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 [ ]:

```python
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)
```

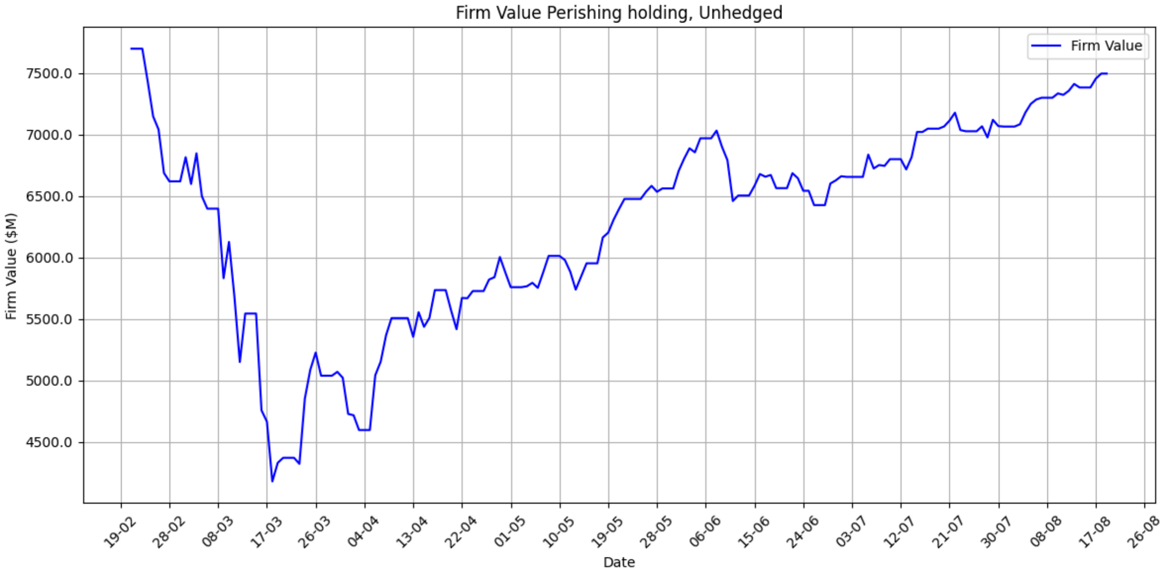
Out[ ]:

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.