FE 620 Project Report

Aidana Bekboeva, Rachel Besecker, Meher Kohli, Eden Luvishis

2023-04-23

1 Exposition

An American option is a financial derivative that grants the holder the right, but not the responsibility, to purchase (call option) or sell (put option) a specific asset, such as a stock or commodity, at any time up to and including the option's expiration date. In contrast to a European option, which may only be executed on the expiration date, the option can be exercised at any time before it expires.

In this project we chose American option on GOOG for several reasons: firstly, this financial derivative does not pay dividends, which makes it easier to price using Binomial tree model; secondly, the underlying asset has large trading volumes (381,877 avg 30 Day option volume) and is a good representation of the market. We believe the high volume will make the option pricing more efficient in the market and therefore make our models more accurate.

Alphabet Inc., Google's parent firm, is denoted by the ticker sign GOOG. The holder of an American option on GOOG would have the opportunity to buy or sell GOOG shares at any time before the expiration date.

If the market price of GOOG shares is higher than the strike price, the payoff of an American call option on GOOG would be the difference between the market price and the strike price (the price at which the option can be exercised), or zero if the market price is lower than the strike price. For example, if a call option's strike price is 1,000 and the current market price of GOOG shares is 1,100, the payment from exercising the option is $100.

The economic rationale for purchasing an American call option on GOOG is to profit from a rise in the price of GOOG shares. If the market price of GOOG shares rises over the option's strike price, the holder can exercise the option and purchase the shares at a lower price, then sell them at the higher market price to profit. The economic rationale for purchasing an American put option on GOOG is to profit from a drop in the price of GOOG shares. If the market price of GOOG shares falls below the option's strike price, the holder can exercise the option and sell the shares at a higher price before buying them back at the lower market price to profit.

Options on GOOG in the United States can be traded on options exchanges such as the Chicago Board Options Exchange (CBOE) and the International Securities Exchange (ISE) (ISE). Broker-dealers can also trade them over-the-counter (OTC).

2 Pricing Algorithm

For the pricing algorithm, we have chosen to use a *binomial tree* model to value both the put and call options. Because we will be pricing an *American option*, we cannot use a Monte-Carlo simulation of Black Scholes since the option can be exercised early. However, because our stock, GOOG, does not pay dividends, the call price of the binominal tree should converge to the Black Scholes price.

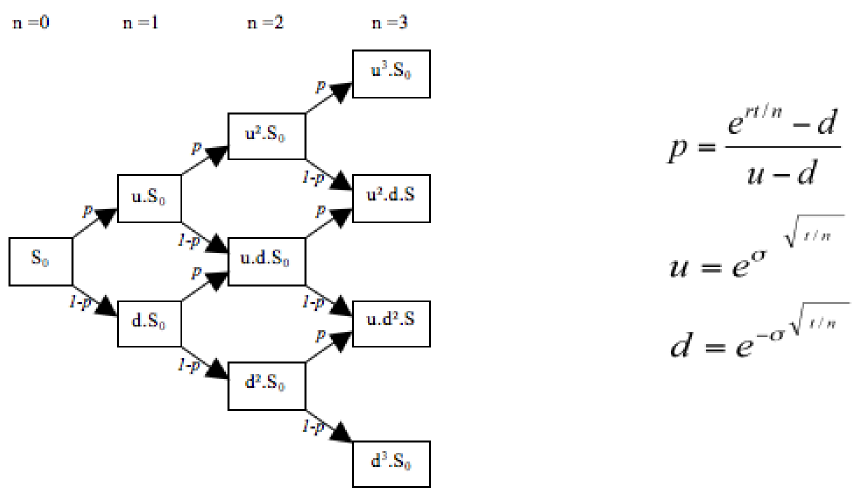


Figure 1: Binomial Tree Depiction[3]

We will start with a theoretical simulation of data. We will use the *derivmkts* package's `binomopt` function for this.

We will set

$$S = 100,\ K = 101,\ \sigma = 0.25,\ r = 0.03,\ \tau = 1,\ d = 0$$

In order to test the efficacy of the binomial tree model that we are using, we will compare the call option price that the tree converges to to the Black Scholes options pricing model.

For the binomial model, we will use risk neutral pricing with 'up' and 'down' factors as follows:

$$u = e^{\sigma\sqrt{\Delta t}}$$

$$d = e^{-\sigma\sqrt{\Delta t}}$$

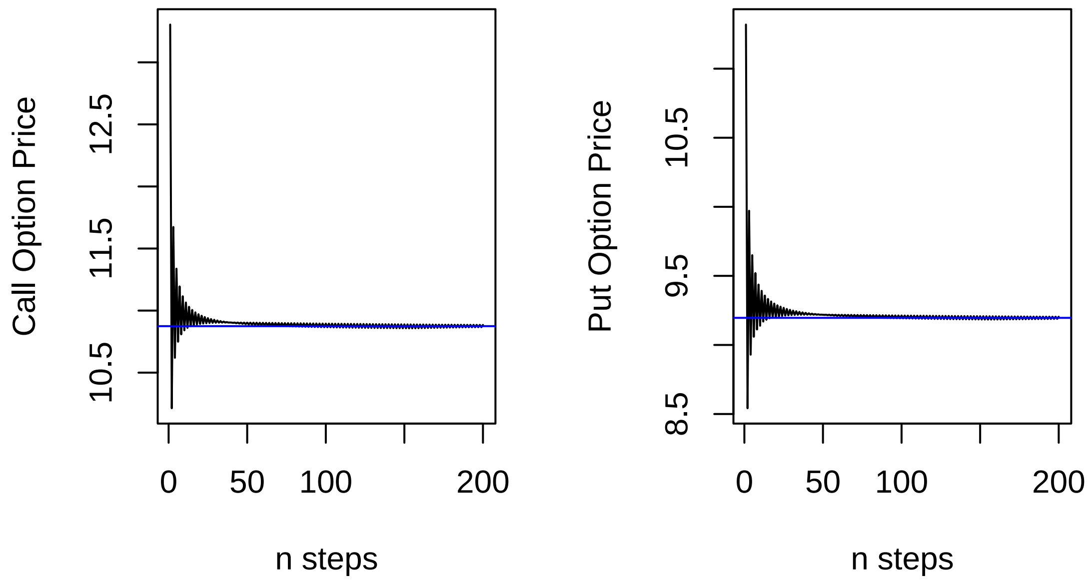


Figure 2: We can see the price converging to a stable value as n increases. The left figure represents call options and the right one represents put options.

Black Scholes Comparison:

We will now use the Black Scholes Merton model to price the theoretically simulated option. In order to do so, we wll again use the *derivmkts* package.

Using Black Scholes, we find that the European option with the same simulated parameters would cost 10.87.

$$BSM(Call) \approx Binomial\ Model(Call) = 10.87$$

Now that we are confident in our pricing model, we can test the sensitivity of other parameters. We will use $n = 200$ for further testing.

Sensitivity Testing

We will begin by testing sensitivity to K, the strike price, for a range of values: $[60, 130]$

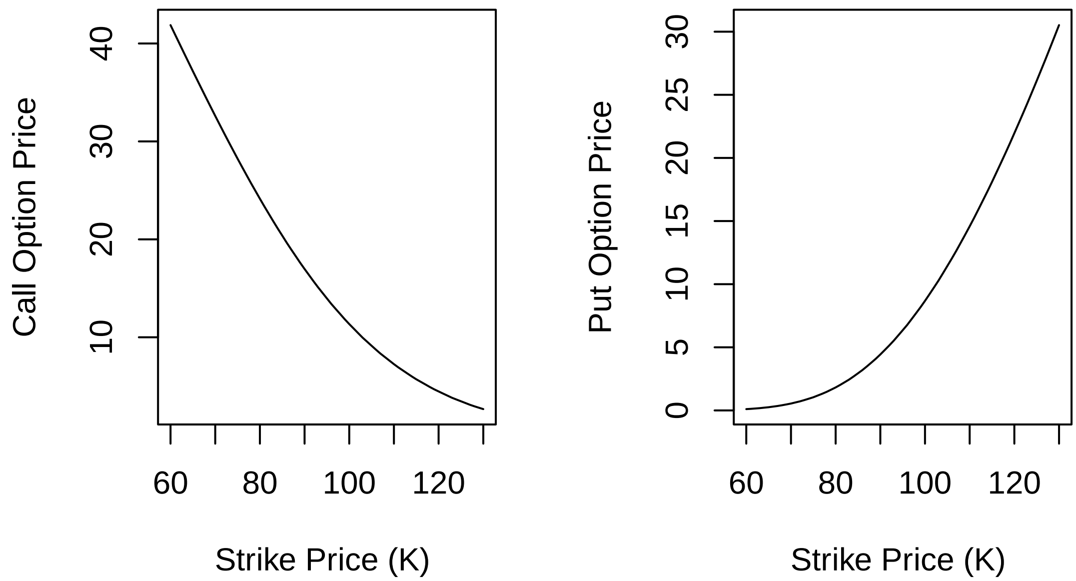


Figure 3: As expected, the strike K has an inverse relationship with the call price and a positive relationshp with the put price.

We will now test variations in the price of the stock S_0. This is representative of Delta (Δ).

Now we will examine how changes in volatility affect option price.

Finally, we will examine the correlation between time to expiry and option price. We expect that as time to expiry goes down, so will the options' value.

3 Data Analysis

Now, we will download actual data from Yahoo Finance for historical Google stock prices and option prices. We will use two expiry dates for this project: April 28th, 2023 and May 5th, 2023. We downloaded all data on April 24th, 2023 when there were $n = 5$ days remaining until expiry of the 4/28 options and $n = 10$ business days so the time to expiry can be summarized as:

28-April-2023: $T = \frac{5}{252}$ Years

5-May-2023: $T = \frac{10}{252}$ Years

We should recall that American options have bounds for their prices that follow from the put-call parity:

$$S_0 - K < C(K) - P(K) < S_0 - Ke^{-rT}$$

Since we are dealing with such short maturities, the factor $e^{-rT} \approx 1$. We will now visualize this for strike

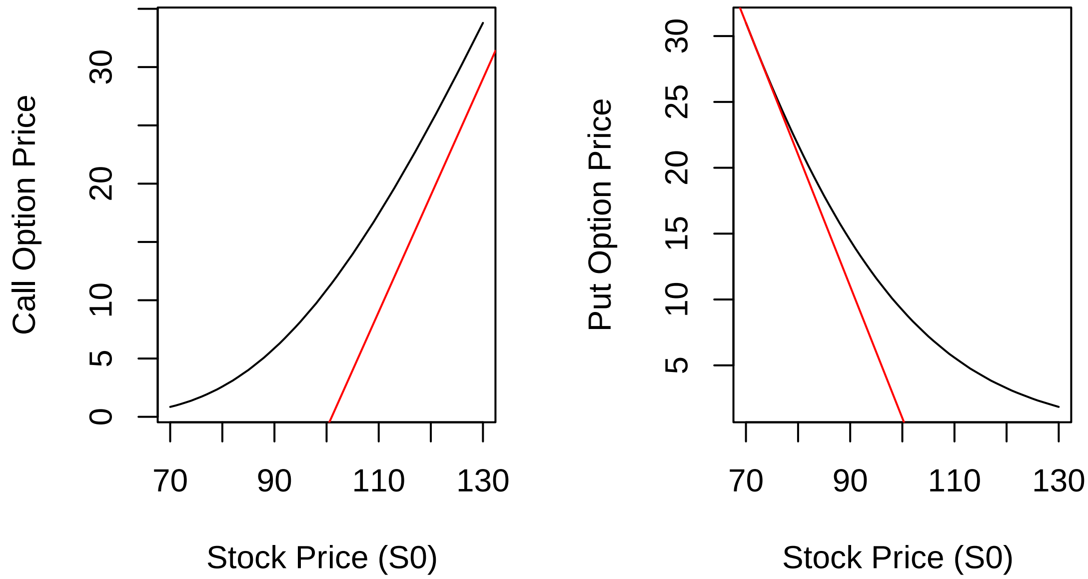


Figure 4: As expected, the stock price has a positive relationship with the call price and an inverse relationshp with the put price.

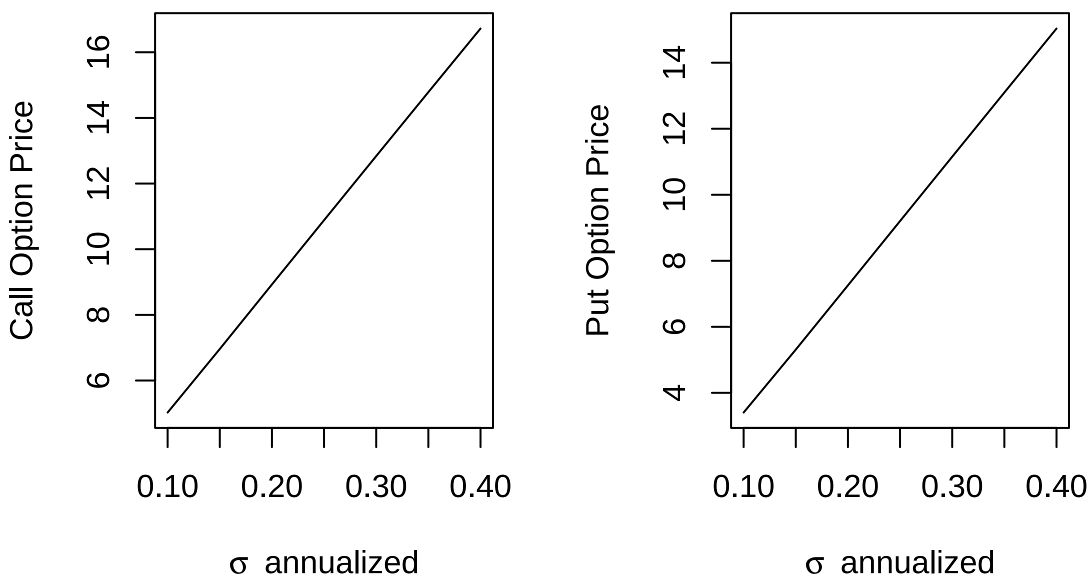


Figure 5: For both call and put options, an increase in volatility will cause an increase in option prices as there is a greater probability that they will expire in the money.

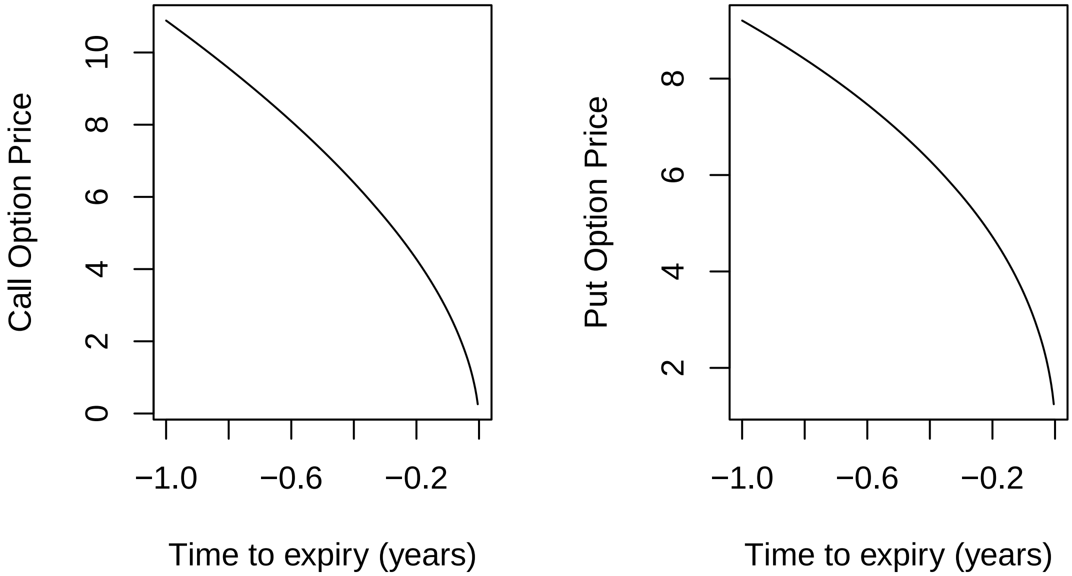


Figure 6: For both call and put options, as time to expiry goes down, so does option value.

	Ticker	Bid	Ask	Last	IVM	Volm	Strike		Ticker	Bid	Ask	Last	IVM	Volm
	28-Apr-23 (7d); CSize 100						5		28-Apr-23 (7d); CSize 100					
1)	GOOG 4/28/23 C104	3.90	4.00	3.85	48.80	116	104.00	78)	GOOG 4/28/23 P104	1.96	1.97	1.96	49.04	926
2)	GOOG 4/28/23 C105	3.35	3.40	3.39	48.76	555	105.00	79)	GOOG 4/28/23 P105	2.38	2.40	2.40	49.06	1395
3)	GOOG 4/28/23 C106	2.83	2.85	2.87	48.47	2542	106.00	80)	GOOG 4/28/23 P106	2.86	2.88	2.86	48.95	1778
4)	GOOG 4/28/23 C107	2.35	2.37	2.39	48.14	2970	107.00	101)	GOOG 4/28/23 P107	3.35	3.45	3.37	48.73	880
5)	GOOG 4/28/23 C108	1.93	1.94	1.95	47.84	1680	108.00	102)	GOOG 4/28/23 P108	3.95	4.00	4.01	48.35	321
	05-May-23 (14d); CSize 100						5		05-May-23 (14d); CSize 100					
6)	GOOG 5/5/23 C104	4.40	4.60	4.35	40.83	24	104.00	103)	GOOG 5/5/23 P104	2.42	2.44	2.46	41.06	288
7)	GOOG 5/5/23 C105	3.85	4.00	3.85	40.64	479	105.00	104)	GOOG 5/5/23 P105	2.84	2.86	2.89	40.87	1336
8)	GOOG 5/5/23 C106	3.35	3.40	3.38	40.18	306	106.00	105)	GOOG 5/5/23 P106	3.30	3.35	3.30	40.70	229
9)	GOOG 5/5/23 C107	2.88	2.91	2.78	40.00	207	107.00	106)	GOOG 5/5/23 P107	3.80	3.85	3.82	40.25	166
10)	GOOG 5/5/23 C108	2.43	2.46	2.44	39.61	569	108.00	107)	GOOG 5/5/23 P108	4.30	4.85	4.14	42.23	5
	12-May-23 (21d); CSize 100						5		12-May-23 (21d); CSize 100					
11)	GOOG 5/12/23 C104	4.55	5.45	5.26	37.92	3	104.00	108)	GOOG 5/12/23 P104	2.83	2.86	2.96	38.22	282
12)	GOOG 5/12/23 C105	3.90	4.50	4.31	35.41	18	105.00	109)	GOOG 5/12/23 P105	3.25	3.30	3.36	38.05	34
13)	GOOG 5/12/23 C106	3.85	3.95	3.80	37.52	126	106.00	110)	GOOG 5/12/23 P106	3.70	3.75	3.80	37.67	105
14)	GOOG 5/12/23 C107	3.35	3.45	3.25	37.20	151	107.00	111)	GOOG 5/12/23 P107	4.20	4.25	4.15	37.37	6
15)	GOOG 5/12/23 C108	2.91	2.95	2.84	36.77	129	108.00	112)	GOOG 5/12/23 P108	4.35	5.15	4.65	36.86	45
	19-May-23 (28d); CSize 100						7		19-May-23 (28d); CSize 100					
16)	GOOG 5/19/23 C90	16.25	16.70	16.68	37.25	3	90.00	113)	GOOG 5/19/23 P90	.37	.38	.38	41.65	1099

Figure 7: GOOG Option Prices as seen in Bloomberg[4]

prices that are close to S_0. Based on the put-call graph, we will use the strike price closest to S_0 in our modeling. We will use $K = 106$.

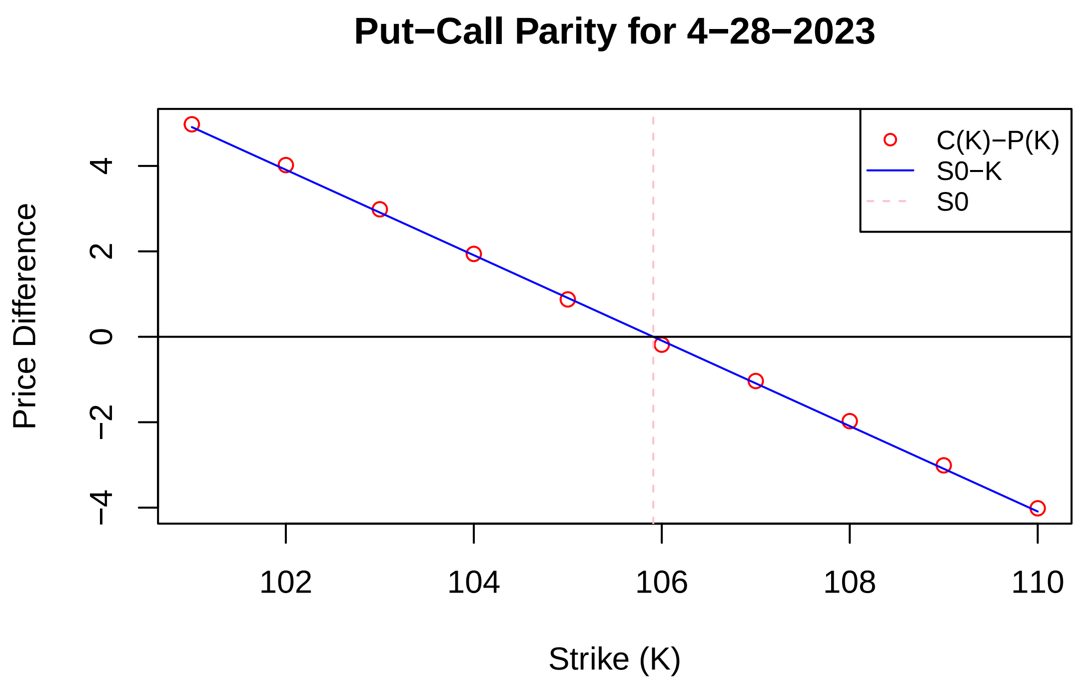
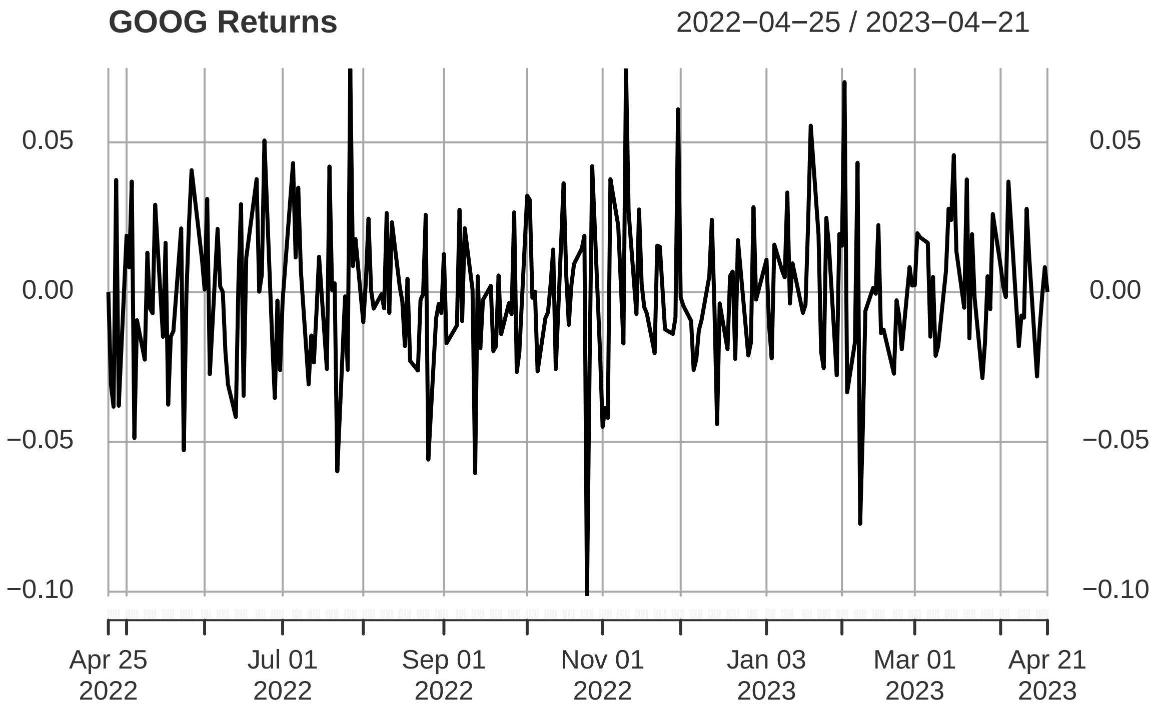


Figure 8: As expected, the line representing $S_0 - K$ converges closely with $C(K) - P(K)$

Analyzing GOOG returns and volatility

We will first estimate returns:



We now want to make sure that Google's returns follow a normal distribution in order to apply our modeling techniques including Black Scholes for the American call on a non-dividend paying asset.

We will use a QQ plot in Figure 7 in order to do so. We expect to see a straight line if the returns do follow a relatively normal distribution.

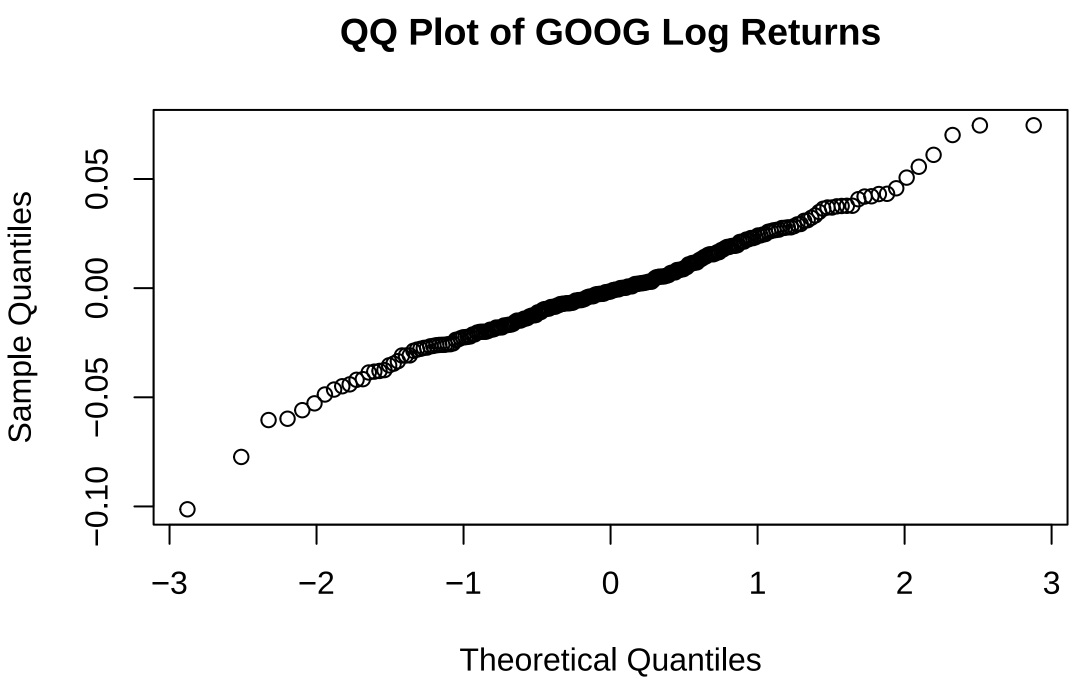


Figure 9: The relatively straight line confirms our hypothesis, that the returns are in fact normal

Estimating Historical Volatility

Now we can estimate volatility using the following property:

$$\hat{\sigma} = \sqrt{\frac{1}{\tau} Var(r_i)} = 252 * stdev(r_i)$$

The error of this estimate is:

$$\frac{\delta \hat{\sigma}}{\hat{\sigma}} = \frac{1}{\sqrt{2n_{days}}}$$

Now we can calculate the historical volatility for four lookback periods: 1, 3, 6, and 12 months.

	Volatility	Error
1 month	0.2959401	0.0581441
3 months	0.3798609	0.0351317
6 months	0.4077233	0.0249411
1 year	0.3943528	0.0175659

With a longer lookback window, we can see a lower error estimate, however we also introduce systemic error. Therefore, we will use the 3 month estimate going forward.

Risk Free Rate

Finally, we need to estimate r the risk-free rate. We can once again pull this data from Yahoo Finance by looking at the Treasury Bill rates with symbol $\hat{}IRX$.

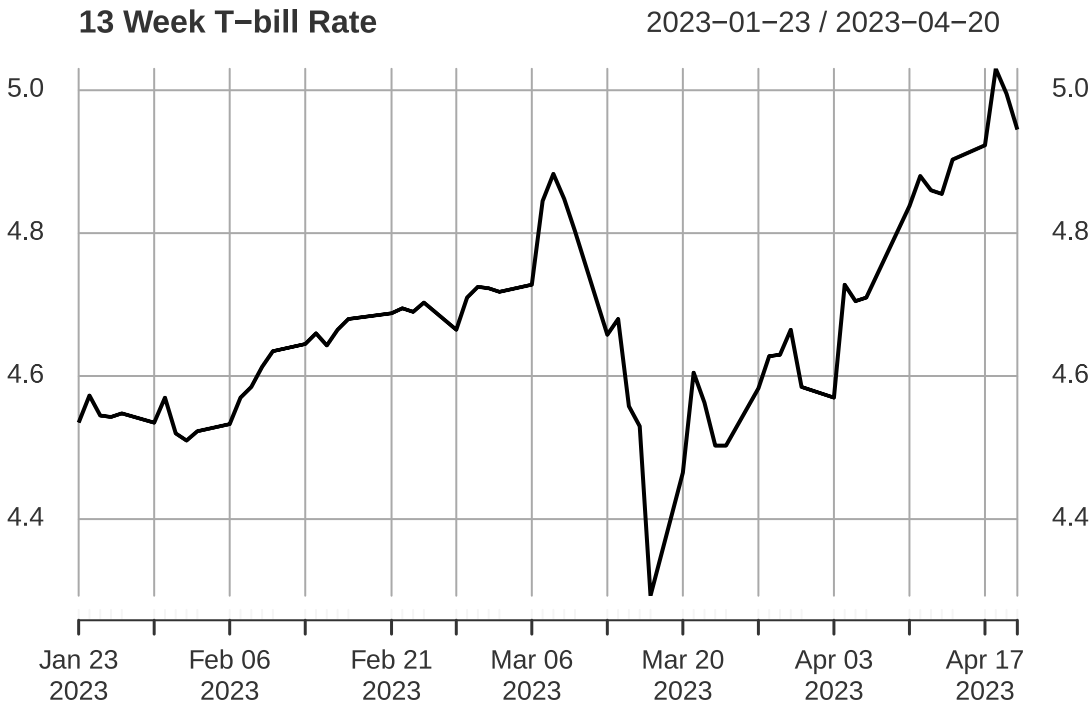


Figure 10: Last 3 months of 13 week T-bill rates

	IRX.Close
2023-04-14	4.903
2023-04-16	NA
2023-04-17	4.923
2023-04-18	5.030
2023-04-19	4.995
2023-04-20	4.945

We will use the most recent value $r = 0.04945$ as our risk-free rate estimate.

Now that we have our $S_0, K, r, \hat{\sigma}$ we are ready to begin modeling.

4 Modeling with Realistic Financial Data

We will need to perform the same sensitivity testing as we did earlier for real option data to find our ideal n for both options.

We will start with the 5-day to expiry option where:

$$S_0 = 105.91$$

$$K = 106$$

$$r = 0.0495$$

$$T = \frac{5}{252}$$

$$\hat{\sigma} = 0.3798$$

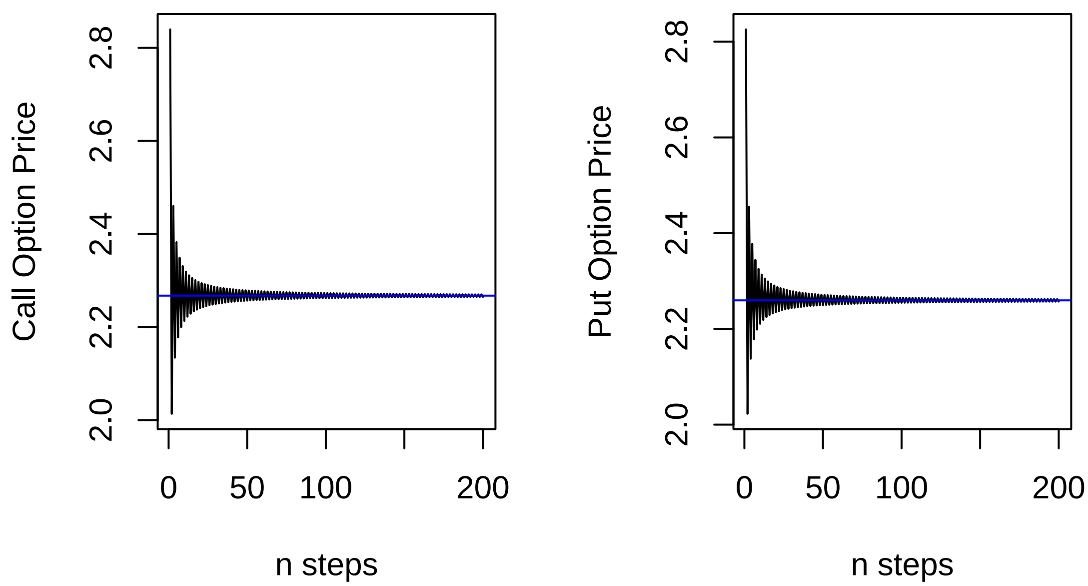


Figure 11: Option Pricing with $T = 5/252$. We are once again confident that by $n = 200$ steps the price converges.

Finally we will compare to the Black Scholes Pricing for the Call Option in particular. Once again, we expect the prices of the American option and the BSM model to converge because GOOG does not pay dividends

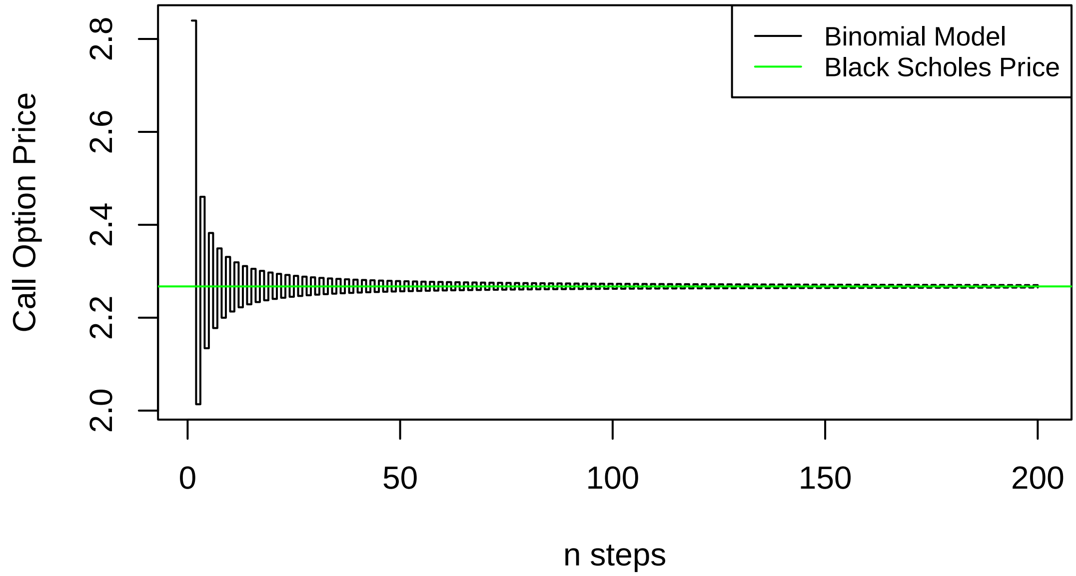


Figure 12: We can conclude that the binomial model does in fact converge to the Black Scholes price for large n.

Option Pricing Comparison

The following tables shows results from the Binomial Option Pricing Model for both April 28 and May 5th expiries. The call options we can compare to the BSM (Black Scholes Merton) model as we can expect the prices to converge since our asset does not pay dividends. For all of the Binomial models we use $n = 200$ steps as we are certain that the price converges at this point.

We will start with options expiring on 4-28-23.

Table 3: Call Option

K	Binomial	BSM	Actual_Bid	Actual_Ask
100	6.39	6.39	5.50	6.90
101	5.56	5.56	5.90	6.10
102	4.77	4.78	5.20	5.35
103	4.05	4.05	4.50	4.70
104	3.39	3.39	3.90	4.00
105	2.79	2.79	3.30	3.45
106	2.26	2.27	2.79	2.90
107	1.81	1.81	2.31	2.42
108	1.43	1.43	1.95	2.00
109	1.10	1.10	1.52	1.61
110	0.84	0.84	1.21	1.26

Table 4: Put Option

Binomial	Actual_Bid	Actual_Ask
0.39	0.75	0.84
0.55	0.97	1.08
0.77	1.21	1.30
1.04	1.53	1.70
1.38	1.92	2.10
1.79	2.29	2.71
2.26	2.76	3.30
2.81	3.30	3.50
3.42	3.85	4.05
4.10	4.50	4.65
4.84	5.15	5.35

Now we will repeat for 5-5-23 expiry:

Table 5: Call Option

K	Binomial	BSM	Actual_Bid	Actual_Ask
100	7.07	7.07	7.15	6.90
101	6.32	6.32	6.40	6.10
102	5.61	5.61	5.70	5.35
103	4.95	4.95	5.05	4.70
104	4.33	4.33	4.45	4.00
105	3.76	3.77	3.85	3.45
106	3.25	3.25	3.30	2.90
107	2.79	2.79	2.54	2.42
108	2.38	2.37	2.23	2.00
109	2.01	2.00	1.85	1.61
110	1.68	1.68	1.55	1.26

Table 6: Put Option

Binomial	Actual_Bid	Actual_Ask
0.97	1.00	1.37
1.21	1.27	1.61
1.51	1.17	1.93
1.84	2.02	2.56
2.23	2.16	2.93
2.66	2.53	3.35
3.15	3.25	3.35
3.69	3.75	5.10
4.27	4.30	5.65
4.91	4.40	6.00
5.58	5.55	6.35

As we can see in the tables, the binomial model consistently prices options with strikes close to S_0 at lower values than the market does. This is a phenomenon that is directly tied to implied volatility.

When pricing with a model, we use just one volatility value $\hat{\sigma}$ which corresponds to historical volatility. However, the actual implied volatility, which is represented as the σ value used within a Black Scholes Model to accurately price the option, is consistently higher than the historical volatility we used. We will demonstrate this in Figure 11 below.

The implied volatility is higher because of a phenomenon called *risk aversion* in which buyers of an option pay a premium to sellers in order to be able to hedge costs. Therefore, sellers will charge a premium to take on more risk, which leads to a higher σ value needed in models to achieve the market price.

Interestingly, we do not observe a clear volatility smile form for either option[1]. However, there is a clearer smile for the May 5th option. This is likely due to the fact that the option data for certain strikes does not have volume and therefore, the price and implied volatility is not reflective of the actual price of each option.

Volatility effect:

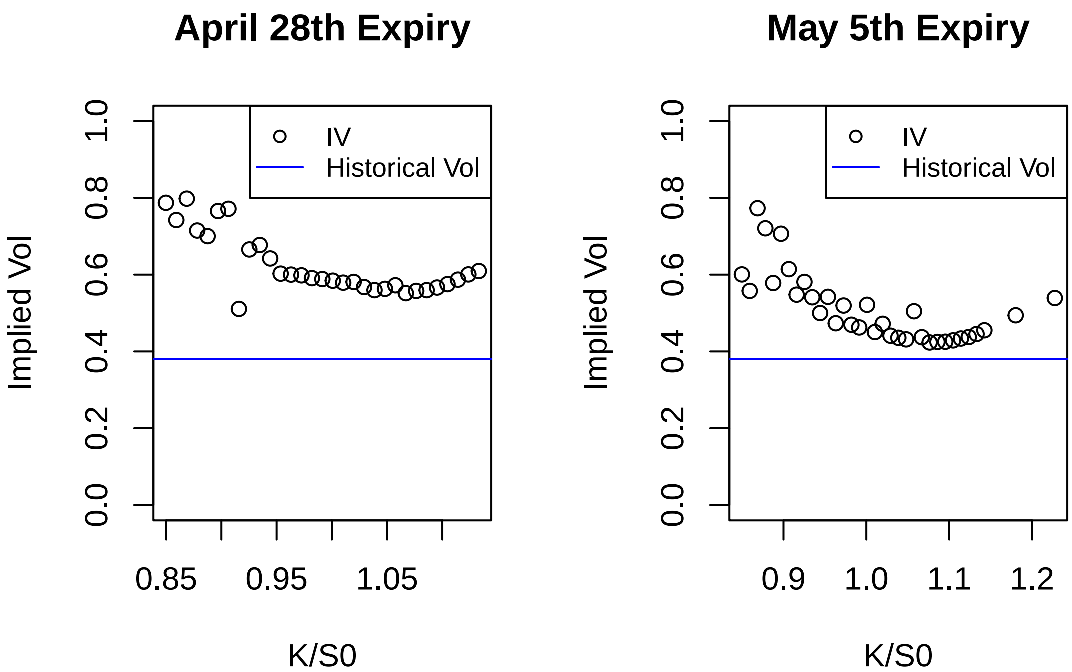


Figure 13: We can clearly see that in the case of both options, the implied volatility is consistently higher than the historical volatility. This is consistent with compensation for risk that option sellers take on.

5 Hedging Analysis

Finally, we are now ready to perform a hedging analysis using our Binomial Pricing model. In order to do so, we will begin by looking at the Greeks of the derivatives produced by the model. Luckily, the *derivmkts* package makes it very easy to do so. In this hedging analysis, we will do a 5 day analysis for the option expiring on April 28th, 2023.

We will begin by observing the convergence of the delta, gamma, and theta values as the number of time steps approaches our desired $n = 200$.

As a quick review:

$$\Delta = \frac{\delta c}{\delta S}$$

$$\Gamma = \frac{\delta^2 c}{\delta S^2}$$

$$\Theta = \frac{\delta c}{\delta T}$$

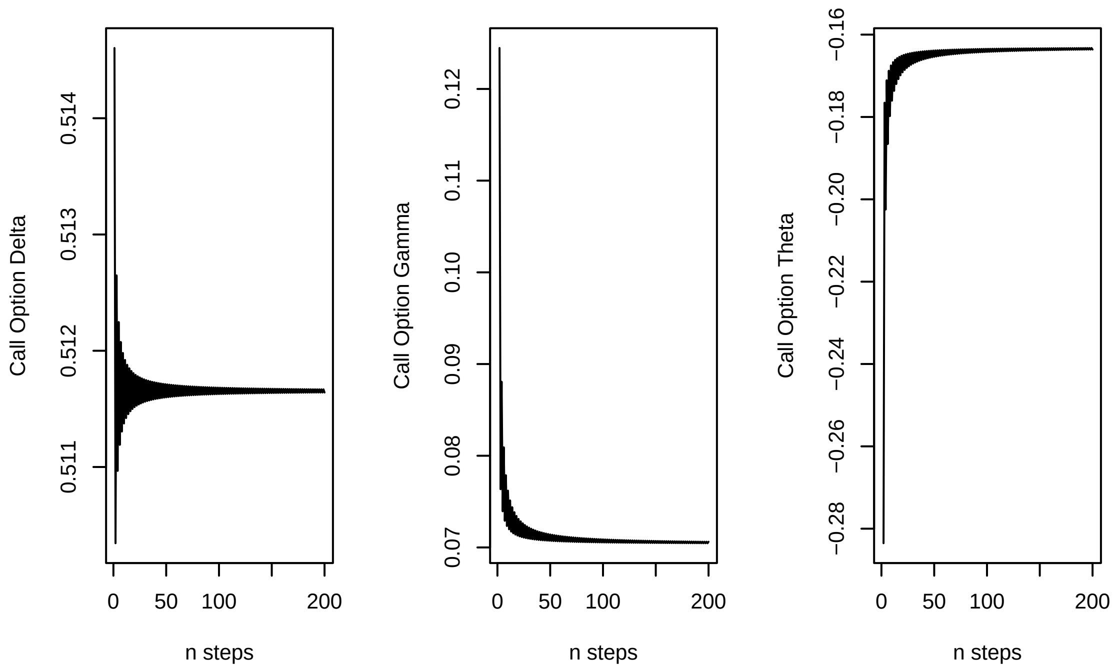


Figure 14: Greeks of the call option on GOOG as of April 21st, 2023 with 5 days remaining until expiry.

Now let's observe delta and gamma for different strike prices:

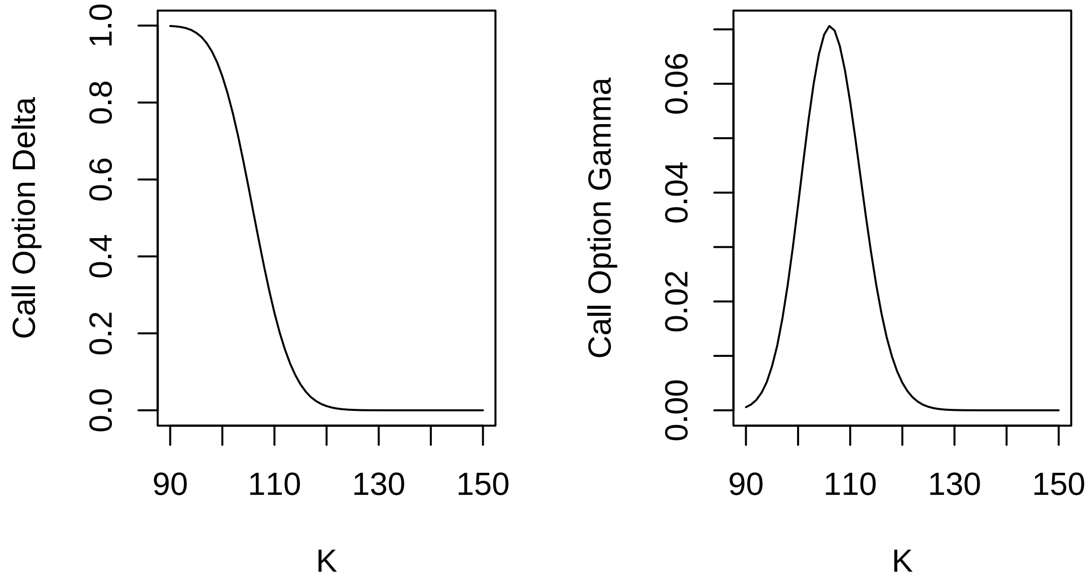


Figure 15: Delta and Gamma vs different Strike prices. The graphs behave as expected with Delta peaking for deep ITM calls and leveling out to 0 for deep OTM calls.

We will repeat the same analysis for put options:

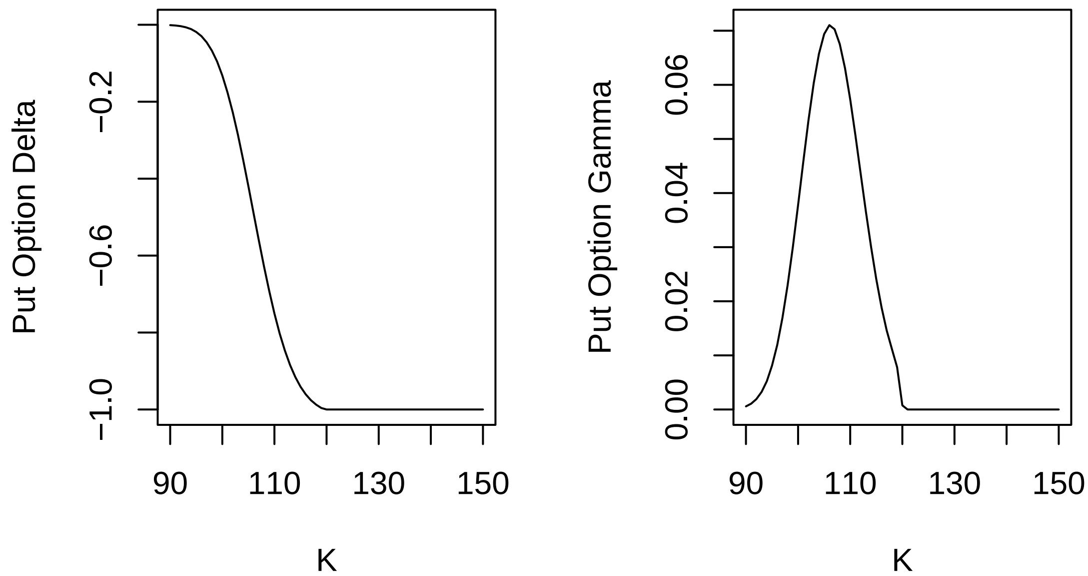


Figure 16: Delta and Gamma vs different Strike prices. The graphs behave as expected with Delta peaking at -1 for deep ITM puts and leveling out to 0 for deep OTM puts.

We can see that all three Greeks converge, meaning the model is reliable to be used at $n = 200$ steps. We can now begin hedging.

Let's examine the stock prices throughout the week that we will be hedging:

	Date	Stock Price
250	2023-04-21	105.91
251	2023-04-24	106.78
252	2023-04-25	104.61
253	2023-04-26	104.45
254	2023-04-27	108.37
255	2023-04-28	108.22

We see that the call option with $K = 106$ will end in the money. Let's examine hedging this call option. We know that call options will have positive delta, so we will need to remove this risk by *shorting* the underlying asset, essentially zeroing out the delta effects. We will end up with the following portfolio[2]:

$$\Pi(t) - \Pi(t-1) = C(t) - C(t-1) - \Delta(t-1)(S(t) - S(t-1))$$

Let's observe the close price data for the last 5 days of trading for the $K = 106$ GOOG Call with expiry 4/28/23:

\begin{center}

\end{center}

GPO GOOG 4 C106 Equity BNEW What's New

GOOG 4 C106 Equity GPO Related Functions Menu Message

GOOG US 04/28/23 C106 $ C 2.16 -0.34 B2.02/2.91B 60x61

On 28 Apr d OpInt 4,902 Vol 3,133 O 1.90I H 2.18T L .40Z Prev 2.50

GOOG US 04/28/23 94) Suggested Charts 96) Actions 97) Edit Bar Chart

05/01/2022 - 05/02/2023 Last Px Local CCY Mov Avgs Key Events

1D 3D 1M 6M YTD 1Y 5Y Max Daily Chart + Related Data Add Data Edit Chart

GOOG US 04/28/23 C106 Equity

Date	Open	High	Low	Close	Open Int	SMAVG (15)	Volume	SMAVG
Tu 05/02/2023								
Fr 04/28/2023	1.90	2.18	.40	2.16			3133	
Th 04/27/2023	.75	3.25	.41	2.50	4902	2317	5697	
We 04/26/2023	1.32	2.00	.45	.57	6334	2020	14890	
Tu 04/25/2023	3.00	3.60	2.15	2.20	6141	1625	6663	
Mo 04/24/2023	2.88	3.50	2.57	3.10	4846	1238	5058	
Fr 04/21/2023	3.25	3.50	2.60	2.87	3068	935	2542	
Th 04/20/2023	2.56	3.70	2.56	3.18	2321	746	1796	
We 04/19/2023	2.50	3.05	2.28	2.73	1571	605	801	
Tu 04/18/2023	3.80	4.10	2.85	3.04	1265	514	486	
Mo 04/17/2023	3.55	4.00	3.35	3.75	1015	443	972	
Fr 04/14/2023	4.91	5.80	4.86	5.77	607	387	53	
Th 04/13/2023	3.87	5.01	3.87	4.95	581	356	190	
We 04/12/2023	4.74	4.75	3.40	3.45	605	326	168	
Tu 04/11/2023	4.55	4.55	3.80	3.80	563	294	183	
Mo 04/10/2023	5.20	5.20	4.04	4.55	473	264	141	
Th 04/06/2023	3.95	6.60	3.95	6.00	458	237	210	
We 04/05/2023	4.39	4.39	3.50	3.80	451	211	143	

Suggested Functions G Manage your charts in a central library OVME Price equity & equity index options

Figure 17: Call Option Data

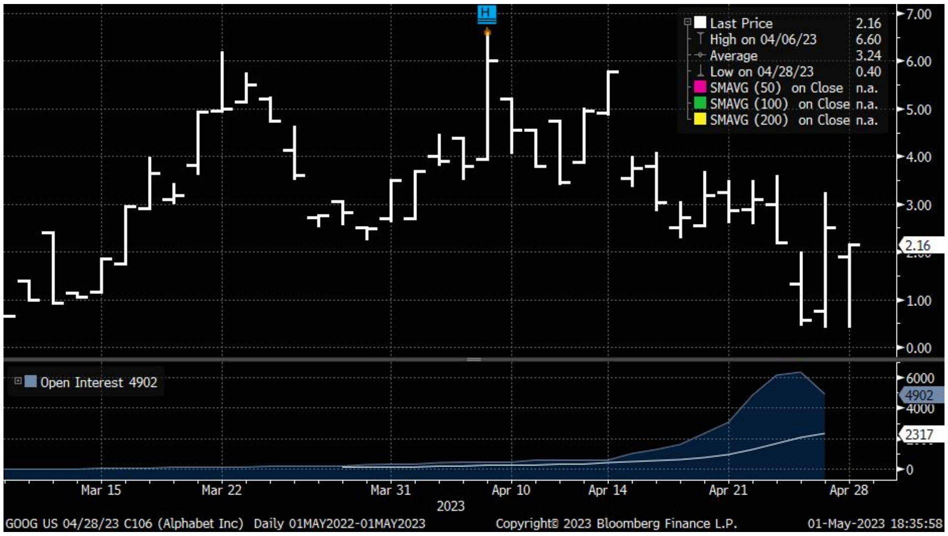


Figure 18: Call Option Historical Graph

We are now ready to construct this portfolio:

Day	S_0	Call Price	Delta	$C(t) - C(t-1)$	$\Pi(t) - \Pi(t-1)$
2023-04-21	105.91	2.87	0.5116411	NA	NA
2023-04-24	106.78	3.10	0.5767447	0.23	-0.2151253
2023-04-25	104.61	2.20	0.3882919	-0.90	0.3515349
2023-04-26	104.45	0.57	0.3417476	-1.63	-1.5678719
2023-04-27	108.37	2.50	0.8274870	1.93	0.5903475
2023-04-28	108.22	2.16	NaN	-0.34	-0.2158757

As we can observe in the table, the addition of delta-shares of stock significantly reduces portfolio value variations as the price of the call option fluctuates. This is exactly the idea behind Delta Hedging. We can use options and stock shares in tandem to counterbalance price swings in each asset as they are directly correlated.

We should note, however, that the portfolio still has small price variation that is likely linked to the presence of gamma and theta effects that are not eliminated through delta hedging alone.

6 Next Steps

Having completed the project, we discovered multiple ideas for further development. Here are some of the potential directions we could explore:

- Incorporating dividends.
  - While the stock used in this project, GOOG, does not pay dividends, many other stocks do. Incorporating dividends into the pricing model could provide a more accurate estimate of option values
- Expanding sensitivity testing.
  - This project tested the sensitivity of option values to changes in strike price, stock price, volatility, and time to expiry, there are other factors that could be explored. For example, testing the impact of interest rates or changes in market conditions on option values could provide valuable insights.
- Conducting real-world testing.
  - Real-world testing using historical market data could provide more accurate estimates of option values. This could also help to identify any potential weaknesses or limitations in the pricing model.

- Considering additional applications.
  - While this project focused on pricing options, the techniques used could be applied to other areas of finance as well. For example, the binomial tree model could be used to value bonds or other fixed-income securities, or to estimate the value of assets in other industries such as real estate or energy.

References

[1] https://www.theoptionsguide.com/volatility-smile.aspx

[2] John C. Hull, Options, Futures and Other Derivatives, Pearson, 10th Edition, 2018

[3] https://en.wikipedia.org/wiki/Binomial_options_pricing_model

[4] Bloomberg Finance L.P

[5] Yahoo Finance

Appendix A - Code

Packages

```
library(derivmkts)
library(knitr)
library(quantmod)
library(xtable)
```

Pricing Algorithm

Fig 1

```
S0 <- 100
K <- 101
r <- 0.03
sig <- 0.25
T_exp <- 1
d <- 0
n <- 10

prices <- c()
steps <- c()

#Call Option
for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

par(mfrow=c(1,2))
plot(steps, prices, xlab = "n steps", ylab = "Call Option Price", type = "l")
abline(h=mean(prices[150:200]), col="blue")
```

```r
#Put Option
prices <- c()
steps <- c()

for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i,
                     putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

plot(steps, prices, xlab = "n steps", ylab = "Put Option Price", type = "l")
abline(h=mean(prices[150:200]), col="blue")
```

Black Scholes Comparison

```r
#Black Scholes Simulation
bscall(s = S0, k = K, r = r, v = sig, tt = T_exp, d)
```

Sensitivity Testing

We will begin by testing sensitivity to K, the strike price, for a range of values: $[60, 130]$

Fig 2

```r
k = seq(60, 130, 1)
S0 <- 100
r <- 0.03
sig <- 0.25
T_exp <- 1
d <- 0
n <- 10

prices <- c()
```

```r
steps <- c()

for (i in k) {
  nextopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

par(mfrow=c(1,2))
plot(steps, prices, xlab = "Strike Price (K) ", ylab = "Call Option Price", type = "l")

prices <- c()
steps <- c()

for (i in k) {
  nextopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200,
                      putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

plot(steps, prices, xlab = "Strike Price (K) ", ylab = "Put Option Price", type = "l")
```

We will now test variations in the price of the stock S_0. This is representative of Delta (Δ).

Fig 3

```r
S0 <-  seq(70, 130, 1)
K <- 101
r <- 0.03
sig <- 0.25
T_exp <- 1
d <- 0
n <- 10
```

```r
prices <- c()
steps <- c()

for (i in S0) {
  nextopt <- binomopt(s = i, k = K, r = r, v = sig, tt = T_exp, d, nstep = 200)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

par(mfrow=c(1,2))
plot(steps, prices, xlab = "Stock Price (S0) ", ylab = "Call Option Price", type = "l")
abline(a=-K, b=1, col = "red")



prices <- c()
steps <- c()

for (i in S0) {
  nextopt <- binomopt(s = i, k = K, r = r, v = sig, tt = T_exp, d, nstep = 200,
                      putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

plot(steps, prices, xlab = "Stock Price (S0) ", ylab = "Put Option Price", type = "l")
abline(a=K, b=-1, col = "red")
```

Now we will examine how changes in volatility affect option price.

Fig 4

```r
library(latex2exp)
S0 <-  100
K <- 101
```

```r
r <- 0.03
sig <- seq(0.10, 0.40, 0.05)
T_exp <- 1
d <- 0
n <- 10

prices <- c()
steps <- c()

for (i in sig) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = i, tt = T_exp, d, nstep = 200)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

par(mfrow=c(1,2))
plot(steps, prices, xlab = TeX(r'($\sigma \ annualized$)'), ylab = "Call Option Price", type = "l")

prices <- c()
steps <- c()

for (i in sig) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = i, tt = T_exp, d, nstep = 200,
                      putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

plot(steps, prices, xlab = TeX(r'($\sigma \ annualized$)'), ylab = "Put Option Price", type = "l")
```

Finally, we will examine the correlation between time to expiry and option price. We expect that as time to expiry goes down, so will the options' value.

Fig 5

```r
S0 <-  100
K <- 101
r <- 0.03
sig <- 0.25
T_exp <- seq(252/252, 0, -1/252)
d <- 0
n <- 10

prices <- c()
steps <- c()

for (i in T_exp) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = i, d, nstep = 200)
  prices <- c(prices,nextopt)
  steps <- c(steps,-i)
}

par(mfrow=c(1,2))
plot(steps, prices, xlab = "Time to expiry (years)", ylab = "Call Option Price", type = "l")

prices <- c()
steps <- c()

for (i in T_exp) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = i, d, nstep = 200,
                      putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,-i)
}

plot(steps, prices, xlab = "Time to expiry (years)", ylab = "Put Option Price", type = "l")
```

Data Analysis

Fig 7

```
call = read.csv("Project Data/GOOG_call_20230428")
put = read.csv("Project Data/GOOG_put_20230428")
stock = read.csv("Project Data/GOOG_historical_prices")

S0 = stock[stock$dates == "2023-04-21","GOOG.Close"]



call_val = call[call$Strike > (S0-5) & call$Strike < (S0 + 5), c("Strike", "Bid", "Ask") ]
put_val = put[put$Strike > (S0-5) & put$Strike < (S0 + 5), c("Strike", "Bid", "Ask") ]

call_price = 0.5*(call_val$Bid + call_val$Ask)
put_price = 0.5*(put_val$Bid + put_val$Ask)

plot(call_val$Strike, call_price-put_price, col = "red", xlab = "Strike (K)",
     ylab = "Price Difference", main = "Put-Call Parity for 4-28-2023")
lines(call_val$Strike, S0- call_val$Strike, col = "blue")
abline(v=S0, col= "pink", lty = "dashed" )
abline(h=0)
legend("topright", legend=c("C(K)-P(K)", "S0-K", "S0"), pch=c(1,NA, NA), lty = c(NA,1,2),
       col=c("red", "blue", "pink"), cex=0.8)
```

Analyzing GOOG returns and volatility

We will first estimate returns:

```
library(quantmod)
getSymbols("GOOG", from = "2022-04-23", to = "2023-04-23")
```

```
plot(Cl(GOOG), main = "Google Prices")
```

```
logret <- periodReturn(GOOG$GOOG.Close, period = "daily", type = "log")
```

```
plot(logret, main = "GOOG Returns")
```

We will use a QQ plot in Figure 8 in order to do so. We expect to see a straight line if the returns do follow a relatively normal distribution.

Fig 8

```
qqnorm(logret, main = "QQ Plot of GOOG Log Returns")
```

Estimating Historical Volatility

Now we can calculate the historical volatility for four lookback periods: 1, 3, 6, and 12 months.

```
yearvol = sqrt(252.0)*sd(logret)
ndays = 252
yearhistvolerr <- yearvol/sqrt(2*ndays)

#last day = 4-21-21

threemonthvol = sqrt(252.0)*sd(logret[index(logret) >= "2023-01-21"])
threemonthhistvolerr <- yearvol/sqrt(2*length(logret[index(logret) >= "2023-01-21"]))

sixmonthvol = sqrt(252.0)*sd(logret[index(logret) >= "2022-10-21"])
sixmonthhistvolerr <- yearvol/sqrt(2*length(logret[index(logret) >= "2022-10-21"]))

onemonthvol=sqrt(252.0)*sd(logret[index(logret) >= "2023-03-21"])
onemonthhistvolerr <- yearvol/sqrt(2*length(logret[index(logret) >= "2023-03-21"]))



kable(data.frame(Volatility = c(onemonthvol, threemonthvol, sixmonthvol, yearvol), Error = c(onemonthhis
```

Risk Free Rate

```
getSymbols("^IRX", src="yahoo", from="2023-01-21", to="2023-04-21")
```

```
## Warning: ^IRX contains missing values. Some functions will not work if objects
```

```
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

Fig 9

```
rfrate <- na.omit(Cl(IRX))
nrf <- length(rfrate)


plot(rfrate, main = "13 Week T-bill Rate")
```

```
kable(as.data.frame(tail(IRX$IRX.Close)))
```

Modeling with Realistic Financial Data

Fig 10

```
#CALL
#Our parameters
S0=105.91
d=0 #dividend yield
K=106 #found above
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945


prices <- c()
steps <- c()


#Call Option
for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}
```

```r
par(mfrow=c(1,2))

plot(steps, prices, xlab = "n steps", ylab = "Call Option Price", type = "l")
abline(h=mean(prices[150:200]), col="blue")



#Put Option

#PUT
#Our parameters
S0=105.91
n=200
d=0 #dividend yield
K=106
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945

prices <- c()
steps <- c()



for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i,
                      putopt = T)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}



plot(steps, prices, xlab = "n steps", ylab = "Put Option Price", type = "l")
abline(h=mean(prices[150:200]), col="blue")
```

Fig 11

```
prices <- c()
steps <- c()

#Call Option
for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i)
  prices <- c(prices,nextopt)
  steps <- c(steps,i)
}

plot(steps, prices, xlab = "n steps", ylab = "Call Option Price", type = "s")



#Black Scholes Simulation
bs = bscall(s = S0, k = K, r = r, v = sig, tt = T_exp, d)

abline(h=bs, col="green")

legend("topright", legend=c("Binomial Model", "Black Scholes Price"), lty = c(1,1),
       col=c("black", "green"), cex=0.8)
```

Option Pricing Comparison

We will start with options expiring on 4-28-23.

```
call5 = read.csv("Project Data/GOOG_call_20230428")
put5 = read.csv("Project Data/GOOG_put_20230428")

call10  = read.csv("Project Data/GOOG_call_20230505")
put10  = read.csv("Project Data/GOOG_put_20230505")

strike = seq(100, 110, 1)

```

```r
bidaskcall5 = call5[call5$Strike >= 100 & call5$Strike <= 110, c("Strike", "Bid", "Ask")]
bidaskput5 = put5[put5$Strike >= 100 & put5$Strike <= 110, c("Strike", "Bid", "Ask")]

callprices <- c()
putprices <- c()
bsprices <- c()
steps <- c()

#Both Options
for (i in strike) {
  nextcopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200)
  nextpopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200,
                       putopt = T)
  bs = bscall(s = S0, k = i, r = r, v = sig, tt = T_exp, d)

  callprices <- c(callprices,nextcopt)
  putprices <- c(putprices, nextpopt)
  bsprices <- c(bsprices, bs)

  steps <- c(steps,i)
}

calldata = kable(data.frame(K = bidaskcall5$Strike, Binomial = round(callprices,2),
                    BSM = round(bsprices,2), Actual_Bid = bidaskcall5$Bid, Actual_Ask = bidaskcall5$Ask)


putdata = kable(data.frame(Binomial = round(putprices,2),
                    Actual_Bid = bidaskput5$Bid, Actual_Ask = bidaskput5$Ask),
                 format = "latex", booktabs = TRUE)
```

Tables 3 and 4:

```
cat(c("\\begin{table}[!htb]
    \\begin{minipage}{.5\\linewidth}
      \\caption{Call Option}
      \\centering",
        calldata,
    "\\end{minipage}%
    \\begin{minipage}{.5\\linewidth}
      \\centering
        \\caption{Put Option}",
        putdata,
    "\\end{minipage}
\\end{table}"
))
```

Now we will repeat for 5-5-23 expiry:

```
bidaskcall10 = call10[call10$Strike >= 100 & call10$Strike <= 110, c("Strike", "Bid", "Ask")]
bidaskput10 = put10[put10$Strike >= 100 & put10$Strike <= 110, c("Strike", "Bid", "Ask")]

callprices <- c()
putprices <- c()
bsprices <- c()
steps <- c()

T_exp = 10/252

#Both Options
for (i in strike) {
  nextcopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200)
  nextpopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200,
                      putopt = T)
  bs = bscall(s = S0, k = i, r = r, v = sig, tt = T_exp, d)

  callprices <- c(callprices,nextcopt)
```

```
  putprices <- c(putprices, nextpopt)
  bsprices <- c(bsprices, bs)


  steps <- c(steps,i)
}


calldata = kable(data.frame(K = bidaskcall10$Strike, Binomial = round(callprices,2),
                    BSM = round(bsprices,2), Actual_Bid = bidaskcall10$Bid, Actual_Ask = bidaskcall5$Ask



putdata = kable(data.frame(Binomial = round(putprices,2),
                    Actual_Bid = bidaskput10$Bid, Actual_Ask = bidaskput10$Ask),
                 format = "latex", booktabs = TRUE)
```

Tables 5 and 6:

```
cat(c("\\begin{table}[!htb]
    \\begin{minipage}{.5\\linewidth}
      \\caption{Call Option}
      \\centering",
        calldata,
    "\\end{minipage}%
    \\begin{minipage}{.5\\linewidth}
      \\centering
        \\caption{Put Option}",
        putdata,
    "\\end{minipage}
\\end{table}"
))
```

Volatility effect:

Fig 12

```
par(mfrow=c(1,2))

iv5 = call5[c("Strike", "IV")][call5$Strike >= 90 & call5$Strike <= 120, ]
plot(x=iv5$Strike/S0, y = iv5$IV, ylim = c(0,1), main = "April 28th Expiry", xlab = "K/S0", ylab = "Impl
abline(h= sig, col = "blue")

legend("topright", legend=c("IV", "Historical Vol"), lty = c(NA,1), pch = c(1,NA),
       col=c("black", "blue"), cex=0.8)

iv10 = call10[c("Strike", "IV")][call10$Strike >= 90 & call10$Strike <= 130, ]
plot(x=iv10$Strike/S0, y = iv10$IV, ylim = c(0,1), main = "May 5th Expiry", xlab = "K/S0", ylab = "Impli
abline(h= sig, col = "blue")



legend("topright", legend=c("IV", "Historical Vol"), lty = c(NA,1), pch = c(1,NA),
       col=c("black", "blue"), cex=0.8)
```

Hedging Analysis

Fig 13

```
delta <- c()
gamma <- c()
theta <- c()
steps <- c()
prices <- c()

S0=105.91
n=200
d=0 #dividend yield
K=106
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945
```

```r
#Call Option
for (i in 1:200) {
  nextopt <- binomopt(s = S0, k = K, r = r, v = sig, tt = T_exp, d, nstep = i, returngreeks = T)
  prices <- c(prices,nextopt[1])
  delta <- c(delta, nextopt[2])
  gamma <- c(gamma, nextopt[3])
  theta <- c(theta, nextopt[4])

  steps <- c(steps,i)
}

par(mfrow=c(1,3))
plot(steps, delta, xlab = "n steps", ylab = "Call Option Delta", type = "l")
plot(steps, gamma, xlab = "n steps", ylab = "Call Option Gamma", type = "l")
plot(steps, theta, xlab = "n steps", ylab = "Call Option Theta", type = "l")
```

Now let's observe delta and gamma for different strike prices:

Fig 14

```r
delta <- c()
gamma <- c()
theta <- c()
strikes <- c()
prices <- c()

S0=105.91
n=200
d=0 #dividend yield
K=106
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945
```

```r
#Call Option
for (i in 90:150) {
  nextopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200, returngreeks = T)
  prices <- c(prices,nextopt[1])
  delta <- c(delta, nextopt[2])
  gamma <- c(gamma, nextopt[3])

  strikes <- c(strikes,i)
}

par(mfrow=c(1,2))
plot(strikes, delta, xlab = "K", ylab = "Call Option Delta", type = "l")
plot(strikes, gamma, xlab = "K", ylab = "Call Option Gamma", type = "l")
```

We will repeat the same analysis for put options:

Fig 15

```r
delta <- c()
gamma <- c()
theta <- c()
strikes <- c()
prices <- c()

S0=105.91
n=200
d=0 #dividend yield
K=106
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945

#Call Option
for (i in 90:150) {
```

```
  nextopt <- binomopt(s = S0, k = i, r = r, v = sig, tt = T_exp, d, nstep = 200, returngreeks = T, putop
  prices <- c(prices,nextopt[1])
  delta <- c(delta, nextopt[2])
  gamma <- c(gamma, nextopt[3])

  strikes <- c(strikes,i)
}

par(mfrow=c(1,2))
plot(strikes, delta, xlab = "K", ylab = "Put Option Delta", type = "l")
plot(strikes, gamma, xlab = "K", ylab = "Put Option Gamma", type = "l")
```

Let's examine the stock prices throughout the week that we will be hedging:

```
GOOG = read.csv("Project Data/GOOG_historical_prices")
stock = GOOG[GOOG$dates >= "2023-04-21", ]
stock = stock[,c("dates", "GOOG.Close")]
kable(stock[,c("dates", "GOOG.Close")], col.names = c("Date", "Stock Price"))
```

We are now ready to construct this portfolio:

```
calls = c( 2.87, 3.10, 2.20, 0.57, 2.50, 2.16) #FILL IN WITH REAL DATA
stock = stock
delta = c()
prices = c()

S0=105.91
n=200
d=0 #dividend yield
K=106
T_exp = 5/252
sig = .3798609 #3 month volatility for 3 month option
r=.04945

for(i in seq(5,0,-1)) {
```

```r
  nextopt <- binomopt(s = stock[5-i+1,2], k = K, r = r, v = sig, tt = i/252, d, nstep = 200, returngreek
  delta <- c(delta, nextopt[2])

}

calldiff = calls[2:length(calls)] - calls[1:length(calls)-1]

calldiff = c(NA, calldiff)

stockdiff = stock$GOOG.Close[2:length(stock$GOOG.Close)] -
          stock$GOOG.Close[1:length(stock$GOOG.Close)-1]


portdiff = calldiff[-1] - delta[-length(delta)]*(stockdiff)

portdiff = c(NA, portdiff)

test = data.frame(stock$dates, stock$GOOG.Close, calls, delta, calldiff, portdiff)
colnames(test) = c("Day", "S0", "Call Price", "Delta", "C(t) - C(t-1)", "P(t) - P(t-1)")

kable(test, col.names = c("Day", "$S_0$", "Call Price", "Delta", "$C(t) - C(t-1)$", "$\\Pi(t) - \\Pi(t-1
```