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

- 2.1 Value of Partially Liquidated Portfolio
- 2.2 Acknowledgements and Tooling

Tooling

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

```
Requirement already satisfied: yfinance in /Users/daniel.gohh/opt/anaconda3/lib/p
ython3.9/site-packages (0.2.43)
Requirement already satisfied: openpyxl in /Users/daniel.gohh/opt/anaconda3/lib/p
ython3.9/site-packages (3.0.10)
Requirement already satisfied: pandas in /Users/daniel.gohh/opt/anaconda3/lib/pyt
hon3.9/site-packages (1.4.4)
Requirement already satisfied: matplotlib in /Users/daniel.gohh/opt/anaconda3/li
b/python3.9/site-packages (3.5.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /Users/daniel.gohh/opt/a
naconda3/lib/python3.9/site-packages (from yfinance) (4.11.1)
Requirement already satisfied: frozendict>=2.3.4 in /Users/daniel.gohh/opt/anacon
da3/lib/python3.9/site-packages (from yfinance) (2.4.4)
Requirement already satisfied: platformdirs>=2.0.0 in /Users/daniel.gohh/opt/anac
onda3/lib/python3.9/site-packages (from yfinance) (2.5.2)
Requirement already satisfied: lxml>=4.9.1 in /Users/daniel.gohh/opt/anaconda3/li
b/python3.9/site-packages (from yfinance) (4.9.1)
Requirement already satisfied: multitasking>=0.0.7 in /Users/daniel.gohh/opt/anac
onda3/lib/python3.9/site-packages (from yfinance) (0.0.11)
Requirement already satisfied: requests>=2.31 in /Users/daniel.gohh/opt/anaconda
3/lib/python3.9/site-packages (from yfinance) (2.32.3)
Requirement already satisfied: pytz>=2022.5 in /Users/daniel.gohh/opt/anaconda3/l
ib/python3.9/site-packages (from yfinance) (2024.1)
Requirement already satisfied: peewee>=3.16.2 in /Users/daniel.gohh/opt/anaconda
3/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/li
b/python3.9/site-packages (from openpyxl) (1.1.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/daniel.gohh/opt/a
naconda3/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/daniel.gohh/opt/anacond
a3/lib/python3.9/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /Users/daniel.gohh/opt/anaconda3/l
ib/python3.9/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/daniel.gohh/opt/anacon
da3/lib/python3.9/site-packages (from matplotlib) (1.4.2)
Requirement already satisfied: packaging>=20.0 in /Users/daniel.gohh/opt/anaconda
3/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/anacon
da3/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/p
ython3.9/site-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /Users/daniel.gohh/opt/anaconda3/l
ib/python3.9/site-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/daniel.gohh/op
t/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/anaco
nda3/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/l
ib/python3.9/site-packages (from requests>=2.31->yfinance) (3.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/daniel.gohh/opt/anaco
nda3/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

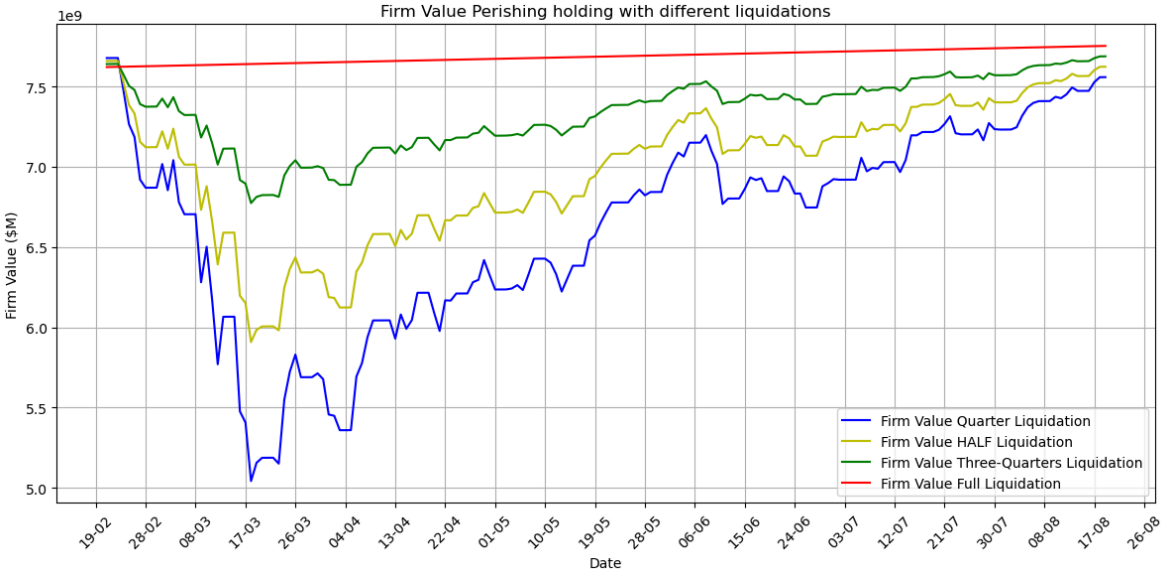
In [ ]:
```
def modifyNumShares(share_tracker, constant, soldShares):
    if (constant == QUARTER_LIQUIDATION):
        liquidation_fact = QUARTER
    elif (constant == HALF_LIQUIDATION):
        liquidation_fact = HALF
    elif (constant == THIRD_QUARTILE_LIQUIDATION):
        liquidation_fact = THIRD_QUARTER
    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
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': 30116
96, 'HHH': 4790127, 'A': 6605820, 'FNMA': 98253384, 'FMCC': 54007893}
Shares Remaining
{'CMG': 21547866, 'HLT': 2639201, 'LOW': 2153303, 'QSR': 3867045, 'BRK-B': 100389
8, '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': 4015
594, '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.

```
OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
SOFTWARE.
```