Computational Problem Set 3 Solutions

Problem 1

First we import the options module we will use as well as Numpy and MatPlotLib

```
In [2]: import options as op
        import numpy as np
        import matplotlib.pyplot as plt
```

(a)

(i)

For this first problem the calculations are simple enough that we will use the raw option functions directly, using the options module as an options calculator. We start by calculating the delta of the described option:

```
In [3]: delta = op.BSCall_Delta(60, 0, 50, 0.5, 20, 5)
        print(delta)
```

```
0.937816048914623
```

The delta is the fraction of the optioned stock that must be shorted for a delta neutral portfolio. Our option position is in 200 shares, so we must short

```
In [4]: delta * 200
```

```
Out[4]: 187.5632097829246
```

or, rounding to the nearest whole number, 188 shares.
So our position now consists of a long call postion on 200 shares and a short cash position of 188 shares. In the first proposed scenario, the spot price of the underlying 1 month later is \$70. Our P&L 1 month later is the sum of the P&Ls of the two positions:

```
In [5]: PL1 = 200 * (op.BSCall(70, 1/12, 50, 0.5, 20, 5) - \
                     op.BSCall(60, 0, 50, 0.5, 20, 5)) - 188 * (70 - 60)
        print(PL1)
```

```
32.21246557980885
```

In the second scenario, the spot price of the underlying asset 1 month later is $50:

```
In [6]: PL2 = 200 * (op.BSCall(50, 1/12, 50, 0.5, 20, 5) - \
                     op.BSCall(60, 0, 50, 0.5, 20, 5)) - 188 * (50 - 60)
        print(PL2)
```

```
203.82148285929225
```

(ii)

In the described scenario, we hold 50 long call positions on a stock, or, equivalently, a long call position on 50 shares. Similar to part (a), we need to calculate the delta of this position to know how to build a delta neutral portfolio:

```
In [7]: delta = 50 * op.BSCall_Delta(110, 0, 105, 1.0, 15, 4)
        print(delta)
```

```
37.13674663977103
```

Thus, rounding to the nearest whole number of shares, we need to short 37 shares for a delta neutral position.
Now we calculate the P&L of the delta neutral portfolio in each of the 2 proposed scenarios. In the first proposal, 3 months later the stock is trading for $100. The P&L of our portfolio is

```
In [8]: PL1 = 50 * (op.BSCall(100, 0.25, 105, 1.0, 15, 4) - \
                    op.BSCall(110, 0, 105, 1.0, 15, 4)) - 37 *(100 - 110)
        print(PL1)
```

```
-7.309616468419051
```

In the second scenario, the stock is trading for $120 3 months later:

```
In [9]: PL2 = 50 * (op.BSCall(120, 0.25, 105, 1.0, 15, 4) - \
                    op.BSCall(110, 0, 105, 1.0, 15, 4)) - 37 *(120 - 110)
        print(PL2)
```

```
-23.235325810604877
```

In the alternative scenario proposed, we assume the changes in the spot price of the stock have happened in 1 month instead of 3 and we calculate the P&Ls of the portfolio in 1 month under this alternative hypothesis:

```
In [10]: PL3 = 50 * (op.BSCall(100, 1/12, 105, 1.0, 15, 4) - \
                     op.BSCall(110, 0, 105, 1.0, 15, 4)) - 37 *(100 - 110)
         print(PL3)
```

```
34.48844869155084
```

```
In [11]: PL4 = 50 * (op.BSCall(120, 1/12, 105, 1.0, 15, 4) - \
                     op.BSCall(110, 0, 105, 1.0, 15, 4)) - 37 *(120 - 110)
         print(PL4)
```

```
20.696490418551434
```

So, in this scenario, the positions have both profited modestly, but over a shorter period of time. To understand this, we compute the theta of the option:

```
In [13]: theta = op.BSCall_Theta(110, 0, 105, 1.0, 15, 4)
         print(theta)
```

```
-0.014943088606749760
```

We scale the theta up for the 50 options in the portfolio:

```
In [14]: 50 * theta
```

```
Out[14]: -0.747154430337488
```

According to this the option position loses 75 cents per day from option time decay. This implies a loss for the additional 2 months of

```
In [16]: 60 * 0.75
```

```
Out[16]: 45.0
```

Based on the theta at time 0 we estimate the option positon will lose $45 from option decay alone in 2 months. Generally estimating time decay based on the theta calculated at one point in time is not very accurate but in this case it is very close to the additional loss we observe between holding the option for 1 month vs. 3 months. It seems likely that time decay is, at least, a major factor in the losses in the position we observe afater 3 months.

(b)

We begin by calculating all the proposed option positions using the option functions directly. The spot price and risk free rate are fixed throughout, so we stort them in fixed variables:

```
In [8]: spot = 110
        ir = 3
```

Option # 1:

```
In [9]: op.BSCall(spot, 0, 100, 0.25, 20, ir)
```

```
Out[9]: 11.571486479406985
```

```
In [10]: op.BSCall_Delta(spot, 0, 100, 0.25, 20, ir)
```

```
Out[10]: 0.8595058353426077
```

```
In [11]: op.BSCall_Gamma(spot, 0, 100, 0.25, 20, ir)
```

```
Out[11]: 0.020282728873482814
```

```
In [12]: op.BSCall_Vega(spot, 0, 100, 0.25, 20, ir)
```

```
Out[12]: 0.12271050968457103
```

```
In [13]: op.BSCall_Theta(spot, 0, 100, 0.25, 20, ir)
```

```
Out[13]: -0.0202675206839212
```

The delta neutral position requires that we short

```
In [14]: 100 * op.BSCall_Delta(spot, 0, 100, 0.25, 20, ir)
```

```
Out[14]: 85.95058353426077
```

or, to the nearest whole number, 86 shares.
Option # 2:

```
In [15]: op.BSCall(spot, 0, 125, 1/12, 30, ir)
```

```
Out[15]: 0.3346531600153675
```

```
In [16]: op.BSCall_Delta(spot, 0, 125, 1/12, 30, ir)
```

```
Out[16]: 0.08017076405822277
```

```
In [17]: op.BSCall_Gamma(spot, 0, 125, 1/12, 30, ir)
```

```
Out[17]: 0.015631082929619393
```

```
In [18]: op.BSCall_Vega(spot, 0, 125, 1/12, 30, ir)
```

```
Out[18]: 0.04728402586209866
```

```
In [19]: op.BSCall_Theta(spot, 0, 125, 1/12, 30, ir)
```

```
Out[19]: -0.02401547556649159
```

The delta neutral position requires that we short

```
In [20]: 100 * op.BSCall_Delta(spot, 0, 125, 1/12, 30, ir)
```

```
Out[20]: 8.017076405822277
```

or, to the nearest whole number, 8 shares.
Option # 3:

```
In [22]: op.BSCall(spot, 0, 110, 2/12, 15, ir)
```

```
Out[22]: 2.9634333614506048
```

```
In [23]: op.BSCall_Delta(spot, 0, 110, 2/12, 15, ir)
```

```
Out[23]: 0.5446946541212022
```

```
In [24]: op.BSCall_Gamma(spot, 0, 110, 2/12, 15, ir)
```

```
Out[24]: 0.058852482489266514
```

```
In [25]: op.BSCall_Vega(spot, 0, 110, 2/12, 15, ir)
```

```
Out[25]: 0.17802875953003117
```

```
In [26]: op.BSCall_Theta(spot, 0, 110, 2/12, 15, ir)
```

```
Out[26]: -0.02662981790851466
```

The delta neutral position requires that we short

```
In [27]: 100 * op.BSCall_Delta(spot, 0, 110, 2/12, 15, ir)
```

```
Out[27]: 54.46946541212022
```

or, to the nearest whole number, 54 shares.
Option # 4:

```
In [34]: op.BSPut(spot, 0, 100, 2/12, 20, ir)
```

```
Out[34]: 0.45158884067974014
```

```
In [35]: op.BSPut_Delta(spot, 0, 100, 2/12, 20, ir)
```

```
Out[35]: -0.10215481284833611
```

```
In [36]: op.BSPut_Gamma(spot, 0, 100, 2/12, 20, ir)
```

```
Out[36]: 0.01984606381248308
```

```
In [37]: op.BSPut_Vega(spot, 0, 100, 2/12, 20, ir)
```

```
Out[37]: 0.08004579071034844
```

```
In [38]: op.BSPut_Theta(spot, 0, 100, 2/12, 20, ir)
```

```
Out[38]: -0.012197503821920561
```

The delta neutral position requires that we short

```
In [39]: 100 * op.BSPut_Delta(spot, 0, 100, 2/12, 20, ir)
```

```
Out[39]: -10.215481284833611
```

or, to the nearest whole number, we take a long position in 10 shares.
Option # 5:

```
In [40]: op.BSPut(spot, 0, 130, 6/52, 35, ir)
```

```
Out[40]: 20.098622636950594
```

```
In [41]: op.BSPut_Delta(spot, 0, 130, 6/52, 35, ir)
```

```
Out[41]: -0.9060077975160257
```

```
In [42]: op.BSPut_Gamma(spot, 0, 130, 6/52, 35, ir)
```

```
Out[42]: 0.012822848851338641
```

```
In [43]: op.BSPut_Vega(spot, 0, 130, 6/52, 35, ir)
```

```
Out[43]: 0.06265934409856054
```

```
In [44]: op.BSPut_Theta(spot, 0, 130, 6/52, 35, ir)
```

```
Out[44]: -0.01619328614804643
```

The delta neutral position requires that we short

```
In [46]: 100 * op.BSPut_Delta(spot, 0, 130, 6/52, 35, ir)
```

```
Out[46]: -90.60077975160257
```

or, to the nearest whole number, we take a long position in 91 shares.
In the second part of this problem we repeat the previous calculations for all the option positions given, but this time use option objects rather than using the option pricing function directly. Since the answers are exactly the same, I will only carry out the full procedure for option #1, but I will instantiate the option object and present the plot for each one.
Option # 1:

```
In [47]: option1 = op.option(strike=100, expiry=0.25, type='call')
```

```
In [49]: option1.price(spot, 0, 20, ir)
```

```
Out[49]: 11.571486479406985
```

```
In [50]: option1.delta(spot, 0, 20, ir)
```

```
Out[50]: 0.8595058353426077
```

```
In [51]: option1.gamma(spot, 0, 20, ir)
```

```
Out[51]: 0.020282728873482814
```

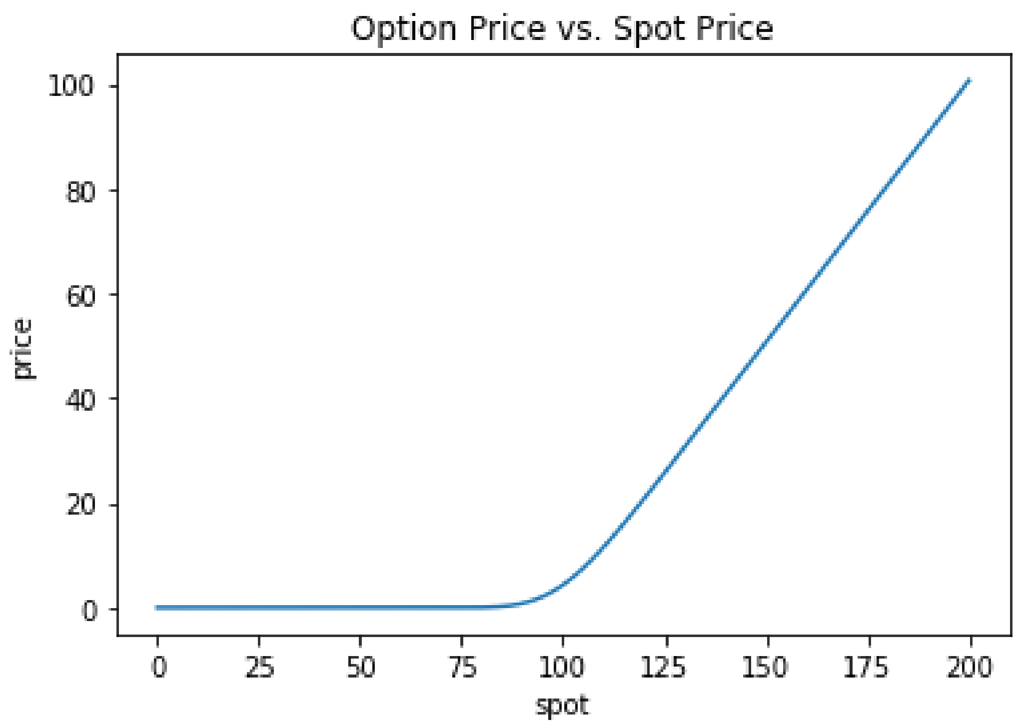
```
In [52]: option1.vega(spot, 0, 20, ir)
```

```
Out[52]: 0.12271050968457103
```

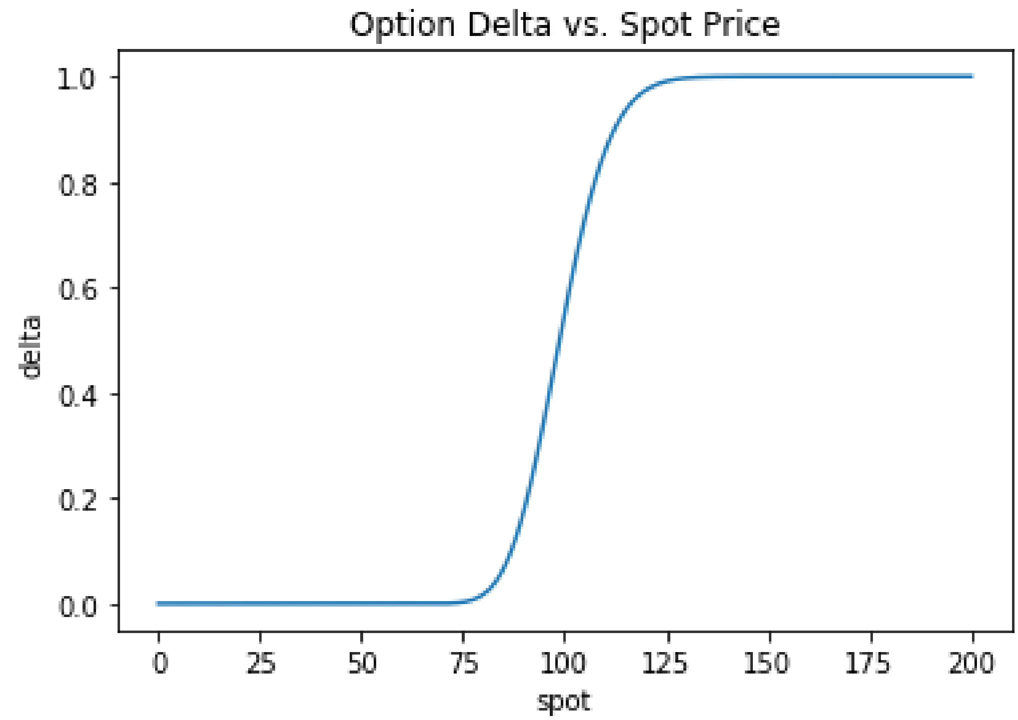
```
In [53]: option1.theta(spot, 0, 20, ir)
```

```
Out[53]: -0.0202675206839212
```

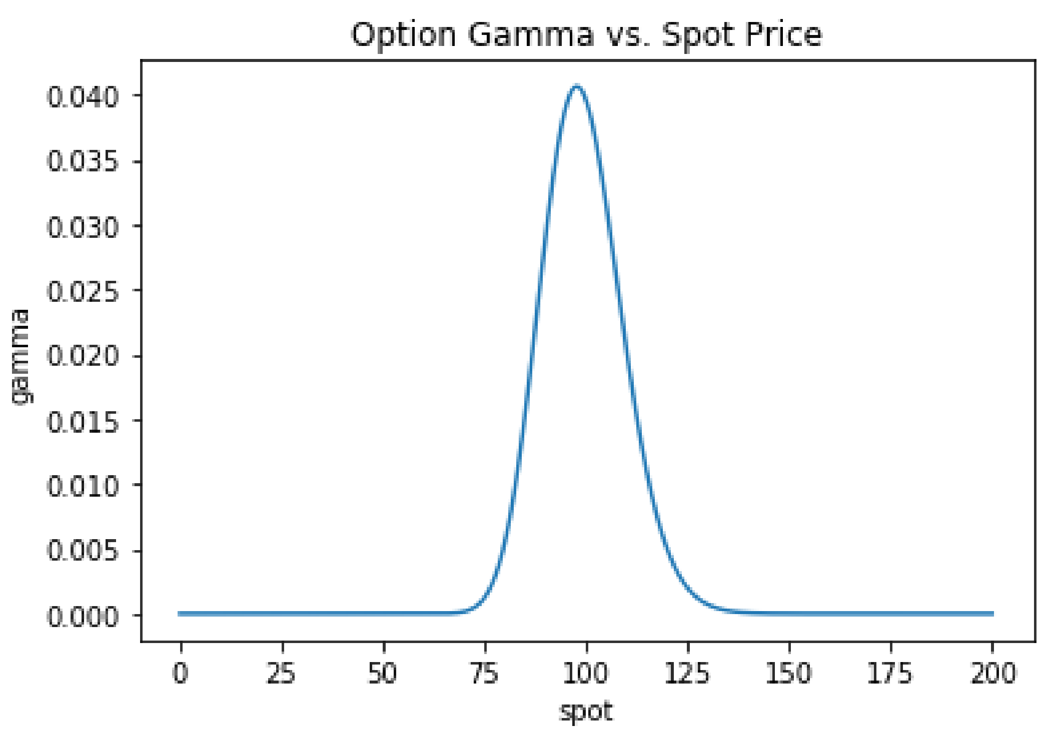
```
In [56]: option1.plot_price(0, 20, ir)
```



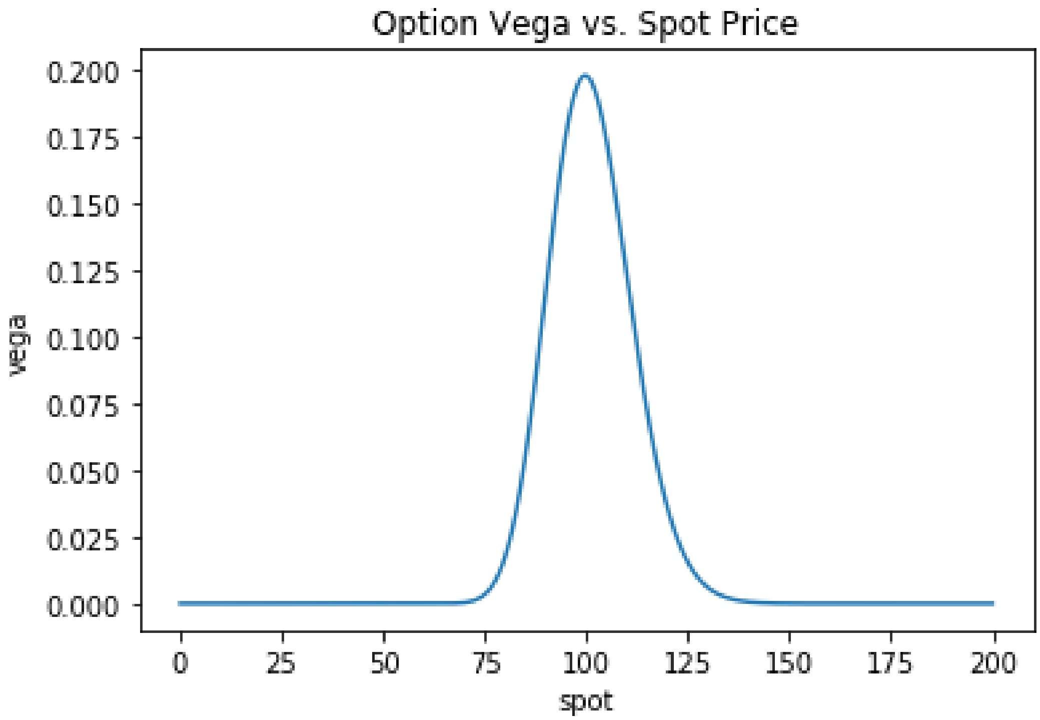

```
In [57]: option1.plot_delta(0, 20, ir)
```



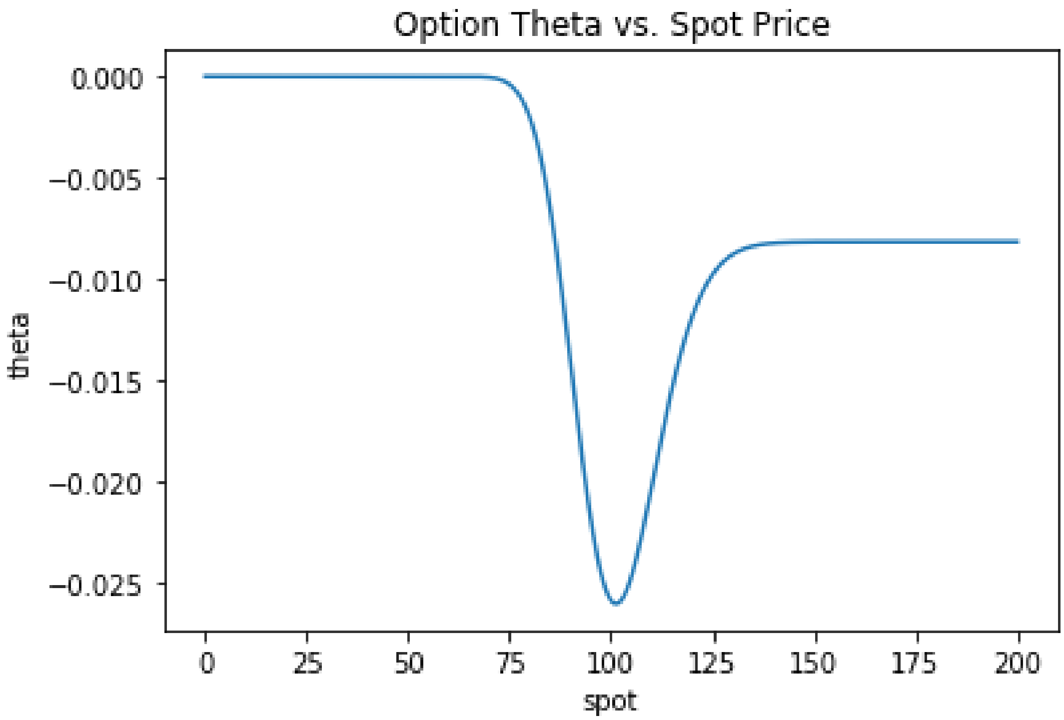

```
In [58]: option1.plot_gamma(0, 20, ir)
```



```
In [59]: option1.plot_vega(0, 20, ir)
```

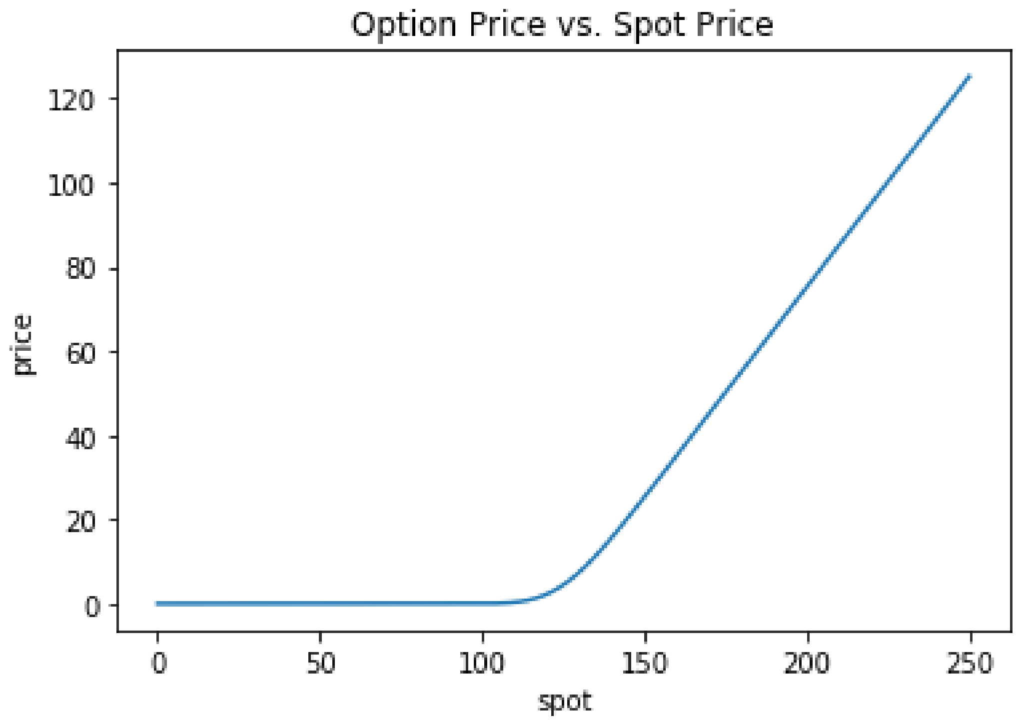



```
In [60]: option1.plot_theta(0, 20, ir)
```

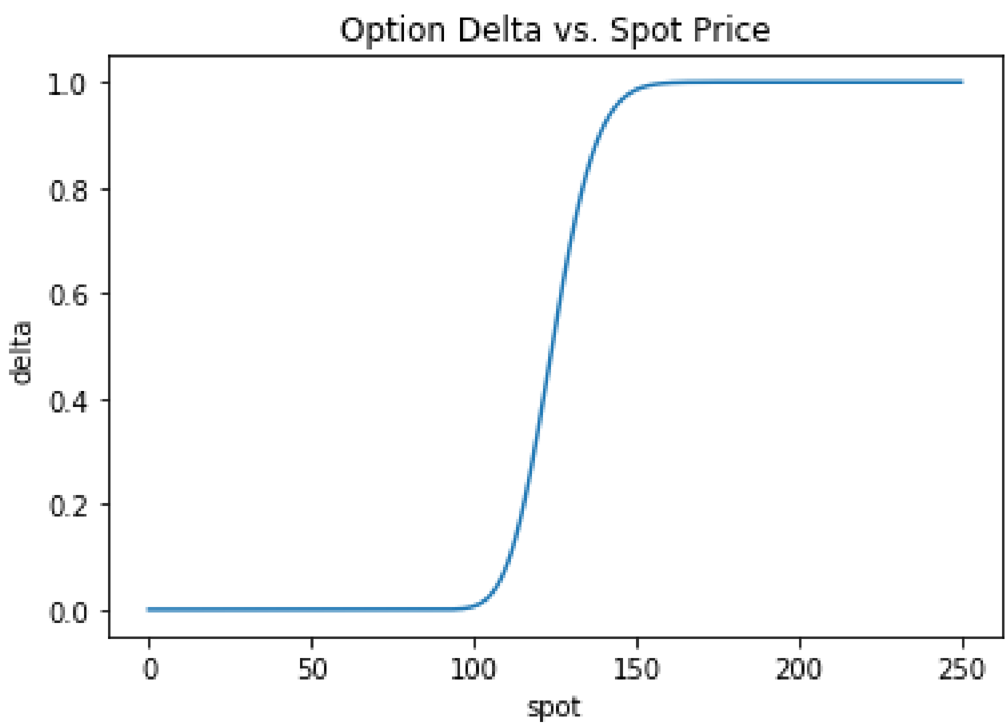


Option # 2:

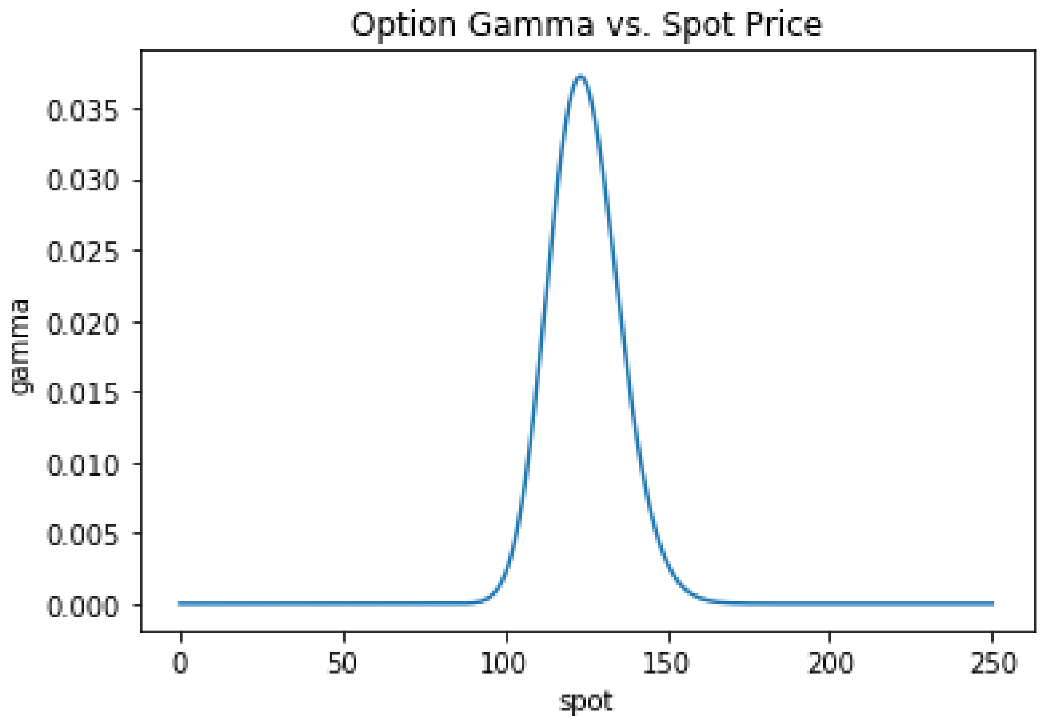
```
In [67]: option2 = op.option(strike=125, expiry=1/12, type='call')

In [68]: option2.plot_price(0, 30, ir)
```



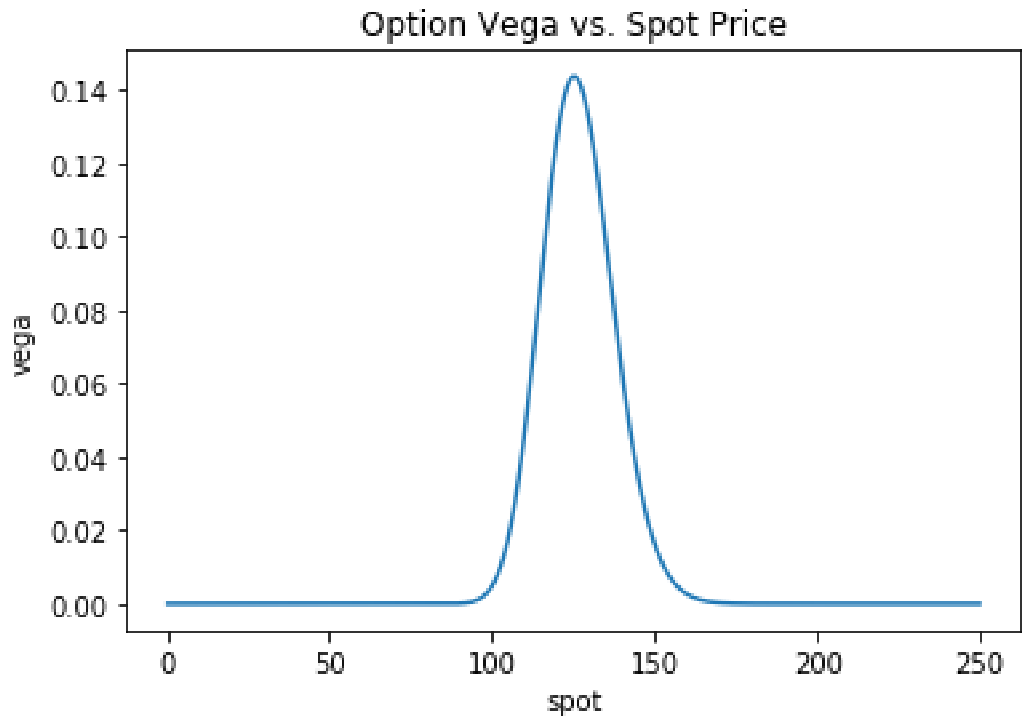

```
In [69]: option2.plot_delta(0, 30, ir)
```



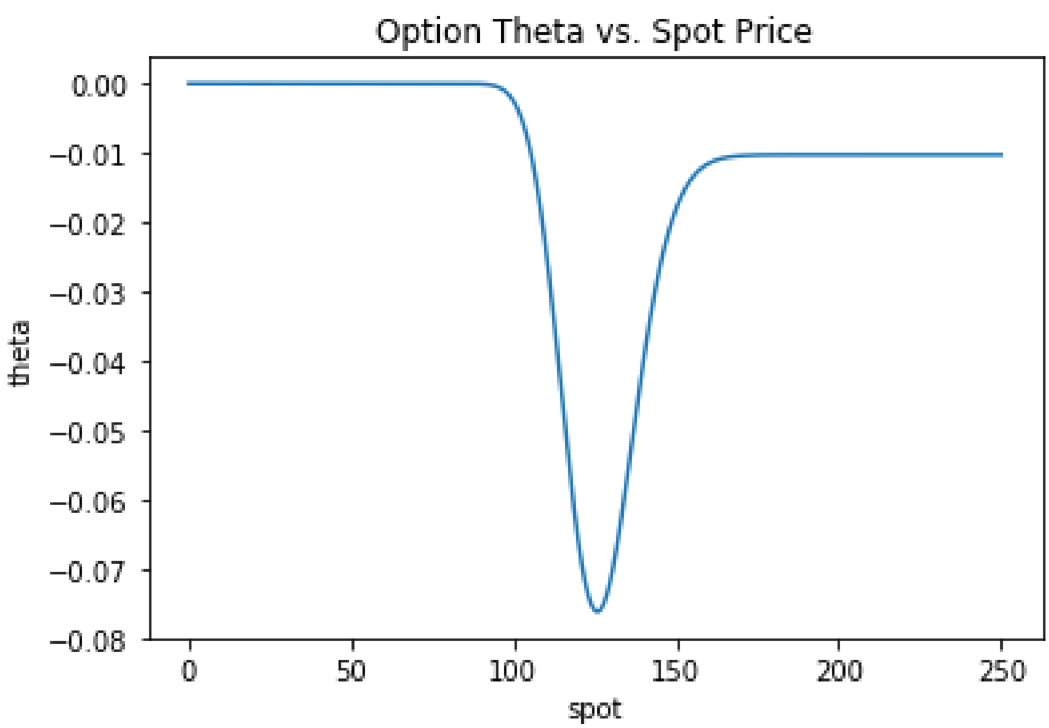
```
In [70]: option2.plot_gamma(0, 30, ir)
```




```
In [71]: option2.plot_vega(0, 30, ir)
```



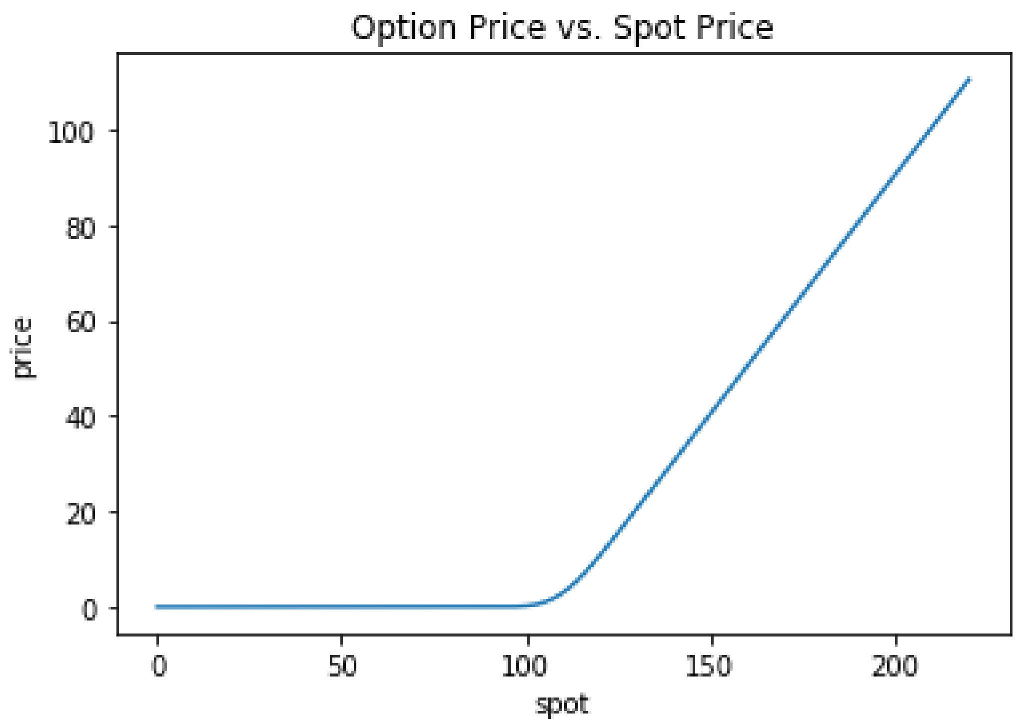

```
In [73]: option2.plot_theta(0, 30, ir)
```



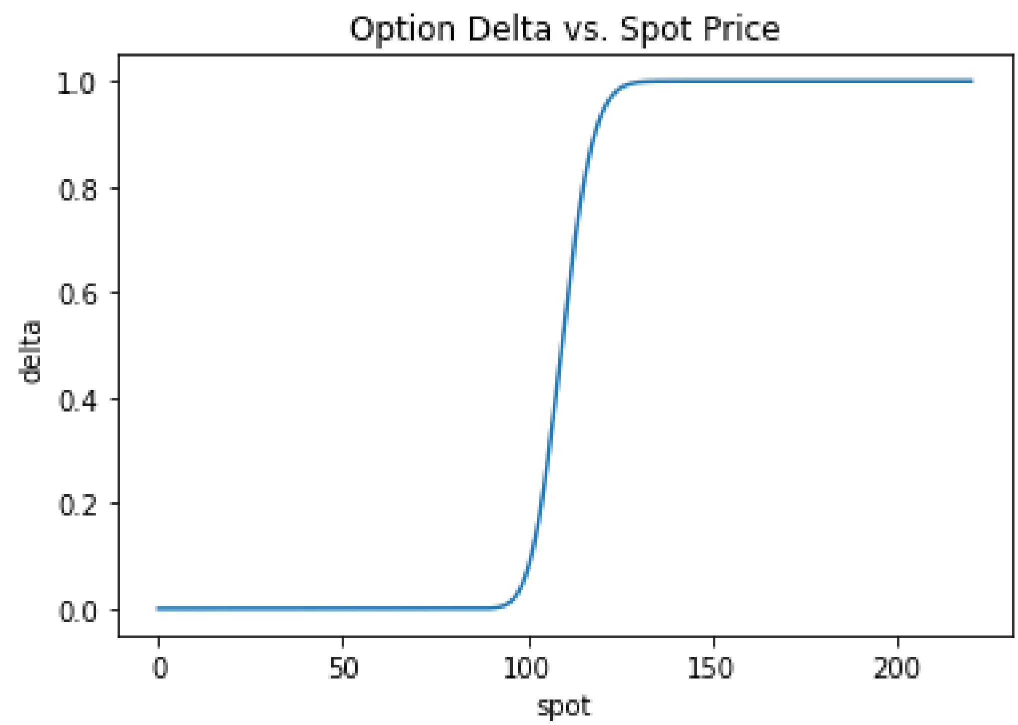
Option # 3:

```
In [74]: option3 = op.option(strike=110, expiry=2/12, type='call')
```

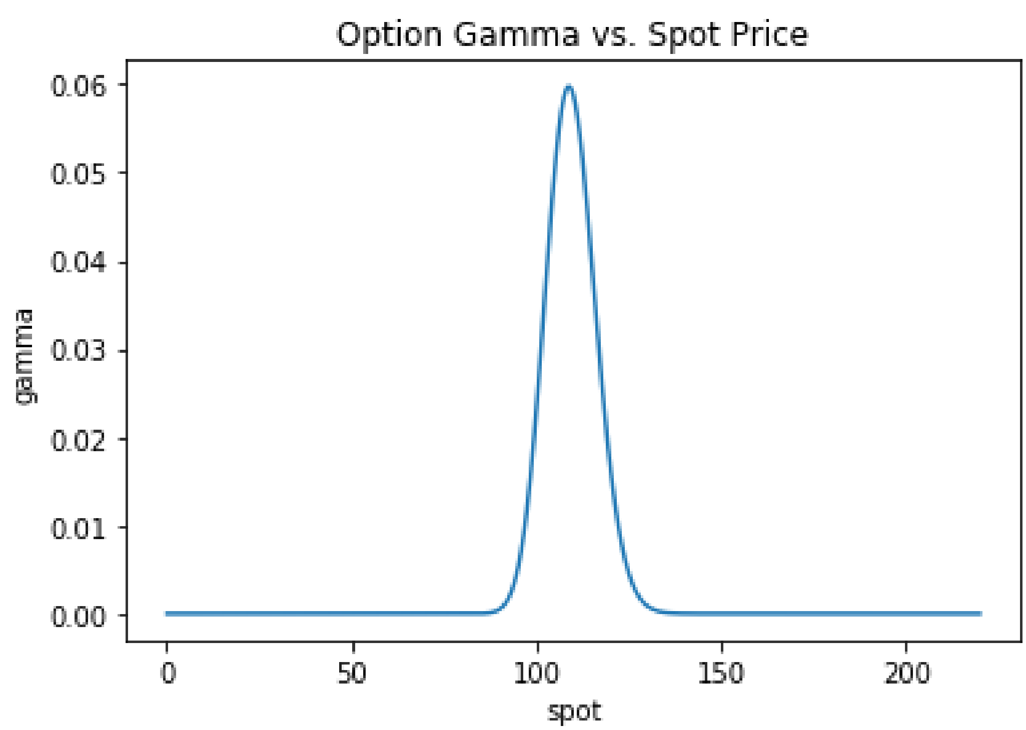
```
In [75]: option3.plot_price(0, 15, ir)
```



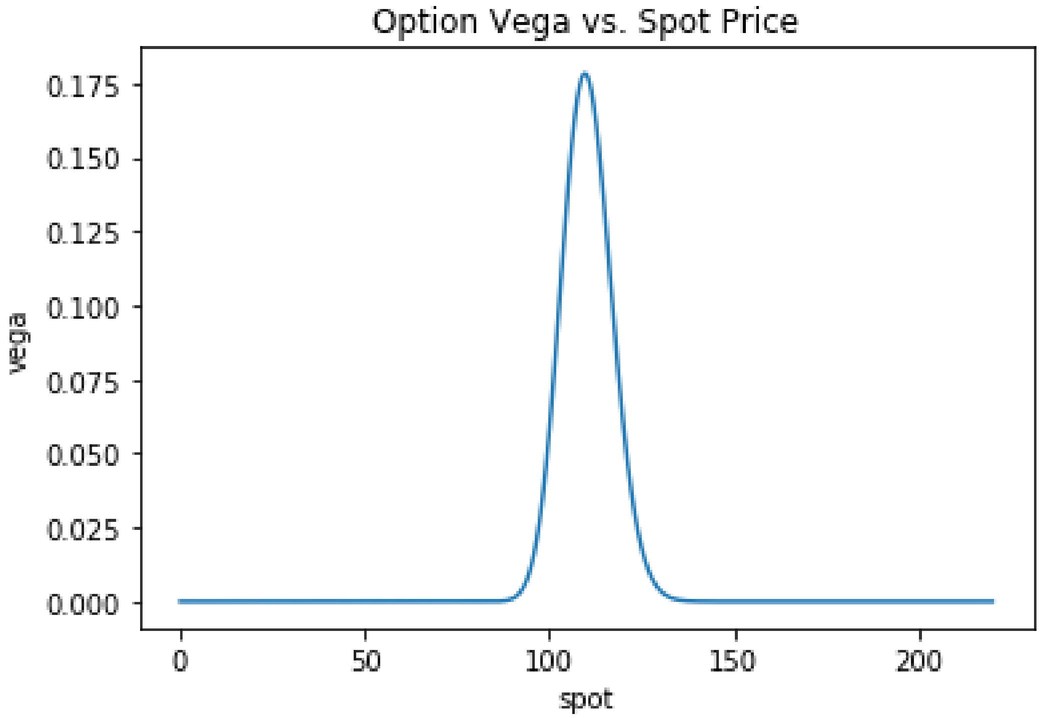

```
In [76]: option3.plot_delta(0, 15, ir)
```



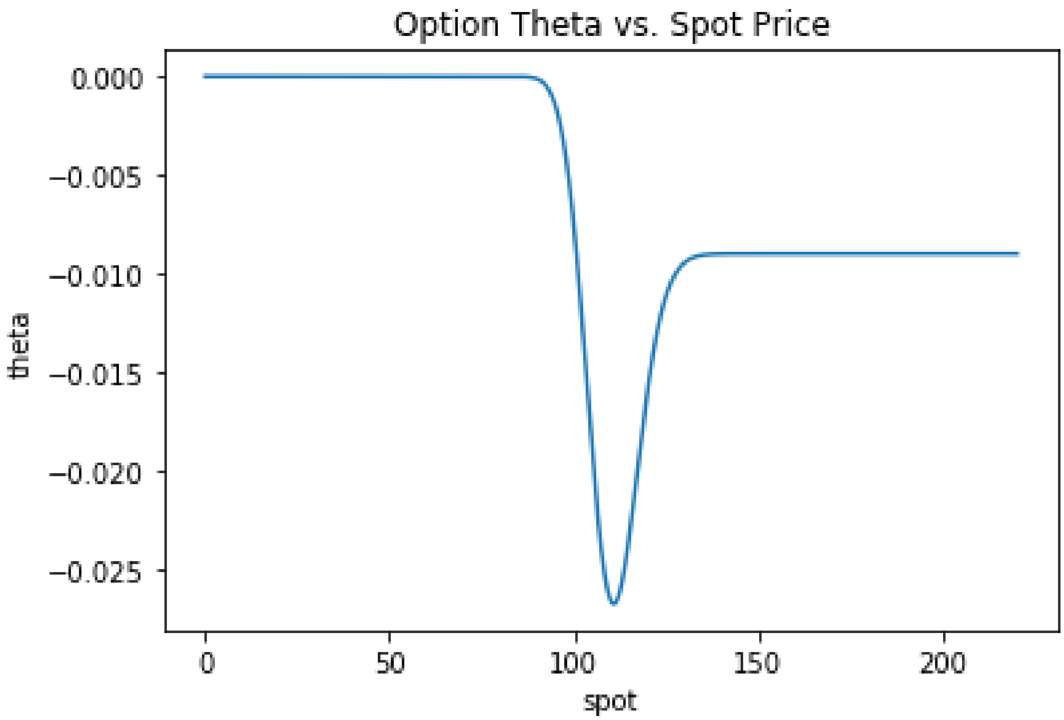

```
In [77]: option3.plot_gamma(0, 15, ir)
```



```
In [80]: option3.plot_vega(0, 15, ir)
```

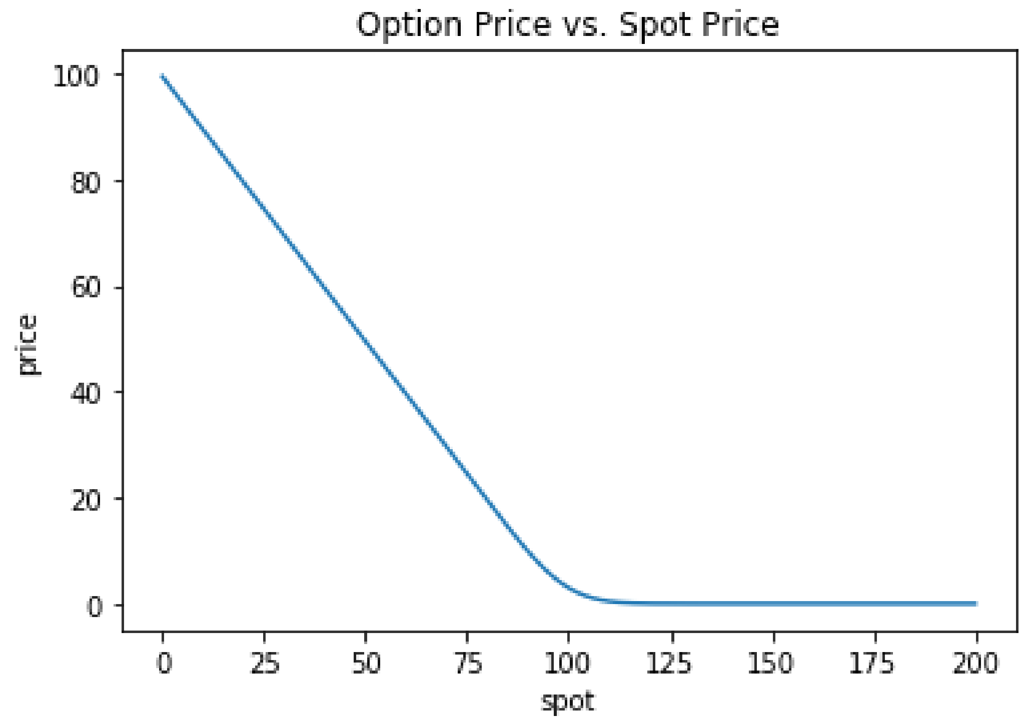



```
In [81]: option3.plot_theta(0, 15, ir)
```

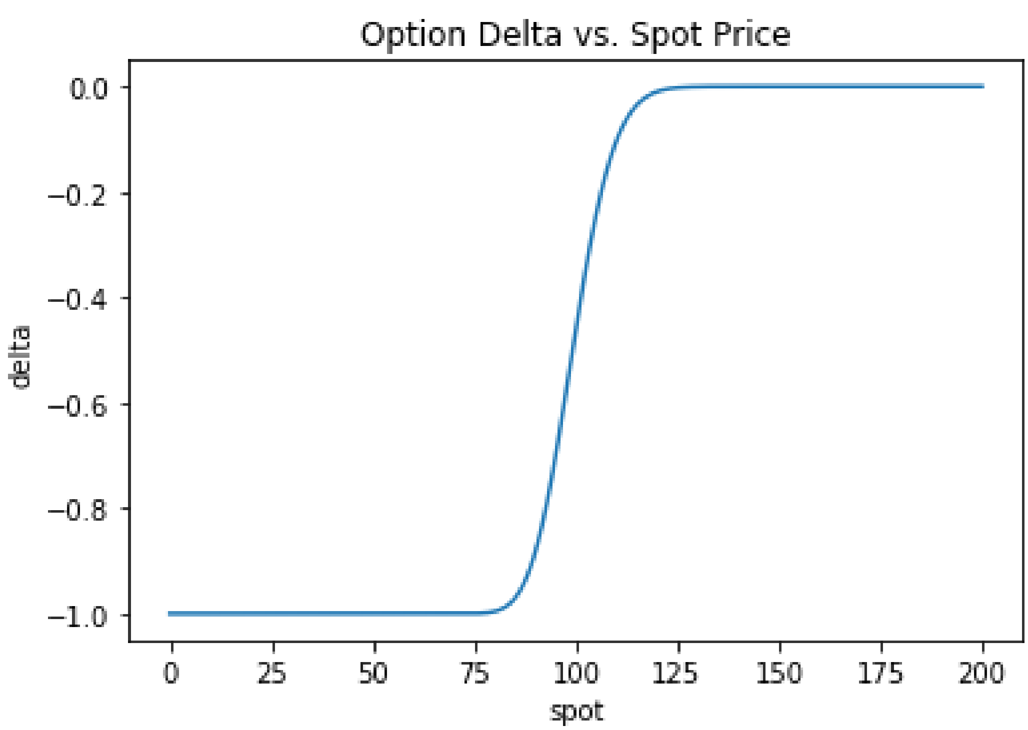


Option # 4:

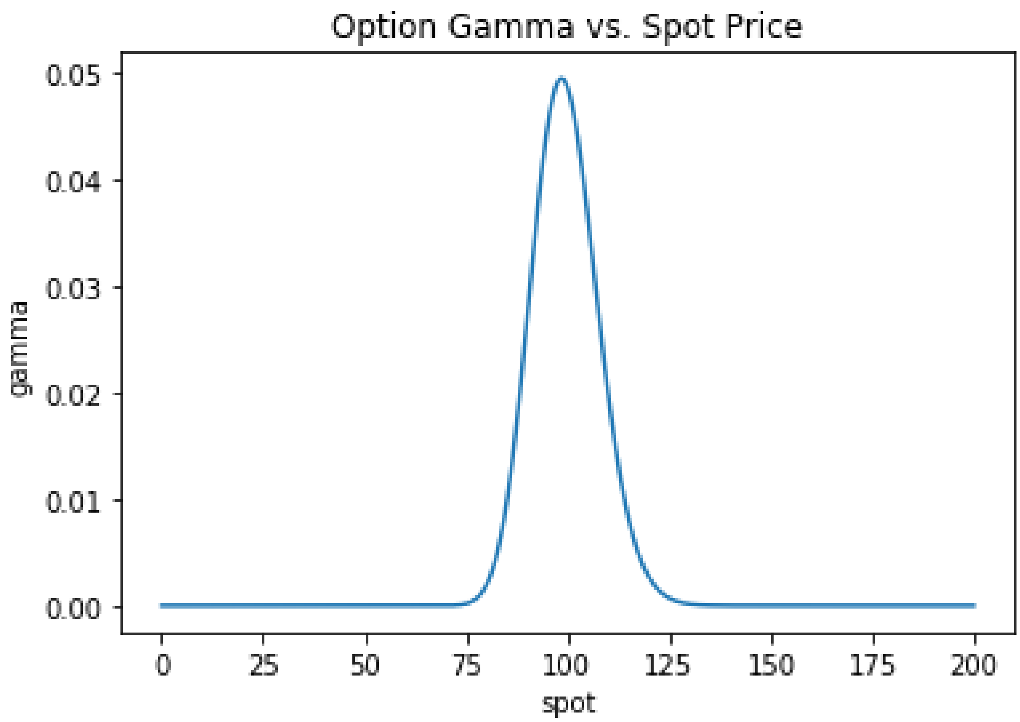
```
In [82]: option4 = op.option(strike=100, expiry=2/12, type='put')

In [83]: option4.plot_price(0, 20, ir)
```



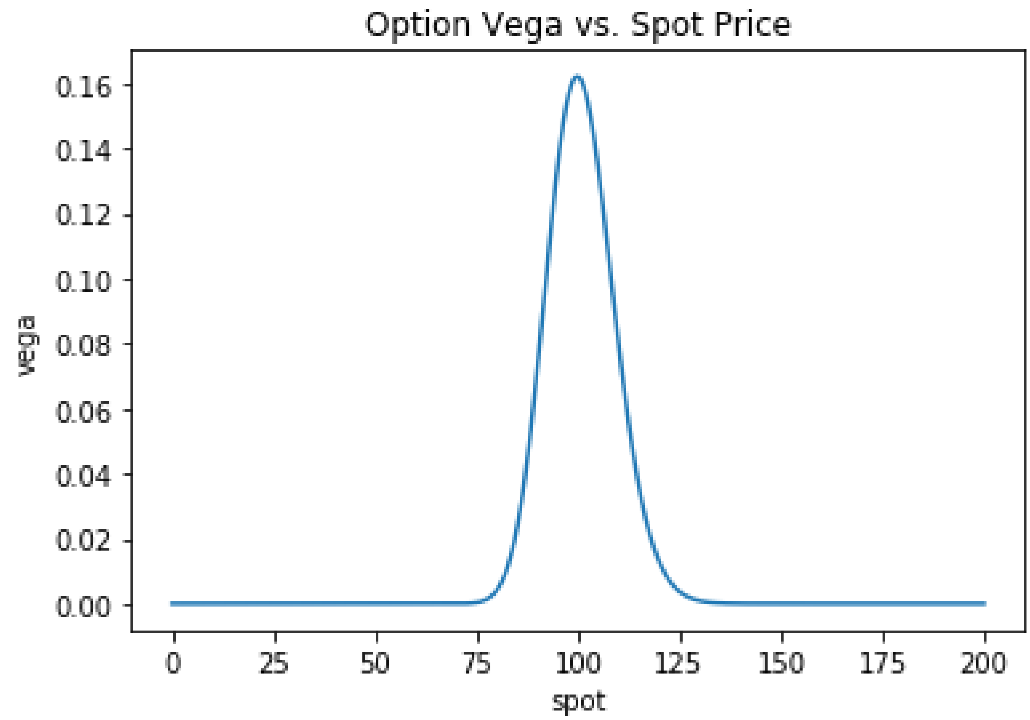
In [84]: `option4.plot_delta(0, 20, ir)`



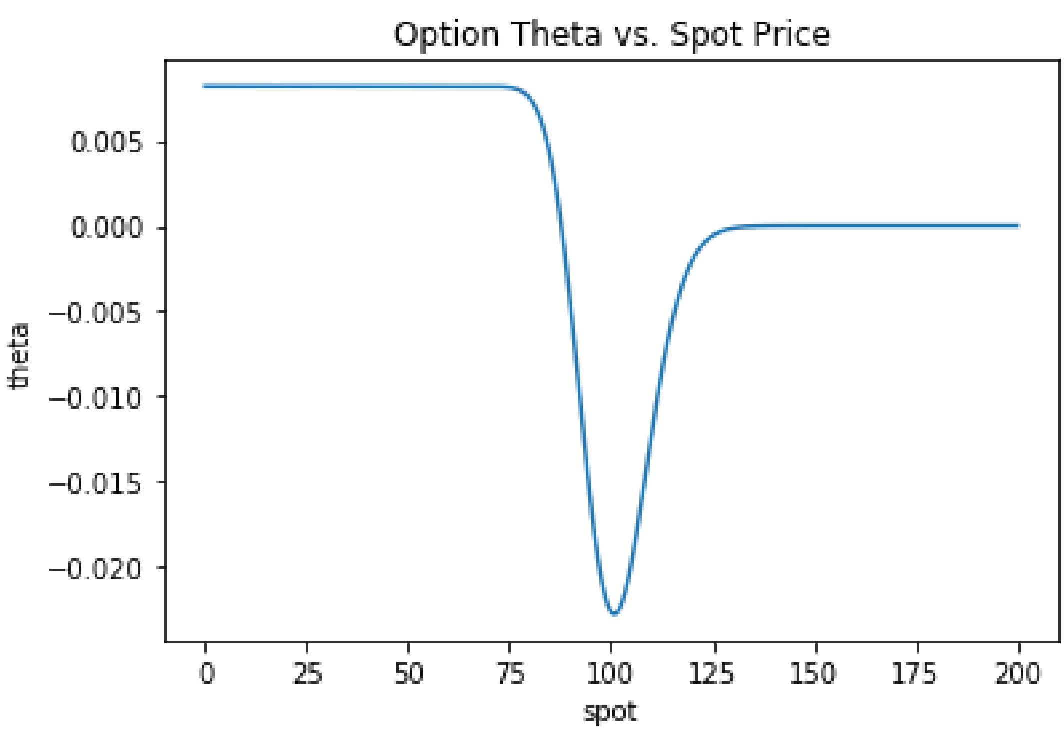
```
In [85]: option4.plot_gamma(0, 20, ir)
```




```
In [86]: option4.plot_vega(0, 20, ir)
```



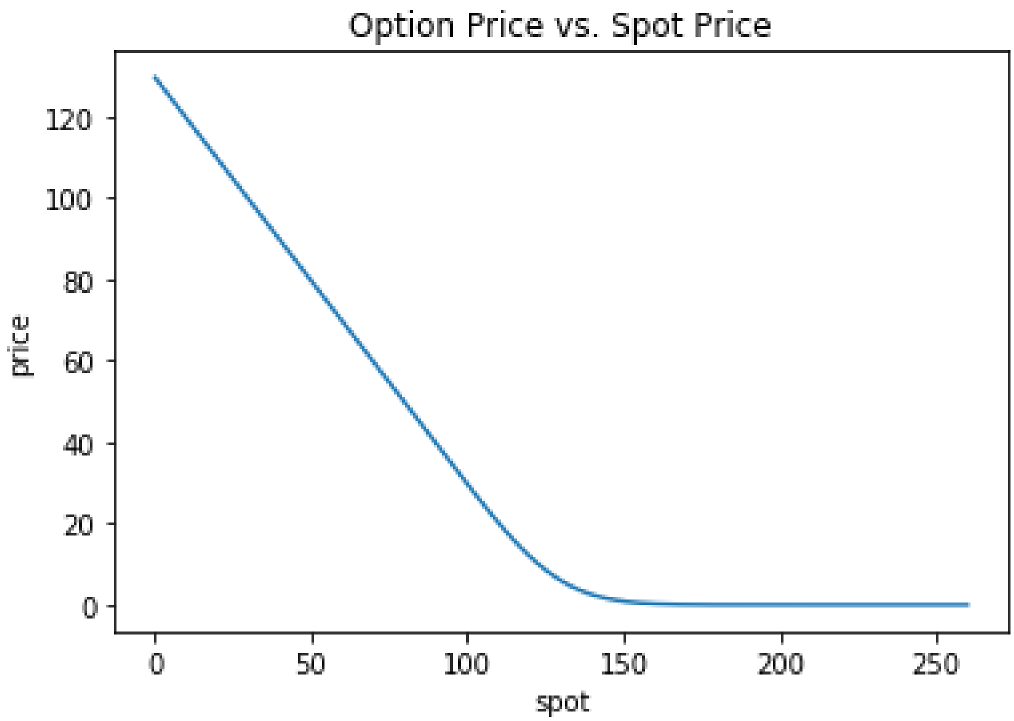

```
In [88]: option4.plot_theta(0, 20, ir)
```



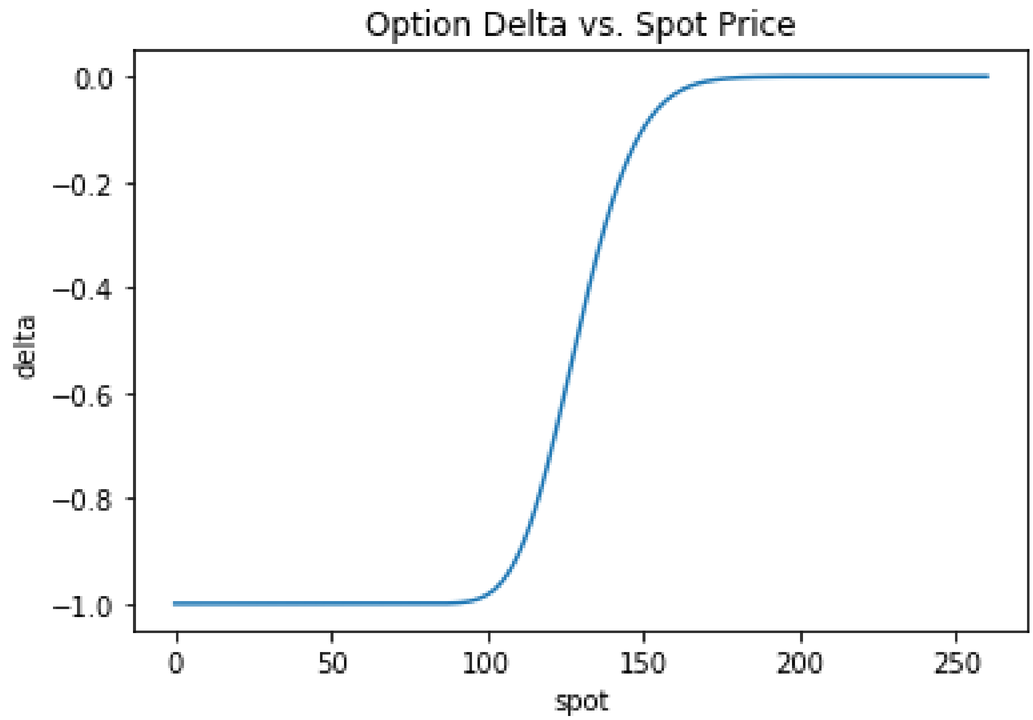
Option # 5:

```
In [89]: option5 = op.option(strike=130, expiry=6/52, type='put')
```

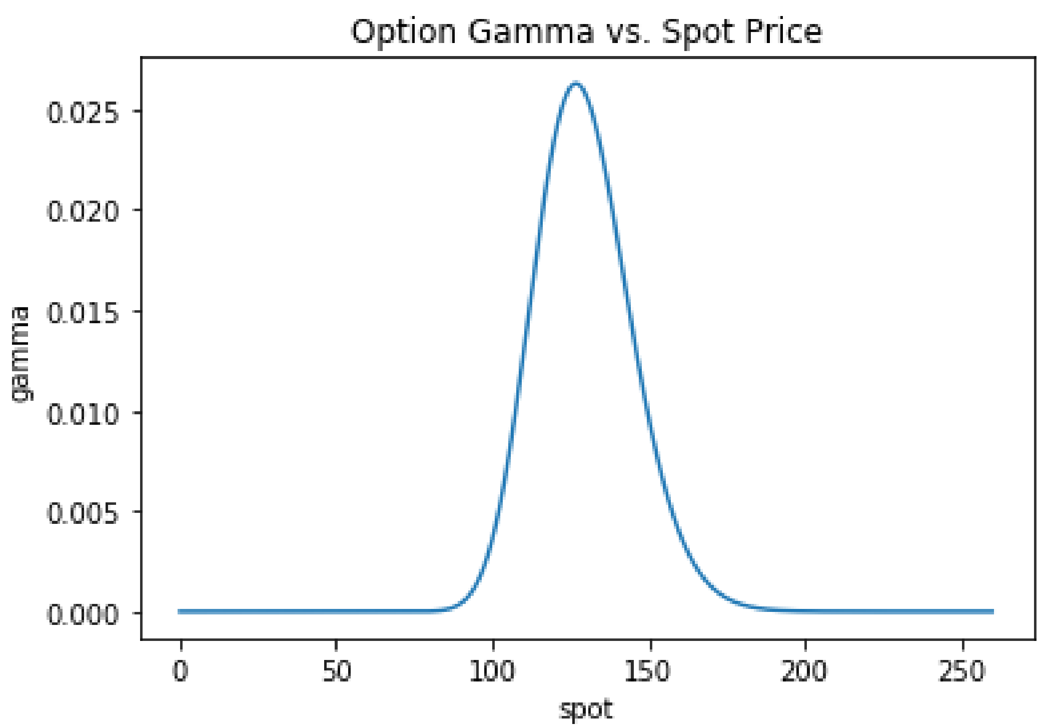
```
In [90]: option5.plot_price(0, 35, ir)
```



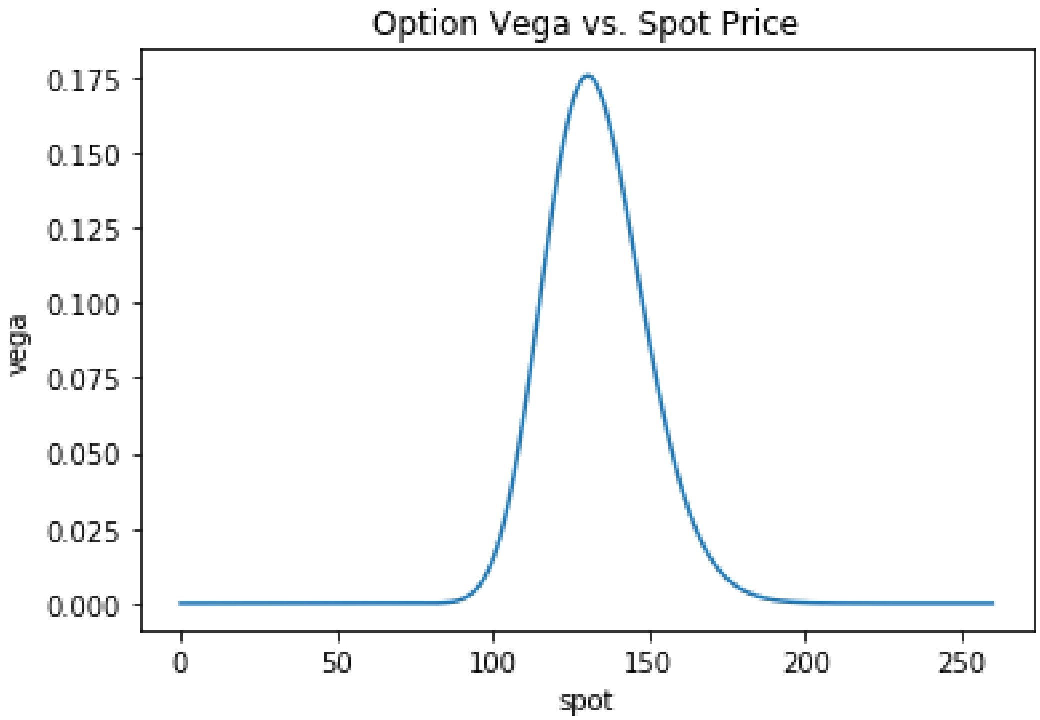

```
In [91]: option5.plot_delta(0, 35, ir)
```



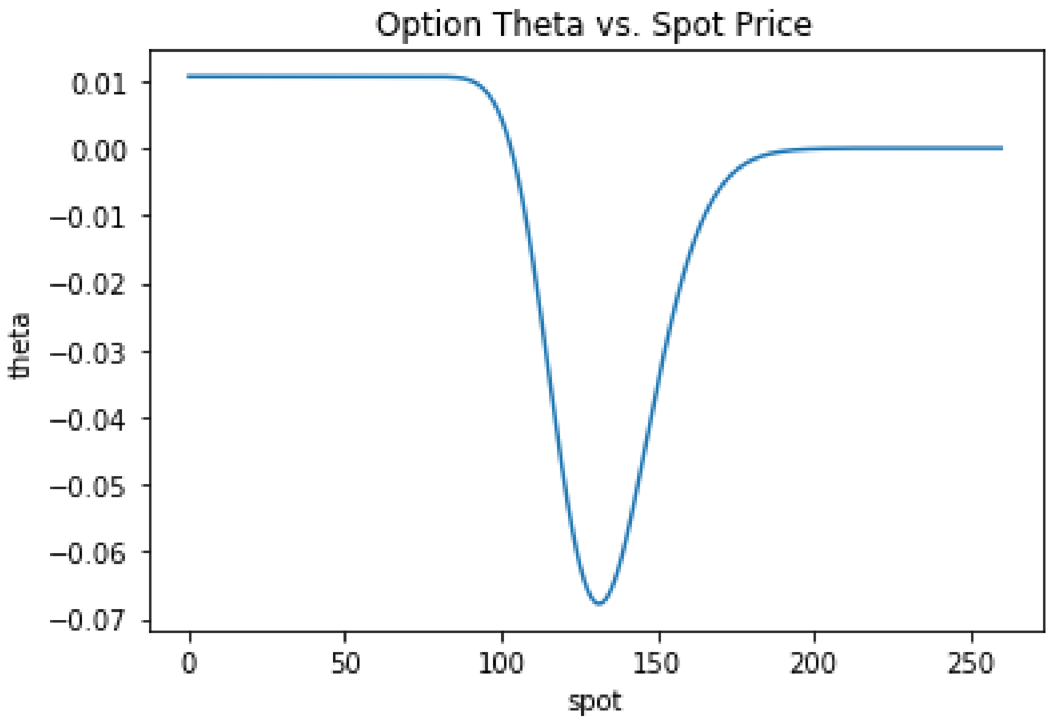

```
In [92]: option5.plot_gamma(0, 35, ir)
```



```
In [93]: option5.plot_vega(0, 35, ir)
```




```
In [95]: option5.plot_theta(0, 35, ir)
```



Problem 2

(a)

We begin by storing some of the initial data, and instantiate an option object:

```
In [43]: stock_price = 160
         expiration = 2/12
         volatility = 15
         ir = 6
         position = 100
         call = op.option(strike=160, expiry=expiration, type='call')
```

(i)

The cost to enter the position is the size of the position (100 shares) times the call premium:

```
In [44]: cost = position * call.price(spot=stock_price, time=0, vol=volatility, rate=ir)
         print(cost)

473.64976513917867
```

On the other hand, the cost to enter into an equivalent (ie the same exposure) cash position is simply the size of the position (the number of shares) times the stock price:

In [45]: position * stock_price

Out[45]: 16000

We see that the equivalent cash position has more than 30 times the cost of the option position. Yet the exposure in the option position is almost identical the same, provided that the stock price goes up. Since this is a long position, this represents a directional bet that the stock is going up in price. If the stock price increases by $10, with the option expiring $10 in the money, we will make the same profit as if we had held the outright stock position, yet with option we attained this exposure for less than a 30th the cost of the cash position.
On the other hand, this position comes with a much higher degree of risk. If the stock price ends up going down, even if only by a few pennies, and the option expires out of the money, our entire investment can get wiped out.

(ii)

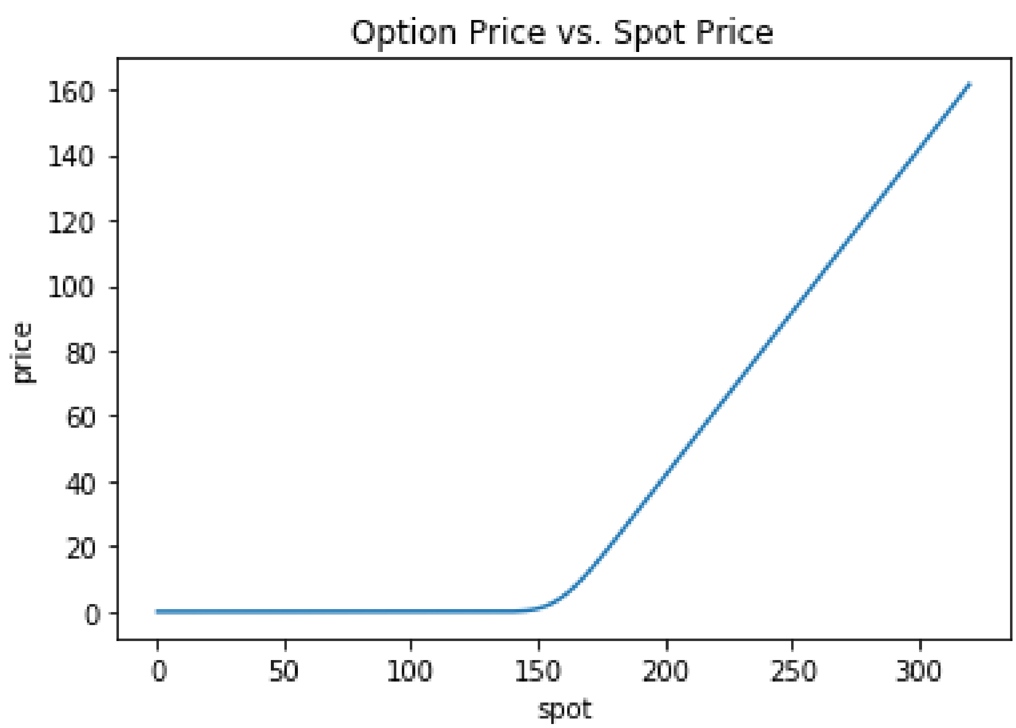
We calculate the option delta:

```
In [46]: call.delta(spot=stock_price, time=0, vol=volatility, rate=ir)
```

```
Out[46]: 0.576879931083044
```

and we plot the option price

```
In [47]: call.plot_price(time=0, vol=volatility, rate=ir)
```



From this plot, we see that once the stock price rises much above the initial price of $160 the option price is virtually a straight line of slope 1, and the option behaves almost exactly the same as the stock itself. In particular, any gains in the stock will result in an almost identical gain in the option. Of course this all presupposes that the stock price rises and the option moves further in-the-money.
The delta of about 0.58, which is simply the slope of the option premium at $160 does not reflect this, because of the nonlinearity of the premium. The ATM point is pretty much exactly at the kink in the plot where the slope has not yet approached it's limiting value. However, it does not take long to get there, as one can see by calculating a few deltas for increasing stock prices:

```
In [48]: call.delta(spot=162, time=0, vol=volatility, rate=ir)

Out[48]: 0.6542339846074849

In [49]: call.delta(spot=164, time=0, vol=volatility, rate=ir)

Out[49]: 0.7247952504292139

In [50]: call.delta(spot=166, time=0, vol=volatility, rate=ir)

Out[50]: 0.7867187268959027

In [51]: call.delta(spot=168, time=0, vol=volatility, rate=ir)

Out[51]: 0.8390736926924574

In [55]: call.delta(spot=170, time=0, vol=volatility, rate=ir)

Out[55]: 0.8817764197652076

In [57]: call.delta(spot=175, time=0, vol=volatility, rate=ir)

Out[57]: 0.9512683950013326

In [58]: call.delta(spot=180, time=0, vol=volatility, rate=ir)

Out[58]: 0.9828830932954439
```

Using the delta to approximate the change in the option price for a given change in the share price is equivalent to approximating the option price by it's straight tangent line at the point on the curve where the delta is calculated, and the above results suggest that, for this situation at least, using the delta at $160 is likely to seriously underestimate the increase in the option value. We will demonstrate this by following the suggested procedure of using the delta to approximate the change in the option value under a proposed change in the spot price, and then checking this against the actual change in the option value by repricing the option.
We carry out these two procedures under a $10 increase in the spot price of the stock from $160 to $170. First, the delta approximation:

```
In [59]: PL_delta = 10 * call.delta(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_delta)
```

```
5.76879931083044
```

And we compare this with the fully accurate calculation from repricing the option:

```
In [60]: PL_full = call.price(spot=170, time=1/12, vol=volatility, rate=ir) - \
         call.price(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_full)
```

```
6.262958128900394
```

And, as expected, the delta approximation underestimates the true P&L of the option position. Note, also, that, the P&L have suffered to some degree under time decay, so the delta approximation has probably underestimated the position's profit by even more.
Now we consider the case of a $10 loss on the stock. The delta approximation for the option price is:

```
In [13]: PL_delta = -10 * call.delta(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_delta)
```

```
-5.76879931083044
```

And from full repricing:

```
In [15]: PL_full = call.price(spot=150, time=1/12, vol=volatility, rate=ir) - \
         call.price(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_full)
```

```
-4.477341470187332
```

In this case, the delta approximation overestimates the loss in the option position, which again follows directly from approximating the option price by it's tangent line at the strike price.
Under the scenario where the stock price has increased by $10 after 1 month the delta is now

```
In [16]: call.delta(spot=170, time=1/12, vol=volatility, rate=ir)
```

```
Out[16]: 0.9378761599666189
```

The closer the delta is to 1, the closer the option is to behaving exactly like the stock. With delta of about 0.94, the option value will go up by 94 cents for every dollar the stock goes up. Since this is a bullish position on the stock, ie a bet that the stock will go up, if we still have this bullish conviction, we might want to hold on to the position. On the other hand, perhaps our view is that this stock will not keep going up for much longer. Or, perhaps we are uncertain, and would just like to lock in the profits we have made so far. So if these were our views, we might want to sell the option now.
Now, we consider the case where the stock price has dropped $10. The new delta is

```
In [18]: call.delta(spot=150, time=1/12, vol=volatility, rate=ir)
```

```
Out[18]: 0.0879746973930165
```

In this case our bullish bet has, so far, not panned out, since the stock has lost $10 per share. Moreover, the option is now so out of the money that, with a delta of about 0.09, the option value will only go up by 9 cents for every dollar the stock price rises. Even if we are still bullish on the stock, the stock will have to rise dramatically before the option position will make much money. It might be a good idea to cut our losses now, and sell the option. But, if we still have a firmly bullish conviction on the stock, we could decide to hold on and hope the stock recovers from this loss, and ultimately, gains over its initial price.

(iii)

The theta of the option is

```
In [25]: call.theta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[25]: -0.045923001947846306
```

This means the option position will lose about 4.6 cents per day if all other market factors stay fixed. Extrapolating this out for 1 month, the option will lose $30 \times 0.046 = \$1.38$. We check if this holds out by full repricing the option. The actual P&L will be

```
In [26]: call.price(spot=160, time=1/12, vol=volatility, rate=ir) - \
         call.price(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[26]: -1.562295150450467
```

So our estimate of the time decay is not too bad, but stil off by a bit more than 10%. To understand this, remember that the theta is a linear approximation to the time dependence of the option price. It is, after all, just the derivative of the option price with respect to time. To see if it helps explain why it underestimates the time decay of the option, we'll calculate the theta for 10, 20, and 30 days ahead:

```
In [27]: call.theta(spot=160, time=10/365, vol=volatility, rate=ir)
```

```
Out[27]: -0.04889789991279146
```

```
In [28]: call.theta(spot=160, time=20/365, vol=volatility, rate=ir)
```

```
Out[28]: -0.05291532781405266
```

```
In [29]: call.theta(spot=160, time=30/365, vol=volatility, rate=ir)
```

```
Out[29]: -0.058767668145517434
```

So the theta will increase (in absolute value) as we proceed into the month, implying a faster time decay than what is implied by the theta simply on the first date of the month. This explains why the theta approximation underestimated the loss on the option from time decay.

(iv)

The vega of our option is

```
In [31]: call.vega(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[31]: 0.2557341829255597
```

Vega is the sensitivity of the option price to implied volatility, but implied volatility is simply a reflection of the traded prices for options, so of course, if we hold an option position, we are exposed to the option market, that is, we have exposure to implied volatility. Our positive vega exposure reminds us that our position could lose money if implied volatililty declines. What this amounts to is that our long call position could lose money if the option market's expectations about future volatility decline.

(v)

We assume that our intentions in this situation are the same: we are implementing a long position on the underlying stock using options, that is, we expect the stock price to rise and we want to take a position that will profit if these expectations are realized. We instantiate a new call object using a strike of $170:

```
In [36]: call2 = op.option(strike=170, expiry=expiration, type='call')
```

Given our stated objectives, the primary advantage of an out of the money option is price. The full cost to enter into this option position, on 100 shares, is

```
In [37]: position * call2.price(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[37]: 115.18504152001654
```

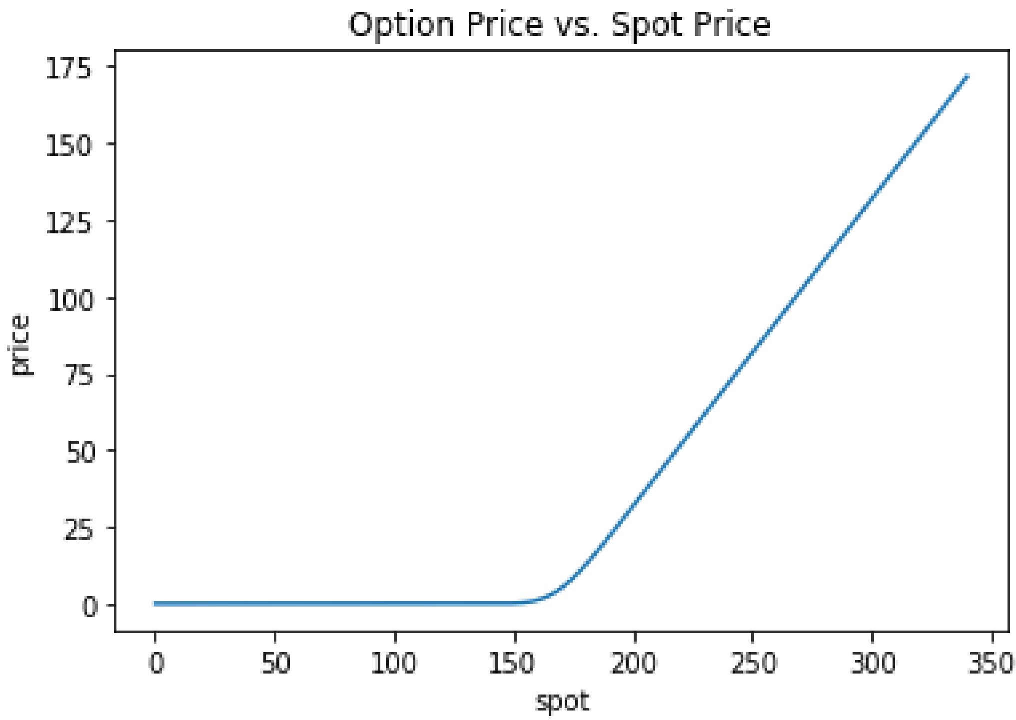
So it is less than 1/4 the cost to enter into this out-of-the-money option than tne $474 for the at-the-money option, and we still have an exposure on 100 shares of the underlying stock.
On the other hand, we do not, quanitatively, have the same exposure that we would have with the at-the-money option. the delta of this position is

```
In [38]: call2.delta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[38]: 0.2129933529279303
```

So a $1 rise in the stock only implies a 21 cent profit for the option. We are still long the stock, but we require a much larger change in the stock price before we we will enjoy significant profits. The situation is further elucidated by viewing a plot of the price:

```
In [39]: call2.plot_price(time=0, vol=volatility, rate=ir)
```

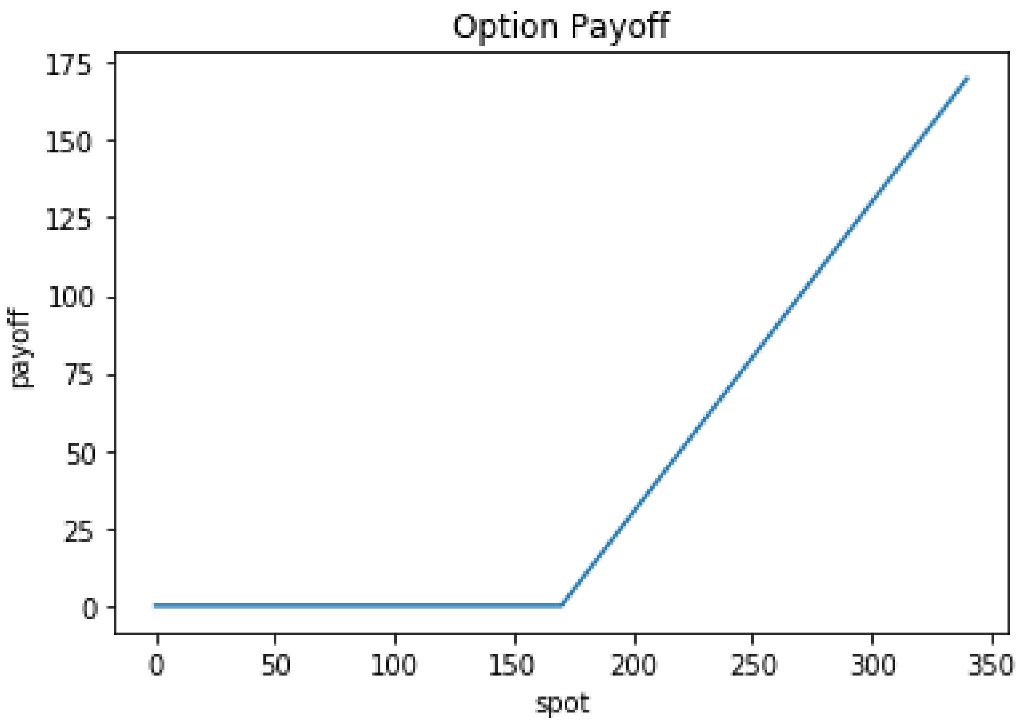


Considering a few specific numerical values, we have

```
In [40]: call2.price(spot=160, time=0, vol=volatility, rate=ir)

Out[40]: 1.1518504152001654

In [41]: call2.price(spot=162, time=0, vol=volatility, rate=ir)

Out[41]: 1.6400032274095295

In [42]: call2.price(spot=164, time=0, vol=volatility, rate=ir)

Out[42]: 2.2627182518454134

In [43]: call2.price(spot=166, time=0, vol=volatility, rate=ir)

Out[43]: 3.0320875039039947

In [44]: call2.price(spot=168, time=0, vol=volatility, rate=ir)

Out[44]: 3.95499370577852

In [45]: call2.price(spot=170, time=0, vol=volatility, rate=ir)

Out[45]: 5.0325287546037885
```

So a \$4 rise in the stock price is required before the option position yields a \$1 profit, a \$6 rise in the stock for \$2 on the option, and \$10 on the stock for \$4 on the option. So when buying out of the money options, we are betting on a much larger rise in the stock price before our position will pay off. There is also a significant risk that the option will expire out of the money, as illustruted by the payoff:

```
In [46]: call2.plot_payoff()
```



If the stock does not rise by \$20 before expriation in 2 months the option will expire worthless, and we will lose our entire investment. Furthermore, the option's theta:

```
In [47]: call2.theta(spot=160, time=0, vol=volatility, rate=ir)

Out[47]: -0.028815004088278265
```

reminds us that if we just sit on this position and neither the stock price nor the option market moves significantly, we will slowly bleed losses through time decay. So in order to not lose money on an out-of-the-money option position, there will need to be a sharp rise in the stock price, and in relatively short order. That is the risk inherent in picking up the option so cheaply.

(vi)

We continue to view the option position as a bullish play on the stock, but now we will see what happens when we implement it with an in-the-money option. We instantiate a new option object:

```
In [5]: call3 = op.option(strike=150, expiry=expiration, type='call')
```

For an in-the-money option, we expect more expense necessary to enter into the position:

```
In [6]: position * call3.price(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[6]: 1200.3519989010954
```

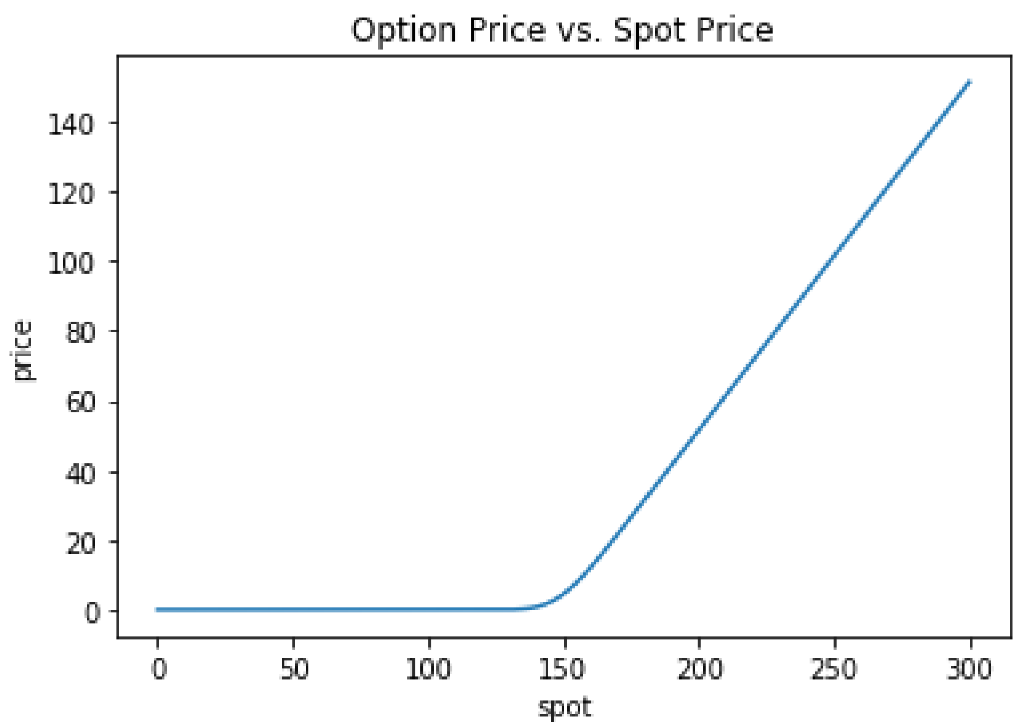
The option is more expensive now, but still much cheaper than taking an equivalent cash position in the stock.
To probe the potential profitability of this investment, we calculate the delta:

```
In [7]: call3.delta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[7]: 0.8939528959522145
```

and we plot the premium:

```
In [9]: call3.plot_price(time=0, vol=volatility, rate=ir)
```



While the cost to enter this portfolio is higher than either the out-of-the-money option or the at-the-money option, in return we have greater exposure to the underlying stock. The option makes 89 cents for every dollar the stock goes up. And if our expectations pan out and the stock does increase, the situation will only get more favorable, as we can see by calculating some of the values of the option price:

```
In [ ]: call3.price(spot=160, time=0, vol=volatility, rate=ir)
```

```
In [11]: call3.price(spot=162, time=0, vol=volatility, rate=ir)

Out[11]: 13.825590656167066

In [12]: call3.price(spot=164, time=0, vol=volatility, rate=ir)

Out[12]: 15.704085651522917

In [13]: call3.price(spot=166, time=0, vol=volatility, rate=ir)

Out[13]: 17.623513544879387

In [14]: call3.price(spot=168, time=0, vol=volatility, rate=ir)

Out[14]: 19.571603939730807

In [15]: call3.price(spot=170, time=0, vol=volatility, rate=ir)

Out[15]: 21.53909229263354
```

So if the stock price rises by even a few dollars we are getting closer and closer to earning almost a dollar for every dollar the stock goes up. So, for \$1200, only 7.5% of the \$16000 required for an equivalent cash position, we have practically the same exposure.
On the other hand, there are still other risks an investor must consider. While if our prediction comes true, and the stock price goes up, we stand to profit significantly from this position, we have much greater downside risk in this position if the stock ends up dropping. There is still a significant risk that the stock could drop by \$10 in 2 months, in which case the option will expire worthless and we will lose our whole investment.
In addition, there is time decay, which is measured by the theta:

```
In [19]: call3.theta(spot=160, time=0, vol=volatility, rate=ir)

Out[19]: -0.03628791094454507
```

If we scale this up to our position of 100 options, this implies that we will lose \$3.63 per day on the position from time decay, or \$218 over the 2 months remaining in the option's life (keep in mind this calculation, based only on extrapolating the theta based estimate for 2 months, may underestimate the true time decay). Given our \$1200 initial investment, this may sound like an acceptable risk. The point, though, is that if we simply hold this position, we can only lose money unless the stock price moves in our favor (or if implied volatility increases due to developments in the options market). One should not take a position like this, unless one has firm conviction in the prediction that the stock price will rise.

(b)

We will carry out a similar analysis as in part (a) for puts. Once again, we gather our starting data and intantiate a put object.

```
In [22]: stock_price = 160
         expiration = 2/12
         volatility = 15
         ir = 6
         position = 100
         put = op.option(strike=160, expiry=expiration, type='put')
```

In many ways, the considerations an investor must take into account with the put position are the same, or similar, as for the call. However, one major difference is that with the put, we are taking a short position on the stock, that is, with the put, we are making a bearish bet, a bet that the stock price will drop over the life of the option.

(i)

The cost to enter the position is the size of the position (100 shares) times the put premium:

```
In [23]: cost = position * put.price(spot=stock_price, time=0, vol=volatility, rate=ir)
         print(cost)
```

```
314.4471051258705
```

When comparing with the equivalent cash position, one must remember that by entering into a put we are now shorting the stock. So the equivalent cash position to compare to is a short sale of the stock. As far as the dollar magnitude of the cash position, it is the same as for the long position:

```
In [102]: position * stock_price
```

```
Out[102]: 16000
```

We are accustomed to treating this amount as a cash receipt when carrying out arbitrage arguments to price securities. However, when analyzing things from an investor's point of view, the circumstances are different. While it is true that the short sale will generate a $16000 payment, the margin requirements of the exchange will require most or all of these proceeds to be kept in a margin account. In fact, regulations, as well as the policies of the exchange itself, will usually require us to deposit even more funds in our margin account, possibly as much as 25-50% or possibly even more. So, entering into the short sale of the stock will likely require us to deposit an amount of cash roughly in the range $400-$800. When compared to this, one can see the attraction of establishing the short using puts, requiring, for this position, only the payment of the $314 premium

As for the call, there are risks implicit in the put position that are fundamental option risks, that the pure cash position is not exposed to. There is the possibility of the stock price rising, if our forecast turns out to be wrong, resulting in losses on the put position, and the risk the puts could expire worthless. There is also time decay as well as vega risk, ie the risk in the options market itself. However, in this case, the uncovered short cash position we are comparing with is such a risky position, that one could argue that risk consideratinos alone justify entering into the option position rather than the cash position.

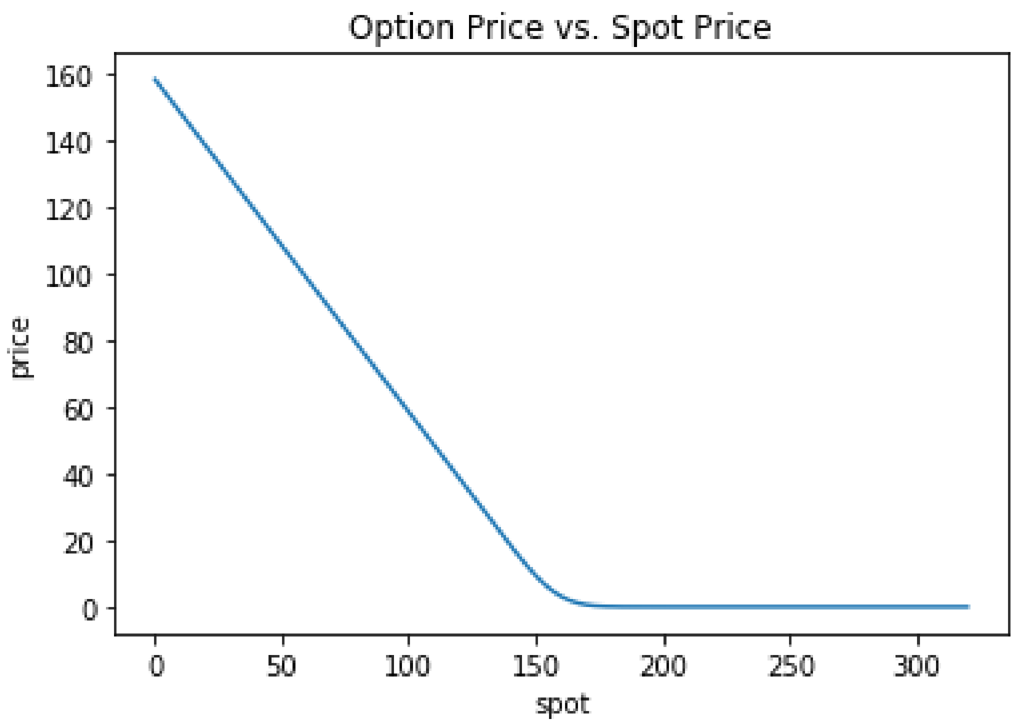
(ii)

We calculate the option delta:

```
In [24]: put.delta(spot=stock_price, time=0, vol=volatility, rate=ir)

Out[24]: -0.4231200689169559
```

and we plot the option price

```
In [25]: put.plot_price(time=0, vol=volatility, rate=ir)
```



Perhaps the most striking thing about this position when compared to the call position is that it is a short position on the underlying. This is reflected both in the delta (because it is negative) and in the plot, where clearly the option position loses money if the stock price increases. These observations remove any doubt that buying a put implements a short position on the stock.
So we look at these circumstances from the point of view of an investor who expects the stock price to drop and wants to profit from such a drop. The delta of -0.42 means that the option position will profit by 42 cents if the stock drops by $1. However, similar to the case of a call position on a rising stock, if the stock drops, and the put moves further into the money (corresponding to the almost linear portion and negatively sloped portion of the call price) we expect the delta to approach -1, and we will eventually reach a point where a $1 drop in the stock price results in nearly a $1 increase in the put value. To illustrate this, we calculate the delta over a range of prices moving further into the money:

```
In [26]: put.delta(spot=158, time=0, vol=volatility, rate=ir)
```

```
Out[26]: -0.5045848291252172
```

```
In [27]: put.delta(spot=156, time=0, vol=volatility, rate=ir)
```

```
Out[27]: -0.586877551613205
```

```
In [28]: put.delta(spot=154, time=0, vol=volatility, rate=ir)
```

```
Out[28]: -0.666486503566712
```

```
In [29]: put.delta(spot=152, time=0, vol=volatility, rate=ir)
```

```
Out[29]: -0.7401143732732686
```

```
In [30]: put.delta(spot=150, time=0, vol=volatility, rate=ir)
```

```
Out[30]: -0.8051031908243667
```

```
In [32]: put.delta(spot=145, time=0, vol=volatility, rate=ir)
```

```
Out[32]: -0.921260561944751
```

```
In [33]: put.delta(spot=140, time=0, vol=volatility, rate=ir)
```

```
Out[33]: -0.9765188791418613
```

Using the delta to approximate the change in the option price for a given change in the share price is equivalent to approximating the option price by it's straight tangent line at the point on the curve where the delta is calculated, and the above results suggest that, for this situation at least, using the delta at $160 is likely to seriously underestimate the increase in the option value. We will demonstrate this by following the suggested procedure of using the delta to approximate the change in the option value under a proposed change in the spot price, and then checking this against the actual change in the option value by repricing the option.
Just like in the case of the call, using the delta to approximate the gain in the put position is the same as approximating the put price by it's tangent line at the point the delta is calculated, and the same geometric condsiderations as for the put lead us to expect this to underestimate the profit from a dropping stock price. To check this, we again compare the P&L calculated using the delta approximation to one based on full repricing of the option for a $10 drop in the stock price:

```
In [35]: PL_delta = -10 * put.delta(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_delta)
```

```
4.231200689169559
```

And we compare this with the fully accurate calculation from repricing the option:

```
In [36]: PL_full = put.price(spot=150, time=1/12, vol=volatility, rate=ir) -\
         put.price(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_full)
```

```
6.316681800774887
```

Once again, the delta approximation underestimates the true P&L.
Next we calculate the P&L for a $10 gain in the stock implying a loss in the option position. The delta approximation for the option price is:

```
In [38]: PL_delta = 10 * put.delta(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_delta)
```

```
-4.231200689169559
```

And from full repricing:

```
In [40]: PL_full = put.price(spot=170, time=1/12, vol=volatility, rate=ir) - \
         put.price(spot=160, time=0, vol=volatility, rate=ir)
         print(PL_full)
```

```
-2.9430186001373393
```

In this case, the delta approximation overestimates the loss in the option position, which again follows directly from approximating the option price by it's tangent line at the strike price.
The analysis of what investment decisions should be made under either of the proposed scenarios is similar to the call. The only difference is that the put will be out-of-the-money if the stock gains and in-the-money if the stock falls. But, similar to the case of the call, once we have fallen deeply out of the money, the move in the stock price will have to reverse itself significantly before we will start to recover much of the money we have lost. And if the stock drops and the option is now in-the-money then our exposure to the stock is almost one for one, and if we expect the stock to keep dropping we might want to hold on to the option position. But if we expect the stock has exhausted the potential to drop much further, or if we just want to lock in the profits we have gained so far, we might want to sell the option now. In either case, the decision continue to hold or to liquidate, fully or partially, the position depends on the strenght of our conviction that the stock will drop or continue to drop.

(iii)

Similar to the call, the put has theta exposure, ie it is subject to time decay. The theta of this option position is

```
In [41]: put.theta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[41]: -0.01988333508759421
```

This indicates that, if the stock price and implied volatility remain fixed and we just hold the position, the put will lose about 2 cents a day, implying a loss of about $1.20 over the two months until expiration. Once again, we suspect this linear approximation may be innaccurate, so we check the loss using a full repricing approach.

```
In [44]: put.price(spot=160, time=60/365, vol=volatility, rate=ir) -\
         put.price(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[44]: -2.697885177270848
```

And once again, the theta based approximation is a fair bit off, and underestimates how much the option will lose from time decay. As for the call, this is due to the inaccuracy of approximating the nonlinear option price by a straight line.
In any case, our put is exposed to time decay risk, and this must be a consideration in our investment decisions.

(iv)

The vega of our option is

```
In [45]: put.vega(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[45]: 0.2557341829255597
```

As in the case of the call, vega exposure means exposure to the option market itself and it also a risk factor we must think about in making our investment decisions.

(v), (vi)

The considerations for an out of the money or in the money put are similar to the considerations for calls. An out of the money put, would for instance be one with a strike price of $150. Instantiating an option object like this,

```
In [48]: put2 = op.option(strike=150, expiry=expiration, type='put')
```

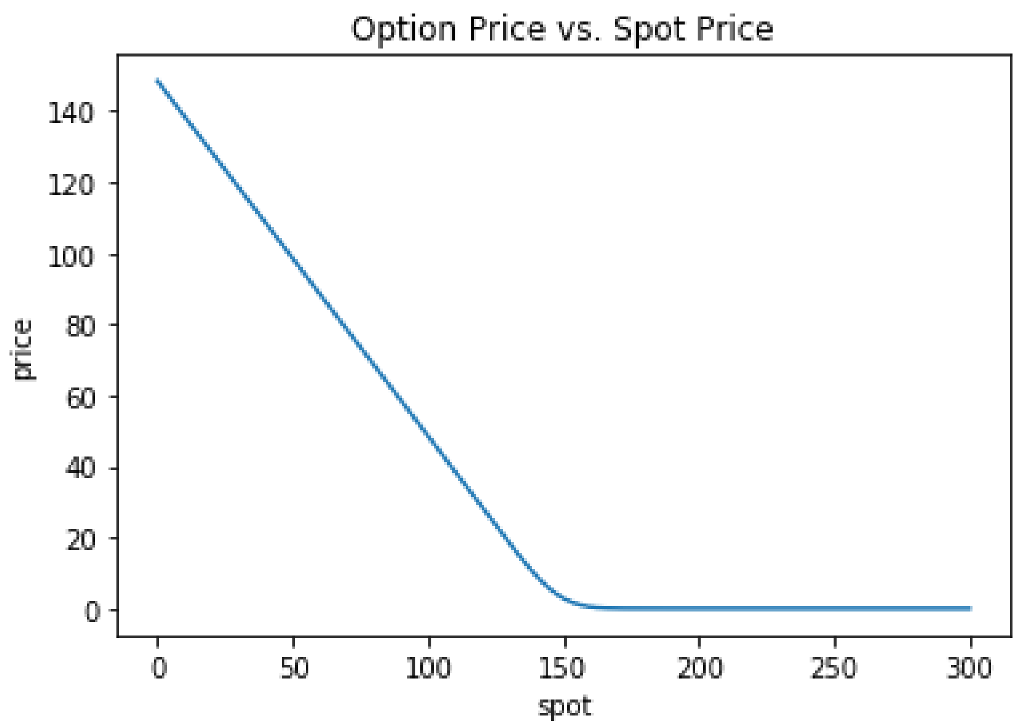
we may now compute the delta

```
In [49]: put2.delta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[49]: -0.1060471040477855
```

which shows at this level of moneyness the stock will have to decline significantly before the option position will profit much. This is illustrated further by inspecting a plot of the price

```
In [50]: put2.plot_price(time=0, vol=volatility, rate=ir)
```



The stock price will have to decline about $10 (which is, of course, down to the strike price) before the positon will start paying much profit.
An in the money position would, for example, be one with a strike price of $170:

```
In [51]: put3 = op.option(strike=170, expiry=expiration, type='put')
```

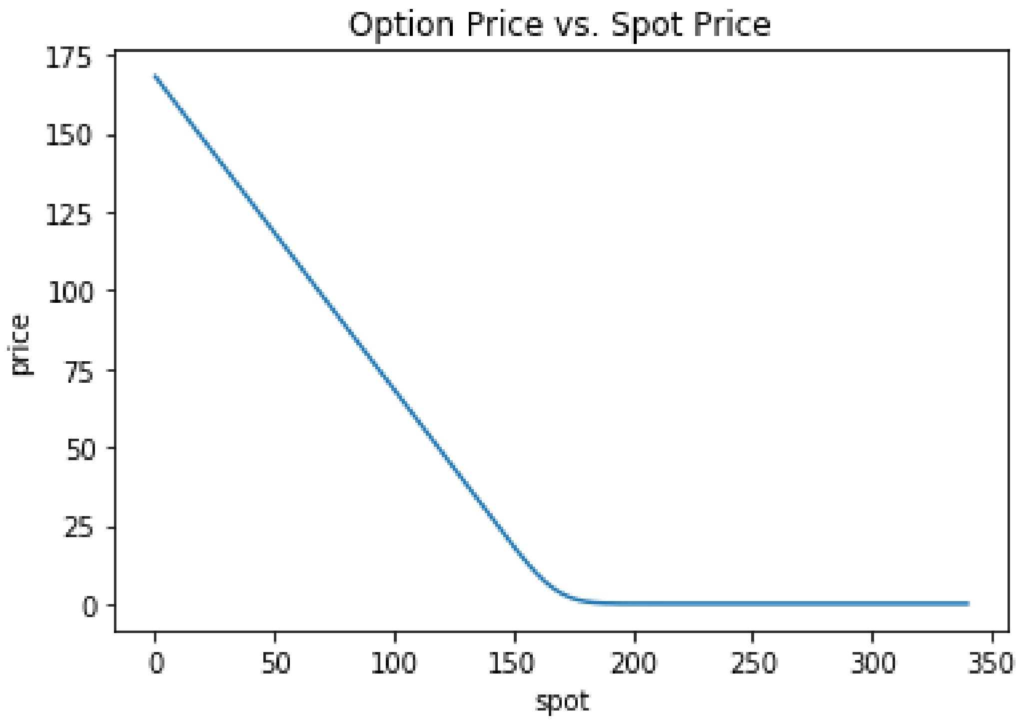
Calculating the delta

```
In [52]: put3.delta(spot=160, time=0, vol=volatility, rate=ir)
```

```
Out[52]: -0.7870066470720697
```

and plotting the price

```
In [53]: put3.plot_price(time=0, vol=volatility, rate=ir)
```



shows that, like the in-the-money call, this position will be much more profitable, and much sooner.
Of course, the out-of-the-money option will be much cheaper to enter into in the first place, whereas the in-the-money put will cost more, so these considerations must be remembered as well. But, like the call, it is matter of balancing the cost with our conviction about how much the stock price will move while we hold the position.

Problem 3

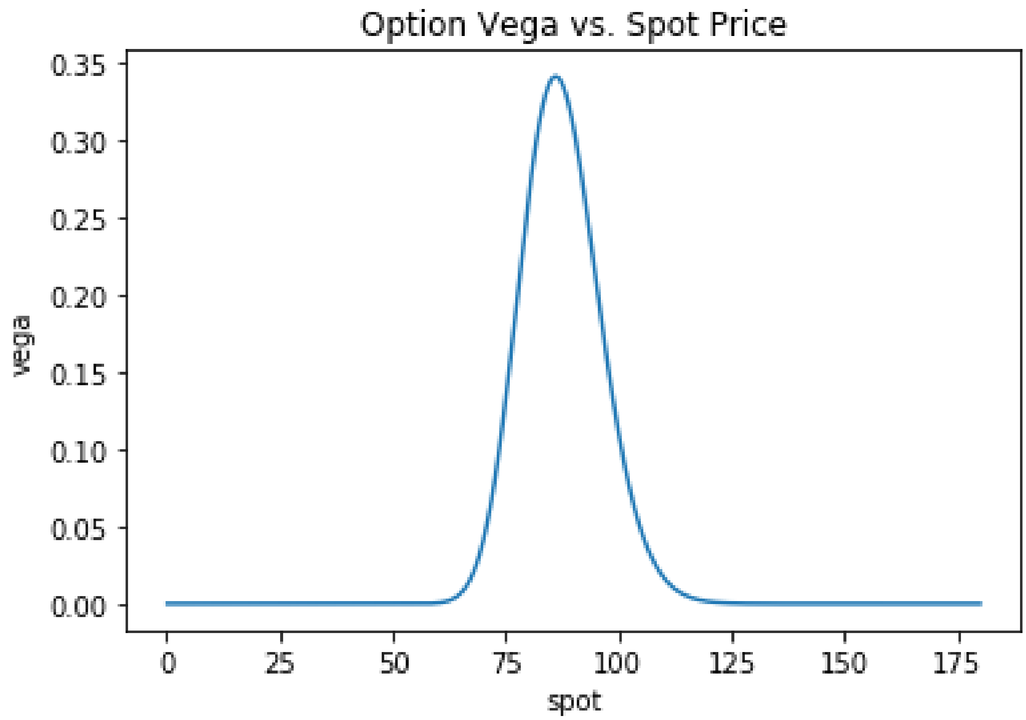
(a)

We will attempt to construct an option position that takes advantage of our volatility forecast. We will implement the position with calls, as done in lecture–though a volatility position could be implemented just as well with puts.

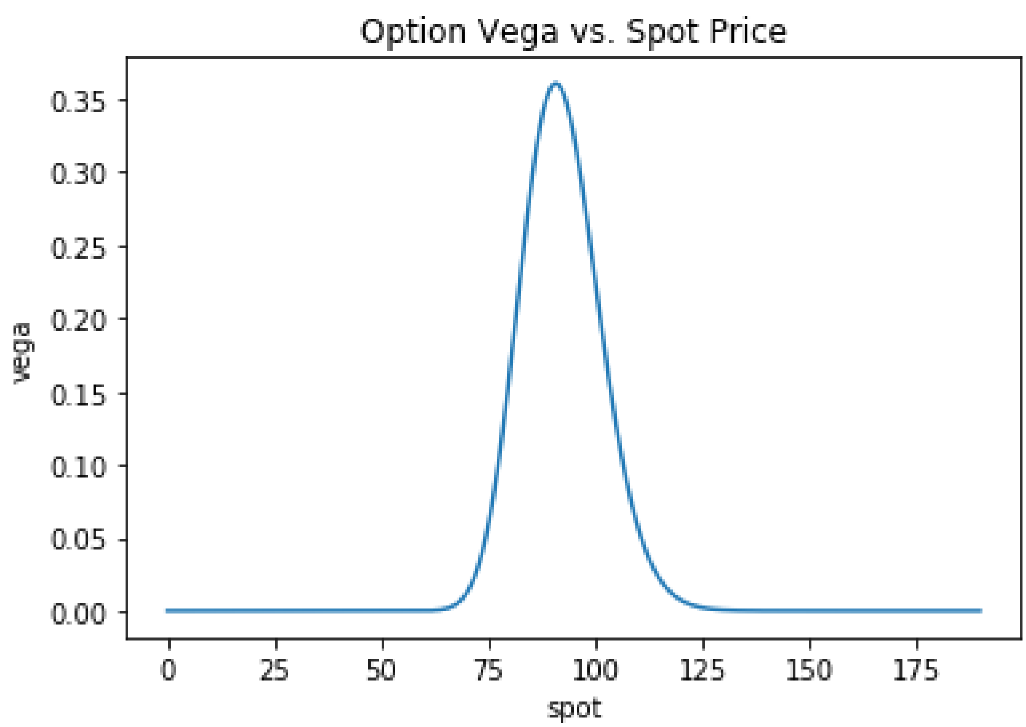
(i)

We first set up a position with options alone, and ignore the delta exposure of the position. This will demonstrate how much extra risk factors, like the delta (exposure to the underlying) and the theta (time decay) can interfere with the attempt to obtain vega exposure.
Following the suggestions in the problem, we will inspect the vega of a few candidate options to identify the one with the maximum exposure to the implied volatility at our spot price of $100:

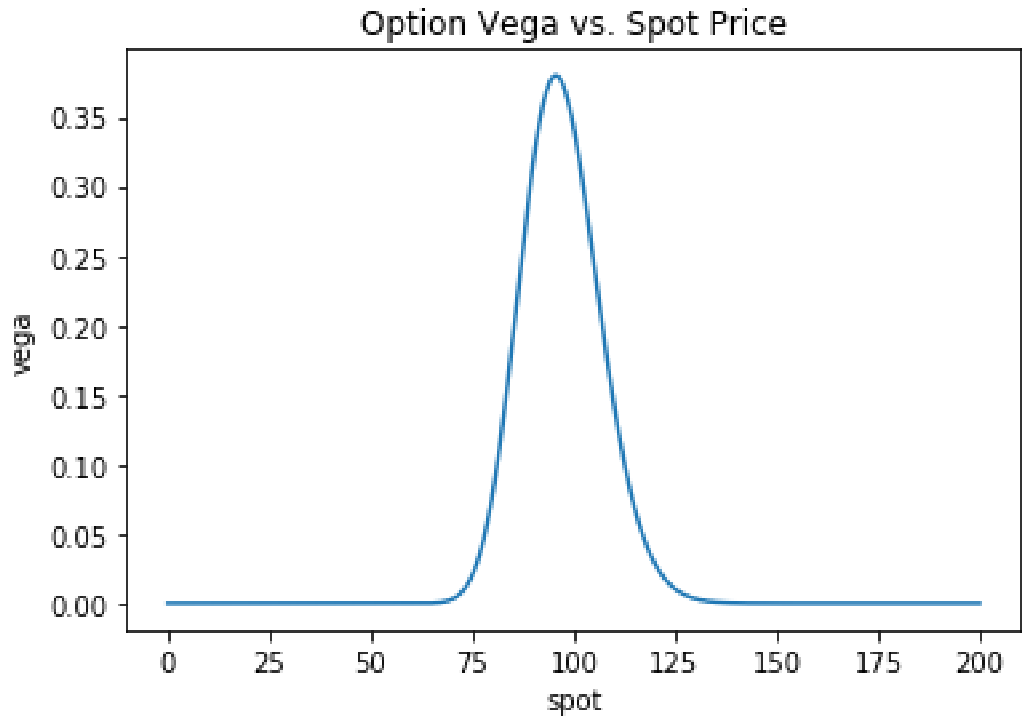
```
In [2]: call90 = op.option(strike=90, expiry=1.0, type='call')
        call90.plot_vega(time=0, vol=10, rate=5)
```



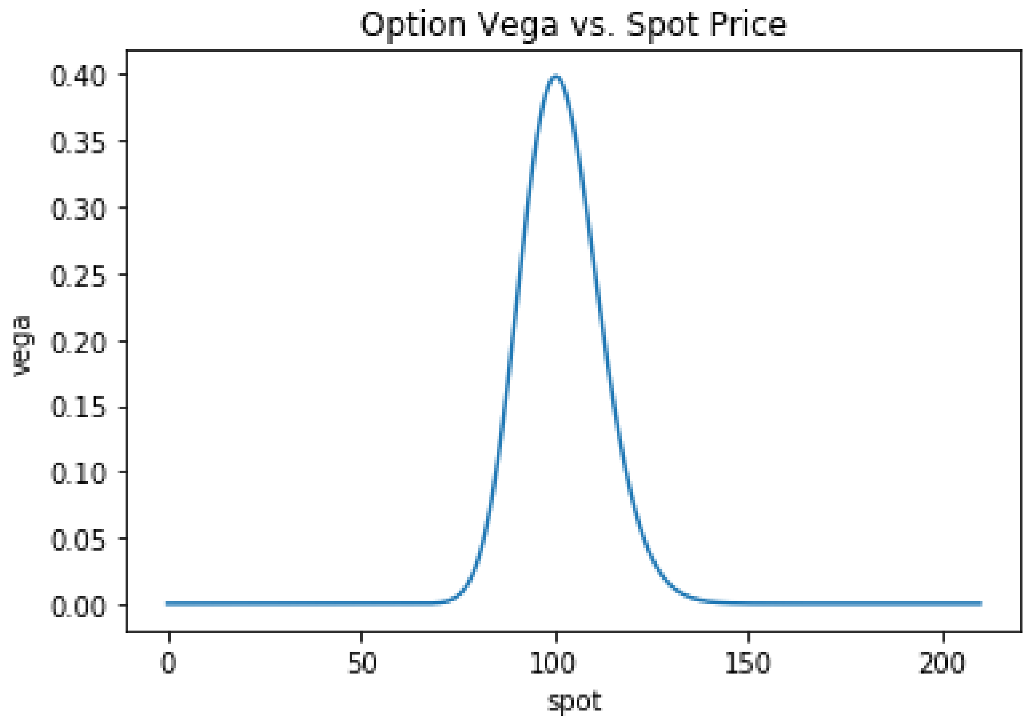

```
In [3]: call95 = op.option(strike=95, expiry=1.0, type='call')
        call95.plot_vega(time=0, vol=10, rate=5)
```



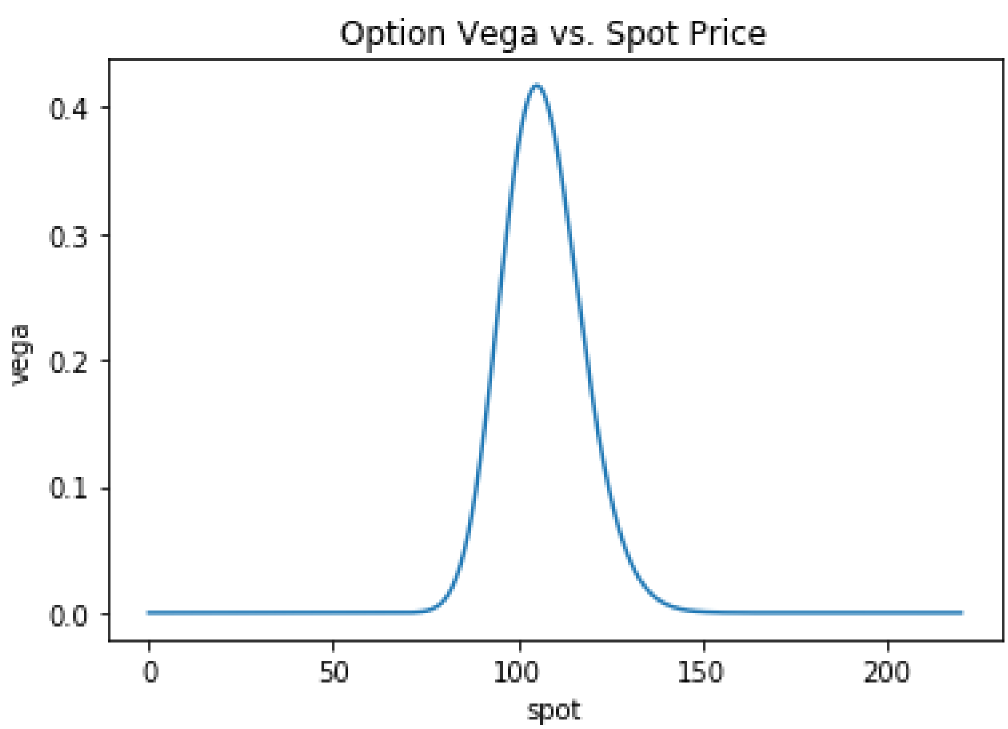
```
In [4]: call100 = op.option(strike=100, expiry=1.0, type='call')
        call100.plot_vega(time=0, vol=10, rate=5)
```




```
In [5]: call105 = op.option(strike=105, expiry=1.0, type='call')
        call105.plot_vega(time=0, vol=10, rate=5)
```




```
In [6]: call110 = op.option(strike=110, expiry=1.0, type='call')
        call110.plot_vega(time=0, vol=10, rate=5)
```



Based on these plots, the call with a strike price of $105 seems to come closest to maximizing the vega at the current spot price of $100, so we will use these calls as our main instrument to obtain the vega exposure we are seeking.
Thus, our portfolio will consist of 1000 call options on our underlying stock, with a strike price of $105 and expiring in 1 year. Now we'll check the P&L of this portfolio in 6 months, under the 3 proposed scenarios: a spot price of $90, $100, and $110. Recall that in all 3 cases we are supposing that in 6 months the volatility has risen 40%

```
In [13]: PL1 = 1000*(call105.price(90, 0.5, 40, 5) - call105.price(100, 0, 10, 5))
         print(PL1)
```

```
1668.925856159376
```

```
In [14]: PL2 = 1000*(call105.price(100, 0.5, 40, 5) - call105.price(100, 0, 10, 5))
         print(PL2)
```

```
6171.770810660802
```

```
In [15]: PL3 = 1000*(call105.price(110, 0.5, 40, 5) - call105.price(100, 0, 10, 5))
         print(PL3)
```

```
12071.213225494383
```

As it turns out, the increase in implied volatility in the proposed scenario is so strong that the raw option position makes money in all 3 case, even though, in the first 2 the call is well out of the money. But the results are very uneven, with the profit in the case of a rising stock price being almost 8 times as much as the profit when the stock price drops by the same amount. This shows that the position has other significant exposures than just the vega.

(ii)

Now we will add to our option position a position in the underlying stock so as to create a delta neutral position. We need the delta of our option:

```
In [19]: delta = call105.delta(100, 0, 10, 5)
         print(delta)
```

```
0.5247577478203188
```

Thus, for a delta neutral position we must short

```
In [20]: delta * 1000
```

```
Out[20]: 524.7577478203189
```

shares. Rounding to the nearest whole numbers, we add a short position on 525 shares.
Our updated portfolio consists of 1000 call options and a short on 525 shares. We compute the P&L of our updated portfolio in each of the 3 given scenarios 6 months hence:

```
In [24]: PL1 = 1000*(call105.price(90, 0.5, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * (90 - 100)
         print(PL1)

6918.925856159376
```

```
In [25]: PL2 = 1000*(call105.price(100, 0.5, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * 0
         print(PL2)

6171.770810660802
```

```
In [26]: PL3 = 1000*(call105.price(110, 0.5, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * (110 - 100)
         print(PL3)

6821.213225494383
```

So the main effect of adding the delta hedging is to make our P&L much more uniform in all 3 cases. In other words, by adding delta hedging we largely insulate our portfolio from changes in the underlying stock price. Which is exactly what should happen, since delta is the exposure to the underlying price. If it is zero then, at least up to the best linear approximation, changes in the underlying price should not effect the value of the position.
In this case, we observe that the position enjoys more profit in the cases where the stock price moved a significant amount than when it remains fixed. This is a reflection of the fact that the position has positive gamma. Computing the gamma (note that the short stock position contributes nothing to the gamma, since it is linear):

```
In [28]: call105.gamma(100, 0, 10, 5)

Out[28]: 0.03981738196467981
```

The positive gamma indicates that the position's value, as a function of the underlying, has a convex U shape, so that profits are increased if the stock price moves, regardless of the direction of the move. This is one reflection of the fact that this position is long volatility.
If you compare the results calculated here with what we found in lecture, you will find they are reasonably close, but not exactly the same. This is because in lecture we used a strike price of \$100 in comparison to the \$105 we have used here. I will leave it as an exercise for the student to modify the calculations we have done here to use a \$100 strike and check that then you will exactly replicate the results from class.

(iii)

We calculate the theta of our position:

```
In [29]: 1000 * call105.theta(100, 0, 10, 5)

Out[29]: -12.088638322332146
```

So, at least initially, the position loses about $12 per day due simply to option time decay. Extrapolating this out to 6 months ($\approx$ 180 days) our position could have lost, roughly

```
In [31]: 180 * 12

Out[31]: 2160
```

from time decay alone. Using the theta approximation over a 6 month period is a crude approximation which likely underestimates the amount of time decay, but it shows that it has likely taken a significant bite out of our profits.

(iv)

We consider the same delta neutral position we implemented in part (ii). The only difference is we suppose the change in market factors happen in 3 days, not 6 months:

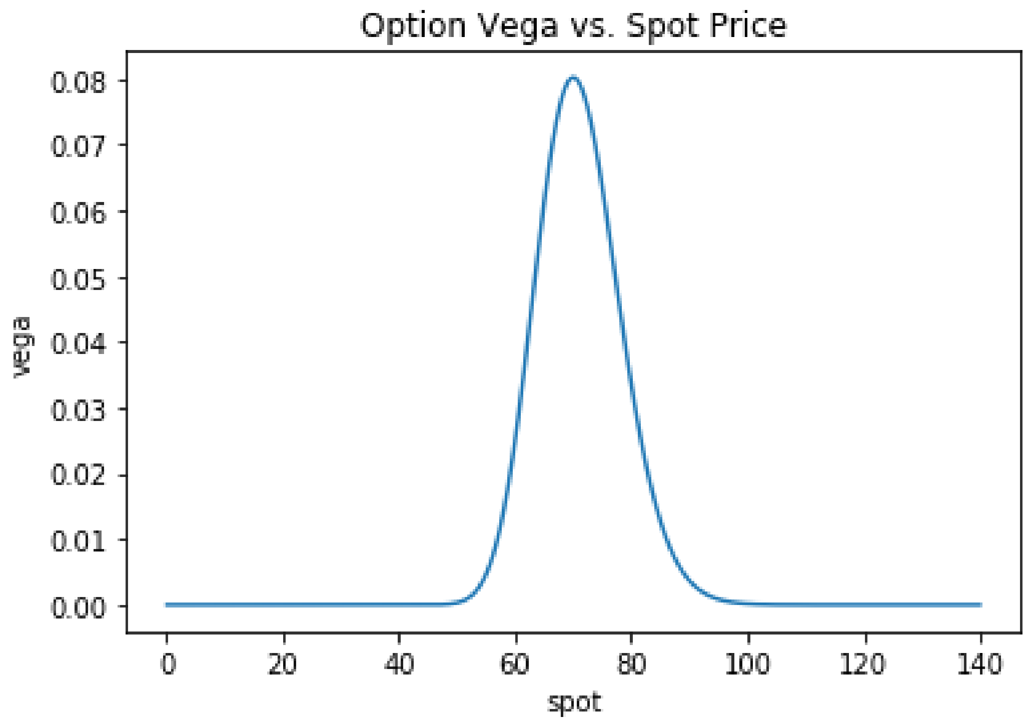
```
In [35]: PL1 = 1000*(call105.price(90, 3/365, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * (90 - 100)
         print(PL1)

11741.855283205708


In [36]: PL2 = 1000*(call105.price(100, 3/365, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * 0
         print(PL2)

11775.081562360263


In [37]: PL3 = 1000*(call105.price(110, 3/365, 40, 5) - \
                     call105.price(100, 0, 10, 5)) - 525 * (110 - 100)
         print(PL3)

12787.614157110129
```

The significantly better profits in this scenario are an indication that the time decay of this position is significant, and we have likely severely underestimated it by relying on the theta approximation alone.

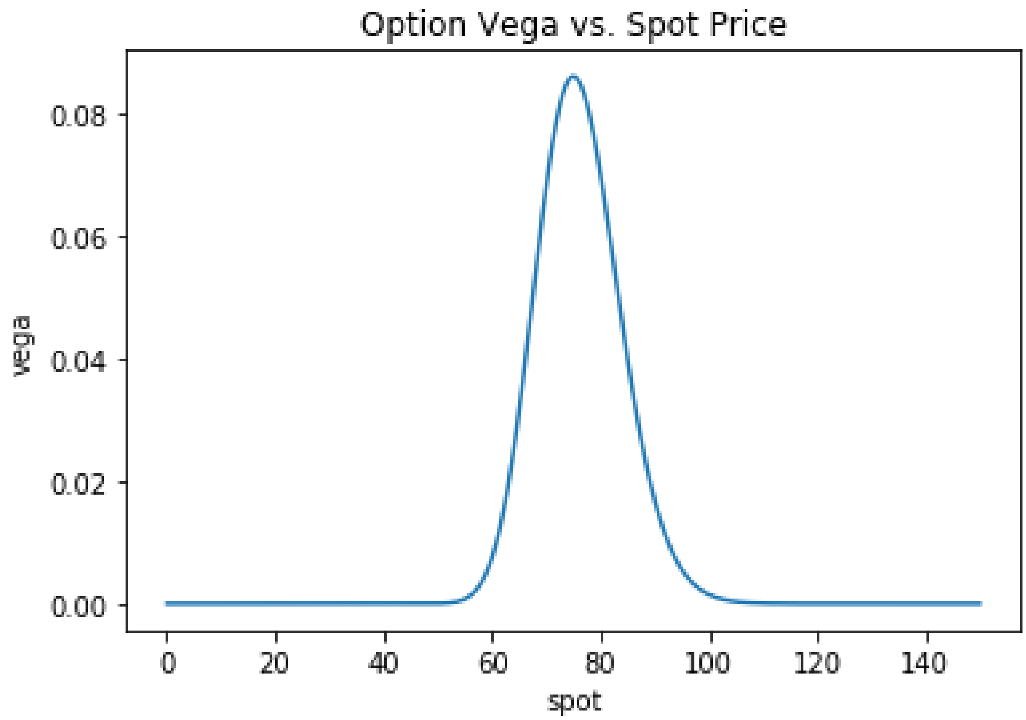
(b)

We start by plotting the vega of a few all options in the range where we expect to find a candidate option.

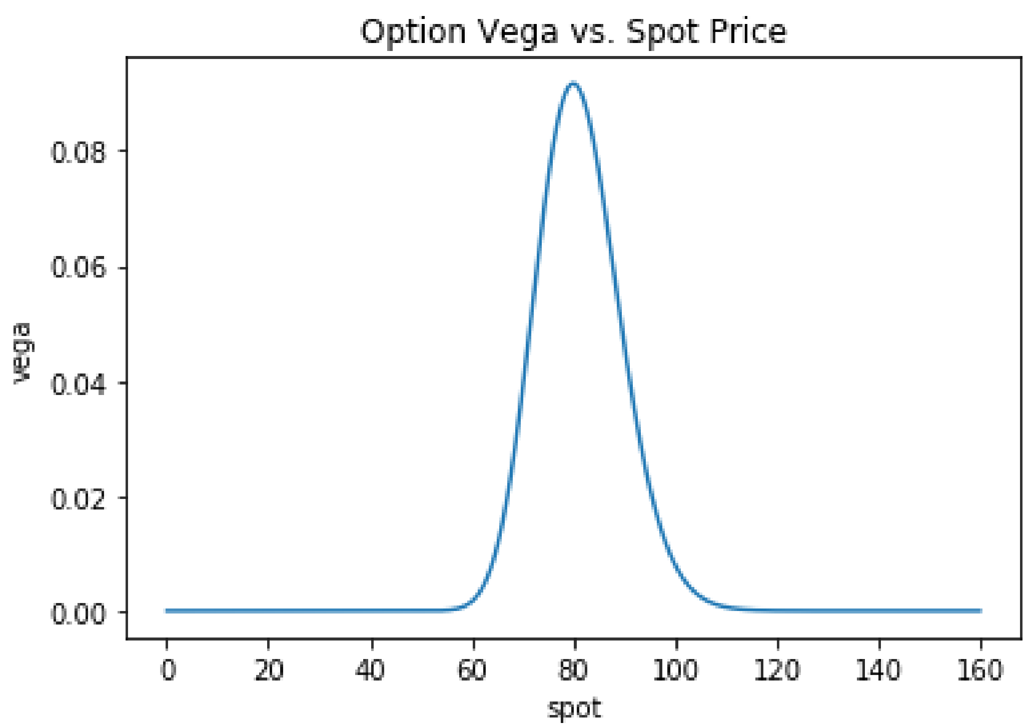
```
In [18]: call70 = op.option(strike=70, expiry=1/12, type='call')
         call70.plot_vega(time=0, vol=35, rate=6)
```



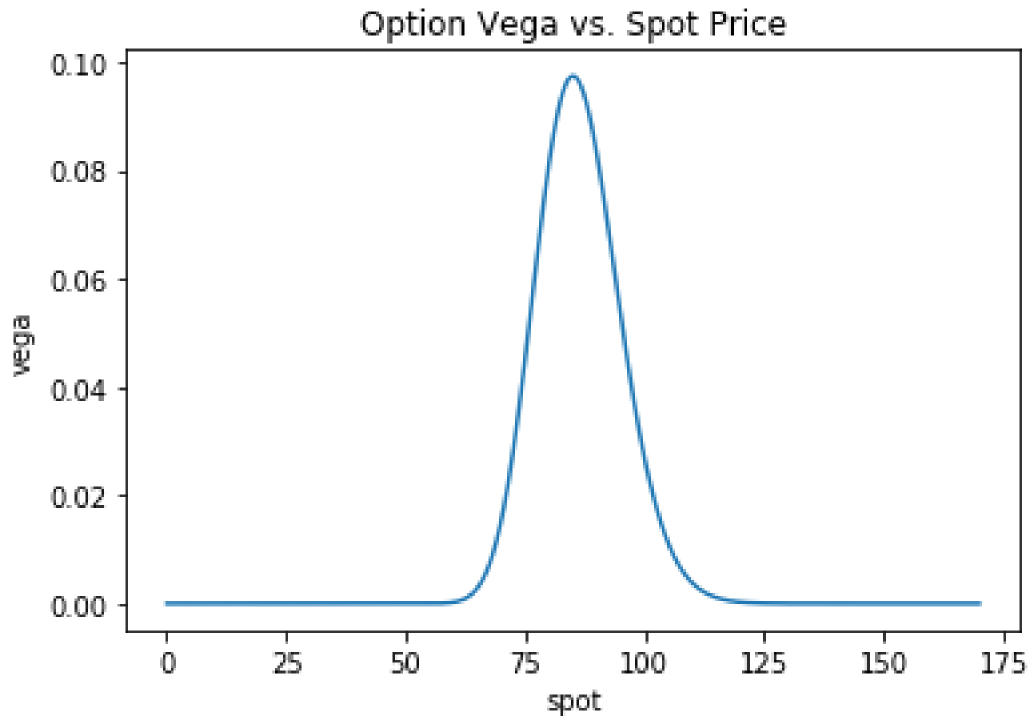

```
In [19]: call75 = op.option(strike=75, expiry=1/12, type='call')
         call75.plot_vega(time=0, vol=35, rate=6)
```



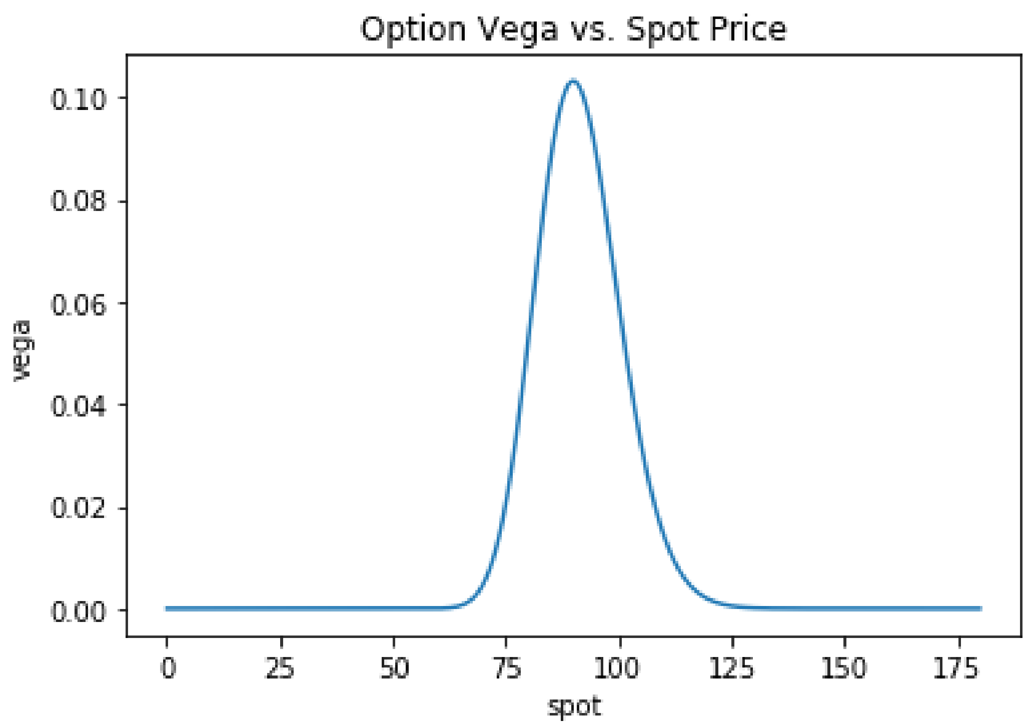

```
In [20]: call80 = op.option(strike=80, expiry=1/12, type='call')
         call80.plot_vega(time=0, vol=35, rate=6)
```



```
In [21]: call85 = op.option(strike=85, expiry=1/12, type='call')
         call85.plot_vega(time=0, vol=35, rate=6)
```




```
In [22]: call90 = op.option(strike=90, expiry=1/12, type='call')
         call90.plot_vega(time=0, vol=35, rate=6)
```



We will use the 70 strike option as our in the money option and the 90 strike option as our out of the money option. Of course, strike 80 is the at the money option.
We will calculate the delta of each option in order to build a delta neutral portfolio for each one:

```
In [23]: delta70 = call70.delta(spot=80, time=0, vol=35, rate=6)
         print(delta70)
```

```
0.9224321889326137
```

```
In [24]: delta80 = call80.delta(spot=80, time=0, vol=35, rate=6)
         print(delta80)
```

```
0.5398299468746814
```

```
In [25]: delta90 = call90.delta(spot=80, time=0, vol=35, rate=6)
         print(delta90)
```

```
0.14326959784392523
```

For a position size of 500 shares a piece, we will short (ie sell) 500 of each of these options, and for each strike we combine the option position with a long position in the stock so that the combined position is delta neutral. These long positions should be

```
In [26]: delta70 * 500
```

```
Out[26]: 461.21609446630686
```

or 461 shares with the 70 strike call

```
In [27]: delta80 * 500
```

```
Out[27]: 269.91497343734073
```

or 270 shares with the 80 strike call

```
In [28]: delta90 * 500
```

```
Out[28]: 71.63479892196261
```

or 72 shares with the 90 strike call.
In the first scenario we are considering, we assume only the implied volatility changes after the company releases its report, with no change in the stock price. So under this scenario, the P&L of each position will be determined entirely by the change in option value. These P&Ls as follows:
70 strike call:

```
In [29]: PL1 = -500 * (call70.price(80, 1/365, 20, 6) - call70.price(80, 0, 35, 6))
         print(PL1)
```

```
147.75393227146694
```

80 strike call:

```
In [30]: PL2 = -500 * (call80.price(80, 1/365, 20, 6) - call80.price(80, 0, 35, 6))
         print(PL2)
```

```
705.8252642427157
```

90 strike call:

```
In [31]: PL3 = -500 * (call90.price(80, 1/365, 20, 6) - call90.price(80, 0, 35, 6))
         print(PL3)
```

```
262.8699282329869
```

So the shrt at the money option with the $80 strike profits the most from the drop in volatility, as we would expect from the vega plots.

(iii)

We will now focus on the 80 strike call and consider a portfolio made up of a short position on 500 of these calls plus a long position of 270 shares of the underlying stock. We calculate the P&L under the 8 proposed scenarios:

```
In [32]: PL1 = 270 * (70 - 80) - 500 * (call80.price(70, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL1)
```

```
-999.2311757061689
```

```
In [33]: PL2 = 270 * (75 - 80) - 500 * (call80.price(75, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL2)
```

```
193.82965040496856
```

```
In [34]: PL3 = 270 * (77 - 80) - 500 * (call80.price(77, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL3)
```

```
517.1161242089349
```

```
In [35]: PL4 = 270 * (79 - 80) - 500 * (call80.price(79, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL4)
```

```
686.4258196791435
```

```
In [36]: PL5 = 270 * (81 - 80) - 500 * (call80.price(81, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL5)
```

```
681.7609229372299
```

```
In [37]: PL6 = 270 * (83 - 80) - 500 * (call80.price(83, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL6)
```

```
515.0382005883584
```

```
In [41]: PL7 = 270 * (85 - 80) - 500 * (call80.price(85, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL7)
```

```
222.21419823974952
```

```
In [39]: PL8 = 270 * (90 - 80) - 500 * (call80.price(90, 1/365, 20, 6) -\
                                        call80.price(80, 0, 35, 6))
         print(PL8)
```

```
-796.7999115934026
```

It turns out this position profits the most if the stock price does not change at all. As the stock price moves, the profits of this position are depleted, and once the stock price moves as much as $10 in either direction, the position loses money. This is exactly the opposite of what we have found with other delta neutral positions we have considered. So what's the explanation?

The difference is that in this position we are shorting the options, whereas in all the other delta neutral positions we have considered so far we have always been long the options. The U shaped profit curve we see from a delta neutral position that's long the options is upside down in this portfolio, so that a stock price that changes a lot, ie a volatile stock, hurts this position. That is related to why this position is considered shorting volatility. This is quantitatively demonstrated most distinctly by the fact that this position has a negative gamma. Recalling that this position is short 500 call options, the gamma of the position is:

```
In [40]: -500 * call80.gamma(80, 0, 35, 6)
```

```
Out[40]: -24.555057679333583
```

This indicates that the price function of this position is curved upwards and that large moves in the price will reduce the profitability of the position.