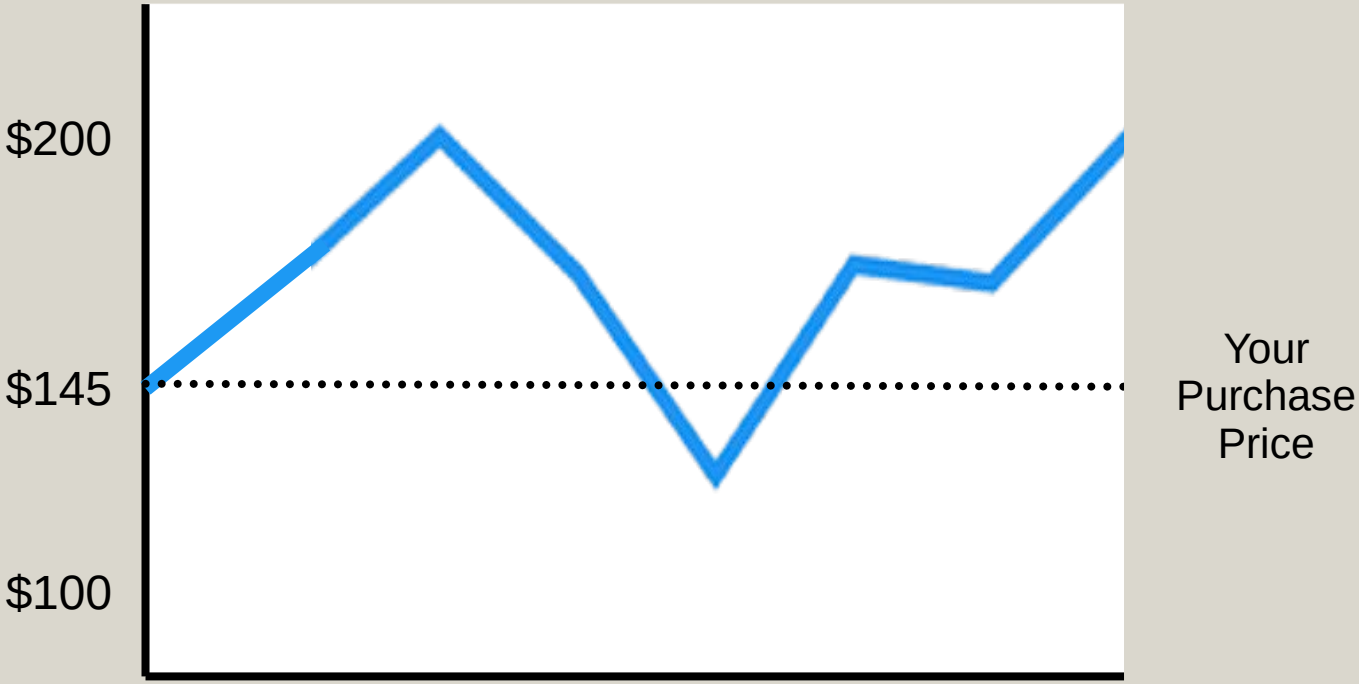
Options Theory & Collecting/Analyzing Data

Kim Pham

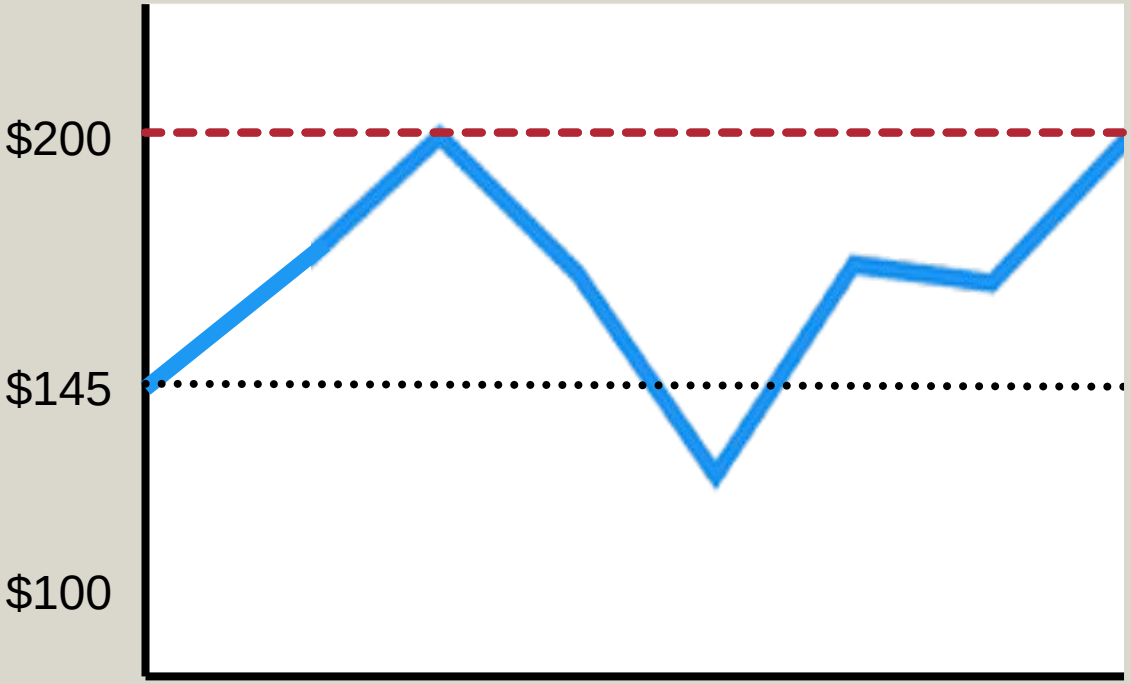
What is an Option?

- Let’s first understand how stocks work



What is an Option?

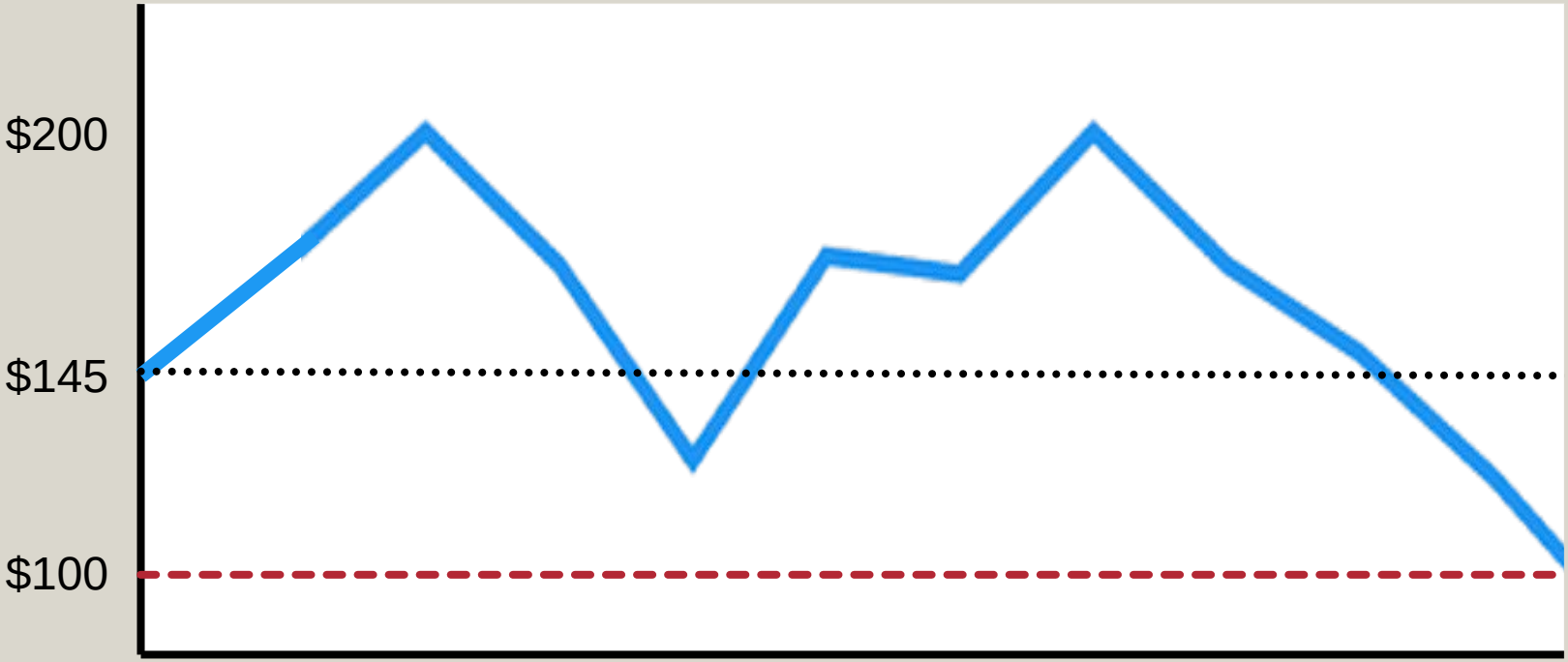
- Let’s first understand how stocks work



If the stock goes up:
$200 - $145 = $55 profit

What is an Option?

- Let’s first understand how stocks work

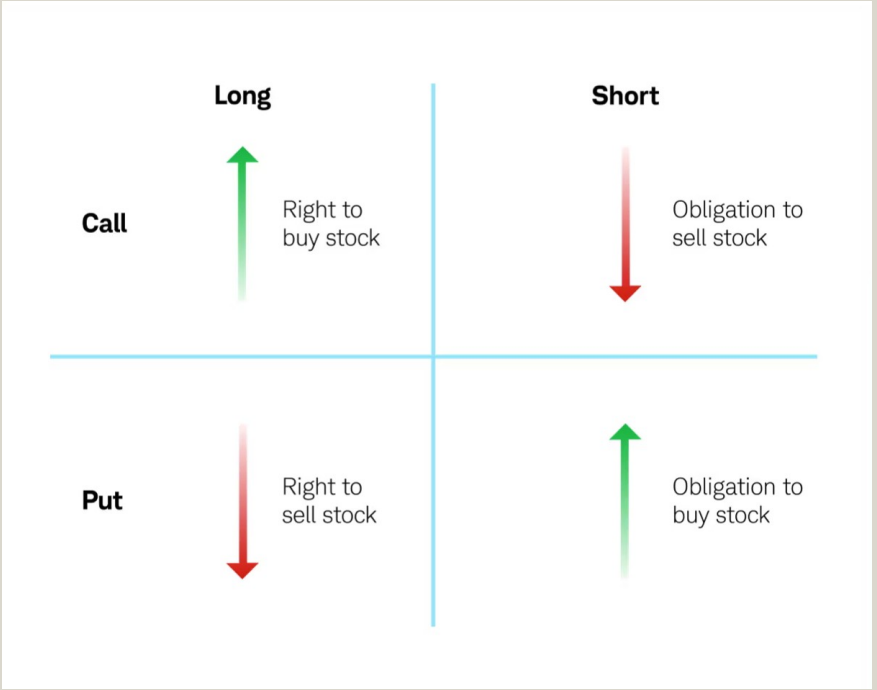


Your Purchase Price

If the stock goes down: $100 - $145 = $45 loss

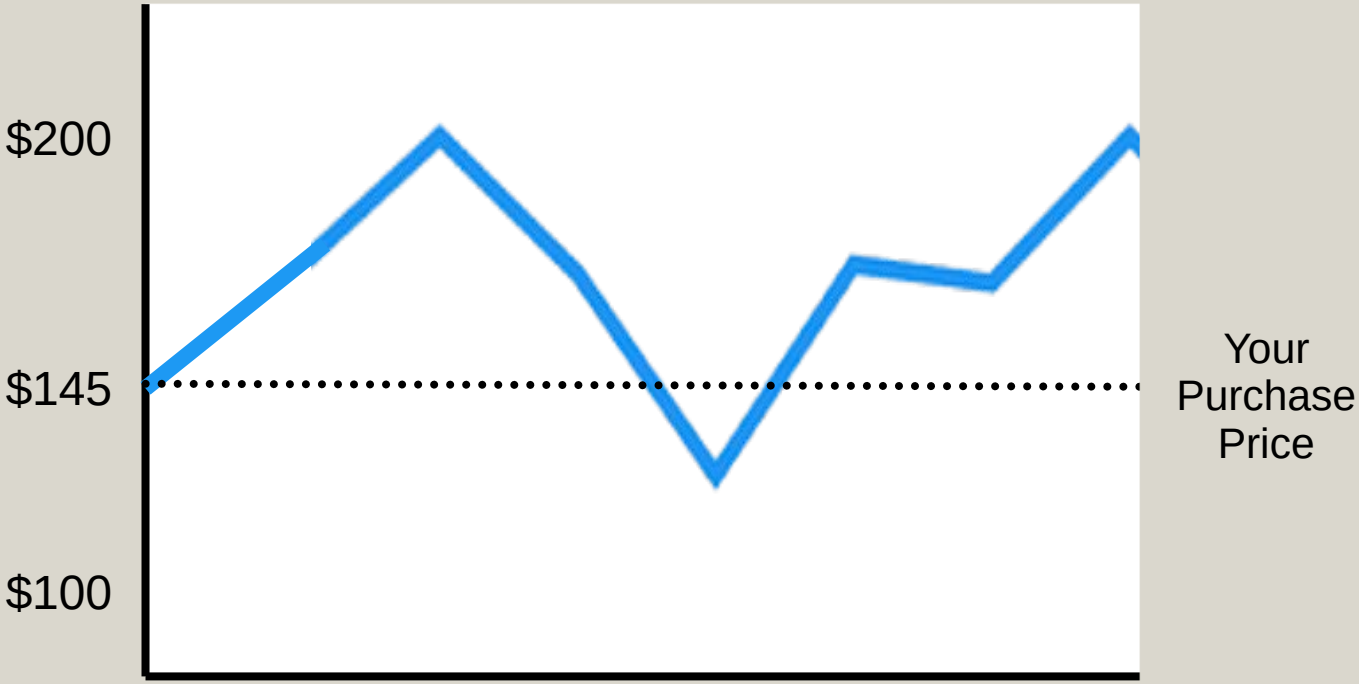
What is an Option? Cont.

- Options are similar but different
- Put Option: Right to sell stocks at the strike price up to expiry
- Call Option: Right to buy stocks at the strike price up to expiry
- You can either long (buy) or short (write/sell) a put/call



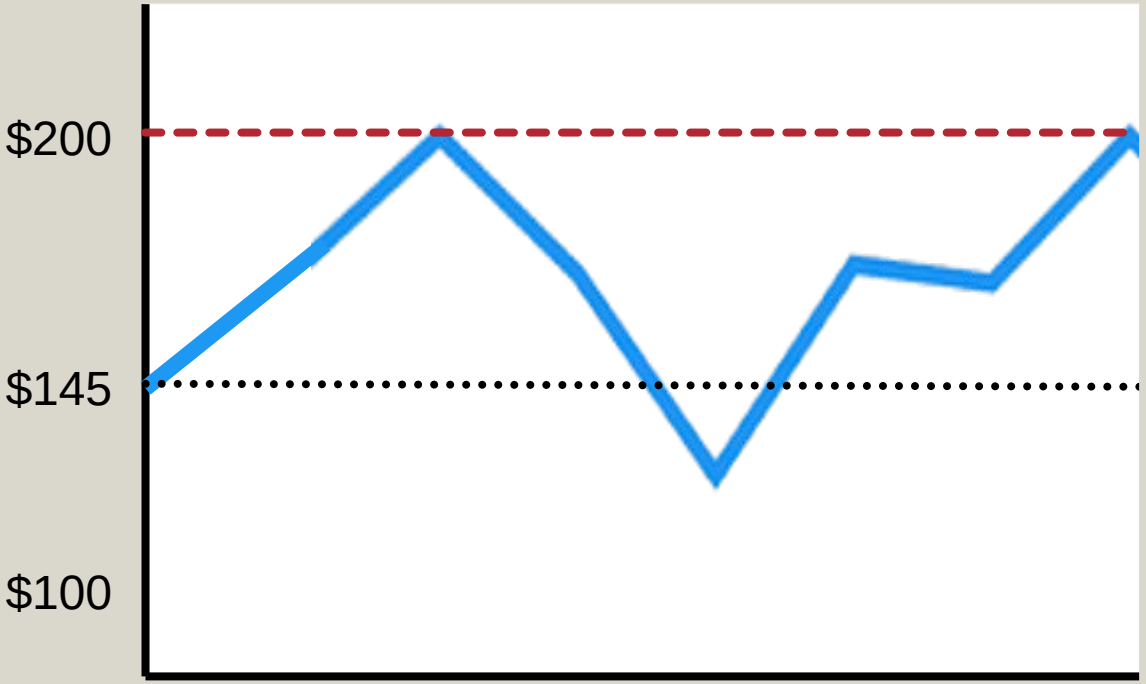
What is an Option? Cont.

- So what if you decide to buy a call option for $5 instead of buying the underlying directly?



What is an Option? Cont.

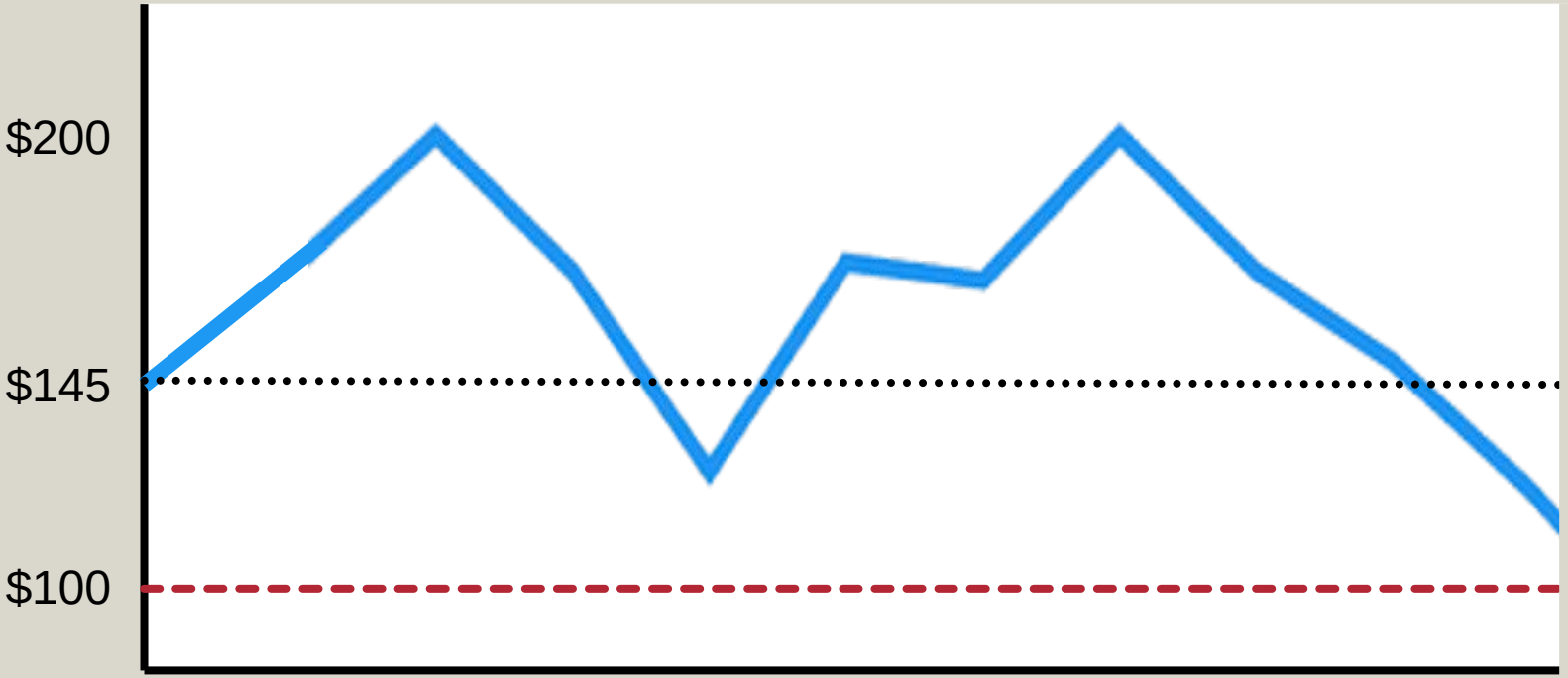
- So what if you decide to buy a call option for $5 instead of buying the underlying directly?



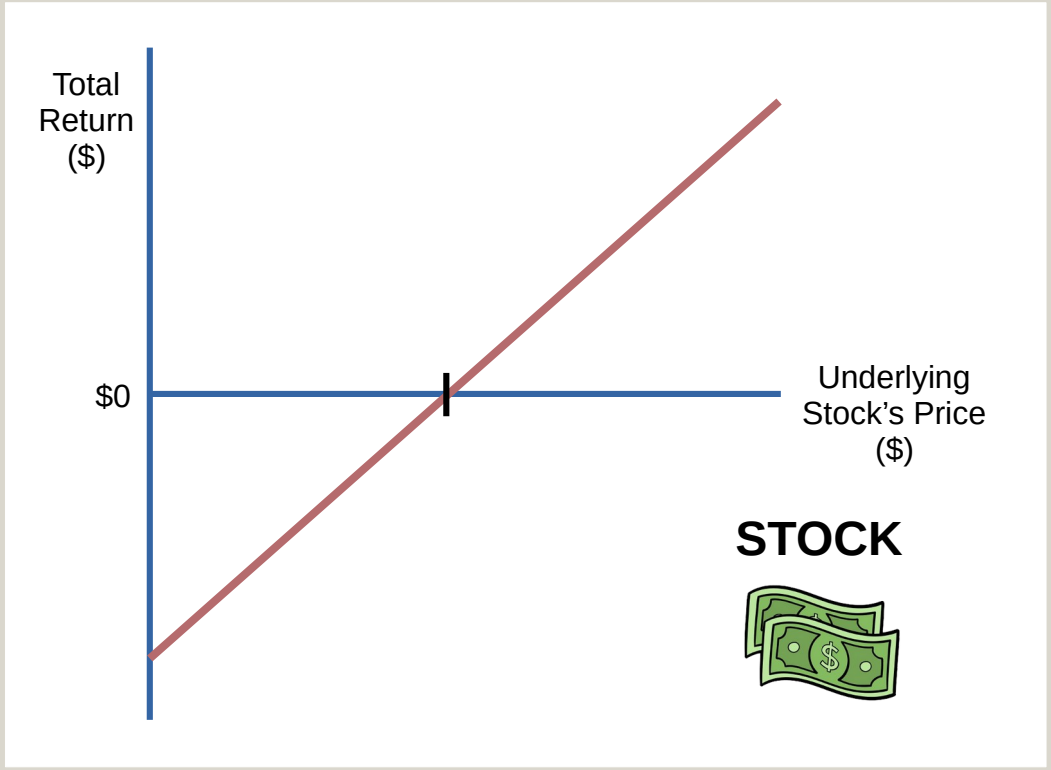
If the stock goes up:
$200 - $145 - $5 = $50 profit

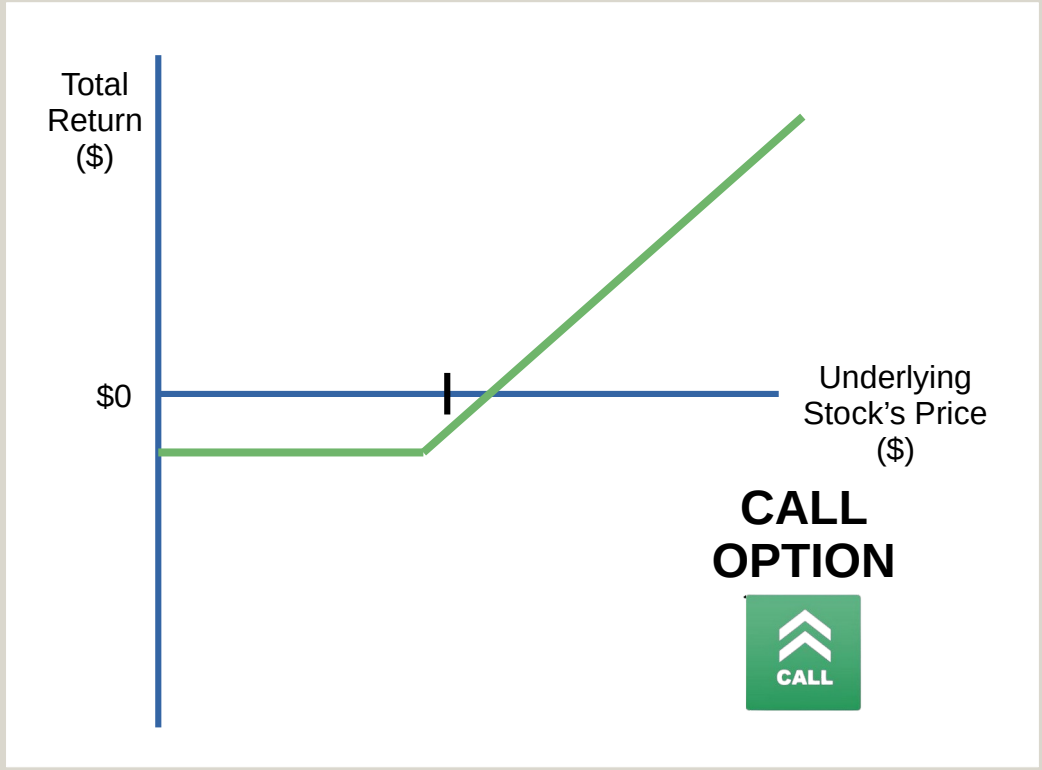
What is an Option? Cont.

- So what if you decide to buy a call option for $5 instead of buying the underlying directly?

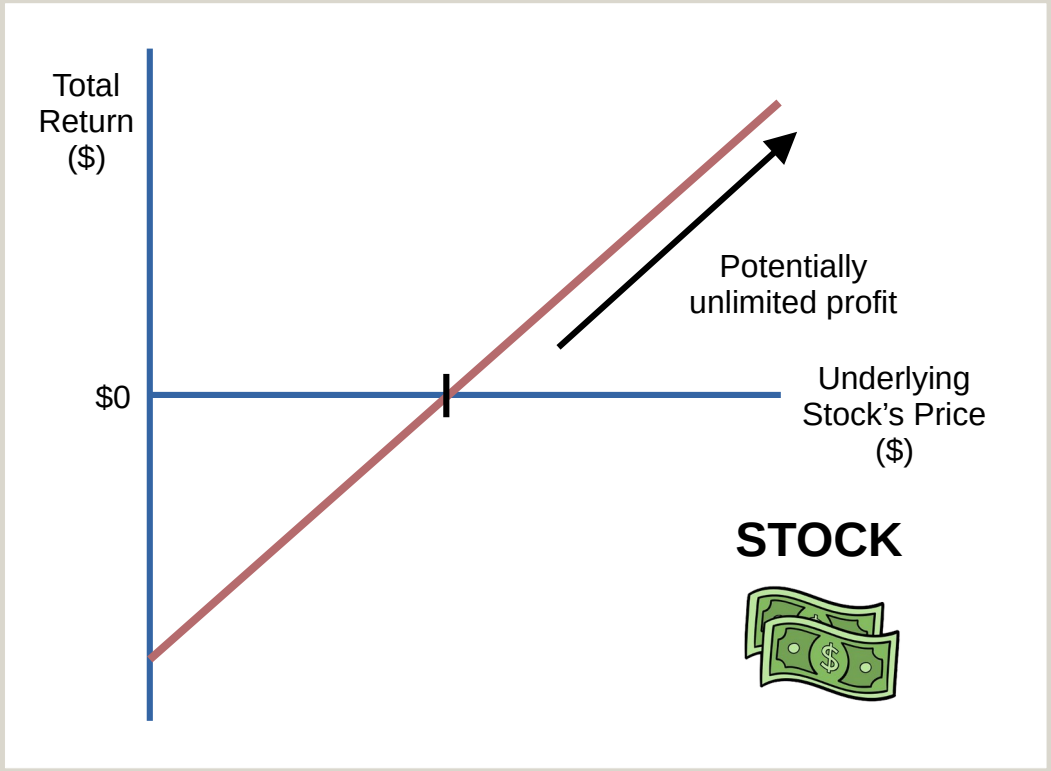


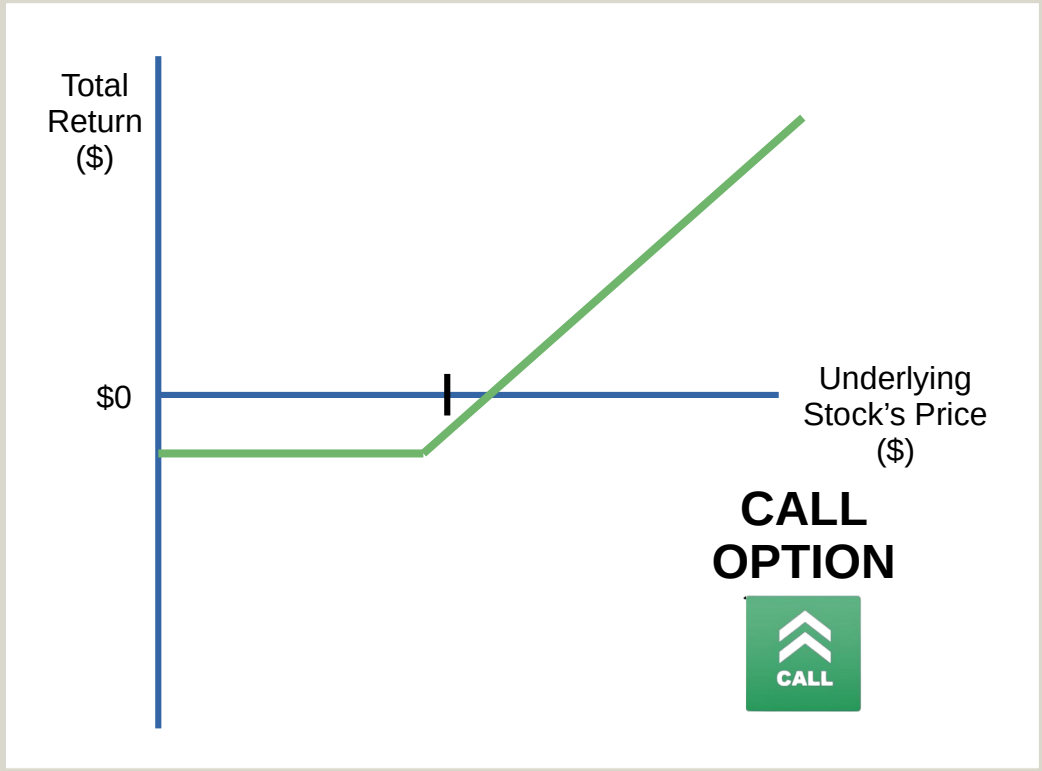
Wait...Why do I Even Care?



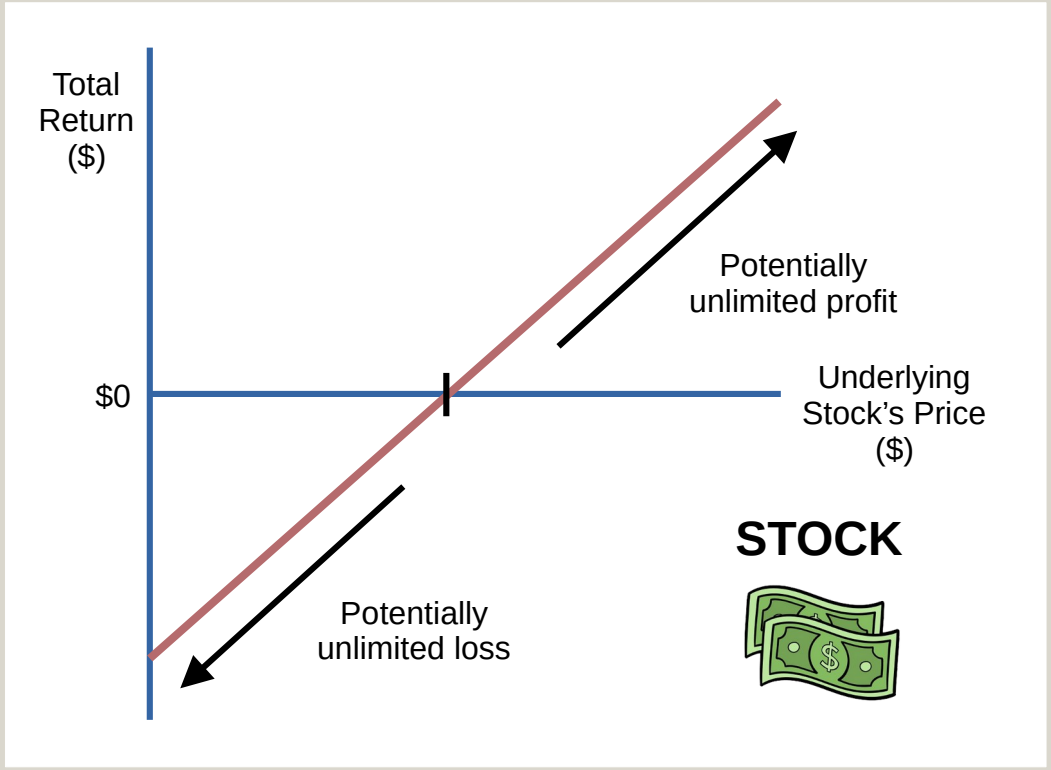


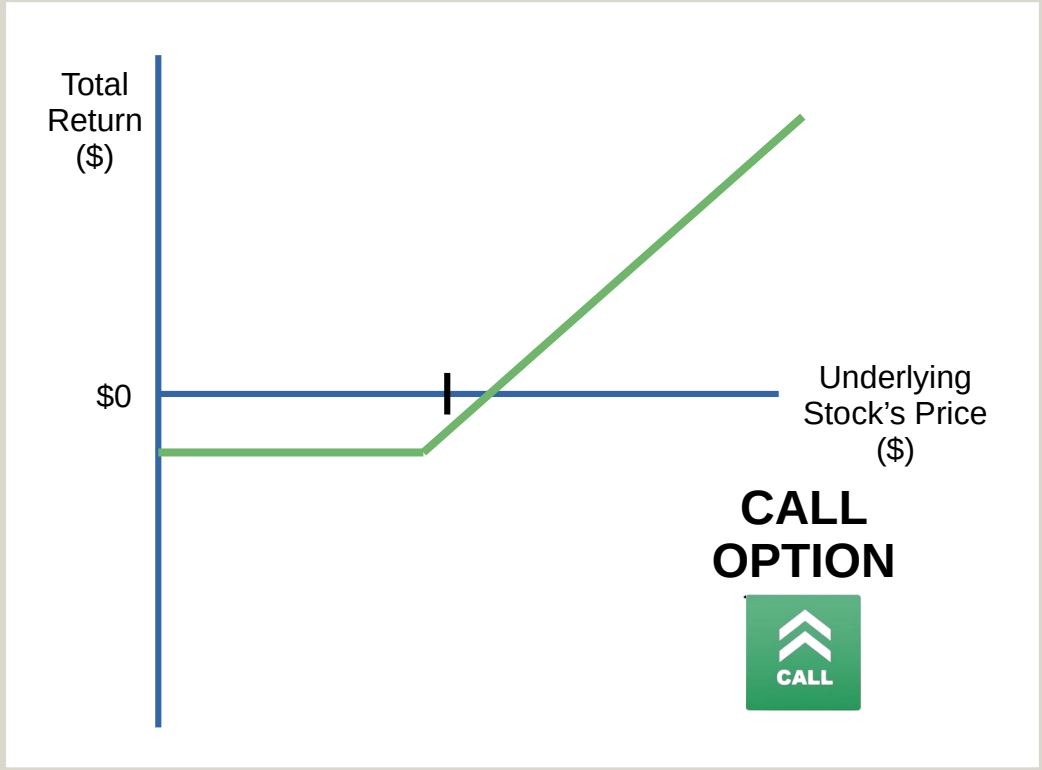
Wait...Why do I Even Care?



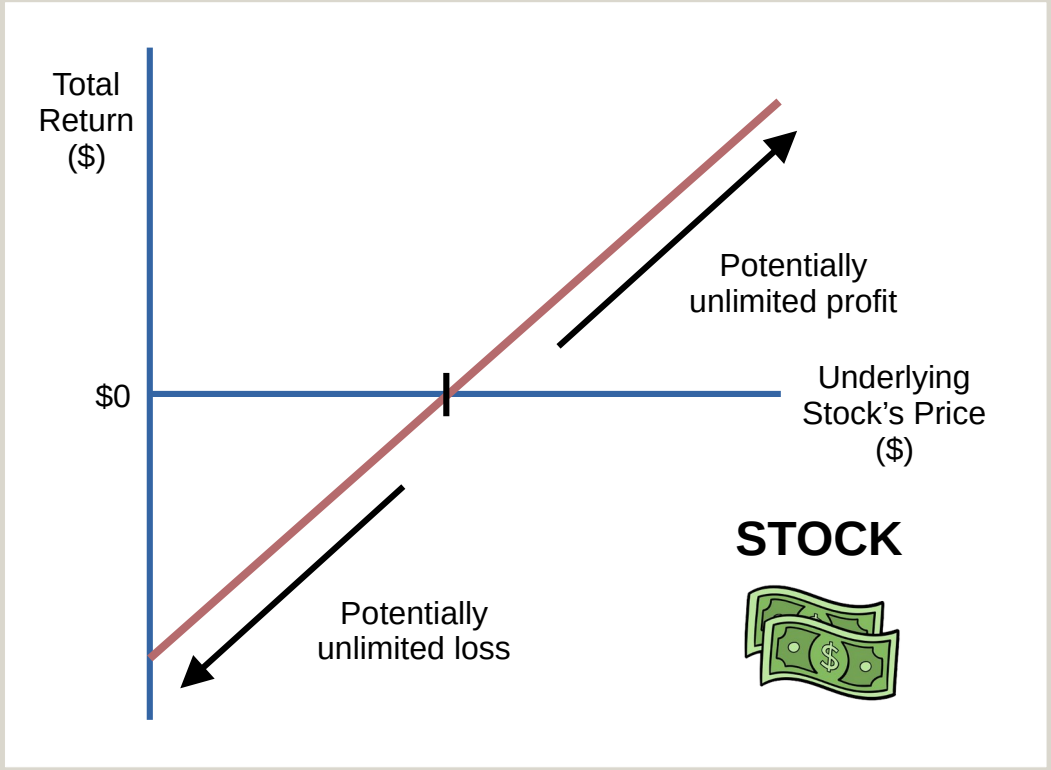


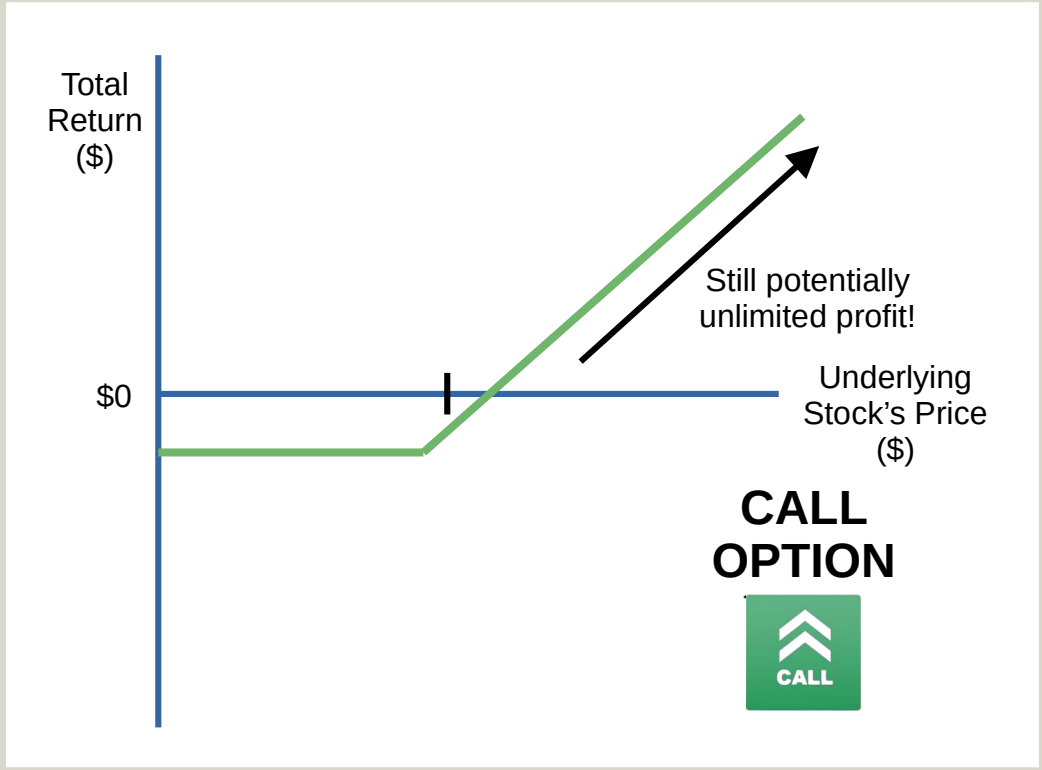
Wait...Why do I Even Care?



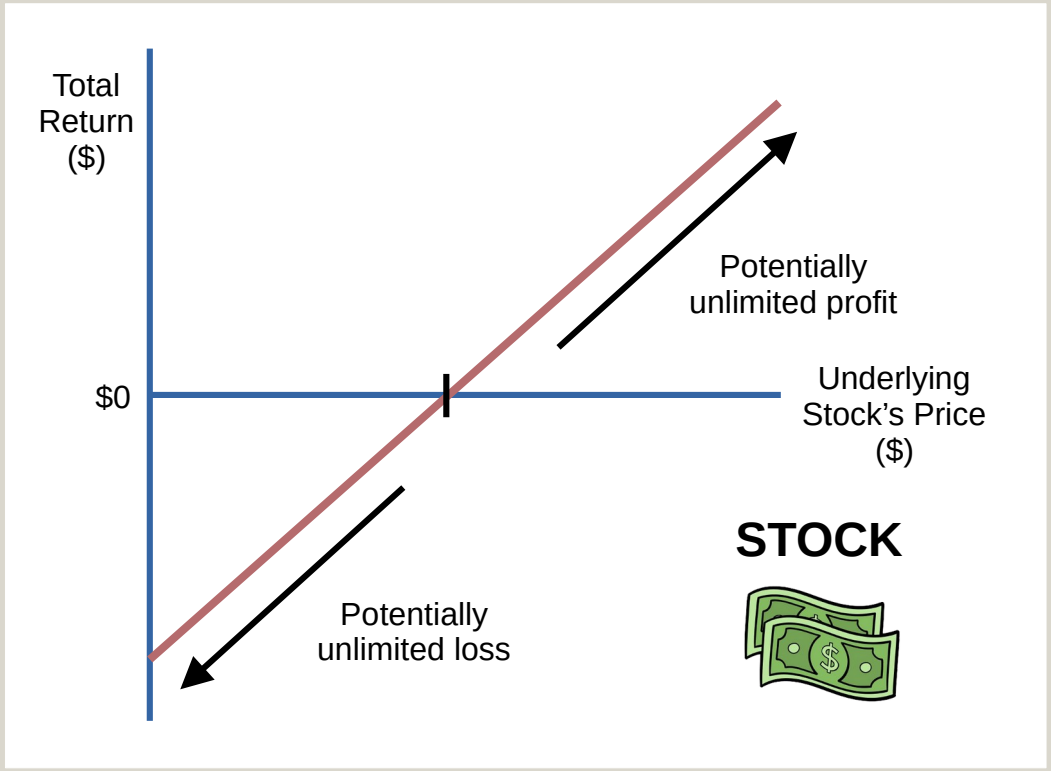


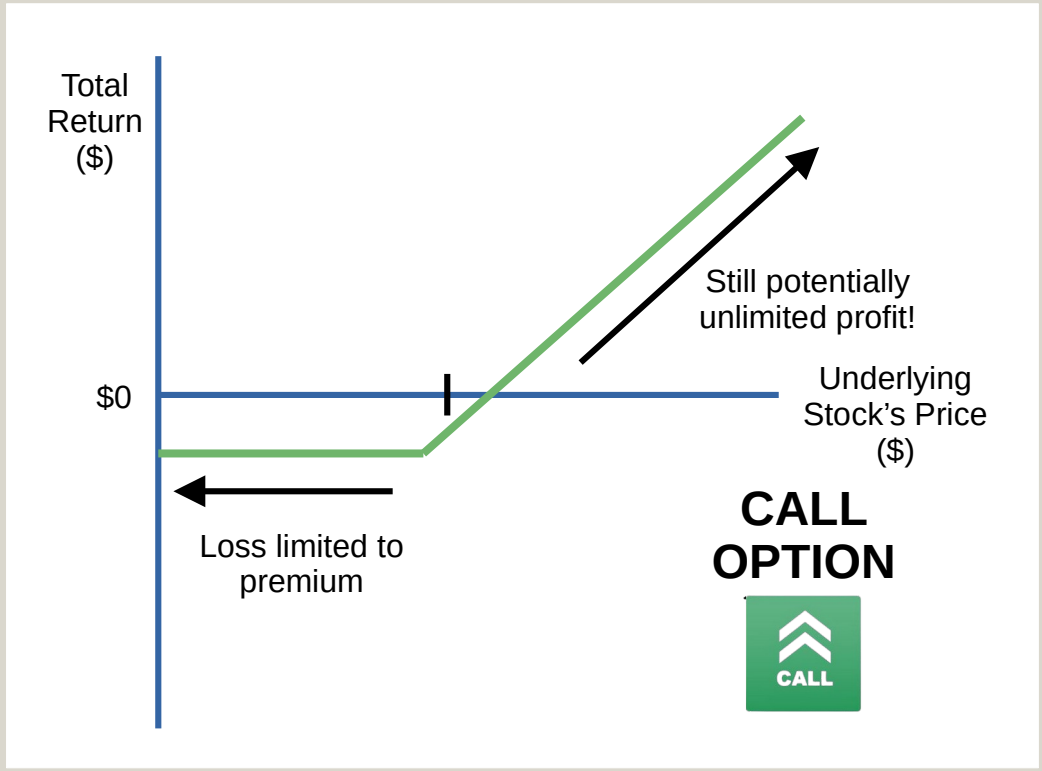
Wait...Why do I Even Care?





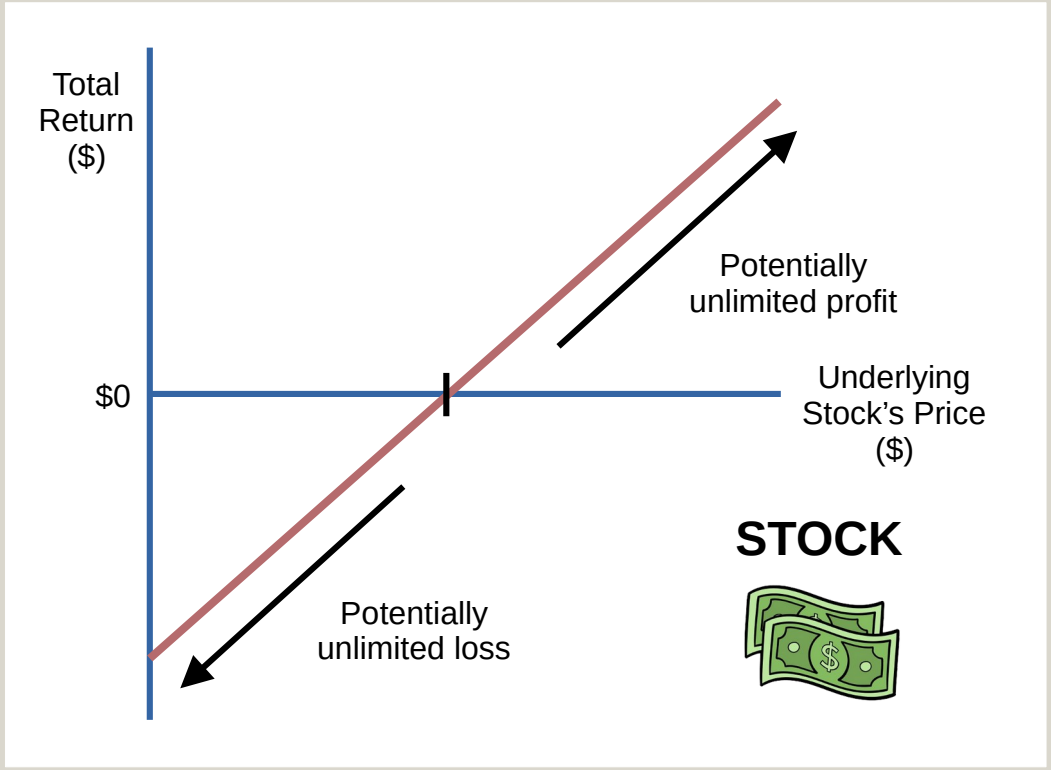
Wait...Why do I Even Care?

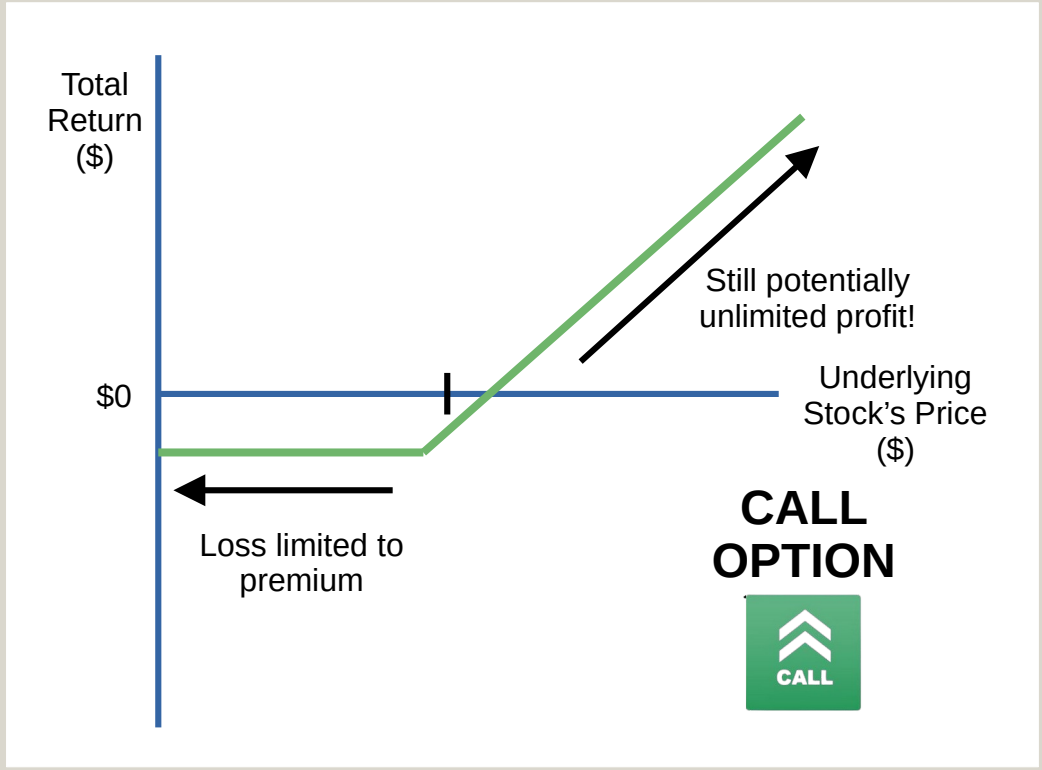


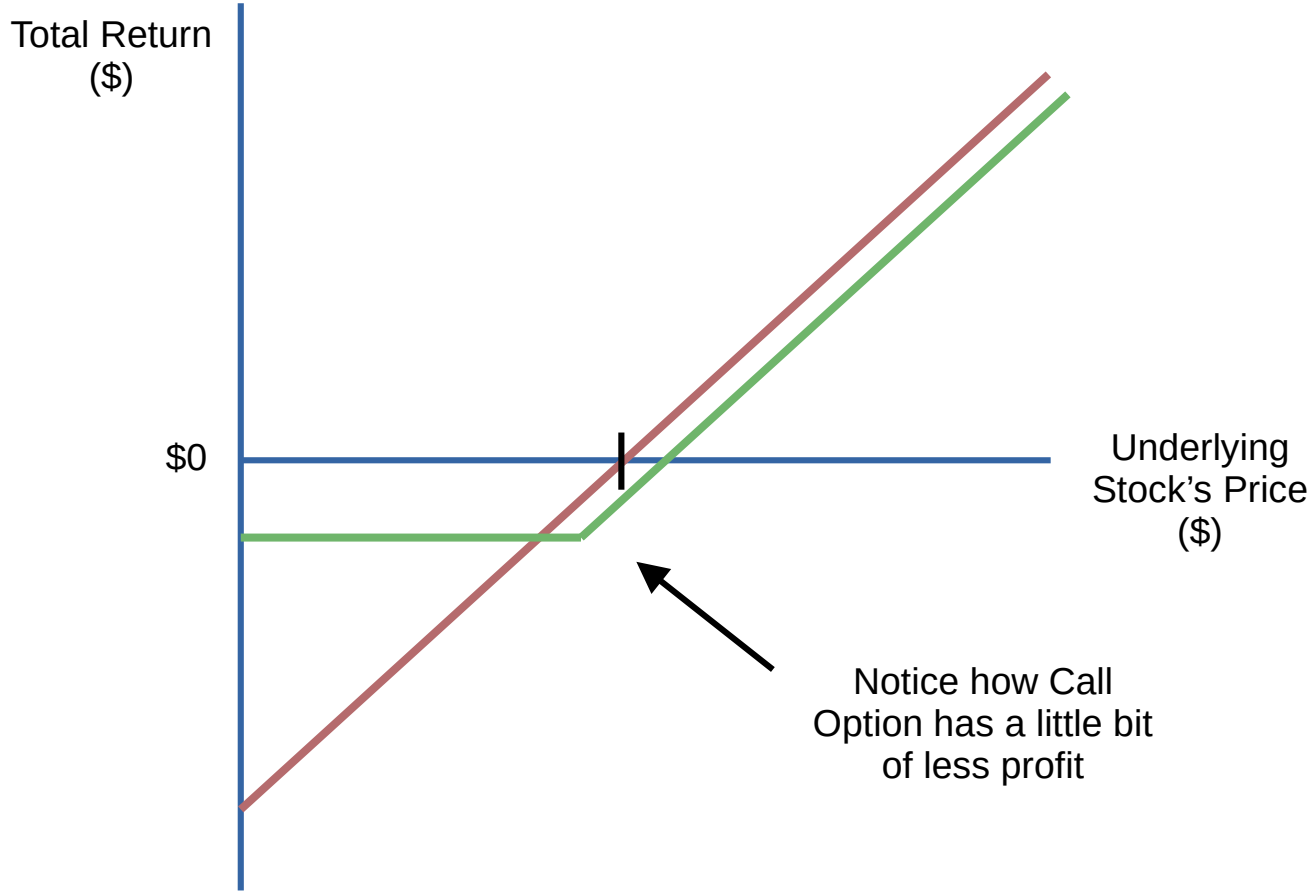


Wait...Why do I Even Care?

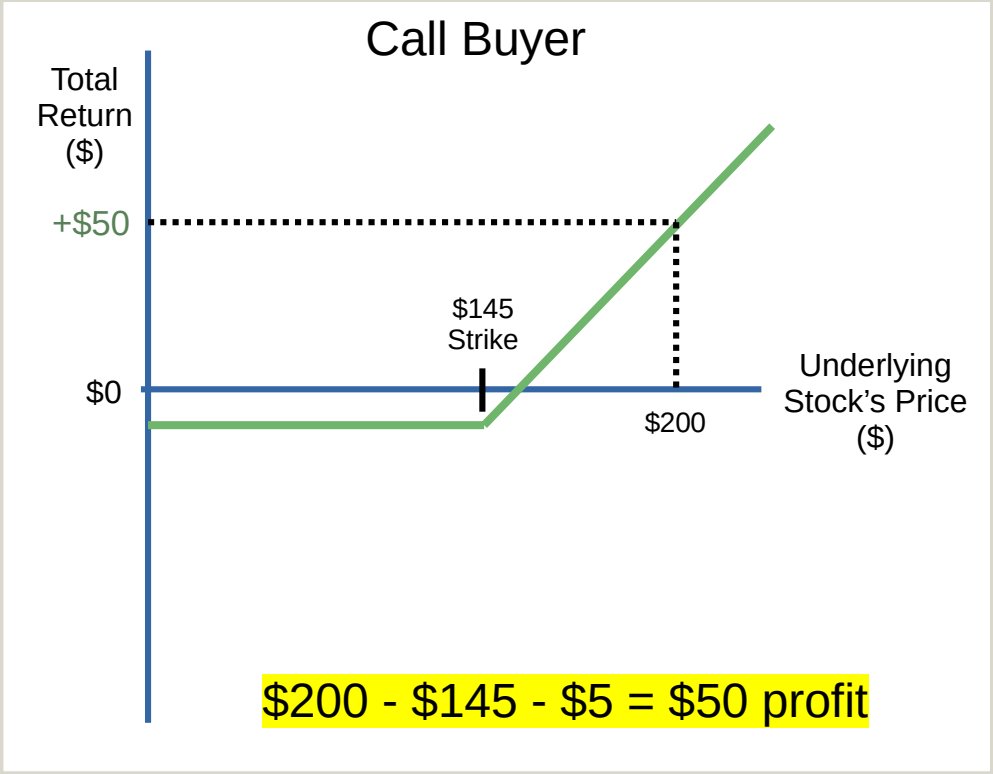
Unlimited profit with limited risks! Who doesn't like that?

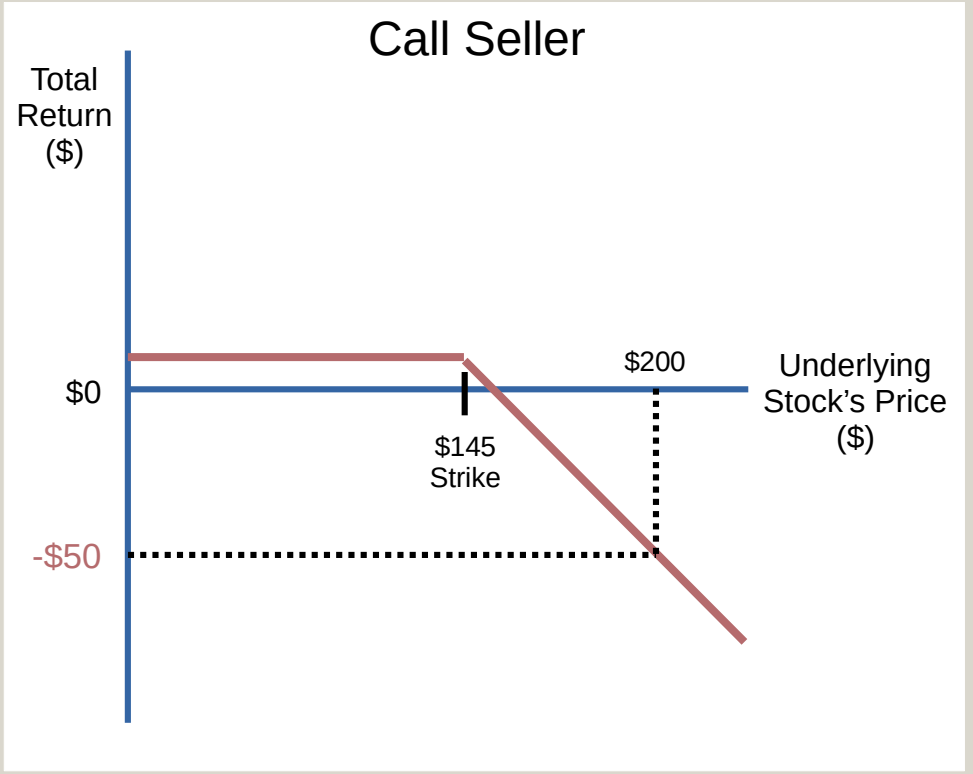




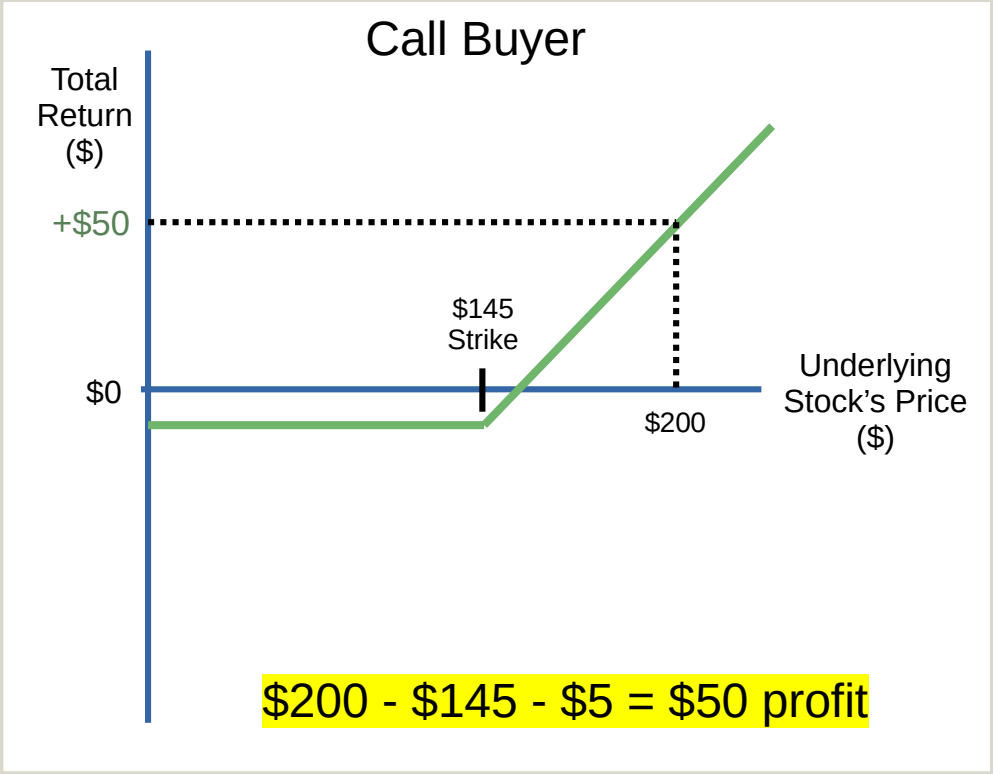

Total Return
($)
$0
Underlying
Stock's Price
($)
Notice how Call
Option has a little bit
of less profit

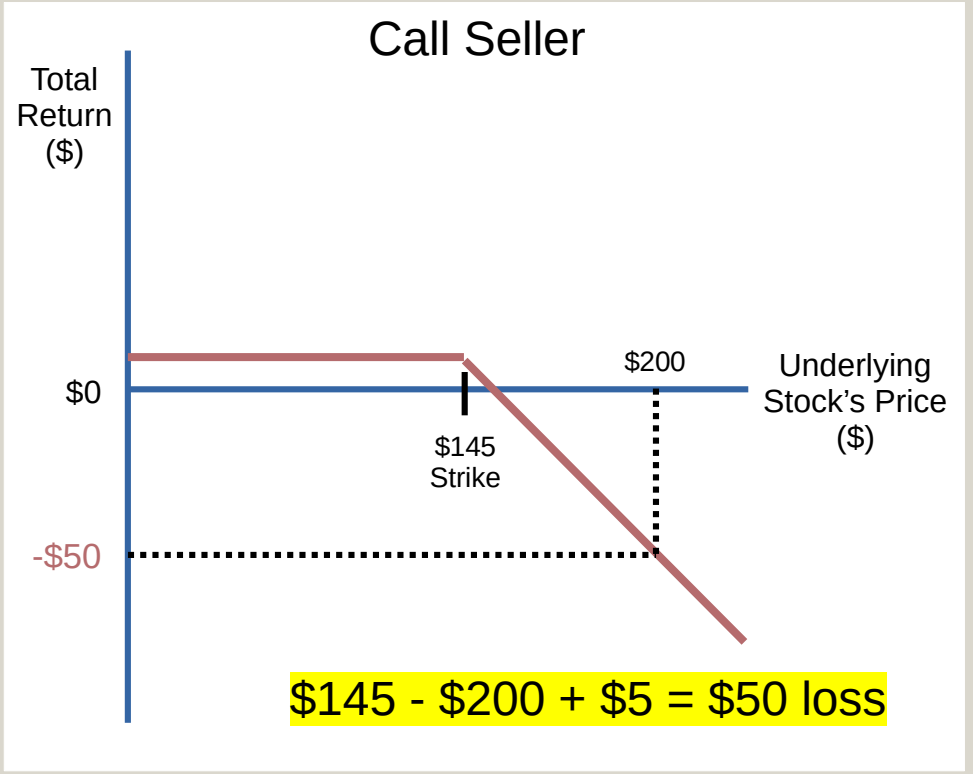
Call Options



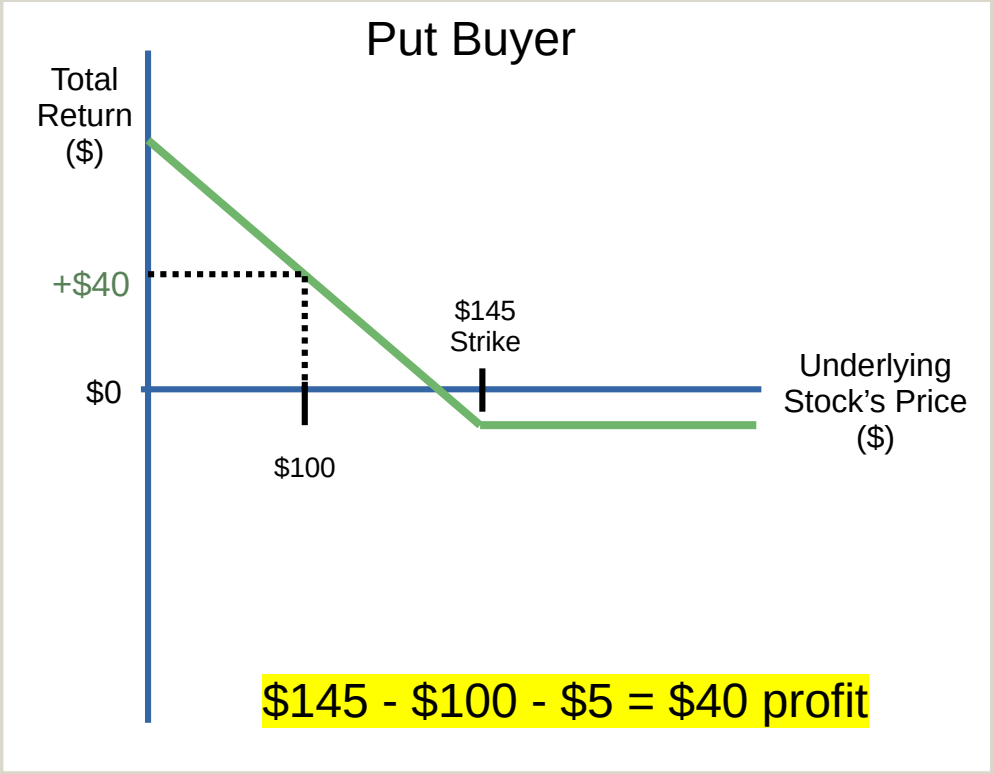


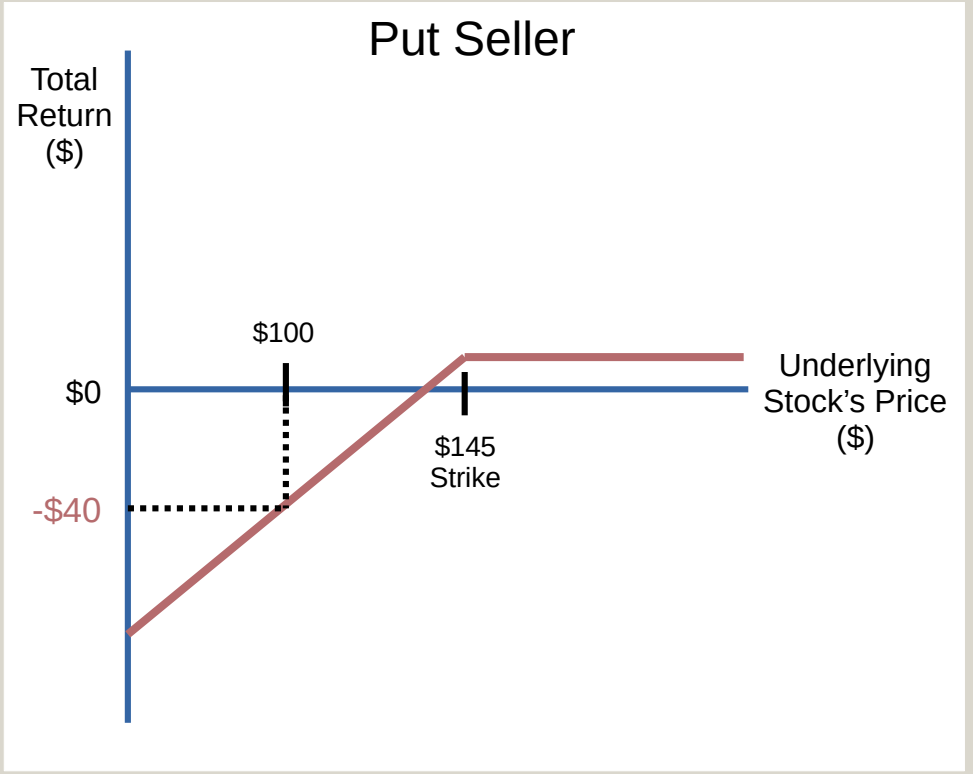
Call Options



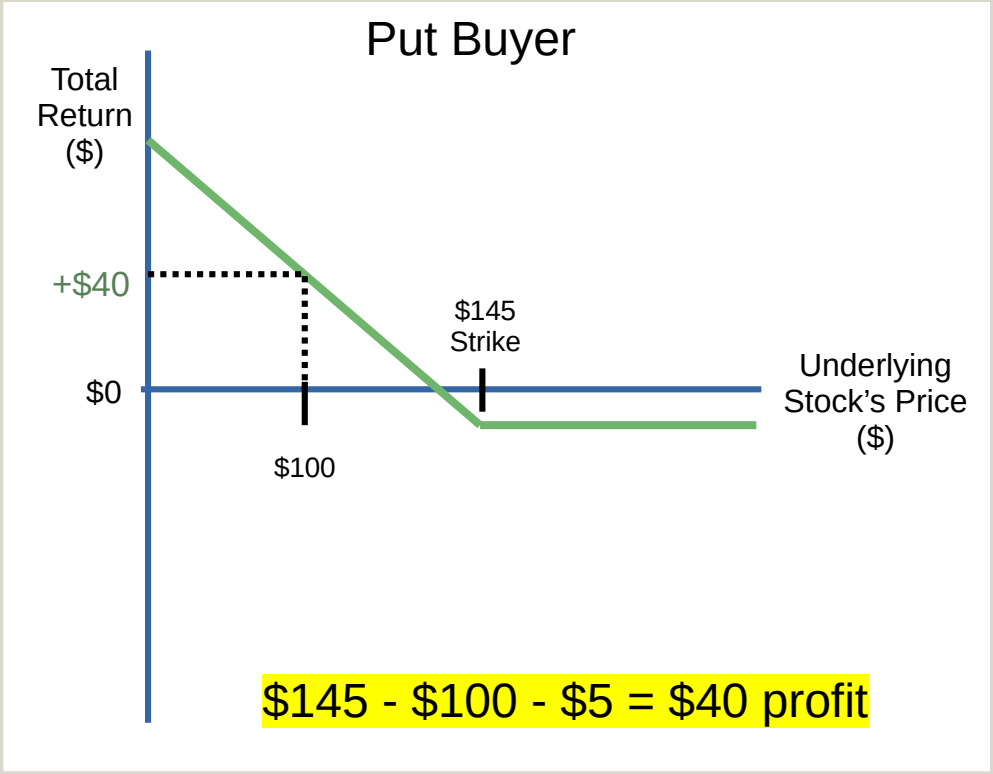


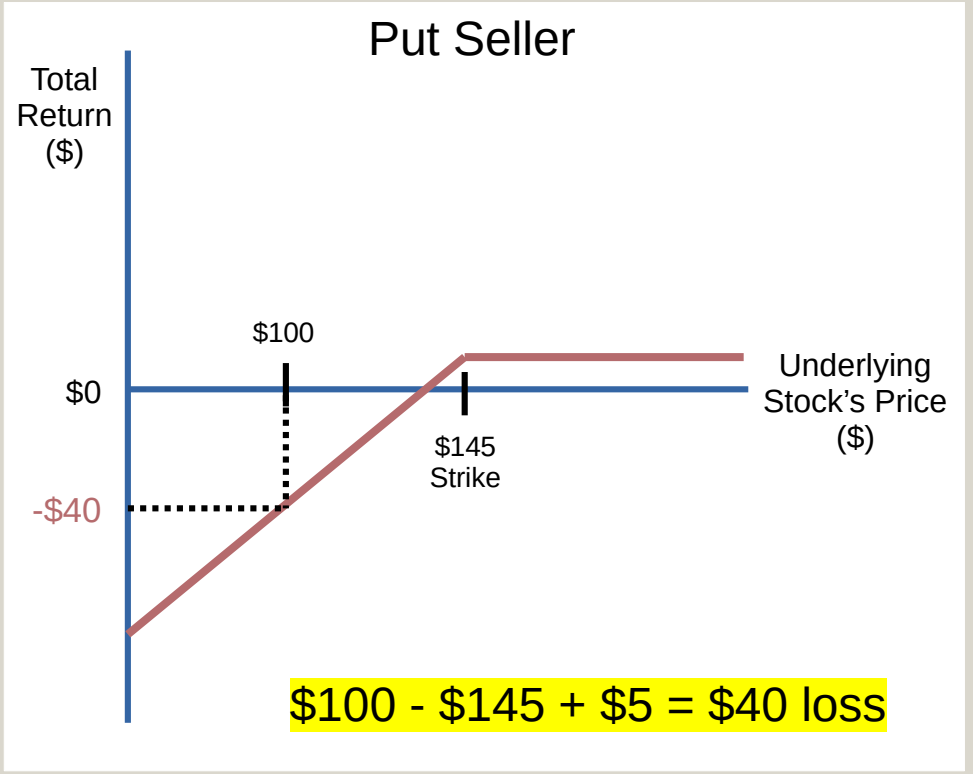
Put Options


Put Buyer
Total Return ($)
+$40
$0
$145 Strike
$100
Underlying Stock's Price ($)
$145 - $100 - $5 = $40 profit


Put Seller
Total Return ($)
$100
$0
$145 Strike
-$40
Underlying Stock's Price ($)

Put Options


Put Buyer
Total Return ($)
+$40
$0
$145 Strike
$100
Underlying Stock's Price ($)
$145 - $100 - $5 = $40 profit


Put Seller
Total Return ($)
$100
$0
$145 Strike
-$40
Underlying Stock's Price ($)
$100 - $145 + $5 = $40 loss

Options Styles

Project Background

- What is the goal of my project?
- Data collection issues
- Calculating implied volatility
- Graphing volatility surfaces
- Plotting volatility smiles

Implied Volatility

- What is implied volatility and the Black-Scholes model?
- N = CDF of normal distribution
- S = Price of underlying
- K = strike price
- r = risk-free interest rate
- T = time to maturity
- σ = volatility

$$C(S_t, t) = S_t N(d_1) - Ke^{-r(T-t)} N(d_2)$$

$$P(S_t, t) = Ke^{-r(T-t)} N(-d_2) - S_t N(-d1)$$

$$d_1 = \frac{\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

```
#-- Black-Scholes formula to find price --#
def bs(S, K, T, r, sigma, flag):
    # Calculates d1 and d2
    d1 = np.float64(np.log(S / K) + (r + 0.5 * np.float64(sigma)**2) * T) / (np.float64(sigma) * np.sqrt(T))
    d2 = np.float64(d1 - np.float64(sigma) * np.sqrt(T))

    # Selects formula based on call or put
    if flag == 'C':
        return S * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    else:
        return K * np.exp(-r * T) * norm.cdf(-d2) - S * norm.cdf(-d1)
```

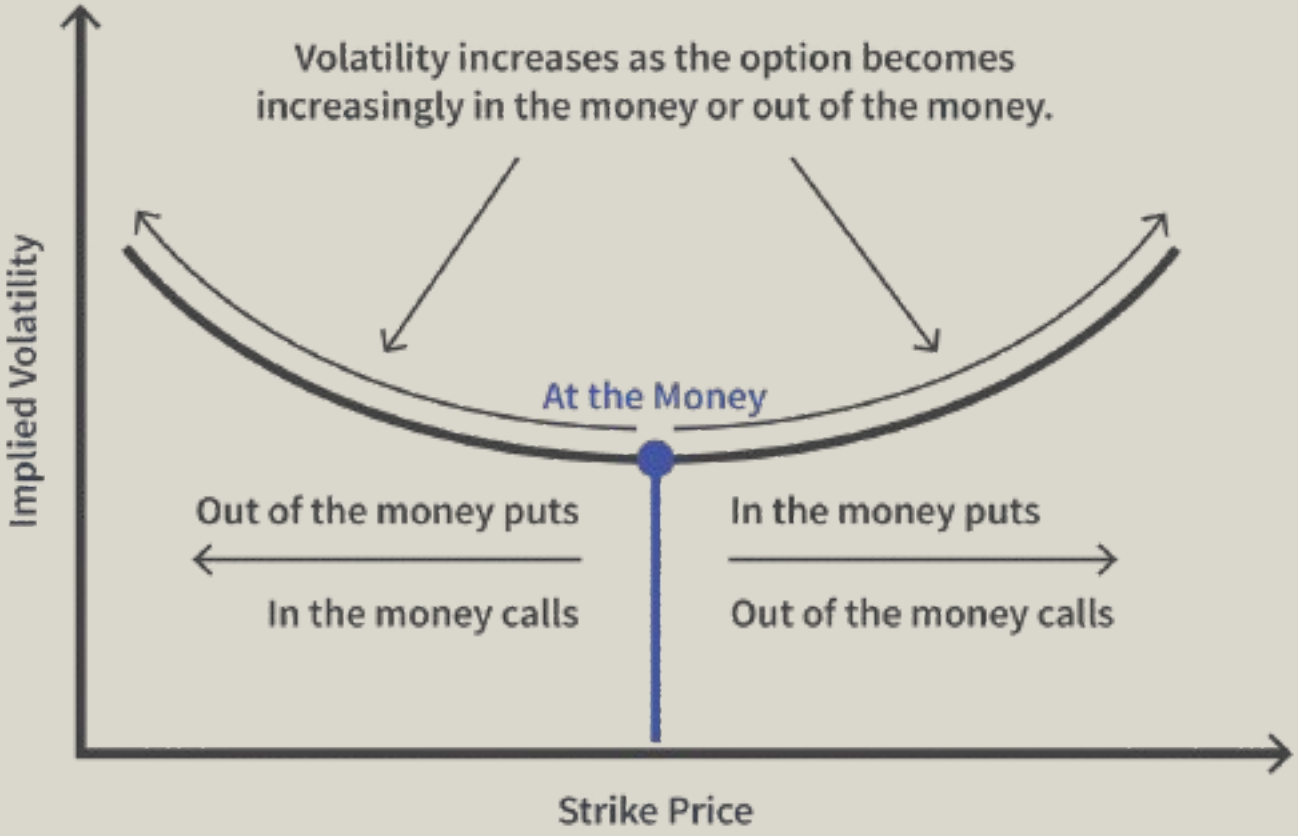
$$Vega = S\sqrt{T - t}\, N'(d_1)$$

```
#-- Calculates options Greek Vega --#
def vega(S, K, T, r, sigma):
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    vega = S * norm.pdf(d1) * np.sqrt(T)

    if vega == 0:
        vega = 0.1

    return vega
```

```python
#-- Implied volatility using Newton's method --#
def iv(row):
    flag = row['P/C']
    S = row['ADJ-CLOSE']
    K = row['STRIKE-PRICE']
    T = row['TIME-TO-EXPIRATION']/365.0
    r = 0.0
    market_price = row['MARK-PRICE']

    # Initial sigma guess based on historical volatility
    sigma = 5
    min_sigma = 1e-4
    max_sigma = 5
    price_diff = 0.05

    # Newton's method with max iterations to find sigma
    while price_diff > 1e-6:
        price = bs(S, K, T, r, sigma, flag)
        vega_val = vega(S, K, T, r, sigma)
        price_diff = price - market_price

        sigma_update = sigma - price_diff / vega_val

        # Ensure sigma stays within bounds
        sigma = max(min_sigma, min(max_sigma, sigma_update))

        if sigma==0.0001:
            return 0.0

    print(row.name)
    return sigma
```

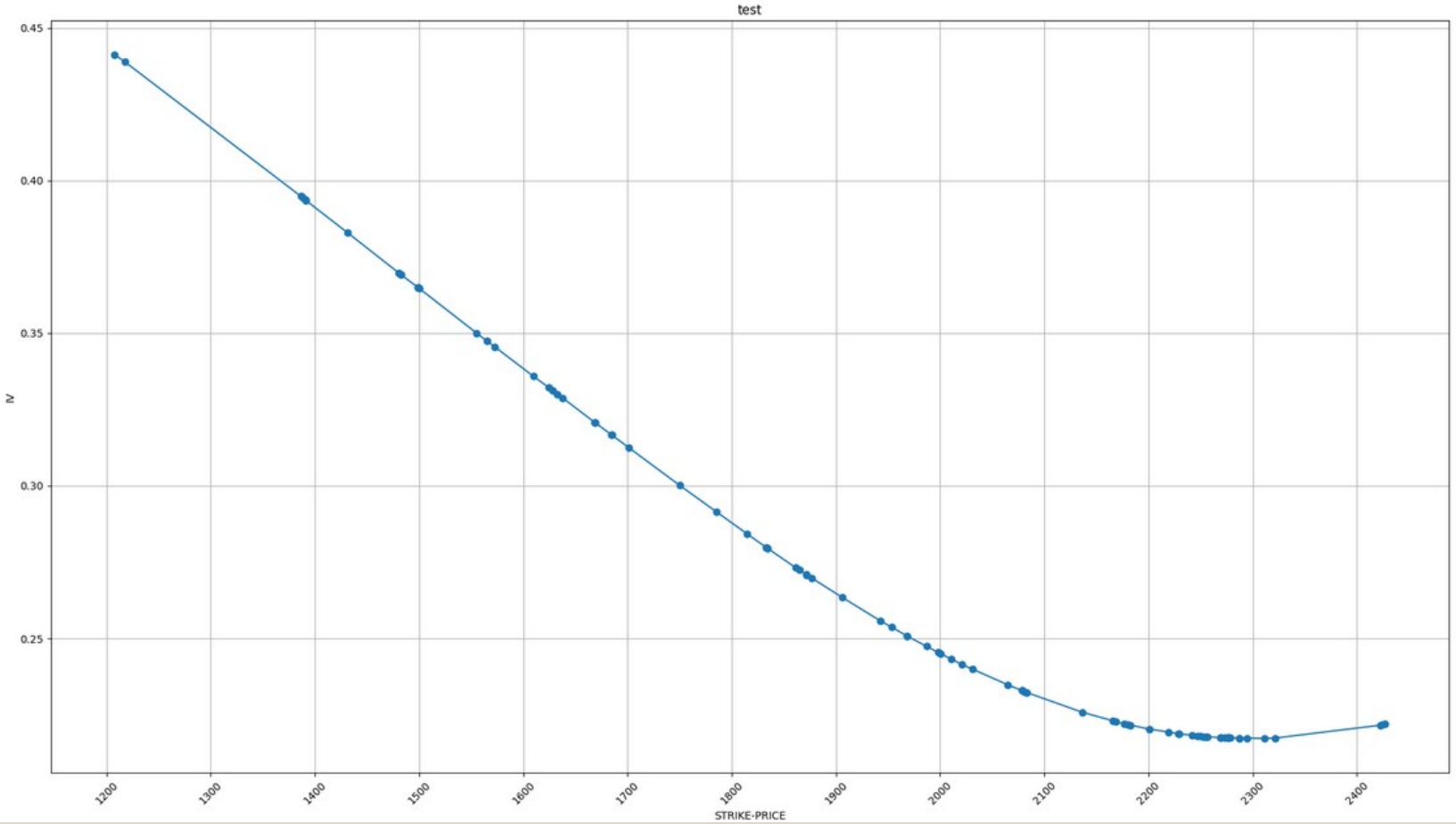

Implied Volatility
Strike

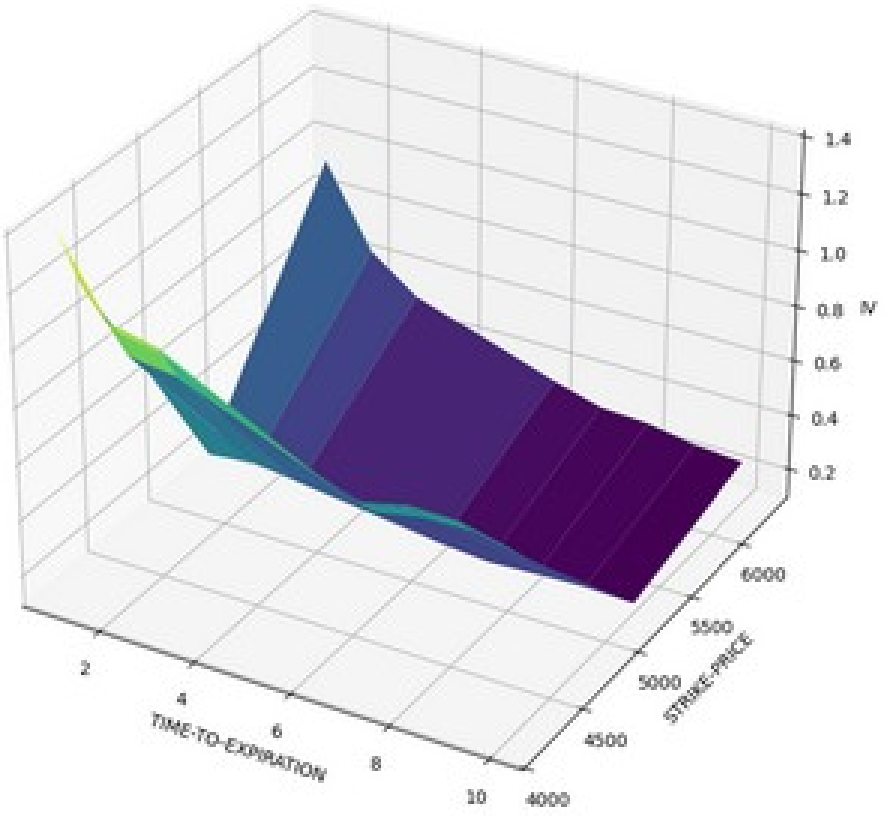
Volatility Smile



Investopedia

RUT 2022-12-16 137 C




1.4
1.2
1.0
0.8
0.6
0.4
0.2
IV
2
4
6
8
10
TIME-TO-EXPIRATION
4000
4500
5000
5500
6000
STRIKE-PRICE

Steps Beyond the Internship

- Using Time-Series models to predict future stock prices and implied volatility using the data
- Create a more organized structure in the code for future improvements
- Fix volatility surfaces
- Questions?