Prof. Lorenzo Naranjo Spring 2026

Homework 3

This **group** assignment is due no later than Sunday 2/22/2026 at 11:59pm CST. The assignment must be produced in Python as a Jupyter notebook, which you will save as an HTML file and upload to Canvas. Format your Jupyter notebook **professionally**, as the presentation of your results will count towards your grade in this assignment. Your deliverable will be an HTML file generated from a Jupyter notebook.

Make sure your code is well-written and easy to read. Use small blocks of code with clear written explanations in between, and use brief comments in your code to explain what each part does. You can use the code snippets provided in this assignment as a starting point, but you should feel free to modify and expand upon them as needed to complete the assignment.

In this assignment, you will implement a minimum variance portfolio strategy and backtest it on the CRSP universe of stocks. You will then compare the performance of your strategy to the CRSP value-weighted market index and compare its relative performance to the Fama-French factor model.

Getting the Data

The following code snippet shows how to download the CRSP monthly stock data from WRDS and save it as a parquet file. It also downloads the CRSP value-weighted market index returns. You can run this code snippet in a Jupyter notebook to get the data you need for this assignment. Please make sure to create a WRDS account using the instructions I posted on Canvas if you don't have one already.

```
import pandas as pd
import wrds

conn = wrds.Connection()

crsp_m = conn.raw_sql("""
select a.permno, a.date, b.shrcd, b.exchcd, a.ret, a.vol, a.shrout, a.prc
from crsp.msf as a
left join crsp.msenames as b
  on a.permno = b.permno
 and b.namedt <= a.date
 and a.date <= b.nameendt
where a.date between '1963-01-01' and '2025-12-31'
  and b.exchcd between 1 and 3
  and b.shrcd between 10 and 11
""")

crsp_m['date'] = pd.to_datetime(crsp_m['date'])

crsp_m.to_parquet('crsp_msf_1963_2025.parquet', index=False)

crsp_vw = conn.raw_sql("""
    select date, vwretd
    from crsp.msi
    where date between '1963-01-01' and '2025-12-31'
""")

crsp_vw['date'] = pd.to_datetime(crsp_vw['date'])
crsp_vw['vwretd'] = pd.to_numeric(crsp_vw['vwretd'], errors='coerce')

crsp_vw.to_parquet('crsp_msi_vwretd_1963_2025.parquet', index=False)
```

Minimum Variance Portfolio

The code below implements a minimum variance portfolio strategy using projected gradient descent. The `backtest_minvar_longonly` function backtests the strategy on the CRSP universe of stocks, using a rolling window of 60 months to estimate the covariance matrix and selecting the top 50 stocks by market capitalization. The function returns a pandas Series of the portfolio returns. Note that the code requires you to have the CRSP data saved as a parquet file in the same directory as your Jupyter notebook. You can adjust the parameters of the backtest (e.g., window size, number of stocks) as needed.

You can choose to run this code on the WRDS server, Google Colab, or your local machine, depending on your preference and computational resources. Just make sure to have the necessary libraries installed and the CRSP data file accessible.

This implementation imposes **short-sale constraints**. Specifically, the optimizer projects candidate weights onto the unit simplex (`proj_simplex`), which enforces nonnegative portfolio weights that sum to one. As a result, each position is long-only ($w_i \geq 0$) and the portfolio is fully invested ($\sum_i w_i = 1$) at every rebalancing date.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf

crsp_m = pd.read_parquet('crsp_msf_1963_2025.parquet')

start_date = '1963-01-01'
df = crsp_m
convert_dict = {'permno': int,
                'shrcd': int,
                'exchcd': int,
                'shrout': int
                }
df = df.astype(convert_dict)
df['date'] = pd.to_datetime(df['date'])
df = df.loc[df['date'] >= start_date]
df['mcap'] = abs(df['prc']) * df['shrout'] / 1000000

rets = df.pivot(index='date', columns='permno', values='ret').sort_index()
mcaps = df.pivot(index='date', columns='permno', values='mcap').reindex(rets.index)

def proj_simplex(v):
    u = np.sort(v)[::-1]
    cssv = np.cumsum(u)
    rho = np.nonzero(u * np.arange(1, len(v) + 1) > (cssv - 1))[0][-1]
    theta = (cssv[rho] - 1) / (rho + 1)
    w = np.maximum(v - theta, 0.0)
    return w / w.sum()

def minvar_longonly_pgd(S, tol=1e-4, max_iters=10000, ridge=1e-8):
    S = 0.5 * (S + S.T) + ridge * np.eye(S.shape[0])
    n = S.shape[0]
    w = np.full(n, 1/n)
    L = np.linalg.eigvalsh(S).max()
    eta = 1.0 / L

    for _ in range(max_iters):
        w_new = proj_simplex(w - eta*(S @ w))
        if np.linalg.norm(w_new - w) < tol:
            return w_new
        w = w_new
    return w

def backtest_minvar_longonly(rets, mcaps, window=60, top_n=50, min_obs=48):
    dates = rets.index
    port = {}

    for t in range(window, len(dates) - 1):
        dt = dates[t]
        dt_next = dates[t + 1]

        elig = (
            mcaps.loc[dt]
            .dropna()
            .sort_values(ascending=False)
            .head(top_n)
            .index
        )

        X = rets.loc[dates[t-window:t-1], elig]
        keep = X.count() >= min_obs
        X = X.loc[:, keep]
        S = X.cov().to_numpy()
        w = pd.Series(minvar_longonly_pgd(S), index=X.columns)

        r_next = rets.loc[dt_next, X.columns].dropna()
        if r_next.empty:
            continue
        w_valid = w.reindex(r_next.index)
        port[dt_next] = float(np.dot((w_valid / w_valid.sum()).to_numpy(), r_next.to_numpy()

    return pd.Series(port, name='minvar_ret')

port_rets = backtest_minvar_longonly(rets, mcaps, top_n = 100)
```

Questions for the Assignment

1. Analyze the performance of the minimum variance portfolio strategy starting in 1965, 1985, and 2005 using the top 100, 500, and 1000 stocks ranked by market cap. Compare the cumulative returns of the strategy to the CRSP value-weighted market index for each starting date. Plot the cumulative returns and discuss any differences in performance across the different time periods and universe of selected stocks.

2. Compute summary statistics and the Sharpe ratio of the minimum variance portfolio strategy for each starting date and compare it to the Sharpe ratio of the CRSP value-weighted market index for the different periods and universe of selected stocks. Discuss the risk-adjusted performance of the strategy relative to the market index.

3. Estimate the Fama-French five factor model using the minimum variance portfolio returns as the dependent variable and the Fama-French factors as independent variables. Use the Fama-French factor data available on Kenneth French's website as I showed you in class. Report the estimated coefficients, their statistical significance, and interpret the results in terms of the factor exposures of the minimum variance portfolio.

4. Using the results from the Fama-French factor model estimation, compute the alpha of the minimum variance portfolio strategy and test whether it is statistically different from zero.

Discuss the economic significance of the alpha and what it implies about the performance of the strategy after controlling for common risk factors.

5. Comment on the limitations of the minimum variance portfolio strategy and discuss potential improvements or alternative approaches that could be explored in future research. Consider issues such as estimation error, transaction costs, and the stability of the covariance matrix over time.

6. Explain in detail the optimization strategy used in this assignment to obtain constrained portfolio weights. In your explanation, describe the projected gradient descent update step, the simplex projection, and why this procedure enforces the long-only and fully invested constraints.