TESLA OPTION CHAIN SIMULATION (16 JUNE 2023)

LONG STRADDLE

HEMANT THAPA

In [56]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import yfinance as yf
import warnings
warnings.filterwarnings("ignore")
```

A long straddle is an options trading strategy that involves buying both a call option and a put option with the same strike price and expiration date. This strategy is used when the trader expects a significant price movement in the underlying asset but is uncertain about the direction of the movement.

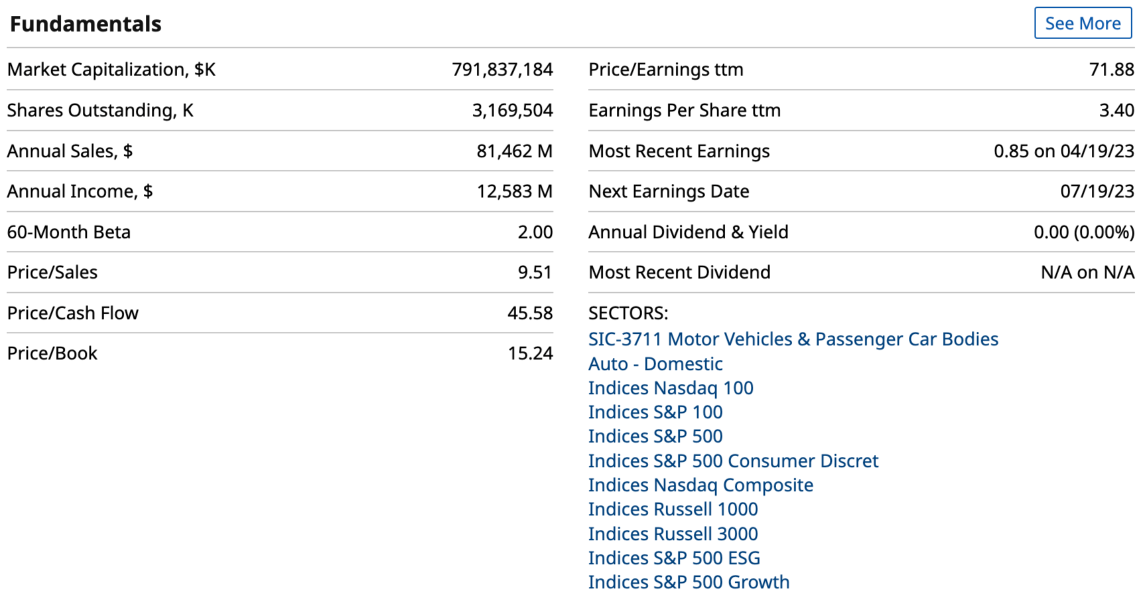
1. Buying a call option: By purchasing a call option, the trader gains the right to buy the underlying asset at the strike price within a specified time frame. This allows them to profit from an upward price movement.

1. Buying a put option: Simultaneously, the trader also buys a put option, which gives them the right to sell the underlying asset at the strike price within a specified time frame. This allows them to profit from a downward price movement.

The main idea behind a long straddle is to profit from a significant price swing, regardless of whether it goes up or down. The strategy benefits from high volatility, as a substantial price movement increases the chances of one of the options being in the money and offsetting the loss on the other option.

Long straddles are often used when traders anticipate events that can cause substantial price volatility, such as earnings announcements, regulatory decisions, or significant news releases. Traders employ this strategy to potentially capitalize on a large price move while limiting their downside risk to the initial cost of purchasing the options.

TESLA



Fundamentals

See More

Market Capitalization, $K	791,837,184	Price/Earnings ttm	71.88
Shares Outstanding, K	3,169,504	Earnings Per Share ttm	3.40
Annual Sales, $	81,462 M	Most Recent Earnings	0.85 on 04/19/23
Annual Income, $	12,583 M	Next Earnings Date	07/19/23
60-Month Beta	2.00	Annual Dividend & Yield	0.00 (0.00%)
Price/Sales	9.51	Most Recent Dividend	N/A on N/A
Price/Cash Flow	45.58	SECTORS: SIC-3711 Motor Vehicles & Passenger Car Bodies Auto - Domestic Indices Nasdaq 100 Indices S&P 100 Indices S&P 500 Indices S&P 500 Consumer Discret Indices Nasdaq Composite Indices Russell 1000 Indices Russell 3000 Indices S&P 500 ESG Indices S&P 500 Growth	
Price/Book	15.24		

```
In [57]: df = pd.read_csv(r'unusual-stock-options-activity-06-13-2023.csv')
```

```
In [58]: tesla = df[df['Symbol']=='TSLA']
```

```
In [59]: tesla[:10]
```

Out[59]:

	Symbol	Price	Type	Strike	Exp Date	DTE	Bid	Midpoint	Ask	Last	Volume	Open In
7	TSLA	259.44	Put	257.50	2023-06-16	3.0	5.65	5.70	5.75	5.75	18850.0	354.
12	TSLA	259.44	Put	253.33	2023-06-16	3.0	3.80	3.83	3.85	3.85	19470.0	524.
13	TSLA	259.44	Put	257.50	2023-06-23	10.0	8.90	8.95	9.00	8.95	4609.0	133.
19	TSLA	259.44	Call	290.00	2023-06-30	17.0	3.90	3.95	4.00	3.95	22413.0	950.
26	TSLA	259.44	Put	195.00	2023-11-17	157.0	9.75	9.83	9.90	9.85	9499.0	487.
30	TSLA	259.44	Put	255.00	2023-06-16	3.0	4.50	4.53	4.55	4.51	29935.0	1721.
35	TSLA	259.44	Put	255.00	2023-06-23	10.0	7.65	7.70	7.75	7.75	7470.0	464.
39	TSLA	259.44	Call	262.50	2023-06-23	10.0	7.90	7.95	8.00	7.90	3855.0	258.
42	TSLA	259.44	Call	295.00	2023-06-30	17.0	3.30	3.33	3.35	3.35	10563.0	757.
43	TSLA	259.44	Put	252.50	2023-06-16	3.0	3.45	3.50	3.55	3.48	16907.0	1226.

In [60]:

```
tesla['Exp Date'].unique()
```

Out[60]:

```
array(['2023-06-16', '2023-06-23', '2023-06-30', '2023-11-17',
       '2023-07-07', '2023-07-21', '2023-07-28', '2023-08-18',
       '2023-07-14', '2023-09-15'], dtype=object)
```

In [61]:

```
tesla['Open Int'].unique()
```

Out[61]:

```
array([  354.,   524.,   133.,   950.,   487.,  1721.,   464.,   258.,
         757.,  1226.,   465.,   113.,   135.,   611.,  3326.,  1655.,
         361.,  1442.,   134.,  3006., 10112., 22574., 16269.,   427.,
        2361.,   160.,   311.,  3122.,  2247.,   240.,   507.,  4783.,
         428.,  1588.,  2611.,   744., 11908.,  8172.,   203.,   353.,
         555.,   227.,   270.,   462.,  1948.,   696.,   808.,   184.,
        3260.,   211.,   594.,  3999.,  1446.,  1487.,   288.,   467.,
        2628.,  1725.,  2331., 15645.,   420.,  4752.,  2989., 10942.,
        8458.,  4150.,  1018.,   268.,   226.,   424.,  8943.,   381.,
         350.,  3628.,   834.,  2390.,  2580.,  1275.,   917.,  2534.,
         468.,   847., 34069.,   809.,  1333.,  8842.,   548.,  1146.,
         431.,   851.,  3733.,   533.,   632.,  1170., 25171.,   395.,
         451.])
```

In [62]:

```
class OptionPayoffCalculator:
    def __init__(self, st):
        self.st = st

    def calculate_call_payoff(self, strike_price, ask_long_call):
        return np.where(self.st > strike_price, self.st - strike_price, 0) -
    def calculate_put_payoff(self, strike_price, ask_long_put):
        return np.where(self.st < strike_price, strike_price - self.st, 0) -

class OptionPayoffPlotter:
    def __init__(self, st, payoff):
        self.st = st
        self.payoff = payoff

    def _set_ylim(self):
        max_payoff = max(self.payoff)
        min_payoff = min(self.payoff)
        ymax = max(abs(max_payoff), abs(min_payoff)) + 5
        self.ax.set_ylim(min_payoff - 500, max_payoff + 500)

    def plot(self):
        self.ax = plt.style.use('dark_background')
        self.fig, self.ax = plt.subplots(figsize=(15, 8))
        self.ax.spines['top'].set_visible(False)
        self.ax.spines['right'].set_visible(False)
        self.ax.spines['bottom'].set_position('zero')
        self.ax.grid(color='gray', linestyle='--', linewidth=0.5)
        self.ax.set_xlabel('Stock Price')
        self.ax.set_ylabel('Profit and Loss')
        self.ax.margins(x=0)

        self.ax.plot(self.st, self.payoff)
        self.ax.fill_between(self.st, 0, self.payoff, where=(self.payoff >=
                             interpolate=True, alpha=0.3, color='green')
        self.ax.fill_between(self.st, 0, self.payoff, where=(self.payoff < 0
                             interpolate=True, alpha=0.3, color='red')

        self._set_ylim()

        plt.subplots_adjust(left=0.07, right=0.95, bottom=0.07, top=0.95)
        plt.show()

    def print_final_profit(self):
        final_profit = self.payoff[-1]
        print(f"Final Profit/Loss: {final_profit}")
```

In [63]:

```
tesla_june_16 = tesla[tesla['Exp Date'] == '2023-06-16']
```

In [64]:

```
tesla_june_16
```

Out[64]:

	Symbol	Price	Type	Strike	Exp Date	DTE	Bid	Midpoint	Ask	Last	Volume
7	TSLA	259.44	Put	257.50	2023-06-16	3.0	5.65	5.70	5.75	5.75	18850.0
12	TSLA	259.44	Put	253.33	2023-06-16	3.0	3.80	3.83	3.85	3.85	19470.0
30	TSLA	259.44	Put	255.00	2023-06-16	3.0	4.50	4.53	4.55	4.51	29935.0
43	TSLA	259.44	Put	252.50	2023-06-16	3.0	3.45	3.50	3.55	3.48	16907.0
59	TSLA	259.44	Call	257.50	2023-06-16	3.0	6.65	6.70	6.75	6.70	38889.0
71	TSLA	259.44	Put	260.00	2023-06-16	3.0	7.00	7.05	7.10	7.08	17143.0
164	TSLA	259.44	Call	253.33	2023-06-16	3.0	8.95	9.00	9.05	8.95	18525.0
177	TSLA	259.44	Put	250.00	2023-06-16	3.0	2.68	2.69	2.70	2.68	60494.0
185	TSLA	259.44	Call	260.00	2023-06-16	3.0	5.50	5.55	5.60	5.54	130663.0
189	TSLA	259.44	Call	255.00	2023-06-16	3.0	7.95	8.00	8.05	8.02	92347.0
217	TSLA	259.44	Call	262.50	2023-06-16	3.0	4.55	4.57	4.60	4.62	16383.0
236	TSLA	259.44	Put	267.50	2023-06-16	3.0	12.05	12.13	12.20	12.12	662.0
237	TSLA	259.44	Call	297.50	2023-06-16	3.0	0.32	0.33	0.33	0.39	2517.0
283	TSLA	259.44	Call	267.50	2023-06-16	3.0	3.05	3.10	3.15	3.10	6933.0
297	TSLA	259.44	Call	265.00	2023-06-16	3.0	3.75	3.78	3.80	3.82	50340.0
319	TSLA	259.44	Put	245.00	2023-06-16	3.0	1.50	1.51	1.52	1.50	33228.0
338	TSLA	259.44	Put	277.50	2023-06-16	3.0	20.35	20.45	20.55	20.25	798.0
340	TSLA	259.44	Put	262.50	2023-06-16	3.0	8.55	8.60	8.65	8.59	2175.0
364	TSLA	259.44	Put	233.33	2023-06-16	3.0	0.37	0.38	0.38	0.38	7127.0
418	TSLA	259.44	Put	247.50	2023-06-16	3.0	2.01	2.03	2.04	2.02	13401.0
474	TSLA	259.44	Put	243.33	2023-06-16	3.0	1.23	1.23	1.24	1.23	7880.0
507	TSLA	259.44	Put	240.00	2023-06-16	3.0	0.82	0.83	0.83	0.82	45248.0
510	TSLA	259.44	Put	275.00	2023-06-16	3.0	18.15	18.23	18.30	18.45	1205.0
537	TSLA	259.44	Put	237.50	2023-	3.0	0.60	0.61	0.62	0.60	13092.0

	Symbol	Price	Type	Strike	Exp Date	DTE	Bid	Midpoint	Ask	Last	Volume
					06-16						
568	TSLA	259.44	Call	280.00	2023-06-16	3.0	1.17	1.17	1.18	1.17	28924.0
574	TSLA	259.44	Call	252.50	2023-06-16	3.0	9.45	9.50	9.55	9.45	22137.0
596	TSLA	259.44	Put	242.50	2023-06-16	3.0	1.11	1.12	1.13	1.12	10552.0
681	TSLA	259.44	Put	280.00	2023-06-16	3.0	22.60	22.70	22.80	22.75	759.0
725	TSLA	259.44	Put	246.67	2023-06-16	3.0	1.83	1.84	1.85	1.85	4849.0
767	TSLA	259.44	Put	265.00	2023-06-16	3.0	10.20	10.27	10.35	10.03	2454.0
807	TSLA	259.44	Call	270.00	2023-06-16	3.0	2.53	2.55	2.56	2.55	62355.0
841	TSLA	259.44	Put	235.00	2023-	3.0	0.45	0.46	0.46	0.45	15532.0

In [65]:
```
tesla_june_16_put = tesla_june_16[tesla_june_16['Type']=='Put']
tesla_june_16_call = tesla_june_16[tesla_june_16['Type']=='Call']
```

TESLA 16 JUNE PUT

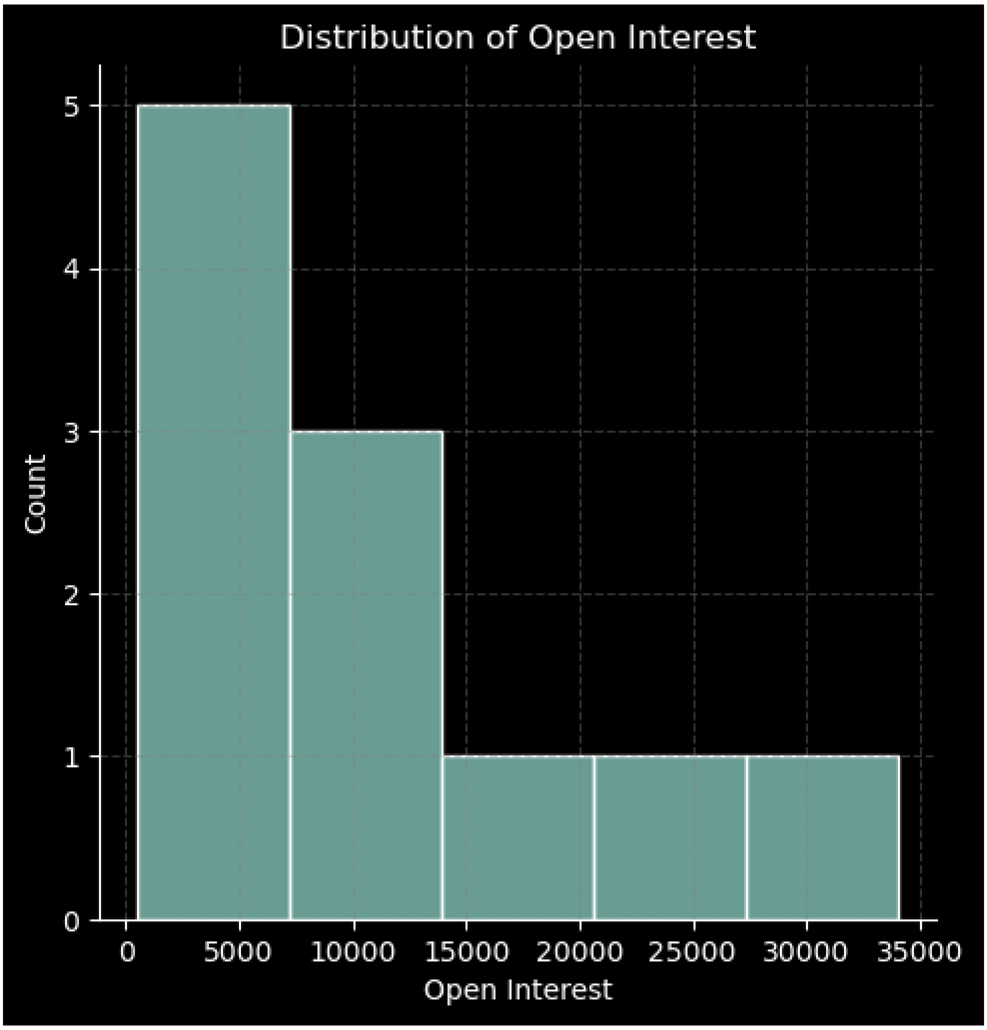
In [66]:
```
tesla_june_16_put.sort_values(by='Strike')
```

Out[66]:

	Symbol	Price	Type	Strike	Exp Date	DTE	Bid	Midpoint	Ask	Last	Volume
364	TSLA	259.44	Put	233.33	2023-06-16	3.0	0.37	0.38	0.38	0.38	7127.0
841	TSLA	259.44	Put	235.00	2023-06-16	3.0	0.45	0.46	0.46	0.45	15533.0
537	TSLA	259.44	Put	237.50	2023-06-16	3.0	0.60	0.61	0.62	0.60	13092.0
507	TSLA	259.44	Put	240.00	2023-06-16	3.0	0.82	0.83	0.83	0.82	45248.0
596	TSLA	259.44	Put	242.50	2023-06-16	3.0	1.11	1.12	1.13	1.12	10552.0
474	TSLA	259.44	Put	243.33	2023-06-16	3.0	1.23	1.23	1.24	1.23	7880.0
319	TSLA	259.44	Put	245.00	2023-06-16	3.0	1.50	1.51	1.52	1.50	33228.0
725	TSLA	259.44	Put	246.67	2023-06-16	3.0	1.83	1.84	1.85	1.85	4849.0
418	TSLA	259.44	Put	247.50	2023-06-16	3.0	2.01	2.03	2.04	2.02	13401.0
177	TSLA	259.44	Put	250.00	2023-06-16	3.0	2.68	2.69	2.70	2.68	60494.0
43	TSLA	259.44	Put	252.50	2023-06-16	3.0	3.45	3.50	3.55	3.48	16907.0
12	TSLA	259.44	Put	253.33	2023-06-16	3.0	3.80	3.83	3.85	3.85	19470.0
30	TSLA	259.44	Put	255.00	2023-06-16	3.0	4.50	4.53	4.55	4.51	29935.0
7	TSLA	259.44	Put	257.50	2023-06-16	3.0	5.65	5.70	5.75	5.75	18850.0
71	TSLA	259.44	Put	260.00	2023-06-16	3.0	7.00	7.05	7.10	7.08	17143.0
340	TSLA	259.44	Put	262.50	2023-06-16	3.0	8.55	8.60	8.65	8.59	2175.0
767	TSLA	259.44	Put	265.00	2023-06-16	3.0	10.20	10.27	10.35	10.03	2454.0
236	TSLA	259.44	Put	267.50	2023-06-16	3.0	12.05	12.13	12.20	12.12	662.0
510	TSLA	259.44	Put	275.00	2023-06-16	3.0	18.15	18.23	18.30	18.45	1205.0
338	TSLA	259.44	Put	277.50	2023-06-16	3.0	20.35	20.45	20.55	20.25	798.0
681	TSLA	259.44	Put	280.00	2023-06-16	3.0	22.60	22.70	22.80	22.75	759.0

In [67]:

```
sns.displot(tesla_june_16_call['Open Int'])
plt.grid(linestyle="--", color="grey", alpha=0.4)
plt.xlabel('Open Interest')
plt.ylabel('Count')
plt.title('Distribution of Open Interest')
plt.show()
```



TESLA 16 JUN CALL

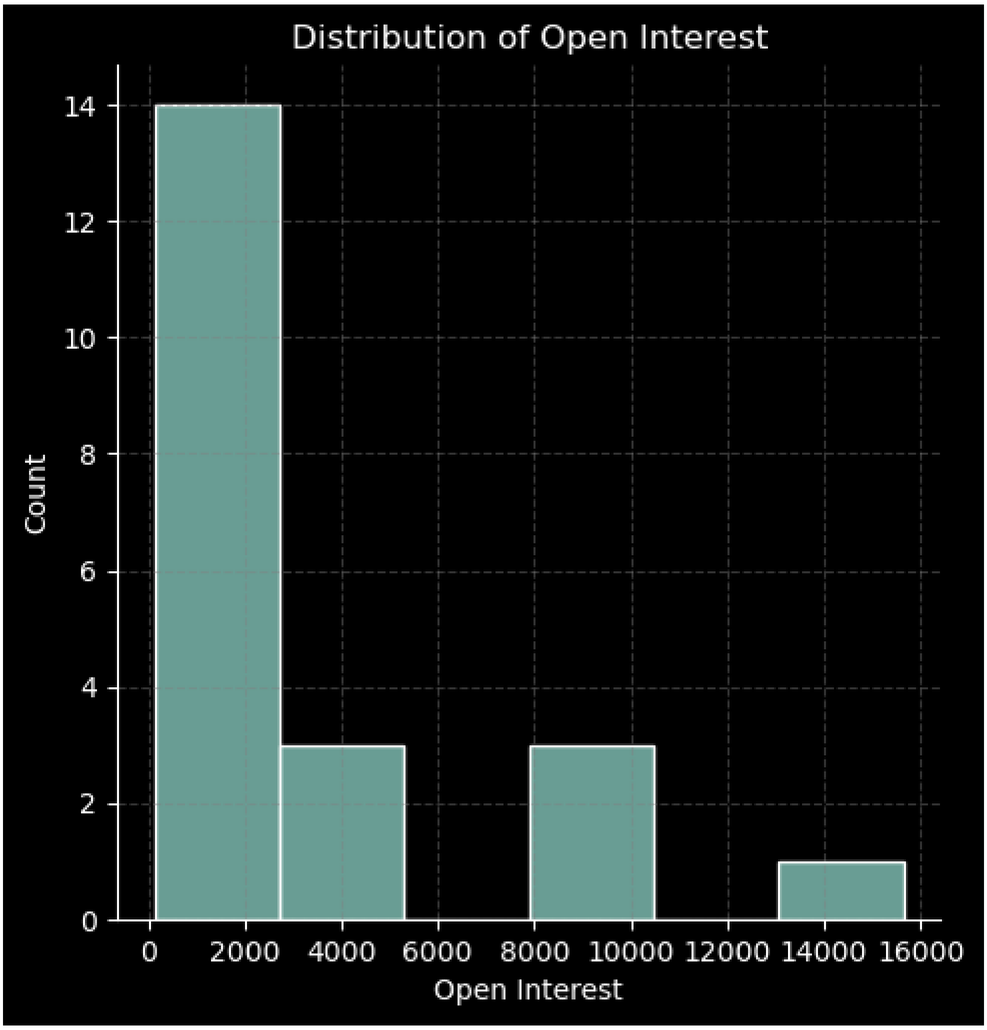
```
In [68]: tesla_june_16_call.sort_values(by='Strike')
```

Out[68]:

	Symbol	Price	Type	Strike	Exp Date	DTE	Bid	Midpoint	Ask	Last	Volume	O
574	TSLA	259.44	Call	252.50	2023-06-16	3.0	9.45	9.50	9.55	9.45	22137.0	84
164	TSLA	259.44	Call	253.33	2023-06-16	3.0	8.95	9.00	9.05	8.95	18525.0	30
189	TSLA	259.44	Call	255.00	2023-06-16	3.0	7.95	8.00	8.05	8.02	92347.0	162
59	TSLA	259.44	Call	257.50	2023-06-16	3.0	6.65	6.70	6.75	6.70	38889.0	33
185	TSLA	259.44	Call	260.00	2023-06-16	3.0	5.50	5.55	5.60	5.54	130663.0	225
217	TSLA	259.44	Call	262.50	2023-06-16	3.0	4.55	4.57	4.60	4.62	16383.0	31
297	TSLA	259.44	Call	265.00	2023-06-16	3.0	3.75	3.78	3.80	3.82	50340.0	119
283	TSLA	259.44	Call	267.50	2023-06-16	3.0	3.05	3.10	3.15	3.10	6933.0	15
807	TSLA	259.44	Call	270.00	2023-06-16	3.0	2.53	2.55	2.56	2.55	62355.0	340
568	TSLA	259.44	Call	280.00	2023-06-16	3.0	1.17	1.17	1.18	1.17	28924.0	109
237	TSLA	259.44	Call	297.50	2023-06-16	3.0	0.32	0.33	0.33	0.39	2517.0	5

In [69]:

```
sns.displot(tesla_june_16_put['Open Int'])
plt.grid(linestyle="--", color="grey", alpha=0.4)
plt.xlabel('Open Interest')
plt.ylabel('Count')
plt.title('Distribution of Open Interest')
plt.show()
```



LONG STRADDLE

```
In [70]: tesla = yf.Ticker("TSLA").history(period="1d", interval="15m")
```

```
In [71]: tesla
```

Out[71]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Datetime							
2023-06-13 09:30:00-04:00	253.949997	255.830002	253.199997	254.285004	19495258	0.0	0.0
2023-06-13 09:45:00-04:00	254.270004	254.649994	252.770004	253.190002	8691268	0.0	0.0
2023-06-13 10:00:00-04:00	253.229996	255.589996	251.830002	252.921204	9124192	0.0	0.0
2023-06-13 10:15:00-04:00	252.940002	253.490005	251.399994	252.199997	7492686	0.0	0.0
2023-06-13 10:30:00-04:00	252.179993	252.720001	251.460007	251.725204	5629981	0.0	0.0
2023-06-13 10:45:00-04:00	251.750000	253.520004	251.339996	253.330093	6612857	0.0	0.0
2023-06-13 11:00:00-04:00	253.350006	254.809998	253.240005	254.655197	6255336	0.0	0.0
2023-06-13 11:15:00-04:00	254.668198	254.970001	253.679993	254.621902	5663999	0.0	0.0
2023-06-13 11:30:00-04:00	254.634995	254.759995	253.779999	254.429993	4124501	0.0	0.0
2023-06-13 11:45:00-04:00	254.462296	257.799988	254.399994	256.069885	9994980	0.0	0.0
2023-06-13 12:00:00-04:00	256.067291	258.540009	255.845001	258.100006	7073384	0.0	0.0
2023-06-13 12:15:00-04:00	258.070007	258.970001	257.810089	258.619995	6116182	0.0	0.0
2023-06-13 12:30:00-04:00	258.649994	259.190002	257.950012	258.029999	5320554	0.0	0.0
2023-06-13	258.037903	258.700012	257.471008	258.329987	3896337	0.0	0.0

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Datetime							
12:45:00-04:00							
2023-06-13 13:00:00-04:00	258.339996	258.950012	257.619995	257.929901	3648977	0.0	0.0
2023-06-13 13:15:00-04:00	257.910095	258.779999	257.700012	258.739990	3113637	0.0	0.0
2023-06-13 13:30:00-04:00	258.760010	259.429993	258.429993	259.234985	4475358	0.0	0.0
2023-06-13 13:45:00-04:00	259.250000	259.679993	259.000000	259.220001	3856138	0.0	0.0
2023-06-13 14:00:00-04:00	259.228485	259.239990	256.600006	257.845001	6122404	0.0	0.0
2023-06-13 14:15:00-04:00	257.859009	257.880005	255.630005	255.919998	4656210	0.0	0.0
2023-06-13 14:30:00-04:00	255.955002	256.809998	255.699997	256.600006	4464609	0.0	0.0
2023-06-13 14:45:00-04:00	256.595001	258.450012	256.220001	258.010010	4369555	0.0	0.0
2023-06-13 15:00:00-04:00	258.000092	258.668915	257.660004	258.489899	3454740	0.0	0.0
2023-06-13 15:15:00-04:00	258.480103	259.380005	258.160004	258.244995	3753801	0.0	0.0
2023-06-13 15:30:00-	258.260010	258.489807	258.141602	258.269989	415018	0.0	0.0

```
In [72]: tesla_june_16_put.sort_values(by='Strike').iloc[20]
```

Out[72]:

```
Symbol              TSLA
Price             259.44
Type                 Put
Strike             280.0
Exp Date      2023-06-16
DTE                  3.0
Bid                 22.6
Midpoint            22.7
Ask                 22.8
Last               22.75
Volume             759.0
Open Int           350.0
Vol/OI              2.17
IV                76.51%
Delta          -0.870582
Time            14:03 ET
Name: 681, dtype: object
```

In [73]:

```
tesla_june_16_call.sort_values(by='Strike').iloc[9]
```

Out[73]:

```
Symbol              TSLA
Price             259.44
Type                Call
Strike             280.0
Exp Date      2023-06-16
DTE                  3.0
Bid                 1.17
Midpoint            1.17
Ask                 1.18
Last                1.17
Volume           28924.0
Open Int         10942.0
Vol/OI              2.64
IV                77.07%
Delta           0.132715
Time            14:05 ET
Name: 568, dtype: object
```

In [84]:

```
current_market_price = 258
print(current_market_price)

# Long put
strike_price_long_put = 280

# Premium long put
ask_long_put = 22.8

# Long call
strike_price_long_call = 280
# Premium long call
ask_long_call = 1.18

# Range at expiration of the put
st = np.arange(0, 2 * current_market_price, 1)
```

```
258
```

In [85]:

```
calculator = OptionPayoffCalculator(st)
calculator
```

Out[85]:

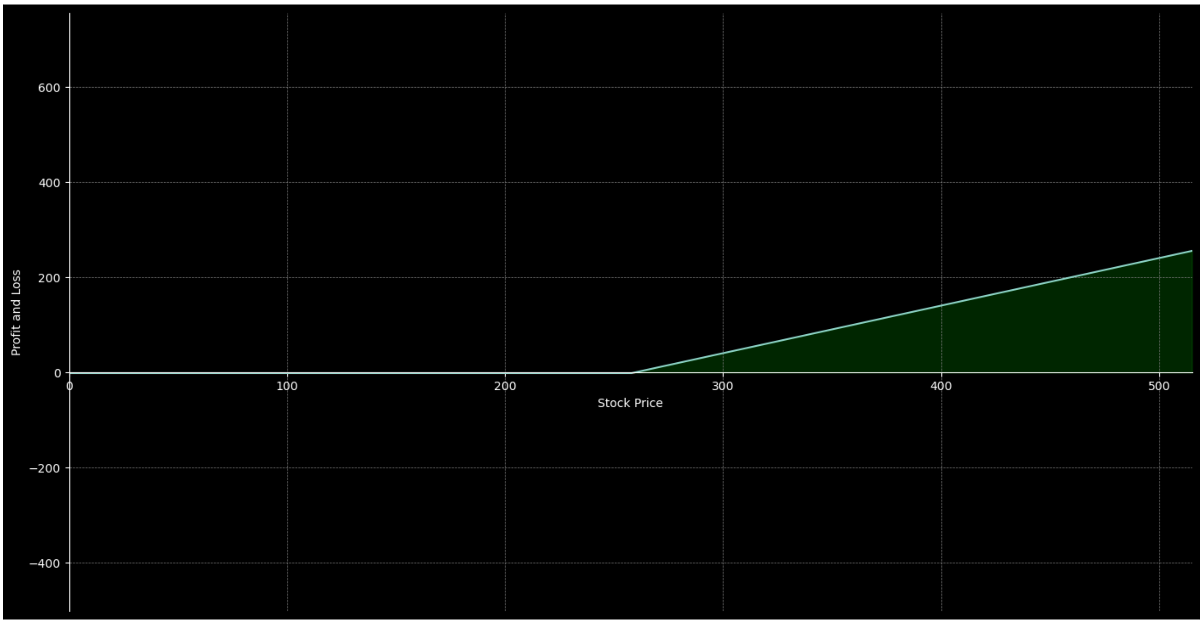
```
<__main__.OptionPayoffCalculator at 0x7fd148d783a0>
```

EXAMPLE 1

LONG CALL

```
In [86]: payoff_long_call = calculator.calculate_call_payoff(current_market_price, as
         plotter = OptionPayoffPlotter(st, payoff_long_call)
```

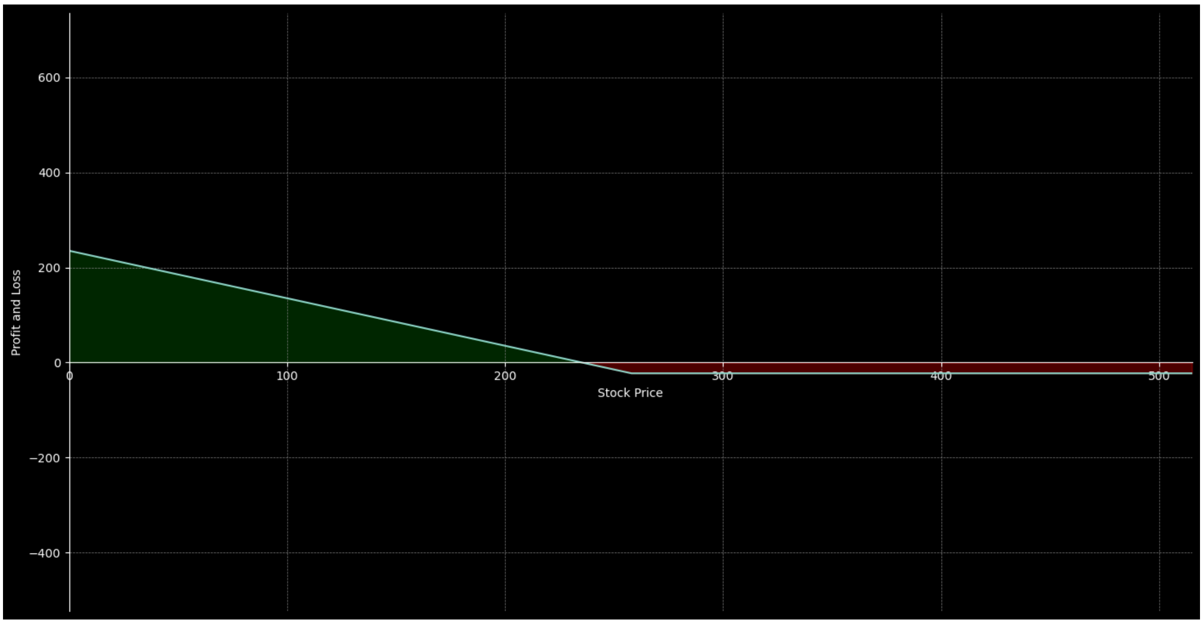
```
In [87]: plotter.plot()
         plotter.print_final_profit()
```



```
Final Profit/Loss: 255.82
```

LONG PUT

```
In [88]: payoff_long_put = calculator.calculate_put_payoff(current_market_price , ask
         plotter = OptionPayoffPlotter(st, payoff_long_put)
         plotter.plot()
         plotter.print_final_profit()
```



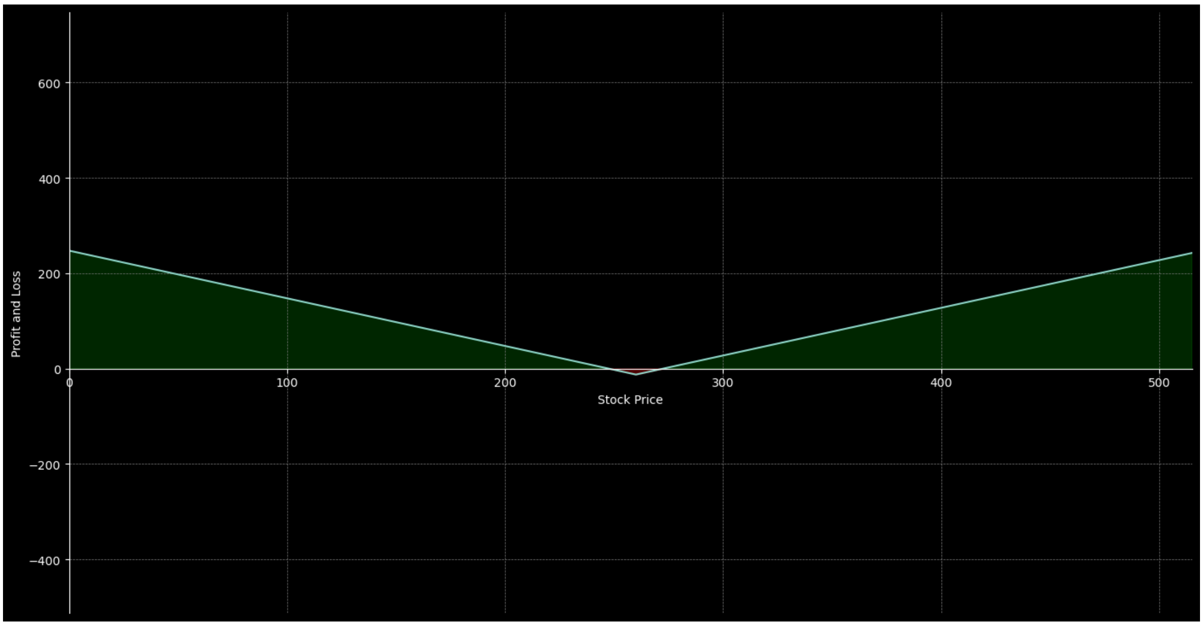
```
Final Profit/Loss: -22.8
```

EXAMPLE 2

In [80]:
```
calculator = OptionPayoffCalculator(st)

strike_price = 260.00
ask_long_call = 5.60

call_payoff = calculator.calculate_call_payoff(strike_price, ask_long_call)

strike_price = 260.00
ask_long_put = 7.10

put_payoff = calculator.calculate_put_payoff(strike_price, ask_long_put)
```

In [81]:
```
plotter = OptionPayoffPlotter(st, call_payoff + put_payoff)
```

In [82]:
```
plotter.plot()
```



In [83]:
```
plotter.print_final_profit()
```

```
Final Profit/Loss: 242.3
```

EXAMPLE 3

In [99]:
```
current_market_price = 258
print(current_market_price)

# Long put
strike_price_long_put = 280
ask_long_put = 22.8

# Long call
strike_price_long_call = 280
ask_long_call = 1.18

# Range at expiration
st = np.arange(0, 2 * current_market_price, 1)
```

```
258
```

In [100…

```
#long put profit/loss
put_payoff = np.where(strike_price_long_put - st > 0, strike_price_long_put
```

In [101…

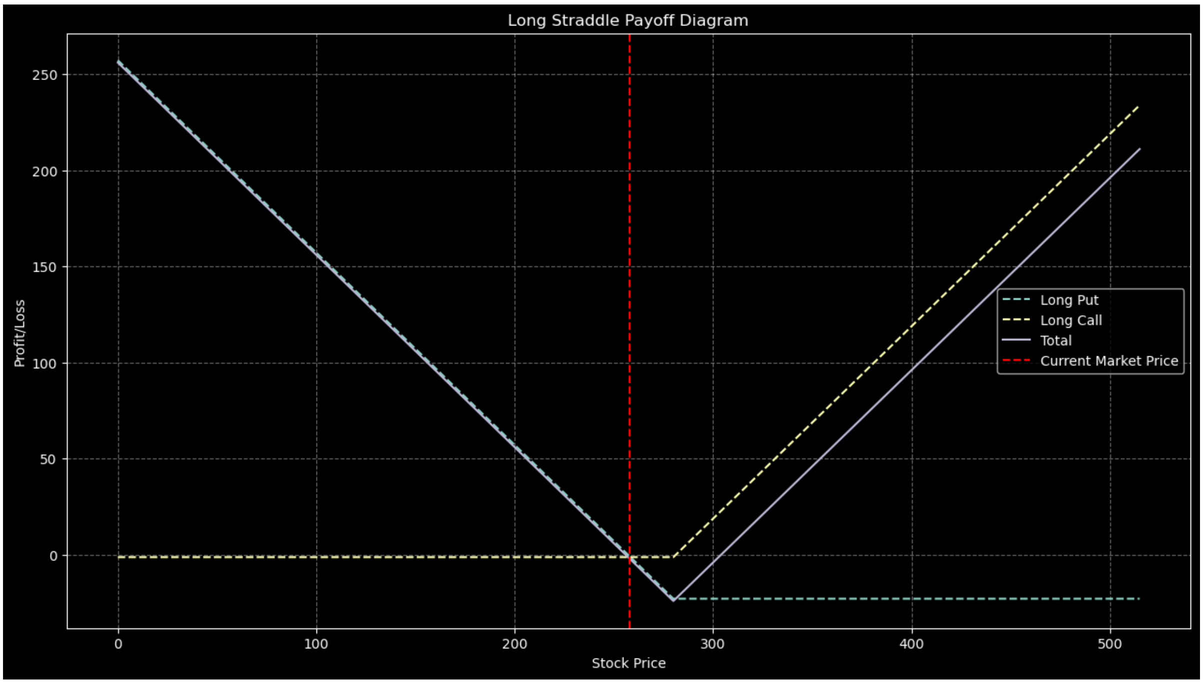
```
#long call profit/loss
call_payoff = np.where(st - strike_price_long_call > 0, st - strike_price_lc
```

In [102…

```
#total profit/loss (put + call)
total_payoff = put_payoff + call_payoff
```

In [105…

```
# Plot the payoff diagram
plt.figure(figsize=(15, 8))
plt.plot(st, put_payoff, label='Long Put', linestyle="--")
plt.plot(st, call_payoff, label='Long Call', linestyle="--")
plt.plot(st, total_payoff, label='Total')

# Add a vertical line for the current market price
plt.axvline(x=current_market_price, color='r', linestyle='--', label='Curren

plt.xlabel('Stock Price')
plt.ylabel('Profit/Loss')
plt.legend()
plt.title('Long Straddle Payoff Diagram')
plt.grid(True, linestyle='--', alpha=0.4)
plt.show()
```


Long Straddle Payoff Diagram
Profit/Loss
Stock Price
Long Put
Long Call
Total
Current Market Price