**Computer Assignment #3:**

**Option Trading Strategies: Construction, Pricing and Simulation**

Group 4

Beier Chen

Chunlei Meng

Yibing Xie

Yi Zhu

# Executive Summary

This report constructs, prices and simulates six option trading strategies on Exxon Mobil Corporation stock. We first simulate the price path of the stock, then we construct six trading strategies including bull call spread, bear put spread, butterfly spread, straddle, strip and collar. We calculate the expected payoff of each strategy under both physical measure and risk-neutral measure, and analyze the payoff of each strategy by plotting the payoff diagram, analyze the profit and loss, and discuss when to employ each strategy.

# Introduction

Investors can use different trading strategies when they have different preferences on risk and return. For example, when investors are moderately bullish about the underlying asset, they can use the bull spread strategy. When they think the underlying asset price is not volatile, they can use butterfly spread strategy. This report analyze six option trading strategies and compare them.

Ito’s Lemma tells us  
Therefore we can simulate stock path by

Specifically, we simulate the stock price path from 11/06/2015 to expiration and then calculate the expected payoff under physical measure and risk-neutral measure for each strategy, and compare the simulated price of the strategy and the Bloomberg price. Then we would plot the payoff diagram for each strategy and analyze the maximum profit, maximum loss and breakeven point. For each strategy, we also discuss when we would use the strategy.

# Data

We choose the common stock of Exxon Mobil Corporation (XOM) as the underlying assets and the options that will expires on Jan 15, 2016 to implement these strategies. The close price on Nov 6, 2015 was 84.475. We use 3-month LIBOR as the risk-free rate, which is 0.3414% annually, and divide it by 252 to give us approximate daily risk-free rate, which will be used in the CAPM later.

In this report, we compute the drift rate based on the following three rates: 1) arithmetic mean of historical log return; 2) estimated rate of return using CAPM regression; 3) annual risk-free rate. We use the weighted average implied volatility as the diffusion, which is 0.2141 according to Bloomberg’s option monitor screen.

The sample period we choose for the first two methods is 05/06/2015 – 11/06/2015, which covers 129 trading days. We can also recover the corresponding estimated volatility from those two methods. The results are summarized as follows:

Arithmetic Mean: Since , we have

For the CAPM, we use S&P 500 index as the market index, and run OLS of the following form:

where , .

Note that the estimation given by those two methods differs quite a bit, though their volatility reconciles and is close to the implied volatility. It may come as surprise to see the distinction between two methods, however, it makes sense by observing Graph 1 in Appendix.

# Methodology

The bull call strategy involves buying one call at low strike and selling one call at higher strike, and both call options have the same expiration and underlying asset. This strategy is used when investors expect a moderate rise of the underlying asset. The bear put strategy is similar and involves buying one put at higher strike and selling one put at lower strike, and investors use this strategy when they expect that the underlying asset price will decrease. Both strategies will limit gains and losses. The butterfly spread is the combination of a bull spread strategy and a bear spread strategy and use three different strike prices.

The straddle strategy is performed by buying one call and one put with same strike price and expiration, and the strategy is used when the investor expects a significant move on the underlying asset price. It will limit the loss while the gain is unlimited. The strip strategy is similar but requires buying two puts and one call, and will have steeper payoff when the underlying asset price is lower than the strike price. The collar strategy is to buy one out-of-the-money put and sell one out-of-the-money call and long the underlying asset, and enables investors to lock in profit without selling the stocks.

# Results and Analysis

In this report, we simulate the stock price path 10,000 times, and the results are showed in Graph 2 in Appendix. And the expected future stock price given different expected rates of return are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Arithmetic Mean  (-4.99%) | CAPM Regression  (1.87%) | Annual risk-free rate  (0.3414% ) |
| ***Expected Rate of Return*** | 83.53 | 84.74 | 84.47 |

To measure the expected payoff, we only use CAPM as the physical measure, and the expected payoff given different strategies are as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Strategy*** | Bull Spread | Bear Spread | Butterfly Spread | Straddle | Strip | Collar |
| ***Expected Payoff*** | 4.76 | 5.23 | 1.17 | 6.45 | 9.94 | 84.69 |

The discounted payoff under the risk-neutral measure for each strategy is summarized as follows along with the corresponding figure in the Appendix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Strategy*** | Bull Spread | Bear Spread | Butterfly Spread | Straddle | Strip | Collar |
| ***Expected Payoff*** | 4.57 | 5.33 | 1.21 | 6.48 | 9.83 | 84.57 |
| ***Bloomberg***  ***Price*** | 4.83 | 6.02 | 1.85 | 6.11 | 9.86 | 85.64 |

The compositions of each strategy and the price of each option are shown Table 1 in Appendix. These options are European options. When we long an option, the bid price is used. When we short an option, the ask price is used. This is because the market makers rely on bid-ask spread to make a profit, so the price at which we buy an asset is always higher than the price at which we sell the asset.

The payoff diagram of each strategy is showed in Appendix. The maximum profit, maximum loss and breakeven point(s) are noted in the P&L diagram of each strategy.

The bull call spread strategy is implemented by buying a call option with a relative low strike price () and selling a call option with a higher strike price (). In our case, is 80 and is 90. We can see from the diagram that we still benefit from the underlying asset price going up but the upside is limited.

We want to employ a bull call spread strategy when we think that the price of the underlying asset will go up moderately in the future, i.e., we believe the underlying asset price will increase but not beyond a certain level (). In this case, we are unwilling to pay for the part of the premium of the call option with strike price that protects us against the risk of the underlying asset price going up beyond, so we write a call option with strike price, which has little chance to be exercised, to partly offset the cost. Since this strategy doesn’t cover the risk of underlying asset price going up beyonda certain level, it is riskiest when the underlying asset price is very likely to go up beyond.

The bear put spread strategy is just the opposite of the bull call spread strategy. This strategy involves buying a put option with a relative high strike () price and selling a put option with a lower strike price (). In our case, is 90 and is 80. We can see from the diagram in the Appendix that we benefit from the underlying asset price going down but the upside is limited.

We want to employ a bear put spread strategy when we think that the price of the underlying asset will go down moderately in the future, i.e., we believe the underlying asset price will decrease but not below a certain level (). We are unwilling to pay for the part of the premium of the put option with strike price that protects us against the risk of the underlying asset price going down below, so we write a put option with strike price to partly offset the cost. Since this strategy doesn’t cover the risk of underlying asset price going down belowa certain level, it is riskiest when the underlying asset price is very likely to go down below.

The butterfly spread strategy can be implemented by selling 2 ATM call options, buy 1 ITM call option and 1 OTM call option. In our case, the butterfly spread strategy involves selling 2 call options with strike price 85, buying 1 call option with strike price 80 and buying 1 call option with strike price 90. This strategy can be thought of as a combination of bull call strategy and bear put strategy, because the payoff diagram of bear put strategy is similar to the payoff diagram of a portfolio consisted by buying 1 OTM call and 1 ATM call. We can see from the diagram below that in this strategy we benefit from the underlying asset price not moving from the current level.

Since this is a short-volatility strategy, we employ it when we believe the underlying asset price will remain in the narrow range around the current level. This strategy doesn’t cover the risk of underlying asset price deviating much from the current level, so it is riskiest when the underlying asset price is very volatile and is very likely to go up beyond the strike price of the OTM call option and go down below the strike price of the ITM call option.

The straddle strategy is the opposite of the butterfly spread strategy and is constructed by buying 1 ATM call option and 1 ATM put option. In our case, we buy 1 call option and 1 put option and both have a strike price of 85. From the payoff diagram below we can see that in this strategy we benefit from the underlying asset price deviating form the current level.

This is a long-volatility strategy, we use this strategy when we think the underlying asset price will change much in either direction. This strategy is riskiest if the underlying asset price is least volatile and tends to remain in a narrow range around the current level.

The strip strategy is similar to the straddle strategy except that strip is more bearish and is consisted of 1 ATM call option and 2 ATM put options. We construct this strategy by buying 1 call option and 2 put options and all have a strike price of 85.

The collar strategy is very much like the bull call strategy. We construct this strategy by buying 1 put option with strike price 80, selling 1 call option with strike price 90 and buying 1 underlying asset.

# Conclusions

The payoff diagrams show different patterns for different strategies, and we would like to use different strategies in different circumstances. For the bull call spread strategy, we would like to use it when we think that the price of the underlying asset will go up moderately in the future. When the price of the underlying asset is expected to go down moderately in the future, we want to employ a bear put spread strategy. When the market is believed to be stable, we would use butterfly spread strategy. And we would like to use straddle strategy when we think the underlying asset price will change much in either direction.

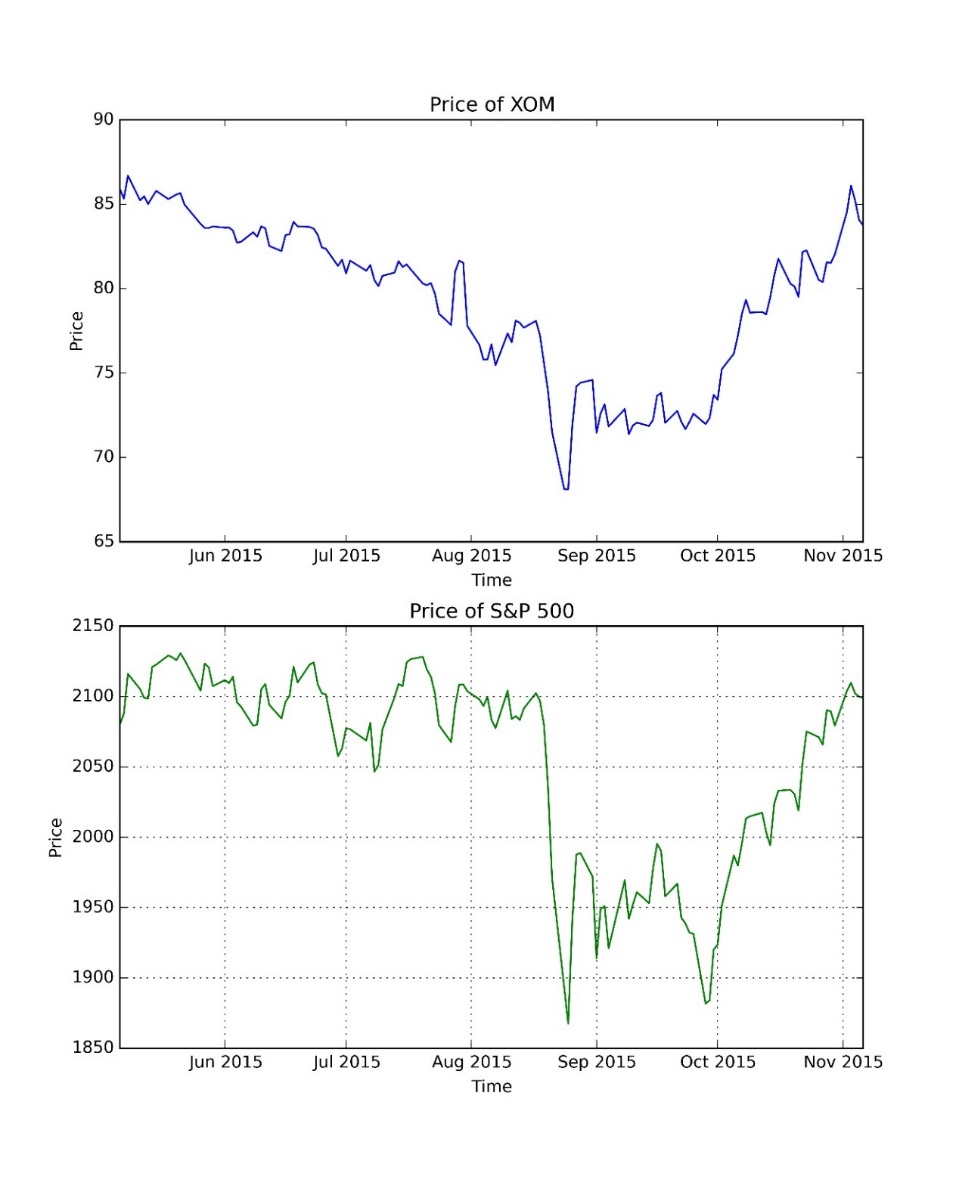
# Appendix

## **Tables and Graphs**

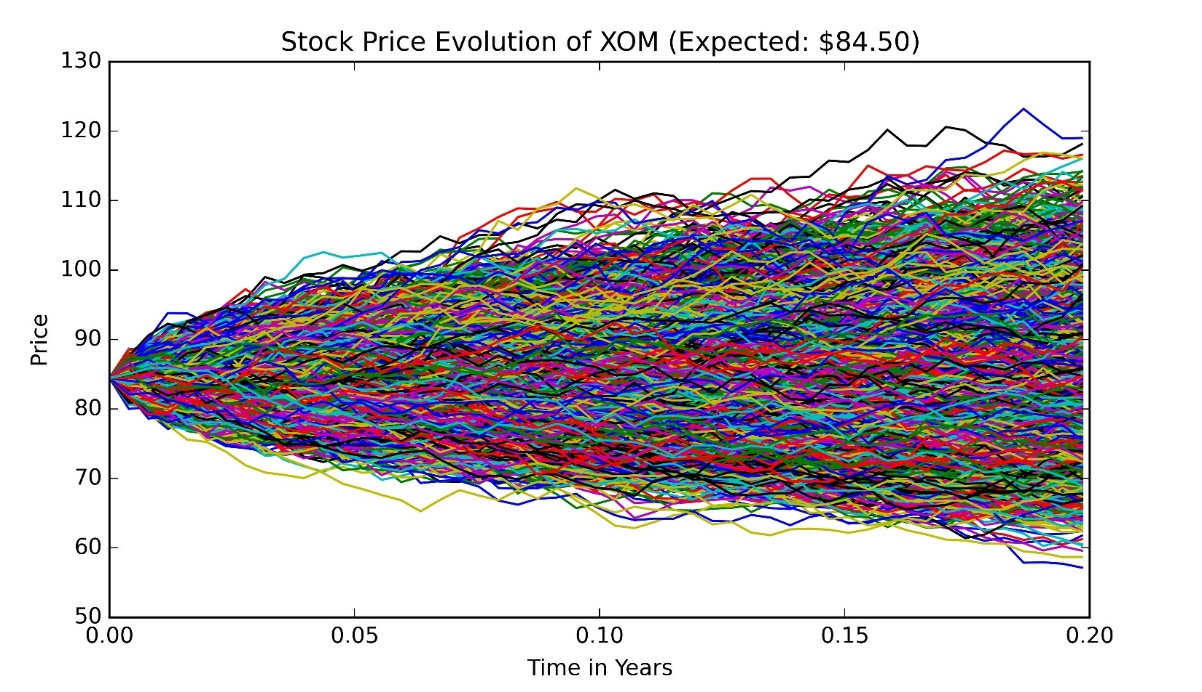
Table 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **A** | **B** | **C** | **D** | **E** | **F** |
| Call @80 | bid | 5.45 | 1 |  | 1 |  |  |  |
| ask | 5.2 |  |  |  |  |  |  |
| Call @85 | bid | 2.36 |  |  |  | 1 | 1 |  |
| ask | 2.16 |  |  | 2 |  |  |  |
| Call @90 | bid | 0.72 |  |  | 1 |  |  |  |
| ask | 0.62 | 1 |  |  |  |  | 1 |
| Put @80 | bid | 1.78 |  |  |  |  |  | 1 |
| ask | 1.68 |  | 1 |  |  |  |  |
| Put @85 | bid | 3.75 |  |  |  | 1 | 2 |  |
| ask | 3.55 |  |  |  |  |  |  |
| Put @90 | bid | 7.7 |  | 1 |  |  |  |  |
| ask | 6.85 |  |  |  |  |  |  |
| Spot | 84.475 | |  |  |  |  |  | 1 |
| **strategy price** |  |  | **4.83** | **6.02** | **1.85** | **6.11** | **9.86** | **85.64** |

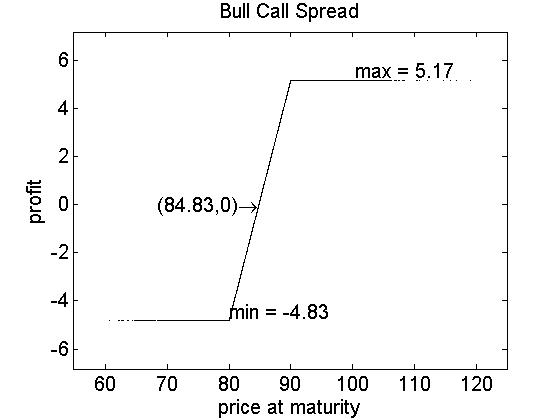
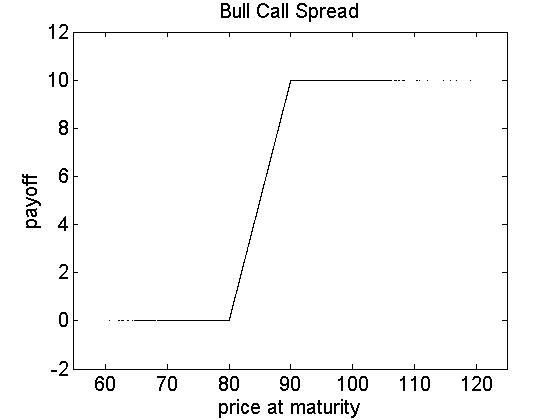
Graph 1: Price of the Underlying Asset and S&P 500

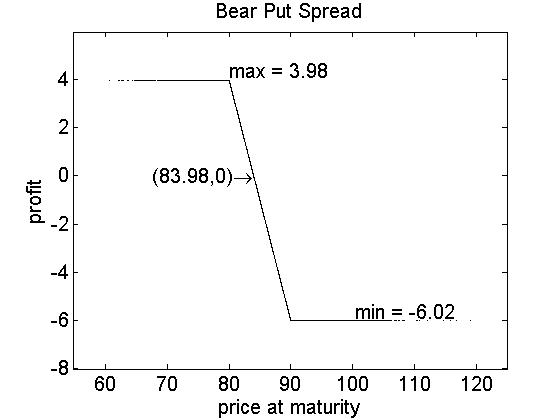
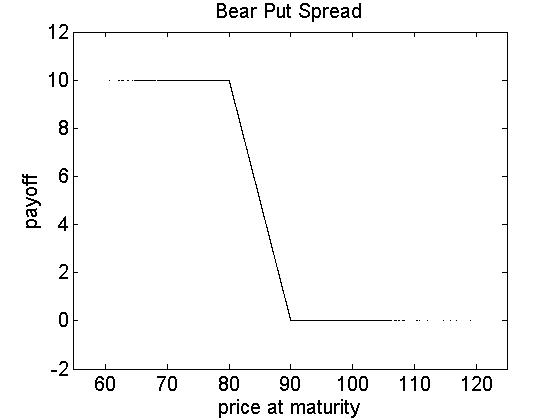


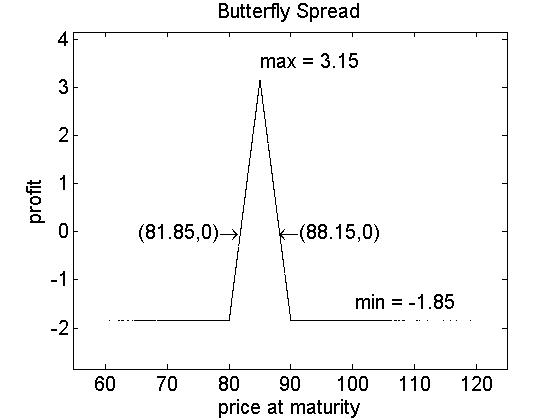
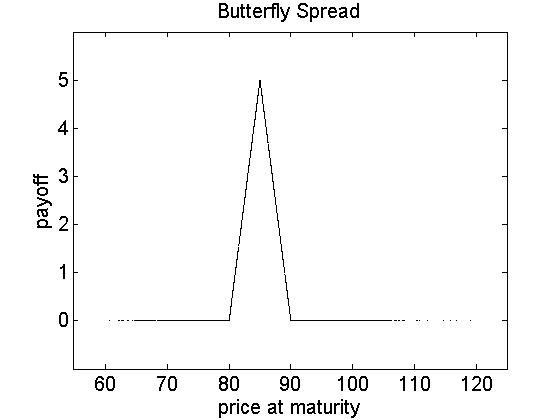
Graph 2: Stock Price Evolution of the Underlying Asset

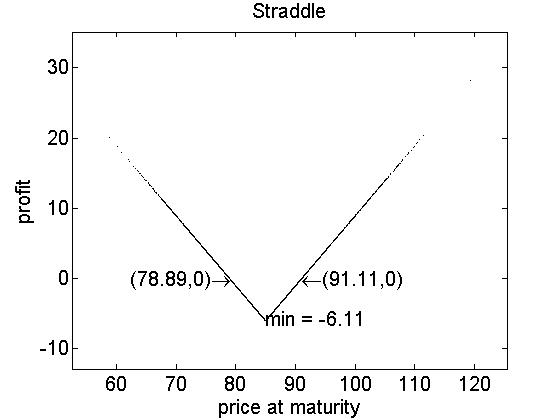
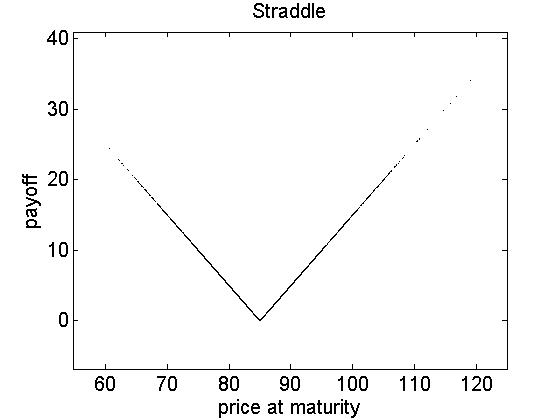


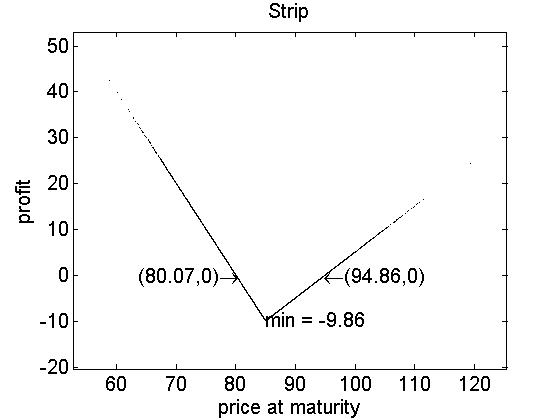
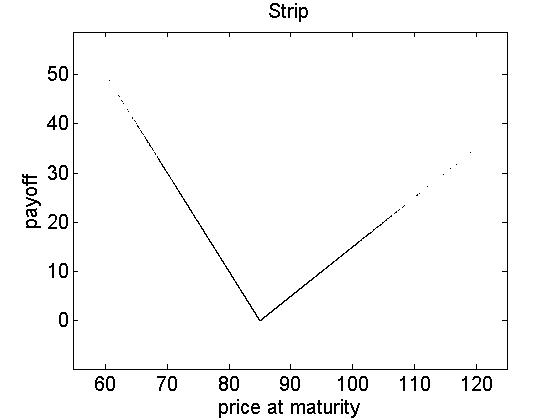
Graph 3: Payoff Diagram for Each Strategy

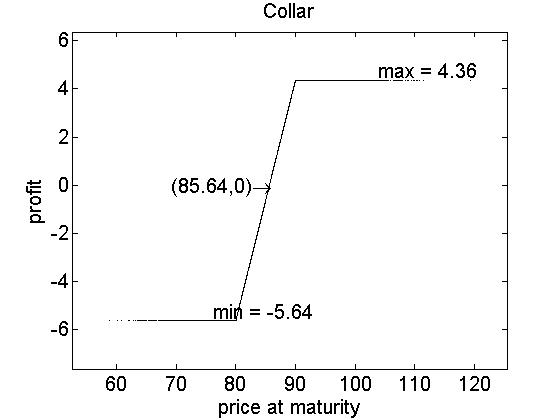
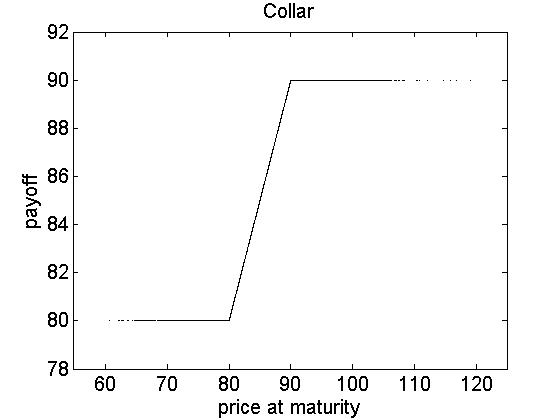












## **Spreadsheet Snippet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Calls | 15 Jan 16 (70d); CSize 100 | | | | |
| Strike | Ticker | Bid | Ask | Last | IVM |
| 75 | XOM 1/15/16 C75 | 9.199999809 | 9.649999619 | 8.949999809 | 24.76065254 |
| 77.5 | XOM 1/15/16 C77.5 | 6.900000572 | 7.5 | 6.850000381 | 22.9758091 |
| 80 | XOM 1/15/16 C80 | 5.199999809 | 5.449999809 | 5.400000572 | 22.18484879 |
| 82.5 | XOM 1/15/16 C82.5 | 3.5 | 3.75 | 3.650000572 | 20.76688004 |
| 85 | XOM 1/15/16 C85 | 2.159999847 | 2.359999657 | 2.270000458 | 19.57300377 |
| 87.5 | XOM 1/15/16 C87.5 | 1.279999733 | 1.350000381 | 1.350000381 | 18.95758438 |
| 90 | XOM 1/15/16 C90 | 0.620000005 | 0.720000029 | 0.660000026 | 18.17798042 |
| 92.5 | XOM 1/15/16 C92.5 | 0.290000021 | 0.350000024 | 0.270000041 | 17.81218719 |
| 95 | XOM 1/15/16 C95 | 0.100000024 | 0.170000017 | 0.150000036 | 17.42953873 |
| Puts | 15 Jan 16 (70d); CSize 100 | | | | |
| Strike | Ticker | Bid | Ask | Last | IVM |
| 75 | XOM 1/15/16 P75 | 0.800000012 | 0.870000005 | 0.870000005 | 26.28457451 |
| 77.5 | XOM 1/15/16 P77.5 | 1.149999619 | 1.260000229 | 1.279999733 | 24.47932434 |
| 80 | XOM 1/15/16 P80 | 1.680000305 | 1.779999733 | 1.710000038 | 22.6422596 |
| 82.5 | XOM 1/15/16 P82.5 | 2.449999809 | 2.600000381 | 2.470000267 | 21.13739014 |
| 85 | XOM 1/15/16 P85 | 3.550000191 | 3.75 | 3.650000572 | 19.87463379 |
| 87.5 | XOM 1/15/16 P87.5 | 5.050000191 | 5.25 | 5.150000572 | 18.82579422 |
| 90 | XOM 1/15/16 P90 | 6.850000381 | 7.699999809 | 7.079999924 | 20.58883858 |
| 92.5 | XOM 1/15/16 P92.5 | 9 | 10.35000038 | 0 | 23.8316288 |
| 95 | XOM 1/15/16 P95 | 11.30000019 | 12.69999981 | 0 | 25.58172226 |

|  |  |  |  |
| --- | --- | --- | --- |
| Underlying | XOM US Equity |  |  |
| Underlying description | EXXON MOBIL CORP |  |  |
| Und. Price | 84.475 |  |  |
| Underlying currency | USD |  |  |
| Trade | 11/06/2015 |  |  |
| Trade time | 16:30 |  |  |
| Settle | 11/06/2015 |  |  |
| Price (Total) | 483 |  |  |
| Price (Share) | 4.83 |  |  |
| Price (%) | 5.7177 |  |  |
| Currency | USD |  |  |
| Delta (%) | 54.15 |  |  |
| Gamma (%) | 0.4084 |  |  |
| Vega | 2.13 |  |  |
| Theta | -0.33 |  |  |
| Rho | 0.04 |  |  |
| Time Value | 35.5 |  |  |
| Gearing |  |  |  |
| Break-Even (%) |  |  |  |
| Strategy | Call/Put Spread | Warrant |  |
| Ticker |  | XOM US 1/15/16 C80 | XOM US 1/15/16 C90 |
| Style |  | Vanilla | Vanilla |
| Call/Put |  | Call | Call |
| Direction |  | Buy | Sell |
| Strike |  | 80 | 90 |
| Strike | % Money | 5.30% ITM | 6.54% OTM |
| Contracts |  | 1 | 1 |
| Expiry |  | 01/15/2016 16:30 | 01/15/2016 16:30 |
| Time to Expiry |  | 70 - 00:00 | 70 - 00:00 |
| Model |  | BS - continuous | BS - continuous |
| Vol | Implied | 21.361 | 17.308 |
| Forward | Carry | 83.7974 | 83.7974 |
| USD||Rate | MMkt | 0.3414 | 0.3414 |
| Dividend Yield |  | 4.5477 | 4.5477 |
| Discounted Div Flow |  | 0.73 | 0.73 |
| Borrow Cost |  | 0 | 0 |
| Leg Prc (Total) |  | 545 | -62 |
| Leg Prc (Share) |  | 5.45 | -0.62 |
| Leg Price (%) |  | 6.4516 | -0.7339 |
| Leg Delta (%) |  | 72.65 | -18.5 |
| Leg Gamma (%) |  | 3.9684 | -3.56 |
| Leg Vega |  | 11.95 | -9.83 |
| Leg Theta |  | -1.39 | 1.06 |
| Leg Rho |  | 0.07 | -0.03 |
| Leg Time Value |  | 97.5 | -62 |
| Leg Gearing |  | 15.5 | -136.25 |
| Leg Break-Even |  | 1.15 | 7.27 |

## **Python Code**

# -\*- coding: utf-8 -\*-

"""

Created on Sun Nov 8 23:18:29 2015

@author: Yibing

"""

**from** abc **import** ABCMeta, abstractmethod

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** datetime

**import** data\_handler

**class** MonteCarlo(metaclass = ABCMeta):

"""

    Abstract class for General Monte Carlo method.

    """

**def** \_\_init\_\_(self):

**pass**

@abstractmethod

**def** \_simulate\_once(self):

**pass**

@abstractmethod

**def** \_run\_simulation(self):

**pass**

@abstractmethod

**def** \_output\_results(self):

**pass**

@abstractmethod

**def** do\_simulation(self):

"""

        Execute Monte Carlo Method, output results.

        """

**pass**

**class** MonteCarloStrategy(MonteCarlo):

"""

    Abstract class for Monte Carlo Option Strategy.

    """

**def** \_\_init\_\_(self, strategy\_type, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, basis):

"""

        Parameters:

        exp\_return - Expected annual log return.

        volatility - Annual volatility.

        expiration - Maturity in years

        num\_of\_periods - Number of small intervals for one path. d

        all\_paths - A list of list containing initial\_price plus subsequent

            prices. (num\_of\_periods + 1) in total.

        all\_payoffs - A list ot all payoffs at expiration.

        all\_future\_stock\_prices - A list of all stock prices at expiration.

        """

self.strategy\_type = strategy\_type

self.ticker = ticker

self.initial\_stock\_price = initial\_stock\_price

self.exp\_return = exp\_return

self.volatility = volatility

self.risk\_free\_rate = risk\_free\_rate

self.expiration = expiration

self.num\_of\_sims = num\_of\_sims

self.num\_of\_periods = int(np.round(self.expiration \* basis))

self.delta\_t = self.expiration / self.num\_of\_periods # Delta t

self.is\_finished = False

self.all\_paths = []

self.all\_payoffs = []

self.all\_future\_stock\_prices = []

@abstractmethod

**def** \_simulate\_once(self):

"""

        This method will be overridden by the child classes.

        """

**pass**

**def** \_generate\_stock\_path(self):

"""

        Generate one path of stock evolution.

        """

stock\_path = [0] \* (self.num\_of\_periods + 1)

stock\_path[0] = self.initial\_stock\_price

**for** i **in** range(1, self.num\_of\_periods + 1):

noise = np.random.normal(0.0, 1.0)

stock\_path[i] = stock\_path[i - 1] \* np.exp((self.exp\_return -

0.5 \* self.volatility \*\* 2) \* self.delta\_t + \

self.volatility \* np.sqrt(self.delta\_t) \* noise)

self.all\_paths.append(stock\_path)

self.all\_future\_stock\_prices.append(stock\_path[-1])

**def** \_run\_simulation(self):

"""

        Run the entire simulation.

        """

**for** i **in** range(self.num\_of\_sims):

self.\_simulate\_once()

self.is\_finished = True

**def** \_output\_results(self):

"""

        Plot figure, save figure to the file.

        """

**if** self.is\_finished == False:

self.do\_simulation()

mean\_payoff = np.mean(self.all\_payoffs)

mean\_stock\_price = np.mean(self.all\_future\_stock\_prices)

discounted\_payoff = np.exp(-self.risk\_free\_rate \* self.expiration) \* \

mean\_payoff

**print**("Expected Future Stock Price := $%.2f" % mean\_stock\_price)

**print**("Expected Payoff of %s Strategy := $%.2f" % (self.strategy\_type,

mean\_payoff))

**print**("Present Value of %s Strategy := $%.2f" % (self.strategy\_type,

discounted\_payoff))

index = [x \* self.delta\_t **for** x **in** range(self.num\_of\_periods + 1)]

# Plot stock price evolution

plt.figure(1, figsize = (8, 10))

plt.subplot(2, 1, 1)

plt.title("Stock Price Evolution of %s (Expected: $%.2f)" % (self.ticker,

mean\_stock\_price))

plt.xlabel("Time in Years")

plt.ylabel("Price")

plt.subplot(2, 1, 2)

plt.title("Payoff of %s Strategy ($%.2f Per Share)" % (

self.strategy\_type, discounted\_payoff))

plt.xlabel("Price at Maturity")

plt.ylabel("Payoff")

plt.grid(True)

**for** i, curve **in** enumerate(self.all\_paths):

plt.subplot(2, 1, 1)

plt.plot(index, curve)

plt.subplot(2, 1, 2)

plt.scatter(curve[-1], self.all\_payoffs[i])

plt.savefig("%s.jpg" % self.strategy\_type, dpi = 600)

plt.show()

plt.close(1)

**def** do\_simulation(self):

"""

        Execute Monte Carlo Method, output results.

        """

self.\_run\_simulation()

self.\_output\_results()

result = pd.DataFrame(np.array([self.all\_future\_stock\_prices,

self.all\_payoffs]).transpose(),

columns = ['Stock Price', 'Payoff'])

result.to\_csv('%s.csv' % self.strategy\_type)

**class** MonteCarloNaiveOption(MonteCarloStrategy):

"""

    This is simpy the option pricing for European call and put.

    """

**def** \_\_init\_\_(self, option\_type, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike,

basis = 252):

"""

        Paramaters:

        option\_type - Could be 'call' or 'put'

        """

super().\_\_init\_\_('Option', ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims,

basis)

self.option\_type = option\_type

self.strike = strike

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = None

**if** self.option\_type == 'call':

payoff = max(0.0, self.all\_future\_stock\_prices[-1] - self.strike)

**elif** self.option\_type == 'put':

payoff = max(0.0, self.strike - self.all\_future\_stock\_prices[-1])

self.all\_payoffs.append(payoff)

**class** MonteCarloBullSpread(MonteCarloStrategy):

"""

    Bull spread strategy by call options:

    Buy one European call with K1 and sell one European call

    with the same maturity with K2, where K1 < K2.

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike\_1,

strike\_2, basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Bull Spread', ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims,

basis)

self.strike\_1 = strike\_1

self.strike\_2 = strike\_2

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_1) - \

max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_2)

self.all\_payoffs.append(payoff)

**class** MonteCarloBearSpread(MonteCarloStrategy):

"""

    Bear spread strategy by put options:

    Sell one European put with K1 and buy one European put

    with the same maturity with K2, where K1 < K2.

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike\_1,

strike\_2, basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Bear Spread', ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims,

basis)

self.strike\_1 = strike\_1

self.strike\_2 = strike\_2

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = -max(0.0, self.strike\_1 - self.all\_future\_stock\_prices[-1]) + \

max(0.0, self.strike\_2 - self.all\_future\_stock\_prices[-1])

self.all\_payoffs.append(payoff)

**class** MonteCarloButterflySpread(MonteCarloStrategy):

"""

    Butterfly spread strategy by call options:

    Buy two European calls with K1 and K3 respectively, and sell two

    European calls with the same maturity with K2, where K1 < K2 < K3.

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike\_1,

strike\_2, strike\_3, basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Butterfly Spread', ticker, initial\_stock\_price,

exp\_return, volatility, risk\_free\_rate, expiration,

num\_of\_sims, basis)

self.strike\_1 = strike\_1

self.strike\_2 = strike\_2

self.strike\_3 = strike\_3

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_1) + \

max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_3) - \

2 \* max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_2)

self.all\_payoffs.append(payoff)

**class** MonteCarloStraddle(MonteCarloStrategy):

"""

    Straddle:

    Buy one European call with K and one European put with

    the same strike price and maturity.

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike,

basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Straddle', ticker, initial\_stock\_price,

exp\_return, volatility, risk\_free\_rate, expiration,

num\_of\_sims, basis)

self.strike = strike

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = max(0.0, self.all\_future\_stock\_prices[-1] - self.strike) + \

max(0.0, self.strike - self.all\_future\_stock\_prices[-1])

self.all\_payoffs.append(payoff)

**class** MonteCarloStrip(MonteCarloStrategy):

"""

    Strip:

    Buy one European call and two European puts with the K and the same

    and maturity.

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike,

basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Strip', ticker, initial\_stock\_price,

exp\_return, volatility, risk\_free\_rate, expiration,

num\_of\_sims, basis)

self.strike = strike

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = max(0.0, self.all\_future\_stock\_prices[-1] - self.strike) + \

2 \* max(0.0, self.strike - self.all\_future\_stock\_prices[-1])

self.all\_payoffs.append(payoff)

**class** MonteCarloCollar(MonteCarloStrategy):

"""

    Collar:

    Buy one stock and one European put with K1, and sell one European call

    with K2, where K1 < S < K2. (Both call and put are OTM.)

    """

**def** \_\_init\_\_(self, ticker, initial\_stock\_price, exp\_return,

volatility, risk\_free\_rate, expiration, num\_of\_sims, strike\_1,

strike\_2, basis = 252):

"""

        Paramaters:

        """

super().\_\_init\_\_('Collar', ticker, initial\_stock\_price,

exp\_return, volatility, risk\_free\_rate, expiration,

num\_of\_sims, basis)

self.strike\_1 = strike\_1

self.strike\_2 = strike\_2

**def** \_simulate\_once(self):

self.\_generate\_stock\_path()

payoff = -max(0.0, self.all\_future\_stock\_prices[-1] - self.strike\_2) + \

max(0.0, self.strike\_1 - self.all\_future\_stock\_prices[-1]) + \

self.all\_future\_stock\_prices[-1]

self.all\_payoffs.append(payoff)

**if** \_\_name\_\_ == '\_\_main\_\_':

ticker = 'XOM'

start = datetime.date(2015, 5, 6)

end = datetime.date(2015, 11, 6)

s\_0 = 84.475

r = 0.003414

sigma = 0.2141

K1 = 80

K2 = 85

K3 = 90

T = (50.0 / 252)

N = 10000

mean\_return = data\_handler.calculate\_mean\_variance(ticker, start,

end)[0]

capm\_return = data\_handler.calculate\_capm\_return(ticker, start, end,

r / 252)[0]

strategy = []

strategy.append(MonteCarloBullSpread(ticker, s\_0, r,

sigma, r, T, N, K1, K3))

strategy.append(MonteCarloBearSpread(ticker, s\_0, r,

sigma, r, T, N, K1, K3))

strategy.append(MonteCarloButterflySpread(ticker, s\_0, r,

sigma, r, T, N, K1, K2, K3))

strategy.append(MonteCarloStraddle(ticker, s\_0, r,

sigma, r, T, N, K2))

strategy.append(MonteCarloStrip(ticker, s\_0, r,

sigma, r, T, N, K2))

strategy.append(MonteCarloCollar(ticker, s\_0, r,

sigma, r, T, N, K1, K3))

**for** s **in** strategy[1:]:

s.do\_simulation()

# ==============================================

quote = [4.83, 6.02, 1.83, 5.90, 9.77, 85.63]

names = ['Bull Spread', 'Bear Spread', 'Butterfly Spread', 'Straddle',

'Strip', 'Collar']

plt.figure(1, figsize = (12, 14))

**for** i, s **in** enumerate(names):

stock = pd.read\_csv("%s.csv" % s)['Stock Price']

payoff = pd.read\_csv("%s.csv" % s)['Payoff'] - quote[i]

plt.subplot(3, 2, i + 1)

plt.title(names[i])

plt.xlabel("Price at Maturity")

plt.ylabel("Net Payoff")

plt.scatter(stock, payoff)

plt.grid(True)

plt.savefig("Net Payoff.jpg", dpi = 600)

plt.close(1)

plt.figure(2, figsize = (12, 14))

**for** i, s **in** enumerate(names):

stock = pd.read\_csv("%s.csv" % s)['Stock Price']

payoff = pd.read\_csv("%s.csv" % s)['Payoff']

plt.subplot(3, 2, i + 1)

plt.title(names[i])

plt.xlabel("Price at Maturity")

plt.ylabel("Payoff")

plt.scatter(stock, payoff)

plt.grid(True)

plt.savefig("Payoff.jpg", dpi = 600)

plt.close(2)