

Jannate Raddadi

Revised September 2026

Working Paper / Preprint

Value at Risk and Expected Shortfall: Measuring Tail Risk in a Technology

Equity Portfolio

Abstract

This paper applies Value at Risk (VaR) and Expected Shortfall (CVaR) to a portfolio of four large-cap tech stocks: Apple (AAPL), Alphabet/Google (GOOGL), Microsoft (MSFT), and NVIDIA (NVDA). I pulled the actual daily price data from Yahoo Finance using the yfinance Python library with *auto_adjust=True*, which automatically corrects for stock splits and dividends so that each return reflects what an investor actually experienced. The data covers January 2, 2022 through December 31, 2024, giving 751 trading days.

I chose this three-year window deliberately because it covers three structurally different market environments: the 2022 tech selloff, the 2023 recovery driven by AI, and the 2024 extended rally. Having all three in the same dataset makes the risk estimates more realistic and more robust than if I had only looked at a single market regime.

The method I used is historical simulation. You sort the actual historical returns and read off the loss at a given quantile, without assuming any particular distribution. This is the most intuitive

approach, and it is also widely used in practice because it captures the heavy tails and skewness that are present in real financial return data, especially for tech stocks.

I chose an equally weighted portfolio to isolate the effect of diversification and risk measurement without introducing bias from a specific optimization strategy. Each stock receives a 25% weight, representing \$25,000 each within a \$100,000 portfolio.

For this portfolio, the 1-day VaR at 95% confidence comes out to \$3,139.14. That means on about one out of every 20 trading days (roughly 5% of the time), you'd expect to lose more than that amount. I also computed the Expected Shortfall (CVaR), which tells you how bad losses actually are on average on those worst days. At 95%, that number is \$4,101.54, about 31% above the VaR threshold.

That gap is the most important finding in this paper: VaR tells you where the threshold is, while CVaR shows how severe losses actually are once that threshold is crossed. In a tech-heavy portfolio, that difference has significant implications for risk management.

The paper also breaks down the VaR for each individual \$25,000 position, where NVDA alone carries nearly twice the tail risk of AAPL or MSFT. I also look at the diversification benefit you get from holding all four stocks together rather than treating each position separately, which comes out to \$356.81, roughly a 10% reduction in risk.

Finally, I extend the 1-day estimates to a 10-day horizon using the square-root-of-time rule, giving a 95% VaR of \$9,926.83, which is consistent with Basel II and Basel III reporting frameworks. The limitations of historical simulation are discussed openly in the final section.

1. Introduction

Risk is something every investor has to deal with, but most people don't have a precise mathematical way to quantify it. VaR tries to solve this by giving a single number: how much can I lose, with a certain probability, over a given time period? It was developed by JPMorgan in the early 1990s (Longestaey & Spencer, 1996) and is now a standard tool in financial risk management, required under the Basel regulatory framework.

That said, VaR has one important blind spot: it tells you the loss threshold, but nothing about what happens beyond it. For a tech-heavy portfolio where a single stock like NVDA can drop 10% in a day, that gap matters. Expected Shortfall (CVaR) fills it by averaging the losses on the days that actually cross that threshold, giving a more complete picture of what the worst-case scenarios actually look like.

This paper tests whether historical simulation accurately captures tail risk in a correlated tech-sector portfolio, and quantifies how much additional insight CVaR provides beyond VaR alone. The goal isn't just to estimate risk, but to understand how that risk actually behaves in real market conditions: volatile, heavy-tailed, and far from normally distributed.

The paper is organized as follows. Section 2 covers the mathematical framework behind VaR and CVaR. Section 3 explains the historical simulation methodology. Section 4 describes the data. Sections 5 and 6 present the results. Section 7 covers diversification, Section 8 extends the analysis to a 10-day horizon, and Section 9 discusses limitations. Section 10 concludes.

2. Mathematical Framework

2.1. Formal Definition of VaR

Before presenting the formal definition, I first define the key variables.

- **W(Wealth)** = the current value of the portfolio in dollars. In this paper, $W = \$100,000$.
- **R(Return)** = the portfolio's return on a single trading day, written as a decimal. So if the portfolio goes up 2% on a given day, $R = +0.02$. If it drops 3%, $R = -0.03$. R has no units.
- **L(Loss)** = the dollar amount lost on a given day. This is the number we actually care about.

Loss is defined as:

$$L = -W \times R$$

This comes directly from percentage changes. Since losses correspond to negative returns, multiplying by $-W$ converts them into positive dollar values.

For example:

If $R = -0.03$, then $L = 3,000$

If $R = 0.02$, then $L = -2,000$

So positive L represents a loss, and negative L represents a gain.

As a more concrete example, suppose:

$$AAPL - 2.0\%, GOOGL - 1.0\%, MSFT + 0.5\%, NVDA - 3.0\%$$

The equally-weighted portfolio return (25% in each stock) is:

$$R = 0.25(-0.020) + 0.25(-0.010) + 0.25(+0.005) + 0.25(-0.030) = -0.01375$$

The portfolio dropped 1.375% that day. The dollar loss is:

$$L = -(\$100,000)(-0.01375) = +\$1,375$$

So we lost \$1,375 on that day. That's exactly what L is measuring.

Across the 751 trading days in the dataset, we can compute $L_t = -W \times R_t$ for each day, giving us 751 dollar loss observations. VaR summarises this into one number

Let α is the confidence level written as a decimal, $\alpha = 0.95$ means 95% confidence. The term that drives everything is $(1 - \alpha)$, the tail probability. At $\alpha = 0.95$, we are looking for the loss threshold exceeded on only the worst 5% of days.

The formal definition (Artzner et al., 1999):

$$VaR_\alpha = \inf \{ l \in \mathbb{R} : P(L > l) \leq 1 - \alpha \}$$

This is the tightest loss threshold such that the probability of exceeding it is at most $(1 - \alpha)$. We use infimum rather than minimum because for a continuous distribution the boundary may not be

an actual member of the set, though in practice with 751 finite observations, this simply means picking the 37th worst return in the sorted data.

We use this definition instead of a Z-score approach because Z-scores assume normally distributed returns. Tech stocks, especially NVDA, which gained +24.4% and lost −10.0% in single days, violate that assumption badly. The infimum definition works for any return distribution.

To see how this works in practice, consider a simple example with $n = 10$ daily portfolio losses, already sorted from worst to best:

Day (Sorted)	Loss
Day 1 (worst)	\$4,800
Day 2	\$4,200
Day 3	\$3,900
Day 4	\$3,500
Day 5	\$3,100
Day 6	\$2,800
Day 7	\$2,300
Day 8	\$1,700
Day 9	\$900
Day 10 (Best/Least bad)	\$200

At $\alpha = 0.90$: $\lfloor 0.10 \times 10 \rfloor = 1 \text{ day} \rightarrow \text{VaR}_{0.90} = \$4,800$,

In our actual dataset at $\alpha = 0.95$: $\lfloor 0.05 \times 751 \rfloor = 37$ days. The 37th worst daily loss in the sorted dataset is \$3,139.14. That is the 95% VaR.

We can also express VaR directly using quantiles. The p -th quantile of a random variable X is formally defined as (Rudin, 1976; Rockafellar and Uryasev, 2000):

$$Q_{p(X)} = \inf \{ x \in \mathbb{R} : F_{X(x)} \geq p \}$$

Where $F_X(x) = P(X \leq x)$ is the cumulative distribution function of X . In plain terms, $Q_{p(X)}$ is the smallest value x such that at least $p \times 100\%$ of observations fall at or below x .

Since $L = -W \cdot R$. Since W is a positive constant, multiplying by $-W$ reverses inequality direction, which flips the quantile index:

$$Q_{0.05(L)} = Q_{0.05}(-W \cdot R)$$

$$Q_{0.05}(-W \cdot R) = -W \cdot Q_{0.95}(R)$$

Therefore:

$$VaR_{0.95} = -W \cdot Q_{0.95}(R)$$

In plain terms: take the 5th percentile of daily returns, flip the sign since a -3% return is a $+3\%$ loss, and multiply by the portfolio value. That gives you the dollar VaR.

Using our data, the 5th percentile of daily portfolio returns across the 751-day dataset is -3.1391% , so:

$$VaR_{0.95} = -(-0.031391) \times \$100,000 = \$3,139.14$$

On a typical "bad day", this portfolio lost at least \$3,139.14. This happened roughly once every 20 trading days.

2.2.Computing Daily Returns

Before calculating any risk numbers, I first need to convert the raw stock prices into daily returns, because what we actually care about is not what a stock costs on a given day, but how much it moved.

V_t is the adjusted closing price of a stock on trading day t , in dollars. The reason I use adjusted prices instead of raw prices is important. Adjusted prices account for things like stock splits and dividends, so the return series reflects what an investor actually experienced. For example, if a stock does a 2-for-1 split, the raw price drops by 50% overnight, but the investor hasn't lost anything, they just have twice as many shares at half the price. If I used raw prices, my model would read that as a massive one-day loss, which is completely wrong.

The daily return on day t is:

$$R_t = \frac{V_t - V_{t-1}}{V_{t-1}}$$

This is just asking: compared to yesterday's price, how much did today's price change, expressed as a fraction? For example, if AAPL closed at \$150.00 on Monday and \$153.00 on Tuesday:

$$R_{tuesday} = \frac{(153.00 - 150.00)}{150.00} = + 0.02 \text{ (a 2\% gain)}$$

If it closed at \$147.00 instead:

$$R_{tuesday} = \frac{(147.00 - 150.00)}{150.00} = - 0.02 \text{ (a 2\% loss)}$$

For the equally-weighted portfolio:

$$R_{port,t} = 0.25 \times (R_{AAPL,t}) + 0.25 \times (R_{GOOGL,t}) + 0.25 \times (R_{MSFT,t}) + 0.25 \times (R_{NVDA,t})$$

Do this for all 751 trading days and you get a time series of 751 daily portfolio returns. That time series is the main input to every single calculation in this paper, the VaR, the CVaR, the diversification benefit, all of it starts here.

2.3. Why Correlation Matters (Diversification)

Before getting into the results, I want to explain something that might seem counterintuitive at first. If you hold four stocks, you might assume the total portfolio risk is just the four individual risks added together. But that's only true if all four stocks crash by the exact same amount on the exact same days, meaning they move in perfect lockstep with each other. In

real life that never happens, and that gap between what you'd expect by just adding things up and what the portfolio actually risks is called the diversification benefit.

Correlation is what measures this. It's a number between -1 and 1 that tells you how much two stocks tend to move together. A correlation of 1 means perfect lockstep, when one drops 3% , the other drops exactly 3% too, every single time. A correlation of 0 means they move completely independently of each other. In practice it always falls somewhere in between, and the key insight is this: as long as correlations are below 1 , holding multiple stocks together will always reduce your overall risk compared to holding each one separately.

Here's the intuition. Say NVDA has a terrible day and drops 10% . On that same day AAPL might only drop 2% and MSFT might drop 1.5% . They're all tech stocks so they tend to move in the same direction, but not by the same amount at the same time. When you combine all four into one equally-weighted portfolio, those differences start to cancel each other out. NVDA's terrible day gets partially softened by the less-terrible days of the other three. Do that across 751 trading days and the extreme edges of the return distribution get compressed.

You can see this directly in the numbers. The pairwise correlations between our four stocks range from 0.55 to 0.69 ; fairly high, which makes sense since they're all in the same sector and all respond to the same macro factors like interest rates and tech sentiment. But none of them hit 1.0 , and that's enough to produce a real diversification benefit. The next section shows exactly why that's true mathematically.

2.3.1 Why Portfolio Variance Is Less Than the Sum of Individual Variances

To see why diversification reduces risk mathematically, start with how portfolio variance is actually calculated. For an equally-weighted portfolio of four assets with weights $w = 0.25$ each, the portfolio variance is:

$$\sigma_{port}^2 = \sum_{i=1}^4 \sum_{j=1}^4 \omega_i \omega_j \sigma_i \sigma_j \rho_{ij}$$

where σ_i is the standard deviation of asset i and ρ_{ij} is the pairwise correlation between assets i and j . When $i = j$, $\rho_{ii} = 1$ by definition, so the diagonal terms just give you each asset's own variance scaled by its weight squared. The off-diagonal terms are where correlation enters.

If all four stocks moved in perfect lockstep ($\rho_{ij} = 1$ for all pairs), this simplifies to:

$$\sigma_{port}^2 = \left(\sum_{i=1}^4 \omega_i \sigma_i \right)^2$$

meaning the portfolio standard deviation would just be the weighted average of individual standard deviations, and there would be zero diversification benefit. VaR, which scales proportionally with the spread of the return distribution, would equal the sum of individual VaRs exactly.

But when correlations are below 1, the off-diagonal terms shrink, pulling portfolio variance down. The four standard deviations from Table 1 are:

$$\sigma_{AAPL} = 1.7076\%, \sigma_{GOOGL} = 2.0637\%, \sigma_{MSFT} = 1.7381\%, \sigma_{NVDA} = 3.4828\%$$

Under perfect correlation ($\rho = 1$ everywhere):

$$\sigma_{port,perfect} = 0.25(1.7076 + 2.0637 + 1.7381 + 3.4828) = 0.25 \times 8.9922 = 2.2481\%$$

Using the actual correlation matrix from Table 5, the realized portfolio standard deviation comes out to 1.9032%; 15.3% lower than the perfect-correlation benchmark. That compression in volatility is exactly what produces the \$356.81 diversification benefit we observe in Section 7. Holding all four stocks together rather than treating each position separately reduces effective risk by roughly the same proportion that realized volatility falls below its perfect-correlation ceiling.

2.4.Expected Shortfall (CVaR)

VaR has one blind spot: it tells you the loss threshold but nothing about what happens beyond it. A portfolio with a 95% VaR of \$3,139 could have typical tail losses of \$3,200 or \$10,000. VaR treats both situations exactly the same way.

Expected Shortfall (ES), also called CVaR, fixes this by averaging the losses in the tail (Rockafellar and Uryasev, 2000):

$$CVaR_{\alpha} = E[L \mid L \geq VaR_{\alpha}]$$

This reads as: the expected value of L on the days when L is at least as large as VaR. In other words, on the days when you do end up in that worst $(1 - \alpha)$ fraction, what's your average loss?

At $\alpha = 0.95$, we look at the worst 5% of trading days in our 751-day dataset and average the losses on those days. That average comes out to \$4,101.54, so about 31% worse than the

\$3,139.14 threshold itself. That gap tells you the tail of this distribution isn't trivial, with several severely adverse days in the dataset coming mostly from the 2022 bear market and a few extreme NVDA drops.

CVaR also has better theoretical properties than VaR. Artzner et al. (1999) showed that VaR fails the subadditivity property, meaning you could technically construct a situation where the VaR of a combined portfolio is higher than the sum of the individual VaRs, which completely contradicts the logic of diversification. CVaR doesn't have that problem. It satisfies all four axioms of a coherent risk measure, and Rockafellar and Uryasev (2000) showed that it is convex in portfolio weights, meaning you can minimize it using linear programming. That makes CVaR not just a better descriptive tool but a more useful one if you ever want to optimize a portfolio around risk.

3. Methodology: Historical Simulation

3.1. What Historical Simulation Is

There are three main ways to compute VaR: the parametric (variance-covariance) method, Monte Carlo simulation, and historical simulation. I used historical simulation because it's the most widely used in practice and it doesn't require you to assume any particular distribution for returns.

The parametric method is mathematically convenient but assumes returns follow a normal distribution. The problem is that tech stocks don't behave that way. Real financial returns have heavy tails, meaning extreme crashes happen more often than a bell curve would predict, and they tend to be negatively skewed, meaning the bad days are more extreme than the good days.

NVDA alone had a single-day gain of +24.4% and a single-day loss of −10.0% in our dataset, that kind of range doesn't fit neatly into a normal distribution, and forcing it to would cause you to underestimate how bad the worst days actually get.

Historical simulation bypasses that problem entirely. Whatever skewness, heavy tails, or extreme moves exist in the 751-day dataset get captured automatically because we're reading directly from what actually happened rather than modeling it. This makes historical simulation one of the most transparent and popular approaches in practice, favored specifically for its simplicity and its lack of distributional assumptions. Hendricks (1996) found that it performs reliably across different market conditions, and Sharma (2012) confirms that it matches actual loss frequencies well over long time horizons.

3.2. Step-by-Step Procedure

The procedure is straightforward:

Step 1: Download the actual daily adjusted closing prices for AAPL, GOOGL, MSFT, and NVDA from Yahoo Finance and compute the daily percentage return for each stock, as defined in Section 2.2. In the implementation, this was handled using `pct_change()`.

Step 2: Combine the four individual stock returns into a single portfolio return for each day using equal weights of 0.25 per stock, since each position represents exactly one quarter of the \$100,000 portfolio.

Step 3: Sort all 751 portfolio return observations in ascending order, from the worst day to the best day.

Step 4: Identify the cutoff index using the general formula $\lfloor (1 - \alpha) \times T \rfloor$, where α is the confidence level and T is the total number of observations. For 95% VaR with $\alpha = 0.95$ and $T = 751$:

$$\lfloor (1 - 0.95) \times 751 \rfloor = \lfloor 37.55 \rfloor = 37$$

So the 95% VaR corresponds to the return at index 37 in the sorted list, the 37th worst day out of 751. This floor method is used consistently across all VaR and CVaR calculations so that every number in this paper is computed exactly the same way.

Step 5: Multiply the absolute value of that 37th return by the portfolio value of \$100,000 to get the final dollar VaR. The absolute value is used because that return is negative; a loss; and we want the result to come out as a positive dollar figure, consistent with the sign convention established in Section 2.

3.3. Scaling to Multi-day Horizons

To extend from a 1-day to a 10-day VaR, I used the square-root-of-time rule, which is the standard approach used for bank capital requirements under Basel II and Basel III. The justification is straightforward: if daily returns are independent and identically distributed with variance σ^2 , then the variance of a 10-day return is the sum of ten independent daily variances,

which equals $10\sigma^2$. Since standard deviation scales with the square root of variance, and VaR scales with standard deviation, the 10-day VaR is just the 1-day VaR multiplied by $\sqrt{10}$:

$$VaR_{10-day} = VaR_{1-day} \times \sqrt{T}$$

For $T = 10$:

$$VaR_{10-day} = \$3,139.14 \times \sqrt{10} = \$3,139.14 \times 3.1623 \approx \$9,926.83$$

So over any 10-trading-day window there's a 5% chance of losing more than \$9,926.83, about 9.93% of the total portfolio value. An investor could expect a loss of roughly this size about once a year.

The key assumption here is that daily returns are IID, each day's return is drawn from the same distribution and is unaffected by what happened the day before. In reality, and especially with tech stocks, big moves tend to cluster together. During a volatile stretch like early 2022 the actual 10-day risk was likely higher than this formula suggests. This limitation is discussed further in Section 9.

4. Data and Descriptive Statistics

The data consists of 751 actual daily return observations per stock, pulled from Yahoo Finance using the yfinance Python library. The period runs from January 2, 2022 through December 31, 2024. Returns were computed from adjusted closing prices to account for splits and dividends, and everything was saved to a CSV file so the results are fully reproducible.

What makes this three-year window particularly useful is that it covers three completely different market environments. In 2022 tech stocks got hit hard when interest rates rose sharply, NVDA fell over 50% and both GOOGL and MSFT dropped more than 30%. Then 2023 was a sharp reversal driven by the AI buildout, with NVDA leading the way at +239%. And 2024 kept the momentum going with NVDA up another 171%. Having all three regimes in the same dataset makes the VaR estimates much more realistic and robust than if I had only looked at a bull market or a bear market in isolation.

Table 1: Descriptive Statistics for Daily Returns (2022–2024)

Stock	Mean Daily Return	Std. Dev (Daily)	Min Return	Max Return
AAPL	+0.0601%	1.7076%	-5.8680%	+8.8975%
GOOGL	+0.0586%	2.0637%	-9.5094%	+10.2244%
MSFT	+0.0502%	1.7381%	-7.7156%	+8.2268%
NVDA	+0.2623%	3.4828%	-10.0046%	+24.3696%
Portfolio	+0.1078%	1.9032%	-6.6844%	+9.7587%

Source: Yahoo Finance via yfinance. All values in percent.

A few things stand out right away. NVDA has by far the highest mean return at +0.2623% per day, but it also has the highest volatility at 3.4828% and the most extreme single-day moves in the whole dataset, a worst day of −10.0% and a best day of +24.4%. That kind of range is unusual even for tech stocks and is exactly why a normal distribution would fail to capture its risk accurately. On the other end, MSFT was the most consistent of the four with the lowest volatility and the smallest daily swings.

What's particularly interesting is that the portfolio standard deviation of 1.9032% ends up closer to AAPL and MSFT's individual volatility than to NVDA's 3.4828% or even GOOGL's 2.0637%. We're holding one of the most volatile large-cap stocks in the market at a 25% weight, and yet the overall portfolio risk gets pulled down almost to the level of the least volatile stocks in the group. As explained in Section 2.3, this is exactly what imperfect correlations between assets produce, and the numbers here show it happening in practice before we've calculated a single VaR figure.

5. VaR Results

5.1. Portfolio VaR Results.

Table 2 shows the 1-day historical simulation VaR at three confidence levels for the \$100,000 portfolio.

Table 2: Portfolio 1-Day VaR at 90%, 95%, and 99% Confidence

Confidence Level		1-Day VaR (%)	1-Day VaR (\$100K Portfolio)
90%		-2.1797%	\$2,179.70
95%		-3.1391%	\$3,139.14
99%		-4.7614%	\$4,761.43

Method: Historical Simulation. Portfolio value: \$100,000.

These results make intuitive sense. At 90% confidence there's a 10% chance of losing more than \$2,179.70 in a day, roughly one bad day every two weeks. At 95% the threshold rises to \$3,139.14, which is about one bad day per month. And the 99% VaR of \$4,761.43 represents the really extreme days, the kind you'd only expect to see maybe two or three times a year. The

numbers scale the way you'd expect as you move up in confidence level, which is a good sign that the results are internally consistent.

One important thing to keep in mind: VaR only tells you the threshold you are likely to breach, not how severe losses can get once you are past it. That's exactly why we look at CVaR in the next section.

5.2. Individual Stock VaR

To see where the risk is coming from, Table 3 breaks down the VaR by individual stock position, assuming a \$25,000 investment in each.

Table 3: Individual Stock VaR at 95% Confidence (\$25,000 per position)

Stock	5th Percentile Loss (%)	Allocated Capital	Individual VaR (\$)
AAPL	-2.7205%	\$25,000	\$680.13
GOOGL	-3.1994%	\$25,000	\$799.84
MSFT	-2.7853%	\$25,000	\$696.32
NVDA	-5.2786%	\$25,000	\$1,319.66
Sum of Individuals	—	\$100,000	\$3,495.94
Portfolio VaR	—	\$100,000	\$3,139.14

Portfolio VaR is less than the sum of individual VaRs because of diversification.

NVDA is the clear outlier. At \$1,319.66 its individual risk contribution is almost double AAPL's \$680.13, which is the safest position in the group. MSFT sits right next to AAPL at \$696.32, making those two the steady anchors of the portfolio while NVDA and GOOGL carry most of the tail risk.

The gap between the sum of individual VaRs (\$3,495.94) and the actual portfolio VaR (\$3,139.14) is \$356.81, about a 10% reduction. This is the diversification benefit introduced in Section 2.3, and Section 7 examines it in full detail using the actual correlation structure of the four stocks.

6. Expected Shortfall (CVaR)

6.1. How I Computed CVaR

To compute Expected Shortfall (CVaR) I identified all portfolio return observations at or below the Value at Risk (VaR) cutoff and took their arithmetic mean. For 95% confidence with $T = 751$ that means averaging the worst 37 daily returns, (37 because $\lfloor 0.05 \times 751 \rfloor = 37$), which is the same floor method used throughout Section 3.2, keeping everything consistent. In Python:

```
idx = int(np.floor(0.05 * len(sorted_port)))
var_pct = sorted_port[idx]
tail = port_returns[port_returns <= var_pct]
cvar = abs(tail.mean()) * portfolio_value
```

The formula is:

$$CVaR_{0.95} = -W \cdot \frac{1}{37} \sum_{R_t \leq Q_{0.05}(R)} R_t$$

This stays consistent with the $L = -W \cdot R$ framework from Section 2, we're averaging the tail returns and converting to dollar losses the same way we defined L throughout. Note that CVaR and Expected Shortfall (ES) are used interchangeably throughout this paper and refer to the same risk measure.

Table 4: VaR vs. CVaR at 95% and 99% Confidence Levels

Confidence Level	Metric	Percentage Loss	Dollar Loss (\$100K)
95%	VaR	-3.1391%	\$3,139.14
95%	CVaR (ES)	-4.1015%	\$4,101.54
99%	VaR	-4.7614%	\$4,761.43
99%	CVaR (ES)	-5.3808%	\$5,380.77

All values computed directly from Python output using real Yahoo Finance data.

6.2. What these Numbers Mean

At the 95% confidence level, the VaR comes out to \$3,139.14 but the CVaR is \$4,101.54, a difference of about 31%. That gap is the most important number in this table. It means that on the days when the portfolio does breach the VaR threshold, the average loss is nearly a third worse than the threshold itself. For a tech-heavy portfolio like this one, where NVDA alone can drop 10% in a single day, that distinction between where the bad days start and how bad those days actually get is exactly the kind of information that VaR alone cannot give you.

At the 99% confidence level, VaR rises to \$4,761.43 and CVaR to \$5,380.77, a difference of approximately 13%. This narrower gap compared to the 95% level is expected, the higher the confidence level, the deeper into the tail we are looking, and the less room there is between the threshold and the average beyond it. However, the 99% CVaR estimate rests on approximately 7 to 8 observations; its statistical reliability is assessed formally in Section 9.1 via bootstrap confidence intervals.

As established in Section 2.4, CVaR is a coherent risk measure and convex in portfolio weights, making it not just a better descriptive tool but a more useful one for anyone looking to optimize a portfolio around a specific risk target.

7. Diversification Benefit

As shown in Section 2.3.1, diversification reduces portfolio risk because pairwise correlations below 1 compress realized portfolio volatility below its perfect-correlation ceiling. VaR gives us a concrete way to measure exactly how much.

Adding up the individual position VaRs from Table 3:

$$\$680.13 + \$799.84 + \$696.32 + \$1,319.66 = \$3,495.94$$

The actual portfolio VaR at 95% is \$3,139.14. So the diversification benefit is:

$$\text{Diversification Benefit} = \$3,495.94 - \$3,139.14 = \$356.81$$

Holding all four stocks together rather than treating each position in isolation reduces the risk estimate by about \$357, roughly a 10% reduction.

Table 5: Pairwise Correlation Matrix of Daily Returns (2022–2024)

	AAPL	GOOGL	MSFT	NVDA
AAPL	1.0000	0.6223	0.6847	0.5545
GOOGL	0.6223	1.0000	0.6941	0.5485
MSFT	0.6847	0.6941	1.0000	0.6378
NVDA	0.5545	0.5485	0.6378	1.0000

All correlations are positive, reflecting shared tech-sector exposure.

All four stocks are moderately to highly correlated, which makes sense, they all respond to the same macro factors like interest rates, tech sector sentiment, and overall market conditions.

AAPL and MSFT have the highest pairwise correlation at 0.6847, while NVDA has somewhat lower correlations with the others (0.55 to 0.64) because its returns are more tied to semiconductor cycles and AI hardware demand specifically.

The 10% diversification benefit is real but not huge, and that's because all four stocks are in the same sector. If you added assets from other sectors, healthcare, consumer staples, bonds, the correlations would drop and the benefit would be significantly larger. This is why the standard advice is to diversify across asset classes, not just within tech.

8. Ten-Day VaR and Regulatory Context

The 1-day VaR is useful for daily risk monitoring, but Basel II and Basel III require banks to hold capital against a 10-day VaR at the 99% confidence level. For individual investors, the 10-day horizon is also meaningful because it's roughly two calendar weeks, enough time for a large loss to accumulate before you'd realistically rebalance.

Applying the square-root-of-time rule from Section 3.3:

$$VaR_{10-day, \%95} = \$3,139.14 \times \sqrt{10} = \$3,139.14 \times 3.1623 \approx \$9,926.83$$

So over any 10-trading-day window there's a 5% chance of losing more than \$9,926.83, so about 9.93% of the total portfolio value. An investor could expect a loss of roughly this size about once per year.

It is worth noting that this calculation uses the 95% confidence level for consistency with the rest of the analysis, while Basel regulations formally require 99%. The structural limitations of the square-root rule, particularly the IID assumption and its implications during volatile periods, are discussed in Section 9.

9. Limitations

Historical simulation is a solid and transparent method, but it has some real limitations worth being upfront about.

The most fundamental one is that it's completely backward-looking. It assumes the next 751 trading days will look more or less like the last 751. If something changes structurally; a new Federal policy, a financial crisis, a major technology shock; the model won't catch it until after it has already happened. The 2022 bear market is a perfect example: it was driven by the fastest rate-hiking cycle in decades, and any model trained only on pre-2022 data would have badly underestimated the downside risk going into that year.

Second, the results are sensitive to the lookback window. I used 751 days as a practical compromise, long enough to capture three different market regimes, short enough that the data is still relevant. But a different window would give different numbers. A natural extension of this study would be a sensitivity analysis showing how the VaR estimates shift across different lookback periods, like one year versus five years.

Third, the 99% CVaR estimate is based on only about 7–8 observations, since 1% of 751 days is roughly 7.5. With so few data points a single extreme day can meaningfully shift the average. To

quantify exactly how unstable this estimate is, I ran a bootstrap simulation with 10,000 resamples of the tail observations and computed a 95% confidence interval around both the 95% and 99% CVaR estimates.

9.1 Bootstrap Confidence Intervals for CVaR Stability

Rather than assuming the tail observations follow any particular distribution, the bootstrap procedure repeatedly resamples the actual tail returns with replacement and recomputes CVaR each time. After 10,000 iterations, the distribution of those recomputed values tells us how much the estimate would move if we had drawn a slightly different set of trading days. The procedure was implemented in Python using 10,000 resamples (Appendix A).

Metric	Point Estimate	95% Bootstrap CI	CI Width
95% CVaR	\$4,101.54	[\$3,854.04 – \$4,369.03]	\$514.99
99% CVaR	\$5,380.77	[\$5,043.22 – \$5,813.70]	\$770.48

The 95% CVaR confidence interval spans \$514.99, staying within roughly $\pm 6\%$ of the point estimate, a reasonably tight range anchored by 37 tail observations. The 99% CVaR interval is noticeably wider at \$770.48, spanning about $\pm 14\%$ of the point estimate, driven entirely by those 7–8 tail observations. This is not a modeling flaw, it is a fundamental data limitation. A dataset covering 10 or more years would provide 25+ observations in the 99% tail and produce a materially more stable estimate. For this paper, the 95% CVaR of \$4,101.54 should be treated as

the reliable risk estimate, and the 99% CVaR of \$5,380.77 should be interpreted as directionally correct but statistically noisy.

Fourth, as established in Section 2.4, VaR fails the subadditivity property and is not a coherent risk measure. In practice this rarely causes problems, but during periods of extreme market stress it can lead to misleading results. This is why Basel IV, currently being phased in, is moving toward CVaR as the primary standard for market risk capital requirements.

Fifth, the square-root-of-time rule used for the 10-day VaR ignores volatility clustering. In reality big losses tend to cluster together, a bad day makes another bad day more likely. The intuition is straightforward: if there was a storm on Monday, Tuesday and Wednesday will almost certainly be affected by it, so you calculate risk accordingly rather than assuming every day starts fresh. During the 2022 selloff, volatility was elevated for weeks at a time, meaning the true 10-day risk during that stretch was likely meaningfully above \$9,926.83. A GARCH model would handle this more accurately by explicitly modeling how volatility evolves over time, and represents the most natural extension of this analysis.

What this data really shows is a tension that comes with any tech-heavy portfolio: the same AI-driven growth that pushes these stocks to massive gains also creates an aggressive risk profile that a simple standard deviation just can't capture. By skipping the bell curve assumption and using historical simulation instead, this study found that the real danger for this portfolio isn't the 3.1% daily loss threshold that VaR gives you; it's the 31% gap between that threshold and what losses look like on the worst days, which is what CVaR measures.

The analysis also makes clear that NVDA is both the portfolio's biggest opportunity and its biggest risk. It carries nearly twice the individual risk of AAPL or MSFT, and that gap is directly tied to how aggressively it moves. At the same time, the 10% diversification benefit we found, even within a single sector where all four stocks are fairly correlated, shows that risk management isn't just about picking safer stocks. Methodologically, bootstrap confidence intervals confirm that the 95% CVaR estimate of \$4,101.54 is statistically stable, while the 99% estimate should be treated as directionally informative rather than precise, a reminder that reliable tail risk measurement requires more data than three years can provide at the deepest confidence levels.

At the end of the day, the whole point of calculating VaR is to replace gut feeling with an actual number. These models are backward-looking and they have real limitations, but they give investors something concrete: a threshold, a confidence level, and a time horizon to plan around. In a sector like tech where extreme moves and sudden crashes are part of the deal, knowing that your 10-day risk sits at \$9,926.83 is the difference between having a plan going into a market correction and being caught completely off guard. Risk in tech is unavoidable, but as this research shows, it is at least something you can measure.

Works Cited

- Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999).
Coherent measures of risk. Mathematical Finance, 9(3), 203–228.
 Available at:
[https://www.ressources-actuarielles.net/EXT/ISFA/1226.nsf/0/0bd2345fdff664eec1256e16004ff688/\\$FILE/ARTZNER%201998.pdf](https://www.ressources-actuarielles.net/EXT/ISFA/1226.nsf/0/0bd2345fdff664eec1256e16004ff688/$FILE/ARTZNER%201998.pdf)
- Hendricks, D. (1996).
Evaluation of value-at-risk models using historical data.
 Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1028807
- Longerstaey, J., & Spencer, M. (1996).
RiskMetrics—Technical document (4th ed.). Morgan Guaranty Trust Company.
- Rockafellar, R. T., & Uryasev, S. (2000).
Optimization of conditional value-at-risk. The Journal of Risk, 2(3), 21–41.
- Rudin, W. (1976). *Principles of Mathematical Analysis* (3rd ed.). McGraw-Hill.
- Sharma, M. (2012).
The historical simulation method for value-at-risk: A research-based evaluation of the industry favorite. SSRN Electronic Journal.
<https://doi.org/10.2139/ssrn.2042594>

Appendix A

```

np.random.seed(42)
n_boot = 10_000

# 95% CVaR bootstrap
idx_95 = int(np.floor(0.05 * len(sorted_port)))
var_pct_95 = sorted_port[idx_95]
tail_95 = port_returns[port_returns <= var_pct_95].values

boot_cvar_95 = []
for _ in range(n_boot):
    resample = np.random.choice(tail_95, size=len(tail_95),
                                replace=True)
    boot_cvar_95.append(abs(resample.mean()) * portfolio_value)

ci_95_low = np.percentile(boot_cvar_95, 2.5)
ci_95_high = np.percentile(boot_cvar_95, 97.5)

# 99% CVaR bootstrap
idx_99 = int(np.floor(0.01 * len(sorted_port)))
var_pct_99 = sorted_port[idx_99]
tail_99 = port_returns[port_returns <= var_pct_99].values

boot_cvar_99 = []
for _ in range(n_boot):
    resample = np.random.choice(tail_99, size=len(tail_99),
                                replace=True)
    boot_cvar_99.append(abs(resample.mean()) * portfolio_value)

ci_99_low = np.percentile(boot_cvar_99, 2.5)
ci_99_high = np.percentile(boot_cvar_99, 97.5)

print(f"95% CVaR: ${abs(tail_95.mean())*portfolio_value:,.2f} "
      f"[95% CI: ${ci_95_low:,.2f} - ${ci_95_high:,.2f}]" )
print(f"99% CVaR: ${abs(tail_99.mean())*portfolio_value:,.2f} "
      f"[95% CI: ${ci_99_low:,.2f} - ${ci_99_high:,.2f}]" )

```