# **Pair Trading Project User Manual**

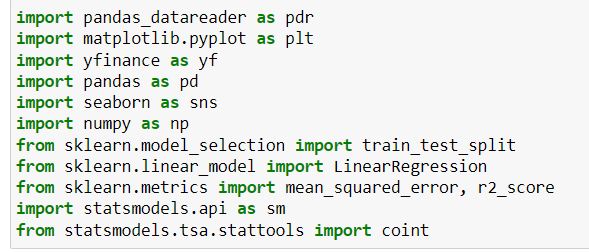
## **Introduction**

This user manual provides a comprehensive guide on implementing a pair trading strategy using historical stock price data. Pair trading involves identifying two stocks with a high correlation and trading them to exploit deviations from their historical price relationship. This project utilizes various statistical and machine learning techniques to identify and trade on these pairs, and includes a backtesting framework to evaluate the strategy's performance.

## **Setup Instructions**

### **Prerequisites**

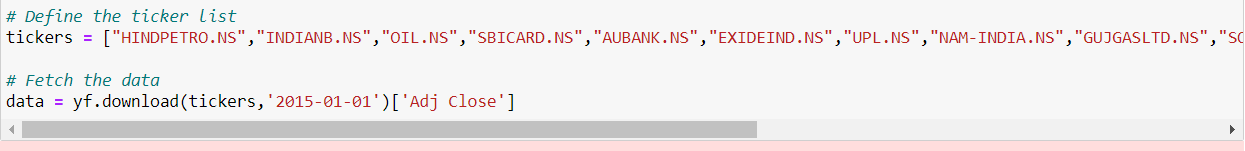
Before running the project, ensure you have Python installed along with the necessary libraries. The required libraries include NumPy, Pandas, SciPy, Statsmodels, Matplotlib, Seaborn, yFinance, Scikit-learn, and Pandas DataReader. These libraries are used for data manipulation, statistical analysis, plotting, and fetching historical stock prices.



## **Data Loading and Preparation**

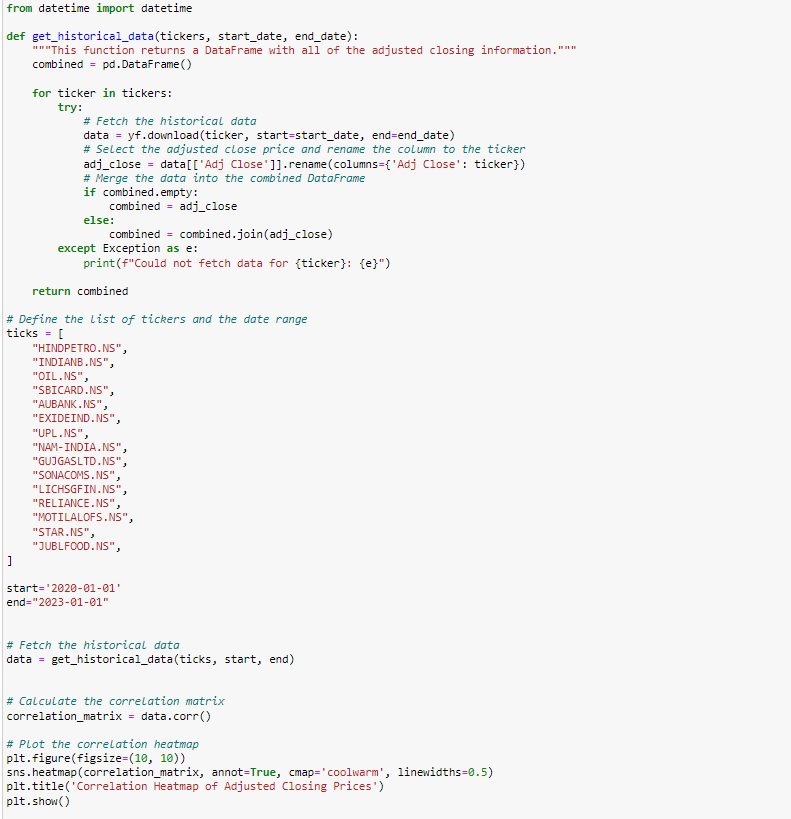
### **Fetching Historical Data**

1. **Define the Ticker List:** A list of stock tickers is defined. These tickers represent the stocks you are interested in for pair trading.
2. **Fetch Data:** Historical stock price data (adjusted closing prices) is fetched from Yahoo Finance using the yfinance library. The data is loaded into a Pandas DataFrame.
3. **Check Data Shape and Types:** The shape and data types of the fetched DataFrame are checked to ensure the data is loaded correctly.



### **Correlation Analysis**

1. **Plot Correlation Matrix:** A correlation matrix is plotted to visualize the correlation between the stock prices. This helps in identifying pairs of stocks with a high correlation, which are potential candidates for pair trading.



## **Pair Trading Strategy**

### **Cointegration Test**

1. **Cointegration Test Function:** A function is defined to fetch historical data for the given list of tickers within a specified date range. This function ensures data is correctly merged into a single DataFrame.
2. **Find Cointegrated Pairs:** Another function performs the cointegration test on each pair of stocks. Cointegration tests whether a pair of stocks move together in the long run, which is crucial for pair trading.
3. **Identify Pairs:** The function identifies pairs of stocks with a p-value less than 0.05 from the cointegration test, indicating they are suitable for pair trading.

### **Visualizing Cointegration Results**

1. **Plot Cointegration Test Results:** The results of the cointegration test are visualized using a heatmap, showing the p-values for each pair of stocks. This helps in easily identifying the best pairs for trading.

### **Implementing the Pair Trading Strategy**

1. **Monitor Spread:** Once the cointegrated pairs are identified, the spread (the price difference between the two stocks) is monitored. Trades are executed when the spread deviates from its historical mean, betting on the spread reverting to the mean.

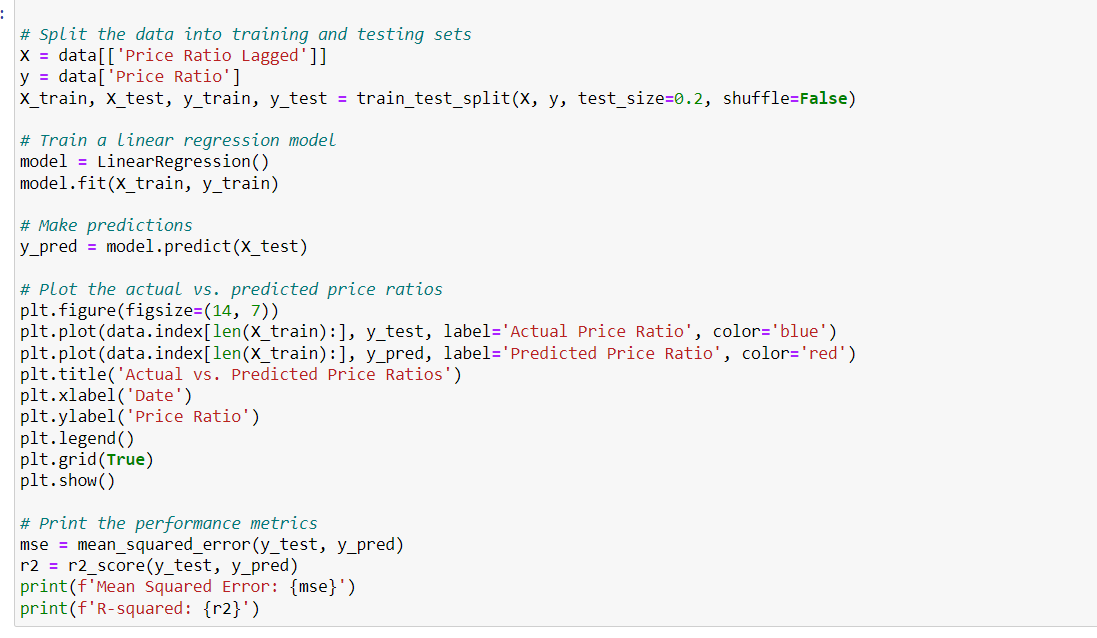
## **Linear Regression Model**

### **Introduction to Linear Regression**

Linear regression is used to model the relationship between two stocks in a pair. It helps in understanding how one stock moves in relation to the other, which is crucial for predicting the spread and making trading decisions.

### **Implementing Linear Regression**

1. **Train-Test Split:** The data is split into training and testing sets. The training set is used to fit the model, and the testing set is used to evaluate its performance.
2. **Fit Linear Regression Model:** The LinearRegression model from Scikit-learn is used to fit the relationship between the two stocks. The model learns the coefficients that best describe the relationship.
3. **Evaluate Model Performance:** The model's performance is evaluated using metrics like Mean Squared Error (MSE) and R-squared (R²). These metrics help in understanding how well the model predicts the relationship between the two stocks.



### **Interpreting Regression Results**

1. **Coefficients:** The coefficients of the regression model indicate the expected change in the dependent variable (one stock's price) for a unit change in the independent variable (the other stock's price).
2. **Model Fit:** The R² value indicates how well the model fits the data. A higher R² value means a better fit.

## **Backtesting the Pair Trading Strategy**

### **Introduction to Backtesting**

Backtesting is the process of testing a trading strategy using historical data to see how it would have performed in the past. It helps in evaluating the effectiveness and robustness of the strategy.

### **Implementing Backtesting**

1. **Define the Trading Strategy:** Establish the rules for entering and exiting trades based on the spread between the pairs. Typically, you enter a trade when the spread deviates significantly from the mean and exit when it reverts to the mean.
2. **Simulate Trades:** Use historical data to simulate trades based on the defined strategy. Track the entry and exit points, as well as the profit or loss from each trade.
3. **Calculate Performance Metrics:** Evaluate the strategy's performance using metrics such as total return, annualized return, Sharpe ratio, maximum drawdown, and win rate. These metrics provide insights into the profitability and risk of the strategy.

### **Analyzing Backtesting Results**

1. **Review Trade Log:** Analyze the trade log to understand the trades made during the backtesting period, including the entry and exit points, and the profit or loss from each trade.
2. **Performance Metrics:** Interpret the performance metrics to evaluate the strategy's effectiveness. A higher Sharpe ratio indicates better risk-adjusted returns, while a lower maximum drawdown indicates lower risk.
3. **Visualize Results:** Plot the cumulative returns over time to visualize the performance of the strategy. This helps in understanding the growth of the investment and the periods of drawdown.