4. Options

Investigate the mechanics and use of S&P 500 options to hedge Pershing Square's portfolio.

For the purposes of this assignment, the put options prices as outlined in Exhibit 4 of Pershing Square's Pandemic Trade (A) will be used for any quantitative analysis. A literature review and coding examples have been provided as part of this analysis.

Key Assumptions And Limitations

- The evaluation date (i.e. the date in which Pershing Square must make a decision on their Hedging strategy) is February 21, 2020
- Limited by the data we were able to find.

This investigation has been divided into three sections, as listed in the table of contents below:

Table of Contents

In [ ]:
```
pip install scipy
```

```
Collecting scipy
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.3,>=1.23.5 in c:\git\finm3405\.conda\lib\s
ite-packages (from scipy) (2.1.1)
Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
   ---------------------------------------- 0.0/44.5 MB ? eta -:--:--
   - -------------------------------------- 1.3/44.5 MB 6.7 MB/s eta 0:00:07
   -- ------------------------------------- 2.9/44.5 MB 7.6 MB/s eta 0:00:06
   ---- ----------------------------------- 4.5/44.5 MB 7.3 MB/s eta 0:00:06
   ----- ---------------------------------- 5.8/44.5 MB 7.0 MB/s eta 0:00:06
   ------ --------------------------------- 7.6/44.5 MB 7.2 MB/s eta 0:00:06
   -------- ------------------------------- 9.4/44.5 MB 7.6 MB/s eta 0:00:05
   ---------- ----------------------------- 11.3/44.5 MB 7.7 MB/s eta 0:00:05
   ------------ --------------------------- 13.4/44.5 MB 8.1 MB/s eta 0:00:04
   ------------- -------------------------- 15.5/44.5 MB 8.2 MB/s eta 0:00:04
   --------------- ------------------------ 17.6/44.5 MB 8.5 MB/s eta 0:00:04
   ----------------- ---------------------- 19.4/44.5 MB 8.4 MB/s eta 0:00:03
   ------------------- -------------------- 21.2/44.5 MB 8.5 MB/s eta 0:00:03
   -------------------- ------------------- 23.3/44.5 MB 8.6 MB/s eta 0:00:03
   ---------------------- ----------------- 25.4/44.5 MB 8.7 MB/s eta 0:00:03
   ------------------------ --------------- 27.5/44.5 MB 8.8 MB/s eta 0:00:02
   -------------------------- ------------- 29.9/44.5 MB 8.9 MB/s eta 0:00:02
   ---------------------------- ----------- 32.0/44.5 MB 8.9 MB/s eta 0:00:02
   ------------------------------ --------- 34.1/44.5 MB 9.0 MB/s eta 0:00:02
   -------------------------------- ------- 36.2/44.5 MB 9.0 MB/s eta 0:00:01
   ---------------------------------- ----- 38.3/44.5 MB 9.0 MB/s eta 0:00:01
   ------------------------------------ --- 40.1/44.5 MB 9.1 MB/s eta 0:00:01
   ------------------------------------- -- 42.2/44.5 MB 9.1 MB/s eta 0:00:01
   ---------------------------------------  44.3/44.5 MB 9.2 MB/s eta 0:00:01
   ---------------------------------------- 44.5/44.5 MB 9.0 MB/s eta 0:00:00
Installing collected packages: scipy
Successfully installed scipy-1.14.1
Note: you may need to restart the kernel to use updated packages.
```

Libraries

In [ ]:
```python
import yfinance as yf
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
import matplotlib.ticker as ticker
import matplotlib.dates as mdates
```

4.1 Hedging Position and Key Details

To protect *Pershing Square's* portfolio from adverse market movements, Bill Ackman could utilise *purchasing* S&P 500 index put options as a hedging instrument.

Literature review

A put option gives the holder the right, but not the obligation, to sell the underlying asset (in this case, the S&P 500 index) at a predetermined price (the strike price) on or before the expiration date. This allows the holder to benefit if the price of the underlying asset falls below the strike price.

Put Option Payoff The payoff of a put option at expiration depends on the relationship between the price of the underlying asset (S_T) and the strike price (K). If the underlying asset price is below the strike price, the payoff is positive. Otherwise, the option expires worthless. The payoff of a put option is given by:

$$\text{Put Option Payoff} = \max(K - S_T, 0)$$

Where:

- (S_T) is the price of the S&P 500 index at expiration.
- K is the strike price of the put option.

Determining the Optimal Number of Contracts

To determine the optimal number of call option contracts required to hedge the portfolio, we can use the optimal hedging ratio. The goal is to align the portfolio's exposure to market movements with the performance of the hedging instrument (the S&P 500).

The optimal number of contracts h can be calculated using the following formula:

$$h = \frac{\beta V}{F} = \bar{\rho}\frac{\bar{\sigma}_A V}{\bar{\sigma}_K F}$$

Where:

- h is the number of futures contracts to purchase or sell.
- β is the beta of the portfolio relative to the S&P 500 (this measures the portfolio's sensitivity to market movements).
- V is the value of the portfolio being hedged.
- F is the notional value of one futures contract on the S&P 500 index, calculated as the index value multiplied by the contract multiplier.
- σ_{A_t} is the standard deviation (volatility) in $A_t - A$.
- σ_{K_t} is the standard deviation (volatility) in $K_t - K$.
- ρ is the correlation between $A_t - A$ and $K_t - K$.

Reference to Beta from Section 3.1

The portfolio's beta (β) was determined in Section 3.1 and is not copied as a constant into the code demonstration.

Contract Choice

In regards to the choice on options contract, the closest to 'At The Money' (ATM) option with a strike price of 3300 was chosen.

In [ ]:
```
# general key info
EVAL_DATE = '2020-02-21'
EVAL_DATE_PLUS_ONE = '2020-02-22'

THIRTY_DAY_SOFR = 0.00154
```

```
# Historic price excel doc
HISTORIC_PRICE_EXCEL = 'portfolio_values_eval_date.xlsx'
HISTORIC_PRICE_EXCEL_SHEET_NAME = 'Values'

# Contract info
EXPIRY_MAR = '2020-03-20'
EXPIRY_MAR_PLUS_ONE = '2020-03-21'

EXPIRY_APR = '2020-04-17'
EXPIRY_APR_PLUS_ONE = '2020-04-18'

P_MAR = 52.80  # Using Ask Price
P_APR = 67.90  # Using Ask Price

T_MAR = (datetime.strptime(EXPIRY_MAR, '%Y-%m-%d') - datetime.strptime(EVAL_DATE
T_APR = (datetime.strptime(EXPIRY_APR, '%Y-%m-%d') - datetime.strptime(EVAL_DATE

K = 3300 # At the money put option

OPT_CONTRACT_SIZE = 50

# hedging info from previous section
BETA_SPX_P = 0.592918
FIRM_VALUE = 7_621.28 * 1_000_000


### Price of the S&P 500 index on the 21st Feb:
spx_price = yf.download("^SPX", start=EVAL_DATE, end=EVAL_DATE_PLUS_ONE)["Adj Cl

# Hedge Pershing Square's portfolio using PUT options with key figures as above:

F = K * OPT_CONTRACT_SIZE

H = BETA_SPX_P * FIRM_VALUE / F
```

```
[*********************100%***********************]  1 of 1 completed
```

4.1.1 Option Costs

In [ ]:

```
# March expiry
cost_mar_expiry = H * P_MAR
cost_apr_expiry = H * P_APR

stats = f'''
############ KEY STATISTICS ############
Beta: {BETA_SPX_P:6f}
Optimal number of contracts to hedge firm position: {int(round(H, 0))}
(unrounded {H:2f})

Face value of futures contract on evaluation date ({EVAL_DATE}): ${F:2f}'


############ UPFRONT COSTS ############
Cost for put options, March ({EXPIRY_MAR}) expiry: ${cost_mar_expiry:.2f}
Cost for put options, April ({EXPIRY_APR}) expiry: ${cost_apr_expiry:.2f}
'''

print(stats)
```

```
############ KEY STATISTICS ############
Beta: 0.592918
Optimal number of contracts to hedge firm position: 27387
(unrounded 27386.630879)

Face value of futures contract on evaluation date (2020-02-21): $165000.000000'


############ UPFRONT COSTS ############
Cost for put options, March (2020-03-20) expiry: $1446014.11
Cost for put options, April (2020-04-17) expiry: $1859552.24
```

4.2 Timing in the Market

Following the discussion above, we assume the the optimal timing to exit the market is at the maximum payoff of exercising the put options contract. For European options, the holder can choose to exercise at maturity T and no time $t <= T$. It would be illogical to graphically illustrate a payoff diagram between the evaluation date T_0 and T since the holder (Pershing Square) would be unable to exercise it.

Thus, retrospectively determine the payoff of the at-the-money call options with the March and April maturities and determine the payoff by discounting at the risk free rate, i.e.

$$\text{Payoff} = e^{-rT}(K - S_T)$$

In [ ]:
```
spx_price_mar_exp = yf.download("^SPX", start=EXPIRY_MAR, end=EXPIRY_MAR_PLUS_ON
payoff_mar = max(0, K - spx_price_mar_exp) * (H * OPT_CONTRACT_SIZE) * np.exp(TH

spx_price_mar_apr = yf.download("^SPX", start=EXPIRY_APR, end=EXPIRY_APR_PLUS_ON
payoff_apr = max(0, K - spx_price_mar_apr) * (H * OPT_CONTRACT_SIZE) * np.exp(TH

print(f'Payoff of Options contract in March: ${payoff_mar:.4f}')
print(f'Payoff of Options contract in April: ${payoff_apr:.4f}')
```

```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Payoff of Options contract in March: $1362431340.9623
Payoff of Options contract in April: $582428791.0586
```

Payoffs

In this analysis, we estimated the payoffs for put options with varying exercise prices (K) for two expiration dates: March and April. The payoffs were calculated using the formula:

$$\text{Payoff} = \max(0, K - S)\left(\frac{\beta V}{KC}\right) Ce^{-\text{r}T}$$

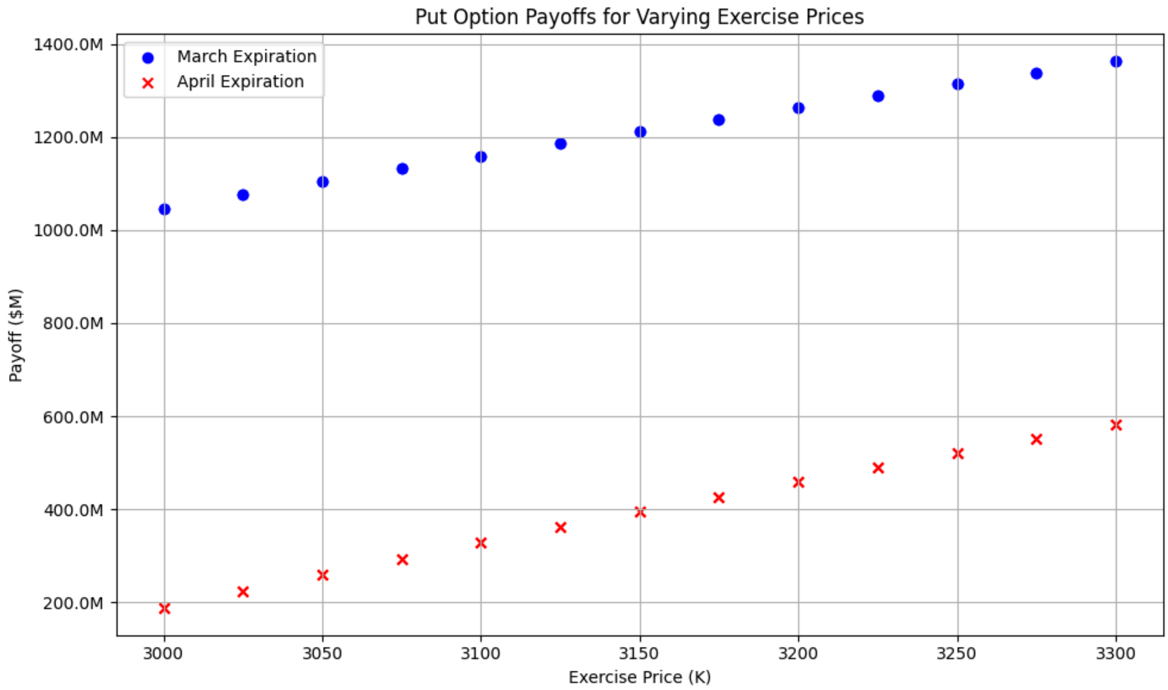
where S is the underlying asset price, β_{SPX} represents the sensitivity of the asset, V_{firm} is the firm's value, C is the options contract size, and T is the time to expiration in years. The payoffs for all exercise prices were visualized using a scatter plot.

In [ ]:
```
file_path = 'options_data/put_option_prices.xlsx'
put_prices_df = pd.read_excel(file_path, sheet_name='prices')

# Extract exercise prices (K) and put option prices from the dataframe
K = put_prices_df['Exercise Price'].values  # Modify the column name as necessar
put_prices = put_prices_df['Ask Price'].values  # Modify the column name as nece

# Calculate payoffs for varying exercise prices
payoffs_mar = np.maximum(0, K - spx_price_mar_exp) * (((BETA_SPX_P * FIRM_VALUE)
    (K * OPT_CONTRACT_SIZE)) * OPT_CONTRACT_SIZE) * np.exp(THIRTY_DAY_SOFR * T_A
payoffs_apr = np.maximum(0, K - spx_price_mar_apr) * (((BETA_SPX_P * FIRM_VALUE)
    (K * OPT_CONTRACT_SIZE)) * OPT_CONTRACT_SIZE) * np.exp(THIRTY_DAY_SOFR * T_A


plt.figure(figsize=(10, 6))
plt.scatter(K, payoffs_mar, label='March Expiration', color='blue', marker='o')
plt.scatter(K, payoffs_apr, label='April Expiration', color='red', marker='x')
plt.title('Put Option Payoffs for Varying Exercise Prices')
plt.xlabel('Exercise Price (K)')
plt.ylabel('Payoff ($M)')
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: f'{x*1e-
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```



3.3 Value of portfolio

Consider that for Pershing Square, the combined change in value of their holdings if they chose to hedge with options contracts would be:

1. The value profit (or loss) of their holdings from evaluation date (`Feb 21 2020`)
2. The value of the options contracts at time t
3. Premium paid on options contracts at time t_0

```
In [ ]: K = 3300

        df = pd.read_excel(HISTORIC_PRICE_EXCEL, sheet_name=HISTORIC_PRICE_EXCEL_SHEET_N
        df_profits = df[df.index >= EVAL_DATE].copy()

        firm_value_eval_date = df_profits['Firm Value'].loc[EVAL_DATE]

        df_profits = df_profits[['Firm Value', '^spx']]
        df_profits['Firm Profit'] = df_profits['Firm Value'] - firm_value_eval_date
        df_profits['Payoff March Option'] = np.where(
            df_profits.index < EXPIRY_MAR,
            np.maximum(0, K - df_profits['^spx']) * (((BETA_SPX_P * FIRM_VALUE) / (K * C
            0)
        df_profits['Payoff April Option'] = np.where(
            df_profits.index < EXPIRY_APR,
            np.maximum(0, K - df_profits['^spx']) * (((BETA_SPX_P * FIRM_VALUE) / (K * C
            0)

        df_profits['Payoff March Option'] = df_profits['Payoff March Option'].replace(0,
        df_profits['Payoff April Option'] = df_profits['Payoff April Option'].replace(0,


        df_profits['Value March'] = df_profits['Firm Value'] + df_profits['Payoff March
        df_profits['Value April'] = df_profits['Firm Value'] + df_profits['Payoff April
        df_profits.head(50)
```

Out[ ]:

Date	Firm Value	^spx	Firm Profit	Payoff March Option	Payoff April Option	Value M
2020-02-21	7.697406e+09	3337.750000	0.000000e+00	NaN	NaN	
2020-02-24	7.430339e+09	3225.889893	-2.670668e+08	1.014813e+08	1.014813e+08	7.530375
2020-02-25	7.146771e+09	3128.209961	-5.506346e+08	2.352375e+08	2.352375e+08	7.380563
2020-02-26	7.040026e+09	3116.389893	-6.573798e+08	2.514231e+08	2.514231e+08	7.290003
2020-02-27	6.685772e+09	2978.760010	-1.011634e+09	4.398841e+08	4.398841e+08	7.124210
2020-02-28	6.618340e+09	2954.219971	-1.079066e+09	4.734875e+08	4.734875e+08	7.090382
2020-03-02	6.814513e+09	3090.229980	-8.828933e+08	2.872447e+08	2.872447e+08	7.100311
2020-03-03	6.596649e+09	3003.370117	-1.100757e+09	4.061847e+08	4.061847e+08	7.001387
2020-03-04	6.845999e+09	3130.120117	-8.514073e+08	2.326219e+08	2.326219e+08	7.077175
2020-03-05	6.496462e+09	3023.939941	-1.200944e+09	3.780177e+08	3.780177e+08	6.873034
2020-03-06	6.396099e+09	2972.370117	-1.301307e+09	4.486339e+08	4.486339e+08	6.843287
2020-03-09	5.830281e+09	2746.560059	-1.867125e+09	7.578428e+08	7.578428e+08	6.586678
2020-03-10	6.124975e+09	2882.229980	-1.572431e+09	5.720657e+08	5.720657e+08	6.695594
2020-03-11	5.680923e+09	2741.379883	-2.016483e+09	7.649361e+08	7.649361e+08	6.444413
2020-03-12	5.148352e+09	2480.639893	-2.549054e+09	1.121976e+09	1.121976e+09	6.268882
2020-03-13	5.542638e+09	2711.020020	-2.154768e+09	8.065089e+08	8.065089e+08	6.347701
2020-03-16	4.755585e+09	2386.129883	-2.941821e+09	1.251391e+09	1.251391e+09	6.005530
2020-03-17	4.663500e+09	2529.189941	-3.033906e+09	1.055495e+09	1.055495e+09	5.717549
2020-03-18	4.176677e+09	2398.100098	-3.520729e+09	1.235000e+09	1.235000e+09	5.410231
2020-03-19	4.328365e+09	2409.389893	-3.369041e+09	1.219541e+09	1.219541e+09	5.546460

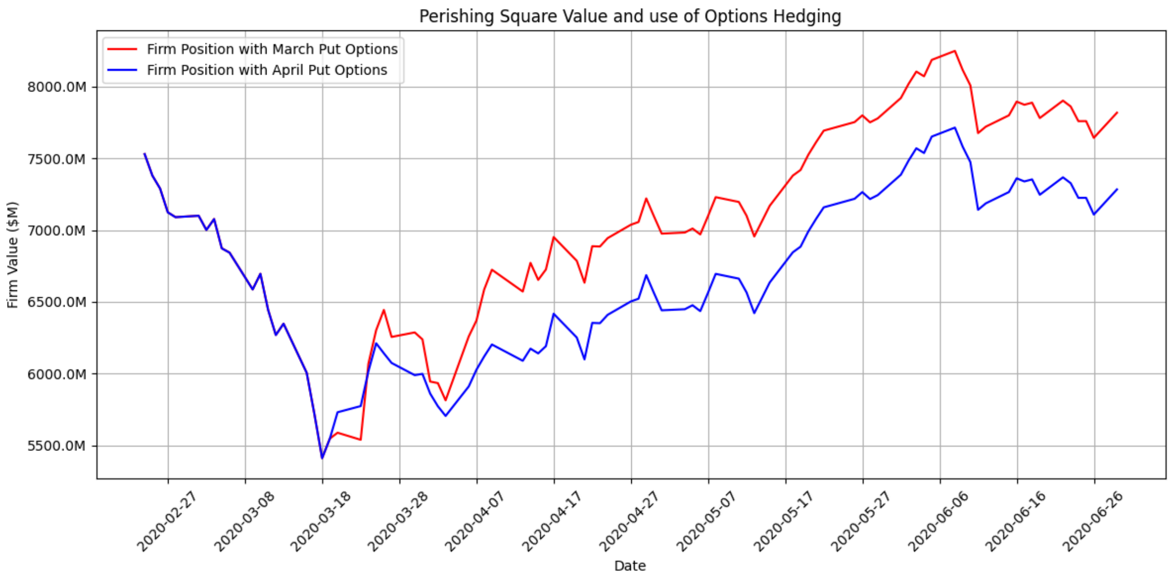
Date	Firm Value	^spx	Firm Profit	Payoff March Option	Payoff April Option	Value M
2020-03-20	4.369312e+09	2304.919922	-3.328094e+09	1.219541e+09	1.362595e+09	5.587407
2020-03-23	4.320024e+09	2237.399902	-3.377382e+09	1.219541e+09	1.455052e+09	5.538118
2020-03-24	4.850671e+09	2447.330078	-2.846735e+09	1.219541e+09	1.167588e+09	6.068765
2020-03-25	5.084053e+09	2475.560059	-2.613353e+09	1.219541e+09	1.128932e+09	6.302147
2020-03-26	5.225480e+09	2630.070068	-2.471926e+09	1.219541e+09	9.173562e+08	6.443575
2020-03-27	5.037013e+09	2541.469971	-2.660393e+09	1.219541e+09	1.038679e+09	6.255108
2020-03-30	5.068500e+09	2626.649902	-2.628906e+09	1.219541e+09	9.220395e+08	6.286594
2020-03-31	5.019991e+09	2584.590088	-2.677415e+09	1.219541e+09	9.796334e+08	6.238085
2020-04-01	4.726205e+09	2470.500000	-2.971201e+09	1.219541e+09	1.135861e+09	5.944300
2020-04-02	4.715196e+09	2526.899902	-2.982210e+09	1.219541e+09	1.058630e+09	5.933291
2020-04-03	4.595193e+09	2488.649902	-3.102214e+09	1.219541e+09	1.111007e+09	5.813287
2020-04-06	5.040590e+09	2663.679932	-2.656817e+09	1.219541e+09	8.713331e+08	6.258684
2020-04-07	5.151624e+09	2659.409912	-2.545782e+09	1.219541e+09	8.771802e+08	6.369718
2020-04-08	5.367697e+09	2749.979980	-2.329710e+09	1.219541e+09	7.531598e+08	6.585791
2020-04-09	5.505769e+09	2789.820068	-2.191637e+09	1.219541e+09	6.986055e+08	6.723864
2020-04-13	5.353649e+09	2761.629883	-2.343757e+09	1.219541e+09	7.372072e+08	6.571743
2020-04-14	5.553753e+09	2846.060059	-2.143653e+09	1.219541e+09	6.215943e+08	6.771847
2020-04-15	5.434965e+09	2783.360107	-2.262441e+09	1.219541e+09	7.074513e+08	6.653060
2020-04-16	5.507518e+09	2799.550049	-2.189888e+09	1.219541e+09	6.852819e+08	6.725613
2020-04-17	5.733559e+09	2874.560059	-1.963847e+09	1.219541e+09	6.852819e+08	6.951653

Date	Firm Value	^spx	Firm Profit	Payoff March Option	Payoff April Option	Value M
2020-04-20	5.567551e+09	2823.159912	-2.129855e+09	1.219541e+09	6.852819e+08	6.785646
2020-04-21	5.415306e+09	2736.560059	-2.282100e+09	1.219541e+09	6.852819e+08	6.633400
2020-04-22	5.669577e+09	2799.310059	-2.027829e+09	1.219541e+09	6.852819e+08	6.887671
2020-04-23	5.667454e+09	2797.800049	-2.029952e+09	1.219541e+09	6.852819e+08	6.885548
2020-04-24	5.725897e+09	2836.739990	-1.971509e+09	1.219541e+09	6.852819e+08	6.943991
2020-04-27	5.819057e+09	2878.479980	-1.878349e+09	1.219541e+09	6.852819e+08	7.037151
2020-04-28	5.838388e+09	2863.389893	-1.859018e+09	1.219541e+09	6.852819e+08	7.056482
2020-04-29	6.002689e+09	2939.510010	-1.694717e+09	1.219541e+09	6.852819e+08	7.220784
2020-04-30	5.877879e+09	2912.429932	-1.819527e+09	1.219541e+09	6.852819e+08	7.095974
2020-05-01	5.757338e+09	2830.709961	-1.940068e+09	1.219541e+09	6.852819e+08	6.975433

In [ ]:

```
plt.figure(figsize=(12, 6))
plt.plot(df_profits.index, df_profits['Value March'], linestyle='-', color='r',
plt.plot(df_profits.index, df_profits['Value April'], linestyle='-', color='b',

plt.title('Perishing Square Value and use of Options Hedging')
plt.xlabel('Date')
plt.ylabel('Firm Value ($M)')
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: f'{x*1e-
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```



4.4 Sensitivity analysis

Changes in implied volatility

- The implied volatility of the options contract was determined using Newton's method. By iteratively solving the Black-Scholes model, the volatility is adjusted until the model's theoretical option price matched the observed market price. Newton's method efficiently converged to the implied volatility by minimizing the difference between the model and market prices.

However, performing a sensitivity analysis on implied volatility was not considered insightful for this analysis. Since the implied volatility is derived directly from market conditions and is used as an input in the option pricing model, further sensitivity analysis on this parameter would not provide meaningful or new information in this context.

The following code (derived from lecture materials) is a implementation of Newtons method to calculate the implied volatility of the March expiry put options contract.

In [ ]:
```
def black_scholes (S, K, r, T, sigma , q):
    d1 = (np.log(S / K) + (r - q + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    C = S * np.exp(-q * T) * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2) #
    P = -S * np.exp(-q * T) * norm.cdf(-d1) + K * np.exp(-r * T) * norm.cdf(-d2)
    return [C, P]

# function to calculate option vega
def vega(S, K, r, T, sigma , q):
    d1 = (np.log(S / K) + (r - q + 0.5 * sigma ** 2) * T) / ( sigma * np.sqrt(T)
    return np.exp(-q * T) * S * norm.pdf(d1) * np.sqrt(T) # same for calls and p

# observed call or put price
obs = P_MAR # put price

# known / observed / given parameter values
S = spx_price
K = K
r = THIRTY_DAY_SOFR
T = T_MAR;
```

```python
q = 0

# Newton ’s method
sigma = np.sqrt (2 * np.abs(np.log(S / (K * np.exp(-r * T))))/T) # initial guess
val = black_scholes(S, K, r, T, sigma, q)[1]

while (abs(val -obs) > 10 ** -8):
    v = vega(S, K, r, T, sigma , q)
    sigma = sigma - (val - obs)/v # Newton step to update / improve estimate of
    val = black_scholes(S, K, r, T, sigma , q)[1]

print(f"Implied volatility: {sigma:.6f}")
```

```
Implied volatility: 0.190229
```

4.5 Acknowledgements and Tooling

This work is licensed under the MIT License and is freely available for distribution. All rights are reserved by the author, DanielCiccC.

- Various tools, including GitHub, GitHub Copilot, and ChatGPT, were utilized in the development and analysis of this project.
- Portions of the code were adapted from examples provided in lectures.

```
DEALINGS IN THE
SOFTWARE.
```