**Strategies For Successful Futures Trading: Leveraging Crude Oil Price Forecasts**

This repository contains Python code for analyzing financial time series data and evaluating two trading strategies. The code uses historical futures contract data, performs data preprocessing, statistical analysis, and implements simulation-based trading strategies.

**Dataset**

The code requires the following dataset, which should be placed in the same directory as the Python script:

\* `Future\_CRUDE\_2005\_2023.csv`: Contains historical data for crude oil futures contracts. Ensure the CSV file has columns named 'Date', 'Expiry Date', and 'Close'.

**Dependencies**

The following Python libraries are required:

\* pandas

\* datetime

\* numpy

\* scipy

\* matplotlib

\* typing

\* seaborn

\* statsmodels

\* arch

You can install these using pip:

```bash

pip install pandas numpy scipy matplotlib seaborn statsmodels arch

**Code Structure**

The Python script is divided into the following sections:

### **1. Data Loading**

This section loads the necessary data from CSV files using pandas. It also defines a helper function for parsing dates.

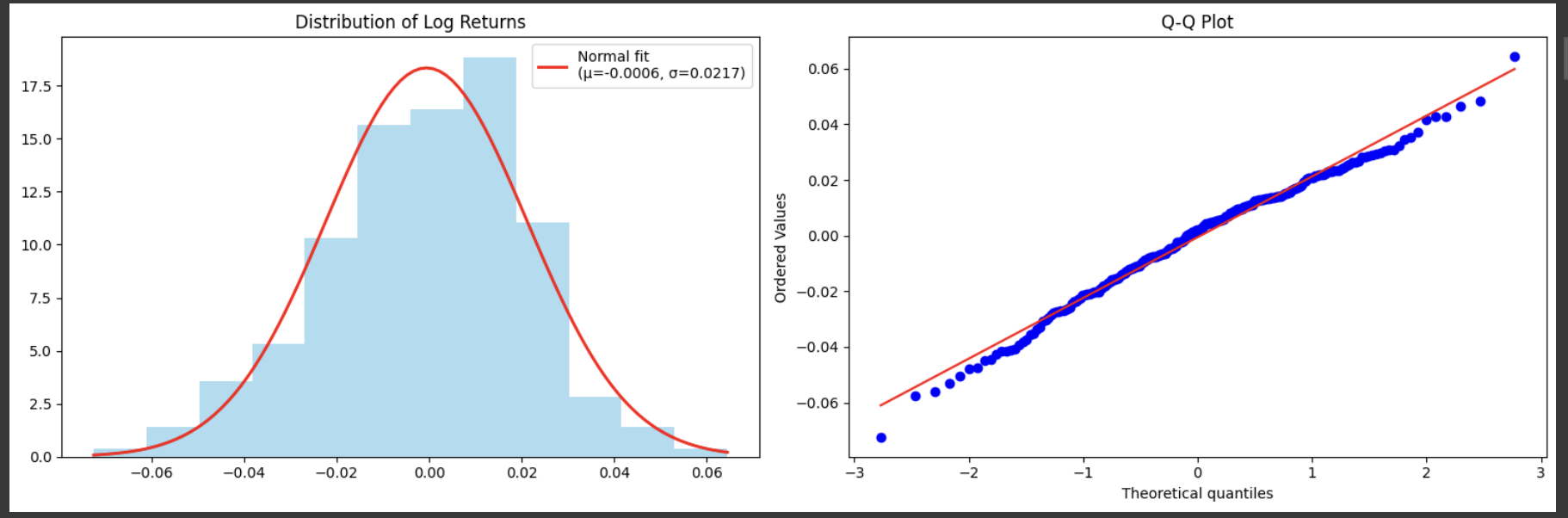
### **2. Data Preprocessing**

This section preprocesses the futures data, including converting date formats, sorting data, removing duplicates, calculating log returns, and handling missing values.



### **3. Data Validation for Geometric Brownian Motion (GBM)**

This section validates the assumption of Geometric Brownian Motion for the log returns. It uses statistical tests like Shapiro-Wilk and Anderson-Darling tests and visualizes the distribution using histograms and Q-Q plots.





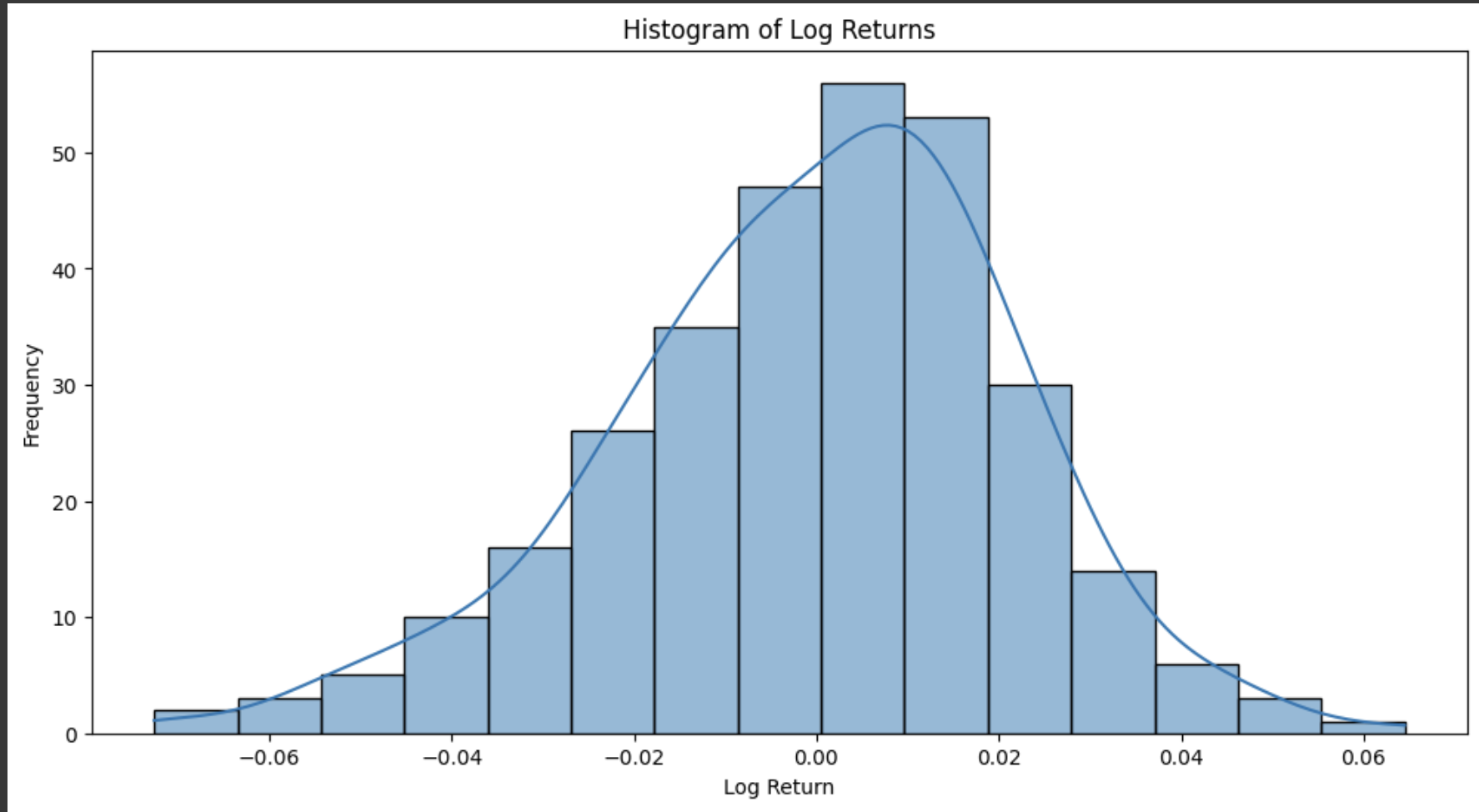
### **4. Descriptive Statistics**

This section generates descriptive statistics (mean, standard deviation, min, max) for the 'Close' and 'Log Return' columns.

[Insert Image of Descriptive Statistics Table Here]

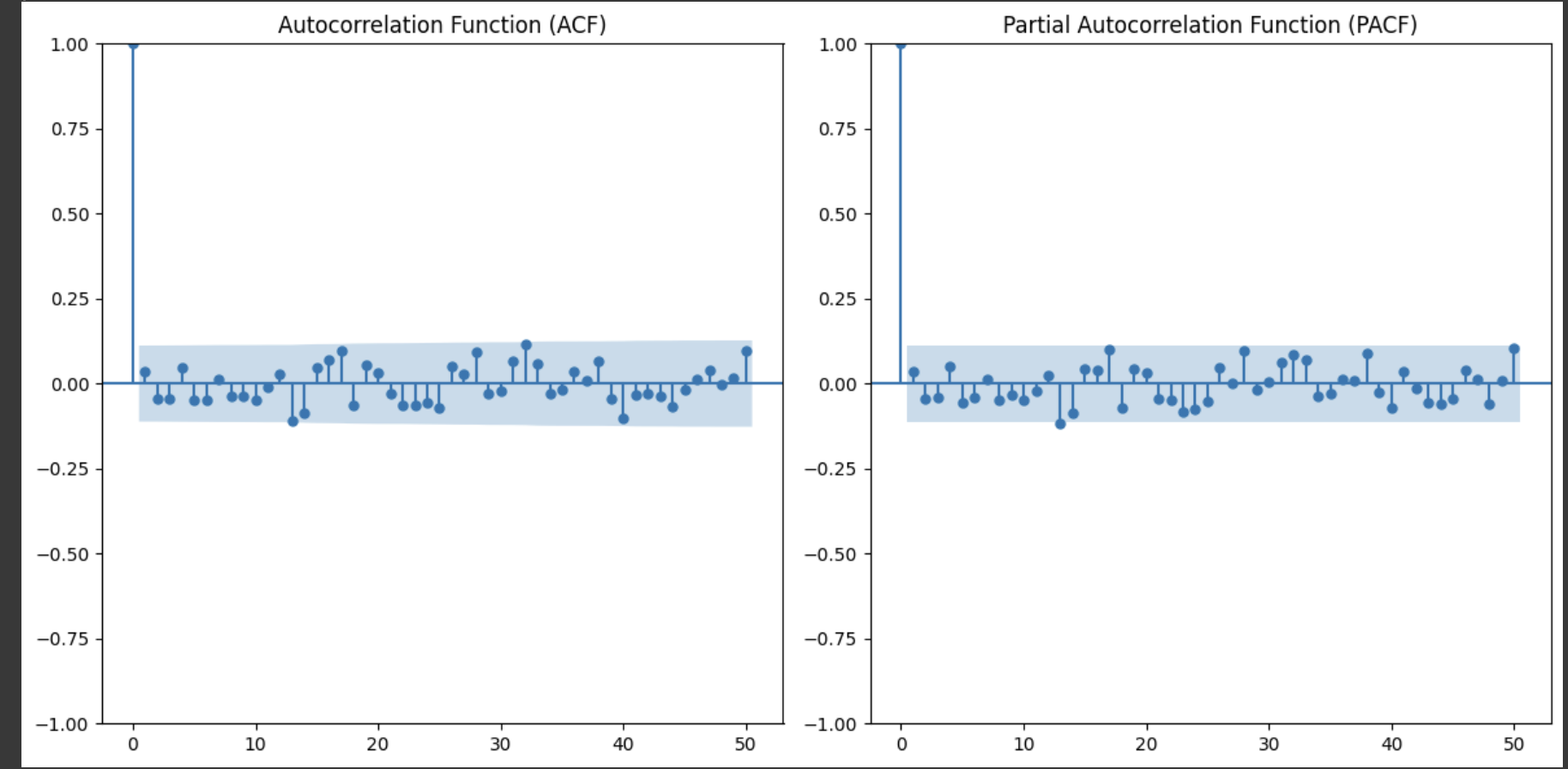
### **5. Log Returns Histogram and Q-Q Plot**

This section visualizes the distribution of log returns using a histogram.



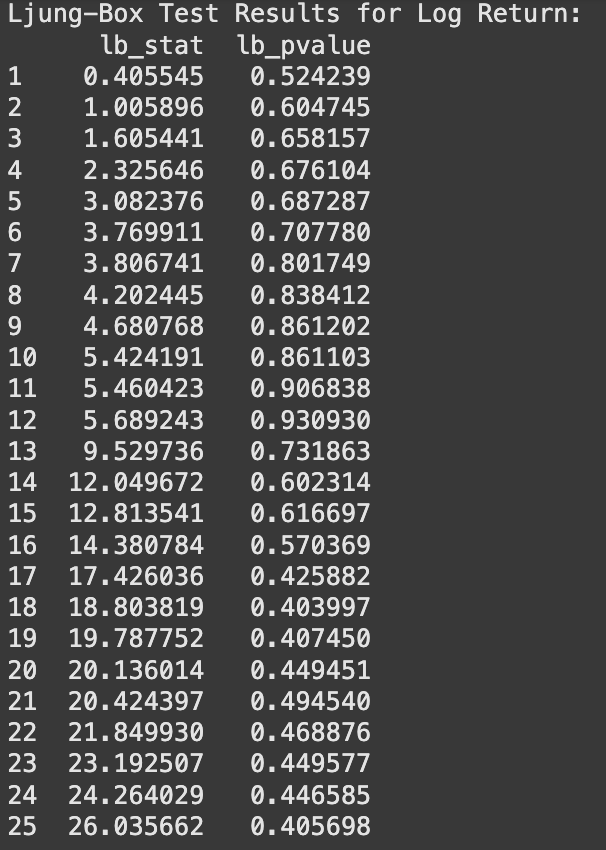
### **6. ACF-PACF Plots**

This section plots the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to analyze the serial correlation of log returns. An Augmented Dickey-Fuller test is also performed to check for stationarity.



### **7. Ljung-Box Test**

This section performs the Ljung-Box test to check for autocorrelation in the log returns.



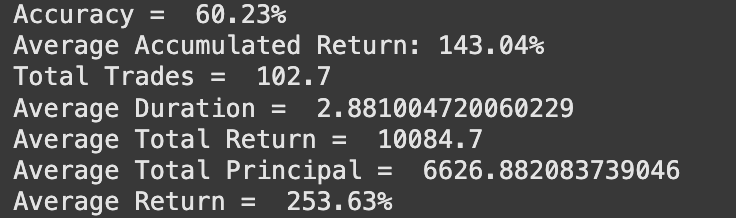
### 

### **8. BDS Test**

This section performs the BDS test to check for non-linearity in the data.

### **9. Strategy 1**

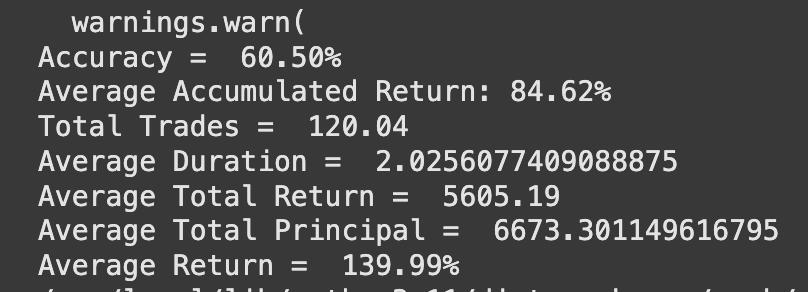
This section implements the first trading strategy, which uses GARCH volatility forecasts and simulated price paths to generate buy/sell signals. It calculates and prints performance metrics like accuracy and average accumulated return across multiple runs.



### 

### **10. Strategy 2**

This section implements the second trading strategy, which is similar to Strategy 1 but may include additional logic or parameters for generating trading signals. It also calculates and prints performance metrics.



## **Usage**

1. Place the Future\_CRUDE\_2005\_2023.csv file in the same directory as the Python script.
2. Install the required libraries.
3. Run all the cells present in the ipynb file.

## **Further Improvements**

* Parameter optimization for trading strategies.
* Backtesting with transaction costs.
* Comparison with other trading strategies.
* More robust risk management.
* Incorporate other market factors.