <= threshold_long[1]) {
    signal[1] <- 1
  } else if (Z_score[1] >= threshold_short[1]) {
    signal[1] <- -1
  }

  # Loop to generate signals
  for (t in 2:nrow(Z_score)) {
    if (signal[t-1] == 0) {
      if (Z_score[t] <= threshold_long[t]) {
        signal[t] <- 1
      } else if (Z_score[t] >= threshold_short[t]) {
        signal[t] <- -1
      } else {
        signal[t] <- 0
      }
    } else if (signal[t-1] == 1) {
      if (Z_score[t] >= 0) signal[t] <- 0
      else signal[t] <- signal[t-1]
    } else {
      if (Z_score[t] <= 0) signal[t] <- 0
      else signal[t] <- signal[t-1]
    }
  }
  return(signal)
}
```

Listing 6: Generate signals

.11 Pairs Trading (PCI)

```
#COMPUTE THE LOG RETURNS OF THE PAIRS TRADING STRATEGY FROM PCI
   MODEL
```

```r
pairs_trading_PCI_tr <- function(Y, beta, name = NULL, threshold =
    0.5, transaction_cost = 0.001, plot = FALSE) {
  # Compute spread using the state history from the PCI model
  w_spread <- cbind(1, -beta) / cbind(1 + beta, 1 + beta)

  # Compute Z-score based on the PCI model spread
  Z_score <- generate_Z_score(result_PCI$M_t)
  threshold_long <- Z_score
  threshold_short <- Z_score
  threshold_short[] <- threshold
  threshold_long[] <- -threshold

  # Generate trading signals
  signal <- generate_signal(Z_score, threshold_long, threshold_
      short)

  # Portfolio weights
  w_portf <- w_spread * lag.xts(cbind(signal, signal), k = 1) #
      IMPORTANT: NOTE THE LAG!!!

  # Compute log-returns and portfolio returns
  X <- diff(log(Y))  # Compute log-returns from log-prices
  portf_return <- xts(rowSums(X * w_portf), index(X))

  # Identify the days where a new trade is initiated (signal
      changes from 0 to 1 or -1)
  previous_signal <- lag.xts(signal, k = 1)  # Previous day's
      signal NOTE THE LAG!!!
  new_trades <- (signal != 0) & (previous_signal == 0)
  closing_trades <- (signal == 0) & (previous_signal != 0)

  # Apply transaction costs only on the days a new trade is
      initiated or closed
```

```
  transaction_costs <- ifelse(new_trades | closing_trades,
     transaction_cost, 0)
  portf_return[new_trades | closing_trades] <- portf_return[new_
     trades | closing_trades] - transaction_costs[new_trades |
     closing_trades]

  # Replace NA values with 0 (initial day)
  portf_return[is.na(portf_return)] <- 0

  colnames(portf_return) <- name

  # plots
  if (plot) {
    tmp <- cbind(Z_score, signal)
    colnames(tmp) <- c("Z-score", "signal")
    par(mfrow = c(2, 1))
    { plot(tmp, legend.loc = "topleft",
           main = paste("Z-score and trading on spread based on",
              name))
      lines(threshold_short, lty = 2)
      print(lines(threshold_long, lty = 2)) }
        print(plot(exp(cumsum(portf_return)), main = paste("Cum P&L
            for spread based on", name)))  #NOTE  exp(cumsum(portf_
           return)) BECAUSE WE ARE DEALING WITH LOG RETURNS! Note
           that it is just for the plot, because the returns that
           are stored are log-returns!!
  }
  return(portf_return)
}
```

Listing 7: pairs trading

APPENDIX C: Trained Parameters

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWH	0.9425	0.97498	1.1248	0.1345	0.0334
2	EWA	EWW	0.9686	0.97901	0.2459	0.1329	0.0241
3	EWO	EIS	0.9475	0.9726	0.3056	0.1524	0.0361
4	EWC	EWY	1.0000	0.9794	0.2710	0.1634	0.0000
5	EWC	EWT	0.9400	0.9658	0.9180	0.1749	0.0446
6	EWC	RSX	0.8664	0.9243	0.5839	0.1413	0.0566
7	EWC	DBA	0.8069	0.9672	0.4894	0.2066	0.1019
8	EWG	XLB	0.8119	0.9732	0.5949	0.1485	0.0720
9	EWS	THD	0.8004	0.9627	0.1646	0.1220	0.0615
10	XLI	FDN	0.9759	0.9722	0.5768	0.1921	0.0304
11	DBC	DBA	0.8837	0.9791	0.7038	0.1805	0.0658

Table 1: Model Parameters for Selected Stock Pairs (Filtered Data 2014 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWH	0.9843	0.9662	1.0111	0.1376	0.0175
2	EWA	EWW	0.8211	0.9755	0.2288	0.1211	0.0569
3	EWA	EWS	0.9540	0.9789	0.8811	0.1130	0.0249
4	EWA	EWY	0.8057	0.9782	0.2159	0.1102	0.0544
5	EWA	EWT	0.8864	0.9778	0.7470	0.1241	0.0447
6	EWA	UGA	0.8034	0.9780	0.1096	0.1617	0.0804
7	EWO	EIS	0.8893	0.9762	0.2915	0.1410	0.0501
8	EWC	EWT	1.0000	0.9700	0.8716	0.1749	0.0000
9	EWU	XLB	0.9445	0.9338	0.5801	0.1402	0.0346
10	XLB	SOXX	0.8022	0.9626	1.2100	0.2144	0.1075
11	DBC	DBA	0.8498	0.9783	0.6568	0.1706	0.0721

Table 2: Model Parameters for Selected Stock Pairs (Filtered Data 2014 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWY	0.8207	0.9792	0.2091	0.1139	0.0535
2	EWA	AFK	0.8332	0.9744	0.5776	0.1259	0.0567
3	EWC	EWI	1.0000	0.9771	0.4571	0.1721	0.0000
4	EWC	EWP	0.9544	0.9791	0.4067	0.1777	0.0391
5	EWC	EWT	1.0000	0.9744	0.8180	0.1809	0.0000
6	EWC	EIS	0.8605	0.9781	0.3334	0.1745	0.0706
7	EWC	FEZ	0.8479	0.9746	0.4661	0.1491	0.0635
8	EWM	EWL	0.8798	0.9796	0.7828	0.2131	0.0792
9	EWM	EWU	1.0000	0.9794	0.7538	0.2095	0.0000
10	EWM	XLE	1.0000	0.9780	0.2691	0.2263	0.0000
11	EWS	TUR	0.8768	0.9554	0.1348	0.1334	0.0506
12	EWS	THD	0.8260	0.9709	0.1429	0.1208	0.0558
13	EWT	EIS	0.9009	0.9605	0.2536	0.1540	0.0516
14	EWT	AAXJ	0.8587	0.9431	0.3130	0.0937	0.0386

Table 3: Model Parameters for Selected Stock Pairs (Filtered Data 2015 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	TUR	0.8026	0.9735	0.1421	0.1415	0.0707
2	EWA	AFK	0.8933	0.9723	0.5908	0.1316	0.0458
3	EWC	EWP	0.9772	0.9787	0.4130	0.1772	0.0272
4	EWC	AFK	0.8574	0.9024	0.7881	0.1546	0.0646
5	EWQ	EIS	0.8250	0.9489	0.4027	0.1756	0.0819
6	EWM	AFK	0.8885	0.9666	0.8034	0.2206	0.0788
7	EWS	TUR	0.8595	0.9629	0.1356	0.1300	0.0531
8	EWT	EIS	0.8781	0.9591	0.2534	0.1540	0.0580
9	EWT	AAXJ	0.8320	0.9330	0.3075	0.0967	0.0442

Table 4: Model Parameters for Selected Stock Pairs (Filtered Data 2015 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	AFK	0.8857	0.9749	0.4964	0.1298	0.0469
2	EWH	EWL	0.9180	0.9781	0.3803	0.1296	0.0390
3	EWD	EZU	0.9242	0.9679	0.7062	0.1351	0.0390
4	EWD	FEZ	0.8990	0.9681	0.6659	0.1403	0.0474
5	EWU	NORW	0.9022	0.9465	1.1132	0.1320	0.0441
6	EWU	FEZ	0.9229	0.9584	0.5882	0.1238	0.0362
7	EWT	AAXJ	0.8878	0.9381	0.3141	0.1013	0.0366
8	EWT	XLB	0.9912	0.9729	0.3250	0.1543	0.0146
9	XLF	VTI	0.8264	0.9710	0.1965	0.0481	0.0222
10	XLF	IWV	0.8210	0.9689	0.1681	0.0480	0.0226
11	XLU	IYR	0.8075	0.9598	0.3867	0.1720	0.0848

Table 5: Model Parameters for Selected Stock Pairs (Filtered Data 2016 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWW	0.8099	0.9792	0.1899	0.1239	0.0604
2	EWA	EWS	0.8572	0.9708	0.7961	0.1177	0.0484
3	EWD	EZU	1.0000	0.9790	0.7065	0.1406	0.0000
4	EWD	FEZ	0.9900	0.9766	0.6810	0.1450	0.0147
5	EWU	NORW	0.8887	0.9368	1.2255	0.1416	0.0509
6	EWU	FEZ	0.9431	0.9481	0.6454	0.1344	0.0334
7	EWT	AAXJ	0.8941	0.9571	0.3233	0.1020	0.0355
8	EWT	XLB	0.9991	0.9788	0.3298	0.1575	0.0047

Table 6: Model Parameters for Selected Stock Pairs (Filtered Data 2016 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWC	1.0000	0.9786	0.5743	0.1297	0.0000
2	EWA	EWZ	0.8866	0.9489	0.2081	0.1421	0.0515
3	EWA	EEM	0.8184	0.9711	0.3510	0.1097	0.0521
4	EWA	ILF	0.8692	0.9532	0.3289	0.1321	0.0519
5	EWA	AFK	0.9173	0.9601	0.4679	0.1418	0.0430
6	EWA	XLE	0.9403	0.9711	0.1518	0.1437	0.0365
7	EWA	GSG	0.8027	0.9778	0.2584	0.1486	0.0741
8	EWK	EIS	0.8355	0.9393	0.1950	0.0990	0.0446
9	EWK	FDN	0.9754	0.9796	0.1050	0.1036	0.0165
10	EWO	GSG	0.8446	0.9788	0.1751	0.1257	0.0542
11	EWG	TAN	0.9898	0.9709	0.1576	0.2220	0.0227
12	EWG	ICLN	1.0000	0.9786	1.2720	0.1981	0.0000
13	EWG	PBW	0.9457	0.9722	0.3386	0.2111	0.0509
14	EWH	EWT	0.9599	0.9704	0.5016	0.1257	0.0259
15	EWH	EZA	0.8084	0.9769	0.1271	0.1234	0.0604
16	EWN	TAN	0.8739	0.9408	0.1397	0.1794	0.0692
17	EWN	ICLN	0.9024	0.9565	1.1530	0.1600	0.0532
18	EWS	EEM	0.8127	0.9720	0.3284	0.0883	0.0427
19	EWS	VWO	0.8016	0.9649	0.3444	0.0888	0.0446
20	EWD	EZU	1.0000	0.9777	0.7049	0.1382	0.0000
21	EWD	EIS	0.8435	0.9226	0.4096	0.2061	0.0905
22	EWD	FEZ	0.9948	0.9772	0.6903	0.1427	0.0104
23	EWD	ICLN	0.9681	0.9562	1.3538	0.2172	0.0398
24	EWD	PBW	0.9824	0.9503	0.3701	0.2346	0.0318
25	EWU	NORW	0.8419	0.9372	1.2092	0.1391	0.0613
26	EWU	FEZ	0.9363	0.9457	0.6611	0.1334	0.0353
27	EWU	TAN	0.9848	0.9634	0.1781	0.2281	0.0286
28	EWU	ICLN	0.8911	0.9581	1.4077	0.1917	0.0677
29	EWU	PBW	1.0000	0.9704	0.3918	0.2207	0.0000
30	EWT	XLB	0.9760	0.9785	0.3218	0.1663	0.0262

Table 7: Model Parameters for Selected Stock Pairs (Filtered Data 2017 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWC	0.9675	0.9616	0.5599	0.1252	0.0232
2	EWA	EWZ	0.8045	0.9264	0.1854	0.1338	0.0672
3	EWA	ILF	0.8110	0.9290	0.3055	0.1259	0.0619
4	EWA	AFK	0.8693	0.9276	0.4588	0.1357	0.0536
5	EWK	EIS	0.8177	0.9528	0.1936	0.0980	0.0468
6	EWH	EWT	0.9495	0.9709	0.4942	0.1243	0.0289
7	EWS	EEM	0.8640	0.9706	0.3245	0.0904	0.0361
8	EWS	VWO	0.8191	0.9658	0.3403	0.0894	0.0424
9	EWD	EWL	0.8805	0.9041	0.9477	0.1533	0.0579
10	EWD	EWY	0.8141	0.9419	0.2922	0.1911	0.0927
11	EWD	EZU	0.8946	0.9331	0.7023	0.1275	0.0445
12	EWD	EIS	0.8117	0.9287	0.4048	0.1981	0.0972
13	EWD	FEZ	0.9012	0.9459	0.6935	0.1317	0.0442
14	EWU	EIS	0.8273	0.9459	0.3932	0.1922	0.0890
15	EWU	NORW	0.8419	0.9252	1.2117	0.1375	0.0607
16	EWU	FEZ	0.9041	0.9416	0.6617	0.1304	0.0431
17	EWU	PBW	0.9513	0.9704	0.3977	0.2130	0.0486
18	EWT	AAXJ	0.8284	0.9472	0.3407	0.1023	0.0472
19	EWT	SOXX	0.8609	0.9733	0.3288	0.1588	0.0642

Table 8: Model Parameters for Selected Stock Pairs (Filtered Data 2017 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWC	0.9371	0.9523	0.5373	0.1203	0.0315
2	EWA	EWZ	0.8118	0.9255	0.1716	0.1314	0.0645
3	EWA	ILF	0.8347	0.9261	0.2941	0.1247	0.0566
4	EWA	AFK	0.8624	0.9043	0.4407	0.1314	0.0538
5	EWH	AAXJ	0.8097	0.9017	0.2662	0.0818	0.0407
6	EWS	EEM	0.8009	0.9747	0.3114	0.0862	0.0433
7	EWD	EWY	1.0000	0.9766	0.2712	0.2122	0.0000
8	EWD	EZU	1.0000	0.9532	0.6917	0.1329	0.0000
9	EWD	FEZ	0.9716	0.9519	0.6901	0.1343	0.0233
10	EWU	FEZ	0.8736	0.9473	0.6625	0.1279	0.0493
11	EWU	PBW	0.9304	0.9673	0.4177	0.2067	0.0570
12	EPI	THD	0.8063	0.9793	0.2111	0.1797	0.0885

Table 9: Model Parameters for Selected Stock Pairs (Filtered Data 2018 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWC	0.9042	0.9435	0.5395	0.1178	0.0389
2	EWA	EWY	0.8155	0.9679	0.1717	0.1187	0.0569
3	EWD	EWL	0.8924	0.9116	0.9302	0.1561	0.0554
4	EWD	EWY	0.9784	0.9763	0.2474	0.2137	0.0319
5	EWD	EZU	0.8250	0.9250	0.6969	0.1208	0.0567
6	EWD	FEZ	0.8958	0.9424	0.7000	0.1290	0.0446
7	EWU	PBW	0.9062	0.9486	0.4975	0.2031	0.0662
8	EPI	EEM	0.9429	0.9781	0.4660	0.1604	0.0397
9	EPI	AAXJ	1.0000	0.9754	0.3093	0.1664	0.0000
10	EPI	SOXX	0.8610	0.9775	0.2235	0.1980	0.0800
11	PIN	EEM	0.9429	0.9733	0.3899	0.1308	0.0324
12	PIN	VWO	0.9616	0.9761	0.4326	0.1282	0.0258
13	PIN	AAXJ	1.0000	0.9657	0.2530	0.1378	0.0000
14	PIN	XLI	0.8576	0.9734	0.2349	0.1552	0.0637
15	PIN	SMH	0.8247	0.9657	0.2208	0.1594	0.0741
16	PIN	SOXX	0.8585	0.9702	0.1864	0.1633	0.0668
17	XLF	SOXX	0.8902	0.9798	0.2184	0.1517	0.0536

Table 10: Model Parameters for Selected Stock Pairs (Filtered Data 2018 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWS	0.8792	0.9775	0.7278	0.1178	0.0439
2	EWA	EWT	0.9812	0.9584	0.4761	0.1281	0.0179
3	EWA	THD	1.0000	0.9760	0.1351	0.1448	0.0000
4	EWA	EEM	1.0000	0.9776	0.3153	0.1175	0.0000
5	EWA	VWO	1.0000	0.9785	0.3481	0.1173	0.0000
6	EWA	ILF	0.9357	0.9744	0.3000	0.1358	0.0359
7	EWA	XLI	0.8149	0.9558	0.2023	0.1221	0.0588
8	EWA	XLB	0.9016	0.9150	0.2487	0.1243	0.0420
9	EWQ	EWN	0.9479	0.9659	0.9331	0.0869	0.0205
10	EWH	EWT	0.8741	0.9798	0.5007	0.1247	0.0476
11	EWD	EWL	0.8478	0.9232	0.9238	0.1539	0.0665
12	EWD	EWY	1.0000	0.9771	0.2505	0.2129	0.0000
13	EWD	EWT	1.0000	0.9785	0.7048	0.2108	0.0000
14	EWD	AAXJ	1.0000	0.9795	0.3020	0.1963	0.0000
15	EWD	FEZ	0.9848	0.9681	0.7208	0.1328	0.0166
16	EWU	EEM	0.8690	0.9731	0.4736	0.1627	0.0636
17	EWU	AAXJ	0.8523	0.9579	0.2939	0.1688	0.0710
18	EWU	XBI	0.8421	0.9434	0.0805	0.2133	0.0937
19	EWU	PBW	0.8918	0.9570	0.5853	0.1958	0.0689
20	EWT	XLB	0.8203	0.9700	0.3398	0.1710	0.0807
21	RSX	XLF	0.9131	0.9708	0.4692	0.1986	0.0617
22	PIN	EEM	0.9946	0.9775	0.3852	0.1387	0.0103
23	PIN	AAXJ	1.0000	0.9741	0.2421	0.1429	0.0000
24	PIN	AFK	0.8321	0.9701	0.4838	0.1659	0.0751
25	PIN	FEZ	0.8748	0.9730	0.3714	0.1580	0.0602

Table 11: Model Parameters for Selected Stock Pairs (Filtered Data 2019 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWH	0.9587	0.9689	0.5868	0.1264	0.0264
2	EWA	EWS	0.8646	0.9779	0.7101	0.1120	0.0446
3	EWA	EWT	0.9765	0.9648	0.4704	0.1221	0.0191
4	EWA	THD	0.9999	0.9800	0.1350	0.1400	0.0003
5	EWA	EEM	0.9153	0.9789	0.3079	0.1084	0.0331
6	EWA	ILF	0.8925	0.9752	0.2849	0.1301	0.0454
7	EWA	XLI	0.8334	0.9540	0.1902	0.1189	0.0538
8	EWA	XLB	0.8974	0.9110	0.2354	0.1206	0.0417
9	EWA	SMH	0.8521	0.9507	0.1506	0.1326	0.0559
10	EWA	SOXX	0.8654	0.9515	0.1257	0.1347	0.0538
11	EWQ	EWN	0.9922	0.9729	0.9229	0.0888	0.0079
12	EWD	EWT	0.8019	0.9359	0.7182	0.1838	0.0929
13	EWD	EEM	1.0000	0.9795	0.4817	0.1864	0.0000
14	EWD	FEZ	1.0000	0.9731	0.7440	0.1303	0.0000
15	EWU	EWT	0.8007	0.9640	0.6709	0.1690	0.0851
16	EWU	EPI	0.8210	0.9631	0.5895	0.1841	0.0868
17	EWU	PIN	0.8476	0.9655	0.7169	0.1876	0.0802
18	EWU	EEM	0.8425	0.9660	0.4550	0.1550	0.0676
19	EWU	VWO	0.8532	0.9711	0.5051	0.1546	0.0646
20	EWU	AAXJ	0.8547	0.9552	0.2813	0.1615	0.0673
21	EWU	XBI	0.8134	0.9414	0.0801	0.1999	0.0972
22	EPI	EEM	0.9467	0.9772	0.4502	0.1654	0.0395
23	EPI	VWO	0.9208	0.9788	0.5076	0.1584	0.0467
24	EPI	AAXJ	0.9558	0.9753	0.2834	0.1686	0.0365
25	EPI	FEZ	0.8563	0.9703	0.4548	0.1884	0.0778
26	RSX	XLF	1.0000	0.9784	0.4488	0.1927	0.0000
27	RSX	XLI	0.9997	0.9762	0.1941	0.1854	0.0030
28	RSX	XLB	0.8548	0.9513	0.2574	0.1647	0.0687
29	RSX	SOXX	1.0000	0.9742	0.1327	0.1968	0.0000
30	PIN	EEM	0.9603	0.9738	0.3690	0.1363	0.0279
31	PIN	VWO	0.9388	0.9763	0.4152	0.1311	0.0337
32	PIN	AAXJ	0.9728	0.9712	0.2305	0.1402	0.0236
33	PIN	AFK	0.8316	0.9688	0.5002	0.1615	0.0733
34	PIN	FEZ	0.8946	0.9671	0.3703	0.1583	0.0548
35	XLF	SMH	0.8090	0.9790	0.2186	0.1644	0.0803
36	XLF	SOXX	0.8198	0.9766	0.1876	0.1649	0.0778

Table 12: Model Parameters for Selected Stock Pairs (Filtered Data 2019 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWA	EWZ	0.8166	0.9776	0.1543	0.1239	0.0590
2	EWA	EWT	0.9863	0.9727	0.4344	0.1174	0.0139
3	EWA	SPY	0.8006	0.9440	0.0594	0.0989	0.0501
4	EWA	RSX	0.8740	0.9624	0.4569	0.1181	0.0453
5	EWA	VWO	0.8966	0.9767	0.3285	0.1005	0.0343
6	EWA	XLF	0.9408	0.9656	0.4072	0.1226	0.0310
7	EWA	XLI	0.9960	0.9679	0.1765	0.1195	0.0076
8	EWA	XLB	0.9138	0.9251	0.2226	0.1127	0.0353
9	EWA	SMH	0.9354	0.9600	0.1322	0.1269	0.0337
10	EWA	VTI	0.8191	0.9440	0.1159	0.1000	0.0477
11	EWA	IWV	0.8256	0.9453	0.0993	0.1004	0.0468
12	EWA	SOXX	0.9292	0.9579	0.1097	0.1276	0.0356
13	EWQ	EWN	0.9938	0.9768	0.9019	0.0907	0.0072
14	EWS	EEM	0.9187	0.9798	0.3262	0.0943	0.0282
15	EWD	EWT	0.8009	0.9507	0.7212	0.1792	0.0905
16	EWU	EWT	0.8505	0.9700	0.6293	0.1707	0.0721
17	EWU	PIN	0.8367	0.9797	0.4198	0.1980	0.0879
18	EWU	VWO	0.8868	0.9725	0.4848	0.1569	0.0565
19	EWU	AAXJ	0.9431	0.9579	0.2653	0.1677	0.0416
20	EWU	XLF	0.9621	0.9778	0.6311	0.1856	0.0370
21	EWU	XLI	0.8083	0.9547	0.2586	0.1657	0.0816
22	EWU	SMH	0.8370	0.9487	0.1933	0.1854	0.0829
23	EWU	XBI	0.8249	0.9437	0.0778	0.1941	0.0907
24	EWU	SOXX	0.8331	0.9464	0.1584	0.1873	0.0850
25	EWU	PNQI	0.9172	0.9580	0.4640	0.1901	0.0577
26	PIN	AAXJ	1.0000	0.9788	0.2206	0.2018	0.0000
27	PIN	AFK	0.8648	0.9717	0.4494	0.2106	0.0838

Table 13: Model Parameters for Selected Stock Pairs (Filtered Data 2020 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWI	0.9428	0.9748	0.4669	0.1005	0.0249
2	EWC	EPI	0.9497	0.9786	0.6075	0.2168	0.0502
3	EWC	XLF	0.9888	0.9729	0.6365	0.1772	0.0190
4	EWC	XLI	0.9728	0.9621	0.2628	0.1654	0.0279
5	EWC	XLB	1.0000	0.9775	0.3592	0.1650	0.0000
6	EWC	UGA	0.8551	0.9538	0.2689	0.2219	0.0924
7	EWQ	EWI	0.9081	0.9727	0.8463	0.1332	0.0427
8	EWQ	EPI	0.8499	0.9781	0.7156	0.2162	0.0914
9	EWH	EWI	0.8095	0.9755	0.4353	0.1611	0.0786
10	EWH	EWS	0.9003	0.9786	0.8814	0.1397	0.0467
11	EWH	EPI	0.8636	0.9584	0.4961	0.1658	0.0666
12	EWH	PIN	0.8313	0.9701	0.4473	0.1807	0.0820
13	EWH	ILF	0.8275	0.9781	0.3400	0.1671	0.0767
14	EWD	AAXJ	0.9463	0.9782	0.3552	0.2164	0.0518
15	EWL	SOXX	1.0000	0.9798	0.2029	0.2338	0.0000
16	EWU	EPI	0.9990	0.9759	0.6792	0.2334	0.0073
17	EWU	XLF	0.8498	0.9784	0.6768	0.1829	0.0773
18	RSX	XLF	0.9189	0.9526	0.5169	0.2055	0.0618
19	RSX	XLI	0.9442	0.9632	0.2125	0.2025	0.0497

Table 14: Model Parameters for Selected Stock Pairs (Filtered Data 2020 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWI	0.9791	0.4885	0.1076	0.0000	
2	EWK	AFK	0.9487	0.9784	0.5329	0.1517	0.0355
3	EWQ	EWI	0.8479	0.9725	0.8865	0.1322	0.0564
4	EWG	AAXJ	0.8767	0.9798	0.2973	0.1917	0.0723
5	EWH	EWM	0.8124	0.9799	0.5950	0.1631	0.0788
6	EWH	EPI	0.8359	0.9533	0.4934	0.1706	0.0765
7	EWH	PIN	0.8394	0.9679	0.4563	0.1894	0.0835
8	EWD	AAXJ	0.9324	0.9726	0.3489	0.2321	0.0629
9	RSX	XLF	0.9023	0.9538	0.4981	0.2115	0.0704

Table 15: Model Parameters for Selected Stock Pairs (Filtered Data 2021 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWQ	0.8021	0.9482	0.5375	0.0835	0.0420
2	EWK	EWI	0.9131	0.9597	0.4977	0.1097	0.0342
3	EWK	XLF	0.8040	0.9664	0.3818	0.1248	0.0621
4	EWQ	XLF	0.9688	0.9747	0.6930	0.1990	0.0359
5	EWQ	XLI	0.9474	0.9713	0.2794	0.1867	0.0443
6	EWH	EZU	0.9319	0.9771	0.3904	0.1778	0.0483
7	EWH	EPI	0.9415	0.9589	0.4704	0.1938	0.0488
8	EWH	PIN	0.8790	0.9696	0.4576	0.2035	0.0761
9	EWH	FEZ	0.9363	0.9800	0.4010	0.1791	0.0470
10	EWH	XLF	1.0000	0.9742	0.3811	0.2035	0.0000
11	EWH	XLI	1.0000	0.9799	0.1562	0.1970	0.0000
12	ICLN	PBW	0.8853	0.9775	0.1937	0.1429	0.0517

Table 16: Model Parameters for Selected Stock Pairs (Filtered Data 2021 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWI	0.9136	0.9606	0.4967	0.1130	0.0351
2	EWK	XLF	0.8313	0.9751	0.3672	0.1320	0.0599
3	EWQ	XLF	1.0000	0.9740	0.6883	0.2161	0.0000
4	EWQ	XLI	0.9231	0.9654	0.2775	0.1998	0.0582
5	EWH	ILF	0.8460	0.9783	0.3657	0.1920	0.0824
6	EWP	ILF	0.8258	0.9617	0.5363	0.2167	0.1005
7	EWL	SPY	0.8248	0.9468	0.0856	0.2029	0.0948
8	EWL	IVV	0.8307	0.9497	0.0842	0.2028	0.0928

Table 17: Model Parameters for Selected Stock Pairs (Filtered Data 2022 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWQ	0.9187	0.9468	0.5014	0.1003	0.0302
2	EWK	EWI	0.9638	0.9521	0.5103	0.1193	0.0234
3	EWK	EPI	0.8831	0.9795	0.3880	0.1644	0.0601
4	EWK	XLI	0.8153	0.9434	0.1490	0.1403	0.0677
5	EWC	XLF	0.8519	0.9573	0.6531	0.2045	0.0862
6	EWQ	XLF	0.9623	0.9737	0.7127	0.2372	0.0473

Table 18: Model Parameters for Selected Stock Pairs (Filtered Data 2022 H2)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWK	EWI	0.9260	0.9557	0.5205	0.1169	0.0334
2	EWQ	EWI	0.8459	0.9167	1.0173	0.1459	0.0636

Table 19: Model Parameters for Selected Stock Pairs (Filtered Data 2023 H1)

Pair	Stock A	Stock B	R^2_{MR}	ρ	Beta	σ_k	σ_z
1	EWC	XLF	0.9178	0.9724	0.6904	0.2296	0.0692
2	EWQ	EWI	0.8175	0.9149	1.0201	0.1463	0.0706
3	EWH	EWD	0.8482	0.9731	0.2772	0.2036	0.0867
4	EWH	EWY	0.8802	0.9695	0.1655	0.1908	0.0709
5	EWH	EEM	0.8510	0.9750	0.3680	0.1550	0.0653
6	EWH	VWO	0.8241	0.9756	0.3951	0.1527	0.0710
7	EWI	FEZ	1.0000	0.9791	0.6851	0.1400	0.0000

Table 20: Model Parameters for Selected Stock Pairs (Filtered Data 2023 H2)