PAIR TRADING USING REINFORCEMENT LEARNING

A Project Report Submitted

for the Course

MA498 Project I

by

Utsav Bhardwaj, Anmol Abhay Jain

(Roll No. 190123062, 190123006)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

November 2022

CERTIFICATE

This is to certify that the work contained in this project report entitled "Pair Trading using Reinforcement Learning" submitted by Utsav Bhardwaj (Roll No.: 190123062) and Anmol Abhay Jain (Roll No.: 190123006) to the Department of Mathematics, Indian Institute of Technology Guwahati towards partial requirement of Bachelor of Technology in Mathematics and Computing has been carried out by them under my supervision.

It is also certified that, along with literature survey, a few new results are established/computational implementations have been carried out/simulation studies have been carried out/empirical analysis has been done by the student under the project.

Turnitin Similarity: 17%

Guwahati - 781 039	Prof. N. Selvaraju
November 2022	Project Supervisor

ABSTRACT

The main aim of the project is to create a statistical arbitrage strategy based on pair trading. Our project entails the entire life cycle from pair selection from a wide universe of securities to training reinforcement learning models to effectively trade on the spread generated from the pairs selected. In our project, we explore various properties of price time series of securities and use these properties to create profitable trading opportunities. Of these, the property of cointegration of the time series of the selected pair is an important step in selecting the pairs for trading as we can create a stationary spread using a weighted difference of the securities. We explore various ways of testing these cointegration properties and then finding ways to generate the best possible stationary spread. In the end, we see what are some ways we can now trade on this stationary spread and how we can model this problem as a reinforcement learning model for our future work.

Contents

Chapter 1

Pair Trading

This chapter mainly discusses the core idea of statistical arbitrage and pair trading as a trading strategy, as well some statistical foundations behind them.

1.1 Statistical Arbitrage and Pair Trading

Statistical arbitrage is a collection of quantitatively driven algorithmic strategies. These strategies are generally used to realize profits in short intervals based on the historical statistical relationship between thousands of financial instruments. They generally take historical prices and stock return time series and other financial indicators as inputs and output different trading signals like buy, hold, sell and so on.

Pair trading is a very famous subcategory of statistical arbitrage strategies. While it is hard to predict individual moments of stocks, it may be easier to predict the relative behaviors of the stocks. In a basic version of this strategy we try to find a pair of financial instruments who 'historically move closely' and if we find short term inconsistency in this trend that is the spread between the two instruments widen, short the increasing instrument and long the decreasing one and if the historical trends repeats then we make a profit.

Before we describe the above process more rigorously and mathematically, we need to understand the inputs that we will be working with, that is the time series data of stock prices.

1.2 Inputs and data characterization

A time series is a sequence of real values taken at successive equally spaced points in time.

$$X_t = (X_1, \ldots, X_n)$$

For most of our models, the only input is the prices of various instruments at various frequencies(1 min, 1 hour, 1 day), with different models utilizing different frequencies of data.

We can think of the observed time series as a discrete stochastic process. Now for a finite set of times, the joint distribution function of $X_t = \{X_i(\omega); i \in T\}$ is defined by

$$F_{t_1,\ldots,t_n}(x_{t_1}, \ldots, x_{t_n}) = P\left(X_{t_1}(\omega) \leq x_{t_1}, \ldots X_{t_n}(\omega) \leq x_{t_n}\right)$$

Stationarity is an important assumption in many statistical procedures and mathematical models that we will be working with so let's have a look at it.

1.2.1 Stationarity

A stationary time series/process has the property that the statistical properties of a process generating the time series does not change over time. Intuitively it means a flat looking series, without trend, constant variance over time, a constant autocorrelation structure over time and no periodic fluctuations. There are two kinds of stationarity: Weak and Strong.

Strict/Strong Stationarity

A strongly stationary stochastic process is one whose unconditional joint probability distribution does not change when shifted in time. Formally, the discrete stochastic process $X_t = \{X_i; i \in Z\}$ is stationary if

$$F_X\left(x_{t_{1+\tau}}, \ldots, x_{t_{n+\tau}}\right) = F_X\left(x_{t_1}, \ldots, x_{t_n}\right)$$

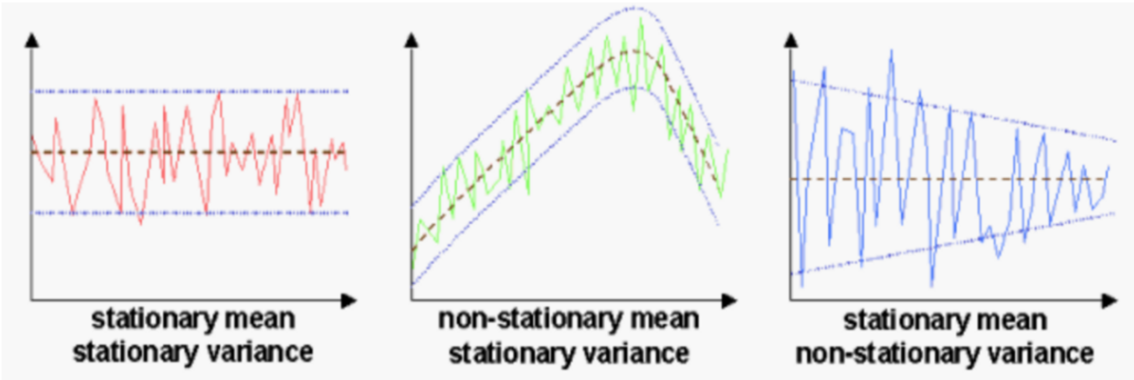
for $T \subset \mathbb{Z}$ with $n \in \mathbb{N}$ and any $\tau \in \mathbb{Z}$.

Weak Stationarity

Weak stationarity only requires the time invariance of the first moment and the cross moment which is the autocorrelation of time series.

This means time series should have the following properties-

1. The first moment of X_i is constant; i.e. $\forall i \in T, E[X_i] = \mu$.
2. The second moment that is the variance of X_t is finite for all t.
3. The auto-covariance depends only on the difference $u - v$; i.e. $\forall u, v, a, cov(X_u, X_v) = cov(X_u + a, X_v + a)$



For most of the applications moving unless stated explicitly; moving forward by stationarity we mean weak stationarity.

1.2.2 Integration Order of Time Series

In statistics, the order of integration, denoted bt I(d), of a time series is a summary statistic, which reports the minimum number of differences required to obtain a weakly stationary series.
A time series is integrated of order d if if taking repeated differences d times yields a stationary process. Stationary process is said to be order 0 integrated.

1.2.3 Mean Reversion in Time Series

A time series is said to be mean reverting if it tends to fall when its level is above its long-run mean and rise when its level is below its long-run mean. A weakly stationary time series is mean reverting.

1.2.4 Log Transformation of Price Series

Most of the time when working with financial price series data rather than working directly with the price we work with log prices that we take the logarithm of time series.

$$\ln(X_t) = (\ln(X_1), .. \ln(X_n))$$

There are a few reasons behind this transformation-

1. Log returns are more convenient to work with as they are time additive that is if we want to calculate the log return of time t_1 to t_3 such that we know the log returns for t_1 to t_2 and t_2 to t_3 then we can just simply add the returns to get the result

2. Log returns have been seen to be normally distributed and this helps because we know a lot about properties of normal distribution.

So moving forward when we will mostly be working with log price series of financial instuments and thus unless specified otherwise when are taking about price and return series we are actually talking about log price and log return series.

1.3 Quantifying if two price series move together

The first thought that might come to our mind is to see if the log return series of the two instruments highly correlated or not. But as we will see cointegration of two time series is much better way to see if in the long term the two price series moving together.

Now lets understand by what it means for two price series to be cointegrated [1].

1.3.1 What is Cointegration?

Now before we answer this question; one basic assumption we will be making is that the log price series is integrated with order 1 that is Y_t is a $I(1)$ series and therefore by extension log returns $(Y_t + 1 - Y_t)$ will be integrated with order 0. Now the fact that returns are stationary is great because we found a time-invariant property, which is the mean of the time series around which it will be reverting thus as create profitable trading strategies. But there is a problem that is we can not trade on returns rather we trade on prices that are integrated with order 1.

Definition 1.3.1. X_t and Y_t are said to be cointegrated with order 1, if X_t and Y_t are $I(1)$ series and there exist a beta such that $Z_t = X_t - \beta * Y_t$ is a $I(0)$ series.

Thus if we can find X_t and Y_t such that the spread $Z_t = X_t - \beta * Y_t$ is stationary then we can successful trade on the spread. We can consider spread a portfolio of two stocks X and Y with weights 1 and $-\beta$ respectively. Now this β is known as The cointegration coefficient.

1.4 Main Steps in a Pair Trading Strategy

Using the concepts developed above now we can describe the broad steps we will taking to execute a pair trading strategy.

1. **Pairs Selection** - We have select pairs for the whole universe of instruments available to us. Since the universe could be large we have to smartly identify stock pairs that could be potentially integrated.

2. **Cointegration Tests** - Once we have identified potential stock pairs we have to check if they are actually really cointegrated or not. We will statistical cointegration tests to check for this.

3. **Spread Generation** - Once we have seen confirmed that the pairs have been historically cointegrated; we need to find the cointegration coefficient for the pairs to generate the spread. Now many a times finding the cointegration coefficient is part of the cointegration test if self but as will see later we can be smart about picking the cointegration coefficient for better spread generation

4. **Optimally trading on the spread** - Once the spread has been generated we have to basically use the mean reverting property of spread to generate profit. If we think of the spread itself as financial instrument which we can trade then our basic trading strategy would be to set two threshold around the mean of the spread denoting when the spead has become overvalued and undervalued. If the spread crosses any of the two threshold then we know that we can take the opposite position because it will revert back to the mean. Thus we will short it when it crosses the overvalued threshold and close the trade when it reaches back to its main and vice versa

Chapter 2

Pair Selection Strategies

In this chapter we'll look into methods and strategies for selecting pairs of securities to generate a spread on. Various simple pair selection strategies such as those based on distance based metrics and correlation exist, we'll be mostly looking into statistical co-integration tests and more complex machine learning based methods to generalise the co-integration approach to larger security universes efficiently.

2.1 Calculating Cointegration

Calculating the amount of cointegration between any two economic variables involves devising methods to analyze the existence of a long-term relationship. As such, there has been much research in the area of calculating cointegrating relationships between various classes and types of variables. We primarily relied on the Engle-Granger test for testing pairs for selection.

Remark 2.1.1. Cointegration as discussed in previous chapters to a statistical long-term relationship between two time series.

2.2 Engle-Granger Test

The idea for cointegration as a statistical property between two non-stationary time-series variables, representing their long-standing relationship was first introduced in the seminal paper by Robert Engle and Clive Granger in 1987 [5]. The paper also introduced the idea of using static linear regression residuals to allow for confirming the stationarity of the spread between the two time series. Along with the test for normality, more statistical tests like the Augmented Dickey-Fuller test also need to be conducted to test for the existence of any unit roots in the residuals i.e. to deal with any scenarios of spurious regressions.

2.2.1 Introduction

Most economic variables tend to be non-stationary $I(1)$ variables, and the existence of a cointegrating relationship requires some existence of a combination of these variables to attain stationarity. Otherwise, any deviation of their spread from equilibrium will not be temporary. Consider the condition for the spread of two stocks to be in equilibrium -

$$Y_t = \beta_1 * X_t + \beta_0 + \epsilon$$

Y_t - Price of stock 1 at time t
X_t - Price of stock 2 at time t

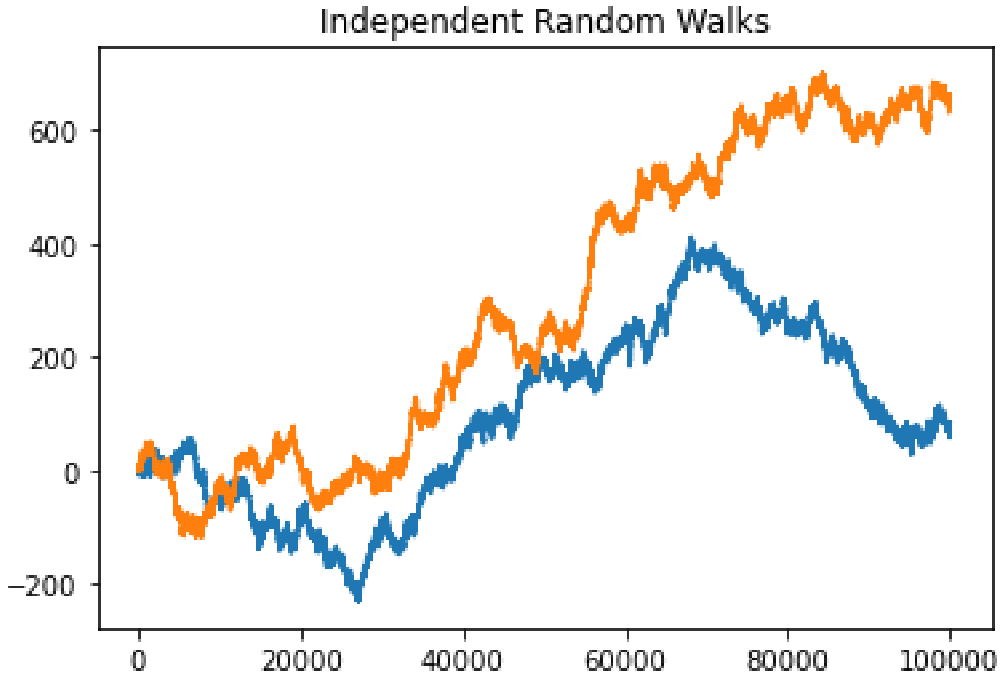
To ensure that this equilibrium is stable in the long-term, we require that the deviation ϵ be stationary i.e., $Y_t - \beta_1 * X_t$ is $I(0)$ (Note that the addition of a constant will not have any effect on the presence or existence of a cointegrating relationship.)

The general idea of the Engle-Granger approach follows from this notion, where we test the residuals obtained from the static regression between two time series variables for stationarity. However, conventional testing procedures can lead to erroneous results due to the spurious regression problem, which we discuss below.

2.2.2 The spurious regression problem

Given two unrelated time series, they might appear linearly related when regressed against each other i.e., their coefficient estimate does not converge towards zero (which is the true value for unrelated time series) but instead follows a non-degenerate distribution with usually very significant t-values. This leads to high regression R^2 values, which might mislead one to this indicating a non-existent relationship.
For example, let us look at the regression results for between two independently generated random walks -

```
# generate two independent random walks
n = 100000
B1 = [0]
B2 = [0]
for i in range(n):
    B1.append(B1[-1] + np.random.standard_normal())
    B2.append(B2[-1] + np.random.standard_normal())
```




```
Gradient and intercept 1.3450148402774478 173.24602131057634
R-squared 0.6205101000091686
```

As we can see in the above figures, the regression results between two independent random walks do not converge the coefficient to the true value of 0, but instead a non-zero erroneous result with very high R^2 values.

Spurious regressions occur when dealing with non-stationary time series because of the existence of stochastic trends, which cause the series to appear to be trending (locally). When two series seem to share similar local trends, this leads to them appearing to be related.

Theoretically, the residual of a spurious regression has a unit root, and thus when determining a relationship between two non-stationary time series, we need to take into account the possibility of a spurious regression and check if the residual is stationary (i.e. cannot reject the null hypothesis for the unit root test).

2.2.3 Cointegration test

The Engle-Granger cointegration test essentially applies the unit root test to check for stationarity to the residual of cointegration regression. Suppose the two time series have a cointegrating relationship. In that case, they will not have a unit root, and in case of a spurious regression with two unrelated time series, the residual must have a unit root. The null hypothesis for the test is that the two series are not cointegrated.

We implement the cointegration test using Augment Dickey Fuller [4] methods provided through the statsmodels library.

2.3 Pair Selection using Machine Learning

Pair trading methods employ a wide variety of pair selection strategies, ranging from simple, some distant metric minimisation for constructing pairs, to more statistical approaches based on correlation or cointegration, as discussed before. Correlation reflects the short-term relation between two time series, whereas cointegration is a measure of a more long-standing relationship between the prices. As discussed apriori, the correlation approach runs into a variety of issues, with highly correlated stocks having no guarantees on the spread being mean reverting. This potential lack of equilibrium also leads to higher divergence risks. Thus, proper cointegration testing methods such as the Engle-Granger tests are necessary to properly check for the validity of any long lasting pattern.

2.3.1 Problems with Multiple Hypothesis testing

With the increase in popularity of pair trading strategies, it becomes harder and harder to find highly rewarding pairs. The naive method of testing all the possible pairs of stocks for mean reversion runs into the following problems -

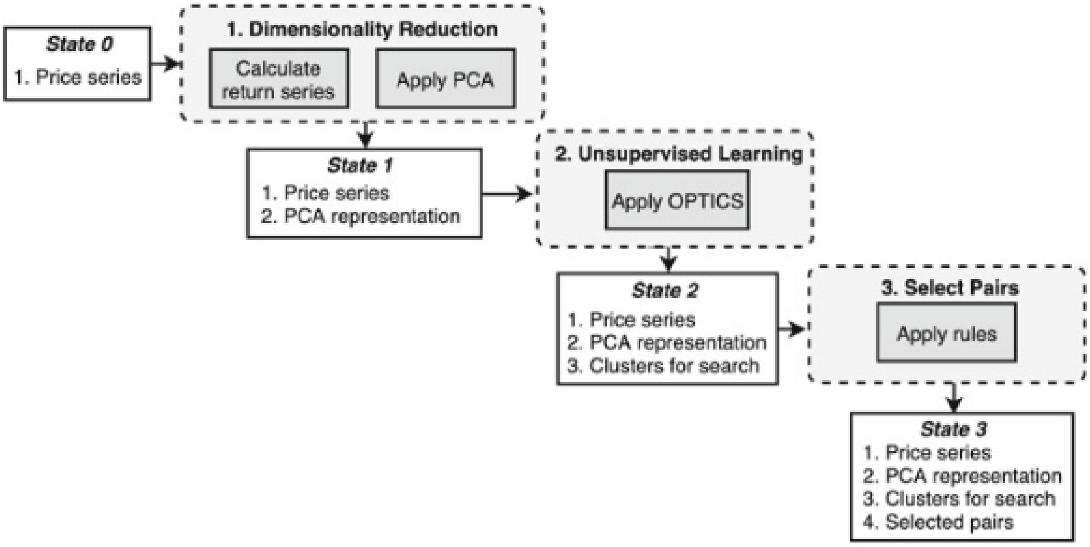
First, for a large universe of stocks testing every possible pair for mean-reversion becomes extremely computationally intensive.
Secondly, the more hypothesis testing we perform at one for every pair, the higher the number of false positives we run into after selection, depending on the chosen levels for the test.

Most strategies solve these issues by clustering or partitioning the universe of stocks and securities into various sub-groups based on manual selection, economic sector-based classification, etc. This helps in reducing the number of pairs required for further mean reversion testing by a lot, but we risk potentially losing some cointegrated pairs. Thus an ideal clustering strategy would lie somewhere in the middle of only looking for apparent pairs and searching the entire universe of pair possibilities.

2.3.2 Machine Learning Approach

The paper by Sarmento and Horta 2020 [6] delved into solving the above dilemma and proposed a three-pronged process built on unsupervised machine learning algorithms to infer meaningful clusters of assets from which to select pairs. The paper's premise is to extract the structure exhibited by the data instead of imposing fundamental groups to which securities should belong. The entire process follows these steps -

1. Dimensionality reduction - find a compact representation for each security
2. Unsupervised learning - use an appropriate clustering algorithm
3. Select pairs - define further rules for the selection of pairs within these clusters

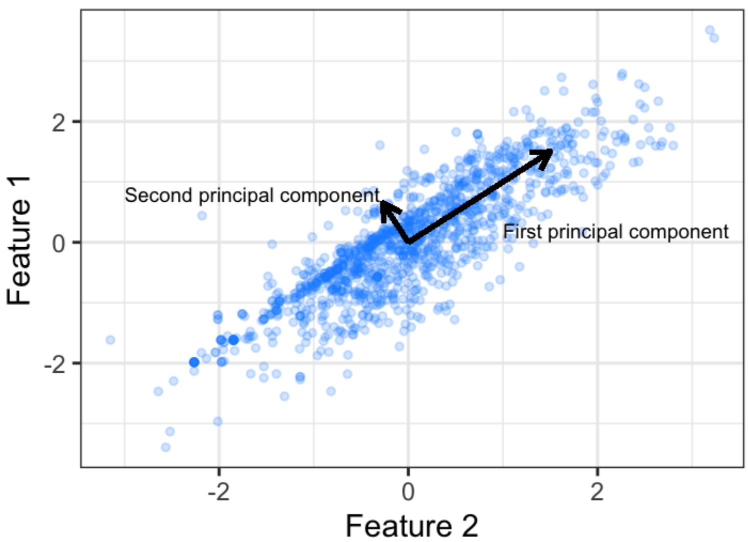


2.3.3 Dimensionality Reduction and PCA

Dimensionality reduction is the transformation of high-dimensional data into a reduced-dimension representation which maximizes the "information" presented in the original data. This helps eliminate the curse of dimensionality and take care of other undesired properties in high dimensional data, such as linearly related features. Fundamentally, any dimensionality reduction algorithm aims to approximate the true intrinsic dimensionality of the data. We utilize Principal Component Analysis as our algorithm for dimensionality reduction.

PCA is a statistical procedure that uses an orthogonal transformation to convert a large set of possibly correlated features into a new linearly independent orthogonal set of features i.e., the principal components. Each successive principal component accounts for the maximum variance w.r.t. the set, with the condition that it is orthogonal to the previous components.

Mathematically, the problem of obtaining the principal components reduces to obtaining the eigenvectors and eigenvalues for the covariance matrix of the features, with the eigenvectors in descending order of eigenvalues being every principal component respectively. Most modern, efficient implementations of PCA utilize single value decomposition (SVD) algorithms.



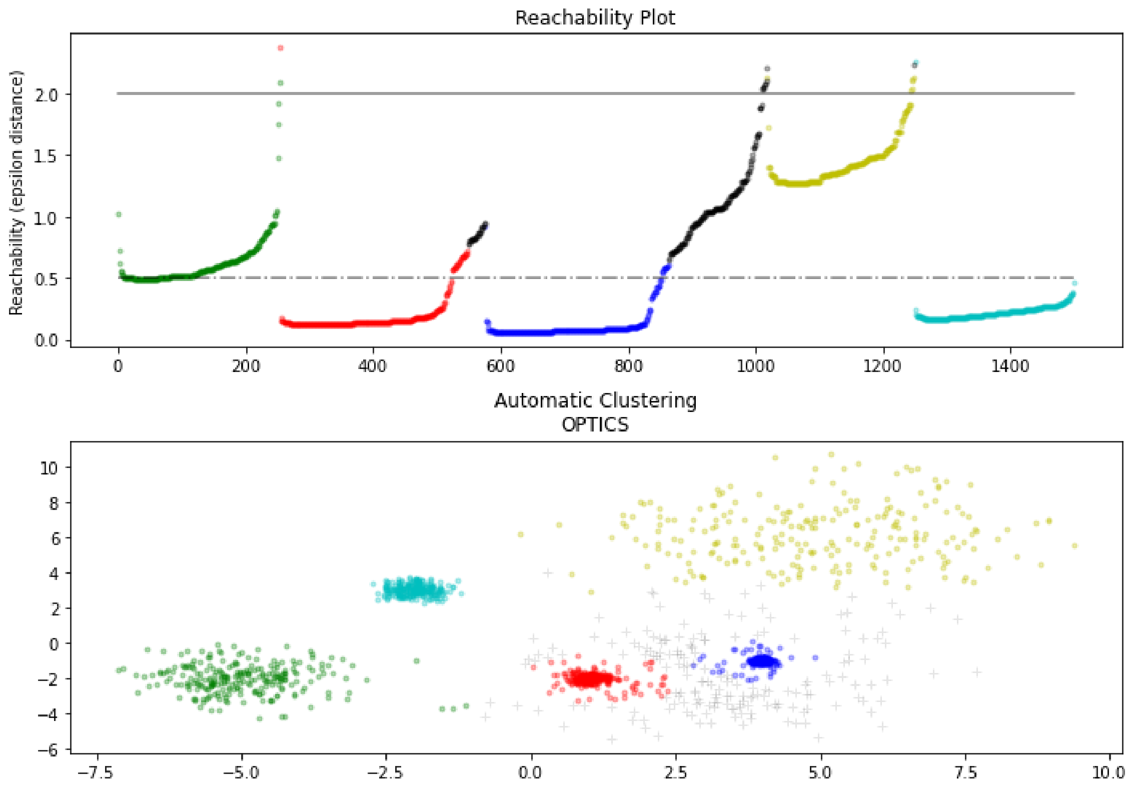
2.3.4 Unsupervised learning - Clustering

There exist a wide variety of clustering algorithms; the primary candidates considered here are two popular density-based clustering algorithms: DB-SCAN and OPTICS [2]

DBSCAN is a density-based clustering algorithm that assigns points to a cluster based on how closely they are packed together.

OPTICS is an improvement and extension of the above DBSCAN algorithm, which allows for using the algorithm for meaningful clustering of data with varying densities. The algorithm utilizes an ordering method for the points, such that spatially close points are neighbours. Also, every point stores a certain distance representing the acceptance density for two points to belong to the same cluster.

OPTICS, while requiring storage of more data during the processing of the algorithm and overall a higher time and computational complexity, allows for less hyperparameter tuning requirements and does not directly segregate the given data into clusters. Instead, it just produces a distance plot and leaves the interpretation of the clusters to the user.



2.3.5 Absolute rules for pair selection

After obtaining clusters of related securities and their dimensionally reduced representations from the previous steps, we use several successive to look for various desired properties in our pair to finally select for trading. The rules that we use for selection are -

1. Co-integration test between the two stocks using the previously discussed Engle-Granger Test.

2. Test further mean-reversion in the spread obtained through the previous cointegration tests be above a given threshold.

3. The variance of the spread should be above a certain threshold to generate significant profits keeping in mind transaction costs, which are

even more critical when considering high-frequency data.

Any pair passing all the above tests successively is chosen as a pair to execute pair trading securities on.

Chapter 3

Spread generation and Optimal Trading

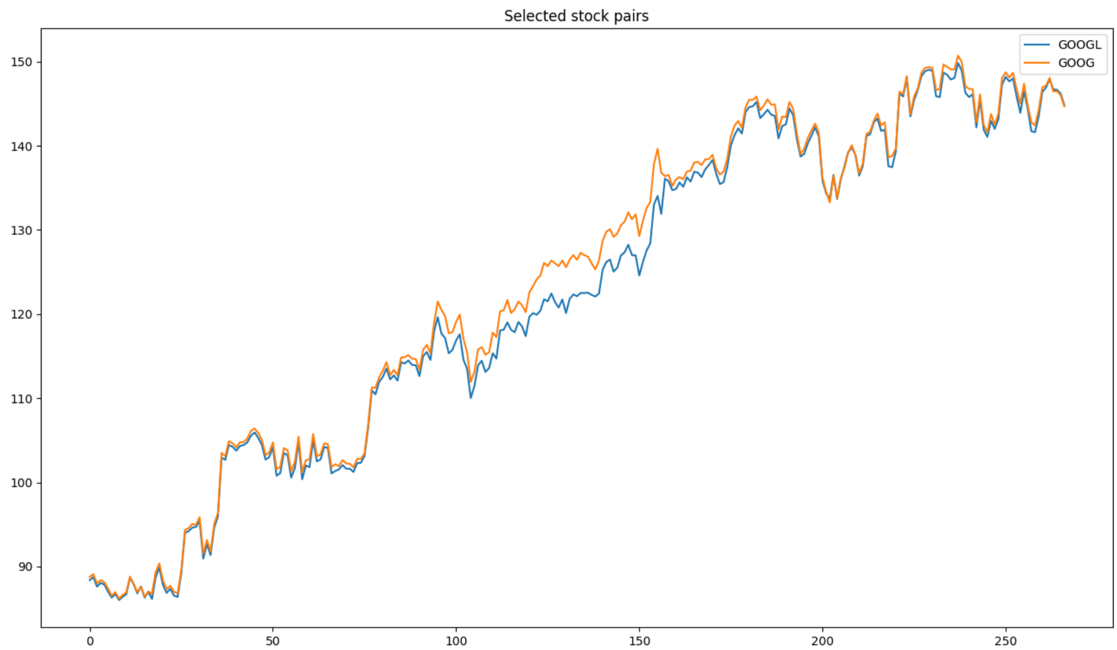
In the previous chapters we have discussed various methods for choosing the pairs on which to execute trades on, based on statistical relationships such as cointegration and machine learning based clustering methods. Now we look into formulating an optimal strategy to generate the spread and trade on this spread using manual and finally reinforcement learning based models.

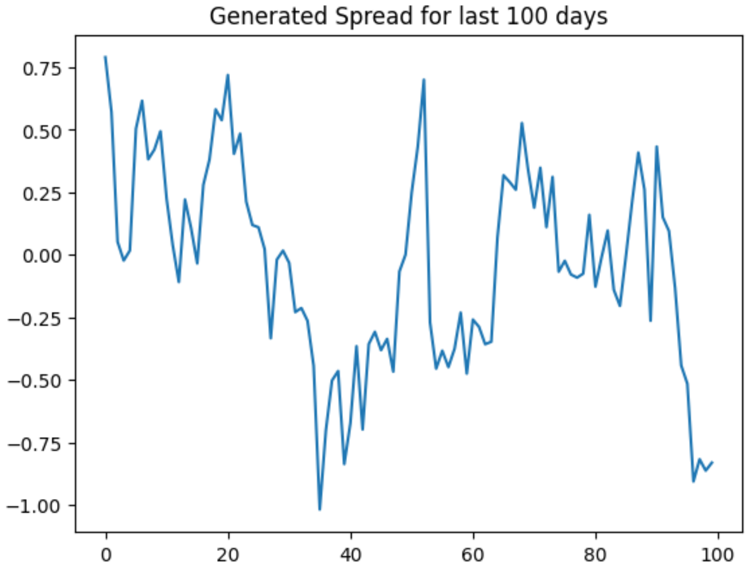
3.1 Spread generation

Generating the spread for any two pairs of securities requires estimating the regression coefficient β let the two price series be Y_t and X_t the the spread can be seen as

$$Z_t = Y_t - \beta * X_t = \mu + \epsilon_t$$

Now if calculate the spread like this then we are assuming that the parameter β of regression remains constant with time. When we study real-life price time series we find that the regression coefficient for a pair can change with time. Hence a way to combat this is through a rolling regression-based method, dividing the data set between windows and using the β's from past windows to generate successive spreads. And hence for each window, we get a different values of β and μ which are more relevant and hence generate a better stationary spread.
The spread generated for the pairs "GOOGL" and "GOOG" looks like follows -

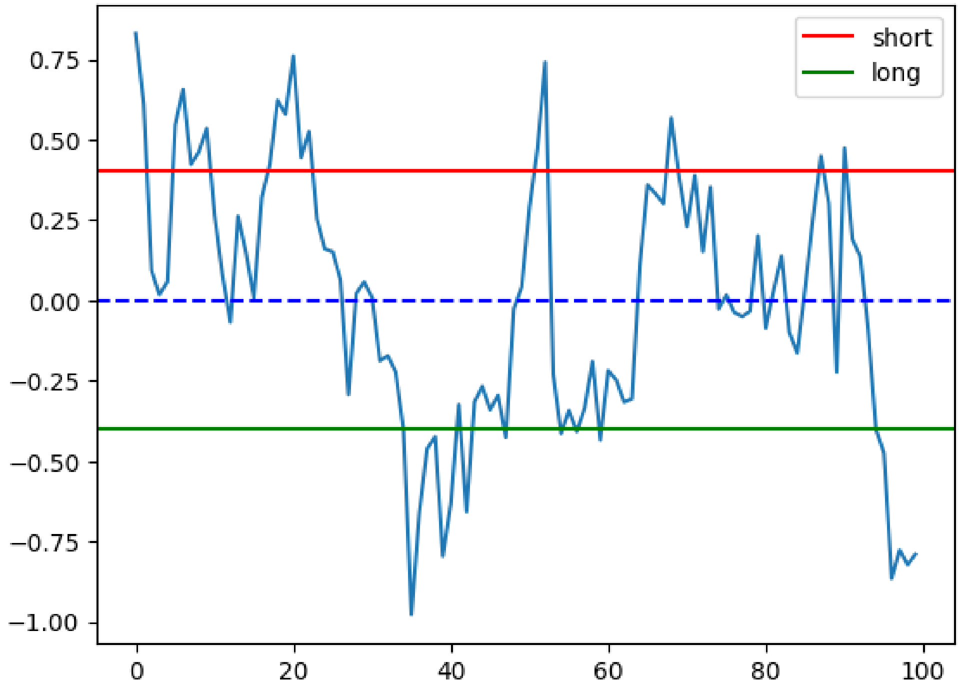





```
ADF Statistic: -3.315148
p-value: 0.014215
Critical Values:
        1%: -3.498
        5%: -2.891
        10%: -2.583
Reject Ho - Time Series is Stationary
```

3.2 Manual Trading Levels

Manually trading on the spread involves various levels for shorting and longing the stocks based on various statistical indicators such as deviation from the mean, and other fundamental methods. A simple manual trading setup on the generated spread from the previous section will look like follows -



3.3 Future Scope - Reinforcement Learning for Optimal Trading

The previous method of deciding manual trading levels can get cumbersome very quickly as the number of pairs increase, with separate tuning required for every individual pair.

Alternatively we can use machine learning algorithms to infer the structure and patter of mean reversion for a spread, helping generalise between the trading strategies for different pairs. Various algorithms ranging from linear regression(and variations such as Lasso etc.), random forest classifiers, gradient boosted trees, neural networks and others can be used for the task.

Here we plan to look into the subclass of Reinforcement Learning algorithms to create optimal trading strategies [3].

Bibliography

[1] Carol Alexander, Ian Giblin, and Wayne Weddington. Cointegration and asset allocation: a new active hedge fund strategy. 2002.

[2] Mihael Ankerst, Markus Breunig, Hans-Peter Kriegel, and Joerg Sander. Optics: Ordering points to identify the clustering structure. volume 28, pages 49–60, 06 1999.

[3] Andrew Brim. Deep reinforcement learning pairs trading with a double deep q-network. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0222–0227, 2020.

[4] David A. Dickey and Wayne A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, 1979.

[5] Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987.

[6] Simão Moraes Sarmento and Nuno Horta. Enhancing a pairs trading strategy with the application of machine learning. *Expert Syst. Appl.*, 158:113490, 2020.