10/4/2024

Bill Ackman’s and Perishing Square Portfolio

FINM3405

Daniel Ciccotosto-Camp, (45857278)
Justin Coles, (46984779)
Matthew Comino, (47494529)
Tze Kheng Goh, (47037542)

Executive Summary:

This report provides a comprehensive analysis of Pershing Square Capital Management's hedging strategies during the COVID-19 pandemic, highlighting its portfolio composition, market exposure, and the critical decisions made to mitigate risks. Under the leadership of Bill Ackman, Pershing Square faced significant challenges due to its concentrated holdings in sectors such as hospitality and retail, which were severely impacted by the pandemic. The report examines the fund's market forecasts, the potential risks posed by the pandemic, and the various hedging strategies considered, including index futures, options, and credit default swaps (CDS).

Ackman's ultimate decision to implement a large-scale CDS strategy proved highly effective, yielding $2.65 billion in profits while safeguarding the fund's investments. The analysis delves into the quantitative aspects of this strategy, including contract sizes, positions, and market timing, and evaluates how the CDS trade met Pershing Square's hedging objectives. Additionally, the report discusses Pershing Square's partial liquidation of the CDS positions to capture gains as market conditions evolved.

The report concludes by demonstrating how the fund's proactive and strategic approach to hedging, combined with Ackman's deep market insights, enabled Pershing Square to navigate one of the most volatile periods in recent history. By effectively managing downside risk while maintaining its long-term investment philosophy, Pershing Square emerged stronger from the crisis, reinforcing the importance of adaptive and timely risk management in hedge fund operations.

Introduction:

Bill Ackman, founder and CEO of Pershing Square Capital Management, is widely recognised for his bold investment strategies, as well as his unique approach to increasing shareholder value (Pershing Square Foundation, 2024). Since being established in 2004, Pershing Square has built and maintained a reputation for being a hedge fund that acquires significant positions in companies, with the aim of influencing their operations to boost the company's value (Rothberg, 2016).

Ackman's investment philosophy is centred around value-oriented activist strategy investing that has a focus on long term investment horizons. Pershing Square's portfolio is composed of large positions in a relatively small number of companies (Hartman, 2024). This focus allows Ackman and his team the time to heavily engage with the management of the companies they invest in, aiming to take an active role in strategic decision-making and cash flow generation.

The fund's approach relies on comprehensive research and a willingness to take seemingly bold positions when market uncertainties are identified by Ackman and the research. In early 2020, Pershing Square's portfolio was heavily concentrated the hospitality, retail, and restaurant sectors. Key holdings of the fund included Hilton Worldwide, Restaurant Brands International, and Chipotle Mexican Grill (Stockcircle, 2020).

However, the outbreak of COVID-19 developed unprecedented challenges for companies within their existing portfolio. With the rapid rate of increase of Covid-19 cases, federal governments were forced to implement strict lockdown measures and public health mandates (Gupta, 2020). These global lockdowns halted economic activity, significantly impacted the sectors in which Perishing Portfolio held investments in. For example, Hilton Worldwide were hit with plummeting occupancy rates. Similarly, Restaurant Brands International and Chipotle Mexican Grill faced operational restrictions (Alvarez, 2023).

The pandemic's impact on consumer behaviour placed immense pressure on company revenues (Chan, 2021). This exposed Pershing Square to substantial risks, requiring an evaluation of hedging strategies to protect the fund's assets. As the pandemic unfolded, the investment decisions made would be critical in shaping Pershing Square's short-term performance and consequential long-term success.

Pershing's portfolio, exposure, forecasts, and objectives:

Pershing Square Capital Management has consistently adhered to a value-oriented, activist investment approach. The firm's concentrated portfolio typically consists of a small number of significant positions in large companies, allowing Ackman to interact with company management and influence strategic decisions (Hartman, 2024). This focus on active engagement with fundamentally strong, undervalued companies is key to Pershing Square's long-term investment strategy. However, this concentration also exposed the fund to substantial risk amongst certain market sectors, particularly as the COVID-19 pandemic began to affect global markets in 2020.

	Firm Name	Ticker Code	Number of Shares	Value (in $ million)	Share price	% Ownership	% of Fund Value	Board Seat	Inception Date
0	Chipotle Mexican Grill	CMG	8.619146e+07	1494.56	17.340000	0.0620	0.213	Yes	2016-08-01
1	Hilton Worldwide	HLT	1.055680e+07	1138.02	107.799661	0.0374	0.162	No	2018-10-01
2	Lowe's Companies	LOW	8.613212e+06	1001.20	116.240028	0.0112	0.143	No	2018-04-01
3	Restaurant Brands International	QSR	1.546818e+07	943.71	61.009759	0.0334	0.134	No	2012-06-01
4	Berkshire Hathaway (B)	BRK-B	4.015594e+06	901.22	224.430059	0.0029	0.128	No	2019-05-01

Table 1. Pershing's portfolio breakdown

As of early 2020, Pershing Square's portfolio included significant investments in sectors such as hospitality, retail, and restaurants; industries particularly vulnerable to the economic downturn caused by the pandemic. Some of the key holdings included Hilton Worldwide, Restaurant Brands International, and Chipotle Mexican Grill, all chosen for their strong brands, business models, and growth potential (Stockcircle, 2020). Under normal market conditions, these companies would align well with Pershing Square's value-investing philosophy.

However, the pandemic's impact on consumer behaviour and economic activity placed these businesses under severe financial pressure. The global lockdowns resulted in severe reductions in occupancy rates for Hilton and restrictions on restaurant operations for companies like Restaurant Brands International and Chipotle Mexican Grill (Alvarez, 2023). As the pandemic ensued, Ackman and his team had to consider adjusting their usual investment approach. The severe reduction in economic activity in the hospitality and restaurant sectors left Pershing Square's portfolio vulnerable to substantial losses. Figure 1 illustrates the impact of the pandemic on the overall value of Pershing Square's holdings during this period.

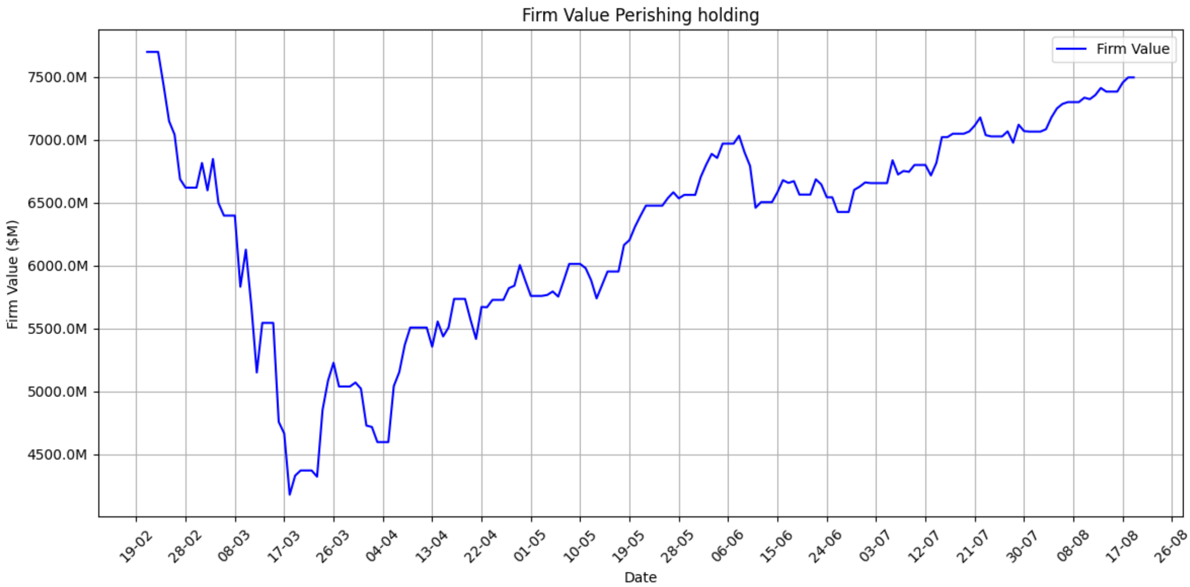


Figure 1. Pershing's firm value (21/02/20 - 21/08/20)

Ackman's team at Pershing Square concluded that COVID-19 was not merely a temporary disruption, but rather a major long-term challenge with profound implications for the global economy. Based on their analysis, Pershing Square anticipated a sustained period of economic uncertainty, with significant downside risks to the market, particularly the industries in which they held major positions. This led the team to re-evaluate the portfolio's exposure and consider the need for an effective hedging strategy to mitigate potential losses.

The primary objective for Pershing Square was to protect the portfolio from the severe market downturn that was expected to ensue following the pandemic. Given the concentrated nature of the fund's holdings, a market decline would disproportionately affect the portfolio, leading to a comprehensive hedging strategy needing to be formed. However, the challenge was to develop a hedging approach that would limit the downside risk without undermining the fund's long-term investment philosophy. Ackman and his team aimed to balance the need for protection with the goal of preserving the potential for future gains once the pandemic subsided.

In response to the global crisis, Pershing Square's hedging objectives centred around reducing exposure to market volatility while maintaining flexibility for future growth. The team explored several hedging options, including the use of S&P 500 index futures, index options, and credit default swaps (CDS). Each of these strategies was evaluated in terms of its ability to limit losses, manage ongoing cashflows, and adapt to market conditions as they evolved. Pershing Square's forecasts indicated that, while the short-term outlook was bleak, the long-term recovery potential for their core holdings remained intact. As such, their hedging strategy needed to protect against immediate downside risks without fully liquidating the portfolio and missing out on potential gains during the eventual recovery.

The decision to hedge was difficult, as Ackman's investment philosophy typically emphasises long-term value creation rather than short-term market timing. However, the unprecedented nature of the pandemic and its immediate impact on key sectors of Pershing Square's portfolio forced such a strategic shift. The fund's hedging strategy aimed to protect the portfolio from the worst of the downturn while positioning the fund to capitalise on opportunities once the market began to recover. This approach allowed Pershing Square to navigate the crisis while remaining aligned with its long-term investment goals.

Pershing Square's exposure to the hospitality and restaurant sectors left the fund highly vulnerable during the COVID-19 pandemic. The decision to hedge the portfolio was driven by the need to protect against short-term market declines while preserving the potential for future gains. This delicate balance between risk management and long-term growth remained central to Pershing Square's approach as the pandemic evolved.

Pershing's hedging options and deliberations:

To evaluate the hedging options, we assumed the positions were opened and assessed from the evaluation date of February 21, 2020. An optimal hedging strategy was applied to futures and options contracts, whilst for CDS and partial liquidation strategies, hedging positions were based on the notional value to Perishing Portfolio's value as of the evaluation date. Considerations of factors such as contract size, positioning, maturities, and market timing are outlined in the relevant Jupyter notebooks attached in Appendices 1–5. Refer to the appendix for a full analysis, including the literature review, graphs, and coding examples.

Partial Liquidation (Appendix 2)

One of the most straightforward options was to liquidate a portion or the entirety of the portfolio and convert the proceeds into cash. This strategy would have effectively removed Pershing Square's market exposure, mitigating the risk of further downside during the market collapse. The degree of liquidation was considered in various scenarios—partial (25%, 50%, 75%) and full liquidation.

The graph below demonstrates how each liquidation level would have affected the firm's portfolio value over time. As the market plummeted, even partial liquidation would have preserved capital, with full liquidation (red line) showing the least drawdown compared to the portfolio's original value. However, while liquidation offered downside protection, it would have also limited the potential for future gains once the market rebounded, which is evident as the portfolio values began recovering in the latter part of 2020.

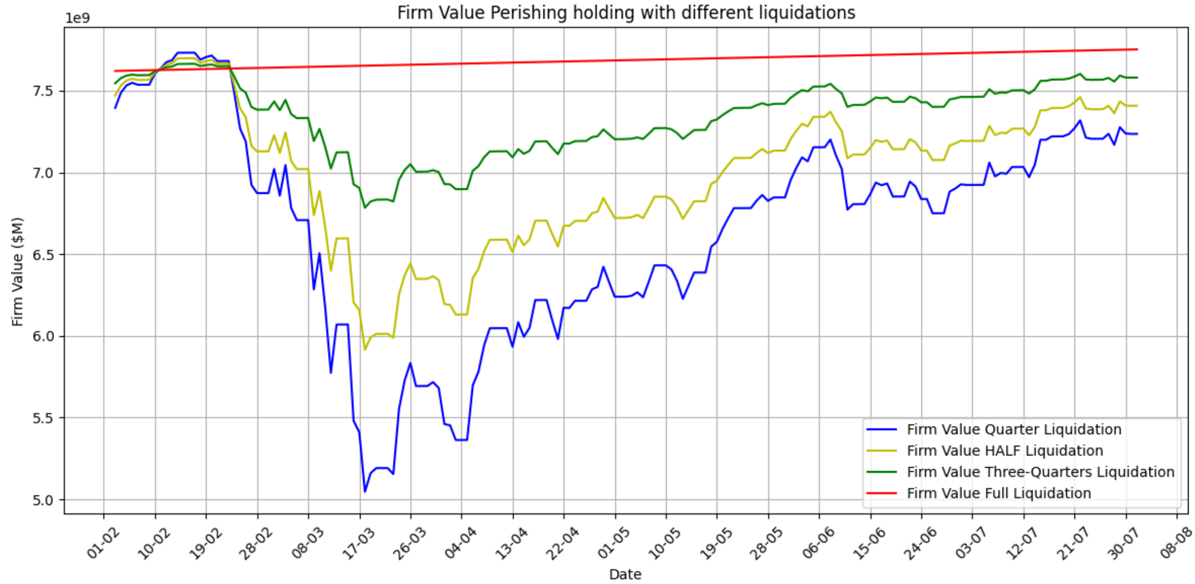


Figure 2. Pershing's firm value with different liquidation scenarios

Despite the safety of full liquidation, locking in losses and sacrificing potential future gains was not an ideal strategy. Liquidating a large volume of shares at once could result in significant additional losses, not only from selling assets at depressed prices but also due to the negative market impact of such a large-scale sale. These additional losses from large-scale liquidation were not incorporated into the figures presented above. Instead, the alternative hedging option involved shorting S&P 500 futures contracts, which would gain value as the market declined and help offset portfolio losses. This approach allowed for hedging without selling underlying assets, maintaining the long-term investment strategy.

Futures (Appendix 3)

The graph below illustrates the profitability of the futures hedging strategy compared to holding the portfolio unhedged. As shown, the futures hedge significantly reduced the firm's losses during the initial market decline in March 2020. The blue line, representing the profits with futures hedging, shows a marked improvement over the red line, which reflects the portfolio's performance without the hedge. This strategy allowed Pershing Square to mitigate short-term losses while maintaining exposure to a market recovery.

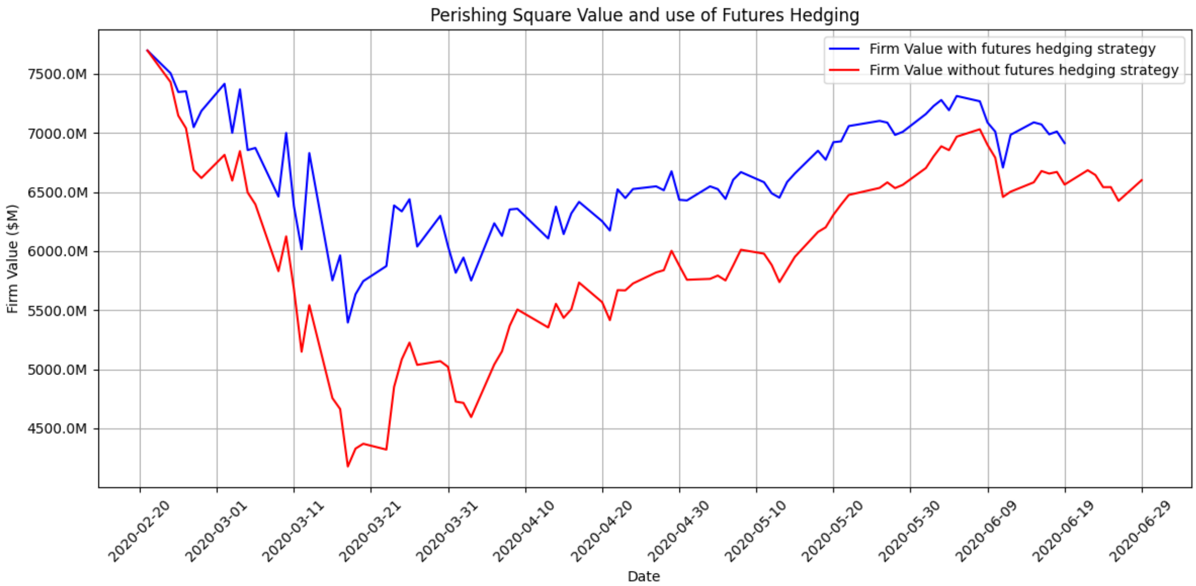


Figure 3. Pershing Square Value with different futures hedging strategies

One of the key considerations in using futures as a hedging instrument was the cost of carry, which is influenced by interest rates, dividends, and storage costs (Appendix 3.4.1). Figure 4 presents a sensitivity analysis of the cost of carry and its impact on future price forecasts at one week, one month, and two months after the evaluation date. Consider that the payoffs of a futures contract can be locked in by executing a reversing trade at a future point, providing flexibility in managing the hedge over time.

For Ackman's team, understanding these dynamics was crucial for determining both the timing and magnitude of the futures contracts necessary for effective hedging. Figure 4 outlines the impact of the cost of carry on the overall payoff, as well as how factors such as interest rates could influence the outcome. The contract price for the March-maturing futures contract was 3367.5, and any changes to dividends, interest rates, or storage costs that lowered the cost of carry (and by extension, the futures price at some later date) would result in a positive payoff. Analyses like these underscores that Ackman's team did not simply employ a futures strategy for its simplicity, but conducted an in-depth evaluation of how external factors could affect the hedge's performance.

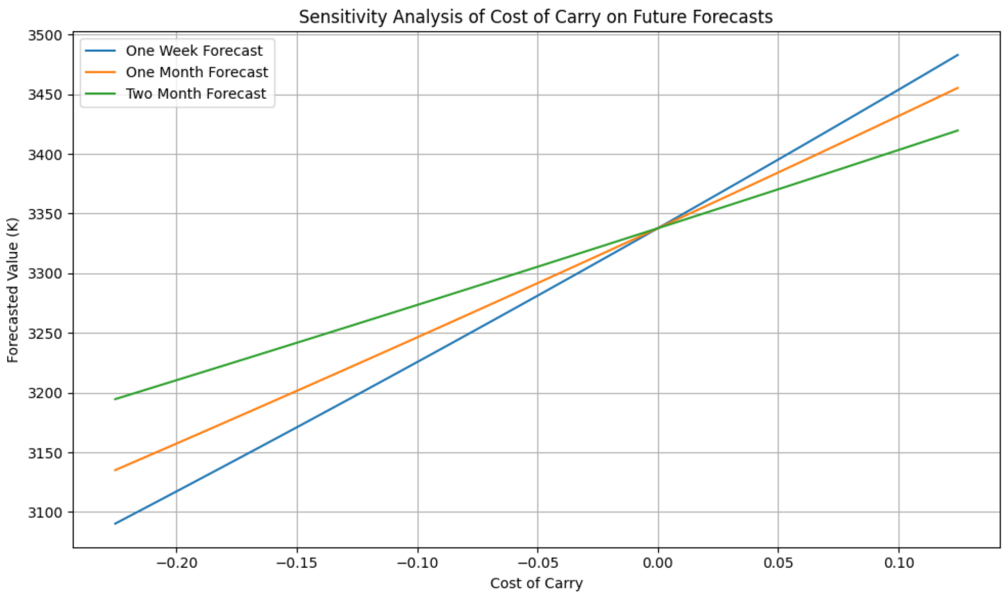


Figure 4. Sensitivity analysis of cost of carry with future forecasts

Put Options (Appendix 4)

The third option that Pershing Square evaluated was purchasing put options on the S&P 500 index. From Figure 5, it can be seen how the well the March put options are better than April put options. Furthermore, put options with March expirations generally offered higher payoffs compared to those expiring in April, particularly at lower strike prices (Figure 6). This is most likely due to the timing, as March captured significantly more of the initial drop in market value, when compared to April. However, the cost of purchasing these options was substantial, particularly for longer expiration dates, and Ackman had to consider whether the hedging benefits outweighed the high upfront premium costs.

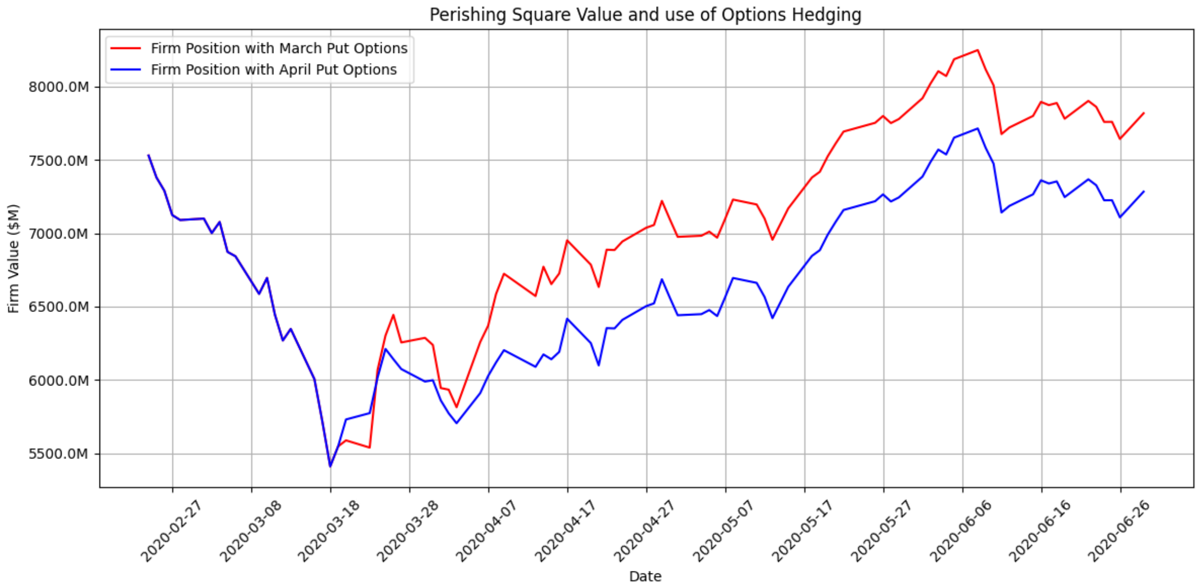


Figure 5. Firm Value When Hedging with March and April Put Options

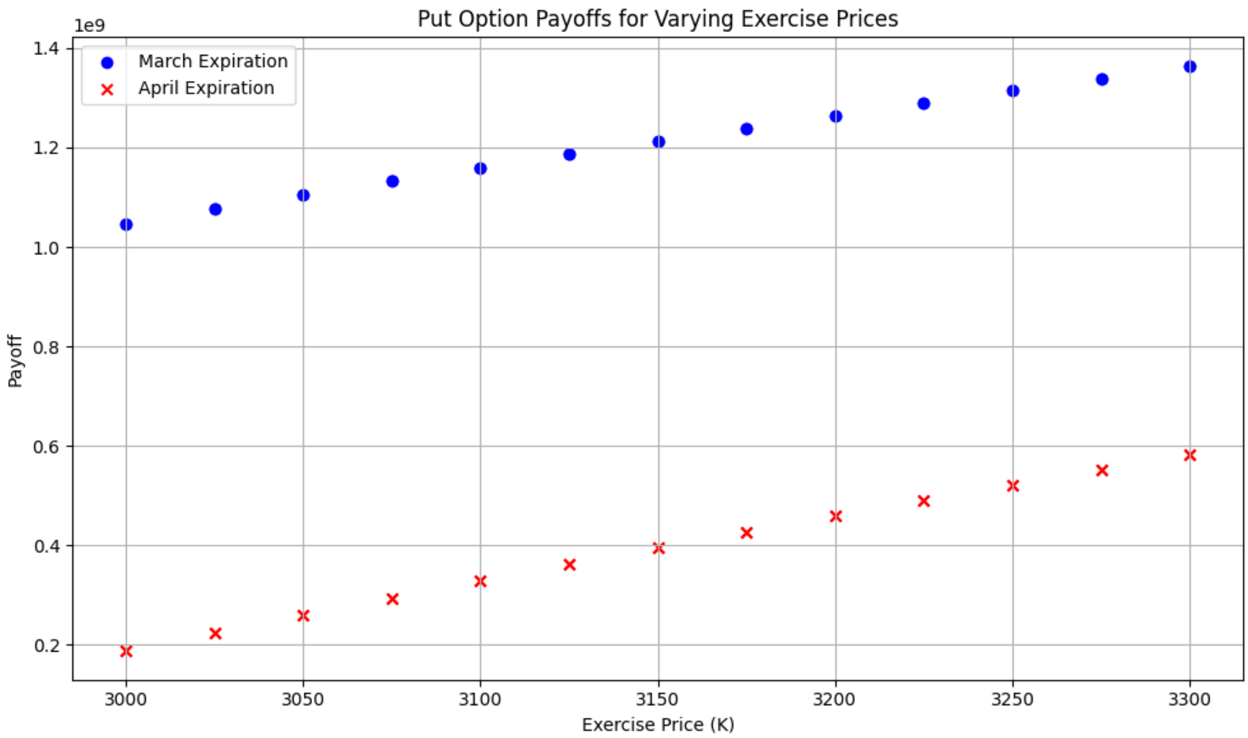


Figure 6. Put option payoffs for varying exercise prices

Credit Default Swaps (Appendix 5)

Credit Default Swaps (CDS) are contracts between two counterparties that insures a payout to the buyer in the event of a default. The mechanics of the CDS contracts, used for hedging, involves recurring premium payments from the buyer of protection to the seller. Additionally, an upfront payment would be paid to the seller of the contract. The CDS spreads function as a proxy for the market's perception of credit risk. As spreads widen, the value of the CDS contracts increases, offering substantial gains for the buyer.

	CDS	Payoffs	Spreads Before	Spreads After	CDS Spread	Quarterly Premiums	Upfront Payment $100 Notional value	Upfront Payment
0	CDX IG	$263,337,103.64	0.59%	1.30%	0.71%	$22,707,347.98	$2.55	$196,283,855.46
1	CDX HY	$1,235,088,105.78	3.46%	6.79%	3.33%	$133,165,125.47	$8.97	$690,457,326.80
2	ITRAXX Main	$248,501,210.47	0.60%	1.27%	0.67%	$23,092,218.29	NaN	NaN

Table 2. Payoffs When Hedging With CDS Indices

Ackman's hedge yielded considerable profits as credit spreads surged in response to the escalating pandemic crisis. Initially, spreads on both IG and ITRAXX indices widened moderately, with payoffs of $263 million and $248 million respectively. However, as the crisis deepened and panic spread through financial markets, CDX HY spreads surged by 3.33%. This significant widening resulted in payoffs of more than $1.23 billion for Pershing's CDS positions These numbers are evident in Table 2. The exponential increase in payoffs illustrates the effectiveness of this hedge as the portfolio's value was preserved amidst global market turbulence.

Analysis of feasibility and suitability of hedging options

During the COVID-19 pandemic, Pershing Square evaluated several hedging options to reduce the market risk of their portfolio. Not hedging was quickly deemed unsuitable, as the portfolio faced significant losses due to its exposure to vulnerable sectors, evident in Figure 1. The steep decline, between February and March, in Perishing's portfolio value highlights the infeasibility of this approach.

Partial liquidation mitigates some losses while preserving capital for potential reinvestment. Full liquidation, although feasible in the short term, would have eliminated market exposure altogether. The liquidation approaches would have allowed Pershing Square to avoid further losses and potentially reinvest at a lower market point. Though, selling such large options would significantly impact share prices, making these strategies less suitable as they don't align with Perishing's long-term strategy, and this was not incorporated into the models presented in Figure 2 or Appendix 2 below.

S&P 500 futures, as shown in Figure 3, provided only moderate protection. While futures contracts lessened the portfolio's exposure to market downturns, they were insufficient to prevent substantial losses, as the portfolio value still declined by approximately $2 billion. S&P 500 put options, in contrast, were a more effective hedge. Figure 6, depicting the payoff of put options demonstrates how this strategy provided downside protection while maintaining the possibility of capitalising on future market gains.

The use of credit default swaps (CDS) proved to be an effective hedging strategy, with the CDX HY yielding payouts of approximately $1.2 billion. While the options strategy allowed Pershing Square's portfolio to recover to pre-crisis levels in less than 180 days—outperforming the returns from the CDS hedging position as detailed in Appendix 5—there were important considerations that influenced the decision to prioritise CDS. The notional value of the portfolio was not a reliable proxy for hedging with CDS, which is what Ackman did not employ. These factors, including the high premiums associated with options, which significantly reduced potential profits, likely contributed to Pershing Square's decision to favor CDS as a more cost-effective hedging strategy.

Pershing Square’s CDS trade

To hedge against the severe economic impact of COVID-19, Pershing Square took a decisive position in Credit Default Swaps (CDS). Ackman acknowledged the mounting risks that the pandemic posed to global markets and executed a hedging strategy designed to mitigate the fund’s exposure to corporate defaults. By using CDS contracts on broad credit indices; CDX Investment Grade (IG), CDX High Yield (HY), and European iTraxx (ITRAXX), Pershing Square was effectively insuring its portfolio against a significant downturn in corporate creditworthiness.

The primary objective of the CDS trades was to protect the fund from losses due to defaults, downgrades, or widening credit spreads in the market. The notional value of the CDS contracts was linked to the portfolio value of Pershing Square as of February 21, 2020, which stood at approximately $7.7 billion. Ackman initiated the CDS contracts at this date, anticipating a significant rise in CDS spreads as the pandemic's economic impact grew.

Ackman and his team deliberated extensively on the timing of when to exit these positions. Initially, they kept the positions open as credit spreads continued to rise. The fund partially closed some of its CDS trades to lock in profits as spreads spiked, generating immediate cash inflows. By doing so, Pershing could ensure that it reaped the benefits of the widening spreads while maintaining some exposure to further market dislocations. However, as markets began to stabilise and spreads tightened, the team decided to close the remaining positions, timing their exit carefully to maximise gains while avoiding the risk of spreads reversing and eroding their profits.

Pershing’s decision to implement CDS as a hedge was highly effective, shielding the fund from catastrophic losses and resulting in significant profits. The deliberations around keeping the positions open initially, then gradually reducing exposure, highlight Ackman’s strategic foresight in navigating an unprecedented crisis. Through these trades, Pershing Square not only protected its portfolio but also positioned itself to benefit from the market's severe downturn, ultimately reinforcing its long-term performance and resilience.

Conclusion:

Pershing Square’s response to the unprecedented market turmoil caused by the COVID-19 pandemic demonstrated a well-executed and strategic approach to risk management and hedging. Led by Bill Ackman, the fund was able to navigate one of the most volatile periods in financial history through a series of calculated decisions. Initially, Pershing’s concentrated portfolio, with heavy exposure to sectors such as hospitality and retail, posed significant risks as the pandemic severely impacted these industries. Faced with the potential for substantial losses, Ackman and his team moved quickly to evaluate various hedging options, each with distinct risks, costs, and potential outcomes.

Among the options considered, Ackman’s bold decision to employ Credit Default Swaps (CDS) proved to be the most effective in protecting the fund from market volatility. By purchasing CDS contracts on major investment-grade and high-yield indices, Pershing Square was able to hedge against the widening credit spreads that materialised as the global economic crisis unfolded. The strategy not only safeguarded the portfolio but also resulted in significant profits as spreads widened dramatically. Through a mix of partial and full liquidation of their CDS positions, the fund secured gains of $2.65 billion, effectively mitigating the downside risk from their equity holdings.

Ackman's decision to carefully manage the timing of both the initial trades and the eventual liquidation of the positions highlights Pershing Square's agility and deep market insight. The success of this hedging strategy underscores the importance of proactive risk management in uncertain times. Ultimately, Pershing Square's response to the crisis allowed the fund to preserve capital, generate substantial returns, and maintain its long-term investment outlook, emerging stronger from one of the most challenging periods in financial markets.

Reference List:

Alvarez, E. (2023). *How did COVID-19 affect tourism occupancy and prices? A spatiotemporal and economic analysis of Madrid and Valencia through Airbnb geospatial data.* International Journal of Applied Earth Observation and Geoinformation. Retrieved from https://www.sciencedirect.com/science/article/pii/S1569843223002728?via%3Dihub

Chan, S. (2024). *An early view of post-COVID-19 discretionary spending in Asia.* McKinsey & Company. Retrieved from https://www.mckinsey.com/industries/retail/our-insights/an-early-view-of-post-covid-19-discretionary-spending-in-asia

Gupta, A. (2020). *Economic Impacts of the COVID-19 Lockdown in a Remittance-Dependent Region.* American Journal of Agricultural Economics. Retrieved from https://onlinelibrary.wiley.com/doi/10.1111/ajae.12178

Hartman, A. (2024). *The Investment Strategies of Bill Ackman's Pershing Square Holdings.* Invest Wizardry. Retrieved from https://investwizardry.com/bill-ackman-pershing-square-holdings/

Pershing Square Foundation. (2024). *Bill Ackman*. The Pershing Square Foundation. Retrieved from https://pershingsquarefoundation.org/people/bill-ackman/

Rothberg, H. (2016). *The Outside Looking in: Do Activist Investor Knowledge Management Actions pay off.* Marist College, USA. Retrieved from https://www.proquest.com/docview/1860279518/fulltextPDF/2F9B9033F1F34D22PQ/1?accountid=14723&sourcetype=Conference%20Papers%20&%20Proceedings

Stockcircle. (2024). *Bill Ackman Portfolio.* Warren Buffett Portfolio. Retrieved from https://stockcircle.com/portfolio/bill-ackman

Appendices:

For a better viewing experience, please visit the following website and click on the links at the bottom of the page: https://danielciccc.github.io/FINM3405/docs/FINM3405.html

This page was intentionally left blank

Pershing Square Hedging Analysis

This section provides a comprehensive quantitative analysis of Pershing Square's hedging strategies, conducted through a series of Jupyter notebooks. It evaluates the fund's various hedging options, which were previously discussed in the context of the report. Key areas of analysis include the examination of contract size, positioning, maturities, market timing, cashflows, and associated costs, as well as expected outcomes and payoffs as specified in the task brief. The analysis also incorporates sensitivity tests for different market scenarios in relation to the chosen hedging contracts. Additionally, a literature review explores the mechanics of each hedging option, with relevant coding examples provided. Assumptions for each hedging strategy are clearly outlined throughout the analysis.

Scope of the Research:

1. **No Hedging**: An analysis of the risks and rewards of not engaging in any hedging activities.
2. **Partial or Full Portfolio Liquidation into Cash**: The implications of converting assets into cash as a risk mitigation strategy.
3. **S&P 500 Index Futures**: The potential of futures contracts on the S&P 500 index as a hedging tool.
4. **S&P 500 Index Options**: An exploration of the strategic use of options contracts on the S&P 500 index.
5. **Index Credit Default Swaps (CDS)**: A detailed evaluation of using CDS contracts to hedge against market downturns.

Each of these options is evaluated not only for its immediate financial impact but also through the lens of potential future market scenarios.

Main Assumptions and Limitations maintained over Hedging concepts:

1. Share Price

One observed limitation includes discrepancies between the share price of Pershing Square's portfolio and the price listed on Yahoo Finance. For example, the share price of Chipotle Mexican Grill (Ticker: CMG) was derived by dividing the value of their stake by the number of shares in the portfolio:

$$\text{Share price (CMG)} = \frac{\text{Value}}{\text{Number of Shares}} = \frac{\$1{,}494{,}560{,}000}{1{,}724{,}310} = \$866.76$$

However, Yahoo Finance reported an *Adjusted Close* share price of $17.34. This analysis assumes that the reported number of shares in the case study was an error, but the valuation of the stake itself is accurate. Consequently, the number of shares has been adjusted in the accompanying Excel file (`Pershing_portfolio.xlsx`), which was imported for this analysis.

2. Evaluation Date
For the purposes of this assignment, we will assume that the evaluation date (i.e. the date in which Pershing Square must make a decision on their Hedging strategy) is February 21, 2020.

3. Hedging amount

- Sections **1** and **2**, only the notional value of the portfolio was considered.
- Sections **3** and **4** utilised optimal hedging techniques to determine the number of contracts to hedge.
- Section **5** used the notional value of Pershing Square's portfolio to hedge with CDS. Further analysis was performed on the CDS positions Pershing Square wrote later in the report.

1. No Hedging

The analysis of the no-hedging strategy disregards several key considerations, such as contract size, number of contracts, position, and ongoing cashflows since no hedging instruments are employed. Instead, the focus is on the valuation of Pershing Square’s portfolio over a six-month (180-day) period and the projected value of their holdings.

Table of Contents

Tooling

In [ ]:
```
pip install yfinance openpyxl pandas matplotlib
```

```
Requirement already satisfied: yfinance in c:\git\finm3405\.conda\lib\site-packag
es (0.2.43)
Requirement already satisfied: openpyxl in c:\git\finm3405\.conda\lib\site-packag
es (3.1.5)
Requirement already satisfied: pandas in c:\git\finm3405\.conda\lib\site-packages
(2.2.3)
Requirement already satisfied: matplotlib in c:\git\finm3405\.conda\lib\site-pack
ages (3.9.2)
Requirement already satisfied: numpy>=1.16.5 in c:\git\finm3405\.conda\lib\site-p
ackages (from yfinance) (2.1.1)
Requirement already satisfied: requests>=2.31 in c:\git\finm3405\.conda\lib\site-
packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in c:\git\finm3405\.conda\lib
\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\git\finm3405\.conda\lib\site-pac
kages (from yfinance) (5.3.0)
Requirement already satisfied: platformdirs>=2.0.0 in c:\git\finm3405\.conda\lib
\site-packages (from yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in c:\git\finm3405\.conda\lib\site-pa
ckages (from yfinance) (2024.2)
Requirement already satisfied: frozendict>=2.3.4 in c:\git\finm3405\.conda\lib\si
te-packages (from yfinance) (2.4.4)
Requirement already satisfied: peewee>=3.16.2 in c:\git\finm3405\.conda\lib\site-
packages (from yfinance) (3.17.6)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\git\finm3405\.conda\l
ib\site-packages (from yfinance) (4.12.3)
Requirement already satisfied: html5lib>=1.1 in c:\git\finm3405\.conda\lib\site-p
ackages (from yfinance) (1.1)
Requirement already satisfied: et-xmlfile in c:\git\finm3405\.conda\lib\site-pack
ages (from openpyxl) (1.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\git\finm3405\.conda\l
ib\site-packages (from pandas) (2.9.0)
Requirement already satisfied: tzdata>=2022.7 in c:\git\finm3405\.conda\lib\site-
packages (from pandas) (2024.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\git\finm3405\.conda\lib\sit
e-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\git\finm3405\.conda\lib\site-pa
ckages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\git\finm3405\.conda\lib\si
te-packages (from matplotlib) (4.54.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\git\finm3405\.conda\lib\si
te-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\git\finm3405\.conda\lib\site
-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\git\finm3405\.conda\lib\site-packa
ges (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\git\finm3405\.conda\lib\sit
e-packages (from matplotlib) (3.1.4)
Requirement already satisfied: soupsieve>1.2 in c:\git\finm3405\.conda\lib\site-p
ackages (from beautifulsoup4>=4.11.1->yfinance) (2.6)
Requirement already satisfied: six>=1.9 in c:\git\finm3405\.conda\lib\site-packag
es (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\git\finm3405\.conda\lib\site-pa
ckages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\git\finm3405\.conda
\lib\site-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\git\finm3405\.conda\lib\site-pa
ckages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\git\finm3405\.conda\lib\s
ite-packages (from requests>=2.31->yfinance) (2.2.3)
```

```
Requirement already satisfied: certifi>=2017.4.17 in c:\git\finm3405\.conda\lib\s
ite-packages (from requests>=2.31->yfinance) (2024.8.30)
Note: you may need to restart the kernel to use updated packages.
```

Libraries

In [ ]:
```
import yfinance as yf
import pandas as pd
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

***Exhibit 3* data**

In [ ]:
```
file_path = 'Perishing_portfolio.xlsx'
sheet_name = 'portfolio'
evaluation_date = '2020-02-21'
days = 180
evaluation_date_next = datetime.strptime(evaluation_date, '%Y-%m-%d') + timedelt

df = pd.read_excel(file_path, sheet_name=sheet_name)
df.head(9)
```

Out[ ]:

	Firm Name	Ticker Code	Number of Shares	Value (in $ million)	Share price	% Ownership	% of Fund Value	Board Seat	I
0	Chipotle Mexican Grill	CMG	8.619146e+07	1494.56	17.340000	0.0620	0.213	Yes	
1	Hilton Worldwide	HLT	1.055680e+07	1138.02	107.799661	0.0374	0.162	No	
2	Lowe's Companies	LOW	8.613212e+06	1001.20	116.240028	0.0112	0.143	No	
3	Restaurant Brands International	QSR	1.546818e+07	943.71	61.009759	0.0334	0.134	No	
4	Berkshire Hathaway (B)	BRK-B	4.015594e+06	901.22	224.430059	0.0029	0.128	No	
5	Howard Hughes	HHH	6.386835e+06	777.15	121.679987	0.1478	0.111	Yes	
6	Agilent Technologies	A	8.807760e+06	727.17	82.560152	0.0284	0.104	No	
7	Fannie Mae	FNMA	1.310045e+08	417.90	3.189966	0.1131	0.060	No	
8	Freddie Mac	FMCC	7.201052e+07	220.35	3.059969	0.1108	0.031	No	

1.1 Unhedged Portfolio Value

In [ ]:
```
share_nums = {}

for index, row in df.iterrows():
    ticker = row['Ticker Code']
    shares = row['Number of Shares']
    share_nums[ticker] = shares


date_range = pd.date_range(start=evaluation_date, end=evaluation_date_next)
portfolio_values = pd.DataFrame(index=date_range)

# Stock prices for the evaluation period
for ticker, shares in share_nums.items():
    historical_data = yf.download(ticker, start=evaluation_date, end=evaluation_

    if not historical_data.empty:
        daily_values = historical_data * shares
        portfolio_values[ticker] = daily_values

portfolio_values.fillna(method='ffill', inplace=True)
portfolio_values['Firm Value'] = portfolio_values.sum(axis=1)

print(f"\nTotal portfolio value on {evaluation_date}: ${portfolio_values['Firm V

# Safekeeping:
portfolio_values.index.name = 'Date'
portfolio_values.to_excel('portfolio_value.xlsx', sheet_name='Portfolio Values')

portfolio_values.head()
```

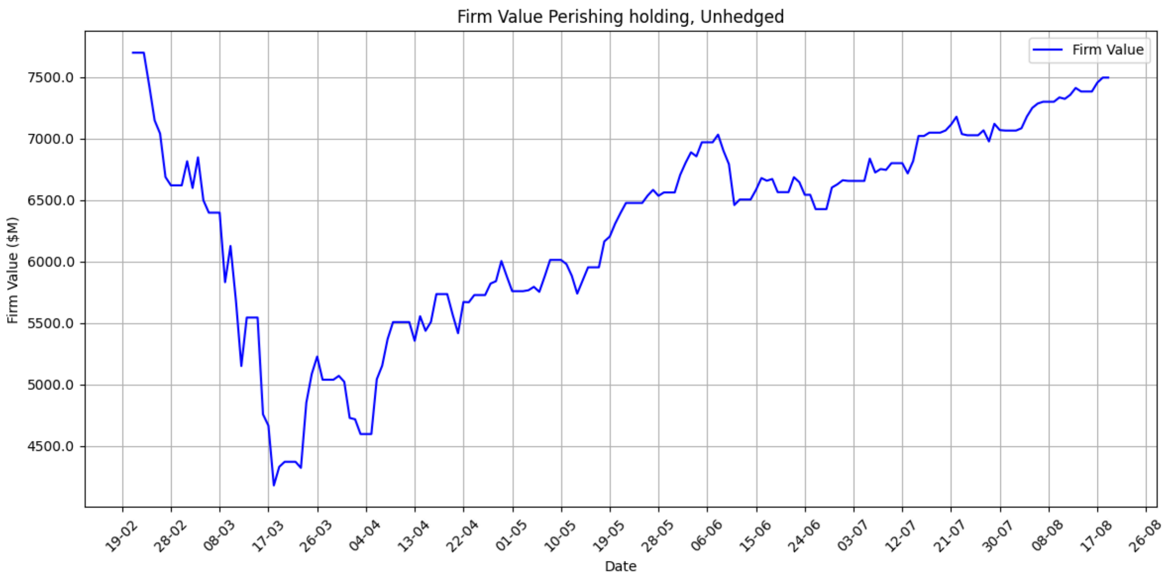
```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Total portfolio value on 2020-02-21: $7697405984.92
C:\Users\campd\AppData\Local\Temp\ipykernel_2208\2353543934.py:20: FutureWarning:
DataFrame.fillna with 'method' is deprecated and will raise in a future version.
Use obj.ffill() or obj.bfill() instead.
  portfolio_values.fillna(method='ffill', inplace=True)
```

Out[ ]:

Date	CMG	HLT	LOW	QSR	BRK-B	
2020-02-21	1.586164e+09	1.143925e+09	9.953334e+08	8.627389e+08	9.208962e+08	7.82372
2020-02-22	1.586164e+09	1.143925e+09	9.953334e+08	8.627389e+08	9.208962e+08	7.82372
2020-02-23	1.586164e+09	1.143925e+09	9.953334e+08	8.627389e+08	9.208962e+08	7.82372
2020-02-24	1.525589e+09	1.085983e+09	9.762703e+08	8.423065e+08	8.902170e+08	7.66542
2020-02-25	1.475615e+09	1.032320e+09	9.414006e+08	8.243628e+08	8.756003e+08	7.29523

In [ ]:

```
import matplotlib.ticker as ticker
plot_x_axis_interval = days / 20

plt.figure(figsize=(12, 6))
plt.plot(portfolio_values.index, portfolio_values['Firm Value'], linestyle='-',
plt.title('Firm Value Perishing holding, Unhedged')
plt.xlabel('Date')
plt.ylabel('Firm Value ($M)')
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=int(plot_x_axis_int
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d-%m'))  # format it
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: f'{x*1e-
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```



1.2 Acknowledgements and Tooling

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

2. Partial or Full Liquidation into Cash

In this strategy, a portion of the portfolio is liquidated and the proceeds are invested at the Secured Overnight Financing Rate (SOFR). The remaining portfolio continues to track the performance of the constituent firms. For a full liquidation, all assets are converted to cash, eliminating exposure to market risk.

Literature Review - Liquidation and Exposure

In this scenario, partial liquidation maintains exposure to market risk, as the portfolio continues to hold the remaining stocks after the liquidation. The extent of the exposure, however, is reduced based on the percentage of the portfolio liquidated (25%, 50%, 75%, or 100%), with less exposure as the liquidation percentage increases. Full liquidation eliminates all market risk since the entirety of the assets are reallocated into cash. This ensures that, while some exposure persists in partial liquidation, it is significantly less compared to an unhedged position. The varying degrees of liquidation reduce the potential downside risk but also limit participation in any positive market movements.

Assumptions

- No transaction costs are incurred during the liquidation of shares.
- Proceeds from the liquidation are invested at the SOFR risk-free rate starting on February 3, 2020.
- When a percentage (x%) of the portfolio is liquidated, the number of shares held in the portfolio is reduced by x%.
- No dividends are paid from any asset in the portfolio, as it is assumed to be a poor year for dividend payouts.
- Liquidation occurs instantly, and there are no delays in reinvesting funds at the SOFR rate.
- Share prices of the portfolio do not change in value at the time of sale. This is outlined in more detail in the relevant section of the report

Table of Contents

Tooling

In [ ]:
```
pip install yfinance openpyxl pandas matplotlib
```

```
Requirement already satisfied: yfinance in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (0.2.43)
Requirement already satisfied: openpyxl in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (3.0.10)
Requirement already satisfied: pandas in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (1.4.4)
Requirement already satisfied: matplotlib in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (3.5.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (4.11.1)
Requirement already satisfied: frozendict>=2.3.4 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (2.4.4)
Requirement already satisfied: platformdirs>=2.0.0 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (2.5.2)
Requirement already satisfied: lxml>=4.9.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (4.9.1)
Requirement already satisfied: multitasking>=0.0.7 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (0.0.11)
Requirement already satisfied: requests>=2.31 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (2.32.3)
Requirement already satisfied: pytz>=2022.5 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (2024.1)
Requirement already satisfied: peewee>=3.16.2 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (3.17.6)
Requirement already satisfied: html5lib>=1.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (1.1)
Requirement already satisfied: numpy>=1.16.5 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from yfinance) (1.21.5)
Requirement already satisfied: et_xmlfile in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from openpyxl) (1.1.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (1.4.2)
Requirement already satisfied: packaging>=20.0 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (21.3)
Requirement already satisfied: pillow>=6.2.0 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (9.2.0)
Requirement already satisfied: fonttools>=4.22.0 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: soupsieve>1.2 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from beautifulsoup4>=4.11.1->yfinance) (2.3.1)
Requirement already satisfied: six>=1.9 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from requests>=2.31->yfinance) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from requests>=2.31->yfinance) (3.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/daniel.gohh/opt/anaconda3/lib/python3.9/site-packages (from requests>=2.31->yfinance) (1.26.11)
Note: you may need to restart the kernel to use updated packages.
```

Libraries

In [ ]:
```
import yfinance as yf
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import math as math
import copy
```

Constants

- The Secured Overnight Financing Rate (SOFR) data used in this analysis was sourced from the official New York Federal Reserve website: SOFR Reference Rates.

Data Required

- **180-Day Average SOFR Rate**: 1.71663%.
- **Reasoning**: The analysis assumes that the investment was made on February 3, 2024. Since the evaluation period spans from February to August, the 180-day average SOFR rate starting on February 3, 2024, was chosen to provide a consistent and relevant benchmark for calculating returns on liquidated cash positions.

In [ ]:
```
file_path = 'Perishing_portfolio.xlsx'
sheet_name = 'portfolio'
evaluation_date = '2020-02-21'
days = 180
evaluation_date_next = datetime.strptime(evaluation_date, '%Y-%m-%d') + timedelt
plot_x_axis_interval = days / 20
share_port = {}
share_nums = {}
share_price = {} # Share price at evaluation date (2020-01-31)
SOFR_RATE_CONS = 0.0171663

# Liquidation constants
QUARTER_LIQUIDATION = 0
HALF_LIQUIDATION = 1
THIRD_QUARTILE_LIQUIDATION = 2
FULL_LIQUIDATION = 3

QUARTER = 0.25
HALF = 0.50
THIRD_QUARTER = 0.75
FULL = 1
```

2.1 Value of Partially Liquidated Portfolio

Four scenarios will be investigated:

- Liquidation of 25%
- Liquidation of 50%
- Liduidation of 75%
- Full Liquidation

In [ ]:
```
def modifyNumShares(share_tracker, constant, soldShares):
    if (constant == QUARTER_LIQUIDATION):
        liquidation_fact = QUARTER
    elif (constant == HALF_LIQUIDATION):
        liquidation_fact = HALF
    elif (constant == THIRD_QUARTILE_LIQUIDATION):
        liquidation_fact = THIRD_QUARTER
    else:
        liquidation_fact = FULL

    for key in share_tracker:
        num_shares = share_tracker[key] #Get number of shares
        shares_sold = math.ceil(num_shares * liquidation_fact) #Apply the liqati
        soldShares[key] = shares_sold
        shares_remaning = num_shares - shares_sold
        share_tracker[key] = math.ceil(shares_remaning)
```

In [ ]:
```
df = pd.read_excel(file_path, sheet_name=sheet_name)
full_liquid = False
frame_arr = []
for index, row in df.iterrows():
    ticker = row['Ticker Code']
    shares = row['Number of Shares']
    price = row['Share price']
    share_port[ticker] = shares
    share_price[ticker] = round(price,  4)


final_value = 0
#Could place this into a for loop so that it graphs can all be graphed at once.
for loop in range(4):
    shares_sold = {}
    share_nums = copy.copy(share_port)
    if (loop == FULL_LIQUIDATION):
        print("FULL LIQUIDATION")
        modifyNumShares(share_nums, FULL_LIQUIDATION, shares_sold) # Method can
        full_liquid = True
    elif (loop == THIRD_QUARTILE_LIQUIDATION):
        print("THIRD QUARTER LIQUIDATION")
        modifyNumShares(share_nums, THIRD_QUARTILE_LIQUIDATION, shares_sold)
    elif (loop == HALF_LIQUIDATION):
        print("HALF LIQUIDATION")
        modifyNumShares(share_nums, HALF_LIQUIDATION, shares_sold)
    elif (loop == QUARTER_LIQUIDATION):
        print("QUARTER LIQUIDATION")
        modifyNumShares(share_nums, QUARTER_LIQUIDATION, shares_sold)

    print("Shares Sold")
    print(shares_sold)
    print("Shares Remaining")
    print(share_nums)

    print("Shares Price")
    print(share_price)

    #Shares sold at shares price at 31-01-2020
    # Invested at the risk free rate.
    #TODO: Do the same thing but now apply the interest rates to liquid cash and
```

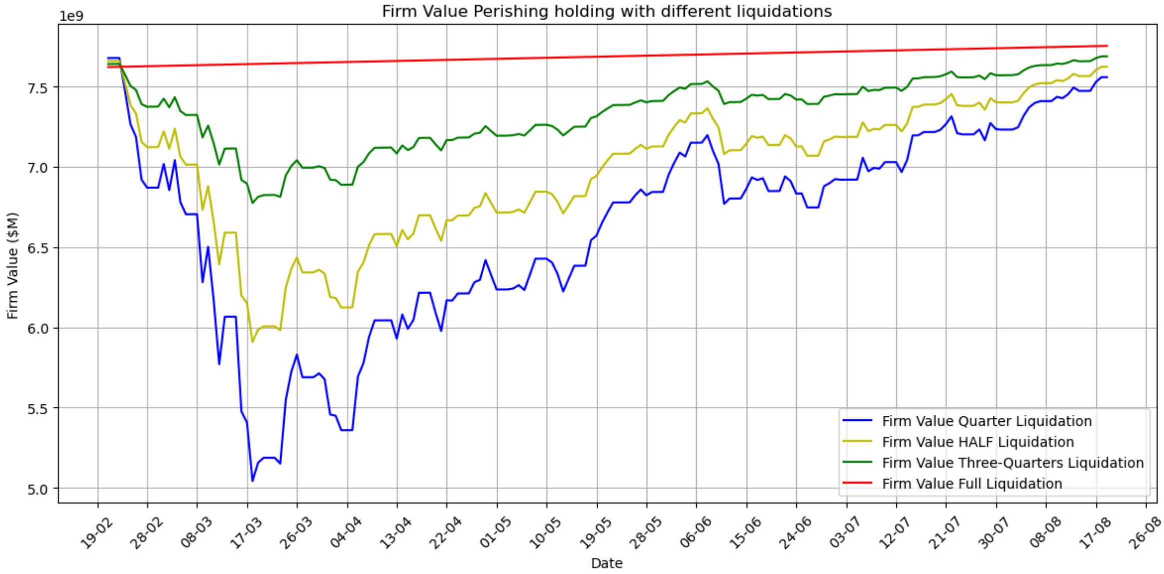
```
    cash_value = 0
    if (full_liquid):
        for index, row in df.iterrows():
            cash_value += row['Value (in $ million)']
        cash_value = cash_value * 1000000

    else:
        for key in shares_sold:
            cash_value += shares_sold[key] * share_price[key]
            print("CASH: ", cash_value)

    print("Cash Value: ", cash_value)
    values = np.zeros((days + 1,), dtype=float) #TO HOLD LIQUIDATION VALUES WITH
    date_range = pd.date_range(start=evaluation_date, end=evaluation_date_next)

    portfolio_values = pd.DataFrame(index=date_range)
    for ticker in share_nums:
        historical_data = yf.download(ticker, start=evaluation_date, end=evaluat
        shares = share_nums[ticker]
        if not historical_data.empty:
            daily_values = historical_data * shares
            portfolio_values[ticker] = daily_values


    portfolio_values.fillna(method='ffill', inplace=True)

    for day in range(0, days + 1):
        values[day] = cash_value * math.exp(SOFR_RATE_CONS * (day/days))
    frame = {'Liquidation Value': values}
    Liquidaion_values = pd.DataFrame(data=frame, index=date_range)
    portfolio_values['Liquidation Value'] = Liquidaion_values['Liquidation Value

    portfolio_values['Firm Value'] = portfolio_values.sum(axis=1)
    frame_arr.append(portfolio_values)

plt.figure(figsize=(12, 6))
plt.plot(frame_arr[QUARTER_LIQUIDATION].index, frame_arr[QUARTER_LIQUIDATION]['F
plt.plot(frame_arr[HALF_LIQUIDATION].index, frame_arr[HALF_LIQUIDATION]['Firm Va
plt.plot(frame_arr[THIRD_QUARTILE_LIQUIDATION].index, frame_arr[THIRD_QUARTILE_L
plt.plot(frame_arr[FULL_LIQUIDATION].index, frame_arr[FULL_LIQUIDATION]['Firm Va
plt.title('Firm Value Perishing holding with different liquidations')
plt.xlabel('Date')
plt.ylabel('Firm Value ($M)')
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=int(plot_x_axis_int
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d-%m'))  # format it
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```

```
QUARTER LIQUIDATION
Shares Sold
{'CMG': 21547867, 'HLT': 2639202, 'LOW': 2153303, 'QSR': 3867046, 'BRK-B': 100389
9, 'HHH': 1596709, 'A': 2201940, 'FNMA': 32751128, 'FMCC': 18002631}
Shares Remaining
{'CMG': 64643598, 'HLT': 7917603, 'LOW': 6459909, 'QSR': 11601135, 'BRK-B': 30116
95, 'HHH': 4790126, 'A': 6605820, 'FNMA': 98253383, 'FMCC': 54007892}
Shares Price
{'CMG': 17.34, 'HLT': 107.7997, 'LOW': 116.24, 'QSR': 61.0098, 'BRK-B': 224.4301,
'HHH': 121.68, 'A': 82.5602, 'FNMA': 3.19, 'FMCC': 3.06}
CASH:  373640013.78
CASH:  658145197.6194
CASH:  908445138.3394
CASH:  1144372841.3902001
CASH:  1369677994.3501
CASH:  1563965545.4701
CASH:  1745758152.2581
CASH:  1850234250.5781
CASH:  1905322301.4380999
Cash Value:  1905322301.4380999
```

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

```
HALF LIQUIDATION
Shares Sold
{'CMG': 43095733, 'HLT': 5278403, 'LOW': 4306606, 'QSR': 7734091, 'BRK-B': 200779
7, 'HHH': 3193418, 'A': 4403880, 'FNMA': 65502256, 'FMCC': 36005262}
Shares Remaining
{'CMG': 43095732, 'HLT': 5278402, 'LOW': 4306606, 'QSR': 7734090, 'BRK-B': 200779
7, 'HHH': 3193417, 'A': 4403880, 'FNMA': 65502255, 'FMCC': 36005261}
Shares Price
{'CMG': 17.34, 'HLT': 107.7997, 'LOW': 116.24, 'QSR': 61.0098, 'BRK-B': 224.4301,
'HHH': 121.68, 'A': 82.5602, 'FNMA': 3.19, 'FMCC': 3.06}
CASH:  747280010.22
CASH:  1316290270.0991
CASH:  1816890151.5391002
CASH:  2288745496.6309004
CASH:  2739355578.1206
CASH:  3127930680.3606005
CASH:  3491515893.9366007
CASH:  3700468090.5766006
CASH:  3810644192.2966003
Cash Value:  3810644192.2966003
```

```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
THIRD QUARTER LIQUIDATION
Shares Sold
{'CMG': 64643599, 'HLT': 7917604, 'LOW': 6459909, 'QSR': 11601136, 'BRK-B': 3011696, 'HHH': 4790127, 'A': 6605820, 'FNMA': 98253384, 'FMCC': 54007893}
Shares Remaining
{'CMG': 21547866, 'HLT': 2639201, 'LOW': 2153303, 'QSR': 3867045, 'BRK-B': 1003898, 'HHH': 1596708, 'A': 2201940, 'FNMA': 32751127, 'FMCC': 18002630}
Shares Price
{'CMG': 17.34, 'HLT': 107.7997, 'LOW': 116.24, 'QSR': 61.0098, 'BRK-B': 224.4301, 'HHH': 121.68, 'A': 82.5602, 'FNMA': 3.19, 'FMCC': 3.06}
CASH:  1120920006.66
CASH:  1974435342.5788002
CASH:  2725335164.7388
CASH:  3433118151.8716
CASH:  4109033386.3212004
CASH:  4691896039.6812
CASH:  5237273860.0452
CASH:  5550702155.0052
CASH:  5715966307.5852
Cash Value:  5715966307.5852
FULL LIQUIDATION
Shares Sold
{'CMG': 86191465, 'HLT': 10556805, 'LOW': 8613212, 'QSR': 15468181, 'BRK-B': 4015594, 'HHH': 6386835, 'A': 8807760, 'FNMA': 131004511, 'FMCC': 72010523}
Shares Remaining
{'CMG': 0, 'HLT': 0, 'LOW': 0, 'QSR': 0, 'BRK-B': 0, 'HHH': 0, 'A': 0, 'FNMA': 0, 'FMCC': 0}
Shares Price
{'CMG': 17.34, 'HLT': 107.7997, 'LOW': 116.24, 'QSR': 61.0098, 'BRK-B': 224.4301, 'HHH': 121.68, 'A': 82.5602, 'FNMA': 3.19, 'FMCC': 3.06}
Cash Value:  7621280000.0
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
```



2.2 Acknowledgements and Tooling

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

3. Futures

Investigate the mechanics and use of short selling S&P 500 futures contracts via the *Chicago Mercantile Exchange* (CME) as outlined in Document *A* of the case study.

The following investigation investigates futures prices of two contracts that would have been available during this time, i.e. *ESH20* and *ESM20*, with maturities in March and June, respectively. The trading prices for these contracts were sourced from BarChart Premiere historical data. A literature review (where relevant) and coding examples have been provided as part of this analysis.

Key Assumptions & limitations

- For the purposes of this assignment, we will assume that the evaluation date (i.e. the date in which Pershing Square must make a decision on their Hedging strategy) is February 21, 2020
- Purchasing of futures on the *Chicago Mercantile Exchange* (CME)
- Where discrepancies between Yahoo Finance share prices and derived share prices from *Exhibit 3* were found, the Yahoo Finance share prices were adopted. Details are outlined further in 3.1

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

Table of Contents

Libraries

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

import warnings
warnings.filterwarnings("ignore")
```

3.1 Hedging Position and Key details

To protect Pershing Square's portfolio from adverse market movements, they could utilise S&P 500 index futures contracts as a hedging instrument. This approach would allow the portfolio to gain protection against potential downside movements in the S&P 500

A futures contract is an agreement to buy or sell the underlying asset (in this case, the S&P 500 index) at a predetermined price on a specific date in the future. This allows the holder to lock in a price for the underlying asset, helping to protect against unfavorable price movements.

Ongoing Cashflows and Costs

Whilst there are no upfront costs or regular payments like options and Credit Default Swaps (CDS) we must consider that an initial margin and maintenance margin must be maintained over the life of ownership of the asset. A coding example calculating the initial margin of this contract is demonstrated in section 3.1. The initial margin figure is sourced from CME risk data.

Since and observation of the payoff yield an immediate positive (and never a non-negative) payoff, we did not calculate any maintenance margin requires for these contracts.

Futures Contract Payoff

The payoff of a futures contract at expiration depends on the relationship between the futures price (S_F) and the spot price of the underlying asset (S_T). If the spot price at expiration is higher than the agreed-upon futures price, the long position benefits, while the short position loses. Conversely, if the spot price is lower, the short position benefits.

The payoff of a short position futures contract is given by:

$$\text{Futures Contract Payoff} = K - S_T$$

Where:

- S_T is the spot price of the S&P 500 index at expiration.
- K is the agreed-upon futures price.

Determining the Optimal Number of Contracts

To determine the optimal number of futures 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 $K_t - K$.

For the purposes of this assignment, σ_{A_t}, σ_{K_t} and ρ were calculated from historical daily returns in the prior four-year period (`HISTORICAL_PRICE_DATE`) to the evaluation date (`EVAL_DATE`).

In [ ]:
```
# Contract information
ESH20_CONTRACT_SIZE = 50
ESM20_CONTRACT_SIZE = 50
ESH20_HIST_PRICES = 'futures_data/esh20_price-history-09-24-2024.csv'
ESM20_HIST_PRICES = 'futures_data/esm20_price-history-09-24-2024.csv'
ESH20_MATURITY = '2020-03-30'
ESM20_MATURITY = '2020-06-30'
ESH20_INITIAL_MARGIN = 11_000.0
ESM20_INITIAL_MARGIN = 11_000.0

# Firm specific information
FIRM_VALUE = 7_621.28 * 1_000_000

# Contextual information
HISTORICAL_PRICE_DATE = '2016-01-04'
EVAL_DATE = '2020-02-21'
EVAL_DATE_PLUS_ONE = '2020-02-22'
EVAL_END_DATE = '2020-06-30'
```

Calculate the portfolio value, weights, historical returns and standard deviations.

This is a bit more complicated because the weighting for each firm in the portfolio varies as per the appendix provided on the assignment task sheet.

Process:

Import the historical prices for each firm, evaluate the firm value based upon the constituent assets owned in the portfolio and calculate the daily returns of the portfolio

In [ ]:
```
file_path = 'Perishing_portfolio.xlsx'
sheet_name = 'portfolio'
share_nums = {}

df = pd.read_excel(file_path, sheet_name=sheet_name)

for index, row in df.iterrows():
    ticker = row['Ticker Code']
    shares = row['Number of Shares']
    share_nums[ticker] = shares
```

```
# Also add the ^spx index
share_nums['^spx'] = 1

date_range = pd.date_range(start=HISTORICAL_PRICE_DATE, end=EVAL_END_DATE)
df = pd.DataFrame(index=date_range)

# stock prices for the evaluation period
for ticker, shares in share_nums.items():
    historical_data = yf.download(ticker, start=HISTORICAL_PRICE_DATE, end=EVAL_

    if not historical_data.empty:
        daily_values = historical_data * shares
        df[ticker] = daily_values

df = df.dropna(how='any')
df['Firm Value'] = df.loc[:, df.columns != '^spx'].sum(axis=1)
df_historic = df[df.index <= EVAL_DATE_PLUS_ONE]

df.index.name = 'Date'

p_returns = np.log(df_historic['Firm Value']).diff(1).dropna()
spx_returns = np.log(df_historic['^spx']).diff(1).dropna()

df_historic['Portfolio Returns'] = p_returns
df_historic['S&P 500 Returns'] = spx_returns
df_historic = df_historic.dropna()

# Portfolio price (we have discovered in previous sections)
p_price_eval_date = df_historic['Portfolio Returns'].loc[EVAL_DATE]

# Portfolio sigma
p_std_dev = np.std(p_returns)

# S&P 500 sigma
spx_std_dev = np.std(spx_returns)

# Safekeeping:
df_historic.to_excel('historic_portfolio_values.xlsx', sheet_name='Values')

df.head()
```

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

Out[ ]:

Date	CMG	HLT	LOW	QSR	BRK-B	
2016-01-04	7.736718e+08	4.357516e+08	5.524744e+08	4.257521e+08	5.250389e+08	6.644987
2016-01-05	7.740511e+08	4.359580e+08	5.536490e+08	4.094645e+08	5.270467e+08	6.575578
2016-01-06	7.355062e+08	4.132306e+08	5.426402e+08	3.970094e+08	5.273680e+08	6.514694
2016-01-07	7.171130e+08	3.977343e+08	5.289166e+08	3.879076e+08	5.199391e+08	6.362481
2016-01-08	7.124415e+08	3.956681e+08	5.201833e+08	3.810812e+08	5.153212e+08	6.355175

Calculate optimal portfolio

> Note: The notional value and initial margin are the same as both contracts have the same contract size

In [ ]:

```
# Calculate rho
rho=np.corrcoef(df_historic['S&P 500 Returns'], df_historic['Portfolio Returns']

# Firm portfolio value and returns
beta = rho * spx_std_dev / p_std_dev

# optimal hedge ratio
spx_price_eval_date = df_historic['^spx'].loc[EVAL_DATE]
F = spx_price_eval_date * ESM20_CONTRACT_SIZE #
H = beta * FIRM_VALUE / F

# Calculate initial margin
margin_cost_esm20 = H * ESM20_CONTRACT_SIZE
margin_cost_esh20 = H * ESH20_CONTRACT_SIZE
```

Key Statistics

In [ ]:

```
stats = f'''
############ KEY STATISTICS ############
Standard deviation of portfolio from historical date ({HISTORICAL_PRICE_DATE}) t
Standard deviation of S&P 500 Sigma from historical date ({HISTORICAL_PRICE_DATE
Rho: {rho:.6f}
Beta: {beta:.6f}

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


Optimal number of contracts to hedge firm position: {int(round(H, 0))}
(unrounded {H:.2f})

############ INITIAL MARGIN ############
Initial Margin (ESM20): {margin_cost_esm20:.2f}
Initial Margin (ESH20): {margin_cost_esh20:.2f}
```

```
'''

print(stats)
```

```
############ KEY STATISTICS ############
Standard deviation of portfolio from historical date (2016-01-04) to evaluation d
ate (2020-02-21): 0.010099
Standard deviation of S&P 500 Sigma from historical date (2016-01-04) to evaluati
on date (2020-02-21): 0.008079
Rho: 0.741110
Beta: 0.592861

Face value of futures contract on evaluation date (2020-02-21): $166887.50


Optimal number of contracts to hedge firm position: 27074
(unrounded 27074.27)

############ INITIAL MARGIN ############
Initial Margin (ESM20): 1353713.70
Initial Margin (ESH20): 1353713.70
```

3.2 (Retrospective) Key Timings

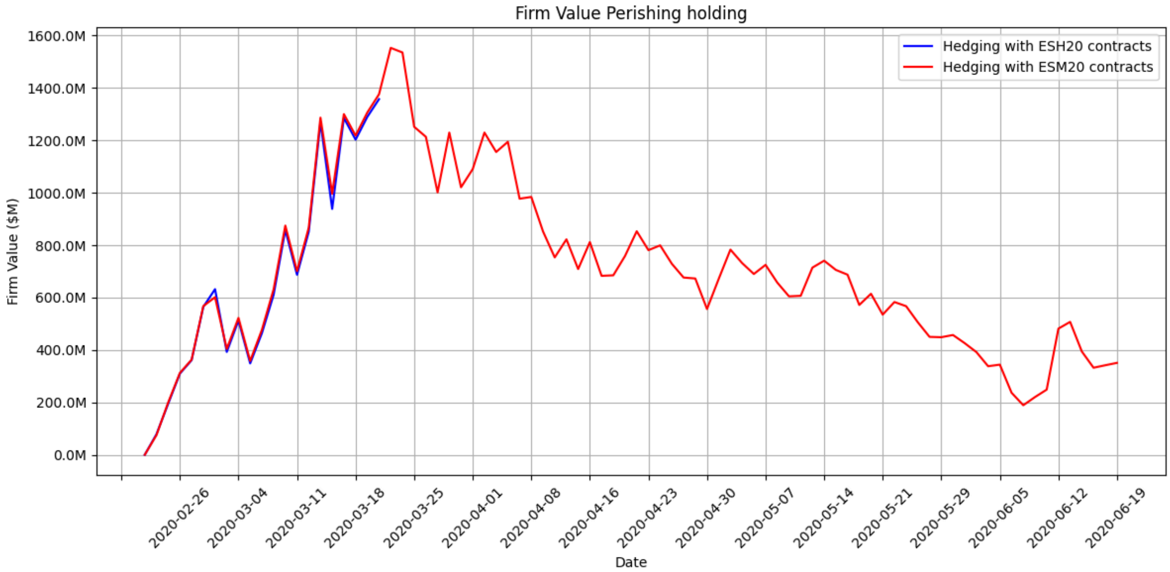
Firstly, we will graphically illustrate the payoffs of the purchasing the contracts starting from the evaluation date to the contract maturity, i.e. 31st March and 30th June.

We will characterise the best exit timing of these securities as the date which would yield the maximum payoff between the evaluation date and maturity.

In [ ]:

```
import matplotlib.ticker as ticker
# Retrieve historic futures data for ESH20 and ESM20

def generate_futures_payoffs(historical_prices_wd, contract_size):

    futures_prices_df = pd.read_csv(historical_prices_wd, index_col='Time', pars
    futures_prices_df = futures_prices_df[futures_prices_df.index >= EVAL_DATE]
    futures_prices_df.index = pd.to_datetime(futures_prices_df.index).strftime('
    futures_prices_df = futures_prices_df.sort_index()
    futures_prices_df = futures_prices_df.rename(columns={'Open': 'Price'})

    esh20_eval_date = futures_prices_df['Price'].loc[EVAL_DATE]

    futures_prices_df['Payoff from EVAL_DATE'] = (esh20_eval_date - futures_pric
    return futures_prices_df

esh20_prices_df = generate_futures_payoffs(ESH20_HIST_PRICES, ESH20_CONTRACT_SIZ
esm20_prices_df = generate_futures_payoffs(ESM20_HIST_PRICES, ESM20_CONTRACT_SIZ

plt.figure(figsize=(12, 6))
plt.plot(esh20_prices_df.index, esh20_prices_df['Payoff from EVAL_DATE'], linest
plt.plot(esm20_prices_df.index, esm20_prices_df['Payoff from EVAL_DATE'], linest
plt.title('Firm Value Perishing holding')
plt.xlabel('Date')
plt.ylabel('Firm Value ($M)')
plt.xticks(rotation=45)
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=5))
```

```
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: f'{x*1e-
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```



In [ ]:
```
# The best date to close the futures contract with a reversing trade would be:
max_price_date = esm20_prices_df['Payoff from EVAL_DATE'].idxmax()

print(f'The best exit timing of the market would be {max_price_date.strftime('%Y
```

```
The best exit timing of the market would be 2020-03-23.
```

3.3 Profits

We now consider the overall profits of Pershing Square maintaining their current portfolio and the futures hedging position.

The profit diagram of their firm is illustrated is the worked coding example below.

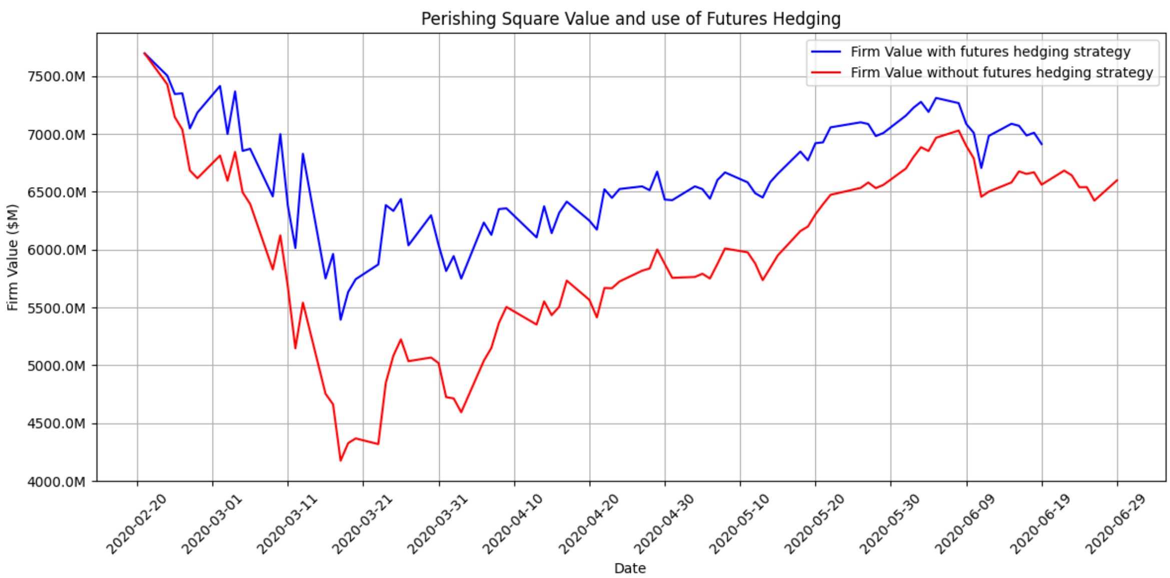
In [ ]:
```
# Given the payoff diagram above, we will only look at the ESM20 contract since
df_profits = df[df.index >= EVAL_DATE].copy()
df_profits = df_profits[['Firm Value']]

# Shuffling because the data isn't perfect
df_profits.index = pd.to_datetime(df_profits.index).normalize()
esm20_prices_df.index = pd.to_datetime(esm20_prices_df.index).normalize()
aligned_esm20_payoff = esm20_prices_df['Payoff from EVAL_DATE'].reindex(df_profi

df_profits.loc[:, 'ESM Payoff'] = aligned_esm20_payoff

firm_value_eval_date = df_profits['Firm Value'].loc[EVAL_DATE]
df_profits['Firm Profit'] = df_profits['Firm Value'] - firm_value_eval_date
df_profits['Firm Profit With Futures Hedge'] = df_profits['ESM Payoff'] + df_prc
df_profits['Firm Value With Futures Hedge'] = df_profits['ESM Payoff'] + df_prof

df_profits.head()
```

Out[ ]:

Date	Firm Value	ESM Payoff	Firm Profit	Firm Profit With Futures Hedge	Firm Value With Futures Hedge
2020-02-21	7.696251e+09	0.000000e+00	0.000000e+00	0.000000e+00	7.696251e+09
2020-02-24	7.429246e+09	7.513111e+07	-2.670049e+08	-1.918738e+08	7.504377e+09
2020-02-25	7.145713e+09	1.996728e+08	-5.505380e+08	-3.508652e+08	7.345386e+09
2020-02-26	7.038965e+09	3.123694e+08	-6.572857e+08	-3.449162e+08	7.351335e+09
2020-02-27	6.684721e+09	3.631337e+08	-1.011530e+09	-6.483962e+08	7.047855e+09

In [ ]:

```
plt.figure(figsize=(12, 6))
plt.plot(df_profits.index, df_profits['Firm Value With Futures Hedge'], linestyl
plt.plot(df_profits.index, df_profits['Firm Value'], linestyle='-', color='r', l
plt.title('Perishing Square Value and use of Futures 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()
```



3.4 Sensitivity Analysis

A futures contract can be priced using the cost of carry approach

$$K = Se^{(r+s-q)T}$$

(using **Compound Interest**)

i.e. that the futures price is dependent on the risk-free interest rate r, storage cost c and dividend yield q in some period T.

3.4.1 Forecast Changes in Cost of Carry

The following code example forecasts changes in the estimated cost of carry:

- The cost of carry is first derived as of the evaluation date (February 21, 2020).
- The payoffs are then evaluated under different cost of carry values, considering forecast periods of one week, one month, and two months into the future.

In [ ]:

```
eval_date_as_date = datetime.strptime(EVAL_DATE, '%Y-%m-%d')
esm20_maturity_as_date = datetime.strptime(ESM20_MATURITY, '%Y-%m-%d')

T = (esm20_maturity_as_date - eval_date_as_date).days / 360

K = esm20_prices_df['Price'].loc[EVAL_DATE]
S = spx_price_eval_date

cost_of_carry = np.log((K / S))* (1 / T)

print(f'K: {K} \nS: {S}\nT: {T}\nCost of Carry: {cost_of_carry:3f}')
```

```
K: 3367.5
S: 3337.75
T: 0.3611111111111111
Cost of Carry: 0.024573
```

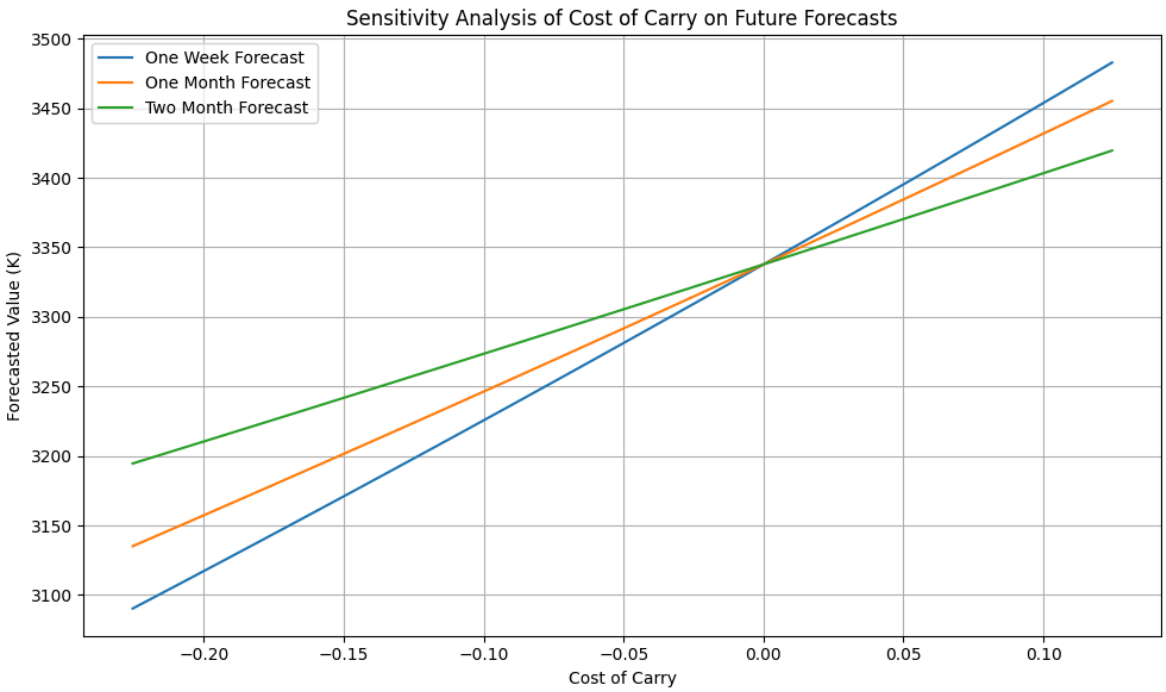
In [ ]:

```
min_value = cost_of_carry - 0.25
max_value = cost_of_carry + 0.1

T_after_one_week = ((esm20_maturity_as_date - eval_date_as_date).days - 7) / 360
T_after_one_month = ((esm20_maturity_as_date - eval_date_as_date).days - 30) / 3
T_after_two_months = ((esm20_maturity_as_date - eval_date_as_date).days - 60) /

sensitivity_values = np.linspace(min_value, max_value, 50)
df_sensitivity = pd.DataFrame(sensitivity_values, columns=['Cost Of Carry'])
df_sensitivity['K One Week Forecast'] = S * np.exp(df_sensitivity['Cost Of Carry
df_sensitivity['K One Month Forecast'] = S * np.exp(df_sensitivity['Cost Of Carr
df_sensitivity['K Two Month Forecast'] = S * np.exp(df_sensitivity['Cost Of Carr

df_sensitivity.head()
```

Out[ ]:

	Cost Of Carry	K One Week Forecast	K One Month Forecast	K Two Month Forecast
0	-0.225427	3090.324643	3135.154350	3194.606642
1	-0.218284	3097.875717	3141.381070	3199.046678
2	-0.211141	3105.445242	3147.620157	3203.492885
3	-0.203998	3113.033263	3153.871634	3207.945272
4	-0.196855	3120.639824	3160.135528	3212.403847

In [ ]:
```python
plt.figure(figsize=(10, 6))
plt.plot(df_sensitivity['Cost Of Carry'], df_sensitivity['K One Week Forecast'],
plt.plot(df_sensitivity['Cost Of Carry'], df_sensitivity['K One Month Forecast']
plt.plot(df_sensitivity['Cost Of Carry'], df_sensitivity['K Two Month Forecast']
plt.title('Sensitivity Analysis of Cost of Carry on Future Forecasts')
plt.xlabel('Cost of Carry')
plt.ylabel('Forecasted Value (K)')
plt.legend()
plt.grid(True)


plt.tight_layout()
plt.show()
```



3.5 Acknowledgements and Tooling

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

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
```

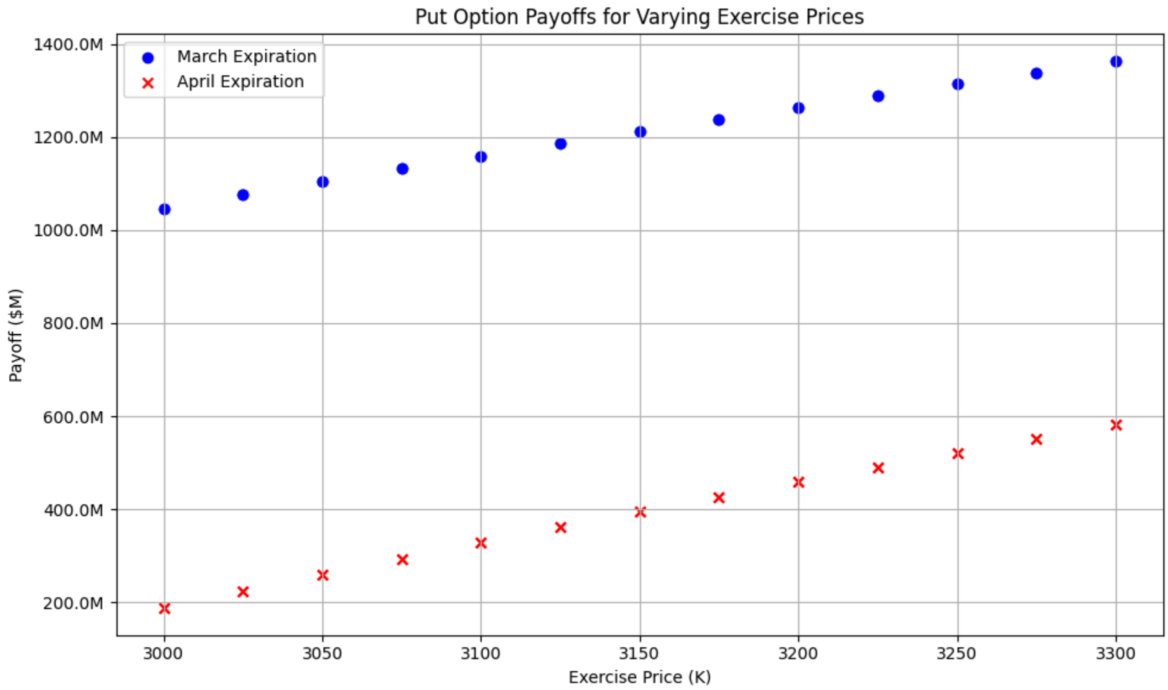
Payoffs

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)
```

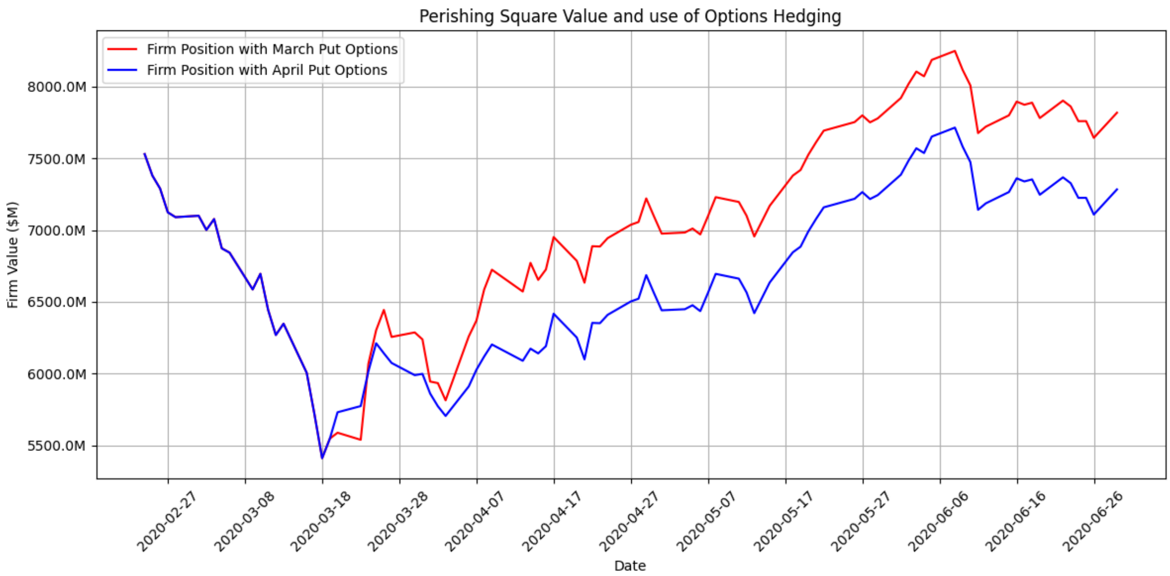
Out[ ]:

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

- 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.
```

5. CDS Hedging

Investigate the use of Credit Default Swaps (CDS) to hedge Perishing Sqaure's portfolio against market risk.

Review

For CDS contracts, the buyer and seller of default protection would initially exchange an “upfront” amount, after which the buyer made quarterly fixed coupon payments to the seller over the life of the contract. In return, for every notional dollar of protection, the seller of CDS would make a payment to the buyer if the bond experienced a so-called “credit event,” such as default or debt restructuring.

The CDS contracts bought by Perishing Square were written on individual bonds were known as single-name CDSs, but there were also contracts known as index CDS's that were written on baskets or portfolios of bonds. The two most common index CDS families were the CDX and iTraxx indices. The CDX indices tracked American Companies, while the iTraxx indices tracked European companies.

In addition, index CDS are centrally cleared which means that a clearing house acted as a counterparty to both sides of the transaction. When purchasing CDS protection on centrally cleared contracts, both parties were required to post initial margin based on the notional value of the contract and the risk of the underlying. This was in addition to any upfront payments made between the parties. Pershing Square estimated that their initial margins on the IG, iTraxx, and HY indices would be 0.8%, 0.9%, and 3.2% of notional, respectively.

In order to hedge Perishing Portfolio from market risk, the entire portfolio value at 2020-02-21 was hedged. This can be found

Information Sourced:

- Since there was limited free information about CDS spreads, an article, written by "The Short Bear" was found. The link to the site: https://theshortbear.substack.com/p/bill-ackmans-2650000000-trade.
- Calculations and spread information was sourced by the author of the document. (Found in excel file used)

Assumptions and Limitations

- With the excel file, it was decided that only one type of Credit Default Swaps (CDS) would be used to hedge its market risk/exposure.
- To see the effectiveness of hedging, we can invest in CDS that has same notional amount of Perishing's portfolio at 02-21.
- We also incoporated an upfront payment that needed to be paid from the buyer to the seller. That was calculated in this notebook.

- We can take a look at the generated profits from each single index and how they hedge against the lose of their portfolio with the use of the graph generated in this notebook.

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

import warnings
warnings.filterwarnings("ignore")
```

Working

In [ ]:
```
QUARTER = 0.25
#THIS STUFF MAY NOT BE NEEDED
NOTIONAL_CONTRACT = 100
# MATURITY = 5
EXPECTED_HOLDING_PERIOD = 60 #Expected to hold CDS for 60 days.

# #Current Data as of 02-21-2020
# RISKLESS_RATE_02_21 = 0.0130 #(Same between both indexes)

BREAKEVEN_CDXHY_SPREAD_02_21 = 0.0295
BREAKEVEN_CDXIG_SPREAD_02_21 = 0.0046

CDX_HY_UPFRONT_PAYMENT_PERCENTAGE_02_21 = 8.97/NOTIONAL_CONTRACT   #Per $100 of
CDX_IG_UPFRONT_PAYMENT_PERCENTAGE__02_21 = 2.55/NOTIONAL_CONTRACT

RECOVERY_RATE_CDXHY = 0.3
RECOVERY_RATE_CDXIG = 0.4
FIXED_CDS_PREMIUM_CDXIG = 0.01
FIXED_CDS_PREMIUM_CDXHY = 0.05


# LIQUIDATION_DATE = '4-21-2020'
# RISKLESS_RATE_04_21 = 0.0034
# BREAKEVEN_CDXHY_SPREAD_04_21 = 0.06
# CDX_HY_UPFRONT_PAYMENT_04_21 = 4.03

# BREAKEVEN_CDXIG_SPREAD_04_21 = 0.016
# CDX_IG_UPFRONT_PAYMENT_02_21 = 2.79

# #Expected Costs and Payoffs
# FIXED_COUPON_PAID_CDXHY = 0.83
# PAYOFF_CHANGE_IN_UPFRONT_CDXHY = 13
# MULTIPLE_ON_PREMIUMS_CDXHY = 15.6

# FIXED_COUPON_PAID_CDXIG = 0.17
# PAYOFF_CHANGE_IN_UPFRONT_CDXIG = 5.33
# MULTIPLE_ON_PREMIUMS_CDXIG = 32.0

VALUE_PORTFOLIO_02_21 = 7697406096.30

DOUBLE_VALUE_PORTFOLIO_02_21 = 15394812192.6
```

```
#Calculate upfront payment for CDS
CDX_HY_UPFRONT_PAYMENT = CDX_HY_UPFRONT_PAYMENT_PERCENTAGE_02_21 * VALUE_PORTFOL
print("CDX HY Upfront Cost:",  CDX_HY_UPFRONT_PAYMENT)

CDX_IG_UPFRONT_PAYMENT = CDX_IG_UPFRONT_PAYMENT_PERCENTAGE__02_21 * VALUE_PORTFO
print("CDX IG Upfront Cost:",  CDX_IG_UPFRONT_PAYMENT)

CDX_HY_DOUBLE_UPFRONT_PAYMENT = DOUBLE_VALUE_PORTFOLIO_02_21 * CDX_HY_UPFRONT_PA
print ("CDX_HY_Upfront Cost (Double):", CDX_HY_DOUBLE_UPFRONT_PAYMENT)

CDX_IG_DOUBLE_UPFRONT_PAYMENT = DOUBLE_VALUE_PORTFOLIO_02_21 * CDX_IG_UPFRONT_PA
print ("CDX_IG_Upfront Cost (Double):", CDX_IG_DOUBLE_UPFRONT_PAYMENT)

#Rest can be found in an excel file
```

```
CDX HY Upfront Cost: 690457326.8381101
CDX IG Upfront Cost: 196283855.45565
CDX_HY_Upfront Cost (Double): 1380914653.6762202
CDX_IG_Upfront Cost (Double): 392567710.9113
```

Payoffs from Hedging Perishing Portfolio Value at Feb-21 2020

In [ ]:

```
file_path = 'Bill Ackman $2.65b trade.xlsx'
sheet_name = 'Hedging'

#Just for formatting purposes
def format_dollars(val):
    if pd.notna(val):
        return "${:,.2f}".format(val)
    else:
        return val

df = pd.read_excel(file_path, sheet_name=sheet_name)
df['Payoffs'] = df['Payoffs'].apply('${:,.2f}'.format)
df['Spreads Before'] = (df['Spreads Before'] * 100).apply('{:.2f}%'.format)
df['Spreads After'] = (df['Spreads After'] * 100).apply('{:.2f}%'.format)
df['CDS Spread'] = (df['CDS Spread'] * 100).apply('{:.2f}%'.format)
df['Quarterly Premiums'] = df['Quarterly Premiums'].apply(format_dollars)
df['Upfront Payment $100 Notional value'] = df['Upfront Payment $100 Notional va
df['Upfront Payment'] = df['Upfront Payment'].apply(format_dollars)

df.head()
```

Out[ ]:

	CDS	Payoffs	Spreads Before	Spreads After	CDS Spread	Quarterly Premiums	Upfront Payment $100 Notional value	
0	CDX IG	$263,337,103.64	0.59%	1.30%	0.71%	$22,707,347.98	$2.55	$196,
1	CDX HY	$1,235,088,105.78	3.46%	6.79%	3.33%	$133,165,125.47	$8.97	$690,
2	ITRAXX Main	$248,501,210.47	0.60%	1.27%	0.67%	$23,092,218.29	NaN	

Payoffs Hedging With a Different Value

- The updated value is twice the value of the Perishing Portfolio as of February 21st.
- Using same excel spreadsheet, these are the new values:

In [ ]:
```
file_path = 'Bill Ackman $2.65b trade.xlsx'
sheet_name = 'Hedging V2'


df2 = pd.read_excel(file_path, sheet_name=sheet_name)
df2['Payoffs'] = df2['Payoffs'].apply('${:,.2f}'.format)
df2['Spreads Before'] = (df2['Spreads Before'] * 100).apply('{:.2f}%'.format)
df2['Spreads After'] = (df2['Spreads After'] * 100).apply('{:.2f}%'.format)
df2['CDS Spread'] = (df2['CDS Spread'] * 100).apply('{:.2f}%'.format)
df2['Quarterly Premiums'] = df2['Quarterly Premiums'].apply(format_dollars)
df2['Upfront Payment $100 Notional value'] = df2['Upfront Payment $100 Notional
df2['Upfront Payment'] = df2['Upfront Payment'].apply(format_dollars)

df2.head()
```

Out[ ]:

	CDS	Payoffs	Spreads Before	Spreads After	CDS Spread	Quarterly Premiums	Upfront Payment $100 Notional value	
0	CDX IG	$526,674,207.27	0.59%	1.30%	0.71%	$50,033,139.63	$2.55	$39
1	CDX HY	$2,470,176,211.57	3.46%	6.79%	3.33%	$133,165,125.47	$8.97	$1,38
2	ITRAXX Main	$497,002,420.95	0.60%	1.27%	0.67%	$23,092,218.29	NaN	