Project for Course: Case Studies in Machine Learning and Finance

Comparing Stability of CAPM Beta and ERoD Beta Using Rolling Window Test

Wenbo Gao

[1]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA, 11794

wenbo.gao@stonybrook.edu

Abstract

This project explores the stability of Capital Asset Pricing Model (CAPM) beta and Expected Regret of Drawdown (ERoD) beta by applying the rolling window test across two different time periods: from January 1, 2000, to January 1, 2024, and during the financial crisis from January 1, 2007, to January 1, 2009. Both CAPM beta and ERoD beta are crucial in measuring risk and the return of the Capital Asset Pricing Model, with ERoD beta incorporating a drawdown based adjustment. The analysis is conducted using the data from SP&500 with the selected notable stocks from different industries. This project contributes to the understanding of how CAPM beta and ERoD beta performs under different market trends and provides the insights of investment strategies and portfolio management.

1 Introduction

1.1 Problem Statement

This project will examine the stability of CAPM beta and ERoD beta through the rolling window test. Since ERoD beta serves the same function as the CAPM beta in the CAPM with drawdown measure, the beta with more stability will provide us with a more stable measure for CAPM or CAPM with drawdown measure. We will be comparing the beta for 2 time periods. Period 1: from 2000/01/01 to 2024/01/01. Period 2: 2007/01/01 to 2009/01/01. The goal of this project is to identify the stability of both betas in a long time period and in a time period where the market is going down such as financial crises.

1.2 Drawdown

Drawdown is a key factor in portfolio management, it measures the difference between the most recent peak of the portfolio's cumulative returns and the current portfolio's cumulative return. The definition of drawdown (Ding and Uryasev (2022)) can be written as:

Suppose that $\{r_t\}_{1\le t\le T}$ is a sample path of scalar returns of some instrument. Let us denote:
$\{w_t\}_{1\le t\le T}$ = vector of uncompounded cumulative returns.

$$w_t = \sum_{v=1}^{t} r_v , \qquad 1\le t\le T .$$

$\{d_t\}_{1\le t\le T}$ = vector of drawdowns,

$$d_t = max_{1\le v\le t} \{w_v\} - w_t , \qquad 1\le t\le T$$

For every time moment t, d_t is the difference between the previous cumulative return peak and the current cumulative return.

1.3 Expected Regret of Drawdown (ERoD)

In this project, we consider using the Expected Regret of Drawdown (ERoD) as the drawdown. The definition of ERoD (Ding and Uryasev (2022)) can be written as:

Let's denote by $w(x)$ the vector of cumulative returns of a portfolio with weights vector x and D($w(x)$) be the random drawdown value for the portfolio x.

$$ERoD_{\varepsilon}(w(x)) = E[(D(w(x))-\varepsilon)^{+}]$$

If we want to calculate the average of all positive drawdowns, we can set ε to a sufficiently small threshold. In this project, we will be looking at all positive drawdowns and drawdowns that are equal to or larger than 0%, $\varepsilon = 0$.
$ERoD_{\varepsilon}(w(x))$ with threshold ε can be calculated as follows:
Let's denote

- x = $(x^1, \ldots, x^I)$ = vector of weights for n assets in the portfolio;
- $(w^1_{st}, \ldots, w^I_{st})$ = vector of uncompounded cumulative returns of portfolio assets at time moment t

on scenario s;

- p_s probability of the scenario (sample path of returns of securities);
- $w_{st}(x) \; = \; \sum_{i=1}^{I} w_{st}^{i} x^{i}$ = cumulative portfolio return at time moment t on scenario s;
- $w(x)$ = vector of cumulative portfolio returns with components $w_{st}(x)$, s=1, … , S; t=1, … , T;

- $d_{st}(x) \; = \; max_{1 \le v \le t}\{w_{sv}(x)\} - w_{st}(x)$ = drawdown of portfolio at time t on sample path s.

$$ERoD_{\varepsilon}(w(x)) \; = \; \frac{1}{T}\sum_{s=1}^{S}\sum_{t=1}^{T} p_s(d_{st}(x) - \varepsilon)^{+} \quad .$$

This project considers a single path, in this case, path s =1 and p_s = 1.

1.4 CAPM and CAPM With Drawdown Measure

The Capital Pricing Model (CAPM) is a financial model that estimates the expected return on an investment based on its risk relative to the overall market, based on the Module_3_Slides (Uryasev), the definition of CAPM and CAPM beta can be written as:

$$E_{rj} = r_0 + \beta_j(E_{rM} - r_0), \quad \beta_j \; = \; \frac{cov(r_j, r_M)}{\sigma_M^{\;2}}$$

- E_{rj} = expected return of asset j;
- r_0 = risk-free rate;
- β_j = beta of asset j;
- E_{rM} = expected return of market;
- $E_{rM} - r_0$ = market risk premium.

With the measurement of ERoD, the CAPM can be expressed in the drawdown measure, the definition of CAPM with drawn down measure ((Ding and Uryasev (2022)) can be written as:

$$\sum_{s=1}^{S} p_s w_{sT}^{i} = \beta^{i}_{ERoD} \sum_{s=1}^{S} p_s w_{sT}^{M} \quad , \qquad \hat{\beta}^{i}_{ERoD} = \frac{\frac{1}{T}\sum_{s=1}^{S}\sum_{t=1}^{T} p_s q_{st}^{*}(w^{i}_{s,\tau(s,t)} - w^{i}_{st})}{\overline{E}_{\epsilon}(w^{M})}$$

- $\overline{E}_{\epsilon}(w^{M}) = \frac{1}{T}\sum_{s=1}^{S}\sum_{t=1}^{T} p_s q_s^{*}(w^{M}_{s,\tau(s,t)} - w^{M}_{st});$
- β^{i}_{ERoD} = ERoD Beta relating the total expected cumulative return, $\sum_{s=1}^{S} p_s w_{sT}^{M}$, of the optimal portfolio (market) and total expected cumulative return, $\sum_{s=1}^{S} p_s w_{sT}^{i}$;
- $\tau(s, t) = max\{K | 1 \le k \le t, \; w_{sk}^{M} = max_{1 \le \zeta \le t} w_{s\zeta}^{M}\};$

- $d_{st}^{M} = w_{s,\tau(s,t)}^{M} - w_{st}^{M}$ = drawdowns of the optimal portfolio;
- $q_{st}^{*} = 1(d_{st}^{M} \geq \epsilon)$ =indicator function which is equal to 1 for $d_{st}^{M} \geq \epsilon$, and 0 otherwise.

In this project, we consider a single path:

$$w_{T}^{i} = \beta_{ERoD}^{i} w_{T}^{M} , \quad \widehat{\beta}_{ERoD}^{i} = \frac{\frac{1}{T}\sum_{t=1}^{T} p_{s}(w_{\tau(t)}^{i} - w_{t}^{i})}{\overline{E}_{\epsilon}(w^{M})}$$

- $\overline{E}_{\epsilon}(w^{M}) = \frac{1}{T}\sum_{t=1}^{T} q_{st}^{*}(w_{\tau(t)}^{M} - w_{t}^{M})$ = ERoD with threshold ϵ for return w^{M};
- $q_{st}^{*} = 1(d_{t}^{M} \geq \epsilon) =$ indicator for drawdowns $d_{t}^{M} = w_{\tau(t)}^{M} - w_{t}^{M}$;
- $\tau(t) = = max\{K | 1 \leq k \leq t,\ w_{k}^{M} = max_{1 \leq \zeta \leq t} w_{\zeta}^{M}\}$.

1.5 CAPM Beta and ERoD Beta

CAPM beta for a security relates its expected return to the market's expected excess return over the risk-free rate, the ERoD Beta relates the average instrument losses during market drawdowns exceeding threshold over average market drawdowns exceeding threshold. In the CAPM and CAPM with drawdown measure, CAPM beta and ERoD Beta they both serve as the estimator. In this project, we will examine their stability and test which beta is more stable during different time periods.

2 Methodology

The approach to the solution is completely done in python with libraries: yfinance, pandas, numpy, and matplotlib. The coding will be in general, we will be able to change the starting time, ending time, tickers, threshold, and rolling window size easily by changing the content of the variables since we are going to repeat the process for different time periods and different thresholds to compare the stability of CAPM beta and ERoD beta through the rolling window test. The rest of the coding can be kept the same for two different periods. The tickers that we are using are: 'AAPL', 'JNJ', 'PG', 'XOM', 'JPM', 'KO', 'MSFT', 'UNH', 'MMM','^GSPC', the '^GSPC' will serve as the market in this project.

Steps for rolling window test for CAPM beta:

- Step1: Import all the necessary libraries: yfinance, pandas, and matplotlib;
- Step 2: Define tickers and choose the tickers that we want to use for comparing the stability of CAPM beta and ERoD beta;
- Step 3: Download the data with the time periods that we want (starting time, ending time) to consider for comparing the stability of CAPM beta and ERoD beta;
- Step 4: Extract the data for adjusted closing price of the tickers;
- Step 5: Define the function for calculating CAPM Beta (1.4 CAPM Beta and ERoD Beta).
- Step 6: Create an empty dataframe.
- Step 7: Perform the rolling window test for CAPM Beta with a specific rolling window size that we want to use for the test.
- Step 8: Save the results from step 7 to the empty data frame created in step 6.
- Step 9: Plot the graphs of rolling window tests for CAPM beta.
- Step 10: Calculate the variances for CAPM betas

Steps for rolling window test for ERoD beta:
- Step1: Import all the necessary libraries: yfinance, numpy, pandas, and matplotlib;
- Step 2: Define tickers and choose the same tickers that we decided for CAPM beta.
- Step 3: Download the data with the time periods that we want (starting time, ending time) to consider for comparing the stability of CAPM beta and ERoD beta;
- Step 4: Extract the data for adjusted closing price of the tickers;
- Step 5: Define the function for calculating ERoD Beta (1.4 CAPM Beta and ERoD Beta) with the threshold that we want to calculate.
- Step 6: Create an empty dataframe.
- Step 7: Perform the rolling window test for ERoD beta with the same rolling window size for CAPM beta.
- Step 8: Save the results from step 7 to the empty data frame created in step 6.
- Step 9: Plot the graphs of rolling window tests for ERoD beta.
- Step 10: Calculate the variance for ERoD betas.

The next step for this project is to focus on comparing two betas with visualization.
- Plot stock's CAPM beta and ERoD beta separately.
- Combine all the rolling CAPM beta in one graph.
- Combine all the rolling ERoD beta in one graph.
- Combine each ticker's CAPM beta and ERoD in one graph.

We are going to repeat all the steps for different time periods and thresholds.

In the first run we are going to compare the stability of CAPM beta and ERoD beta from 2000-01-01 to 2024-01-01, with threshold $\varepsilon=0$ and rolling window size of 252.
In the second run, we are going to change the time period from 2007-01-01 to 2009-01-01, with the same threshold and rolling window size. It is important to see how the betas perform over a long time period and during time periods where markets are going down.

3 Experiment / Results

Provides data, graphical and statistical results for rolling window tests of CAPM beta and ERoD beta.

3.1 Data

The initial data is sp&500 index with tickers: 'AAPL', 'JNJ', 'JPM', 'KO', 'MMM', 'MSFT', 'PG', 'UNH', 'XOM','^GSPC', from 2000/01/01 to 2024/01/01.

The data includes the dates, Open, High, Low, Close, Adjusted Close, and Volume for the tickers. For the rolling window test of CAPM beta and ERoD beta, we are only considering using Adjusted Close Price with dates for the tickers, therefore, we are going to extract the Adjusted Close Price with corresponding dates and tickers and drop other data. We are going to extract it two times, the first time is from 2000/01/01 to 2024/01/01, and the second time is from 2007/01/01 to 2009/01/01.
Figure 1 is the extracted data of adjusted closing price for the tickers from 2000/01/01 to 2024/01/01.
Figure 2 is the extracted data of adjusted closing price for the tickers from 2007/01/01 to 2009/01/01.

(Figure 1: Adjusted Closing Price for the Tickers From 2000/01/01 to 2024/01/01)

```
Ticker             AAPL         JNJ         JPM         KO        MMM  \
Date
2000-01-03    0.846127   24.698620   23.776054  14.434139  17.557297
2000-01-04    0.774790   23.794405   23.254374  14.450142  16.859659
2000-01-05    0.786128   24.045578   23.110825  14.578156  17.348019
2000-01-06    0.718097   24.799088   23.438942  14.594159  18.743296
2000-01-07    0.752113   25.854008   23.869564  15.554296  19.115364
...                ...         ...         ...        ...        ...
2023-12-22  193.353287  154.288544  165.409225  57.857216  87.167412
2023-12-26  192.803986  154.963409  166.387451  58.095314  88.626625
2023-12-27  192.903839  155.171844  167.385437  58.244122  89.143089
2023-12-28  193.333298  155.400101  168.274734  58.283806  89.921883
2023-12-29  192.284637  155.558899  168.077118  58.462376  89.618561

Ticker             MSFT          PG         UNH         XOM        ^GSPC
Date
2000-01-03   36.065571   28.069103    5.400593   18.328701  1455.219971
2000-01-04   34.847260   27.529005    5.331516   17.977631  1399.420044
2000-01-05   35.214703   27.005247    5.318956   18.957701  1402.109985
2000-01-06   34.035065   28.249125    5.513628   19.937754  1403.449951
2000-01-07   34.479851   30.507757    6.160444   19.879248  1441.469971
...                ...         ...         ...         ...          ...
2023-12-22  373.888580  143.442017  518.266907  100.971611  4754.629883
2023-12-26  373.968445  144.093689  517.988037  101.199486  4774.750000
2023-12-27  373.379547  144.212158  520.737183  100.723907  4781.580078
2023-12-28  374.587280  143.886322  522.838928   99.267441  4783.350098
2023-12-29  375.345886  144.686081  524.402710   99.059380  4769.830078

[6037 rows x 10 columns]
```

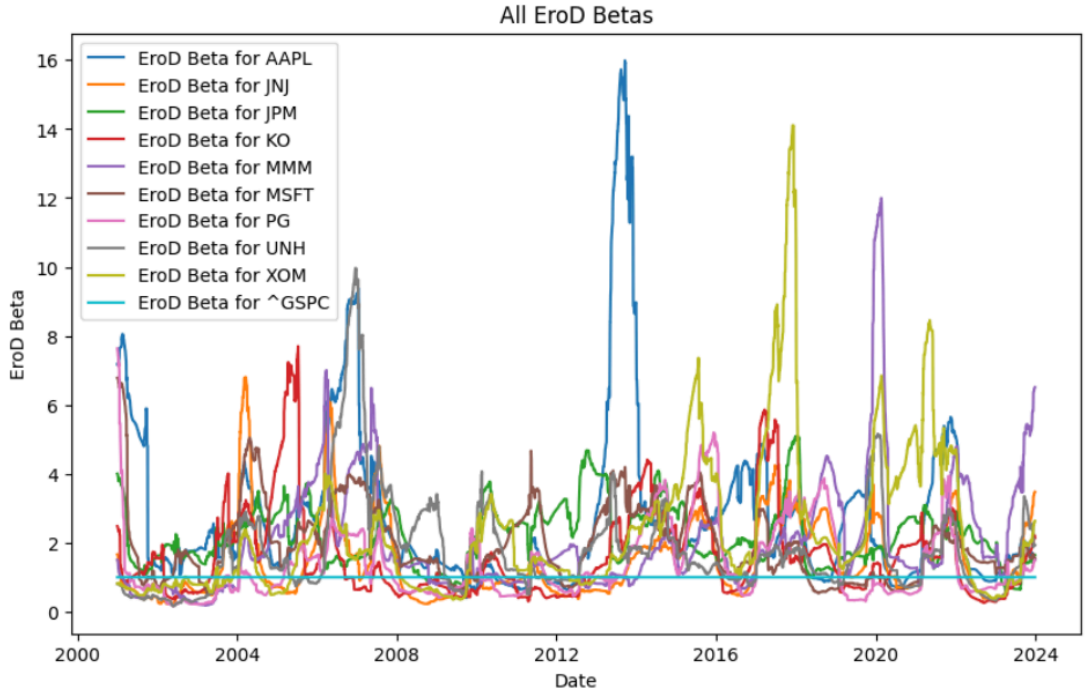
(Figure 2: Adjusted Closing Price for the Tickers From 2007/01/01 to 2009/01/01)

```
Ticker          AAPL        JNJ        JPM         KO        MMM       MSFT  \
Date
2007-01-03  2.533751  40.151031  30.657494  14.311197  34.797928  21.397421
2007-01-04  2.589991  40.652912  30.734028  14.317080  34.660091  21.361603
2007-01-05  2.571546  40.284061  30.478922  14.216932  34.424438  21.239777
2007-01-08  2.584245  40.217552  30.580956  14.308254  34.500000  21.447590
2007-01-09  2.798918  40.066372  30.453409  14.320037  34.540031  21.469086
...              ...        ...        ...        ...        ...        ...
2008-12-24  2.571244  37.360832  20.186495  13.710385  26.233116  14.180634
2008-12-26  2.594525  37.360832  20.152681  13.859618  26.346539  14.151044
2008-12-29  2.618714  37.099251  20.139156  13.806764  26.171669  14.025287
2008-12-30  2.609038  37.749996  20.970968  13.968424  27.022490  14.306395
2008-12-31  2.580616  38.171059  21.322617  14.074129  27.197371  14.380359

Ticker             PG        UNH        XOM        ^GSPC
Date
2007-01-03  39.077045  42.389317  40.657738  1416.599976
2007-01-04  38.780380  42.663479  39.895157  1418.339966
2007-01-05  38.447380  42.373207  40.180443  1409.709961
2007-01-08  38.532139  42.994076  39.856770  1412.839966
2007-01-09  38.435265  42.478020  39.549538  1412.109985
...               ...        ...        ...          ...
2008-12-24  38.216614  20.962343  43.098732   868.150024
2008-12-26  38.305222  21.236891  43.900646   872.799988
2008-12-29  38.096375  20.647425  44.372692   869.419983
2008-12-30  38.678574  21.834433  44.696873   890.640015
2008-12-31  39.121567  21.479141  45.402100   903.250000

[504 rows x 10 columns]
```

3.2 Graphical Results

Figure 3 is all the tickers' rolling ERoD betas from 2000/01/01 to 2024/01/01.
Figure 4 is all the tickers' rolling CAPM betas from 2000/01/01 to 2024/01/01.

(Figure 3)



(Figure 4)

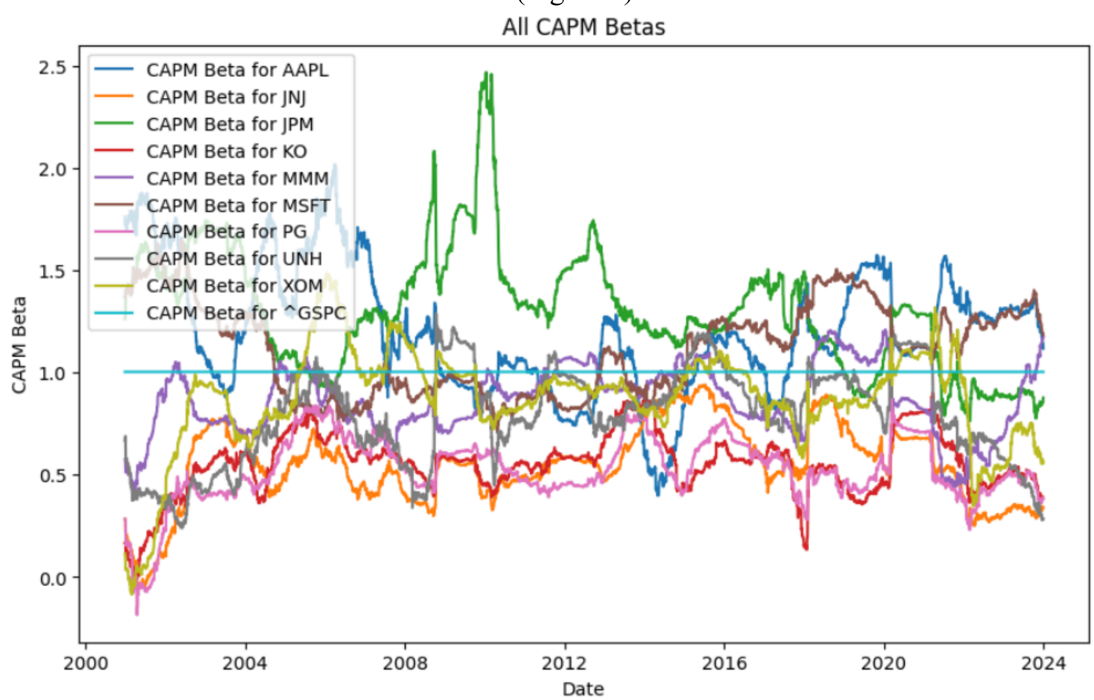
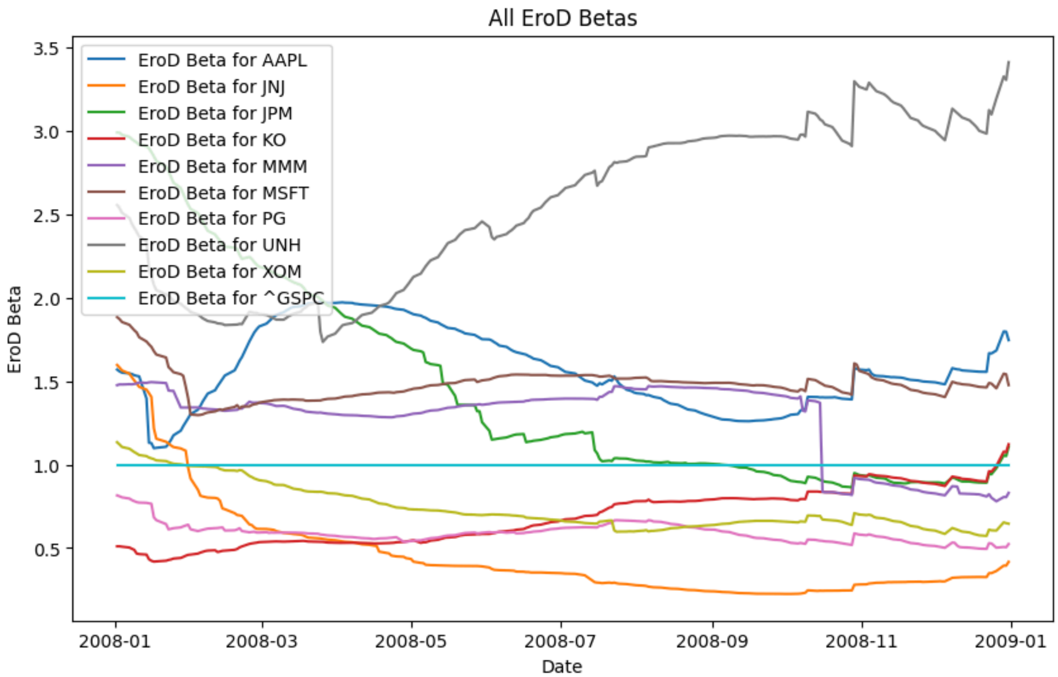
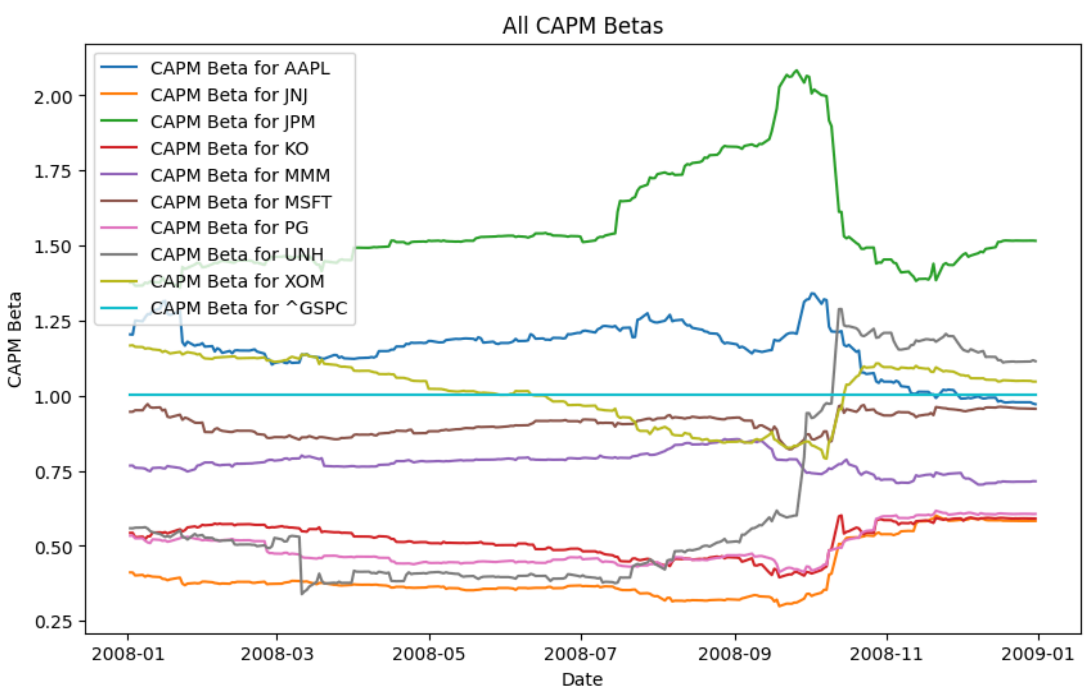


Figure 5 is all the tickers' rolling ERoD betas from 2007/01/01 to 209/01/01.
Figure 6 is all the tickers' rolling CAPM betas from 2007/01/01 to 2009/01/01.

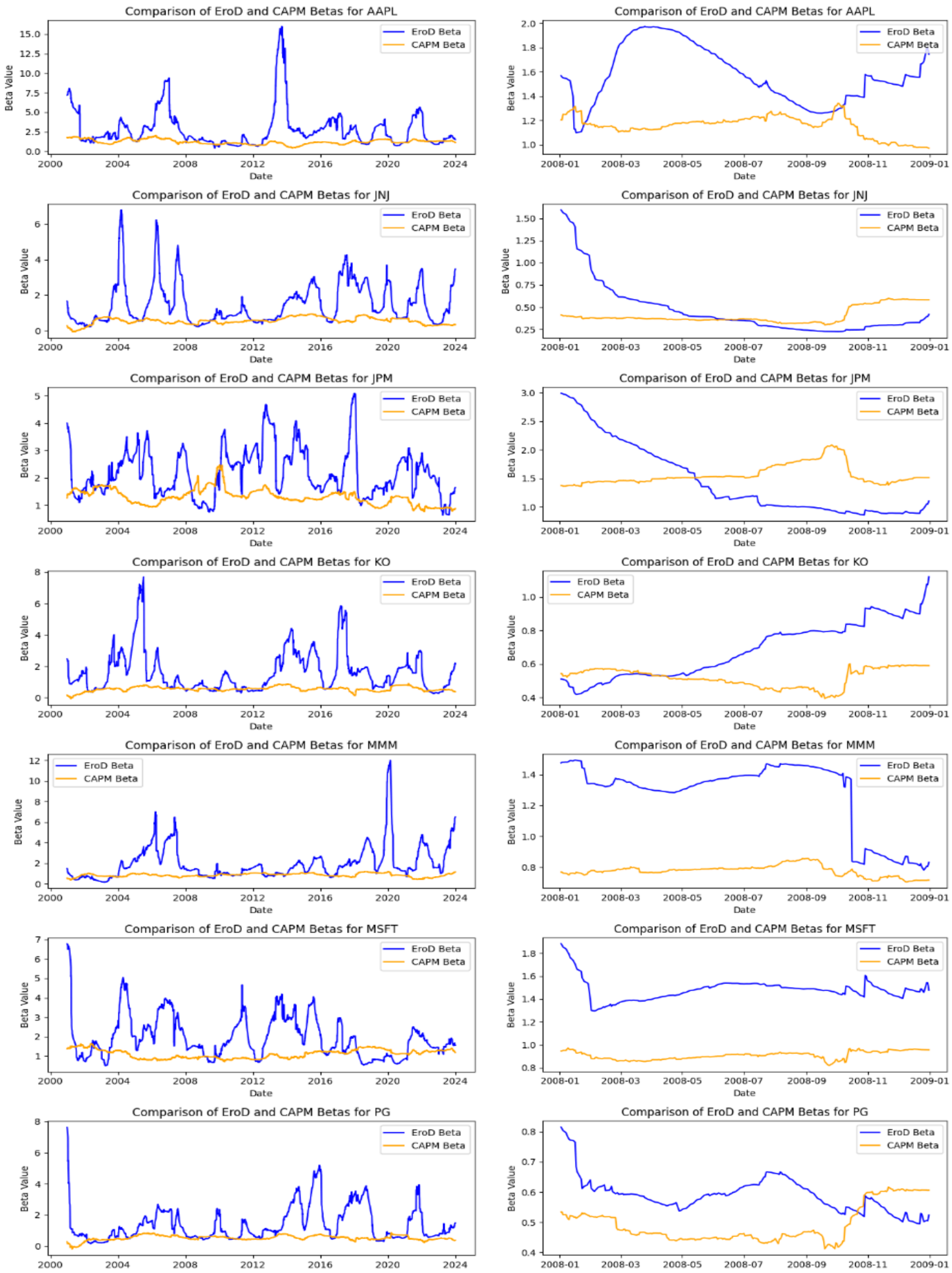
(Figure 5)



(Figure 6)



The figures on the left side is the comparison of rolling CAPM beta with rolling ERoD beta from 2000/01/01 to 2024/01/01 for some notable tickers that we computed.
The figures on the right side is the comparison of rolling CAPM beta with rolling ERoD beta from 2007/01/01 to 2009/01/01 for some notable tickers that we computed.



3.3 Statistical Results

Table 1 provides the variances for rolling CAPM beta and ERoD beta from 2000/01/01 to 2024/01/01
Table 2 provides the variances for rolling CAPM beta and ERoD beta from 2007/01/01 to 2009/01/01

(Table 1)

Ticker	CAPM Beta Variance	EroD Beta Variance
AAPL	0.094103	5.880929
JNJ	0.038017	1.331109
JPM	0.084197	0.750909
KO	0.022278	1.64258
MMM	0.027663	2.597611
MSFT	0.048648	1.181238
PG	0.026244	1.161085
UNH	0.053755	2.100743
XOM	0.060029	4.219792
^GSPC	0.0	0.0

(Table 2)

Ticker	CAPM Beta Variance	EroD Beta Variance
AAPL	0.007291	0.05643
JNJ	0.007976	0.091628
JPM	0.030241	0.393462
KO	0.003021	0.027011
MMM	0.001281	0.051027
MSFT	0.001224	0.008858
PG	0.003598	0.00352
UNH	0.095129	0.236647
XOM	0.010582	0.019145
^GSPC	0.0	0.0

4 Conclusion

In this project, we conducted the rolling window test for CAPM beta and ERoD beta for a long time period and the time period where the market is going down. It is important to see how stable the betas are during different market trends since the ERoD beta only considers the time periods where market has drawdown but the CAPM beta considers all time intervals.

In the rolling window test, we used 9 notable stocks from different industries and the SP&500 index as the market in two betas' calculations. In addition, the rolling window is 252 which is approximately the number of trading days in one year for two betas' rolling window tests.

Based on the rolling window tests for CAPM beta and ERoD beta during the time period from 2000/01/01 to 2024/01/01, CAPM beta performs more stable than ERoD beta. In the time periods from 2007/01/01 to 2009/01/01 which is the financial crisis, the variances of both betas decreased, especially for ERoD beta. The variance of ERoD beta decreased on a much larger scale compared to the CAPM beta. It suggests that the ERoD beta is more stable during the time periods when the market is going down compared to a longer time period. However, during this period, the variance of CAPM beta is still smaller than the variance of ERoD beta even though the variance of CAPM beta did not decrease that much. Overall, this project suggests that the CAPM beta provides a more stable relation between the stock's return and the market's return and the CAPM beta is a more stable estimator compared to ERoD beta in the CAPM.

References

Ding, R., & Uryasev, S. (2022). Drawdown beta and portfolio optimization. *Quantitative Finance*, *22*(7), 1265–1276. https://doi.org/10.1080/14697688.2022.2037698

Uryasev, S. *Module 3 Slides:Calibrating Risk Preferences* [PowerPoint slides]. Stony Brook University.

Appendices

Coding in Python for this project

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
#data
tickers = ['AAPL', 'JNJ', 'JPM', 'KO', 'MMM', 'MSFT', 'PG', 'UNH', 'XOM','^GSPC']
data = yf.download(tickers, start="2000-01-01", end="2024-01-01")
print(data)
```

```
# Adjusted Closing Price
adj_close = data['Adj Close']
print(adj_close)
```

```
# Function to calculate ERoD Beta
def Calculate_EroD_Beta(Market_Return_Series, Stock_Return_Series):
    #daily returns
    stock_dreturns = Stock_Return_Series.pct_change()
    market_dreturns = Market_Return_Series.pct_change()

    #Uncompounded cumulative returns
    stock_ucr = stock_dreturns.cumsum()
    market_ucr = market_dreturns.cumsum()

    #Calculate 2 drawdown series
    market_drawdowns = market_ucr.cummax() - market_ucr
    stock_drawdowns = stock_ucr.cummax() - stock_ucr

    #compute and return beta (selected when drawdown exists for market first)
    threshold = market_drawdowns > 0
    market_drawdowns_selected = market_drawdowns[threshold]
    stock_drawdowns_selected = stock_drawdowns[threshold]
    beta = stock_drawdowns_selected.mean() / market_drawdowns_selected.mean()
    #return beta
    return beta
```

```python
# Function to calculate CAPM Beta
def Calculate_CAPM_Beta(Market_Return_Series, Stock_Return_Series):
    # Calculate daily returns
    stock_returns = Stock_Return_Series.pct_change().dropna()
    market_returns = Market_Return_Series.pct_change().dropna()

    # Align the lengths of the return series
    common_dates = stock_returns.index.intersection(market_returns.index)
    stock_returns = stock_returns.loc[common_dates]
    market_returns = market_returns.loc[common_dates]

    # Calculate covariance between stock returns and market returns
    covariance = np.cov(stock_returns, market_returns)

    # The covariance between the stock and the market is the [0,1] element
    cov_stock_market = covariance[0, 1]

    # Calculate the variance of the market returns
    var_market = np.var(market_returns)

    # Beta is the ratio of covariance to variance
    beta = cov_stock_market / var_market
    return beta
```

```python
# Rolling window size and creating a empty dataframe
window_size = 252
drawdown_betas = pd.DataFrame(index=adj_close.index[window_size-1:], columns=adj_close.columns)
capm_betas = pd.DataFrame(index=adj_close.index[window_size-1:], columns=adj_close.columns)
```

```python
#rolling window test
for ticker in adj_close.columns:
    for end_index in range(window_size - 1, len(adj_close)):
        start_index = end_index - window_size + 1
        current_date = adj_close.index[end_index]
        market_return_series = adj_close['^GSPC'][start_index:(end_index + 1)]
        stock_return_series = adj_close[ticker][start_index:(end_index + 1)]

        # Calculate and store the EroD Beta
        drawdown_betas.at[current_date, ticker] = Calculate_EroD_Beta(market_return_series, stock_return_series)

        # Calculate and store the CAPM Beta
        capm_betas.at[current_date, ticker] = Calculate_CAPM_Beta(market_return_series, stock_return_series)
```

```python
#plots for each stock's CAPM beta and ERoD beta

num_stocks = 9  # The number of stocks you want to plot.
fig, axes = plt.subplots(num_stocks * 2, 1, figsize=(10, 40))  # Adjust the size as needed

for i, ticker in enumerate(drawdown_betas.columns[:num_stocks]):
    # EroD Beta subplot
    axes[i * 2].plot(drawdown_betas.index, drawdown_betas[ticker], label=f'EroD Beta for {ticker}')
    axes[i * 2].set_title(f'EroD Beta for {ticker}')
    axes[i * 2].set_xlabel('Date')
    axes[i * 2].set_ylabel('EroD Beta')
    axes[i * 2].legend()

    # CAPM Beta subplot
    axes[i * 2 + 1].plot(capm_betas.index, capm_betas[ticker], label=f'CAPM Beta for {ticker}', color='orange')
    axes[i * 2 + 1].set_title(f'CAPM Beta for {ticker}')
    axes[i * 2 + 1].set_xlabel('Date')
    axes[i * 2 + 1].set_ylabel('CAPM Beta')
    axes[i * 2 + 1].legend()

plt.tight_layout()  # Adjust the layout to prevent overlap
plt.show()
```

```python
#Plots for all CAPM beta in one graph and all ERoD beta in one graph
num_stocks = len(drawdown_betas.columns)

# Plotting all EroD Betas in one graph
fig, ax1 = plt.subplots(figsize=(10, 6))
for ticker in drawdown_betas.columns[:num_stocks]:
    ax1.plot(drawdown_betas.index, drawdown_betas[ticker], label=f'EroD Beta for {ticker}')

ax1.set_title('All EroD Betas')
ax1.set_xlabel('Date')
ax1.set_ylabel('EroD Beta')
ax1.legend(loc='upper left')  # You may need to adjust the location of the legend
plt.show()

# Plotting all CAPM Betas in one graph
fig, ax2 = plt.subplots(figsize=(10, 6))
for ticker in capm_betas.columns[:num_stocks]:
    ax2.plot(capm_betas.index, capm_betas[ticker], label=f'CAPM Beta for {ticker}')

ax2.set_title('All CAPM Betas')
ax2.set_xlabel('Date')
ax2.set_ylabel('CAPM Beta')
ax2.legend(loc='upper left')  # You may need to adjust the location of the legend
plt.show()
```

```python
#Comparing each stock's CAPM beta and ERoD beta
num_stocks = len(drawdown_betas.columns)  # Assuming you want to plot for all stocks

fig, axes = plt.subplots(num_stocks, 1, figsize=(7, 3 * num_stocks))

if num_stocks == 1:
    axes = [axes]

# Loop through each ticker and plot both EroD and CAPM Betas on the same subplot
for i, ticker in enumerate(drawdown_betas.columns):
    axes[i].plot(drawdown_betas.index, drawdown_betas[ticker], label='EroD Beta', color='blue')
    axes[i].plot(capm_betas.index, capm_betas[ticker], label='CAPM Beta', color='orange')
    axes[i].set_title(f'Comparison of EroD and CAPM Betas for {ticker}')
    axes[i].set_xlabel('Date')
    axes[i].set_ylabel('Beta Value')
    axes[i].legend()

# Adjust the layout
plt.tight_layout()
plt.show()
```

```python
# Calculate variance for each column in both DataFrames
capm_variances = capm_betas.var()
erod_variances = drawdown_betas.var()

variances_df = pd.DataFrame({
    'CAPM Beta Variance': capm_variances,
    'EroD Beta Variance': erod_variances
})

# Print the resulting DataFrame
print(variances_df)
```

The data will be submitted in a spreadsheet to Brightspace.