An Exploration and Deployment of a Quantitative Trading Strategy on HO and COKE Futures Markets

Math Methods for Financial Price Analysis (Spring 2023)

Under guidance of Professor Alexei Chekhlov

Department of Mathematics at Columbia University

Overview

Introduction

- We will be introducing the markets along with their basic descriptive statistics

Statistical Testing

- We will be diving deep into our results from the push-response and variance ratio tests

Backtesting

- We will go into the methodology and trade results obtained from the backtesting

Overview of the Introduction

- Background of the products traded
- HO
- Historical Prices for HO
- Descriptive Statistics for HO
- COKE
- Historical Prices for COKE
- Descriptive Statistics for COKE

Background of the products traded

What are futures?

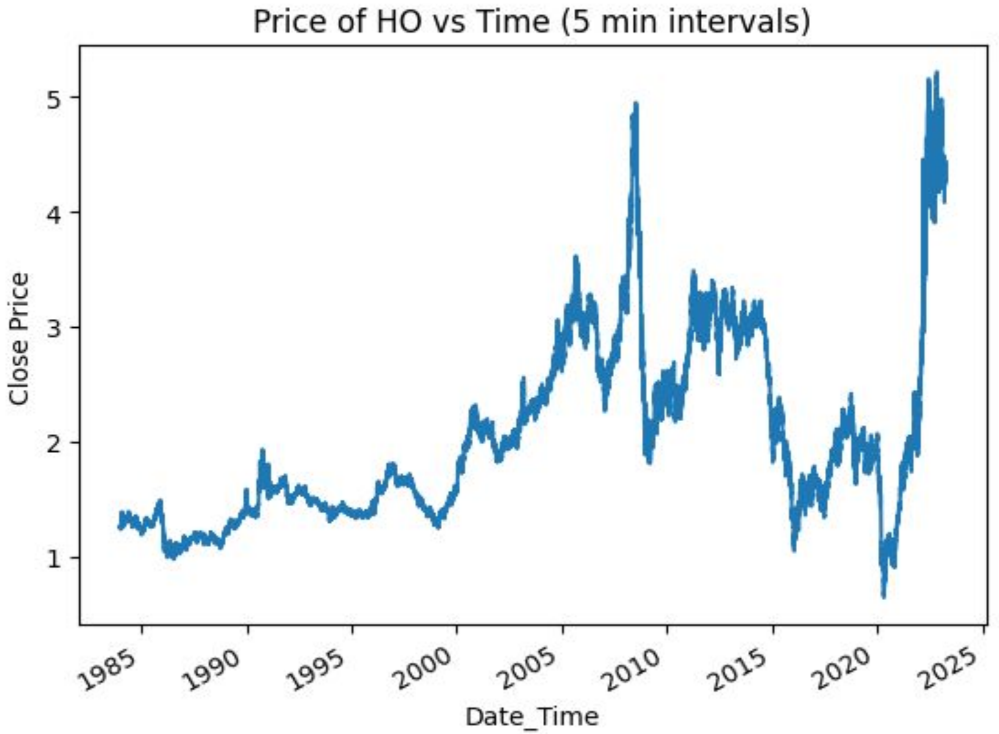
- A futures contract is a binding agreement between a seller and a buyer to make (seller) and to take (buyer) delivery of the underlying commodity (or financial instrument) at a specified future date with agreed upon payment terms.
- Futures contracts protect suppliers and producers from price changes.
- For example, a corn farmer and a cereal company may enter into a futures agreement to lock in a price for a future delivery of corn in a particular month.

HO(Heating Oil)

What do we know about heating oil?

- Heating oil is a heavy fuel oil that is refined from crude oil. Heating oil is also known as No. 2 fuel oil
- US New York Harbor ULSD 62 Grade Future, Access the deep liquidity of Heating Oil products, Currently traded over 180 million barrels every day in 94 different countries on NYMEX
- Heating oil prices are highly correlated with crude oil prices, although heating oil prices are also subject to swift supply and demand shifts due to weather changes or refinery shutdowns.

Historical Close Prices vs Date_Time for HO



- Data is available for past 40 years

Descriptive Statistics for HO

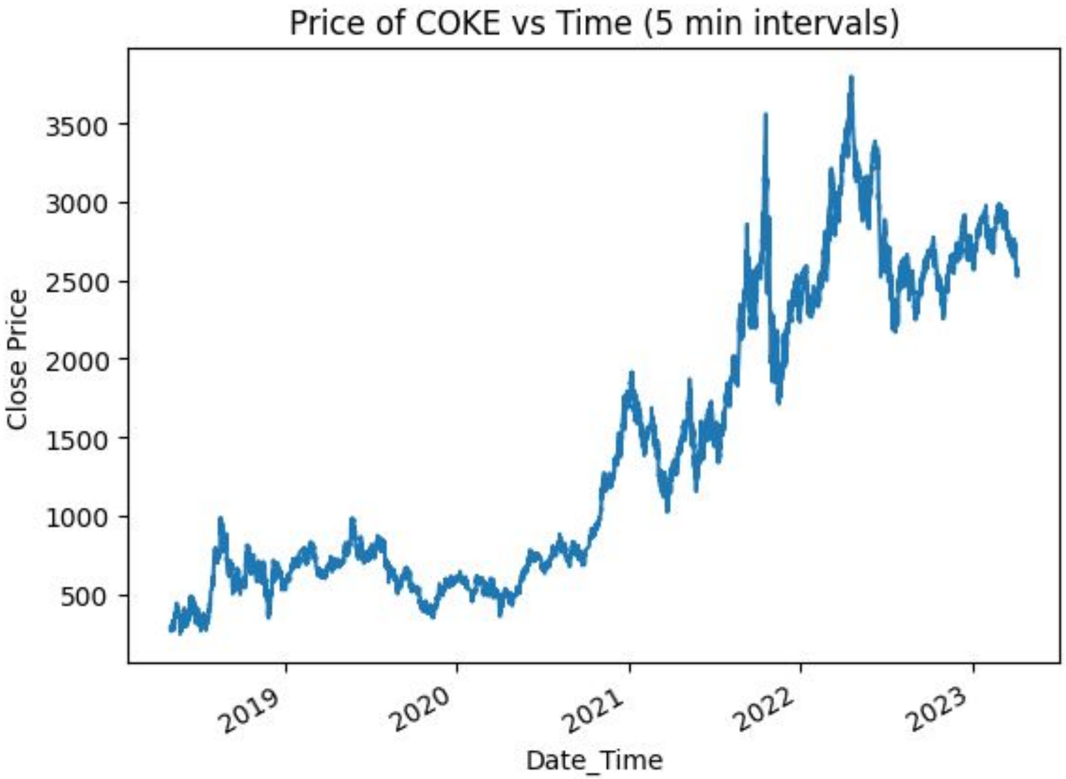
```
Average price change per 5 mins: 5.067469938530l1e-06
Standard deviation of price per 5 mins: 0.820672387147515
Average price of ticker: 2.0805955434811008
Max price: 5.2158 on 2022-11-04 13:40:00
Min price: 0.6427 on 2020-04-27 14:20:00
Max 5-min price increase: 0.41089999999999998
Min 5-min price decrease: -0.53499999999999997
```

```
Average price change per 15 mins: 1.4322639575164828e-05
Standard deviation of price per 15 mins: 0.8225495898259614
Average price of ticker: 2.0786790896988663
Max price: 5.212 on 2022-11-04 13:45:00
Min price: 0.6442 on 2020-04-27 14:15:00
Max 15-min price increase: 0.41140000000000043
Min 15-min price decrease: -0.56929999999999997
```

COKE

- **What do we know about coke?**
- The most-active coking coal futures on the Dalian Commodity Exchange, JKEE on the metals group of DCE.
- Metallurgical coke is produced by destructive distillation of coal in coke ovens. Prepared coal is "coked", or heated in an oxygen-free atmosphere until all volatile components in the coal evaporate. The material remaining is called coke.
- The coke industry plays an important role in China's booming economy, with annual output and consumption both above 300 million tonnes in recent years.

Historical Close Prices vs Date_Time for COKE



- Data is available for past 5 years

Descriptive Statistics for COKE

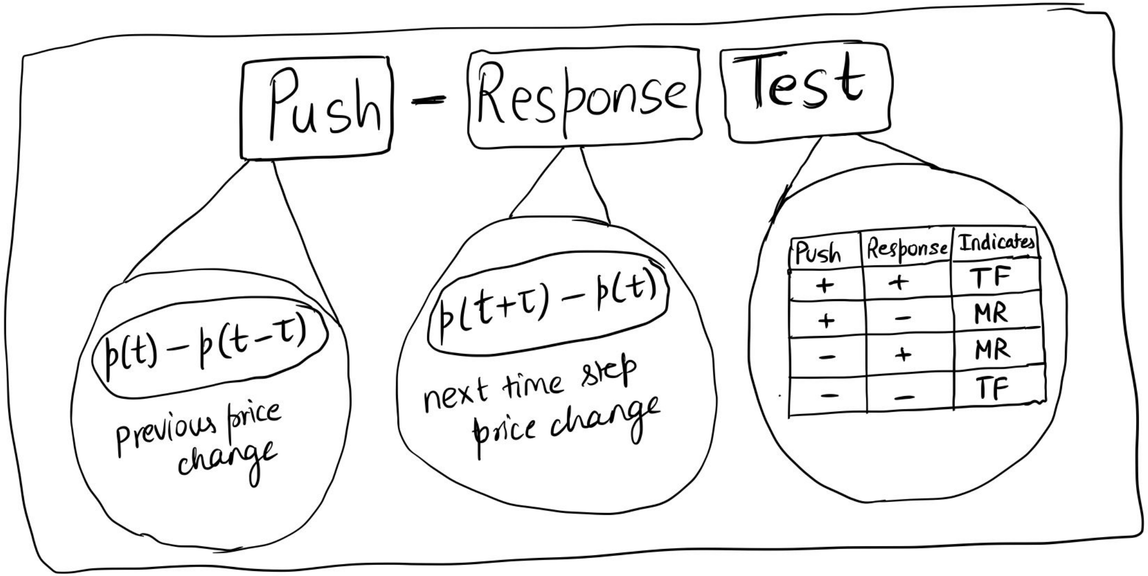
```
Average price change per 5 mins: 0.02611720043056054
Standard deviation of price per 5 mins: 935.9957385270317
Average price of ticker: 1430.5000810185186
Max price: 3798.5 on 2022-04-19 10:05:00
Min price: 245.0 on 2018-05-23 13:30:00
Max 5-min price increase: 238.5
Min 5-min price decrease: -365.0
```

```
Average price change per 15 mins: 0.07381452529357072
Standard deviation of price per 15 mins: 937.1369281667783
Average price of ticker: 1276.8529614325068
Max price: 3631.5 on 2022-04-19 10:15:00
Min price: 82.5 on 2018-05-23 13:30:00
Max 15-min price increase: 213.0
Min 15-min price decrease: -365.0
```

Overview of Statistical Testing on Markets

- Push-response Test Overview
- Push-response Test on HO
- Push-response Test on COKE
- Push-response Test with Regression Overview
- Push-response Test for HO with Regression
- Push-response Test for COKE with Regression
- Variance Ratio Test Overview
- Variance Ratio Test on HO
- Variance Ratio Test on COKE

Push Response Test Overview



⇒ Positive slope of graph means TF

⇒ Negative slope of graph means MR

⇒ Almost zero slope of graph means RW

Please Note: Here, TF is Trend Following
MR is Mean Reversion
RW is Random Walk

Push Response Test with Regression Overview

⊛ **Levy's distribution** $\Rightarrow \sigma(\tau) \propto \tau^{\gamma} \Rightarrow \sigma(\tau) = A\tau^{\gamma}$

$$\Rightarrow \underbrace{\boxed{\log_{10} \sigma(\tau)}}_{\text{dependent variable}} = \underbrace{\boxed{\log_{10} A}}_{\text{intercept}} + \underbrace{\boxed{\gamma}}_{\text{slope}} \underbrace{\boxed{\log_{10} \tau}}_{\text{independent variable}}$$

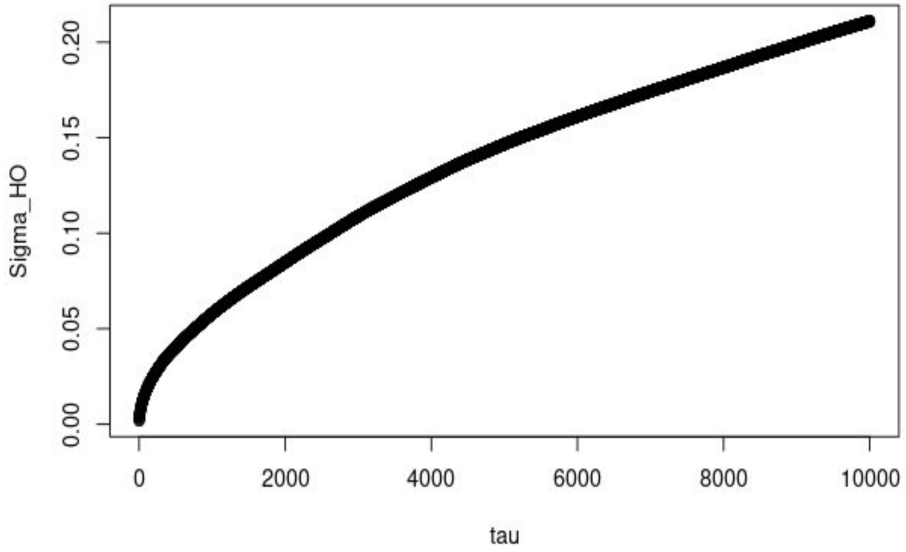
Generally,

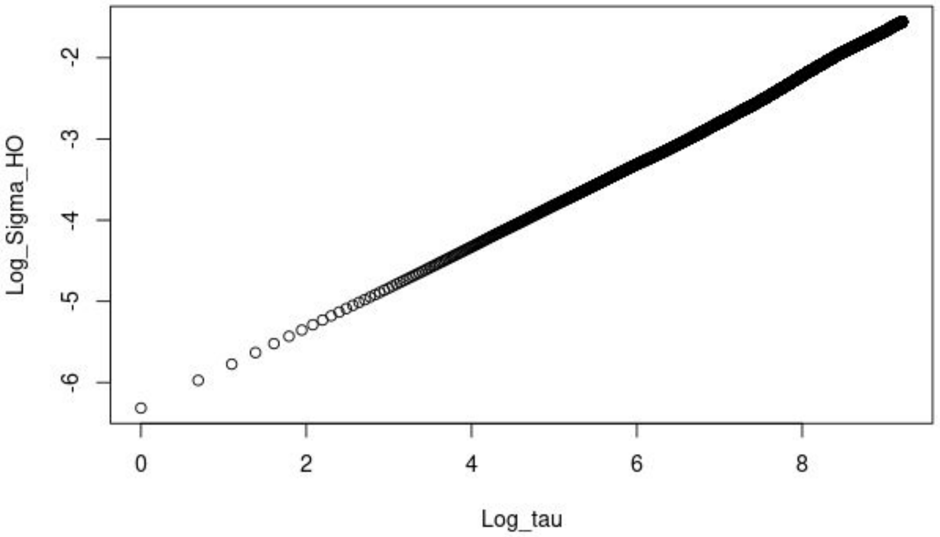
$\boxed{\text{slope}\ (\gamma) \simeq 0.5} \Rightarrow \boxed{\text{Brownian Motion}}$

$\boxed{\text{slope}\ (\gamma) > 0.5} \Rightarrow \boxed{\text{Momentum effect}}$

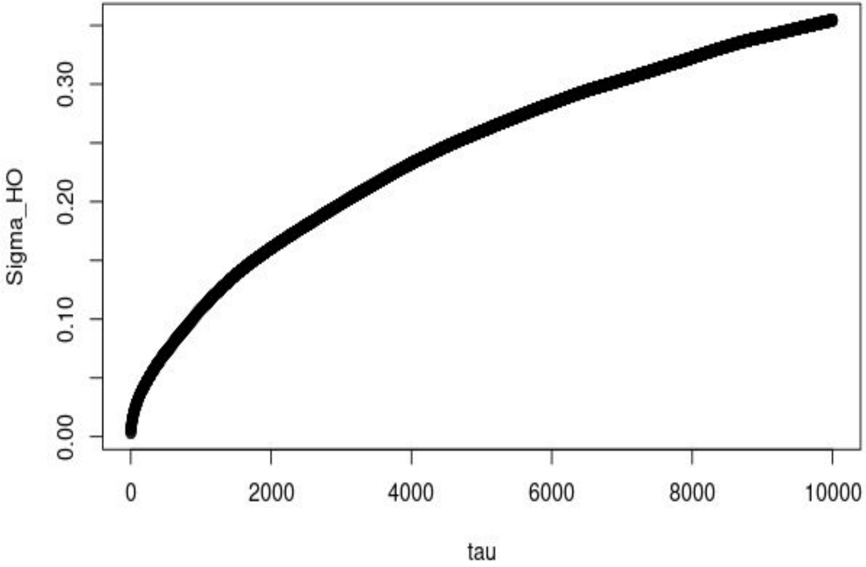
$\boxed{\text{slope}\ (\gamma) < 0.5} \Rightarrow \boxed{\text{Contrarian effect}}$

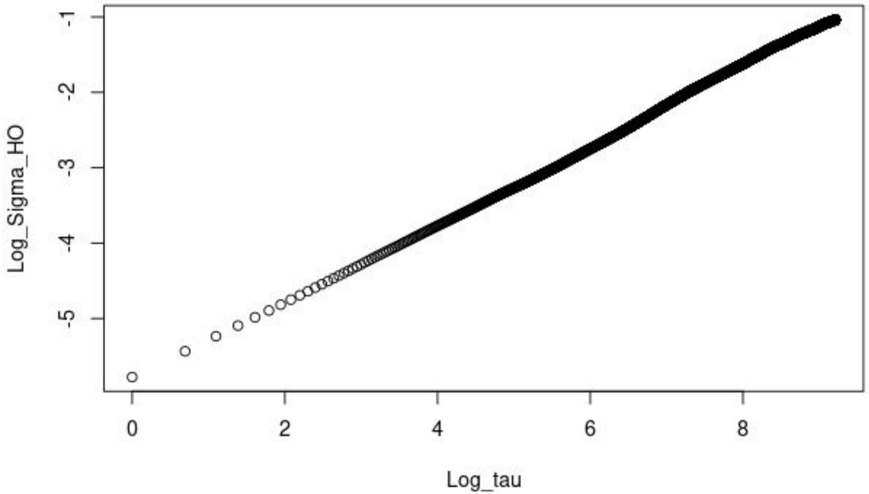
Push Response Test for HO (5 min) with Regression





Push Response Test for HO (15 min) with Regression





Regression Results for HO

(5-min)

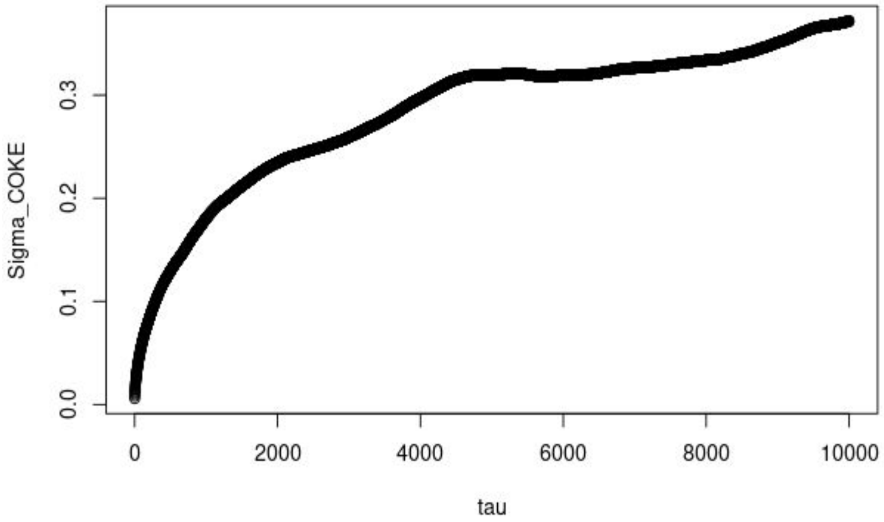
```
> # generating log-log linear model and displaying summariHO
> HO_log_linear_model = lm(Log_Sigma_HO ~ Log_tau, data = log_tau_sigma_df)
> summary(HO_log_linear_model)

Call:
lm(formula = Log_Sigma_HO ~ Log_tau, data = log_tau_sigma_df)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.034868 -0.002494  0.001221  0.010328  0.289142 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.6032707  0.0016206   -4075   <2e-16 ***
Log_tau      0.5479881  0.0001959    2797   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01954 on 9998 degrees of freedom
Multiple R-squared:  0.9987,	Adjusted R-squared:  0.9987 
F-statistic: 7.822e+06 on 1 and 9998 DF,  p-value: < 2.2e-16
```

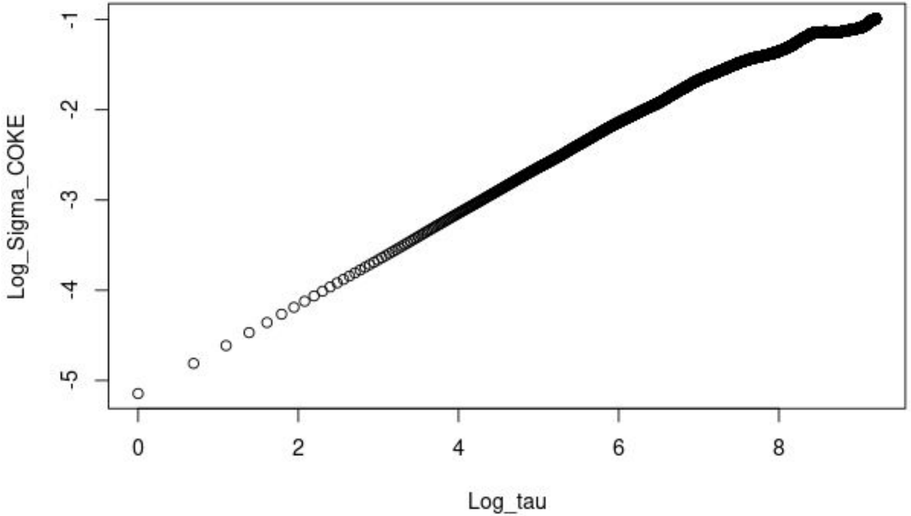
(15-min)

```
> # generating log-log linear model and displaying summariHO
> HO_log_linear_model = lm(Log_Sigma_HO ~ Log_tau, data = log_tau_sigma_df)
> summary(HO_log_linear_model)

Call:
lm(formula = Log_Sigma_HO ~ Log_tau, data = log_tau_sigma_df)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.053793 -0.017044  0.007214  0.019598  0.109876 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.886801   0.001844   -3192   <2e-16 ***
Log_tau      0.530904   0.000223    2381   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02224 on 9998 degrees of freedom
Multiple R-squared:  0.9982,	Adjusted R-squared:  0.9982 
F-statistic: 5.67e+06 on 1 and 9998 DF,  p-value: < 2.2e-16
```

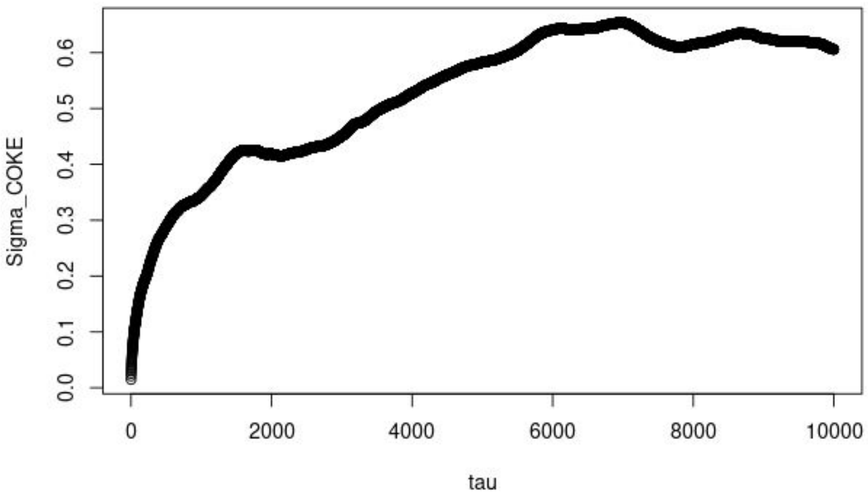
Slope > 0.5 ⇒ Momentum or Trend Following Behavior

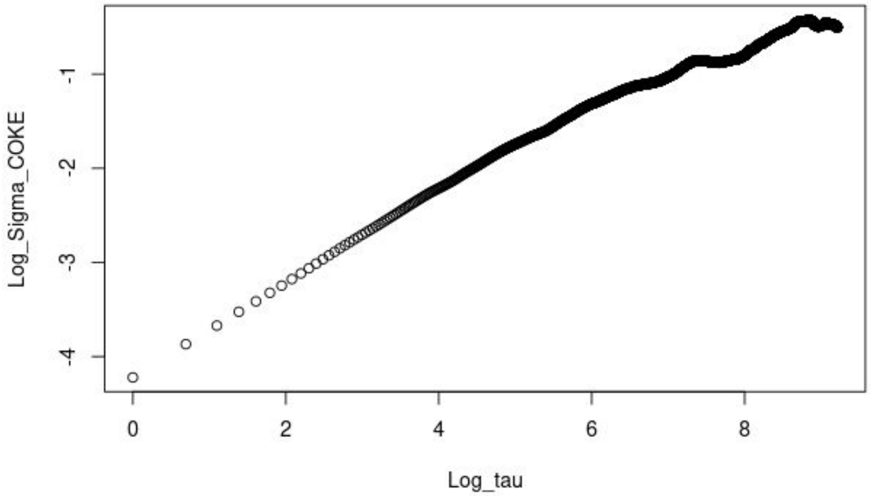
Push Response Test for COKE (5-min) with Regression





Push Response Test for COKE (15-min) with Regression





Regression Results for COKE

(5-min)

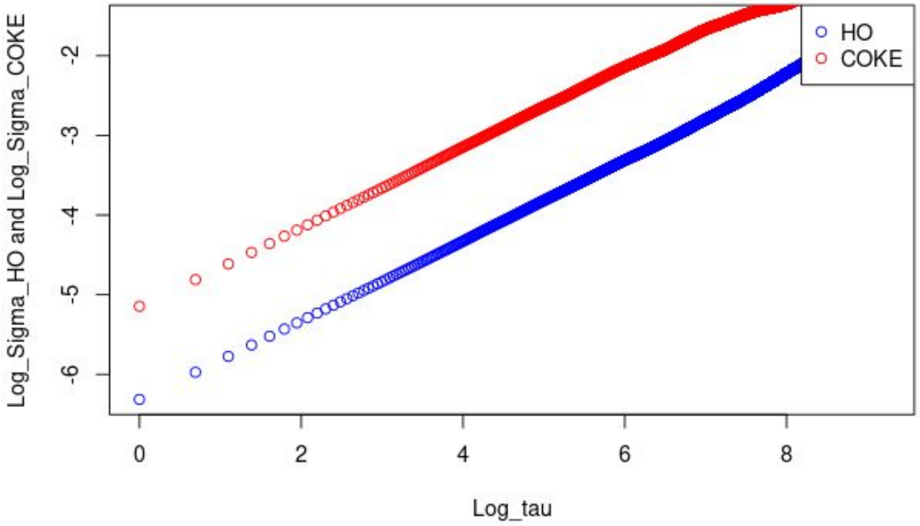
```
> COKE_log_linear_model = lm(Log_Sigma_COKE ~ Log_tau, data = log_tau_sigma_df)
> summary(COKE_log_linear_model)

Call:
lm(formula = Log_Sigma_COKE ~ Log_tau, data = log_tau_sigma_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.84773 -0.05223  0.00646  0.05903  0.08536 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.2968972  0.0058667  -732.4   <2e-16 ***
Log_tau      0.3640468  0.0007093   513.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07075 on 9998 degrees of freedom
Multiple R-squared:  0.9634,	Adjusted R-squared:  0.9634 
F-statistic: 2.634e+05 on 1 and 9998 DF,  p-value: < 2.2e-16
```

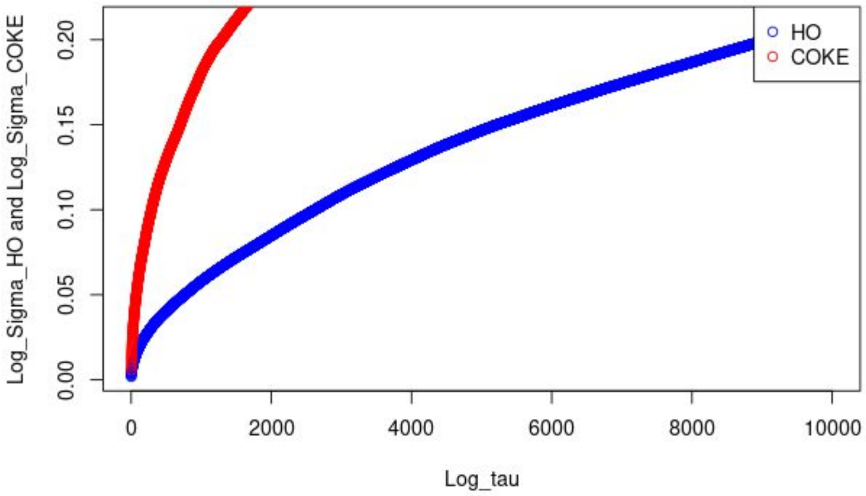
(15-min)

```
> COKE_log_linear_model = lm(Log_Sigma_COKE ~ Log_tau, data = log_tau_sigma_df)
> summary(COKE_log_linear_model)

Call:
lm(formula = Log_Sigma_COKE ~ Log_tau, data = log_tau_sigma_df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.96891 -0.04819  0.01153  0.05080  0.09297 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.2524496  0.0053993  -602.4   <2e-16 ***
Log_tau      0.3131949  0.0006528   479.8   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06511 on 9998 degrees of freedom
Multiple R-squared:  0.9584,	Adjusted R-squared:  0.9584 
F-statistic: 2.302e+05 on 1 and 9998 DF,  p-value: < 2.2e-16
```

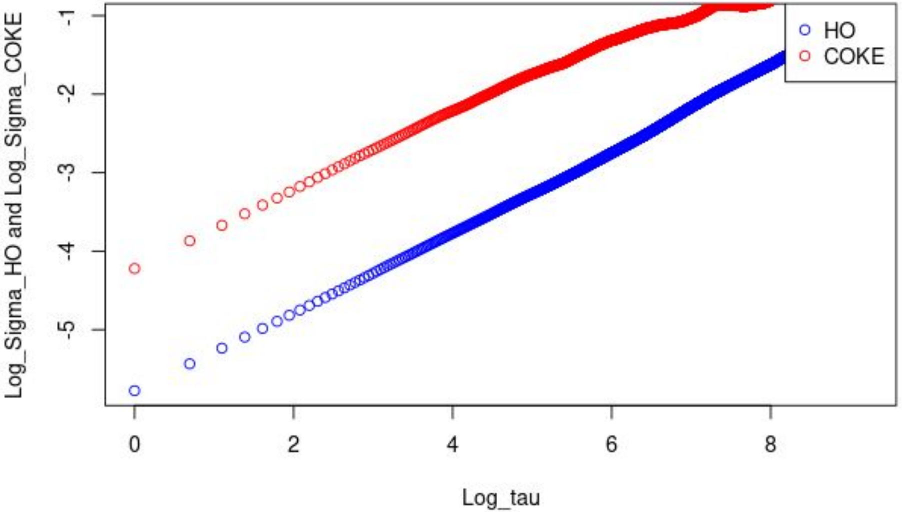
Slope < 0.5 ⇒ Contrarian or Mean Reverting Behavior

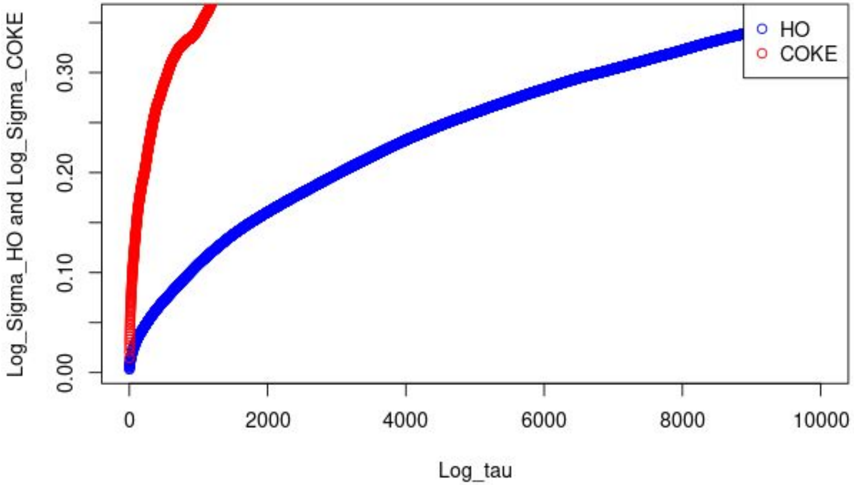
Push Response Test Comparison (5-min) with Regression





Push Response Test Comparison (15-min) with Regression





Variance Ratio Test Overview

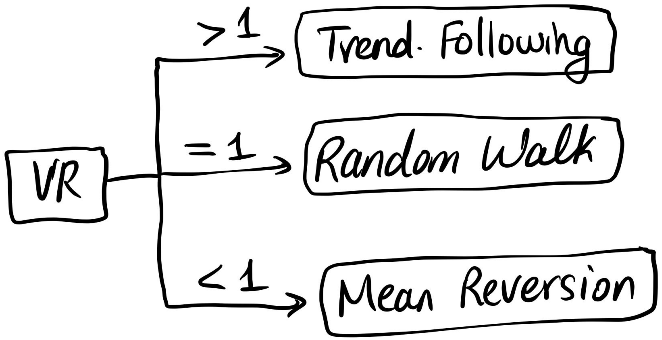
$$VR(q) = \frac{Var(q)}{q \cdot Var(1)}$$

$$VR(q) = 1 + 2\sum_{k=1}^{q-1}\left(1 - \frac{k}{q}\right)\rho_k$$

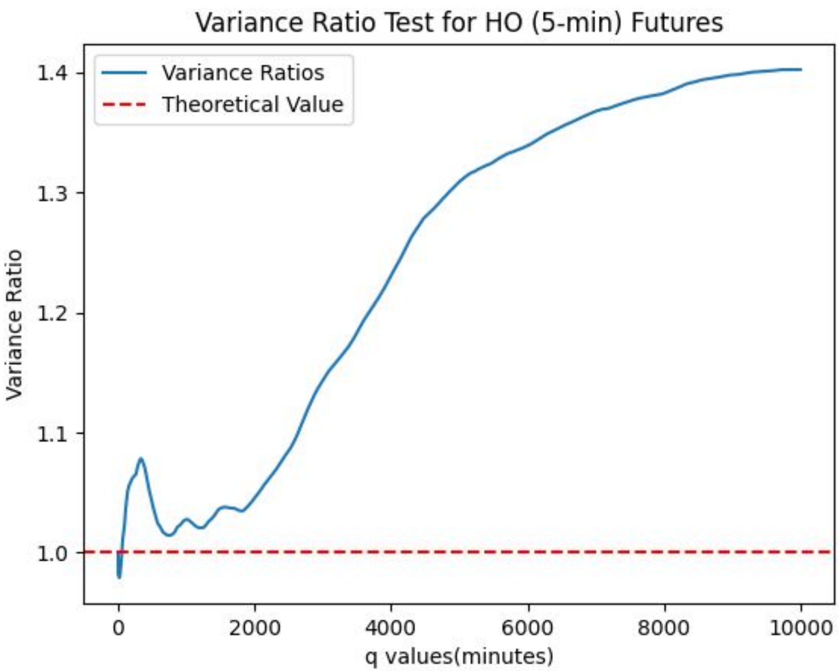
$\rho_1 = corr(\Delta p_1, \Delta p_2)$ for example

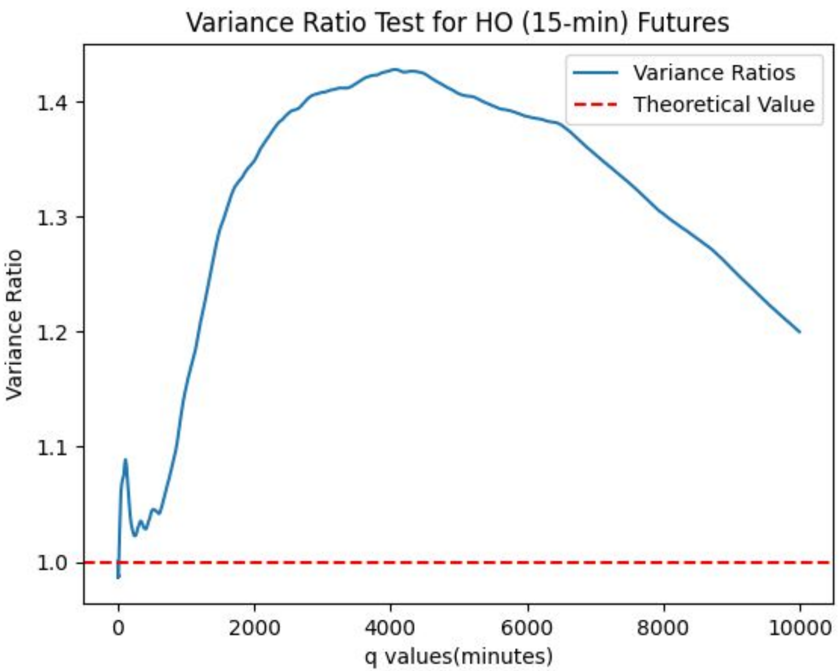
Here, $\Delta p_1 = p(t+\tau) - p(t)$

$\Delta p_2 = p(t+2\tau) - p(t+\tau)$



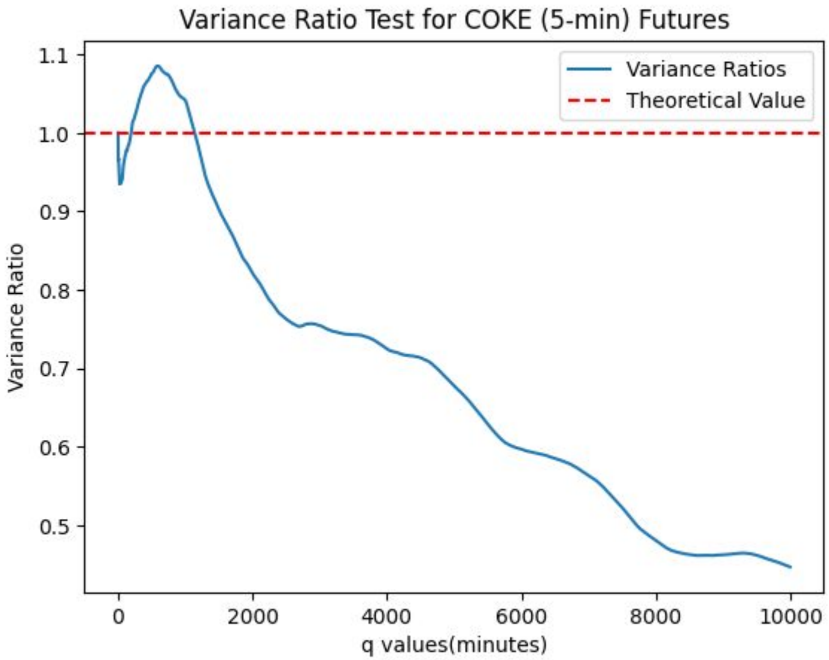
Andrew Lo’s Variance Ratio Test for HO (5 & 15 min)

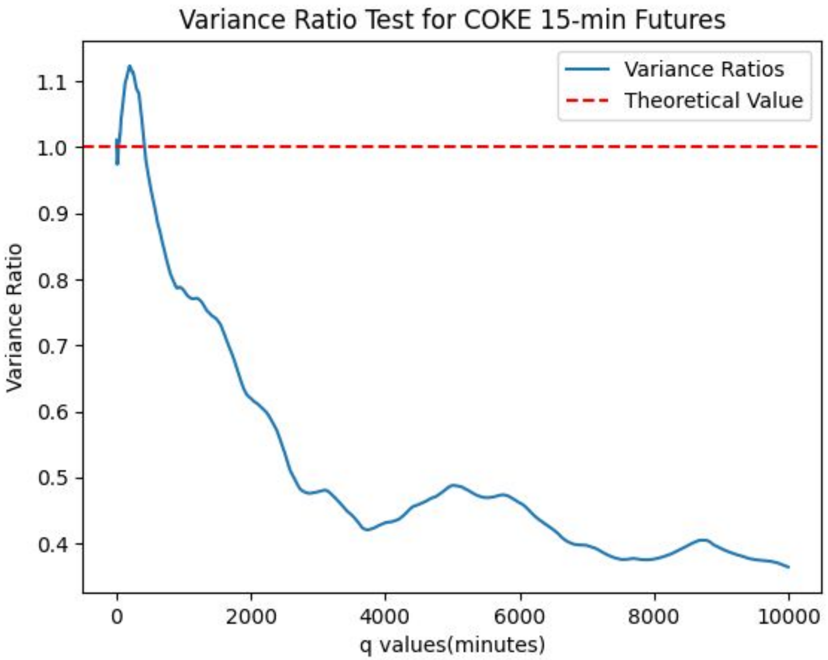




Long Term: VR >1 ⇒ Trend Following Behavior

Andrew Lo’s Variance Ratio Test for COKE (5 & 15 min)





Long Term: VR < 1 ⇒ Mean Reverting Behavior

Overview of Backtesting

- Methodology
- Trade Results for HO
- Trade Results for COKE
- Future Work
- Conclusion and Acknowledgements
- [BONUS] The End

Methodology

- We had originally written the trade code in Python but that's not fast enough so we tried to run the 15 min data from it.

- The trade parameters over 'Channel length' and 'Stop Percent' are optimised using an automatic hyperparameter (parameters in a learning algorithm) framework in Python.

- We also ran some code in MATLAB too because it was a faster than their Python counterparts and was a good sanity check too because it was able to handle 5-min data!

- <u>Technologies used:</u> MATLAB, Python (Numpy, Pandas, Matplotlib, Scikit-learn, Stats), R.

Parameters for MATLAB

```
% For CO %%%%%%%%%%%%%%%
%dataFile='CO-5min.csv';
%%%%%%%%%%%%%%%%%%%%%%%%

% For COKE %%%%%%%%%%%%%%%
dataFile='COKE-5min.csv';
%%%%%%%%%%%%%%%%%%%%%%%%%%

% For HO %%%%%%%%%%%%%%%
%dataFile='HO-5min.csv';
%%%%%%%%%%%%%%%%%%%%%%%%

% For HO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%inSample=[datenum('01/01/1980'),datenum('01/01/2000')];
%outSample=[datenum('06/14/2011'),datenum('04/14/2014')];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% For CO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%inSample=[datenum('08/01/2003'),datenum('08/01/2013')];
%outSample=[datenum('08/02/2013'),datenum('04/06/2023')];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% For COKE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
inSample=[datenum('05/03/2018'),datenum('05/03/2020')];
outSample=[datenum('05/04/2020'),datenum('04/06/2022')];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% For HO %%%%%%
%barsBack=17001;
%slpg=47;
%PV=42000;
%%%%%%%%%%%%%%%

% For CO %%%%%%
%barsBack=17001;
%slpg=48;
%PV=1000;
%%%%%%%%%%%%%%%

% For COKE %%%%
barsBack=17001;
slpg=48;
PV=20000;
%%%%%%%%%%%%%%%
```

Trading strategy - Channel Breakout

SIGNAL:
Breaking
Highest High
and Lowest Low

Optimization for three parameters

- **Length** - The length of the lookback window, i.e., the number of previous bars to consider when calculating the highest high and the lowest low.
- **StopPct** - The stop percentage, used to calculate the stop loss level as a percentage of the entry price.
- **BarsBack** - The number of time bars to look back.

Optimizing Best PnL to Max DD ratio:

- ## Output for HO

```
Optimal parameters: Length=5000, StopPct=0.01, BarsBack=21000
```

- ## Output COKE

```
Optimal parameters: Length=10500, StopPct=0.015, BarsBack=9001
```

Optimization code

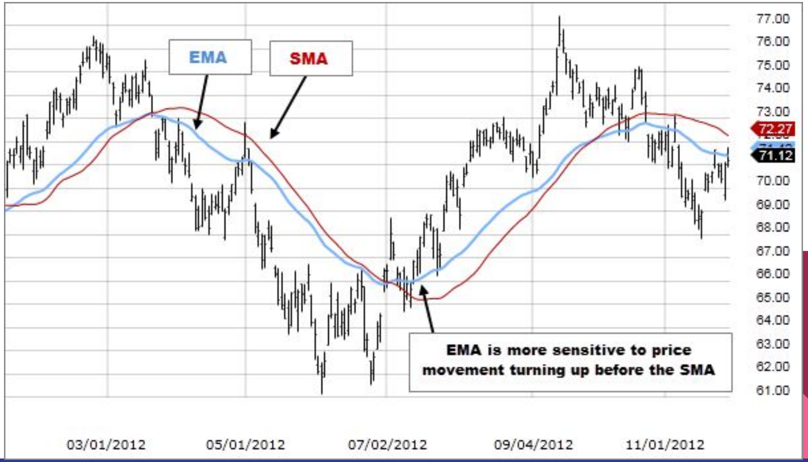
```python
# Define the parameter ranges
Length = np.arange(1000, 15001, 2000)
StopPct = np.arange(0.010, 0.021, 0.004)
BarsBack_range = np.arange(1000, 26001, 5000)
inSample = [pd.Timestamp('2022-03-19').timestamp(), pd.Timestamp('2023-03-20').timestamp()]
outSample = [pd.Timestamp('2023-03-21').timestamp(), pd.Timestamp('2023-04-21').timestamp()]

# Initialize variables to store optimal values
best_pnl_to_max_dd = -np.inf
best_params = None

# Loop through all possible parameter combinations
# (data, length, stop_pct, inSample, outSample, barsBack)
for length, stop_pct, bars_back in product(Length, StopPct, BarsBack_range):
    # Calculate the trading strategy with the current parameter combination
    pnl, max_dd = calculate_trading_strategy(data, length, stop_pct, inSample, outSample, bars_back)
    pnl_to_max_dd = pnl / max_dd

    # Check if the current parameter combination has a higher pnl_to_max_dd
    if pnl_to_max_dd > best_pnl_to_max_dd:
        best_pnl_to_max_dd = pnl_to_max_dd
        best_params = (length, stop_pct, bars_back)

print(f"Optimal parameters: Length={best_params[0]}, StopPct={best_params[1]}, BarsBack={best_params[2]}")
print(f"Best PnL to Max DD ratio: {best_pnl_to_max_dd}")
```

```
Optimal parameters: Length=5000, StopPct=0.01, BarsBack=21000
Best PnL to Max DD ratio: 0.704650446566061
```

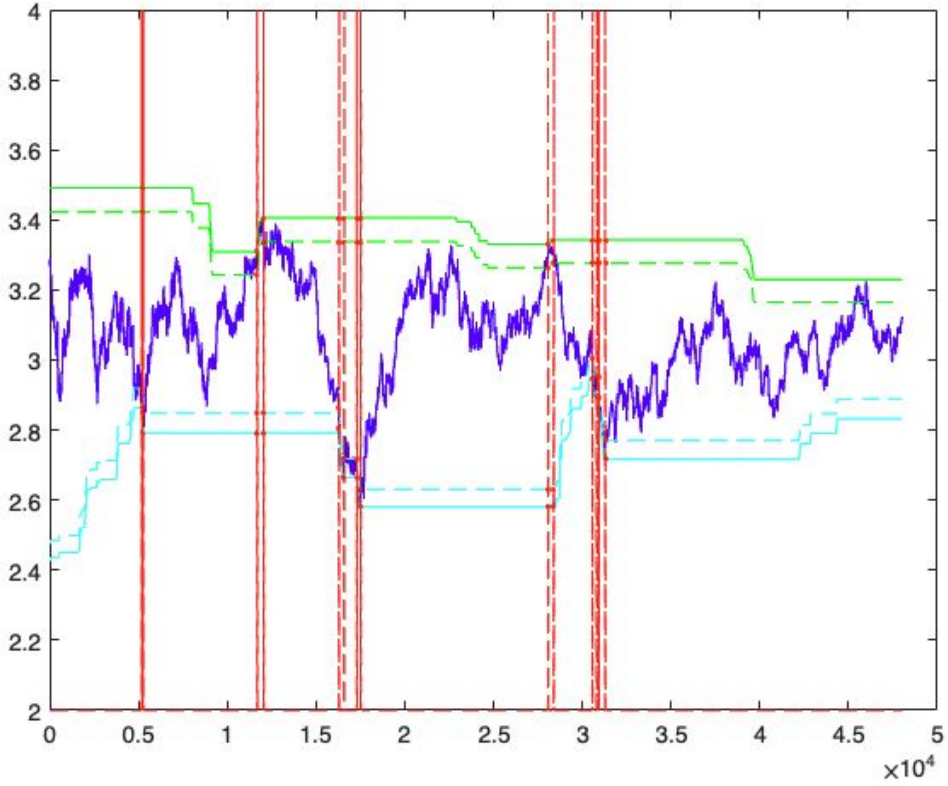
Additional improvements in Signals

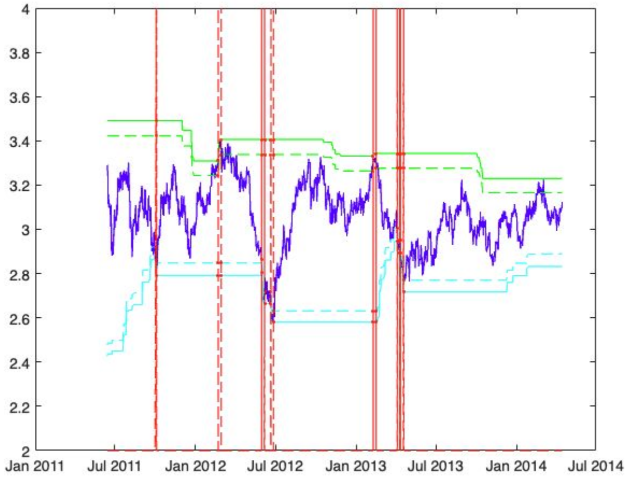
- SMA and Exponential moving average - more trend following signal (for five minutes: 1000/4000)
- Relative Strength Index - RSI = 100 - [100 / (1 + RS)]

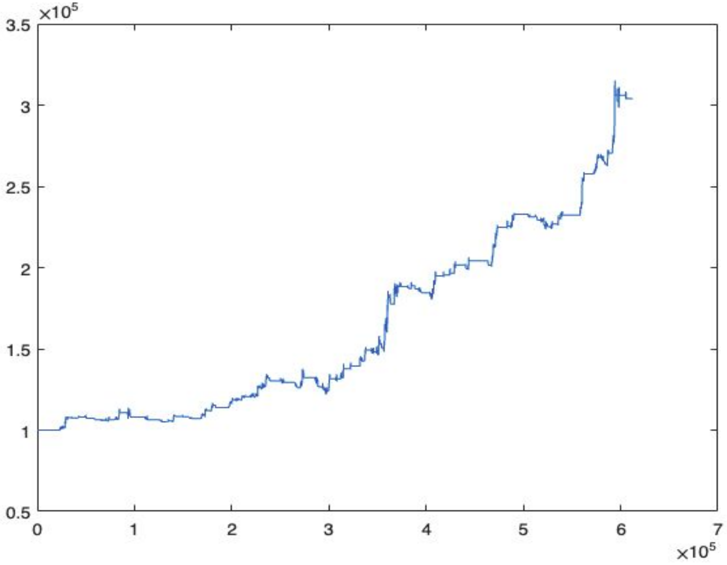
where RS (Relative Strength) is the average gain of up periods divided by the average loss of down periods over a 1000 bins.



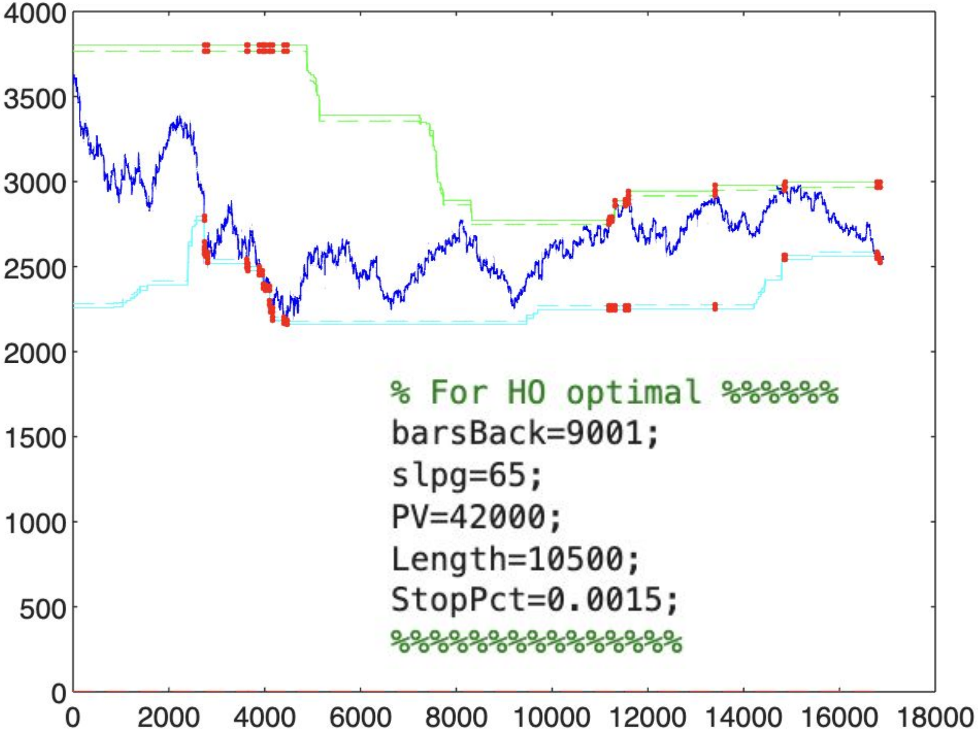
MATLAB: Trade Results for HO (5 min)
20 years / 4 years (default)

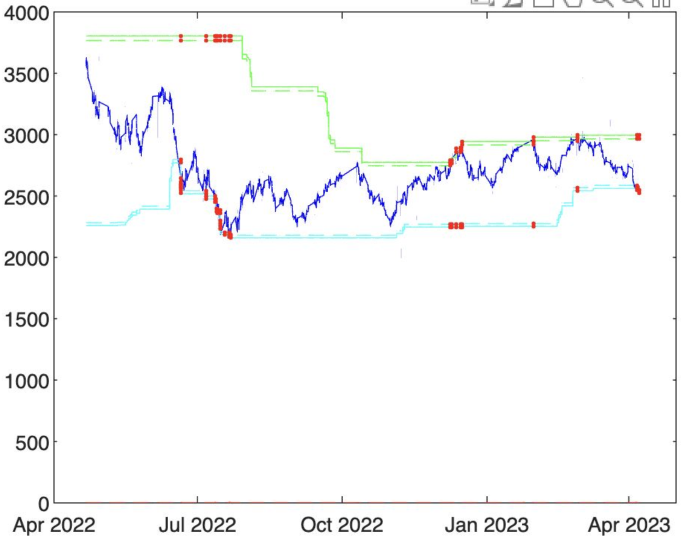


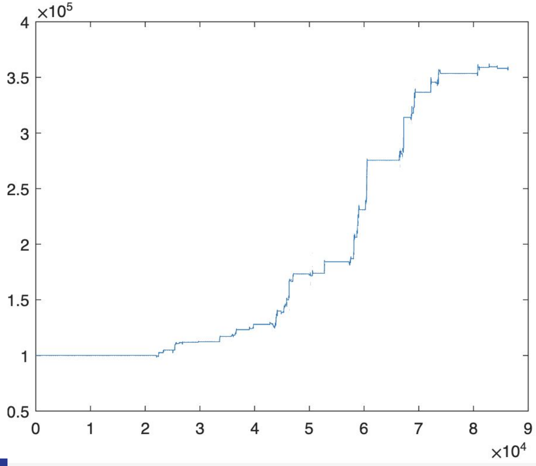




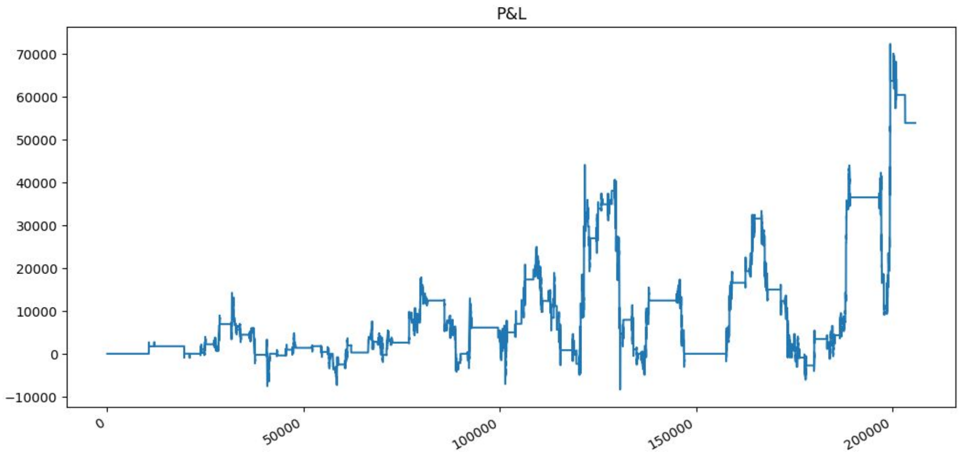
MATLAB: Trade Results for HO (5 min) 10 years / 1 years (optimal parameters)





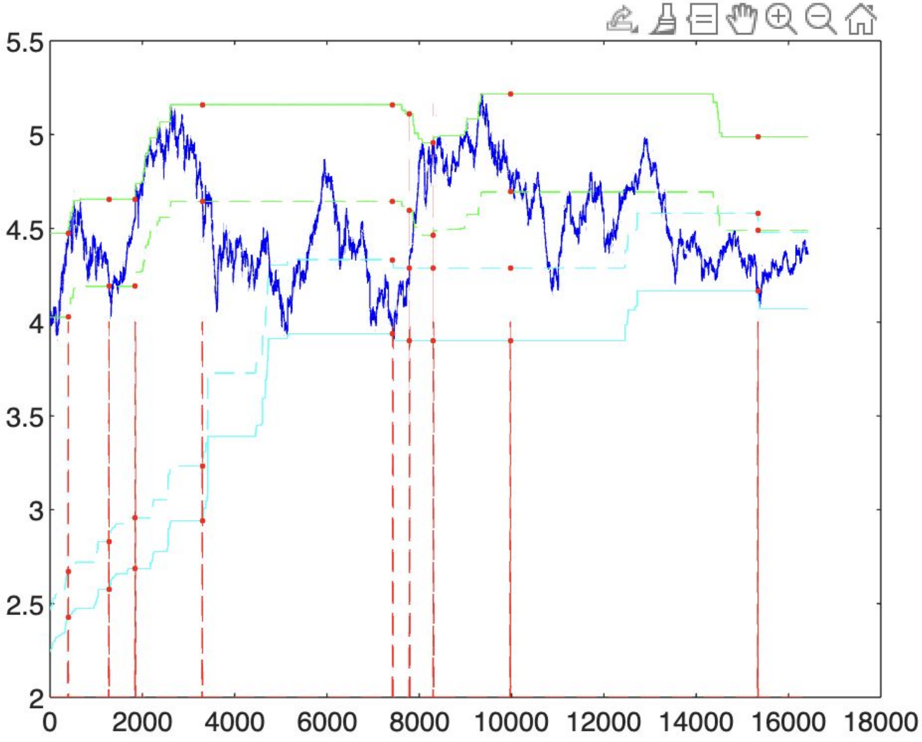


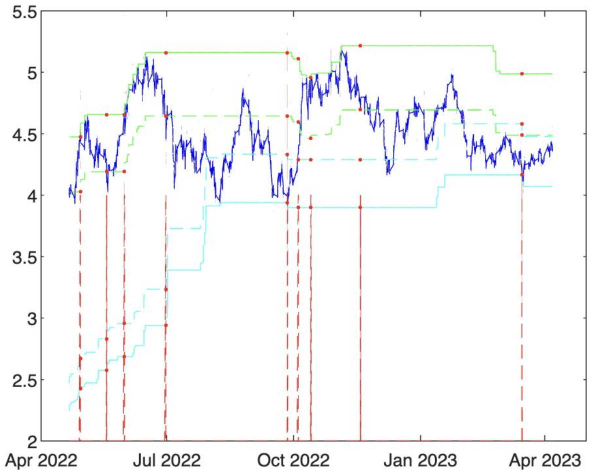
Python: Trade Results for HO (15 min) 1 Year/ 1 Month

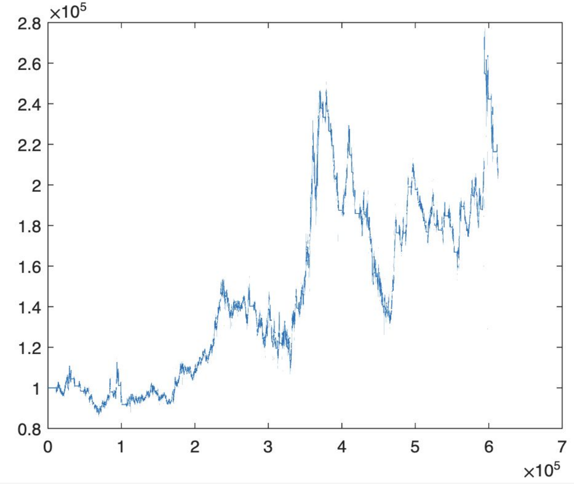


	Value
Average Return	1.079643107360280
Standard Deviation	118.26412771211000
Sharpe Ratio	0.009129083588122800
Total Trades	541.0
Percent Profitable Trades	48.059149722735700
Max Drawdown	-19721.480000000100
Return on Account	16.338186789226700
Average Winner	1763.029230769230
Average Loser	-844.2344341637010
Profit Factor	1.932250725452470

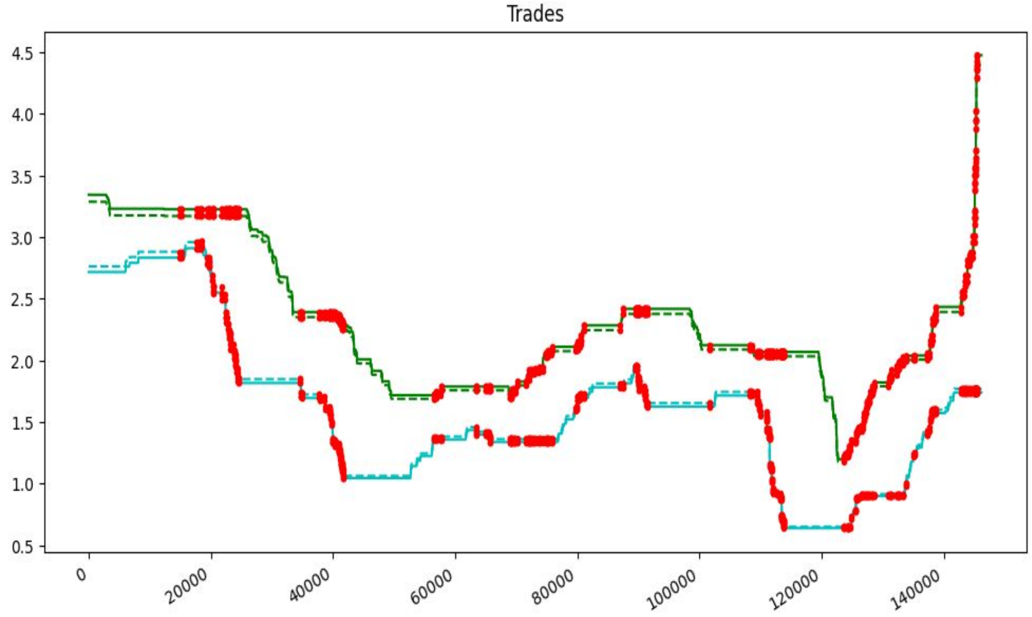
MATLAB: Trade Results for HO (5 min)
10 Years/ 1 Year

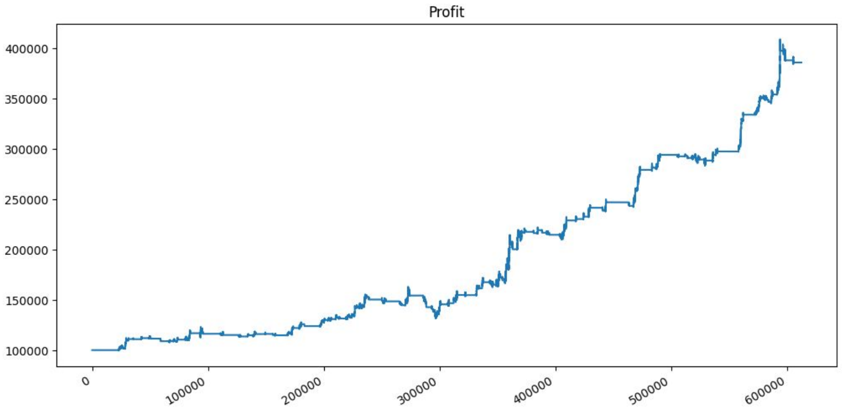


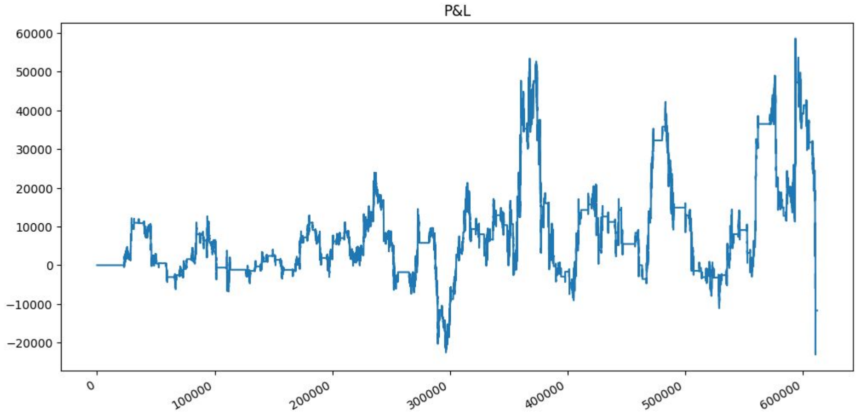




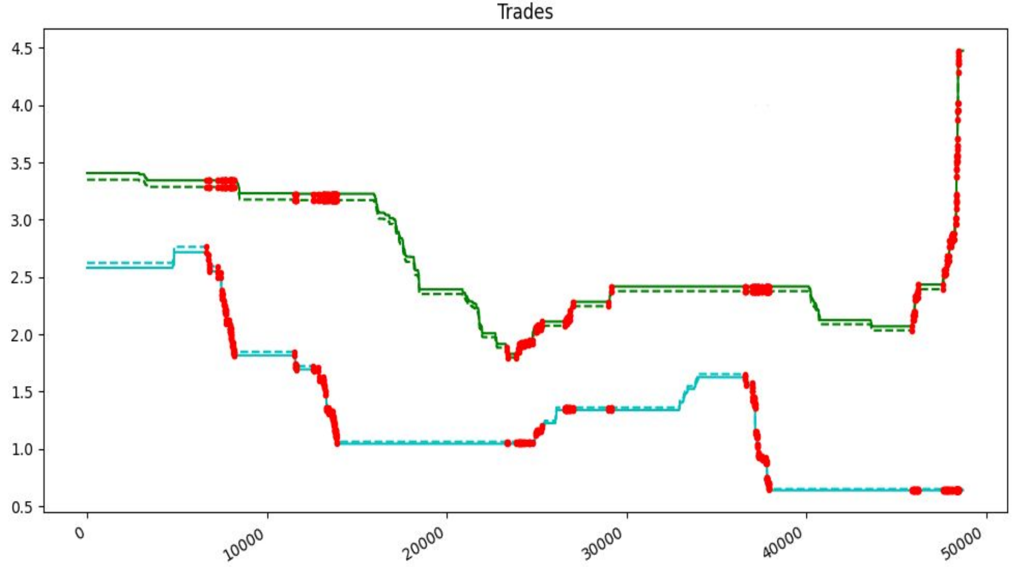
Python: Trade Results for HO (5 min) for 20 years 7 Years

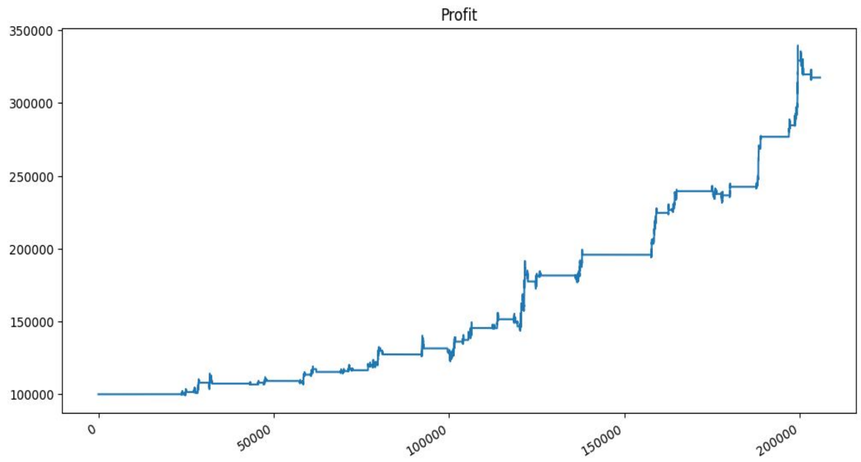


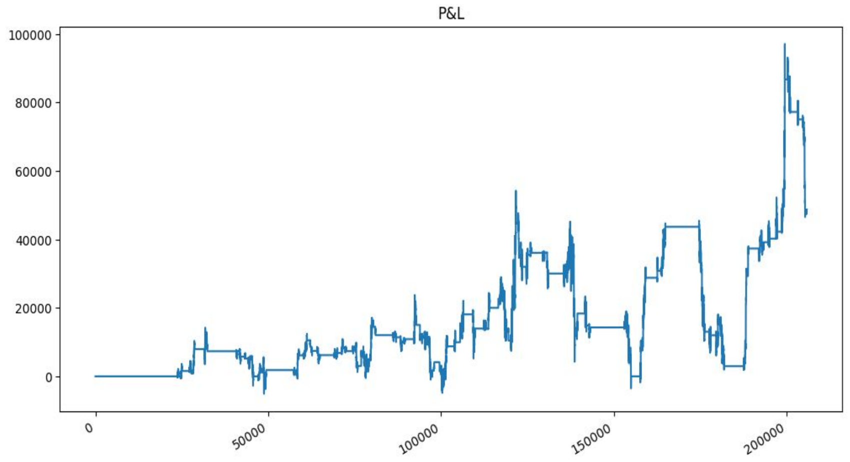




Python: Trade Results for HO (15 min) 10 years/1 year

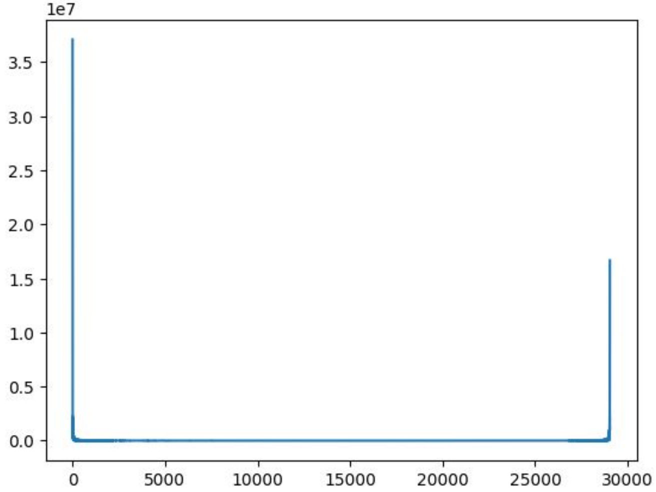






COKE Analysis

Fourier transform on the COKE



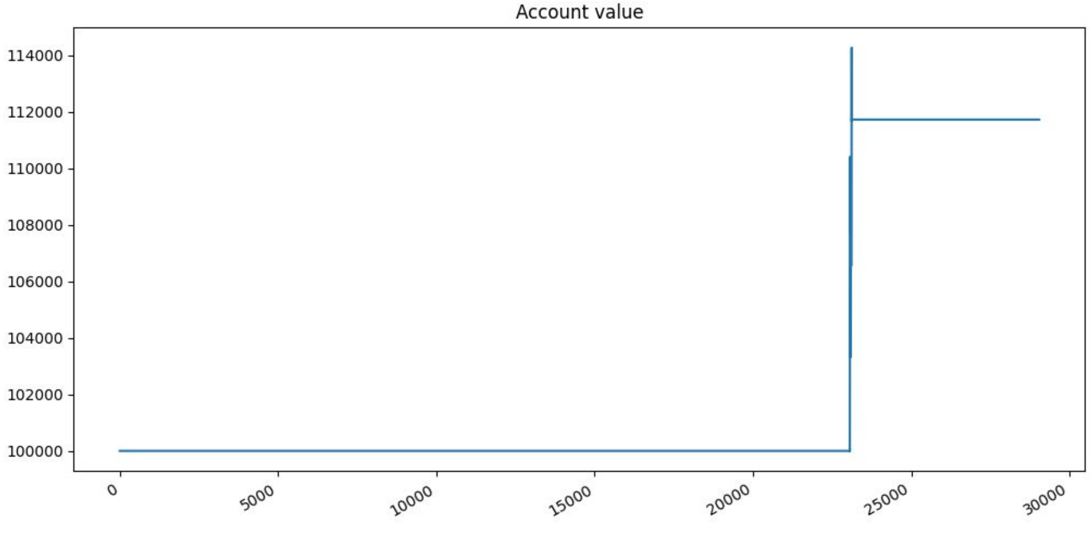
Augmented Dickey-Fuller test

```
ADF Statistic: -1.247097
p-value: 0.653015
```

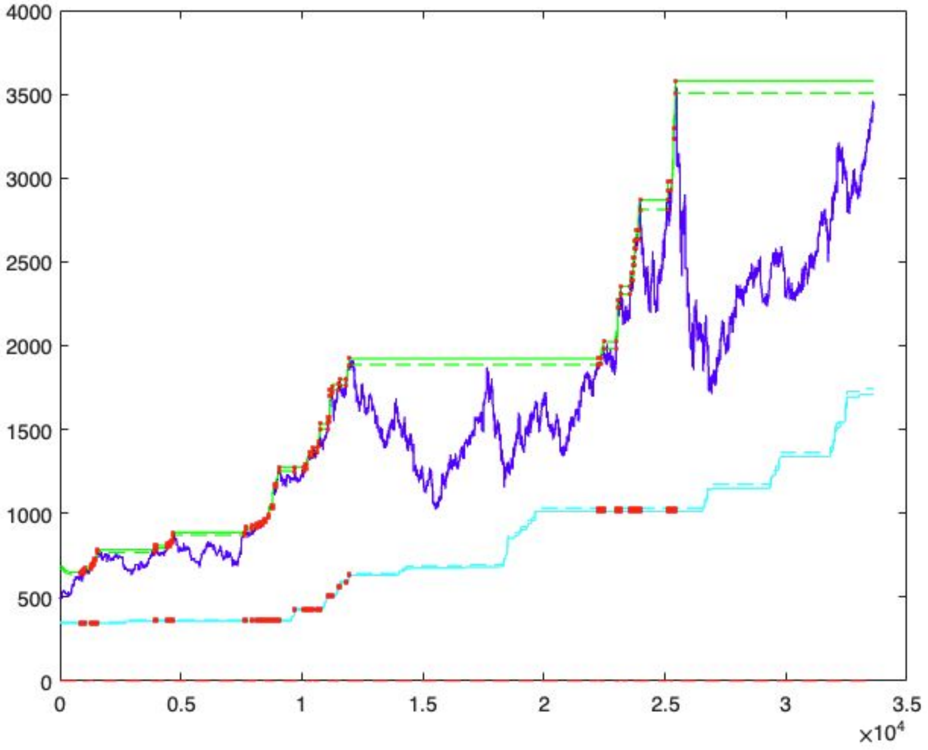
P value is higher than 5% so data is not statinry

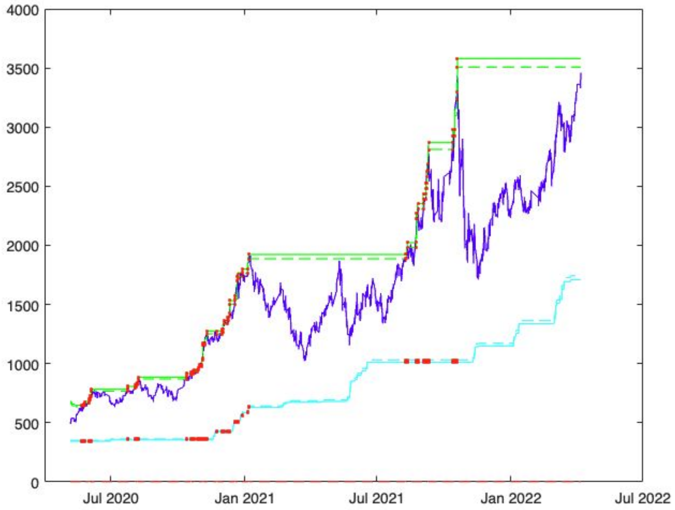
COKE 1 years - 1 month

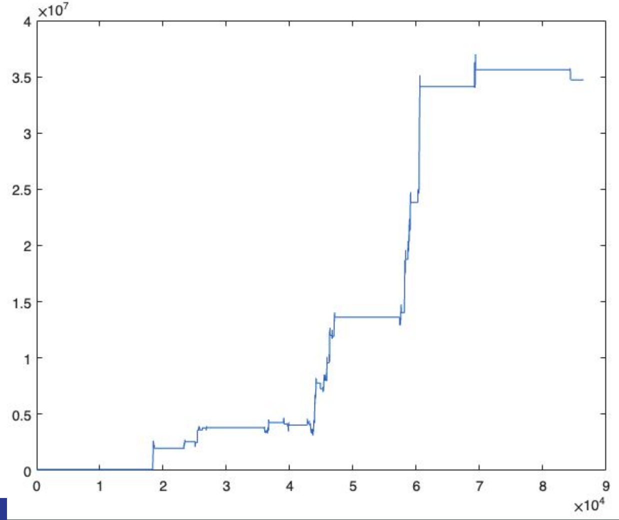
	Value
Average Return	0.40359688694514200
Standard Deviation	67.14553962178040
Sharpe Ratio	0.00601077732368429
Total Trades	5.0
Percent Profitable Trades	80.0
Max Drawdown	-7081.649999999990
Return on Account	15.775991470914300
Average Winner	4030.1750000000000
Average Loser	-4450.325000000000
Profit Factor	3.6223646587608800



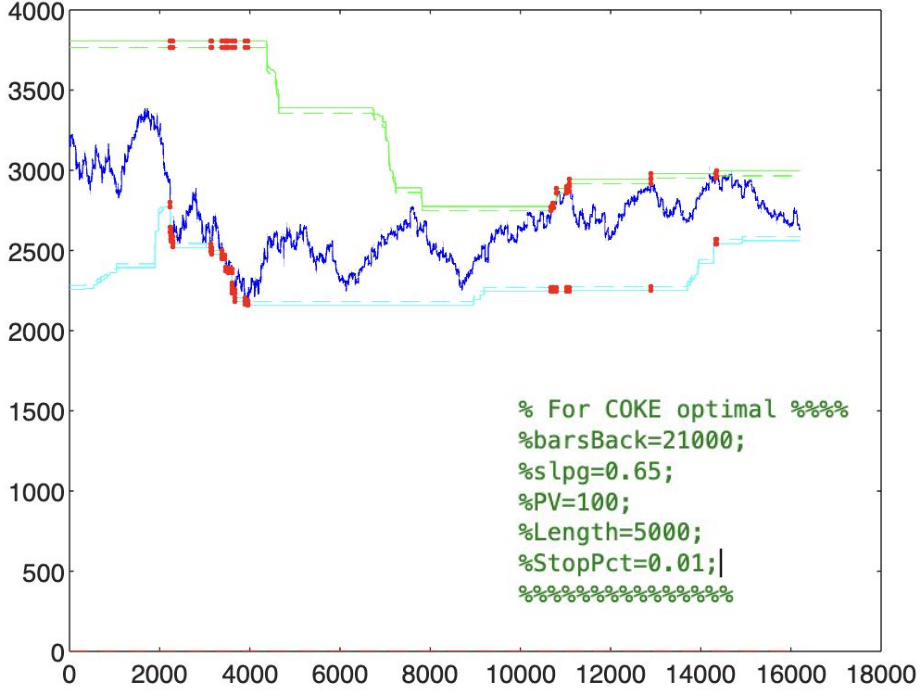
MATLAB: Trade Results for COKE (5 min) (default)

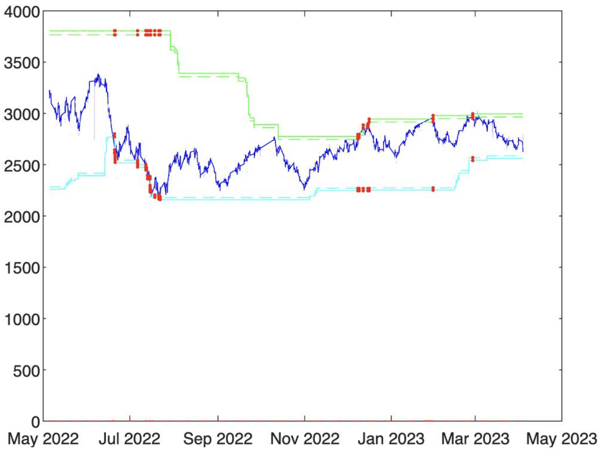


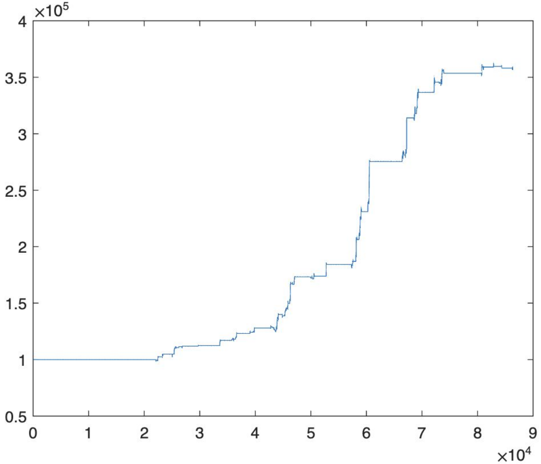




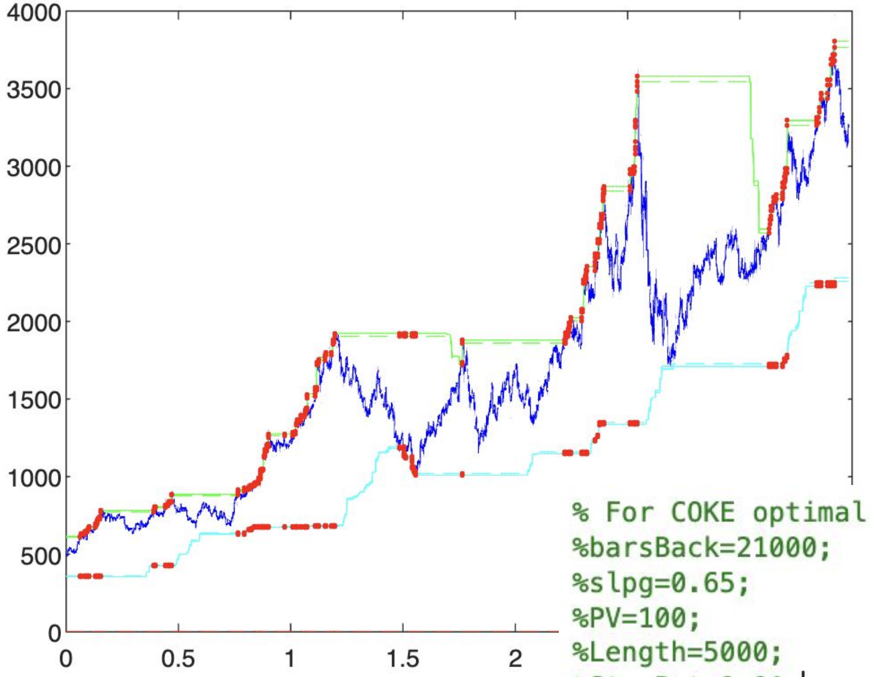
MATLAB Trade Results COKE (5-min)
4 years/ 1 year (optimal parameters)



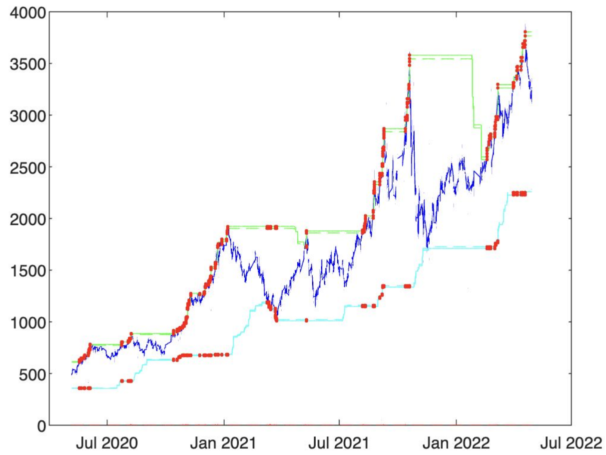


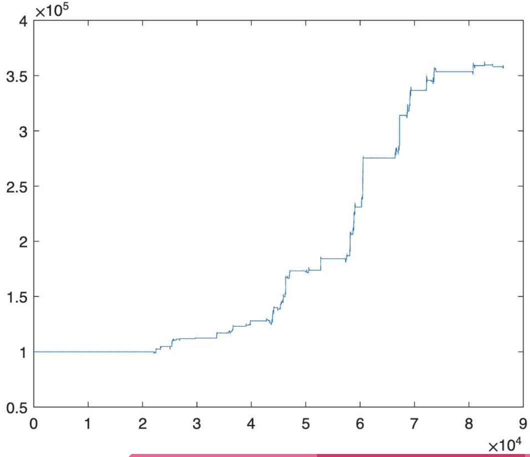


MATLAB Trade Results COKE (5-min) 2 years/ 2 year (optimal parameters)

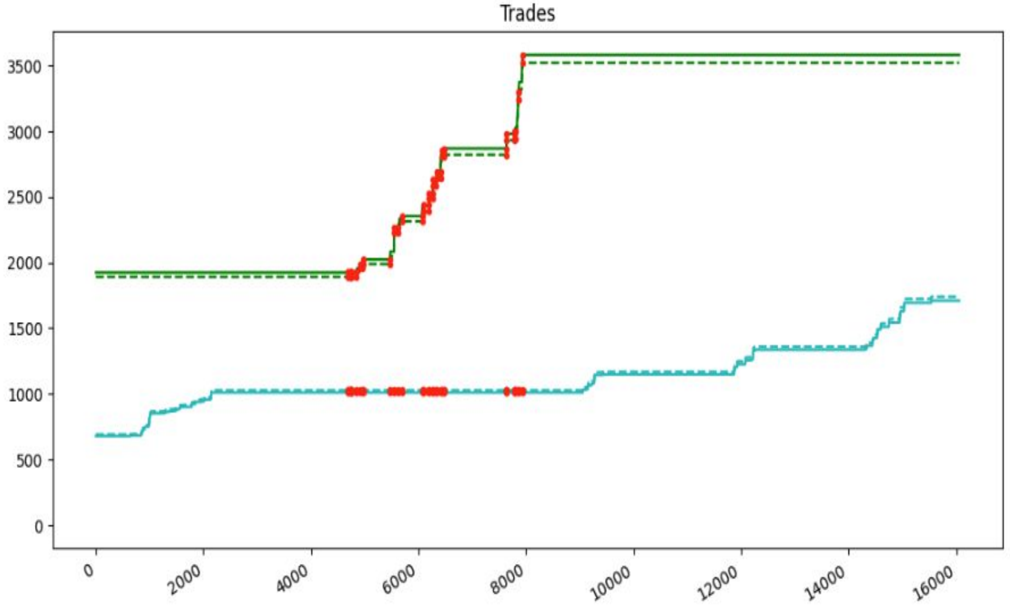


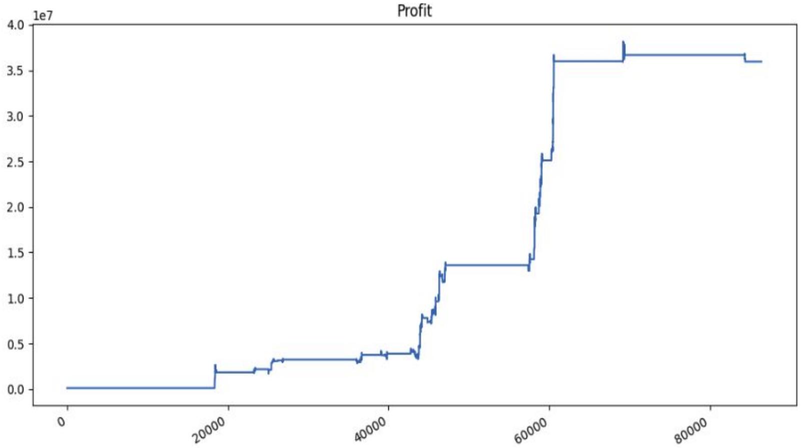

```
% For COKE optimal %%%%
%barsBack=21000;
%slpg=0.65;
%PV=100;
%Length=5000;
%StopPct=0.01;
%%%%%%%%%%%%%%
```

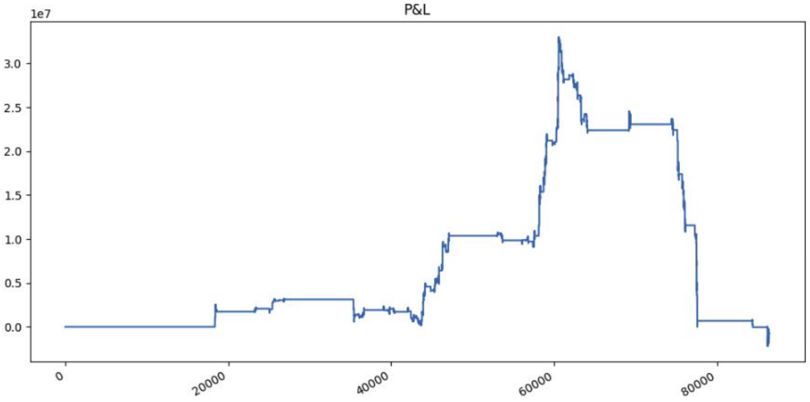




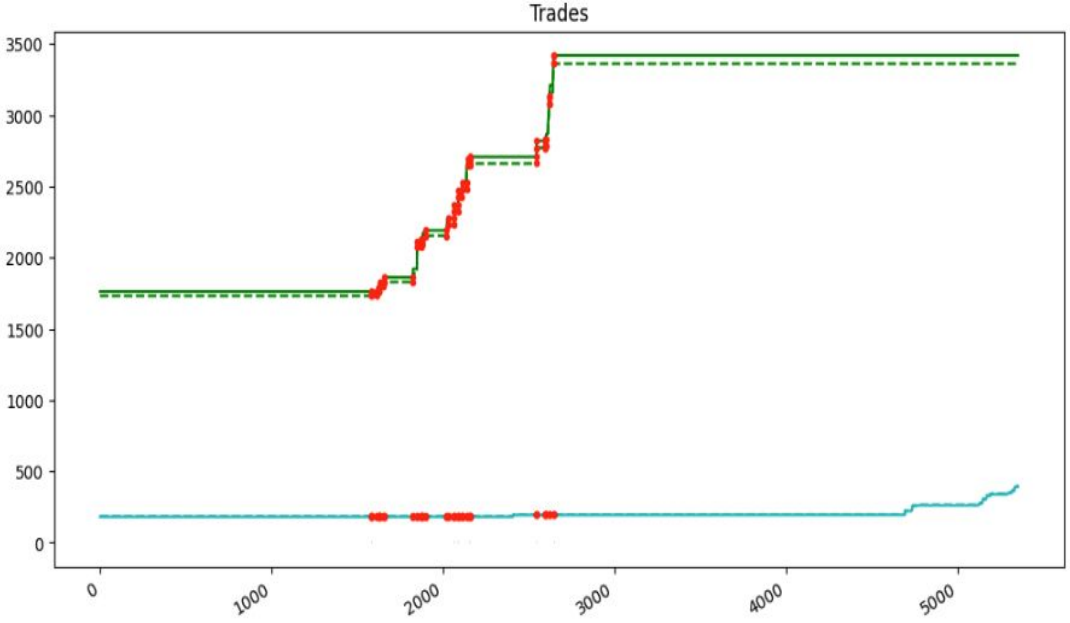
Python: Trade Results for COKE (5 min)

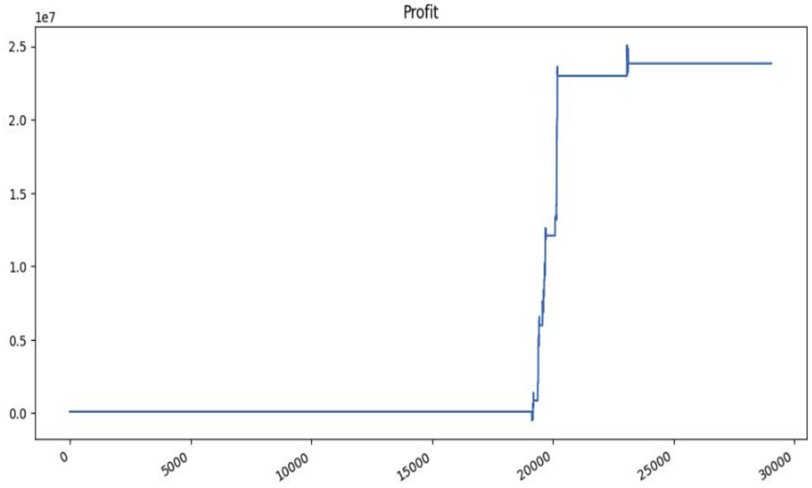


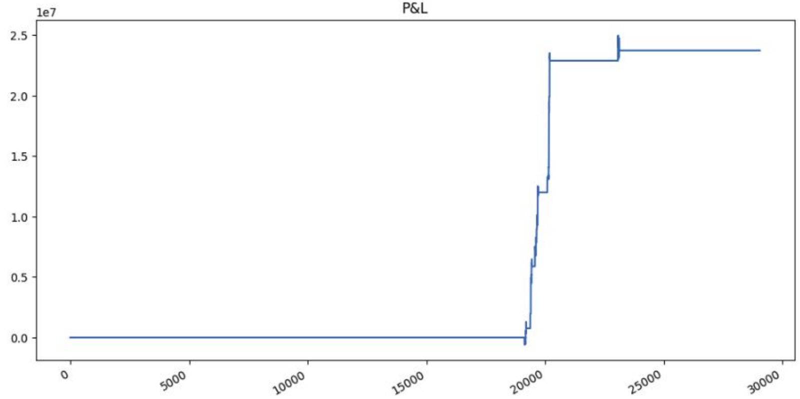




Python: Trade Results for COKE (15 min)







Future Work

- Improve the run time of our algorithm and implementation
- Consider a switch to any of the following infrastructures:
  - Parental C/C++ based
  - Parallelized Python with C under the hood
  - Multiprocessing through threads and locks in Java
- Trying to run the analysis at a higher frequency than ⅕ minute^(-1)
- Instantiating an Automated Data Pipeline for efficient incorporation of the new market data.
- Using a more efficient GridSearch method to learn the optimal parameter values.

[BONUS] The End

Thank You
For Listening!!