Election Impact on Market Dynamics: A Volatility-Driven Portfolio

MF 703 – Programming for Mathematical Finance

Boston University

Jon Lopes, Brock Williams, Christian Archibald,
Aiden Perkins and Ryan Holzschuh

Professor Chris Kelliher, Professor Eugene Sorets

07 December, 2024

Abstract/Introduction

With the United States presidential elections becoming increasingly polarized, many citizens express concerns about the country's future. Gerardo Manzo, in his study, highlights how Italian financial markets respond to politically charged events.[1] This paper uses Manzo's analysis as motivation to analyze market volatility during United States election cycles and leverage this volatility to construct high-performing investment portfolios.

We begin by defining the election period: for this analysis, it includes the 26 weeks preceding an election and the 2 weeks following it. We then examine whether market volatility during these periods is significantly higher compared to similar timeframes in non-election years. Finally, we collect data on multiple assets and apply portfolio optimization techniques to explore potential performance improvements.

Assumptions

For simplicity, we make several assumptions for our data and portfolio construction. We discuss further that we assign specific weights to different assets based on constraints that we give our weight optimizer. We also assume zero transaction costs to the positions that we hold, because the main focus of this project is the weight optimization rather than the simulation of a portfolio. In particular, we assume costless rolling of futures contracts. We finally assume log-normality of returns for each asset class. More assumptions on the government bond data will be discussed in the Treasuries portion of the Financial Instruments section.

Hypothesis Testing

First, we have to check whether or not the volatility even has a statistically significant change over the election period. To check this, we are going to use Levene's Test for Equality of Variances. We are using this test, rather than a standard F-Test, because it is more robust in accounting for non-normal data. Looking at Figure 1, we graphed the density histogram of the returns during the election cycles against different PDFs, scaled with the sample mean and standard deviation of the returns. Looking at this, the data seems more Laplace distributed, when compared to the Normal, but it is very clearly not Cauchy distributed. Levene's Test has three different choices for the test statistic, but looking at the histogram, we will calculate ours using the mean. After running the test, we find that our test statistic is 1.722, and our p-value is 0.189.

[1] Manzo, Gerardo. "Quantifying Political Risk on Financial Markets—Italian Case Study."

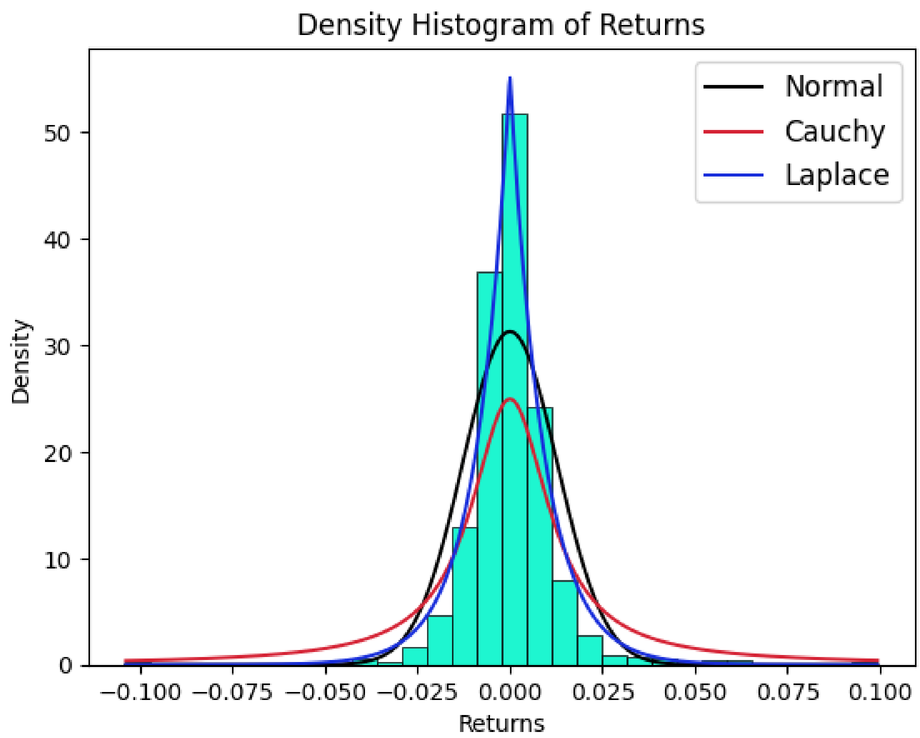


Figure 1: Density Histogram of Returns

Financial Instruments

For our portfolios, we decided to focus on Sector ETFs, (Futures), and U.S. Treasuries. We decided to diversify our portfolio amongst these assets due to the availability and simplicity of data.

Sector ETFs

We chose to include the following ETFs in our portfolio:

- XLB - Materials Select Sector SPDR Fund
- XLF - Financial Select Sector SPDR Fund
- XLP - Consumer Staples Select Sector SPDR Fund
- XLY - Consumer Discretionary Select Sector SPDR Fund
- XLE - Energy Select Sector SPDR Fund
- XLI - Industrial Select Sector SPDR Fund

- XLK - Technology Select Sector SPDR Fund
- XLU - Utilities Select Sector SPDR Fund
- XLV - Health Care Select Sector SPDR Fund

  The data for the ETFs was obtained from Yahoo Finance.[2] This is a daily data for April 2007 to December 2024. The data used from Yahoo Finance was the daily "Adj Close" for each ETF. This data was cleaned by forward filling the very few bits of missing data. This "Adj Close" price was then used in portfolio calculations.

Futures

We chose to include the following futures in our portfolio:

- ES1 - E-mini S&P 500 Futures
- DX1 - U.S. Dollar Index Futures
- SI1 - Silver Futures
- CL1 - Crude Oil Futures
- NG1 - Natural Gas Futures
- GC1 - Gold Futures

This data was obtained from Bloomberg.[3] This is daily data that goes from September 1997 to December 2024.The last price was used. This data was cleaned by forward filling the missing data and dropping days with negative values (CL1 was negative on April 20, 2024). This data was then used in the calculations for the portfolio.

Treasuries

To diversify our portfolio and include the risk-free rate, we decided to add government bonds to our portfolio. We originally collected data from a Bloomberg terminal, but due to the lack of data from multiple years, we decided to focus on the par yield data from `treasury.gov` instead.[4] We collected data points from January 1st 1992 up until November 26th, 2024.

We focused on the 6-month T-bill, 1-year T-bill, 2-year T-note, 3-year T-note, 4-year T-note, 5-year T-note, 7-year T-note, 10-year T-note, and 30-year T-bond due to the availability of data. Any data that were missing were interpolated using Panda's interpolation method.

[2] "Yahoo Finance - Stock Market Live, Quotes, Business & Finance News."
[3] Bloomberg Terminal. Historical Price Data for CL1, GC1, DX1, SI1, NG1.
[4] "Daily Treasury Par Yield Curve Rates." U.S. Department of the Treasury, 19 Nov. 2024,

Since bond data are quoted in yields rather than price, there was a bit of extra work to do after collecting the data. We assumed that treasuries were issued at par on the day we opened our portfolio positions. We then recalculated the price of each bond that was in our portfolio.

To recalculate the price of the bond each day, we first looked at the par yield for each government bond and then plotted them against their maturities for each day in our portfolio. We then interpolated the yield curve using the cubic spline method from Scipy. [5] From that, we got the yields for real or hypothetical bonds with maturities of $i/2$ years for $i \in \mathbb{Z}$ such that $1 \leq i \leq 30$, and bootstrapped the discount rates along each hypothetical or real maturity. [6]

We then used the spot rates to interpolate the spot curve. Let τ_i be the time till maturity, T, per 360 day year. If we let T_i be the the number of days till coupon payment i, then $\tau_i = T_i/360$. Since the equation of pricing a bond is

$$p = F\left(\sum_{i=1}^{2T} c_{i/2} d(i/2) + c_T d(T)\right),$$

Where $c_{i/2}$ is the coupon payment and $2d(t)^{\frac{1}{-2t}} - 2 = r_s(t)$. [7] The equation for recalculating the price (calculating the clean price) is similar:

$$p = F\left(\sum_{i=1}^{2T} c_{\tau_{i/2}} d(\tau_{i/2}) + c_{\tau_T} d(\tau_T)\right)$$

Since interpolations are never perfect estimations of data, we included some error handling in the code provided in the GitHub Repository.[8] If there was ever a time we got negative discount factors from the algorithms, we would search one day forward or one day backward for spot rates of the same tenor, since rates change very little on a daily basis.

Portfolio Construction

General Methodology

The first step for our portfolio construction was to find the daily logarithmic returns for each instrument within the portfolio. This was done using the following equation:

$$r_t = \ln\left(\frac{P_t}{P_t - 1}\right)$$

[5] scipy.interpolate.CubicSpline

[6] full calculation and algorithm can be found in the Treasuries section of the Appendix

[7] derivation of this equation can be found in the Appendix under the Treasuries section

[8] Lopes, Jon Lopes, et al. "JLOPES3/Mf703$_project : Repository for the Final Project for MF$703."

The next step was to find the expected annualized logarithmic return of each of the individual assets. To do this, we used the Capital Asset Pricing Model (CAPM). The following is the equation for the CAPM:

$$E[r] = (E[r_{market}] - r_f) + r_f$$

To use this model to find the expected return of an asset in the model, we needed to get each of the three parts of the equation. As a benchmark for the market, we used a different instrument for each instrument type.

For all ETFs, we used the ETF SPY as a proxy for the market. For all Futures, we used the ETF DBC as a proxy for the market. For all Treasuries, we used the ten year treasury as a proxy for the market.

To find beta, we used the following equation:

$$\beta = \frac{\text{Cov(log returns of asset, log returns of benchmark)}}{\text{Var(log returns of benchmark)}}$$

To find the expected returns for the market, we used the benchmark daily historical logarithmic returns and the following equation where N represents the number of datapoints:

$$E[r_{\text{benchmark}}] = \frac{252}{N} \sum_i r_{i,\text{benchmark}}$$

To find the risk free rate, we used the U.S. Treasury 6-month Yield to Maturity from the earliest date available. In order to use it in the CAPM equation with logarithmic returns, we took the natural logarithm of this value.

Using all of those parts, we calculated the expected annualized logarithmic return for each of the assets in the portfolio.

The next step is to look for the optimal weights of each asset in the portfolio. We used the portfolio with the maximum Sharpe Ratio to be the optimal portfolio. The following is the equation for the Sharpe ratio:

$$\text{Sharpe Ratio} = \frac{E[r] - r_f}{\sigma}$$

To find the expected annualized logarithmic return of the portfolio, we used the weighted sum of the expected annualized logarithmic return of each of the portfolio.
For the risk-free rate, we used the same value as we did for the CAPM equation above.
For the portfolio volatility, we used the following equation where w is the vector of weights of the assets and is the annualized covariance matrix of the daily logarithmic returns of all of the assets within the portfolio:

$$\sigma = \sqrt{w^t \Sigma w}$$

In order to find the weights of the portfolio with the maximum Sharpe Ratio, we needed to be able to find the weights of the minimum volatility portfolio for a specific target return. We used a Sequential Least Square Quadratic Programming optimizer to find the minimum volatility portfolio weights for a given target return. This was done in Python using SciPy's optimize module. We had the constraints that the weights had to add up to 1 for all portfolios. For the long only portfolios, we had a constraint that individual weights must be between 0 and 1 inclusive. For the other portfolios, we had a constraint that individual weights must be between -0.5 and 0.5 inclusive.

We did this for 100 different evenly-spaced out target returns starting at the lowest individual asset expected annualized return and ending with the highest individual asset expected annualized return. We calculated the Sharpe Ratios for each portfolio, and chose the weights of the portfolio with the maximum Sharpe Ratio as the optimal weights.

The last step was to test the portfolio weights on future data. For all of our portfolios, the weights were optimized using election period data from the 2008 election to the 2020 election. The tests were done on periods after 2020. The actual annualized logarithmic returns of the portfolios were calculated using the following equation:

$$r_{\text{portfolio annualized log return}} = \sum_j w_j \left(\frac{252}{N}\right) \left[\sum_i r_{i,\text{ individual asset j daily log return}}\right]$$

This actual return was compared to the original expected return to determine the success of the model.

Different Portfolios

We decided to use different portfolios to determine if our methodology of creating portfolios performed differently with different combinations of asset classes. We decided on eight different portfolios using different subsets of the instruments we described in the Instrument section:

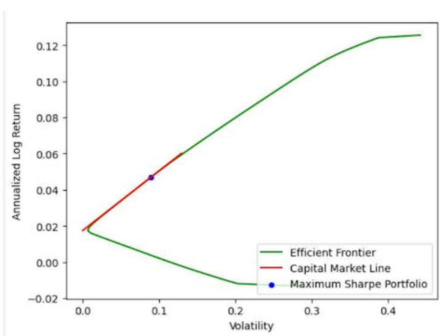
- All assets
- Futures Only
- ETFs Only
- Futures and ETFs
- Long Only All assets
- Long Only Futures Only

- Long Only ETFs Only
- Long Only Futures and ETFs

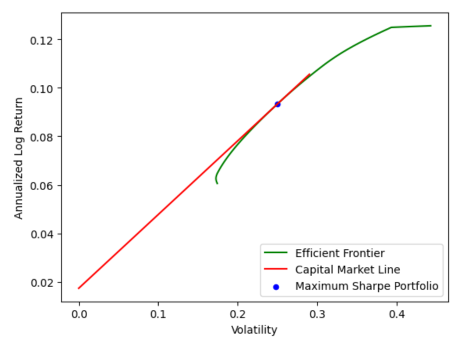
We also wanted to see if our portfolio performance varied if we restricted them to just long positions or allowed them to have short positions. This is why we have “Long Only” versions of the first four portfolios. The first four portfolios have individual weights constraints -0.5 and 0.5 inclusive, while the “Long Only” portfolios have individual weight constraints of 0 and 1 inclusive.

Results

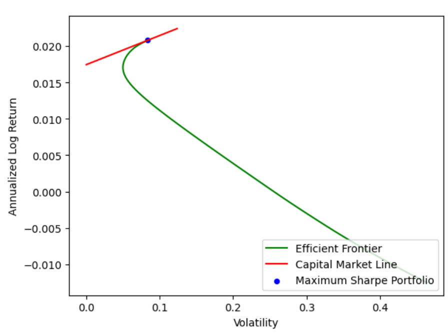
Using our weight optimization algorithm to build the portfolios based on historical data, we then evaluated our portfolios using common metrics in portfolio construction and tested how they would have performed in the 2024 election cycle. To begin with, graphs of the efficient frontiers of ‘Long Only’ portfolios can be seen below, with capital market line and MSR portfolios marked:



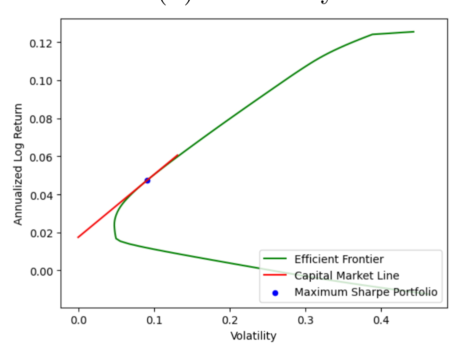
(a) All Assets



(b) ETF Only



(c) Long Only Futures



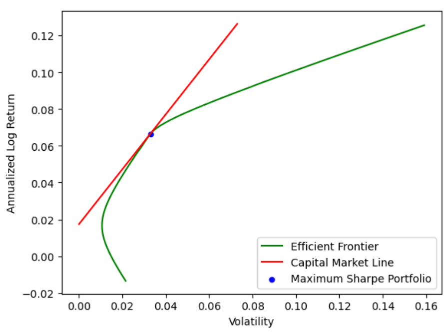
(d) ETF and Futures

Figure 2: Efficient Frontiers of Long Only Portfolios

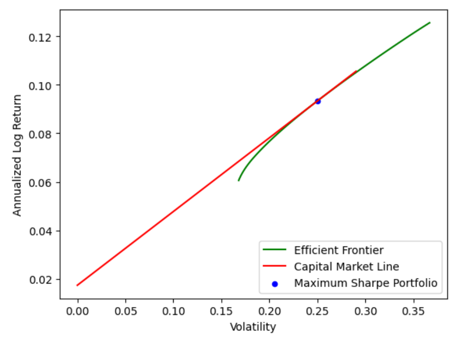
Below is a table showing metrics and results for these portfolios:

	All Assets	ETF Only	Long Only Futures	ETF and Futures
Sharpe Ratio	.33	.30	.04	.33
Exp. Return (%)	4.69	9.34	2.1	4.75
VaR (%)	14.7	41.1	13.8	15.0
ES (%)	18.5	51.6	17.3	18.8
Avg 2021-23 (%)	6.9	11.6	6.7	8.5
2024 Return (%)	9.0	18.5	2.1	9.0

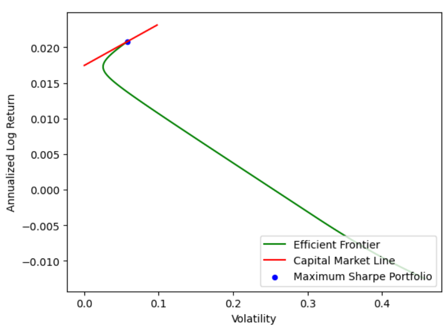
Next, we have our results for portfolios with shorting allowed. However, minimum and maximum weights for individual securities have been restricted to -0.5 and 0.5 respectively.



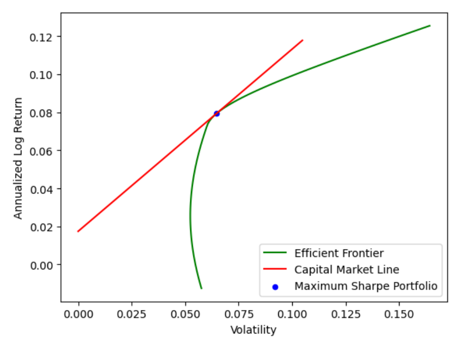
(a) All Assets



(b) ETF Only



(c) Only Futures



(d) ETF and Futures

Figure 3: Efficient Frontiers of Portfolios that Allow for Longs and Shorts

	A: All assets	B: ETF Only	C: Future Only	D: ETF & Future
Sharpe Ratio	1.49	.3	.06	.96
Exp. Return (%)	6.65	9.34	2.08	7.95
VaR (%)	5.4	41.1	9.5	10.7
ES (%)	6.7	51.6	11.9	13.4
Avg 2021-23 (%)	2.4	11.6	2.4	4.0
2024 Return (%)	-1.4	18.5	4.5	7.0

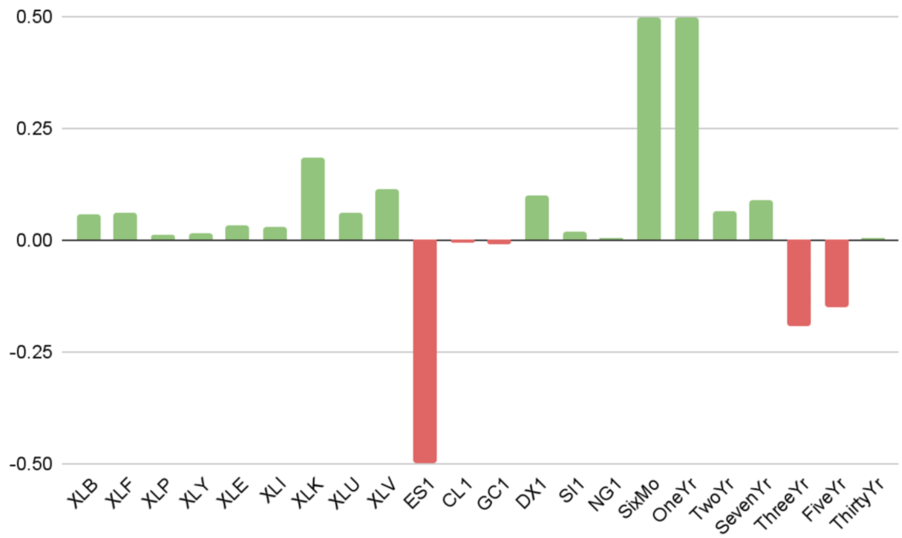


Figure 4: Bar Chart of weights of an All Asset Portfolio (Allows Shorting)

As we can see, portfolios with greater diversification and fewer restrictions tend to have better metrics. In portfolios with shorting allowed, we achieved substantially higher Sharpe ratios than long-only, and our highest portfolio by a wide margin was the All-Asset portfolio at 1.49. We can also see that more diversified portfolios, especially those with a balance of low- and high- volatility assets, have much better risk metrics. While the ETF-only portfolios tend to have high expected returns, their VaR and ES measures are very risky. We can also see how each portfolio would have performed in the 2024 election period, and see that there is a meaningful difference between election-time and non-election-time for each, lending credibility to our initial thesis. However, although our all-asset portfolio had the best risk measures and Sharpe ratio, it actually was the worst-performing portfolio for the 2024 election. This may be explained by our goal of building a portfolio agnostic to the outcome of the election – while other portfolios may have benefited more from one side than the other, perhaps the most diversified portfolio would have performed better in a different outcome.

To achieve a portfolio constrained to a total value of 1 (i.e., the entire value of the portfolio), our weight optimization algorithm constructed long and short parts with total weights of 1.855 and -0.855 respectively. As we can see, the long section of the portfolio is far more diversified with some of its largest key holdings being six-month, one-year, and seven-year treasuries, XLK and XLV sector ETFs, and DX1 futures. By contrast, our short portfolio is much more concentrated, with fewer securities overall, and the majority of its weight being in ES1. However, it also maintains substantial holdings in three- and five-year treasuries. Figure 4 shows a bar chart of the absolute weight of each instrument in our portfolio.

Conclusions

There is a noticeable increase in market volatility during election cycles. The portfolio optimizer suggests a strategy of going long on short-term treasuries while shorting long-term treasuries, which is intuitive given the focus on short-term market windows. However, in recent years, SPY has outperformed the custom portfolios constructed in this analysis. Despite this, we believe that with access to a broader range of securities—including a wider variety of futures, options, individual stocks, and fixed-income products—it may be possible to develop a portfolio capable of outperforming the market during periods of cyclical volatility.

Appendix

Treasuries

1. The bootstrapping algorithm for discount rates:

$$\begin{aligned}
FV &= FV\left(\frac{r_p(t)}{2}\sum_{i=1}^{2T} d(i) + d(T)\right)\\
\iff 1 &= \frac{r_p(t)}{2}\sum_{i=1}^{2T} d(i/2) + d(T)\\
\iff 1 &= \frac{r_p(t)}{2}\sum_{i=1}^{2T-1} d(i/2) + \left(1+\frac{r_p(t)}{2}\right)d(T)\\
\iff \left(1+\frac{r_p(t)}{2}\right)d(T) &= 1 - \frac{r_p(t)}{2}\sum_{i=1}^{2T-1} d(i/2)\\
d(T) &= \left(1 - \frac{r_p(t)}{2}\sum_{i=1}^{2T-1} d(i/2)\right)\left(1+\frac{r_p(t)}{2}\right)^{-1}
\end{aligned}$$

2. Discount factor to spot rate conversion:
   Let τ_i be the time remaining till coupon payment i. If $\tau_i < 0, d(\tau_i) = 0$

$$\begin{aligned}
d(\tau_i) &= \left(1+\frac{r_s(\tau_i)}{2}\right)^{-2\tau_i}\\
d(\tau_i)^{\frac{1}{-2\tau_i}} &= 1 + \frac{r_s(\tau_i)}{2}\\
2d(\tau_i)^{\frac{1}{-2\tau_i}} - 2 &= r_s(\tau_i)
\end{aligned}$$

References

1. Manzo, Gerardo. “Quantifying Political Risk on Financial Markets-Italian Case Study.” Two Sigma, 22 Sept. 2020, www.twosigma.com/articles/quantifying-political-risk-on-financial-markets-italian-case-study/.

2. “Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” Yahoo! Finance, Yahoo!, finance.yahoo.com/.

3. Bloomberg Terminal. Historical Price Data for CL1, GC1, DX1, SI1, NG1. Adjusted close prices, 9 Sep. 1997–4 Dec. 2024. Accessed 7 Dec. 2024.

4. “Daily Treasury Par Yield Curve Rates.” U.S. Department of the Treasury, 19 Nov. 2024, home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value=2024.

5. Lopes, Jon Lopes, et al. “JLOPES3/Mf703_project: Repository for the Final Project for MF703.” GitHub, github.com/jlopes3/MF703_Project.