It can be theorised that WTI price is equal to some function f(USD, inflation, storage). So let's first perform a rolling regression on these factors from in order to choose the optimal parameters:

In [1]:

```
#Importing all necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
import warnings
warnings.filterwarnings('ignore')
```

WTI 1 month future prices 2010-2019:

In [2]:

```
data_prices_crude         = pd.read_excel('PET_PRI_FUT_S1_D.xls', sheet_name =
data_prices_crude.columns = ['Date', 'F1', 'F2', 'F3', 'F4']
data_prices_crude         = data_prices_crude[(data_prices_crude['Date'] >= '20
data_prices_crude         = data_prices_crude.set_index('Date')
data_prices_crude.drop(columns=['F2','F3','F4'],inplace=True)
data_prices_crude.head(100)
```

Out[2]:

	F1
Date	
2009-08-07	70.93
2009-08-10	70.60
2009-08-11	69.45
2009-08-12	70.16
2009-08-13	70.52
...	...
2009-12-22	74.40
2009-12-23	76.67
2009-12-24	78.05
2009-12-28	78.77
2009-12-29	78.87

100 rows × 1 columns

10 year breakeven inflation rate 2010-2019:

In [3]:

```
inflation_rate         = pd.read_excel('T10YIE.xls', skiprows = 10)
inflation_rate.columns = ['Date','Inflation Rate']
inflation_rate         = inflation_rate[(inflation_rate['Date'] >= '2009-08-12'
inflation_rate         = inflation_rate.set_index('Date')
inflation_rate.head()
```

Out[3]:

	Inflation Rate
Date	
2009-08-12	1.88
2009-08-13	1.79
2009-08-14	1.70
2009-08-17	1.69
2009-08-18	1.73

Nominal USD Index 2010-2019:

In [4]:

```
exchange_rate         = pd.read_excel('DTWEXBGS.xls', skiprows = 10)
exchange_rate.columns = ['Date','USD Index']
exchange_rate         = exchange_rate[(exchange_rate['Date'] >= '2009-08-07') &
exchange_rate         = exchange_rate.set_index('Date')
exchange_rate.head()
```

Out[4]:

	USD Index
Date	
2009-08-07	94.4692
2009-08-10	94.6087
2009-08-11	94.8678
2009-08-12	94.5394
2009-08-13	94.2392

Weekly stock data 2010-2019:

In [5]:

```
data_stocks               = pd.read_excel('PET_STOC_WSTK_DCU_NUS_W.xls', sheet_
data_stocks               = data_stocks[['Date', 'Stocks']]
data_stocks               = data_stocks[(data_stocks['Date'] >= '2009-08-07') &
data_stocks         = data_stocks.set_index('Date')
data_stocks.head(100)
```

Out[5]:

Date	Stocks
2009-08-07	1058086
2009-08-14	1049688
2009-08-21	1049816
2009-08-28	1049443
2009-09-04	1043637
...	...
2011-06-03	1073439
2011-06-10	1070033
2011-06-17	1068311
2011-06-24	1063936
2011-07-01	1063063

100 rows × 1 columns

Merging the data together:

In [6]:

```
df1 = pd.merge(data_prices_crude,inflation_rate,how = 'left',on='Date')
df1 = pd.merge(df1,exchange_rate,how = 'left',on='Date')
df1 = pd.merge(df1,data_stocks,how = 'left',on='Date')
df1.head()
```

Out[6]:

Date	F1	Inflation Rate	USD Index	Stocks
2009-08-07	70.93	NaN	94.4692	1058086.0
2009-08-10	70.60	NaN	94.6087	NaN
2009-08-11	69.45	NaN	94.8678	NaN
2009-08-12	70.16	1.88	94.5394	NaN
2009-08-13	70.52	1.79	94.2392	NaN

Stock data is weekly so we must use interpolation:

In [7]:

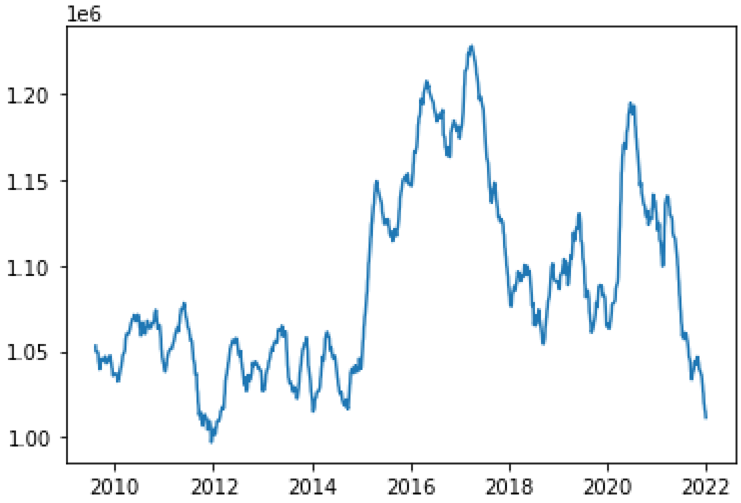
```
df1['Stocks'].interpolate(inplace=True)
df1 = df1.loc['2009-08-12':]
```

In [8]:

```
plt.plot(df1['Stocks'])
```

Out[8]:

```
[<matplotlib.lines.Line2D at 0x12915f640>]
```



In [9]:

```
import pandas_datareader as pdr
import seaborn
import statsmodels.api as sm
from statsmodels.regression.rolling import RollingOLS

seaborn.set_style("darkgrid")
pd.plotting.register_matplotlib_converters()
%matplotlib inline
```

Getting optimal coefficients:

In [10]:

```
w = 60
endog = df1['F1']
exog = sm.add_constant(df1[['Inflation Rate','USD Index','Stocks']])
rols = RollingOLS(endog, exog, window=w)
rres = rols.fit()
params = rres.params.copy()
params.index = np.arange(1, params.shape[0] + 1)
params
```

Out[10]:

	const	Inflation Rate	USD Index	Stocks
1	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN
...	...	...	...	...
3118	-228.150461	20.315538	-0.470935	0.000297
3119	-187.728962	21.927039	-0.507664	0.000258
3120	-150.824993	23.830340	-0.550962	0.000223
3121	-117.391030	26.261025	-0.606247	0.000191
3122	-279.821665	3.975534	-0.088994	0.000345

3122 rows × 4 columns

In [11]:

```
params['Date'] = df1.index
params = params.set_index('Date')
params.tail()
```

Out[11]:

	const	Inflation Rate	USD Index	Stocks
Date				
2021-12-27	-228.150461	20.315538	-0.470935	0.000297
2021-12-28	-187.728962	21.927039	-0.507664	0.000258
2021-12-29	-150.824993	23.830340	-0.550962	0.000223
2021-12-30	-117.391030	26.261025	-0.606247	0.000191
2021-12-31	-279.821665	3.975534	-0.088994	0.000345

Merging with WTI dataframe:

In [12]:

```
df1_reg = pd.merge(df1,params,how = 'left',on='Date')
```

In [13]:

```
df1_regrange = df1_reg.loc['2010-01-04':]
df1_regrange
```

Out[13]:

Date	F1	Inflation Rate_x	USD Index_x	Stocks_x	const	Inflation Rate_y	USD Index_y	Stocks_y
2010-01-04	81.51	2.38	92.3566	1.036539e+06	-223.685957	1.230996	-0.002131	0.000286
2010-01-05	81.77	2.34	92.2236	1.036640e+06	-164.148619	0.685651	0.008764	0.000229
2010-01-06	83.18	2.37	92.0941	1.036740e+06	-109.632478	0.285432	0.017643	0.000177
2010-01-07	82.66	2.41	92.3684	1.036841e+06	-49.771309	-0.418137	0.016535	0.000121
2010-01-08	82.75	2.42	92.1485	1.036941e+06	-20.414187	-0.484475	0.019674	0.000093
...	...	...	...	...	...	...	...	...
2021-12-27	75.57	2.50	115.4964	1.015275e+06	-228.150461	20.315538	-0.470935	0.000297
2021-12-28	75.98	2.50	115.4497	1.014339e+06	-187.728962	21.927039	-0.507664	0.000258
2021-12-29	76.56	2.53	115.3964	1.013404e+06	-150.824993	23.830340	-0.550962	0.000223
2021-12-30	76.99	2.58	115.3163	1.012468e+06	-117.391030	26.261025	-0.606247	0.000191
2021-12-31	75.21	2.56	0.0000	1.011533e+06	-279.821665	3.975534	-0.088994	0.000345

3023 rows × 8 columns

In [14]:

```
df1_regrange['Predicted F1'] = df1_regrange['const'] + (df1_regrange['Inflation
df1_regrange
```

Out[14]:

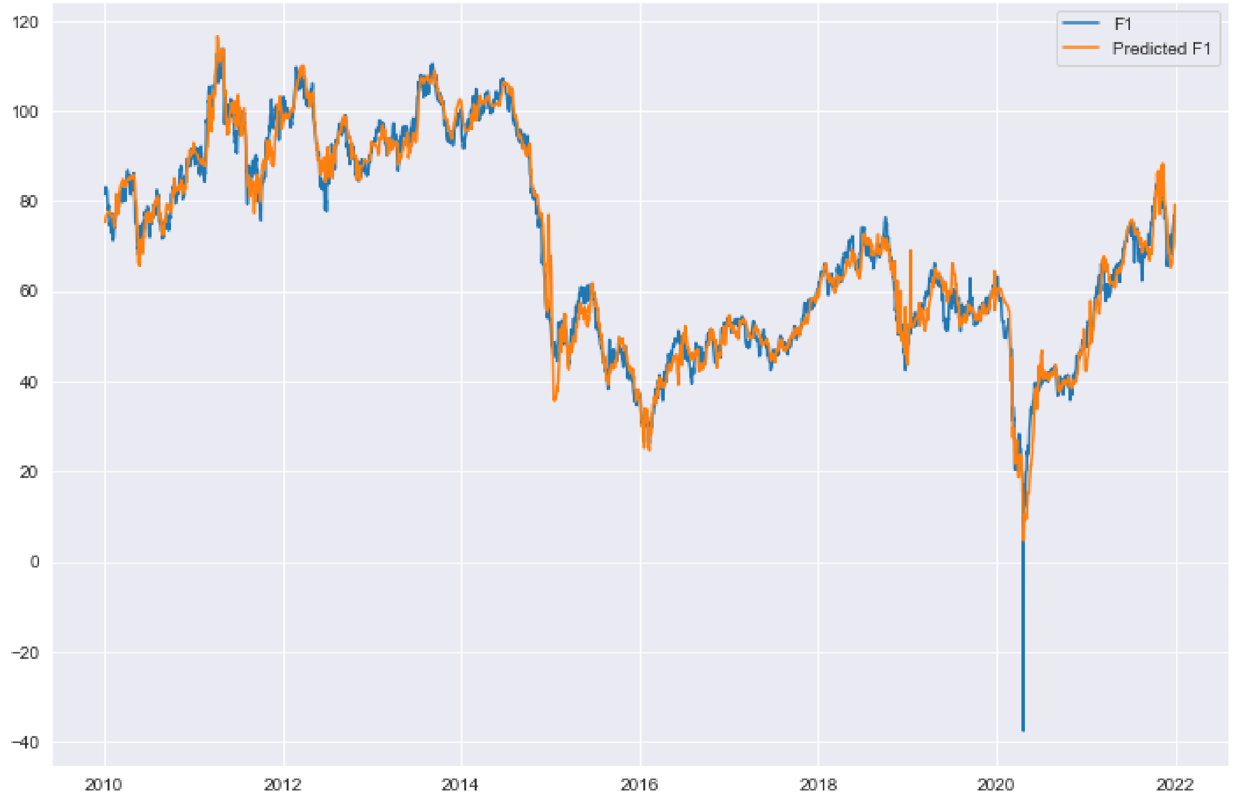
	F1	Inflation Rate_x	USD Index_x	Stocks_x	const	Inflation Rate_y	USD Index_y	Stocks_y
Date								
2010-01-04	81.51	2.38	92.3566	1.036539e+06	-223.685957	1.230996	-0.002131	0.000286
2010-01-05	81.77	2.34	92.2236	1.036640e+06	-164.148619	0.685651	0.008764	0.000229
2010-01-06	83.18	2.37	92.0941	1.036740e+06	-109.632478	0.285432	0.017643	0.000177
2010-01-07	82.66	2.41	92.3684	1.036841e+06	-49.771309	-0.418137	0.016535	0.000121
2010-01-08	82.75	2.42	92.1485	1.036941e+06	-20.414187	-0.484475	0.019674	0.000093
...	...	...	...	...	...	...	...	...
2021-12-27	75.57	2.50	115.4964	1.015275e+06	-228.150461	20.315538	-0.470935	0.000297
2021-12-28	75.98	2.50	115.4497	1.014339e+06	-187.728962	21.927039	-0.507664	0.000258
2021-12-29	76.56	2.53	115.3964	1.013404e+06	-150.824993	23.830340	-0.550962	0.000223
2021-12-30	76.99	2.58	115.3163	1.012468e+06	-117.391030	26.261025	-0.606247	0.000191
2021-12-31	75.21	2.56	0.0000	1.011533e+06	-279.821665	3.975534	-0.088994	0.000345

3023 rows × 9 columns

Plotting F1 against Predicted F1:

In [15]:

```
plt.figure(figsize=(12,8))
plt.plot(df1_regrange['F1'],label='F1')
plt.plot(df1_regrange['Predicted F1'],label='Predicted F1')
plt.legend()
plt.savefig('F1 versus Predicted F1 (window = 60).png')
plt.show()
```



In [16]:
```
df1_regrange['F1 - Predicted'] = df1_regrange['F1'] - df1_regrange['Predicted F
df1_regrange
```

Out[16]:

Date	F1	Inflation Rate_x	USD Index_x	Stocks_x	const	Inflation Rate_y	USD Index_y	Stocks_y
2010-01-04	81.51	2.38	92.3566	1.036539e+06	-223.685957	1.230996	-0.002131	0.000286
2010-01-05	81.77	2.34	92.2236	1.036640e+06	-164.148619	0.685651	0.008764	0.000229
2010-01-06	83.18	2.37	92.0941	1.036740e+06	-109.632478	0.285432	0.017643	0.000177
2010-01-07	82.66	2.41	92.3684	1.036841e+06	-49.771309	-0.418137	0.016535	0.000121
2010-01-08	82.75	2.42	92.1485	1.036941e+06	-20.414187	-0.484475	0.019674	0.000093
...	...	...	...	...	...	...	...	...
2021-12-27	75.57	2.50	115.4964	1.015275e+06	-228.150461	20.315538	-0.470935	0.000297
2021-12-28	75.98	2.50	115.4497	1.014339e+06	-187.728962	21.927039	-0.507664	0.000258
2021-12-29	76.56	2.53	115.3964	1.013404e+06	-150.824993	23.830340	-0.550962	0.000223
2021-12-30	76.99	2.58	115.3163	1.012468e+06	-117.391030	26.261025	-0.606247	0.000191
2021-12-31	75.21	2.56	0.0000	1.011533e+06	-279.821665	3.975534	-0.088994	0.000345

3023 rows × 10 columns

Now let's read in the WTI data with the roll implemented from the first assignment:

In [17]:

```
#Reading in the files
df_wti             = pd.read_excel('HW3input.xlsx', sheet_name = 'WTI')

#transaction cost
t = 0.01
```

In [18]:

```
df_wti = df_wti.set_index('Date')
df_wti = df_wti.loc['2010-01-04':]
df_wti['F1 - Predicted'] = df1_regrange['F1 - Predicted']
df_wti
```

Out[18]:

	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P
Date									
2010-01-04	81.51	82.12	82.65	83.12	0	0	2.15	2.10	
2010-01-05	81.77	82.41	82.99	83.52	0	0	0.26	0.29	
2010-01-06	83.18	83.75	84.31	84.86	0	0	1.41	1.34	
2010-01-07	82.66	83.19	83.75	84.29	0	0	-0.52	-0.56	
2010-01-08	82.75	83.30	83.87	84.47	0	0	0.09	0.11	
...	...	...	...	...	...	...	...	...	
2021-12-27	75.57	75.18	74.67	74.12	0	0	1.78	1.76	
2021-12-28	75.98	75.60	75.13	74.59	0	0	0.41	0.42	
2021-12-29	76.56	76.18	75.71	75.18	0	0	0.58	0.58	
2021-12-30	76.99	76.61	76.13	75.58	0	0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0	0	-1.78	-1.73	

3023 rows × 11 columns

Implementing our strategy and calculating all the necessary values:

In [19]:

```python
df_wti['C(0)']  = 0.0

epsilon = 0

for i in range(len(df_wti)):
    if df_wti['F1 - Predicted'][i] > epsilon:
        df_wti['C(0)'][i] = -1.0
    elif df_wti['F1 - Predicted'][i] < epsilon:
        df_wti['C(0)'][i] = 1.0
```

In [20]:

```python
df_wti['P/L'] = 0.0

for i in range(1,len(df_wti)):
    df_wti['P/L'][i] = df_wti['Cumulative_P/L_barrel'][i]-df_wti['Cumulative_P/
```

In [21]:

```python
df_wti['P/L_short'] = -df_wti['P/L']

# have to reverse the transaction costs for the roll
for i in range(1,len(df_wti)):
    if df_wti['Holding_type'][i] == 1 and df_wti['Holding_type'][i-1] == 0:
        df_wti['P/L_short'][i-1] = df_wti['P/L_short'][i-1] - 0.04
```

In [22]:

```
df_wti.tail()
```

Out[22]:

	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P/
Date									
2021-12-27	75.57	75.18	74.67	74.12	0	0	1.78	1.76	
2021-12-28	75.98	75.60	75.13	74.59	0	0	0.41	0.42	
2021-12-29	76.56	76.18	75.71	75.18	0	0	0.58	0.58	
2021-12-30	76.99	76.61	76.13	75.58	0	0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0	0	-1.78	-1.73	

In [23]:

```
df_wti['P/L_C(0)'] = 0.0
df_wti['Cumulative_P/L_C(0)'] = 0.0

# deciding whether your P&L is from the long or short position
for i in range(1,len(df_wti)):
    if (df_wti['C(0)'][i-1] == 1):
        df_wti['P/L_C(0)'][i] = df_wti['P/L'][i]

    if (df_wti['C(0)'][i-1] == -1):
        df_wti['P/L_C(0)'][i] = df_wti['P/L_short'][i]


# if it is time to switch positions you have to add the transaction costs
for i in range(1,len(df_wti)):
    if df_wti['C(0)'][i] != df_wti['C(0)'][i-1]:
        df_wti['P/L_C(0)'][i] -= 0.02

# calculating cumulative P&L
for i in range(1,len(df_wti)):
    df_wti['Cumulative_P/L_C(0)'][i] = df_wti['P/L_C(0)'][i] + df_wti['Cumulati

#just to see the behaviour when the c(0) sign flips. You can see we account fo
df_wti.tail(25)
```

Out[23]:

	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P
Date									
2021-11-24	78.39	77.81	77.15	76.51	0	0	-0.11	-0.13	
2021-11-29	69.95	69.62	69.28	68.92	0	0	-8.44	-8.19	
2021-11-30	66.18	65.85	65.49	65.15	0	0	-3.77	-3.77	
2021-12-01	65.57	65.37	65.11	64.83	0	0	-0.61	-0.48	
2021-12-02	66.50	66.27	66.03	65.77	0	0	0.93	0.90	
2021-12-03	66.26	66.10	65.93	65.71	0	0	-0.24	-0.17	
2021-12-06	69.49	69.30	69.07	68.80	0	0	3.23	3.20	
2021-12-07	72.05	71.84	71.56	71.26	0	0	2.56	2.54	
2021-12-08	72.36	72.18	71.90	71.64	0	0	0.31	0.34	
2021-12-09	70.94	70.79	70.54	70.29	0	0	-1.42	-1.39	
2021-12-10	71.67	71.48	71.22	70.95	0	0	0.73	0.69	
2021-12-13	71.29	71.06	70.80	70.53	0	0	-0.38	-0.42	
2021-12-14	70.73	70.52	70.26	69.98	0	0	-0.56	-0.54	
2021-12-15	70.87	70.66	70.36	70.05	0	1	0.14	0.14	
2021-12-16	72.38	72.15	71.75	71.35	0	1	1.51	1.49	
2021-12-17	70.86	70.72	70.35	69.95	0	1	-1.52	-1.43	
2021-12-20	68.23	68.61	68.36	68.03	0	1	-2.63	-2.11	
2021-12-21	71.12	70.82	70.44	70.05	1	1	2.89	2.21	
2021-12-22	72.76	72.33	71.83	71.32	0	0	1.64	1.51	
2021-12-23	73.79	73.42	72.93	72.40	0	0	1.03	1.09	

Date	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P
2021-12-27	75.57	75.18	74.67	74.12	0	0	1.78	1.76	
2021-12-28	75.98	75.60	75.13	74.59	0	0	0.41	0.42	
2021-12-29	76.56	76.18	75.71	75.18	0	0	0.58	0.58	
2021-12-30	76.99	76.61	76.13	75.58	0	0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0	0	-1.78	-1.73	

In [24]:

```
df_wti['P(t)_C(0)'] = 0.0

for i in range(1,len(df_wti)):
    df_wti['P(t)_C(0)'][i] = df_wti['P(t)_C(0)'][i-1] +(df_wti['P/L_C(0)'][i]*1

df_wti.tail(2)
```

Out[24]:

Date	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P/
2021-12-30	76.99	76.61	76.13	75.58	0	0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0	0	-1.78	-1.73	

In [25]:

```
df_wti['dF(t)_C(0)'] = df_wti['P/L_C(0)']*100000
```

Calculating performance metrics:

In [26]:

```
N = 12

ann_PL_C = df_wti['P(t)_C(0)'][-1]/N

print(f"C(0) Average Annual P&L: {ann_PL_C}")
```

```
C(0) Average Annual P&L: 414166.66666666535
```

In [27]:

```
ann_SR_C = ann_PL_C/(np.std(df_wti['dF(t)_C(0)'])*np.sqrt(250))

print(f"C(0) Annualised Sharpe Ratio: {ann_SR_C}")
```

```
C(0) Annualised Sharpe Ratio: 0.19147649685959242
```

In [28]:

```
df_wti['HWM_C(0)'] = 0.0
df_wti['DD_C(0)'] = 0.0

for i in range(len(df_wti)):
    df_wti['HWM_C(0)'][i] = np.max(df_wti['P(t)_C(0)'][0:i+1])

    df_wti['DD_C(0)'][i] = df_wti['HWM_C(0)'][i] - df_wti['P(t)_C(0)'][i]
```

In [29]:

```
mdd_C = np.max(df_wti['DD_C(0)'])
mdd_Cdate = df_wti['DD_C(0)'].idxmax()

print(f"C(0) Maximum Drawdown: {mdd_C} at {mdd_Cdate}")
```

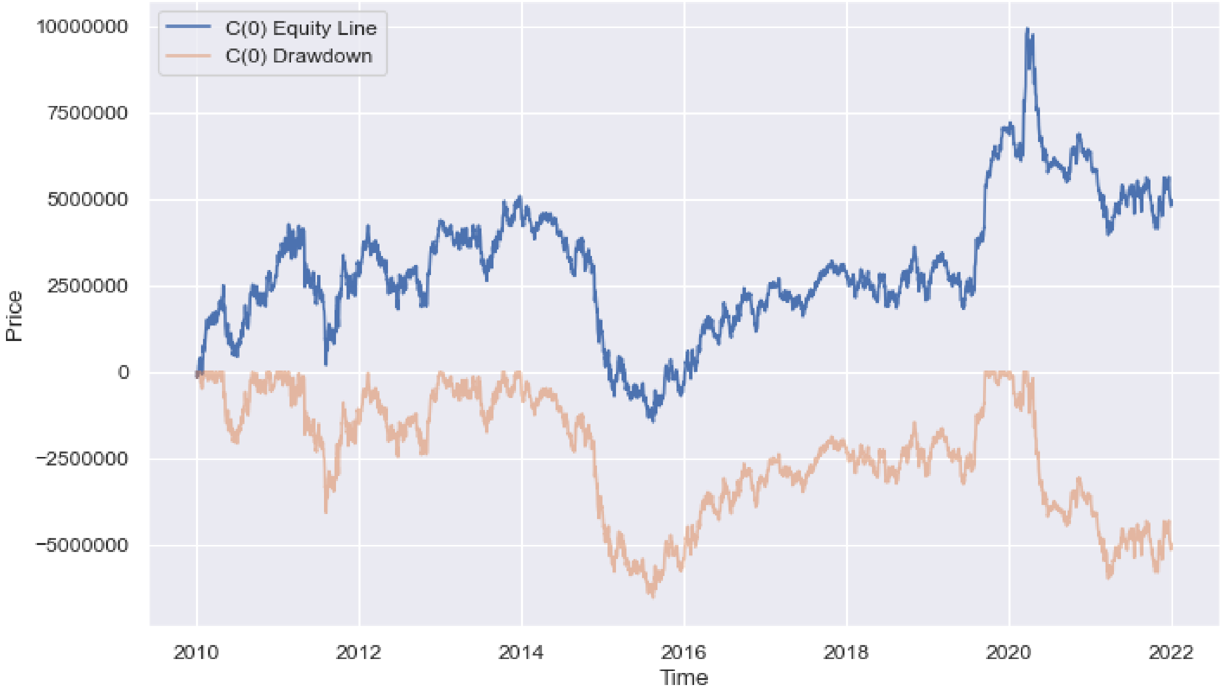
```
C(0) Maximum Drawdown: 6535999.999999996 at 2015-08-18 00:00:00
```

Graphing the equity line:

In [30]:

```
import seaborn as sns
sns.set()

plt.figure(figsize=(10, 6), dpi=80)
plt.ticklabel_format(style='plain')
plt.plot(df_wti.index, df_wti['P(t)_C(0)'], label = 'C(0) Equity Line')
plt.plot(df_wti.index, -df_wti['DD_C(0)'], label = 'C(0) Drawdown',alpha=0.5)
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.savefig('equity line and drawdown not optpitmized')
plt.show()
```



Now let's try to find the optimal parameters for the strategy. First let's find the optimal value for w, which is the rolling window value for the rolling regression calculation:

In [31]:

```
ws = np.arange(11,70,1)

SRoptimal = 0.0
Wopt = 0.0
```

```
epsilon = 0

for w in ws:
    print(w,':')
    print('')
    endog                           = df1['F1']
    exog                            = sm.add_constant(df1[['Inflation Rate','USD
    rols                            = RollingOLS(endog, exog, window=w,min_nobs=
    rres                            = rols.fit()
    params                          = rres.params.copy()
    params.index                    = np.arange(1, params.shape[0] + 1)

    params['Date']                  = df1.index
    params                          = params.set_index('Date')

    df1_reg                         = pd.merge(df1,params,how = 'left',on='Date'

    df1_regrange                    = df1_reg.loc['2010-01-04':]
    df1_regrange['Predicted F1']    = df1_regrange['const'] + (df1_regrange['Inf

    df1_regrange['F1 - Predicted'] = df1_regrange['F1'] - df1_regrange['Predict
    df_wti['F1 - Predicted']        = df1_regrange['F1 - Predicted']

    df_wti['C(0)']  = 0.0
    for i in range(len(df_wti)):
        if df_wti['F1 - Predicted'][i] > epsilon:
            df_wti['C(0)'][i] = -1.0
        elif df_wti['F1 - Predicted'][i] < epsilon:
            df_wti['C(0)'][i] = 1.0

    df_wti['P/L_C(0)'] = 0.0
    df_wti['Cumulative_P/L_C(0)'] = 0.0

    # deciding whether your P&L is from the long or short position
    for i in range(1,len(df_wti)):
        if (df_wti['C(0)'][i-1] == 1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L'][i]

        if (df_wti['C(0)'][i-1] == -1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L_short'][i]


    # if it is time to switch positions you have to add the transaction costs
    for i in range(1,len(df_wti)):
        if df_wti['C(0)'][i] != df_wti['C(0)'][i-1]:
            df_wti['P/L_C(0)'][i] -= 0.02

    # calculating cumulative P&L
    for i in range(1,len(df_wti)):
        df_wti['Cumulative_P/L_C(0)'][i] = df_wti['P/L_C(0)'][i] + df_wti['Cumu

    df_wti['P(t)_C(0)'] = 0.0

    for i in range(1,len(df_wti)):
        df_wti['P(t)_C(0)'][i] = df_wti['P(t)_C(0)'][i-1] +(df_wti['P/L_C(0)'][

    df_wti['dF(t)_C(0)'] = df_wti['P/L_C(0)']*100000

    ann_PL_C = df_wti['P(t)_C(0)'][-1]/N
    print(f"C(0) Average Annual P&L: {ann_PL_C}")
```

```
ann_SR_C = ann_PL_C/(np.std(df_wti['dF(t)_C(0)'])*np.sqrt(250))
print(f"C(0) Annualised Sharpe Ratio: {ann_SR_C}")

df_wti['HWM_C(0)'] = 0.0
df_wti['DD_C(0)'] = 0.0

for i in range(len(df_wti)):
    df_wti['HWM_C(0)'][i] = np.max(df_wti['P(t)_C(0)'][0:i+1])
    df_wti['DD_C(0)'][i] = df_wti['HWM_C(0)'][i] - df_wti['P(t)_C(0)'][i]

mdd_C = np.max(df_wti['DD_C(0)'])
mdd_Cdate = df_wti['DD_C(0)'].idxmax()

print(f"C(0) Maximum Drawdown: {mdd_C} at {mdd_Cdate}")

if ann_SR_C > SRoptimal:
    SRoptimal = ann_SR_C
    Wopt = w
```

```
11 :

C(0) Average Annual P&L: 634833.3333333313
C(0) Annualised Sharpe Ratio: 0.2939753442121571
C(0) Maximum Drawdown: 5416999.999999998 at 2011-05-06 00:00:00
12 :

C(0) Average Annual P&L: 307333.3333333277
C(0) Annualised Sharpe Ratio: 0.14228771172722654
C(0) Maximum Drawdown: 7528000.0 at 2021-12-31 00:00:00
13 :

C(0) Average Annual P&L: 534499.9999999956
C(0) Annualised Sharpe Ratio: 0.24745711623236077
C(0) Maximum Drawdown: 7202000.000000025 at 2021-08-09 00:00:00
14 :

C(0) Average Annual P&L: 54333.333333328366
C(0) Annualised Sharpe Ratio: 0.025147224831182095
C(0) Maximum Drawdown: 7869000.000000026 at 2021-12-31 00:00:00
15 :

C(0) Average Annual P&L: 74499.9999999952
C(0) Annualised Sharpe Ratio: 0.03448356671142607
C(0) Maximum Drawdown: 7736000.0 at 2021-12-20 00:00:00
16 :

C(0) Average Annual P&L: 258666.6666666638
C(0) Annualised Sharpe Ratio: 0.11973683688930364
C(0) Maximum Drawdown: 6170000.0000000205 at 2021-05-28 00:00:00
17 :

C(0) Average Annual P&L: 407999.999999999
C(0) Annualised Sharpe Ratio: 0.18885699054244973
C(0) Maximum Drawdown: 6625999.999999996 at 2021-05-13 00:00:00
18 :

C(0) Average Annual P&L: 315499.99999999785
C(0) Annualised Sharpe Ratio: 0.1460253333335852
C(0) Maximum Drawdown: 6144000.000000006 at 2021-05-13 00:00:00
19 :

C(0) Average Annual P&L: 279166.66666666436
C(0) Annualised Sharpe Ratio: 0.12920661579011092
C(0) Maximum Drawdown: 6656999.999999997 at 2021-08-25 00:00:00
20 :

C(0) Average Annual P&L: -84166.66666666932
C(0) Annualised Sharpe Ratio: -0.03894958166136588
C(0) Maximum Drawdown: 6737000.000000006 at 2021-08-26 00:00:00
21 :

C(0) Average Annual P&L: -131833.33333333678
C(0) Annualised Sharpe Ratio: -0.061006150519148906
C(0) Maximum Drawdown: 6624999.999999995 at 2021-11-01 00:00:00
22 :

C(0) Average Annual P&L: 32499.9999999974
C(0) Annualised Sharpe Ratio: 0.015039716512720427
C(0) Maximum Drawdown: 6224000.000000001 at 2021-03-23 00:00:00
```

```
23 :

C(0) Average Annual P&L: -331833.3333333332
C(0) Annualised Sharpe Ratio: -0.15354980498768445
C(0) Maximum Drawdown: 5990999.99999999 at 2021-11-01 00:00:00
24 :

C(0) Average Annual P&L: -188000.00000000023
C(0) Annualised Sharpe Ratio: -0.08699415440683392
C(0) Maximum Drawdown: 6227000.000000001 at 2014-08-29 00:00:00
25 :

C(0) Average Annual P&L: -378499.9999999995
C(0) Annualised Sharpe Ratio: -0.17513165726588176
C(0) Maximum Drawdown: 5800999.999999993 at 2021-10-20 00:00:00
26 :

C(0) Average Annual P&L: -158500.00000000003
C(0) Annualised Sharpe Ratio: -0.0733368330933691
C(0) Maximum Drawdown: 7145999.999999992 at 2020-05-21 00:00:00
27 :

C(0) Average Annual P&L: -197666.6666666659
C(0) Annualised Sharpe Ratio: -0.09145170783811062
C(0) Maximum Drawdown: 6284999.999999992 at 2014-08-29 00:00:00
28 :

C(0) Average Annual P&L: 195666.6666666672
C(0) Annualised Sharpe Ratio: 0.09052581417471052
C(0) Maximum Drawdown: 5166999.999999995 at 2014-08-29 00:00:00
29 :

C(0) Average Annual P&L: 102166.6666666669
C(0) Annualised Sharpe Ratio: 0.04726008091622853
C(0) Maximum Drawdown: 5438999.999999992 at 2015-07-08 00:00:00
30 :

C(0) Average Annual P&L: 334999.99999999773
C(0) Annualised Sharpe Ratio: 0.15498435016344803
C(0) Maximum Drawdown: 3972000.0000000126 at 2020-07-01 00:00:00
31 :

C(0) Average Annual P&L: -63333.33333333387
C(0) Annualised Sharpe Ratio: -0.029295758090015393
C(0) Maximum Drawdown: 6631999.999999989 at 2015-07-08 00:00:00
32 :

C(0) Average Annual P&L: -201166.6666666655
C(0) Annualised Sharpe Ratio: -0.09304669857358594
C(0) Maximum Drawdown: 7015999.999999983 at 2015-07-08 00:00:00
33 :

C(0) Average Annual P&L: -337166.66666666354
C(0) Annualised Sharpe Ratio: -0.15595388666517532
C(0) Maximum Drawdown: 7800999.999999994 at 2019-06-07 00:00:00
34 :

C(0) Average Annual P&L: -273499.99999999726
C(0) Annualised Sharpe Ratio: -0.1264986202561386
C(0) Maximum Drawdown: 7678999.999999988 at 2019-06-07 00:00:00
```

```
35 :

C(0) Average Annual P&L: -23666.666666664125
C(0) Annualised Sharpe Ratio: -0.010946038222264738
C(0) Maximum Drawdown: 6761999.999999977 at 2015-07-15 00:00:00
36 :

C(0) Average Annual P&L: 64666.66666666954
C(0) Annualised Sharpe Ratio: 0.029908383263308354
C(0) Maximum Drawdown: 6793999.999999989 at 2015-07-15 00:00:00
37 :

C(0) Average Annual P&L: 269000.00000000064
C(0) Annualised Sharpe Ratio: 0.12442585296430632
C(0) Maximum Drawdown: 6063999.999999992 at 2015-07-17 00:00:00
38 :

C(0) Average Annual P&L: 288833.3333333354
C(0) Annualised Sharpe Ratio: 0.13359679351013523
C(0) Maximum Drawdown: 6101999.999999992 at 2015-07-17 00:00:00
39 :

C(0) Average Annual P&L: 346000.000000001
C(0) Annualised Sharpe Ratio: 0.16003790473928725
C(0) Maximum Drawdown: 5563999.99999999 at 2015-07-17 00:00:00
40 :

C(0) Average Annual P&L: 379833.333333334
C(0) Annualised Sharpe Ratio: 0.1756854821535032
C(0) Maximum Drawdown: 5469999.99999999 at 2015-07-17 00:00:00
41 :

C(0) Average Annual P&L: 432999.9999999996
C(0) Annualised Sharpe Ratio: 0.2002744335899954
C(0) Maximum Drawdown: 4854999.999999993 at 2015-07-21 00:00:00
42 :

C(0) Average Annual P&L: 447833.3333333328
C(0) Annualised Sharpe Ratio: 0.20713052888046019
C(0) Maximum Drawdown: 5127999.999999987 at 2015-07-22 00:00:00
43 :

C(0) Average Annual P&L: 586166.6666666671
C(0) Annualised Sharpe Ratio: 0.27112536607390464
C(0) Maximum Drawdown: 5006999.999999986 at 2015-07-22 00:00:00
44 :

C(0) Average Annual P&L: 551499.9999999998
C(0) Annualised Sharpe Ratio: 0.25507654232063903
C(0) Maximum Drawdown: 5442999.999999985 at 2015-07-29 00:00:00
45 :

C(0) Average Annual P&L: 416333.3333333331
C(0) Annualised Sharpe Ratio: 0.19253644674845802
C(0) Maximum Drawdown: 5881000.0000000335 at 2021-11-01 00:00:00
46 :

C(0) Average Annual P&L: 482166.6666666663
C(0) Annualised Sharpe Ratio: 0.2229812728554925
C(0) Maximum Drawdown: 6285000.000000042 at 2021-11-01 00:00:00
```

```
47 :

C(0) Average Annual P&L: 619833.3333333322
C(0) Annualised Sharpe Ratio: 0.28666865318748774
C(0) Maximum Drawdown: 6571000.000000042 at 2021-11-01 00:00:00
48 :

C(0) Average Annual P&L: 573499.9999999992
C(0) Annualised Sharpe Ratio: 0.2652215701008766
C(0) Maximum Drawdown: 6971000.000000034 at 2021-11-01 00:00:00
49 :

C(0) Average Annual P&L: 491666.6666666649
C(0) Annualised Sharpe Ratio: 0.227351566581645
C(0) Maximum Drawdown: 6941000.000000034 at 2021-11-01 00:00:00
50 :

C(0) Average Annual P&L: 909333.3333333308
C(0) Annualised Sharpe Ratio: 0.4206029259602411
C(0) Maximum Drawdown: 5717000.000000048 at 2021-11-01 00:00:00
51 :

C(0) Average Annual P&L: 857833.3333333299
C(0) Annualised Sharpe Ratio: 0.39677636875247685
C(0) Maximum Drawdown: 5874000.00000005 at 2021-03-23 00:00:00
52 :

C(0) Average Annual P&L: 837833.3333333284
C(0) Annualised Sharpe Ratio: 0.387500492360305
C(0) Maximum Drawdown: 6140000.00000005 at 2021-03-23 00:00:00
53 :

C(0) Average Annual P&L: 847666.6666666627
C(0) Annualised Sharpe Ratio: 0.39204909639759267
C(0) Maximum Drawdown: 5322000.00000005 at 2021-03-23 00:00:00
54 :

C(0) Average Annual P&L: 636666.6666666645
C(0) Annualised Sharpe Ratio: 0.294412156667524
C(0) Maximum Drawdown: 6148000.000000043 at 2021-03-23 00:00:00
55 :

C(0) Average Annual P&L: 472833.33333333215
C(0) Annualised Sharpe Ratio: 0.21862151217146758
C(0) Maximum Drawdown: 6732000.000000043 at 2021-03-23 00:00:00
56 :

C(0) Average Annual P&L: 490666.666666666
C(0) Annualised Sharpe Ratio: 0.2268714678166569
C(0) Maximum Drawdown: 6298000.000000039 at 2021-03-23 00:00:00
57 :

C(0) Average Annual P&L: 588999.9999999988
C(0) Annualised Sharpe Ratio: 0.27235026787358746
C(0) Maximum Drawdown: 6605000.000000034 at 2021-11-01 00:00:00
58 :

C(0) Average Annual P&L: 470166.66666666564
C(0) Annualised Sharpe Ratio: 0.21737146189252107
C(0) Maximum Drawdown: 6165000.00000004 at 2021-11-01 00:00:00
```

```
59 :

C(0) Average Annual P&L: 550333.3333333324
C(0) Annualised Sharpe Ratio: 0.2544499412130605
C(0) Maximum Drawdown: 6093999.999999996 at 2015-08-18 00:00:00
60 :

C(0) Average Annual P&L: 414166.66666666535
C(0) Annualised Sharpe Ratio: 0.19147649685959242
C(0) Maximum Drawdown: 6535999.999999996 at 2015-08-18 00:00:00
61 :

C(0) Average Annual P&L: 401333.3333333317
C(0) Annualised Sharpe Ratio: 0.1855388661743373
C(0) Maximum Drawdown: 6497999.999999997 at 2015-08-18 00:00:00
62 :

C(0) Average Annual P&L: 581833.3333333301
C(0) Annualised Sharpe Ratio: 0.269013107392562
C(0) Maximum Drawdown: 6383999.999999994 at 2015-08-18 00:00:00
63 :

C(0) Average Annual P&L: 517999.9999999991
C(0) Annualised Sharpe Ratio: 0.239489604548107
C(0) Maximum Drawdown: 6541999.9999999935 at 2015-08-18 00:00:00
64 :

C(0) Average Annual P&L: 300499.99999999854
C(0) Annualised Sharpe Ratio: 0.13891251897318949
C(0) Maximum Drawdown: 7831000.000000032 at 2021-12-30 00:00:00
65 :

C(0) Average Annual P&L: 78833.3333333315
C(0) Annualised Sharpe Ratio: 0.03643848826958082
C(0) Maximum Drawdown: 7885000.00000003 at 2021-12-30 00:00:00
66 :

C(0) Average Annual P&L: 71666.66666666404
C(0) Annualised Sharpe Ratio: 0.033125058165686554
C(0) Maximum Drawdown: 7662000.0000000335 at 2021-12-01 00:00:00
67 :

C(0) Average Annual P&L: 17499.99999999774
C(0) Annualised Sharpe Ratio: 0.008088581956464789
C(0) Maximum Drawdown: 8041000.0000000335 at 2021-12-01 00:00:00
68 :

C(0) Average Annual P&L: -44333.33333333736
C(0) Annualised Sharpe Ratio: -0.020490591194503912
C(0) Maximum Drawdown: 9275000.000000047 at 2021-12-01 00:00:00
69 :

C(0) Average Annual P&L: 162999.99999999753
C(0) Annualised Sharpe Ratio: 0.07534568330748707
C(0) Maximum Drawdown: 7845000.0000000335 at 2021-12-01 00:00:00
```

In [32]:

```
print(SRoptimal)
print(Wopt)
```

```
0.4206029259602411
50
```

Thus the optimal rolling regression parameter is w=50, giving a Sharpe Ratio of 0.42 (epsilon=0)

Run these cells again:

In [33]:
```
w = 50
endog = df1['F1']
exog = sm.add_constant(df1[['Inflation Rate','USD Index','Stocks']])
rols = RollingOLS(endog, exog, window=w)
rres = rols.fit()
params = rres.params.copy()
params.index = np.arange(1, params.shape[0] + 1)
```

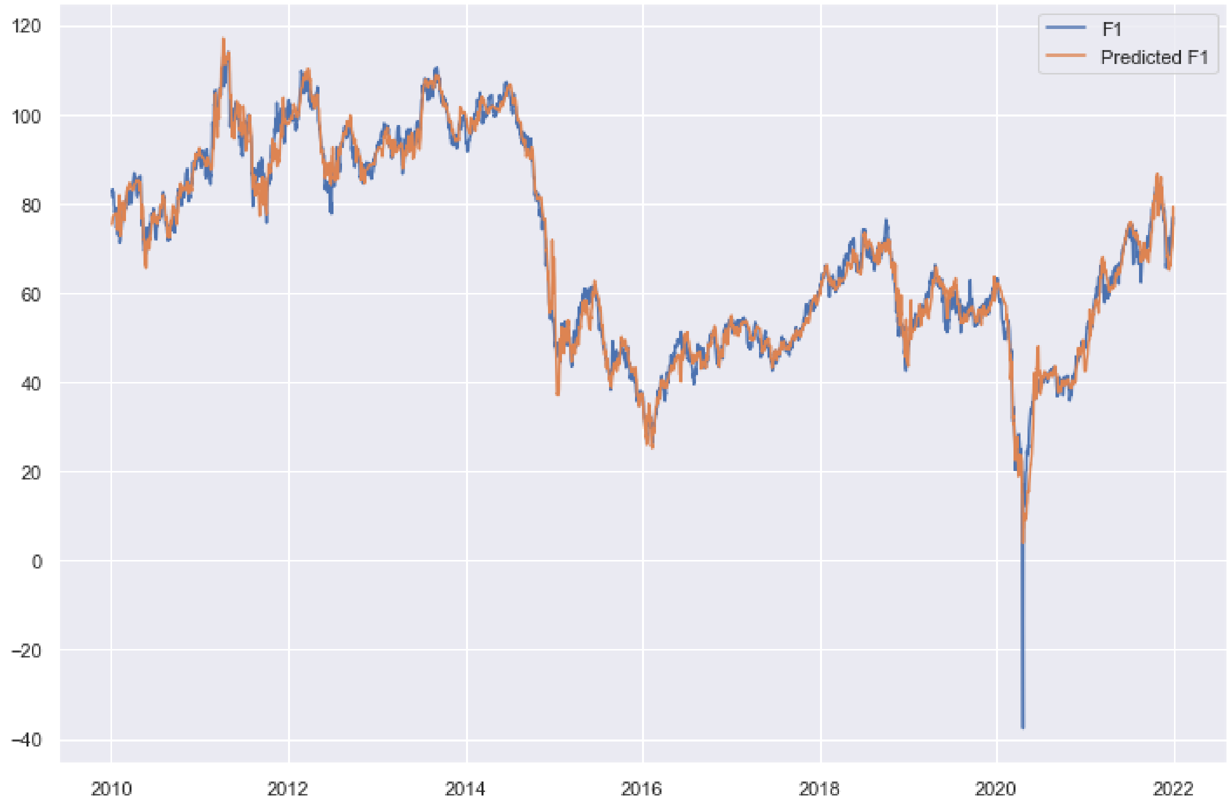
In [34]:
```
params['Date'] = df1.index
params = params.set_index('Date')
```

In [35]:
```
df1_reg = pd.merge(df1,params,how = 'left',on='Date')
```

In [36]:
```
df1_regrange = df1_reg.loc['2010-01-04':]
```

In [37]:
```
df1_regrange['Predicted F1'] = df1_regrange['const'] + (df1_regrange['Inflation
```

In [38]:
```
plt.figure(figsize=(12,8))
plt.plot(df1_regrange['F1'],label='F1')
plt.plot(df1_regrange['Predicted F1'],label='Predicted F1')
plt.legend()
plt.savefig('F1 versus Predicted F1 (opt window=50)')
plt.show()
```



In [39]:
```
df1_regrange['F1 - Predicted'] = df1_regrange['F1'] - df1_regrange['Predicted F
```

In [40]:
```
#Reading in the files
df_wti           = pd.read_excel('HW3input.xlsx', sheet_name = 'WTI')

#transaction cost
t = 0.01
```

In [41]:
```
df_wti = df_wti.set_index('Date')
df_wti = df_wti.loc['2010-01-04':]
df_wti['F1 - Predicted'] = df1_regrange['F1 - Predicted']
```

In [42]:
```
df_wti['P/L'] = 0.0

for i in range(1,len(df_wti)):
    df_wti['P/L'][i] = df_wti['Cumulative_P/L_barrel'][i]-df_wti['Cumulative_P/
```

In [43]:
```
df_wti['P/L_short'] = -df_wti['P/L']

# have to reverse the transaction costs for the roll
for i in range(1,len(df_wti)):
    if df_wti['Holding_type'][i] == 1 and df_wti['Holding_type'][i-1] == 0:
        df_wti['P/L_short'][i-1] = df_wti['P/L_short'][i-1] - 0.04
```

Now let's optimise epsilon, which is the parameter used in our strategy function:

In [44]:
```
epsilons = np.arange(0.0,1.01,0.01)
SRoptimal = 0.0
Eopt = 0.0
```

```
for epsilon in epsilons:
    print(epsilon,":")
    print("")

    df_wti['C(0)']  = 0.0
    for i in range(len(df_wti)):
        if df_wti['F1 - Predicted'][i] > epsilon:
            df_wti['C(0)'][i] = -1.0
        elif df_wti['F1 - Predicted'][i] < epsilon:
            df_wti['C(0)'][i] = 1.0

    df_wti['P/L_C(0)'] = 0.0
    df_wti['Cumulative_P/L_C(0)'] = 0.0

    # deciding whether your P&L is from the long or short position
    for i in range(1,len(df_wti)):
        if (df_wti['C(0)'][i-1] == 1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L'][i]

        if (df_wti['C(0)'][i-1] == -1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L_short'][i]


    # if it is time to switch positions you have to add the transaction costs
    for i in range(1,len(df_wti)):
        if df_wti['C(0)'][i] != df_wti['C(0)'][i-1]:
            df_wti['P/L_C(0)'][i] -= 0.02

    # calculating cumulative P&L
    for i in range(1,len(df_wti)):
        df_wti['Cumulative_P/L_C(0)'][i] = df_wti['P/L_C(0)'][i] + df_wti['Cumu

    df_wti['P(t)_C(0)'] = 0.0

    for i in range(1,len(df_wti)):
        df_wti['P(t)_C(0)'][i] = df_wti['P(t)_C(0)'][i-1] +(df_wti['P/L_C(0)'][

    df_wti['dF(t)_C(0)'] = df_wti['P/L_C(0)']*100000

    ann_PL_C = df_wti['P(t)_C(0)'][-1]/N
    print(f"C({epsilon}) Average Annual P&L: {ann_PL_C}")

    ann_SR_C = ann_PL_C/(np.std(df_wti['dF(t)_C(0)'])*np.sqrt(250))
    print(f"C({epsilon}) Annualised Sharpe Ratio: {ann_SR_C}")

    df_wti['HWM_C(0)'] = 0.0
    df_wti['DD_C(0)'] = 0.0

    for i in range(len(df_wti)):
        df_wti['HWM_C(0)'][i] = np.max(df_wti['P(t)_C(0)'][0:i+1])
        df_wti['DD_C(0)'][i] = df_wti['HWM_C(0)'][i] - df_wti['P(t)_C(0)'][i]

    mdd_C = np.max(df_wti['DD_C(0)'])
    mdd_Cdate = df_wti['DD_C(0)'].idxmax()

    print(f"C({epsilon}) Maximum Drawdown: {mdd_C} at {mdd_Cdate}")
    print("")

    if ann_SR_C > SRoptimal:
        SRoptimal = ann_SR_C
        Eopt = epsilon
```

```
0.0 :

C(0.0) Average Annual P&L: 909333.3333333308
C(0.0) Annualised Sharpe Ratio: 0.4206029259602411
C(0.0) Maximum Drawdown: 5717000.000000048 at 2021-11-01 00:00:00

0.01 :

C(0.01) Average Annual P&L: 865999.9999999981
C(0.01) Annualised Sharpe Ratio: 0.4005405939426985
C(0.01) Maximum Drawdown: 5951000.000000048 at 2021-11-01 00:00:00

0.02 :

C(0.02) Average Annual P&L: 771333.3333333312
C(0.02) Annualised Sharpe Ratio: 0.35671496934899705
C(0.02) Maximum Drawdown: 5743000.000000048 at 2021-11-01 00:00:00

0.03 :

C(0.03) Average Annual P&L: 700499.9999999987
C(0.03) Annualised Sharpe Ratio: 0.3239370141011185
C(0.03) Maximum Drawdown: 6149000.000000046 at 2021-11-01 00:00:00

0.04 :

C(0.04) Average Annual P&L: 564833.3333333329
C(0.04) Annualised Sharpe Ratio: 0.2611731235324725
C(0.04) Maximum Drawdown: 6149000.000000044 at 2021-11-01 00:00:00

0.05 :

C(0.05) Average Annual P&L: 554499.9999999999
C(0.05) Annualised Sharpe Ratio: 0.25639365720997676
C(0.05) Maximum Drawdown: 5835000.00000004 at 2021-11-01 00:00:00

0.06 :

C(0.06) Average Annual P&L: 491833.33333333296
C(0.06) Annualised Sharpe Ratio: 0.22741148826151075
C(0.06) Maximum Drawdown: 5835000.000000038 at 2021-11-01 00:00:00

0.07 :

C(0.07) Average Annual P&L: 565833.3333333333
C(0.07) Annualised Sharpe Ratio: 0.2616442150461933
C(0.07) Maximum Drawdown: 5835000.000000036 at 2021-11-01 00:00:00

0.08 :

C(0.08) Average Annual P&L: 566833.3333333336
C(0.08) Annualised Sharpe Ratio: 0.2621093870149064
C(0.08) Maximum Drawdown: 5835000.000000036 at 2021-11-01 00:00:00

0.09 :

C(0.09) Average Annual P&L: 549333.3333333336
C(0.09) Annualised Sharpe Ratio: 0.2540187162271203
C(0.09) Maximum Drawdown: 5685000.000000044 at 2021-11-01 00:00:00
```

```
0.1 :

C(0.1) Average Annual P&L: 565833.3333333338
C(0.1) Annualised Sharpe Ratio: 0.2616551066489101
C(0.1) Maximum Drawdown: 5417000.000000044 at 2021-11-01 00:00:00

0.11 :

C(0.11) Average Annual P&L: 535500.0000000001
C(0.11) Annualised Sharpe Ratio: 0.24762355090677732
C(0.11) Maximum Drawdown: 5185999.999999985 at 2015-08-31 00:00:00

0.12 :

C(0.12) Average Annual P&L: 497166.6666666672
C(0.12) Annualised Sharpe Ratio: 0.22989038038991239
C(0.12) Maximum Drawdown: 5185999.999999984 at 2015-08-31 00:00:00

0.13 :

C(0.13) Average Annual P&L: 420000.000000001
C(0.13) Annualised Sharpe Ratio: 0.19419717239437836
C(0.13) Maximum Drawdown: 5449999.999999986 at 2015-08-31 00:00:00

0.14 :

C(0.14) Average Annual P&L: 444833.333333334
C(0.14) Annualised Sharpe Ratio: 0.2056832458080909
C(0.14) Maximum Drawdown: 5593999.999999984 at 2015-08-31 00:00:00

0.15 :

C(0.15) Average Annual P&L: 464833.3333333337
C(0.15) Annualised Sharpe Ratio: 0.21493358055537493
C(0.15) Maximum Drawdown: 5593999.999999984 at 2015-08-31 00:00:00

0.16 :

C(0.16) Average Annual P&L: 446666.6666666674
C(0.16) Annualised Sharpe Ratio: 0.20653366042947302
C(0.16) Maximum Drawdown: 5545999.999999985 at 2015-08-31 00:00:00

0.17 :

C(0.17) Average Annual P&L: 425333.333333334
C(0.17) Annualised Sharpe Ratio: 0.1966697797769936
C(0.17) Maximum Drawdown: 5461999.999999987 at 2015-08-31 00:00:00

0.18 :

C(0.18) Average Annual P&L: 394833.3333333339
C(0.18) Annualised Sharpe Ratio: 0.18256521406494358
C(0.18) Maximum Drawdown: 5567999.999999993 at 2015-08-31 00:00:00

0.19 :

C(0.19) Average Annual P&L: 388166.6666666672
C(0.19) Annualised Sharpe Ratio: 0.179484013879144
C(0.19) Maximum Drawdown: 5589999.999999992 at 2015-08-31 00:00:00
```

```
0.2 :

C(0.2) Average Annual P&L: 354166.66666666715
C(0.2) Annualised Sharpe Ratio: 0.16375994552661843
C(0.2) Maximum Drawdown: 5589999.99999999 at 2015-08-31 00:00:00

0.21 :

C(0.21) Average Annual P&L: 338666.6666666678
C(0.21) Annualised Sharpe Ratio: 0.1565915341580261
C(0.21) Maximum Drawdown: 5731999.999999992 at 2015-08-31 00:00:00

0.22 :

C(0.22) Average Annual P&L: 300666.66666666657
C(0.22) Annualised Sharpe Ratio: 0.1390183518532155
C(0.22) Maximum Drawdown: 5800000.000000008 at 2019-06-12 00:00:00

0.23 :

C(0.23) Average Annual P&L: 293500.0
C(0.23) Annualised Sharpe Ratio: 0.1357033169863342
C(0.23) Maximum Drawdown: 5860000.00000001 at 2019-06-12 00:00:00

0.24 :

C(0.24) Average Annual P&L: 315833.33333333326
C(0.24) Annualised Sharpe Ratio: 0.1460302076181666
C(0.24) Maximum Drawdown: 5692000.000000013 at 2019-06-12 00:00:00

0.25 :

C(0.25) Average Annual P&L: 274833.33333333285
C(0.25) Annualised Sharpe Ratio: 0.1270704644321155
C(0.25) Maximum Drawdown: 5932000.000000012 at 2019-06-12 00:00:00

0.26 :

C(0.26) Average Annual P&L: 300833.3333333329
C(0.26) Annualised Sharpe Ratio: 0.13909269421607695
C(0.26) Maximum Drawdown: 5620000.000000013 at 2019-06-12 00:00:00

0.27 :

C(0.27) Average Annual P&L: 309333.33333333273
C(0.27) Annualised Sharpe Ratio: 0.14302469300600776
C(0.27) Maximum Drawdown: 6207999.999999994 at 2015-08-31 00:00:00

0.28 :

C(0.28) Average Annual P&L: 345833.3333333338
C(0.28) Annualised Sharpe Ratio: 0.15990628140492333
C(0.28) Maximum Drawdown: 5631999.999999991 at 2015-08-31 00:00:00

0.29 :

C(0.29) Average Annual P&L: 365833.33333333366
C(0.29) Annualised Sharpe Ratio: 0.1691563948502056
C(0.29) Maximum Drawdown: 5631999.999999992 at 2015-08-31 00:00:00
```

```
0.3 :

C(0.3) Average Annual P&L: 414000.0000000009
C(0.3) Annualised Sharpe Ratio: 0.19143336821827528
C(0.3) Maximum Drawdown: 5399999.999999993 at 2015-08-31 00:00:00

0.31 :

C(0.31) Average Annual P&L: 408833.333333334
C(0.31) Annualised Sharpe Ratio: 0.18904502896617403
C(0.31) Maximum Drawdown: 5461999.999999994 at 2015-08-31 00:00:00

0.32 :

C(0.32) Average Annual P&L: 376666.6666666665
C(0.32) Annualised Sharpe Ratio: 0.17416477383904538
C(0.32) Maximum Drawdown: 5788999.999999992 at 2015-12-24 00:00:00

0.33 :

C(0.33) Average Annual P&L: 482666.6666666663
C(0.33) Annualised Sharpe Ratio: 0.2231904813042629
C(0.33) Maximum Drawdown: 5672999.999999991 at 2015-12-24 00:00:00

0.34 :

C(0.34) Average Annual P&L: 501000.00000000006
C(0.34) Annualised Sharpe Ratio: 0.23166812485105936
C(0.34) Maximum Drawdown: 5672999.99999999 at 2015-12-24 00:00:00

0.35000000000000003 :

C(0.35000000000000003) Average Annual P&L: 526666.6666666676
C(0.35000000000000003) Annualised Sharpe Ratio: 0.24353799496454645
C(0.35000000000000003) Maximum Drawdown: 5494999.999999988 at 2015-12-24 00:0
0:00

0.36 :

C(0.36) Average Annual P&L: 491166.6666666674
C(0.36) Annualised Sharpe Ratio: 0.22711577645682918
C(0.36) Maximum Drawdown: 5388999.999999988 at 2015-12-24 00:00:00

0.37 :

C(0.37) Average Annual P&L: 562833.3333333343
C(0.37) Annualised Sharpe Ratio: 0.2602673795978604
C(0.37) Maximum Drawdown: 5388999.999999989 at 2015-12-24 00:00:00

0.38 :

C(0.38) Average Annual P&L: 557000.0000000013
C(0.38) Annualised Sharpe Ratio: 0.25756977422642974
C(0.38) Maximum Drawdown: 5388999.999999989 at 2015-12-24 00:00:00

0.39 :

C(0.39) Average Annual P&L: 619500.0000000013
C(0.39) Annualised Sharpe Ratio: 0.2864833046396058
C(0.39) Maximum Drawdown: 5388999.99999999 at 2015-12-24 00:00:00
```

```
0.4 :

C(0.4) Average Annual P&L: 656500.0000000008
C(0.4) Annualised Sharpe Ratio: 0.3035997575450391
C(0.4) Maximum Drawdown: 5388999.999999989 at 2015-12-24 00:00:00

0.41000000000000003 :

C(0.41000000000000003) Average Annual P&L: 658000.0000000002
C(0.41000000000000003) Annualised Sharpe Ratio: 0.3042925043372581
C(0.41000000000000003) Maximum Drawdown: 5488999.999999991 at 2015-12-24 00:0
0:00

0.42 :

C(0.42) Average Annual P&L: 619666.6666666673
C(0.42) Annualised Sharpe Ratio: 0.2865580822634739
C(0.42) Maximum Drawdown: 5618999.999999989 at 2015-12-24 00:00:00

0.43 :

C(0.43) Average Annual P&L: 661833.3333333333
C(0.43) Annualised Sharpe Ratio: 0.3060635238764539
C(0.43) Maximum Drawdown: 5198999.999999991 at 2015-12-24 00:00:00

0.44 :

C(0.44) Average Annual P&L: 713166.6666666659
C(0.44) Annualised Sharpe Ratio: 0.32981238742091623
C(0.44) Maximum Drawdown: 5062999.999999993 at 2015-12-24 00:00:00

0.45 :

C(0.45) Average Annual P&L: 688166.6666666662
C(0.45) Annualised Sharpe Ratio: 0.3182491498479676
C(0.45) Maximum Drawdown: 5236999.999999993 at 2016-02-02 00:00:00

0.46 :

C(0.46) Average Annual P&L: 613500.0000000002
C(0.46) Annualised Sharpe Ratio: 0.28370496412765334
C(0.46) Maximum Drawdown: 5150999.999999993 at 2016-02-02 00:00:00

0.47000000000000003 :

C(0.47000000000000003) Average Annual P&L: 615833.3333333331
C(0.47000000000000003) Annualised Sharpe Ratio: 0.2847827257882117
C(0.47000000000000003) Maximum Drawdown: 5150999.999999993 at 2016-02-02 00:0
0:00

0.48 :

C(0.48) Average Annual P&L: 599166.6666666666
C(0.48) Annualised Sharpe Ratio: 0.27707280329255896
C(0.48) Maximum Drawdown: 4656999.999999993 at 2016-02-02 00:00:00

0.49 :

C(0.49) Average Annual P&L: 540166.6666666666
```

```
C(0.49) Annualised Sharpe Ratio: 0.24977940359995002
C(0.49) Maximum Drawdown: 4904999.9999999935 at 2016-02-02 00:00:00

0.5 :

C(0.5) Average Annual P&L: 566999.9999999994
C(0.5) Annualised Sharpe Ratio: 0.26219810469295385
C(0.5) Maximum Drawdown: 4756999.9999999935 at 2016-02-02 00:00:00

0.51 :

C(0.51) Average Annual P&L: 577999.9999999999
C(0.51) Annualised Sharpe Ratio: 0.2672882436924898
C(0.51) Maximum Drawdown: 4780999.999999994 at 2016-02-02 00:00:00

0.52 :

C(0.52) Average Annual P&L: 500499.99999999994
C(0.52) Annualised Sharpe Ratio: 0.2314359781625707
C(0.52) Maximum Drawdown: 5141000.0 at 2016-02-02 00:00:00

0.53 :

C(0.53) Average Annual P&L: 456499.9999999993
C(0.53) Annualised Sharpe Ratio: 0.2110842514001262
C(0.53) Maximum Drawdown: 5141000.0 at 2016-02-02 00:00:00

0.54 :

C(0.54) Average Annual P&L: 459833.33333333296
C(0.54) Annualised Sharpe Ratio: 0.21262644994606353
C(0.54) Maximum Drawdown: 5141000.0 at 2016-02-02 00:00:00

0.55 :

C(0.55) Average Annual P&L: 524166.6666666663
C(0.55) Annualised Sharpe Ratio: 0.24238179486172645
C(0.55) Maximum Drawdown: 4751000.0 at 2016-02-02 00:00:00

0.56 :

C(0.56) Average Annual P&L: 508499.999999999
C(0.56) Annualised Sharpe Ratio: 0.2351338336069687
C(0.56) Maximum Drawdown: 4751000.0 at 2016-02-02 00:00:00

0.5700000000000001 :

C(0.5700000000000001) Average Annual P&L: 533499.9999999997
C(0.5700000000000001) Annualised Sharpe Ratio: 0.24669398690224942
C(0.5700000000000001) Maximum Drawdown: 4750999.999999999 at 2016-02-02 00:00:
00

0.58 :

C(0.58) Average Annual P&L: 525333.3333333336
C(0.58) Annualised Sharpe Ratio: 0.24291577555447733
C(0.58) Maximum Drawdown: 4784999.999999996 at 2011-08-09 00:00:00

0.59 :
```

```
C(0.59) Average Annual P&L: 562333.3333333329
C(0.59) Annualised Sharpe Ratio: 0.26002935589469267
C(0.59) Maximum Drawdown: 4514999.999999996 at 2016-02-02 00:00:00

0.6 :

C(0.6) Average Annual P&L: 578499.9999999993
C(0.6) Annualised Sharpe Ratio: 0.2675065174948227
C(0.6) Maximum Drawdown: 4514999.999999997 at 2016-02-02 00:00:00

0.61 :

C(0.61) Average Annual P&L: 533833.3333333329
C(0.61) Annualised Sharpe Ratio: 0.2468478539174876
C(0.61) Maximum Drawdown: 4858999.999999999 at 2011-08-09 00:00:00

0.62 :

C(0.62) Average Annual P&L: 529833.3333333327
C(0.62) Annualised Sharpe Ratio: 0.2449972355086305
C(0.62) Maximum Drawdown: 4858999.999999999 at 2011-08-09 00:00:00

0.63 :

C(0.63) Average Annual P&L: 523999.9999999995
C(0.63) Annualised Sharpe Ratio: 0.24229649038043255
C(0.63) Maximum Drawdown: 4874999.999999999 at 2011-08-09 00:00:00

0.64 :

C(0.64) Average Annual P&L: 550166.6666666664
C(0.64) Annualised Sharpe Ratio: 0.25441155449343567
C(0.64) Maximum Drawdown: 4962999.999999997 at 2011-08-09 00:00:00

0.65 :

C(0.65) Average Annual P&L: 525499.9999999993
C(0.65) Annualised Sharpe Ratio: 0.2429995380419912
C(0.65) Maximum Drawdown: 4962999.999999997 at 2011-08-09 00:00:00

0.66 :

C(0.66) Average Annual P&L: 590333.3333333331
C(0.66) Annualised Sharpe Ratio: 0.2729924815352075
C(0.66) Maximum Drawdown: 4962999.999999997 at 2011-08-09 00:00:00

0.67 :

C(0.67) Average Annual P&L: 629666.6666666659
C(0.67) Annualised Sharpe Ratio: 0.2911873630482925
C(0.67) Maximum Drawdown: 5056999.999999997 at 2011-08-09 00:00:00

0.68 :

C(0.68) Average Annual P&L: 622833.3333333329
C(0.68) Annualised Sharpe Ratio: 0.2880206272908279
C(0.68) Maximum Drawdown: 5056999.999999997 at 2011-08-09 00:00:00

0.6900000000000001 :
```

```
C(0.6900000000000001) Average Annual P&L: 647499.9999999993
C(0.6900000000000001) Annualised Sharpe Ratio: 0.2994295576065485
C(0.6900000000000001) Maximum Drawdown: 5026999.999999999 at 2011-08-09 00:00:
00

0.7000000000000001 :

C(0.7000000000000001) Average Annual P&L: 648499.9999999997
C(0.7000000000000001) Annualised Sharpe Ratio: 0.2998904424108772
C(0.7000000000000001) Maximum Drawdown: 5026999.999999999 at 2011-08-09 00:00:
00

0.71 :

C(0.71) Average Annual P&L: 681666.6666666664
C(0.71) Annualised Sharpe Ratio: 0.3152370637367297
C(0.71) Maximum Drawdown: 5026999.999999999 at 2011-08-09 00:00:00

0.72 :

C(0.72) Average Annual P&L: 683333.3333333328
C(0.72) Annualised Sharpe Ratio: 0.3160077820017607
C(0.72) Maximum Drawdown: 5026999.999999999 at 2011-08-09 00:00:00

0.73 :

C(0.73) Average Annual P&L: 625833.3333333333
C(0.73) Annualised Sharpe Ratio: 0.28939832017722816
C(0.73) Maximum Drawdown: 5026999.999999999 at 2011-08-09 00:00:00

0.74 :

C(0.74) Average Annual P&L: 657833.3333333333
C(0.74) Annualised Sharpe Ratio: 0.3042055447373633
C(0.74) Maximum Drawdown: 5012999.999999998 at 2011-08-09 00:00:00

0.75 :

C(0.75) Average Annual P&L: 661833.3333333329
C(0.75) Annualised Sharpe Ratio: 0.3060553072490553
C(0.75) Maximum Drawdown: 4542999.999999998 at 2011-08-09 00:00:00

0.76 :

C(0.76) Average Annual P&L: 567500.0000000002
C(0.76) Annualised Sharpe Ratio: 0.26241264896527383
C(0.76) Maximum Drawdown: 4994999.999999995 at 2016-02-02 00:00:00

0.77 :

C(0.77) Average Annual P&L: 624000.0000000003
C(0.77) Annualised Sharpe Ratio: 0.28854434121014266
C(0.77) Maximum Drawdown: 4994999.999999995 at 2016-02-02 00:00:00

0.78 :

C(0.78) Average Annual P&L: 606833.3333333337
C(0.78) Annualised Sharpe Ratio: 0.2806002890278103
C(0.78) Maximum Drawdown: 5146999.999999994 at 2016-02-02 00:00:00
```

```
0.79 :

C(0.79) Average Annual P&L: 565500.0000000007
C(0.79) Annualised Sharpe Ratio: 0.2614791361697736
C(0.79) Maximum Drawdown: 5794999.999999993 at 2016-02-02 00:00:00

0.8 :

C(0.8) Average Annual P&L: 535166.6666666672
C(0.8) Annualised Sharpe Ratio: 0.2474501225446935
C(0.8) Maximum Drawdown: 5608999.999999992 at 2016-02-02 00:00:00

0.81 :

C(0.81) Average Annual P&L: 300333.3333333329
C(0.81) Annualised Sharpe Ratio: 0.13885300925453065
C(0.81) Maximum Drawdown: 5608999.999999993 at 2016-02-02 00:00:00

0.8200000000000001 :

C(0.8200000000000001) Average Annual P&L: 188166.66666666637
C(0.8200000000000001) Annualised Sharpe Ratio: 0.08698962081085376
C(0.8200000000000001) Maximum Drawdown: 5164999.999999993 at 2016-02-02 00:00:
00

0.8300000000000001 :

C(0.8300000000000001) Average Annual P&L: 214499.9999999994
C(0.8300000000000001) Annualised Sharpe Ratio: 0.09916264130560586
C(0.8300000000000001) Maximum Drawdown: 5192999.999999991 at 2016-02-02 00:00:
00

0.84 :

C(0.84) Average Annual P&L: 133166.66666666538
C(0.84) Annualised Sharpe Ratio: 0.06156026274474255
C(0.84) Maximum Drawdown: 5622999.9999999935 at 2016-02-02 00:00:00

0.85 :

C(0.85) Average Annual P&L: 87333.33333333192
C(0.85) Annualised Sharpe Ratio: 0.04037205899007997
C(0.85) Maximum Drawdown: 5640999.999999995 at 2016-02-02 00:00:00

0.86 :

C(0.86) Average Annual P&L: 62499.999999998436
C(0.86) Annualised Sharpe Ratio: 0.02889221645028714
C(0.86) Maximum Drawdown: 5810999.999999996 at 2016-02-02 00:00:00

0.87 :

C(0.87) Average Annual P&L: 82166.66666666506
C(0.87) Annualised Sharpe Ratio: 0.03798395729268973
C(0.87) Maximum Drawdown: 5810999.999999997 at 2016-02-02 00:00:00

0.88 :

C(0.88) Average Annual P&L: 135166.66666666578
C(0.88) Annualised Sharpe Ratio: 0.062485810514852684
```

```
C(0.88) Maximum Drawdown: 5810999.999999995 at 2016-02-02 00:00:00

0.89 :

C(0.89) Average Annual P&L: 107666.66666666527
C(0.89) Annualised Sharpe Ratio: 0.04977171542952236
C(0.89) Maximum Drawdown: 5870999.999999997 at 2016-02-02 00:00:00

0.9 :

C(0.9) Average Annual P&L: 173499.9999999986
C(0.9) Annualised Sharpe Ratio: 0.08020589837744435
C(0.9) Maximum Drawdown: 5870999.999999996 at 2016-02-02 00:00:00

0.91 :

C(0.91) Average Annual P&L: 133499.99999999822
C(0.91) Annualised Sharpe Ratio: 0.06171382171893575
C(0.91) Maximum Drawdown: 5744999.999999994 at 2016-02-02 00:00:00

0.92 :

C(0.92) Average Annual P&L: 153166.6666666648
C(0.92) Annualised Sharpe Ratio: 0.07080580959119168
C(0.92) Maximum Drawdown: 5744999.9999999935 at 2016-02-02 00:00:00

0.93 :

C(0.93) Average Annual P&L: 168999.99999999814
C(0.93) Annualised Sharpe Ratio: 0.07812685802094461
C(0.93) Maximum Drawdown: 5382999.999999997 at 2016-02-02 00:00:00

0.9400000000000001 :

C(0.9400000000000001) Average Annual P&L: 175666.6666666648
C(0.9400000000000001) Annualised Sharpe Ratio: 0.08120780756934516
C(0.9400000000000001) Maximum Drawdown: 5258999.999999998 at 2016-02-02 00:00:
00

0.9500000000000001 :

C(0.9500000000000001) Average Annual P&L: 127333.33333333116
C(0.9500000000000001) Annualised Sharpe Ratio: 0.0588630601826729
C(0.9500000000000001) Maximum Drawdown: 5394999.999999998 at 2016-02-02 00:00:
00

0.96 :

C(0.96) Average Annual P&L: 148166.66666666456
C(0.96) Annualised Sharpe Ratio: 0.06849531530053633
C(0.96) Maximum Drawdown: 5128000.000000002 at 2015-01-16 00:00:00

0.97 :

C(0.97) Average Annual P&L: 114999.99999999806
C(0.97) Annualised Sharpe Ratio: 0.05316195819676684
C(0.97) Maximum Drawdown: 5170999.999999998 at 2016-02-02 00:00:00

0.98 :
```

```
C(0.98) Average Annual P&L: 78999.99999999808
C(0.98) Annualised Sharpe Ratio: 0.03651955481251295
C(0.98) Maximum Drawdown: 5116999.999999997 at 2016-02-02 00:00:00

0.99 :

C(0.99) Average Annual P&L: 36333.333333330804
C(0.99) Annualised Sharpe Ratio: 0.016795670832830217
C(0.99) Maximum Drawdown: 5145000.000000006 at 2016-02-02 00:00:00

1.0 :

C(1.0) Average Annual P&L: 108166.66666666469
C(1.0) Annualised Sharpe Ratio: 0.05000294583472322
C(1.0) Maximum Drawdown: 5102000.000000004 at 2015-01-16 00:00:00
```

In [45]:

```
print(f'Best Sharpe Ratio: {SRoptimal}, Best epsilon: {Eopt}')
```

```
Best Sharpe Ratio: 0.4206029259602411, Best epsilon: 0.0
```

Thus the optimal Sharpe Ratio is 0.42 when $\epsilon = 0$. Let's experiment with higher values of ϵ:

In [46]:

```
epsilons = [10,9,8,7,6,5,4,3,2,0]
SRoptimal = 0.0
Eopt = 0.0

for epsilon in epsilons:
    print(epsilon,":")
    print("")

    df_wti['C(0)']  = 0.0
    for i in range(len(df_wti)):
        if df_wti['F1 - Predicted'][i] > epsilon:
            df_wti['C(0)'][i] = -1.0
        elif df_wti['F1 - Predicted'][i] < epsilon:
            df_wti['C(0)'][i] = 1.0

    df_wti['P/L_C(0)'] = 0.0
    df_wti['Cumulative_P/L_C(0)'] = 0.0

    # deciding whether your P&L is from the long or short position
    for i in range(1,len(df_wti)):
        if (df_wti['C(0)'][i-1] == 1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L'][i]

        if (df_wti['C(0)'][i-1] == -1):
            df_wti['P/L_C(0)'][i] = df_wti['P/L_short'][i]


    # if it is time to switch positions you have to add the transaction costs
    for i in range(1,len(df_wti)):
        if df_wti['C(0)'][i] != df_wti['C(0)'][i-1]:
            df_wti['P/L_C(0)'][i] -= 0.02

    # calculating cumulative P&L
    for i in range(1,len(df_wti)):
        df_wti['Cumulative_P/L_C(0)'][i] = df_wti['P/L_C(0)'][i] + df_wti['Cumu
    df_wti['P(t)_C(0)'] = 0.0

    for i in range(1,len(df_wti)):
        df_wti['P(t)_C(0)'][i] = df_wti['P(t)_C(0)'][i-1] +(df_wti['P/L_C(0)'][

    df_wti['dF(t)_C(0)'] = df_wti['P/L_C(0)']*100000

    ann_PL_C = df_wti['P(t)_C(0)'][-1]/N
    print(f"C({epsilon}) Average Annual P&L: {ann_PL_C}")

    ann_SR_C = ann_PL_C/(np.std(df_wti['dF(t)_C(0)'])*np.sqrt(250))
    print(f"C({epsilon}) Annualised Sharpe Ratio: {ann_SR_C}")

    df_wti['HWM_C(0)'] = 0.0
    df_wti['DD_C(0)'] = 0.0

    for i in range(len(df_wti)):
        df_wti['HWM_C(0)'][i] = np.max(df_wti['P(t)_C(0)'][0:i+1])
        df_wti['DD_C(0)'][i] = df_wti['HWM_C(0)'][i] - df_wti['P(t)_C(0)'][i]

    mdd_C = np.max(df_wti['DD_C(0)'])
    mdd_Cdate = df_wti['DD_C(0)'].idxmax()

    print(f"C({epsilon}) Maximum Drawdown: {mdd_C} at {mdd_Cdate}")
    print("")

    if ann_SR_C > SRoptimal:
        SRoptimal = ann_SR_C
        Eopt = epsilon
```

```
10 :

C(10) Average Annual P&L: -485166.6666666669
C(10) Annualised Sharpe Ratio: -0.22397337097477873
C(10) Maximum Drawdown: 12179999.999999989 at 2020-04-21 00:00:00

9 :

C(9) Average Annual P&L: -663333.333333333
C(9) Annualised Sharpe Ratio: -0.3062417253913218
C(9) Maximum Drawdown: 13925000.000000002 at 2020-05-26 00:00:00

8 :

C(8) Average Annual P&L: -664333.3333333334
C(8) Annualised Sharpe Ratio: -0.30669962088755115
C(8) Maximum Drawdown: 14325000.000000004 at 2020-05-26 00:00:00

7 :

C(7) Average Annual P&L: -592333.3333333323
C(7) Annualised Sharpe Ratio: -0.27346926153005147
C(7) Maximum Drawdown: 13267999.999999993 at 2020-10-30 00:00:00

6 :

C(6) Average Annual P&L: -728166.6666666665
C(6) Annualised Sharpe Ratio: -0.33621751431102104
C(6) Maximum Drawdown: 14457999.999999998 at 2020-10-30 00:00:00

5 :

C(5) Average Annual P&L: -938833.3333333296
C(5) Annualised Sharpe Ratio: -0.43357470823106187
C(5) Maximum Drawdown: 16097999.999999955 at 2020-10-30 00:00:00

4 :

C(4) Average Annual P&L: -935499.9999999953
C(4) Annualised Sharpe Ratio: -0.43210219902485664
C(4) Maximum Drawdown: 15873999.999999948 at 2020-10-30 00:00:00

3 :

C(3) Average Annual P&L: -1097166.6666666635
C(3) Annualised Sharpe Ratio: -0.5069447086428552
C(3) Maximum Drawdown: 16953999.99999996 at 2020-10-30 00:00:00

2 :

C(2) Average Annual P&L: -621333.3333333319
C(2) Annualised Sharpe Ratio: -0.28714565600996433
C(2) Maximum Drawdown: 10602000.000000004 at 2016-02-08 00:00:00

0 :

C(0) Average Annual P&L: 909333.3333333308
C(0) Annualised Sharpe Ratio: 0.4206029259602411
C(0) Maximum Drawdown: 5717000.000000048 at 2021-11-01 00:00:00
```

So $\epsilon = 0$, $w = 50$ is the optimal pairing here. Our final results are:

- C(0) Average Annual P&L: 909333.3333333308
- C(0) Annualised Sharpe Ratio: 0.42060292596024096
- C(0) Maximum Drawdown: 5717000.000000048 at 2021-11-01 00:00:00

Let's finally see the tail end of the dataframe and also graph the equity line for these optimal parameters:

In [47]:

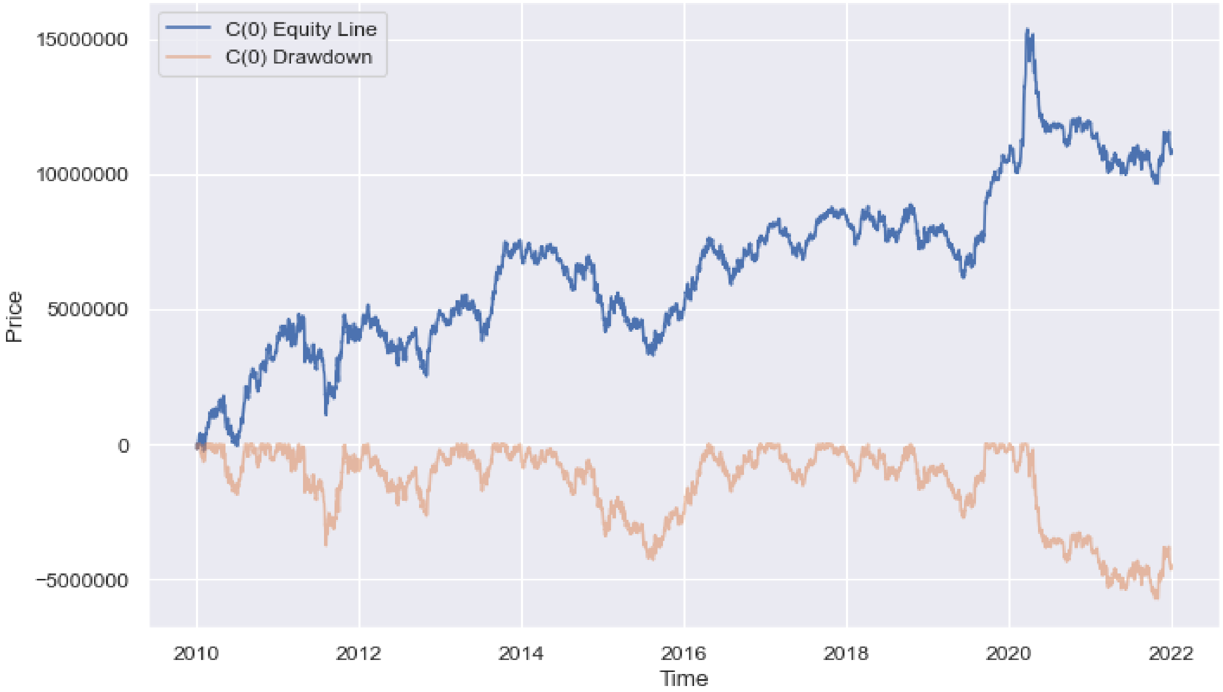
```
df_wti.tail()
```

Out[47]:

	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_P/
Date									
2021-12-27	75.57	75.18	74.67	74.12	0	0	1.78	1.76	
2021-12-28	75.98	75.60	75.13	74.59	0	0	0.41	0.42	
2021-12-29	76.56	76.18	75.71	75.18	0	0	0.58	0.58	
2021-12-30	76.99	76.61	76.13	75.58	0	0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0	0	-1.78	-1.73	

In [48]:

```
plt.figure(figsize=(10, 6), dpi=80)
plt.ticklabel_format(style='plain')
plt.plot(df_wti.index, df_wti['P(t)_C(0)'], label = 'C(0) Equity Line')
plt.plot(df_wti.index, -df_wti['DD_C(0)'], label = 'C(0) Drawdown',alpha=0.5)
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.savefig('equity line and drawdown optimized')
plt.show()
```



Let s downdload the data in an excel file

In [50]:

```
# determining the name of the file
file_name = 'model_strategy_Arthur_Arjun.xlsx'

# saving the excel
df_wti.to_excel(file_name)
print('DataFrame is written to Excel File successfully.')
```

```
DataFrame is written to Excel File successfully.
```

In [ ]:

In [ ]:

On top of this, and in light of the current political landscape, we also thought it would be interesting to add war sentiment into the strategy somehow. So let's first find and import some data to hopefully derive some conclusions on a more complex strategy. The data we will be using is deaths from conflict and terrorism per 100,000 in the Middle East & North Africa:

In [51]:

```
data_war          = pd.read_csv('deaths-conflict-terrorism-per-100000.csv')
data_war          = data_war[data_war['Entity'] == 'Middle East & North Africa']
data_war          = data_war[['Year','Deaths - Conflict and terrorism - Sex: Bot
data_war.rename(columns={"Year": "Date", "Deaths - Conflict and terrorism - Sex
data_war          = data_war.set_index('Date')
data_war.index    = pd.to_datetime(data_war.index,format='%Y')
data_war          = data_war[(data_war.index >= '2010-01-01') & (data_war.index
data_war.head(15)
```

Out[51]:

Date	Deaths
2010-01-01	2.241973
2011-01-01	7.131705
2012-01-01	17.841588
2013-01-01	16.821756
2014-01-01	28.672619
2015-01-01	23.710162
2016-01-01	24.224164
2017-01-01	17.838126
2018-01-01	10.961725
2019-01-01	5.222731

In [52]:

```
df_wti2       = df_wti
wti_war = df_wti2.merge(data_war,how='outer',left_index=True,right_index=True)
wti_war
```

Out[52]:

Date	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_F
2010-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2010-01-04	81.51	82.12	82.65	83.12	0.0	0.0	2.15	2.10	
2010-01-05	81.77	82.41	82.99	83.52	0.0	0.0	0.26	0.29	
2010-01-06	83.18	83.75	84.31	84.86	0.0	0.0	1.41	1.34	
2010-01-07	82.66	83.19	83.75	84.29	0.0	0.0	-0.52	-0.56	
...	...	...	...	...	...	...	...	...	
2021-12-27	75.57	75.18	74.67	74.12	0.0	0.0	1.78	1.76	
2021-12-28	75.98	75.60	75.13	74.59	0.0	0.0	0.41	0.42	
2021-12-29	76.56	76.18	75.71	75.18	0.0	0.0	0.58	0.58	
2021-12-30	76.99	76.61	76.13	75.58	0.0	0.0	0.43	0.43	
2021-12-31	75.21	74.88	74.45	73.94	0.0	0.0	-1.78	-1.73	

3032 rows × 21 columns

Interpolate the death count:

In [53]:

```
wti_war['Deaths'].interpolate(inplace=True)
wti_war = wti_war.loc['2010-01-04':'2019-12-31']
wti_war
```

Out[53]:

	F1	F2	F3	F4	Execution_day	Holding_type	P/L_F1	P/L_F2	Cumulative_F
Date									
2010-01-04	81.51	82.12	82.65	83.12	0.0	0.0	2.15	2.10	
2010-01-05	81.77	82.41	82.99	83.52	0.0	0.0	0.26	0.29	
2010-01-06	83.18	83.75	84.31	84.86	0.0	0.0	1.41	1.34	
2010-01-07	82.66	83.19	83.75	84.29	0.0	0.0	-0.52	-0.56	
2010-01-08	82.75	83.30	83.87	84.47	0.0	0.0	0.09	0.11	
...	...	...	...	...	...	...	...	...	
2019-12-24	61.11	60.94	60.65	60.26	0.0	0.0	0.59	0.59	
2019-12-26	61.68	61.48	61.16	60.73	0.0	0.0	0.57	0.54	
2019-12-27	61.72	61.53	61.21	60.79	0.0	0.0	0.04	0.05	
2019-12-30	61.68	61.44	61.10	60.66	0.0	0.0	-0.04	-0.09	
2019-12-31	61.06	60.77	60.41	59.97	0.0	0.0	-0.62	-0.67	

2528 rows × 21 columns

In [54]:

```
wti_war['Deaths'].dtypes
```

Out[54]:

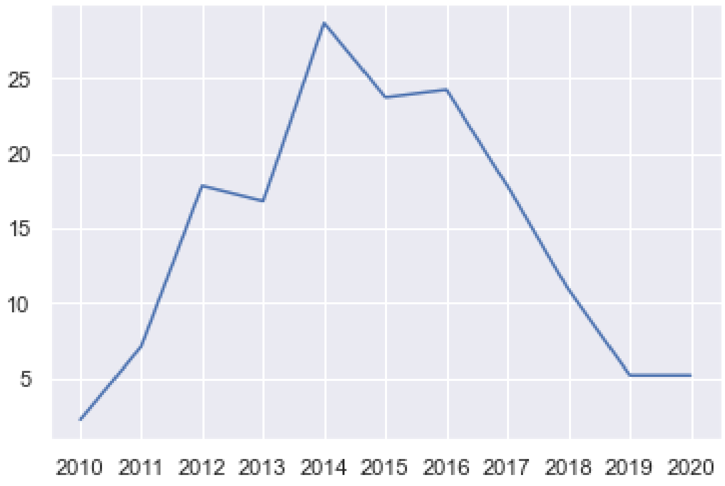
```
dtype('float64')
```

In [55]:

```
plt.plot(wti_war['Deaths'])
```

Out[55]:

```
[<matplotlib.lines.Line2D at 0x282a735b0>]
```



The strategy is to hold nothing when the deaths are above the threshold. We initially did this with no time lag we decided to add a time lag of approximately 1 year as acts of war and terror usually occur in succession:

In [56]:

```
wti_war.dropna(inplace=True)
thresholds = [25.0,24.0,23.0,22.0,21.0]
SRoptimal = 0.0
Topt = 0.0
epsilon = 0

for t in thresholds:
    print(t,":")
    print("")

    wti_war['C(0)']  = None
    for i in range(len(wti_war)):

        if wti_war['C(0)'][i] == 0.0:
            continue
        if wti_war['Deaths'][i] > t:
            wti_war['C(0)'][i+250] = 0.0

        if wti_war['F1 - Predicted'][i] > epsilon:
                wti_war['C(0)'][i] = -1.0
        elif wti_war['F1 - Predicted'][i] < epsilon:
                wti_war['C(0)'][i] = 1.0

    wti_war['P/L_C(0)'] = 0.0
    wti_war['Cumulative_P/L_C(0)'] = 0.0

    # deciding whether your P&L is from the long or short position
    for i in range(1,len(wti_war)):
        if (wti_war['C(0)'][i-1] == 0.0):
            wti_war['P/L_C(0)'][i] = 0.0

        elif (wti_war['C(0)'][i-1] == 1.0):
            wti_war['P/L_C(0)'][i] = wti_war['P/L'][i]

        elif (wti_war['C(0)'][i-1] == -1.0):
            wti_war['P/L_C(0)'][i] = wti_war['P/L_short'][i]
    # if it is time to switch positions you have to add the transaction costs
    for i in range(1,len(wti_war)):
        if wti_war['C(0)'][i] != wti_war['C(0)'][i-1]:
            wti_war['P/L_C(0)'][i] -= 0.02

    # calculating cumulative P&L
    for i in range(1,len(wti_war)):
        wti_war['Cumulative_P/L_C(0)'][i] = wti_war['P/L_C(0)'][i] + wti_war['C

    wti_war['P(t)_C(0)'] = 0.0

    for i in range(1,len(wti_war)):
        wti_war['P(t)_C(0)'][i] = wti_war['P(t)_C(0)'][i-1] +(wti_war['P/L_C(0)

    wti_war['dF(t)_C(0)'] = wti_war['P/L_C(0)']*100000

    ann_PL_C = wti_war['P(t)_C(0)'][-1]/N
    print(f"C(0) Average Annual P&L: {ann_PL_C}")

    ann_SR_C = ann_PL_C/(np.std(wti_war['dF(t)_C(0)'])*np.sqrt(250))
    print(f"C(0) Annualised Sharpe Ratio: {ann_SR_C}")

    wti_war['HWM_C(0)'] = 0.0
    wti_war['DD_C(0)'] = 0.0

    for i in range(len(wti_war)):
        wti_war['HWM_C(0)'][i] = np.max(wti_war['P(t)_C(0)'][0:i+1])
        wti_war['DD_C(0)'][i] = wti_war['HWM_C(0)'][i] - wti_war['P(t)_C(0)'][i

    mdd_C = np.max(wti_war['DD_C(0)'])
    mdd_Cdate = wti_war['DD_C(0)'].idxmax()

    print(f"C(0) Maximum Drawdown: {mdd_C} at {mdd_Cdate}")
    print("")

    if ann_SR_C > SRoptimal:
        SRoptimal = ann_SR_C
        Topt = t
```

```
25.0 :

C(0) Average Annual P&L: 1104750.000000002
C(0) Annualised Sharpe Ratio: 0.5679812470953957
C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00

24.0 :

C(0) Average Annual P&L: 942250.0000000013
C(0) Annualised Sharpe Ratio: 0.49024795448898284
C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00

23.0 :

C(0) Average Annual P&L: 982333.3333333348
C(0) Annualised Sharpe Ratio: 0.5143571954710559
C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00

22.0 :

C(0) Average Annual P&L: 1067750.0000000012
C(0) Annualised Sharpe Ratio: 0.5618169433532283
C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00

21.0 :

C(0) Average Annual P&L: 1164000.0000000016
C(0) Annualised Sharpe Ratio: 0.611558791497197
C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00
```

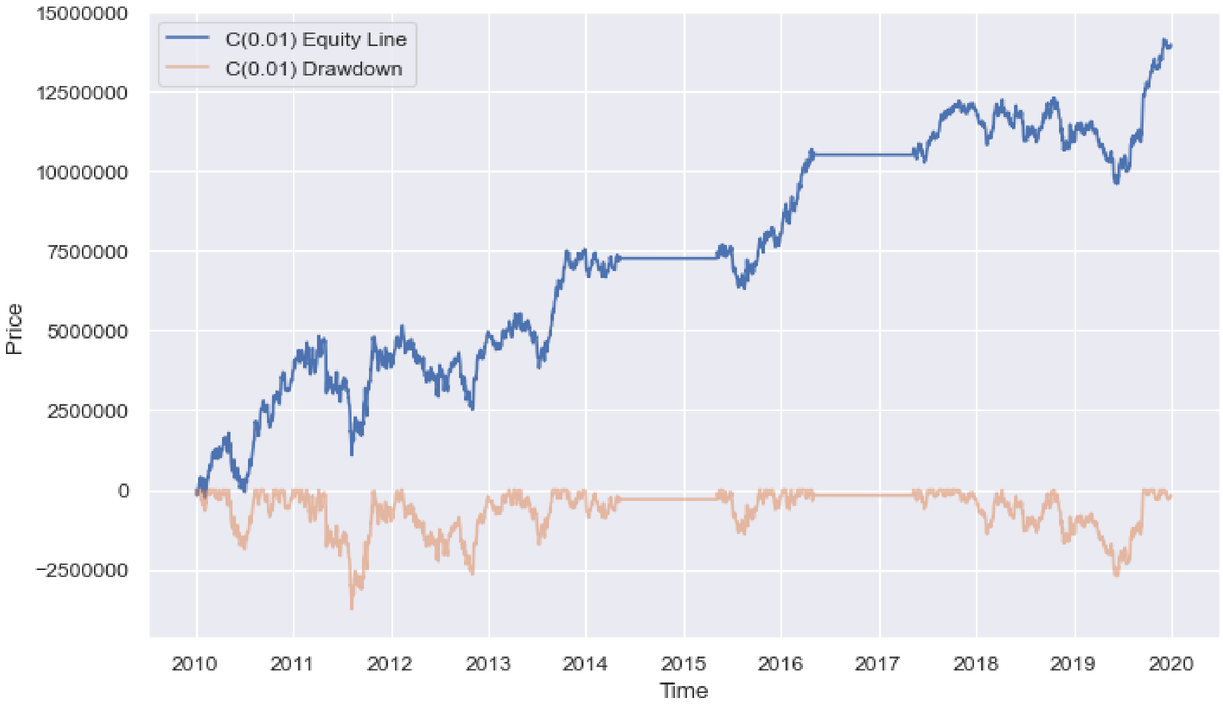
Now our optimal results are with a threshold of 21:

- C(0) Average Annual P&L: 1164000.0000000016
- C(0) Annualised Sharpe Ratio: 0.6115587914972023
- C(0) Maximum Drawdown: 3742999.999999998 at 2011-08-09 00:00:00

Let's also graph the equity line below:

In [57]:

```
plt.figure(figsize=(10, 6), dpi=80)
plt.ticklabel_format(style='plain')
plt.plot(wti_war.index, wti_war['P(t)_C(0)'], label = 'C(0.01) Equity Line')
plt.plot(wti_war.index, -wti_war['DD_C(0)'], label = 'C(0.01) Drawdown',alpha=0
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.savefig('EL + drawdown and geopolitical risk reduction')
plt.show()
```



Although the war-based strategy is obviously not the most robust and may be completely random, we can see that by assigning a threshold to terminate trading we have considerably reduced maximum drawdown and thus reduced risk and improved P&L. We have also improved the Sharpe ratio by so this experiment was surprisingly successful. By sampling over a larger range we may be able to see more reliable patterns, but we may not have spent enough time in a globalising economy to notice patterns/time-lags like these.

In [ ]:

In [58]:

```
# determining the name of the file
file_name = 'model_strategy_Arthur_Arjun_with_geopolitical_risk_reduction.xlsx'

# saving the excel
df_wti.to_excel(file_name)
print('DataFrame is written to Excel File successfully.')
```

```
DataFrame is written to Excel File successfully.
```