Strategic Investment Analysis Report

Identifying market leaders in the global cloud services sector and modeling the volatility of their stock returns to drive investment strategic decisions

Prepared for: Financial & Investment Research Department

SEPTEMBER 9, 2025

Prepared by: Hisham Alboo

Studying Executive Master of Business Administration

"Strategic Leadership" at Quantic School of Business and Technology

Table of Contents

Executive Summary:

This report investigates the intersection of cloud computing market leadership and financial risk modeling to guide strategic investment decisions. It begins by identifying Amazon Web Services (AWS) and Microsoft Azure as dominant global cloud providers, supported by infrastructure breadth, service innovation, and competitive market share. Modeling the volatility of stock returns is crucial for modern finance, moving beyond historical observations to provide a dynamic understanding of risk. This approach offers benefits such as enhanced risk management, optimized portfolio construction, improved derivatives pricing, informed trading strategy development, and better performance evaluation. The expected outcomes of this report include identifying Amazon and Microsoft as the dominant players, profiling their competitive advantages (e.g., AWS's broad services and global infrastructure, and Microsoft's enterprise integration and AI investments), and applying statistical models to analyze the volatility of their daily stock returns.

Objectives/ Problem statement:

The primary objectives of this report are to:

- Examine the critical role of cloud computing in modern business operations, assessing its impact through scalable, flexible, and cost-effective solutions for managing data, applications, and services.
- Identify and analyze the dominant global cloud service providers, evaluating their infrastructure breadth, service innovation, and competitive market share.
- Profile the competitive advantages of these market leaders, including AWS's broad services and global infrastructure, and Microsoft's enterprise integration and AI investments.
- Problem statement : How can we develop statistical models to analyze the volatility of daily stock returns for leading cloud companies, thereby providing a dynamic understanding of their financial risk to drive strategic investment decisions?

Overview:

In this section, we will cover the following topics:

A- Cloud computing is important for modern businesses and plays a key role in their operations.

B- Which global companies dominate the cloud services market?

C- Modeling the volatility of companies' stock returns.

A- Cloud computing is important for modern businesses and plays a key role in their operations:

Cloud computing has emerged as a pivotal technology reshaping the landscape of information technology and business operations, offering scalable, flexible, and cost-effective solutions for managing data, applications, and services (Adeoye & Osibo, 2023)[1] ,(Onwuzurike, 2024) [2].

Its importance lies in its ability to dynamically provide computing resources, shifting expenditure from capital investments to more manageable operational costs for businesses of all sizes (Onwuzurike, 2024)[2] , (Choudhary et al., 2015)[3] . This paradigm is built upon robust cloud infrastructure, which serves as the foundational element enabling the delivery of various cloud services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Mohanasundaram et al., 2018)[4] (Marinescu, 2013)[5] (Casino et al., 2024)[6] (Pathak et al., 2024)[7] (Razaque et al., 2022)[8] .

The cloud infrastructure typically comprises essential components such as servers, storage, network, and management software (GhasemiGol, 2019)[9] (Lal, 2015)[10] . These components are often virtualized, allowing for the dynamic allocation and scaling of resources based on user demand (Sato et al., 2011)[11] . The management system within the cloud infrastructure is crucial for ensuring the normal operation and

dynamic provisioning of these resources (Sato et al., 2011)[11] (Innocent, 2012)[12] . Key cloud service providers like Amazon, Google, and Microsoft have pioneered and continue to support these delivery models, with companies like Amazon being significant in IaaS, Google focusing on SaaS and PaaS, and Microsoft also involved in PaaS (Marinescu, 2013)[5] .

The importance of cloud computing for businesses manifests in numerous benefits. These include significant cost savings by reducing the need for substantial upfront investments in hardware and software (Onwuzurike, 2024)[2] (Choudhary et al., 2015)[3] (Revathi et al., 2024)[13] . It enhances agility and resource utilization by providing scalable computing resources over the internet (Onwuzurike, 2024)[2] . Cloud adoption can revolutionize how enterprises operate, streamlining processes, fostering collaboration, and improving overall efficiency (Adeoye & Osibo, 2023)[1] (Revathi et al., 2024)[13] . Moreover, cloud

computing offers enhanced scalability, improved collaboration, and the capacity to adapt quickly to shifting market conditions (Revathi et al., 2024)[13] . Beyond cost and efficiency, cloud computing contributes to security benefits, such as enhanced data protection, improved disaster recovery capabilities, proactive threat detection, and centralized management (Shriwas et al., 2024)[14] . The global nature of cloud services, accessible on-demand from anywhere with network access, further underscores its importance (Gundu et al., 2020)[15] .

The infrastructure supporting cloud operations involves complex architectural elements. A typical cloud computing architecture includes components such as database servers, application servers, and directory servers, all within the cloud environment (Jyoti et al., 2020)[16] . Different types of clouds exist, including private, public, hybrid, and community clouds, each offering various service models like IaaS, PaaS, and SaaS (Jyoti et al., 2020)[16] (Gundu et al., 2020)[15] . Load balancers are critical

for distributing incoming requests across multiple servers, optimizing resource utilization, and ensuring high availability (Jyoti et al., 2020)[16] (Gundu et al., 2020)[17] (Muteeh et al., 2021)[18] . Management components like cloud controllers, schedulers, and brokers orchestrate the overall operation, resource allocation, and interaction between providers and consumers (Jyoti et al., 2020)[16] (Khodayarseresht et al., 2023)[19] . Firewalls are essential for security, controlling network traffic and protecting the cloud environment (Jyoti et al., 2020)[16] . Image analysis of cloud architectures often depicts this layered approach, showing the interplay between devices, network infrastructure, and the various cloud service. For instance, one visualization details the cloud computing architecture with its different cloud types and the role of components like the load balancer and cloud controller. Another highlights the device connections and the application, platform, and infrastructure layers of cloud services. Mobile cloud computing

architectures, for example, illustrate the interaction between mobile clients, mobile networks, the internet, and cloud service providers.

The importance of cloud infrastructure extends to its role in supporting emerging technologies and trends. It provides the backbone for big data analytics, enabling companies to derive valuable insights from vast amounts of data generated within their systems (Saraswat & Choudhari, 2024)[20] . Cloud computing, with its cost-effective resources, facilitates the hosting and processing of this data (Saraswat & Choudhari, 2024)[20] . Integration with the Internet of Things (IoT) is another crucial area, where cloud infrastructure handles the massive amount of data generated by IoT devices, supporting applications in smart cities and various industries (Huang et al., 2020)[21] (Alhaidari et al., 2020)[22] (Pathak et al., 2024)[7] (Čolaković, 2023)[23] (Alam, 2021)[24] . The cloud also plays a vital role in supporting cutting-edge fields like artificial intelligence, robotics, and upcoming technologies, providing the

computational power and infrastructure required for these advancements . The concept of the "cloud-to-edge continuum" emphasizes the integrated approach, where cloud infrastructure works in tandem with edge computing to meet the stringent requirements of futuristic applications like metaverse and haptic communication (Maia et al., 2024) [25] . Architectures illustrating this continuum show the interaction between cloud data centers, access networks, and edge computing nodes providing localized services.

Furthermore, cloud computing's impact on business value is being actively studied. Research explores how cloud adoption influences different types of innovation, such as exploitative and explorative innovation, and how environmental characteristics like dynamism and complexity moderate these relationships (Li et al., 2024)[26] . A conceptual model highlights how cloud computing influences business value through innovation, considering environmental characteristics

and control variables like industry and firm size. Studies also investigate the practical business impacts of cloud computing adoption, identifying strategic, managerial, operational, and functional benefits (Deed & Cragg, 2022)[27] . The financial services industry, for example, is leveraging cloud computing for various applications (Dandapani, 2017)[28] .

Despite the numerous benefits, the adoption of cloud computing also presents challenges, particularly regarding security and privacy (Singh et al., 2018)[29] (Onwuzurike, 2024)[2] (Sharma & Ahuja, 2017)[30] (Sharma et al., 2023)[31] . Ensuring data confidentiality and integrity within the cloud environment is a core concern (Singh et al., 2018)[29] (Pathak et al., 2024)[7] . Attacks on cloud infrastructure can lead to significant service disruptions (GhasemiGol, 2019)[9] . Therefore, addressing security issues for cloud computing infrastructure is paramount (Sharma & Ahuja, 2017)[30] .

In conclusion, cloud computing, underpinned by a complex and dynamic infrastructure, is of significant importance to modern businesses due to its ability to deliver scalable, flexible, and cost-efficient IT resources. Its benefits, ranging from cost savings and increased efficiency to enhanced security and support for emerging technologies, are driving its widespread adoption across various sectors. While challenges, particularly in security, require careful consideration, the ongoing evolution of cloud infrastructure and its integration with other technological advancements solidify its critical role in the current and future business landscape.

B-Which global companies dominate the cloud services market?

The global cloud services market is primarily dominated by a few major players, notably Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) (Bhatte et al., 2022)[32] (Patil & Sankapal,

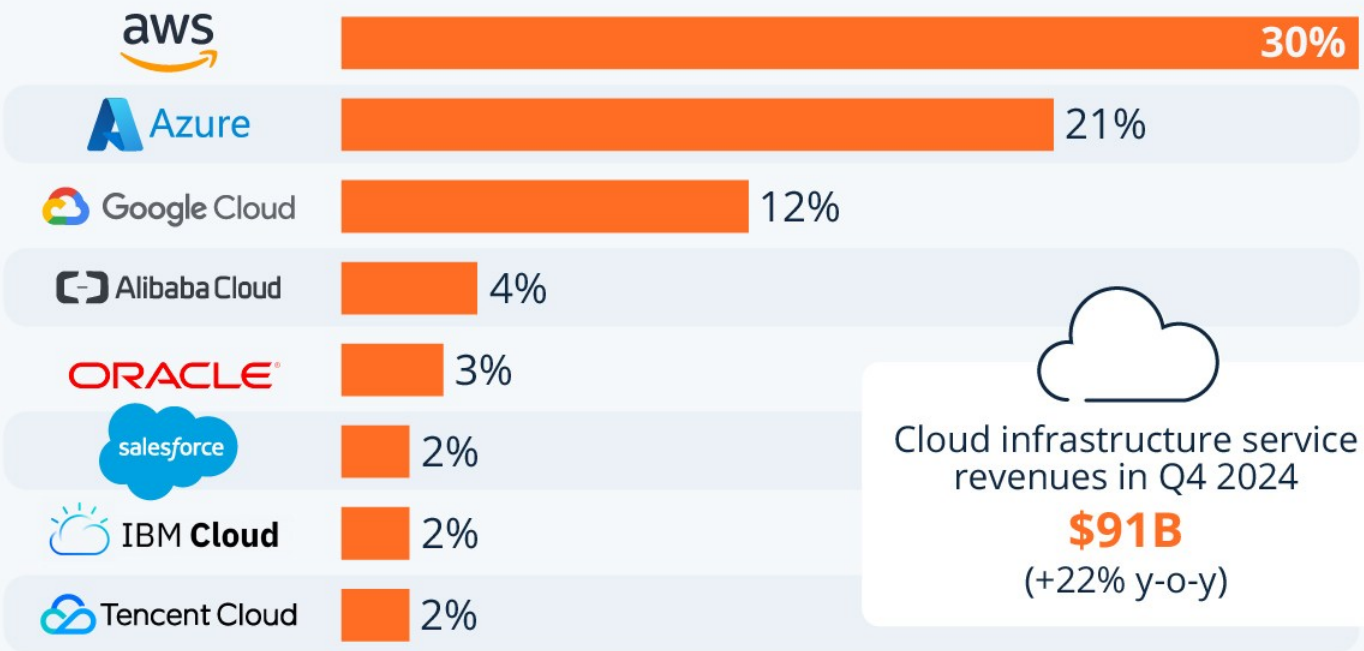
2024)[33] . These providers offer a wide array of services, encompassing Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Data as a Service (DaaS), catering to both enterprises and individual users on a pay-as-you-go basis (Bhatte et al., 2022)[32] (Sajid et al., 2022)[34] (Gundu et al., 2020)[15] . While IBM Cloud and Oracle Cloud are also significant contenders, AWS, Azure, and GCP hold the largest market share (Bhatte et al., 2022)[32] (Sajid et al., 2022)[34] .

But the last news available ,Amazon Web Services (AWS) continues to outperform its competitors, with a 30% market share in Q4 2024 — ahead of Microsoft's Azure at 21% and Google Cloud at 12%. The "Big Three" collectively account for more than 60% of the growing market. Global spending on cloud infrastructure services grew by 22% in Q4 2024, adding $17 billion and bringing total spending for the quarter to $91 billion, while for the full year, service revenues surpassed $330

billion. This strong and sustained growth reflects growing enterprise reliance on the cloud, a trend further driven by the growing influence of Artificial Intelligence (AI) services. According to John Dinsdale, chief analyst at Synergy Research Group, this forms a “virtuous cycle”—new capabilities enable greater demand, which in turn fuels further investment and innovations in the sector (Felix Richter, 2025)[35] .

Amazon and Microsoft Stay Ahead in Global Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q4 2024*



* Includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group

statista

The competitive advantages of these leading cloud service providers stem from various factors, including their extensive infrastructure, diverse service portfolios, continuous innovation, and established market presence.

Below is an analysis of the competitive advantages that position these companies as market leaders, with considerations for the competitive landscape in 2025 based on available literature:

Company	Competitive Advantages
Amazon Web Services (AWS)	Established market leader with the broadest and deepest set of services, extensive global infrastructure, strong focus on innovation (including AI/ML and HPC) (Harika, 2024)[36] (Sochat et al., 2025)[37] (Pacios et al., 2025)[38] , and a mature ecosystem. Known for performance and variability (Ahuja et al., 2025)[39] and robust security features (Guptha et al., 2021)[40] (Rath et al., 2019)[41] .

Microsoft Azure	Strong presence in enterprise markets due to integration with Microsoft ecosystem, competitive pricing, hybrid cloud capabilities, and significant investments in AI/ML and data analytics (Hassan et al., 2022)[42] (Polinati, 2025)[43] (Tamla et al., 2025)[44] . Offers comprehensive security solutions comparable to AWS (Guptha et al., 2021)[40] (Rath et al., 2019)[41]
Google Cloud Platform (GCP)	Strengths in data analytics, machine learning, and open-source technologies like Kubernetes (Borra, 2024)[45] . Known for competitive pricing, performance variability (Ahuja et al., 2025)[39] , and advanced networking capabilities (Lin et al., 2025)[46] . Expanding its service offerings and global footprint (Sajid et al., 2022)[34] .

These providers differentiate themselves through various means. AWS, for instance, is noted for its comprehensive suite of services and its pioneering role in the cloud market, maintaining a leading position in

terms of market share (Bhatte et al., 2022)[32] (Harika, 2024)[36] . Its infrastructure-as-a-service offerings, such as EC2 for virtual machines and S3 for storage, are widely adopted (Varshney & Ujjwal, 2021)[47] (Liu et al., 2023)[48] . AWS also provides robust ETL tools for data management (Borra, 2024)[49] .

Microsoft Azure leverages its strong relationships with enterprises already using Microsoft software and services. It offers a competitive alternative to AWS, particularly in areas like backup programming and support for IoT and retail organizations (Hassan et al., 2022)[42] . Azure's services are also used in specialized applications like medical named entity recognition (Tamla et al., 2025)[44] and residential energy systems interoperability (Leniston et al., 2025)[50] . Azure DevOps and GitHub, both under Microsoft, are prominent CI/CD platforms (Manolov et al., 2025)[51] .

GCP is recognized for its prowess in data analytics and machine learning, making it a preferred choice for data-intensive workloads (Sajid et al., 2022)[34] (Borra, 2024)[45] . Its services are increasingly being used in comparative analyses against AWS and Azure for various applications, including geoinformation data processing (Nedosnovanyi et al., 2023)[52] and high-performance computing (HPC) (Sochat et al., 2025)[37] (Munhoz & Castro, 2023)[53] . Cross-cloud platforms utilizing both AWS and GCP are also being explored for applications like satellite image detection (Pacios et al., 2025)[38] .

The competitive landscape in 2025 continues to be shaped by these major players, with ongoing developments in areas such as serverless computing (Function-as-a-Service) offerings like AWS Lambda and Azure Functions (Ivan et al., 2019)[54] , and the increasing adoption of

containerized microservices orchestrated and provisioned within these cloud environments (Saboor et al., 2022)[55] (Muniswamy & Vignesh, 2024)[56] . Security remains a crucial factor, with providers continually enhancing their security services and compliance measures (Guptha et al., 2021)[40] (Rath et al., 2019)[41] . The concept of hybrid cloud, combining on-premise infrastructure with public cloud services, is also a significant trend (Polinati, 2025)[43] (Gundu et al., 2020)[15] . The market is also seeing increased scrutiny from regulatory bodies, as highlighted by discussions around proposed regulations in Europe impacting data transfer fees and open interfaces to facilitate switching between providers (Gans et al., 2023)[57] .

Competitive advantages in the cloud market, as in other service industries, are also influenced by factors such as the ability to manage large-scale infrastructure efficiently, offer flexible pricing models (pay-as-you-go), provide high reliability, and ensure strong security and

compliance (Prygara & Yarosh-Dmytrenko, 2021)[58] (Yevtushenko & Fedorchenko, 2023)[59] (Perebyynis et al., 2024)[60] (Kucherivska, 2024)[61] (Grechan & Bilik, 2023)[62] (Novikova et al., 2024)[63] . The rapid growth of cloud computing necessitates effective resource provisioning and allocation strategies, often involving complex brokerage mechanisms to help users navigate the heterogeneous market and select the most advantageous services (Yao et al., 2025)[64] (Li et al., 2022)[65] . The ability of organizations to effectively utilize big data analytics and organizational innovation, potentially mediated by these capabilities, also contributes to gaining competitive advantage in the cloud era (Korayim et al., 2023)[66] . As the cloud computing paradigm evolves, including the emergence of edge-cloud computing environments, the competitive dynamics among providers will likely continue to shift, driven by innovation and strategic positioning (Yao et al., 2025)[64] (Marcelino et al., 2025)[67] (Liu et al., 2023)[48] .

Furthermore, optimizing operational costs, particularly in energy-intensive data centers, is becoming increasingly important for cloud service providers to maintain competitiveness (Guo et al., 2025)[68] .

C- Modeling the volatility of companies' stock returns:

Modeling the volatility of companies' stock returns offers profound benefits for informing investment strategy decisions. It moves beyond simple historical observations to provide a more dynamic and forward-looking understanding of risk, which is crucial for modern finance. Here's a detailed breakdown of these benefits:

Benefits of Modeling Volatility for Investment Strategy Decisions:

1. Enhanced Risk Management and Capital Allocation:

Accurate Risk Assessment: Volatility models, particularly GARCH-type models, capture the time-varying nature of financial market volatility (e.g., volatility clustering, where high volatility periods are followed by

high volatility, and low by low). This provides a more realistic and often predictive measure of risk compared to static historical standard deviations. Tsay, R. S. (2010)[69].

Improved Value at Risk (VaR) and Expected Shortfall (ES) Calculations: These widely used risk metrics rely heavily on accurate volatility forecasts. By using sophisticated models, financial institutions and investors can generate more reliable VaR and ES figures, leading to better capital allocation decisions and adherence to regulatory requirements, Hull, J. C. (2018)[70].

Dynamic Risk Budgeting: Investors can dynamically adjust their risk exposure across different asset classes or strategies based on forecasted volatility. During periods of anticipated high volatility, they might reduce their exposure to risky assets, and vice versa Moreira, A., & Muir, R. (2017)[71].

Stress Testing and Scenario Analysis: Volatility models enable the simulation of portfolio performance under extreme market conditions. This helps investors identify potential vulnerabilities in their portfolios and plan for severe downturns.

2. Optimized Portfolio Construction and Diversification:

Efficient Frontier Construction: Modern Portfolio Theory (MPT) relies on expected returns, volatilities, and correlations to construct optimal portfolios. Modeling individual asset volatilities and their co-movements (correlations) allows for more robust and forward-looking portfolio optimization, leading to a more efficient allocation of capital among diverse assets , Bodie, Z., Kane, A., & Marcus, A. J. (2020)[72].

Dynamic Diversification: During periods of high market stress, correlations between assets often increase, reducing the benefits of diversification. Volatility models can capture these changing

correlations, allowing investors to adjust their diversification strategies dynamically to maintain desired risk levels Engle, R. (2002) [73].

3. Enhanced Derivatives Pricing and Hedging Strategies:

Accurate Option Pricing: Volatility is a critical input in options pricing models (e.g., Black-Scholes). Modeled volatility (especially implied volatility derived from option prices) provides a more accurate reflection of market expectations about future price swings, leading to more precise option valuations Hull, J. C. (2018)[74].

Effective Hedging: Investors use volatility forecasts to determine the appropriate size and type of derivatives contracts needed to hedge specific risks in their portfolios. For instance, if higher volatility is expected, more protective puts might be considered.

Volatility Trading: Specialized strategies exist that directly trade volatility itself (e.g., VIX futures and options). Modeling volatility is fundamental for identifying opportunities and managing risk in these strategies.

4. Informing Trading Strategy Development and Execution:

Adaptive Position Sizing: Traders can use volatility forecasts to dynamically adjust the size of their positions. During high volatility, smaller positions might be taken to manage risk, while larger positions might be justified during low volatility.

Market Timing Signals: Changes in modeled volatility can sometimes precede significant market movements or regime shifts (e.g., from calm to turbulent). While not a perfect predictor, it can provide valuable signals for adjusting market exposure or strategy.

Algorithmic Trading: Volatility forecasts are frequently integrated into algorithmic trading systems to fine-tune execution strategies, manage risk exposure, and optimize trade sizing in real-time.

5. Better Performance Evaluation:

Risk-Adjusted Performance Measurement: Simply looking at raw returns can be misleading. Volatility models provide the necessary input for calculating risk-adjusted performance metrics like the Sharpe Ratio or Sortino Ratio, which allow for a more meaningful comparison of investment strategies by accounting for the level of risk taken Bodie, Z., Kane, A., & Marcus, A. J. (2020)[75].

6. Deeper Insights into Market Dynamics:

Understanding Market Sentiment: Implied volatility (derived from option prices) can act as a "fear gauge" (e.g., the VIX index). By modeling and comparing realized volatility with implied volatility, investors gain

insights into collective market expectations and sentiment, which can inform strategic decisions, Peter Gratton, Cboe Volatility Index (VIX)(2025)[76].

Identification of Market Regimes: Shifts in modeled volatility can indicate transitions between different market regimes (e.g., periods of growth, recession, crisis), allowing investors to adapt their strategies to prevailing conditions.

In conclusion, modeling the volatility of stock returns, especially using advanced econometric techniques like GARCH, transforms raw price data into sophisticated insights about risk and market behavior. This enables investors to make more informed, dynamic, and risk-aware decisions across all facets of their investment strategies.

Expected Outcomes:

Identification of Leading Companies:

After what was shown in the Overview section, it became clear to us that Amazon and Microsoft as the two dominant players in the global cloud services market, with respective market shares of 30% and 21% in Q4 2024 (Richter, 2025)[35].

Competitive Advantage Profiling:

For Amazon: Highlight its position as a market leader, noting its broad and deep range of services, global infrastructure, and innovation capabilities in AI/ML and HPC (Harika, 2024)[36]; also emphasize its mature ecosystem, strong performance, and robust security (Sochat et al., 2025)[37]; (Pacios et al., 2025)[38]; (Ahuja et al., 2025)[39]; (Guptha et al., 2021)[40]; (Rath et al., 2019)[41].

For Microsoft: Outline its competitive strengths in data analytics, machine learning, and support for open-source technologies like Kubernetes (Borra, 2024)[45]; mention its pricing competitiveness, performance variability, and networking capabilities (Ahuja et al., 2025)[39]; (Lin et al., 2025)[46]; as well as its ongoing global expansion (Sajid et al., 2022)[34].

Statistical Modeling Objective:

Develop and apply statistical models to analyze the volatility of Amazon and Microsoft's daily stock returns. We will remedy that in Data analysis section.

Analytical Interpretation:

Interpret the outcomes of the statistical modeling in the context of competitive positioning and market share, to drive investment strategic decisions.

Data analysis:

This section includes:

A- Introduction

B- Data Sources

C- Data Gathering

D- Define The Observations

E- Data Wrangling

F- Exploratory

G- Building models to make report contains forecasts time-varying volatility and strategic investment scenarios:

A- Introduction:

We will develop and apply statistical models to analyze the volatility of Amazon and Microsoft's stock returns.

NOTE: For technical details, please refer to the Appendix section, which contains the PYTHON language codes for this analysis, and the datasets are also attached to this report.

B- Data Sources:

Source: collected recent observations from

https://www.alphavantage.co/

Method of gathering: Stock Market Data API Requesting via AlphaVantageAPI

C- Data Gathering:

We create a URL to get recent 2500 observations the stock data from 30-9-2015 to 9-9-2025 for Microsoft and Amazon from AlphaVantage website and save them as Data Frame called df_microsoft for Microsoft and df_amazon for Amazon using python language :

```
df_microsoft type: <class 'pandas.core.frame.DataFrame'>
df_microsoft shape: (2500, 5)
```

date	open	high	low	close	volume
2025-09-09	501.430	502.250	497.70	498.41	14410542.0
2025-09-08	498.105	501.195	495.03	498.20	16771015.0
2025-09-05	509.070	511.970	492.37	495.00	31994846.0
2025-09-04	504.300	508.150	503.15	507.97	15509486.0
2025-09-03	503.790	507.790	502.32	505.35	15995154.0

```
df_amazon type: <class 'pandas.core.frame.DataFrame'>
df_amazon shape: (2500, 5)
```

date	open	high	low	close	volume
2025-09-09	236.355	238.8500	235.08	238.24	27033778.0
2025-09-08	234.940	237.6000	233.75	235.84	33947104.0
2025-09-05	235.190	236.0000	231.93	232.33	36721802.0
2025-09-04	231.185	235.7700	230.78	235.68	59391779.0
2025-09-03	225.210	227.1699	224.36	225.99	26355706.0

D-Define The Observations:

Column Name	Meaning
Date	The date and/or time of the record.
Open	The price of the stock at the beginning of the trading period.
High	The highest price reached during the trading period.
Low	The lowest price reached during the trading period.
Close	The final price at the end of the trading period.
Volume	The total number of shares traded during that period.

We note that stock returns is not available ,so we will calculate it.

E- Data Wrangling

After inspecting of datasets for Microsoft and Amazon stock data if they have any missing data , we create calculate returns stock according to plot that shows us there no logarithm - transformation is needed and volatility clustering is visible of returns stockTsay, R. S. (2010)[69].

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{\Delta P}{P_{t-1}}$$

Where :

R_t=return at time t

P_t=close price at time t

P_{t-1}=close price at previous time t-1

This gives the percentage change between two time periods.

Then we create Amazon and Microsoft datasets that contains most recent returns of 2500 observations using python codes:

```
y_amazon type: <class 'pandas.core.series.Series'>
y_amazon shape: (2500,)
```

date	return
2015-09-30	3.144717
2015-10-01	1.768718
2015-10-02	2.269934
2015-10-05	2.091862
2015-10-06	-1.140377

dtype: float64

```
y_microsoft type: <class 'pandas.core.series.Series'>
y_microsoft shape: (2500,)
```

date	return
2015-09-30	1.749540
2015-10-01	0.927602
2015-10-02	2.151984
2015-10-05	2.326092
2015-10-06	0.257345

dtype: float64

F- Exploratory

We will explore the volatility of returns for Microsoft and Amazon.

Variance (which is sometimes called volatility in finance), is a measurement of how spread out the points in a dataset are around the mean. It's calculated by first summing up the squared different between each data point and mean, then dividing by the number of data points minus one .In other words, volatility in finance is the same thing at standard deviation in statistics.

$$\operatorname{Var}(X) = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \overline{X})^2$$

Let's create the volatility time series plot so that we have a visual aid to talk about what volatility is.

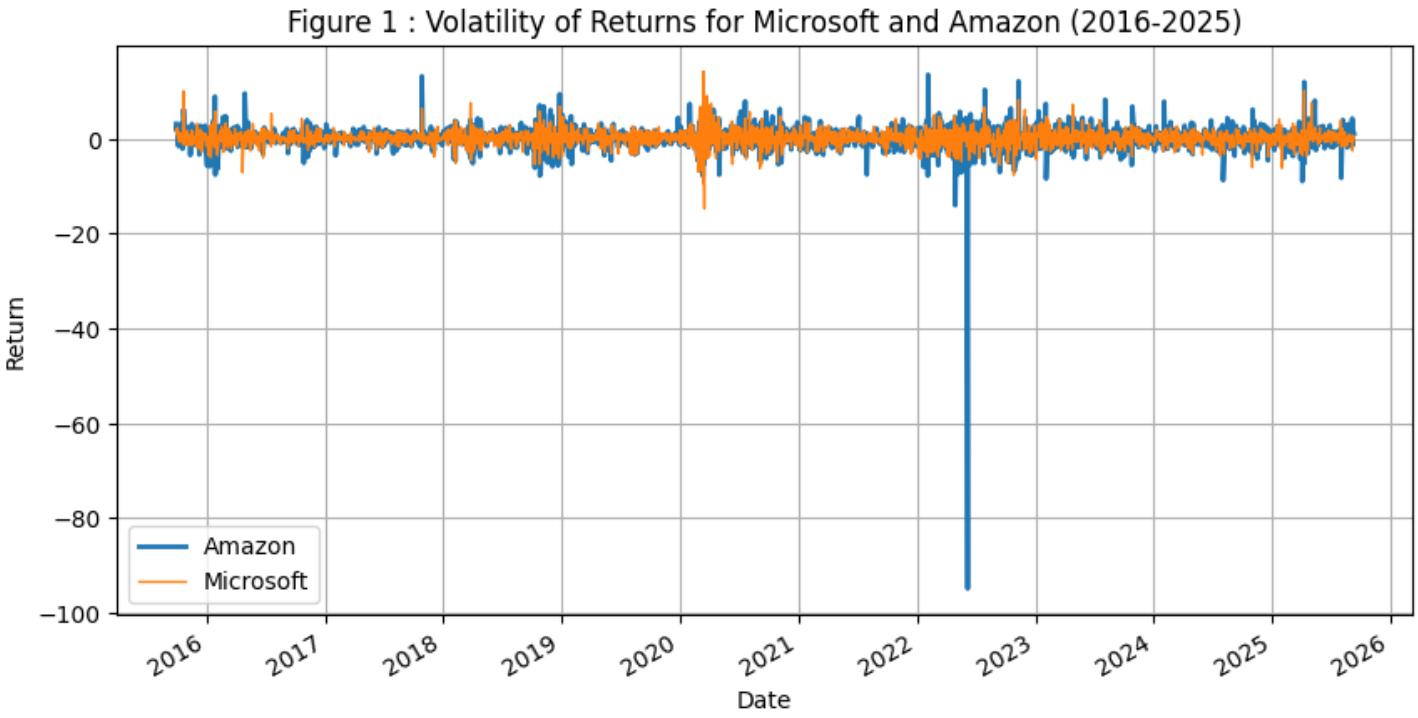
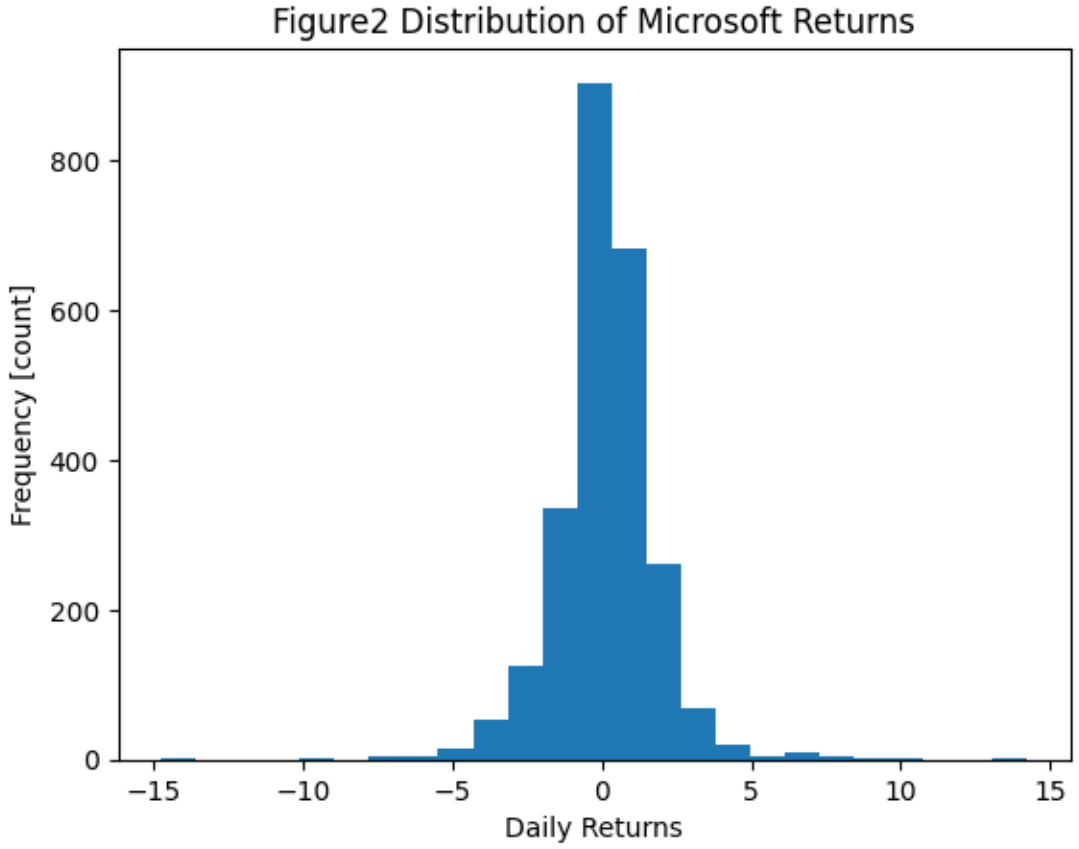


Figure 1 illustrates the volatility of returns for Microsoft and Amazon from 2016 to 2025. Both companies exhibit fluctuating returns, with periods of high volatility. Notably, Amazon experiences a severe decline in returns around 2023, which is more pronounced compared to Microsoft's performance during the same period. The chart effectively highlights the dynamic nature of stock returns and the differing risk profiles of the two companies over the

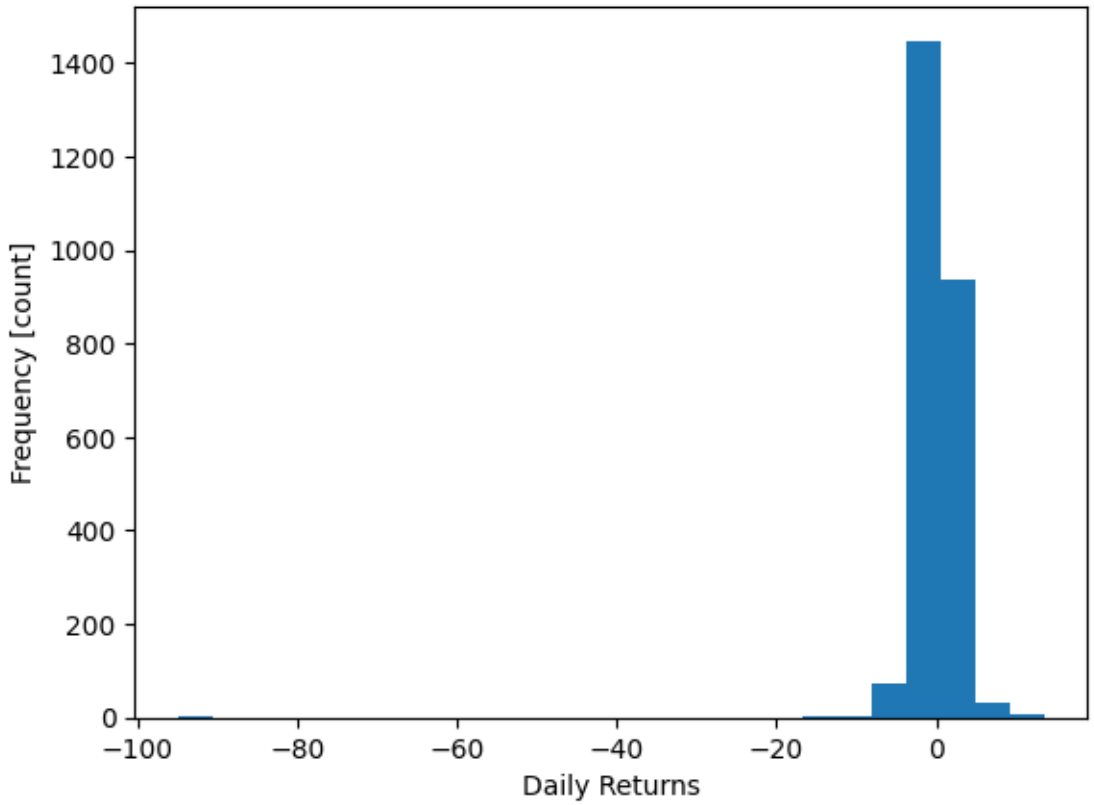
given timeframe. This visualization is valuable for analyzing market behavior, assessing investment risks, and understanding how external factors might impact the financial performance of major tech companies.

Let's create a histogram about Distribution of Microsoft Daily Returns .



Let's create a histogram about Distribution of Amazon Daily Returns .

Figure 3 Distribution of Amazon Returns



We observe that two shapes above(Figure2 and Figure3) turn out that returns follow *an almost normal distribution, centered on 0.* Volatility is the measure of the spread of these returns around the mean.

Let's start by measuring the daily volatility of our two stocks. Since our data frequency is also daily, this will be exactly the same as calculating the standard deviation.

```
Microsoft Daily Volatility: 1.7018376833451128
Amazon Daily Volatility:    2.8058045528789792
```

Looks like Amazon is more volatile than Microsoft. This reinforces what we saw in our Figure 1, where Amazon returns have a much wider spread. While daily volatility is useful, investors are also interested in volatility over other time periods — like annual volatility. Keep in mind that a year isn't 365 days for a stock market,

though. After excluding weekends and holidays, most markets have only 252 trading days.

Let's calculate the annual volatility for Microsoft and Amazon:

```
Microsoft Annual Volatility: 27.015835691576747
Amazon Annual Volatility:    44.540766446223344
```

Again, Amazon has higher volatility than Microsoft.

Since we're dealing with time series data, another way to look at volatility is by calculating it using a rolling window.In financial time-series analysis, rolling windows (with sizes like 20, 50, 100, or even 200) are frequently used to compute technical indicators (moving average, standard deviation, momentum, or volatility). This practice is well documented in financial econometrics and technical analysis texts (Alexander, C. (2001))[77].

For financial return time series, it's very common to compute 50-period rolling statistics, for instance:

Moving average of returns (to gauge momentum).

Rolling standard deviation (volatility).

Rolling beta or correlation with another series.

So 50 for 2,500 daily return observations (about 2% of the data) falls nicely within standard practice — it's frequently used in financial applications.

Let's Create a time series plot showing the daily returns for Amazon and Microsoft the 50-day rolling volatility.

Figure 4 : Volatility of Returns for Amazon(2016-2025)

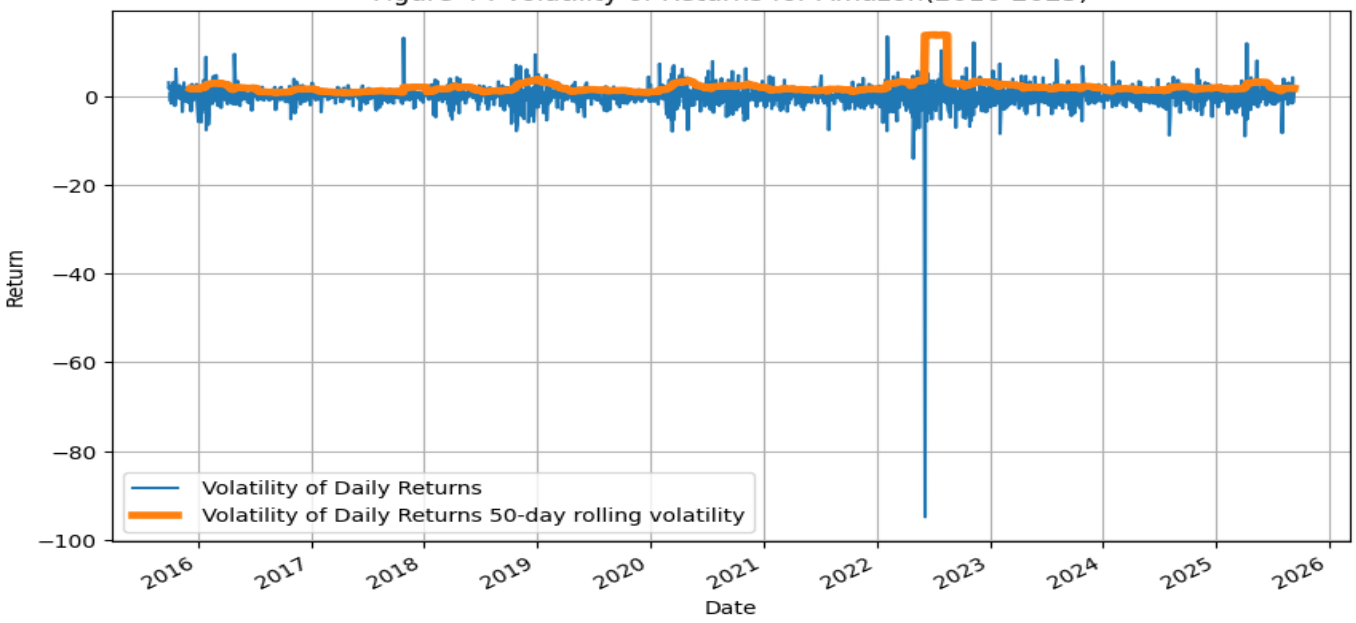


Figure 5 : Volatility of Returns for Microsoft(2016-2025)

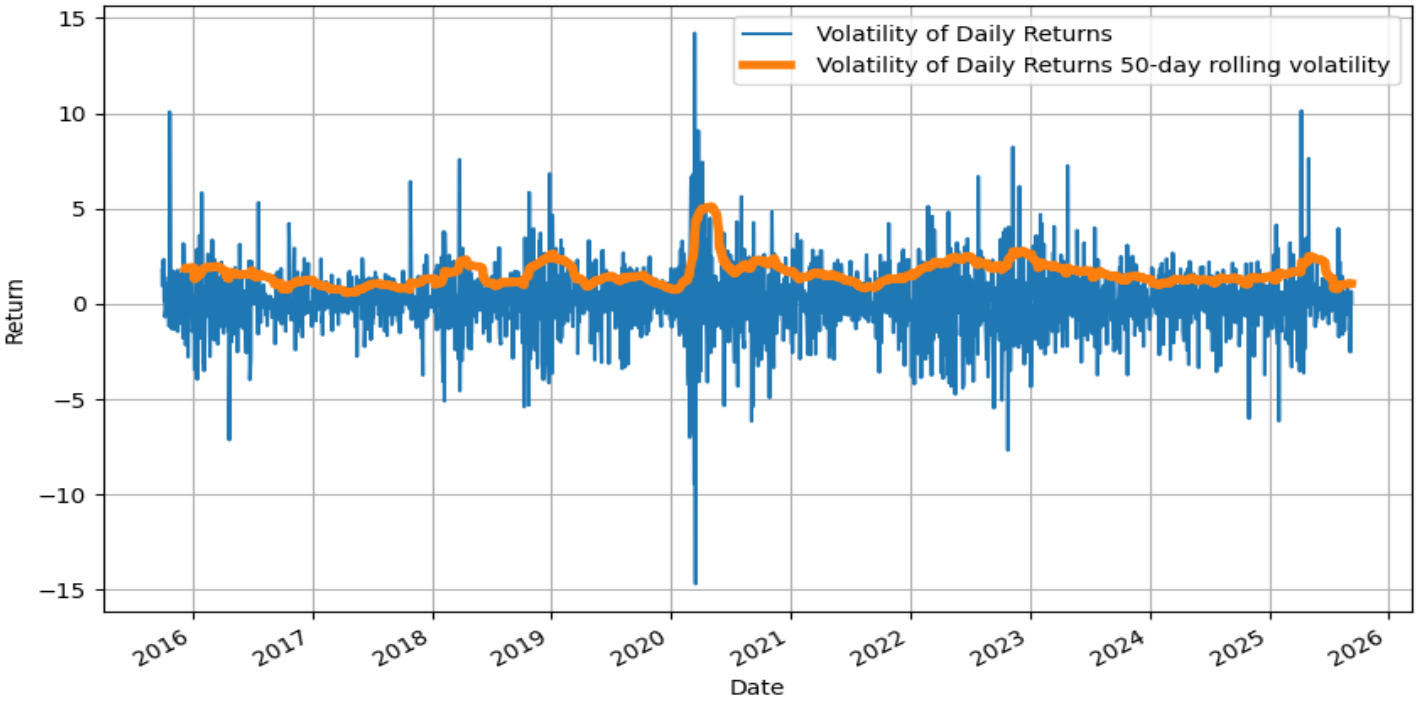
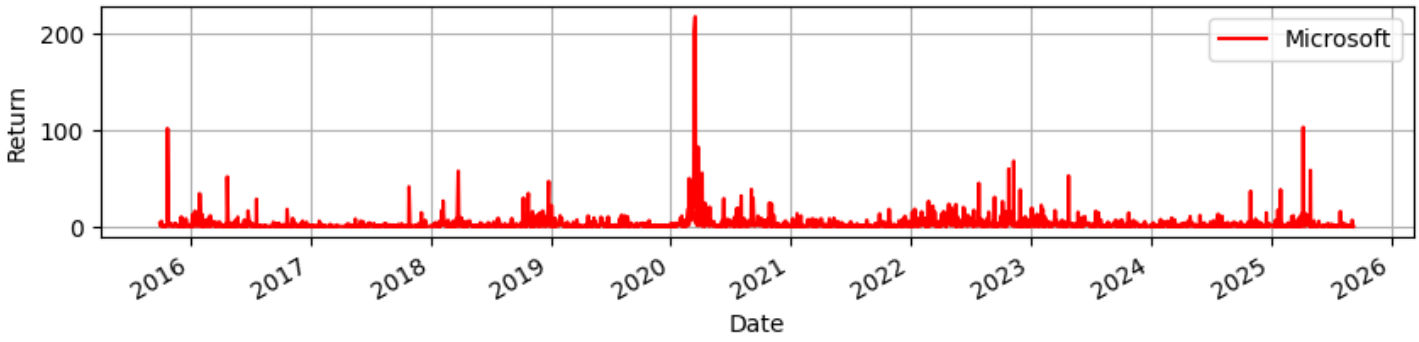
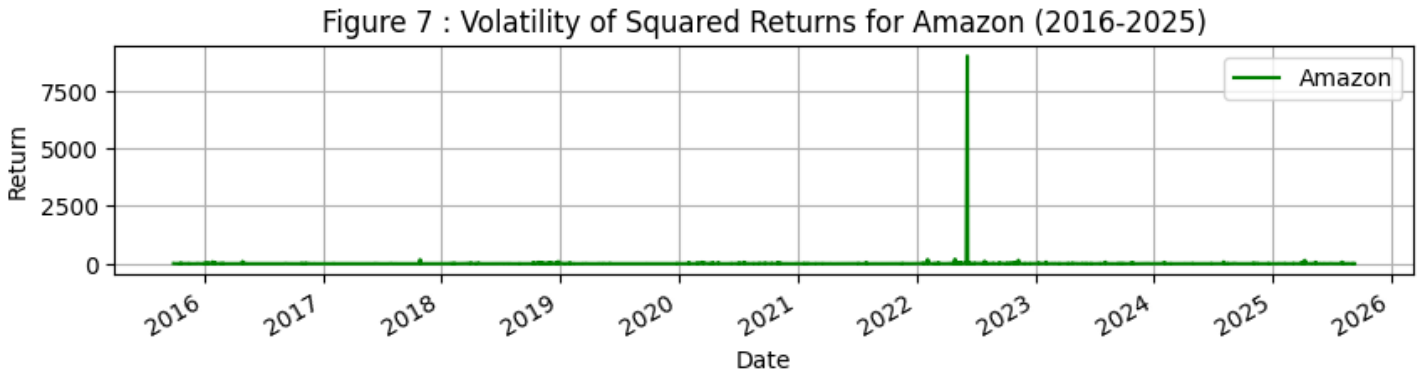


Figure4 and figure5 above reveal a problem. We want to use returns to see if high volatility on one day is associated with high volatility on the following day. But high volatility is caused by large changes in returns, which can be either positive or negative. How can we assess negative and positive numbers together without them canceling each other out? One solution is to take the absolute value of the numbers, which is what we do to calculate performance metrics like mean absolute error. The other solution, which is more common in this context, is to square all the values.

Let's create a time series plot of the squared stock returns for Amazon and Microsoft .

Figure 6 : Volatility of Squared Returns for Mirosoft(2016-2025)





Perfect! from figure6 and figure7 it's much easier to see that we have periods of high and low volatility, and high volatility days *tend to cluster together and no logarithm-transformation is needed* ,Tsay, R. S. (2010)[69]. In addition, the returns of Amazon and Microsoft *follow an almost normal distribution, centered on 0* (from Figure2 and Figure3 please review above) , *this is a perfect situation to use a GARCH model.* The GARCH model (Generalized Autoregressive Conditional Heteroskedasticity) is used to model and forecast volatility in time series data, especially when the variance (volatility) changes over time in a predictable way. So, in next point of this section, we will build GARCH

models to forecasts time-varying volatility (variance) in time series of stock daily returns for Microsoft and Amazon .

G-Building models to make report contains forecasts time-varying volatility and strategic investment scenarios:

As we mentioned above ,the GARCH model (Generalized Autoregressive Conditional Heteroskedasticity) is used to model and forecast volatility in time series data, especially when the variance (volatility) changes over time in a predictable way.

GARCH(p, q) Model Equation Tsay, R. S. (2010)[69]:

Where:

- σ_t^2 = Conditional variance at time t
- ω = Constant
- ε_{t-i}^2 = Past squared residuals (ARCH effects)
- σ_{t-j}^2 = Past conditional variances (GARCH effects)
- α_i, β_j = Coefficients

Let's start to apply the following methodology using python language codes to build GARCH models to forecasts time-varying volatility (variance) in time series of stock daily returns for Microsoft and Amazon for strategic investment scenarios:

- Data Splitting: The dataset is split chronologically into 80% training and 20% testing (For the purposes of preparing this report on strategic investment decisions, we split the dataset of daily returns into a training set comprising 80% of the data and a testing set comprising the remaining 20%. This approach allows us to fit and calibrate the GARCH model using historical data while evaluating its forecasting performance on unseen data, ensuring the robustness and relevance of the model for forward-looking investment decisions.)
- Model Fitting: A GARCH(1,1) model with a Student's t-distribution (a probability distribution that is used to estimate the mean of a

normally distributed population when the sample size is small and/or when the population standard deviation is unknown) is fitted to the training data.

- Diagnostics:
  - Ljung-Box test: Checks for autocorrelation in standardized residuals (p-values > 0.05 indicate no significant autocorrelation).
  - ARCH-LM test: Tests for remaining Autoregressive Conditional Heteroskedasticity “ARCH” effects (p-value > 0.05 suggests the model captures volatility clustering).
  - AIC/BIC: AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion, also called Schwarz Criterion) solve this trade-off by penalizing models for having more parameters. Their goal is to find the model that explains the data well without unnecessary complexity. Lower values indicate better model fit.
- Forecasting: 10-day volatility forecasts are generated, converted to standard deviation (volatility), and plotted.

- Value at Risk (VaR) calculation: is a statistical measure that quantifies the level of financial risk within a firm, portfolio, or position over a specific time frame . 1-day VaR at 95% confidence is computed using the forecasted volatility and t-distribution.
- Investment Implications:
  - *High persistence:* (Alpha (α): measures how sensitive today's volatility is to yesterday's news or shocks + Beta (β): measures how much of yesterday's volatility level carries over into today's volatility it represents the long-term memory of the volatility process> 0.9) "α + β > 0.9": Indicates long-lasting volatility, suggesting conservative strategies (e.g., reduced exposure, hedging).
  - *High α (> 0.3):* Recent shocks drive volatility, warranting short-term risk management (e.g., stop-loss).
  - *High β (> 0.7):* Stable volatility regimes, suitable for long-term risk budgeting.

- *Volatility forecasts*: Guide position sizing, VaR/ Expected Shortfall (ES), and hedging decisions.
- *VaR:* Provides a loss threshold for risk management.

- Strategic Investment Scenarios

After make investment implications based on the GARCH parameters and forecast , we can make the scenarios for strategic investment decisions:

1- High Volatility Persistence ($\alpha + \beta > 0.9$):

- Implication: Volatility shocks persist, increasing risk over longer horizons.
- Action: Reduce exposure to stock, allocate to lower-volatility assets, or use options for hedging (e.g., protective puts).
- Example: If persistence is 0.95, expect volatility to remain elevated after shocks, so maintain robust risk controls.

2- High ARCH Effect ($\alpha > 0.3$):

- Implication: Recent price shocks significantly impact volatility, indicating potential for sudden risk spikes.

- Action: Implement stop-loss orders or reduce position sizes during high-volatility periods. Monitor market news for catalysts.
- Example: If $\alpha = 0.4$, a large return on day t increases next-day volatility, prompting tighter risk management.

3- High GARCH Effect ($\beta > 0.7$):

- Implication: Volatility is driven by past volatility, suggesting stable risk regimes.
- Action: Use for long-term risk budgeting in portfolios. Allocate capital based on stable volatility forecasts.
- Example: If $\beta = 0.8$, volatility remains consistent, allowing predictable risk-adjusted returns.

4- Low Degrees of Freedom ($\nu < 5$):

- Implication: Fat-tailed distribution, indicating higher risk of extreme returns.
- Action: Increase hedging (e.g., options) and focus on tail-risk metrics like Expected Shortfall.
- Example: If $\nu = 4$, expect frequent large price movements, requiring conservative position sizing.

5- Volatility Forecast Insights:

- Implication: Forecasted volatility (e.g., 1.5% daily) informs VaR/ES and position sizing.
- Action: Adjust leverage or exposure based on forecasted volatility. High forecasts (> historical average) suggest reducing risk.
- Example: If day-1 volatility is 1.5% vs. a historical average of 1%, consider reducing position size or hedging.

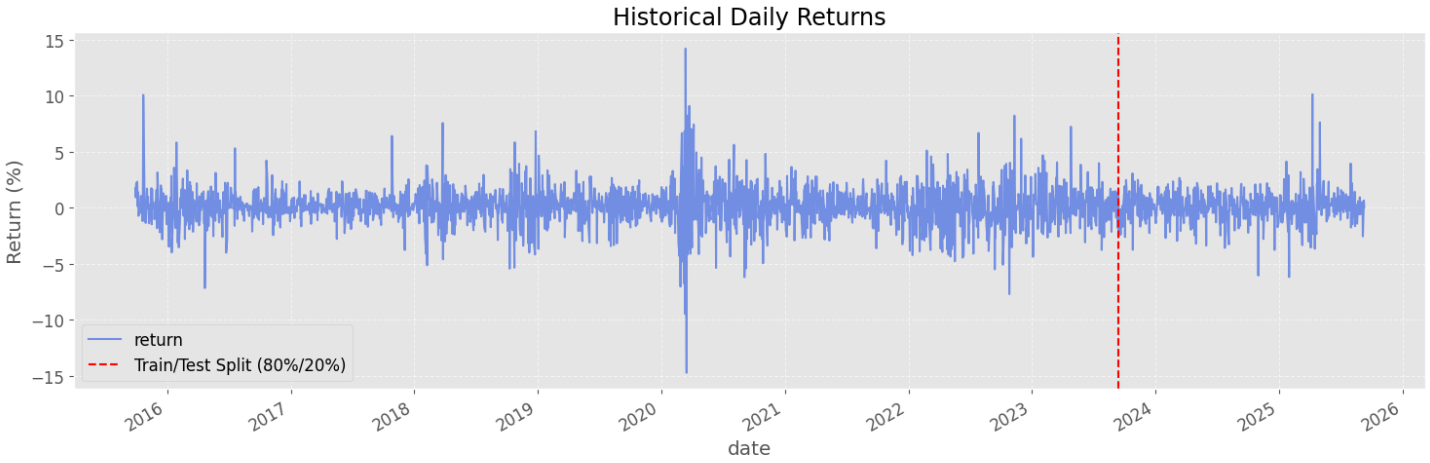
6- VaR and Risk Management:

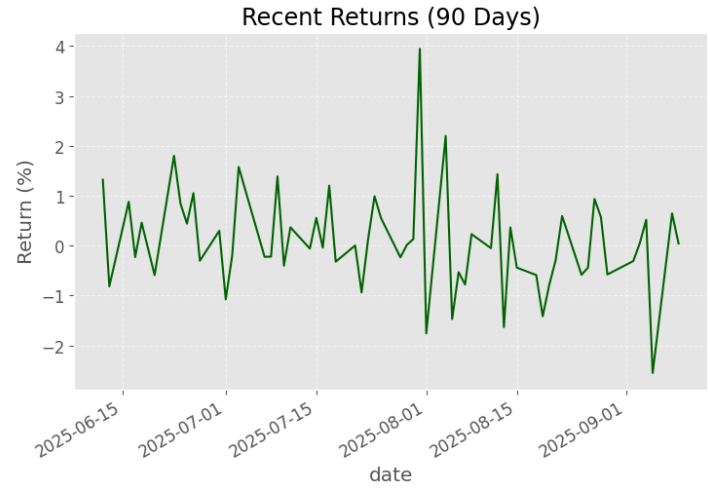
- Implication: VaR (e.g., -2.5% at 95% confidence) sets a loss threshold.
- Action: Use VaR to set stop-loss levels or allocate capital to stay within risk limits.
- Example: If VaR is -2.5%, ensure portfolio loss limits align with this threshold.

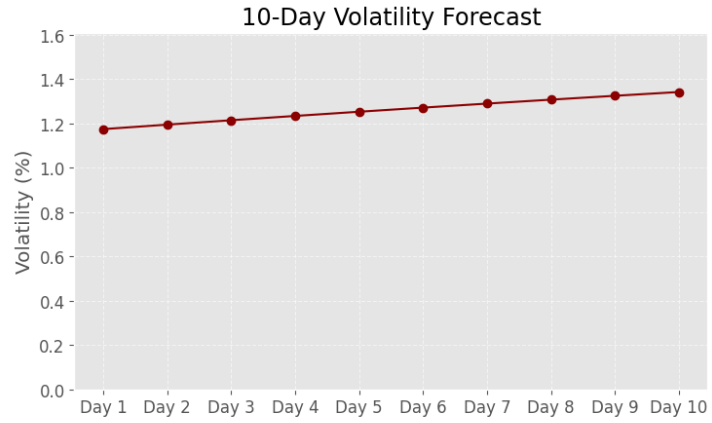
- # MICROSOFT STOCK VOLATILITY ANALYSIS REPORT

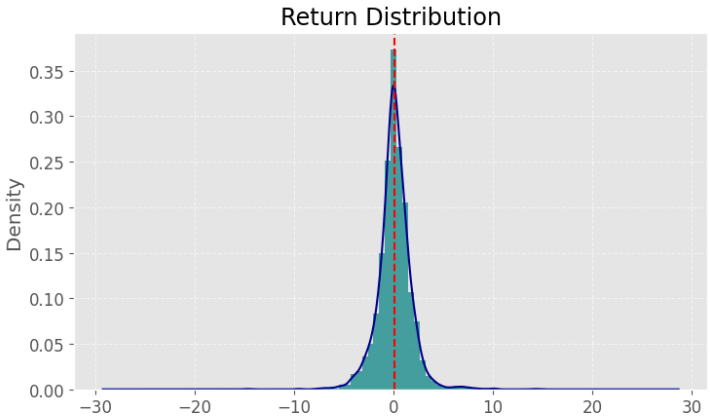
Summary:

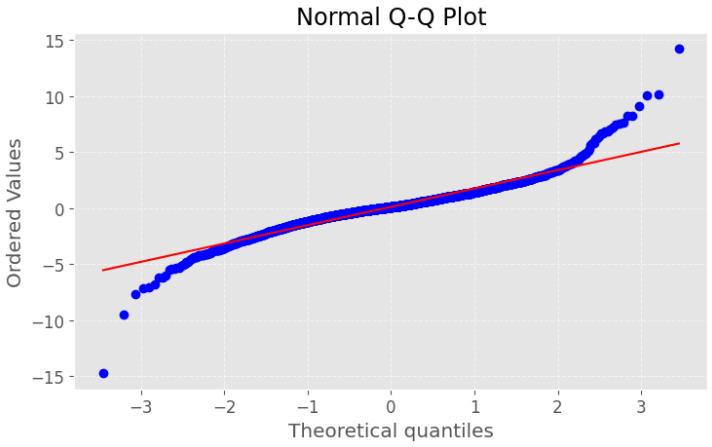
Microsoft demonstrates high volatility persistence ($\alpha+\beta = 0.9934$), meaning market shocks will have extended effects before decaying. The model forecasts gradually rising short-term volatility (1.17% → 1.34% over 10 days), though levels remain below prior peaks. Fat-tailed behavior ($\nu = 4.84$) signals elevated probability of extreme returns, while a low 1-day VaR (95% confidence) of 0.09% indicates manageable daily loss exposure. Strategically, investors should prioritize tail-risk hedging, reduced exposure to volatility-sensitive positions, and tighter stop-loss controls. The strong GARCH component ($\beta = 0.86$) reinforces stable long-term volatility regimes, supporting strategic portfolio risk budgeting.


Historical Daily Returns
Return (%)
return
Train/Test Split (80%/20%)
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
date


Recent Returns (90 Days)
Return (%)
2025-06-15
2025-07-01
2025-07-15
2025-08-01
2025-08-15
2025-09-01
date


10-Day Volatility Forecast
Volatility (%)
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Day 9
Day 10


Return Distribution
Density


Normal Q-Q Plot
Ordered Values
Theoretical quantiles

Model Parameters and Diagnostics:

Parameter	Value	Interpretation
Omega (ω)	0.057268	Baseline volatility level
Alpha (α)	0.130045	Sensitivity to recent shocks
Beta (β)	0.863333	Long-term volatility memory
Persistence (α+β)	0.9934	Prolonged volatility shock duration (>0.9)
Degrees of Freedom (ν)	4.84	Fat-tailed returns (ν < 5 = higher extremes)
AIC	7249.51	Good model fit (lower = better)
BIC	7277.51	Good model fit (lower = better)
ARCH-LM p-value	0.6148	No residual ARCH effects (>0.05 adequate)
1-Day VaR (95%)	0.09%	Daily loss threshold
Day 1 Volatility Forecast	1.17%	Elevated short-term volatility

Key Diagnostics

- **High persistence (α+β ≈ 0.99):** Confirms volatility shocks decay slowly.
- **Fat tails (ν < 5):** Signals increased probability of large swings.
- **ARCH-LM p-value (>0.05):** Confirms model adequacy (no remaining clustering).

Strategic Investment Implications:

Factor	Implication	Action
High Volatility Persistence ($\alpha+\beta>0.9$)	**Long-lasting market shocks**	**Reduce exposure; hedge with protective puts**
High GARCH Term ($\beta>0.7$)	**Stable volatility regimes**	**Enable long-term risk budgeting**
Fat Tails ($\nu<5$)	**Extreme return risk**	**Hedge tail risks; monitor outliers**
Forecasted Volatility (1.17% → 1.34%)	**Rising short-term uncertainty**	**Trim position sizes; deploy short-term hedges**
Low 1-Day VaR (0.09%)	**Manageable daily risk threshold**	**Define stop-losses; align capital allocation**

10-Day Volatility Forecast

Day 1: 1.17% → Day 10: 1.34% (steady upward trend).

Conclusions:

1- **Rising Risk in Near Term: Volatility is expected to increase gradually, warranting defensive positioning.**

2- **Tail-Risk Exposure: Fat tails elevate the probability of outsized market swings; hedging is essential.**

3- **Risk Management Priority: VaR provides a basis for daily stop-loss calibration.**

4- **Stable Long-Term Outlook: Strong β coefficient (0.86) supports risk budgeting for strategic investors.**

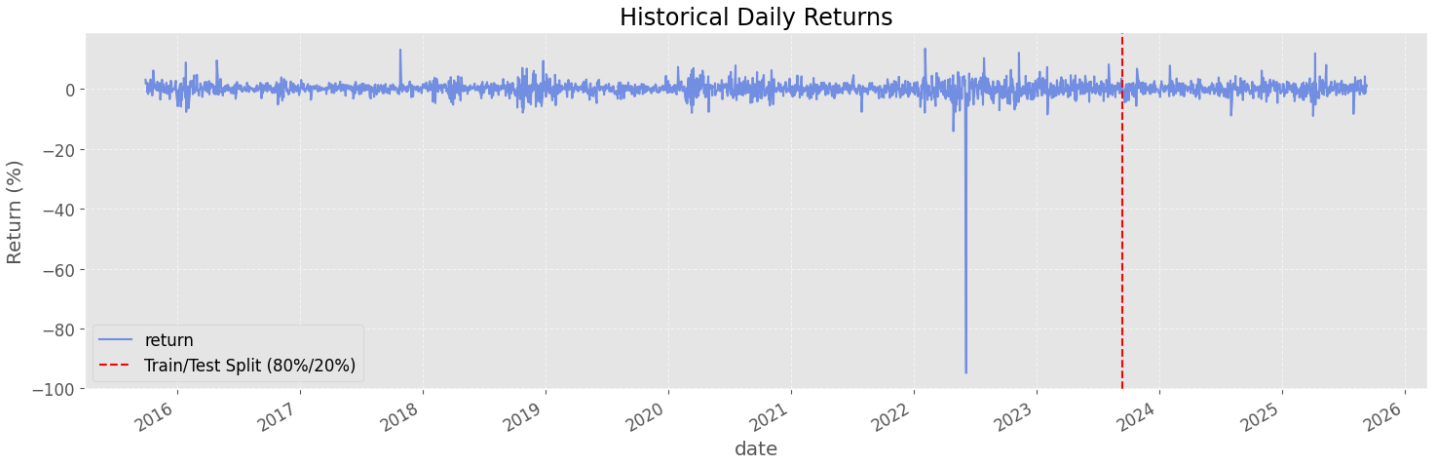
Recommendations:

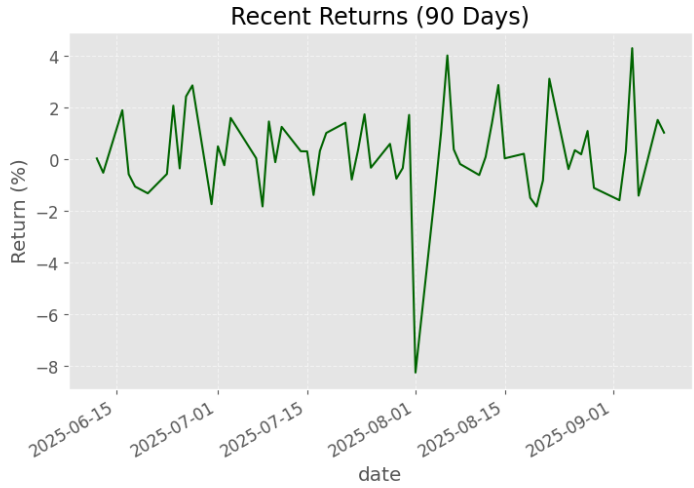
- **Short-term: Reduce positions in high-volatility exposures; use options for hedging.**
- **Long-term: Rebalance portfolios with volatility forecasts in mind; diversify to mitigate fat-tail risks.**

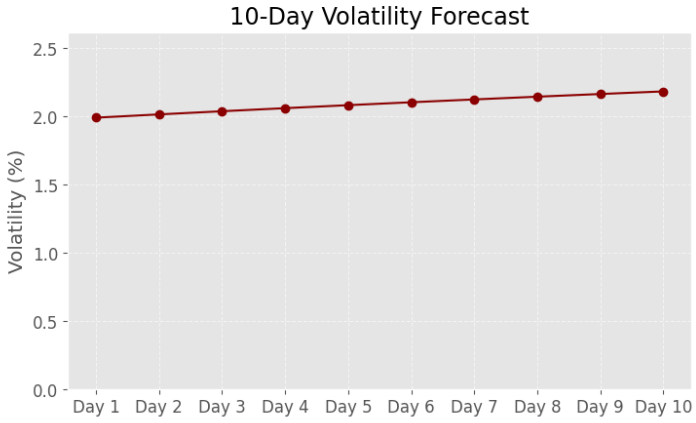
- # AMAZON STOCK VOLATILITY ANALYSIS REPORT

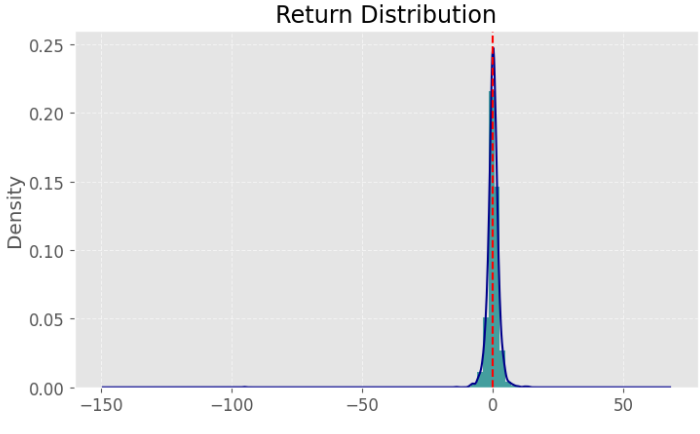
Summary:

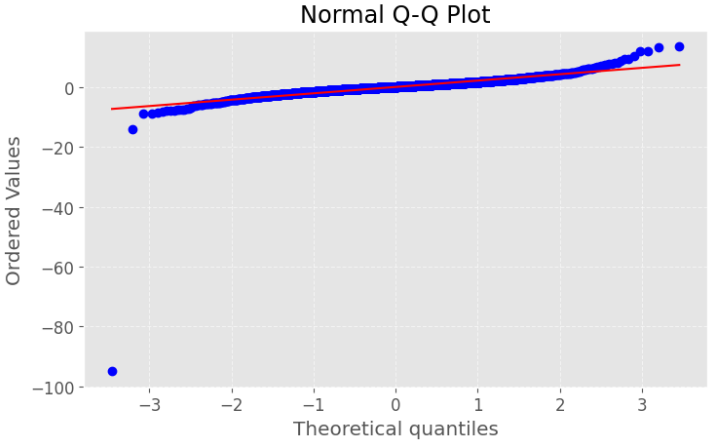
Amazon shows high volatility persistence ($\alpha+\beta = 0.9822$), confirming that market shocks have extended effects. The model forecasts elevated near-term volatility (1.99% → 2.18% over 10 days), well above historical averages (≈1.48%). Fat-tailed returns ($\nu = 4.21$) indicate a heightened risk of extreme price movements. Despite this, the 1-day VaR (95%) remains ultra-low at 0.02%, suggesting limited daily downside in normal conditions. Strategic actions include reducing exposure, deploying protective puts, and reinforcing tail-risk hedging, as the rising volatility path highlights increased uncertainty. Long-term investors can rely on the stable GARCH component ($\beta = 0.82$) for structured risk budgeting.


Historical Daily Returns
Return (%)
0
−20
−40
−60
−80
−100
return
Train/Test Split (80%/20%)
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
date


Recent Returns (90 Days)
Return (%)
4
2
0
−2
−4
−6
−8
2025-06-15
2025-07-01
2025-07-15
2025-08-01
2025-08-15
2025-09-01
date


10-Day Volatility Forecast
Volatility (%)
2.5
2.0
1.5
1.0
0.5
0.0
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Day 9
Day 10


Return Distribution
Density
0.25
0.20
0.15
0.10
0.05
0.00
−150
−100
−50
0
50


Normal Q-Q Plot
Ordered Values
0
−20
−40
−60
−80
−100
−3
−2
−1
0
1
2
3
Theoretical quantiles

Model Parameters and Diagnostics:

Parameter	Value	Interpretation
Omega (ω)	0.166498	Baseline volatility level
Alpha (α)	0.165878	Sensitivity to recent shocks
Beta (β)	0.816288	Long-term volatility memory
Persistence (α+β)	0.9822	Prolonged volatility shock duration (>0.9)
Degrees of Freedom (ν)	4.21	Fat-tailed returns (ν < 5 = higher extremes)
AIC	8051.10	Good model fit (lower = better)
BIC	8079.10	Good model fit (lower = better)
ARCH-LM p-value	1.0000	No residual ARCH effects (>0.05 adequate)
1-Day VaR (95%)	0.02%	Daily loss threshold
Day 1 Volatility Forecast	1.99%	Elevated near-term volatility

Key Diagnostics

- **High persistence (α+β ≈ 0.98):** Volatility shocks decay slowly, extending risk exposure.
- **Fat tails (ν = 4.21 < 5):** Suggests elevated probability of outsized returns.
- **ARCH-LM test (p=1.0):** Confirms model adequacy (no remaining clustering).

Strategic Investment Implications:

Factor	Implication	Action
High Volatility Persistence ($\alpha+\beta>0.9$)	Long-lasting market shocks	Reduce exposure; hedge with protective puts
High GARCH Term ($\beta>0.7$)	Stable volatility regimes	Enable long-term risk budgeting
Fat Tails ($\nu<5$)	Extreme return risk	Hedge tail risks; monitor outliers
Forecasted Volatility (1.99% → 2.18%)	Rising short-term uncertainty	Trim position sizes; deploy short-term hedges
Low 1-Day VaR (0.02%)	Manageable daily risk threshold	Define stop-losses; align capital allocation

10-Day Volatility Forecast

Day 1: 1.99% → Day 10: 2.18% (clear upward trend).

Conclusions:

1- **Acute Near-Term Risk: Volatility is projected to rise significantly, requiring defensive strategies.**

2- **Tail-Risk Exposure: Fat-tailed distribution underscores vulnerability to extreme moves; tail hedges are critical.**

3- **Persistence Challenge: High $\alpha+\beta$ (0.9822) indicates extended volatility shocks, requiring sustained risk controls.**

4- **Daily Risk Contained: Despite high volatility, VaR remains minimal, enabling precise stop-loss planning.**

Recommendations:

- Short-term: Cut exposure in volatility-sensitive assets, deploy tail-risk protection (e.g., Out-of-The-Money Put Options, variance swaps).

- Long-term: Rebalance portfolios with volatility forecasts in mind; diversify into low-correlation assets.

- Ongoing Monitoring: Track ν for tail-thickness shifts; revalidate model if $\nu < 4.0$ or persistence changes materially.

Strategic Proposals (Strategic Investment Scenarios):

Based on volatility diagnostics and cloud market dominance, we propose:

1. Short-Term Investment Strategy

- **Reduce** exposure to Amazon and Microsoft during forecasted periods of elevated volatility (Amazon: 1.99% → 2.18%, Microsoft: 1.17% → 1.34%).
- **Implement protective hedging,** such as put options, variance swaps, or stop-loss orders, especially under conditions of high persistence ($\alpha+\beta \approx 0.99$) and fat tails ($\nu < 5$).
- **Tighten stop-loss thresholds** using each stock's 1-day VaR as a benchmark (Microsoft: 0.09%, Amazon: 0.02%).

2. Long-Term Investment Strategy

- **Rebalance portfolios** in alignment with rolling volatility forecasts to sustain risk-adjusted returns.
- **Prioritize tail-risk management** for Amazon, given its higher near-term volatility path and fat-tail exposure ($\nu = 4.21$).
- **Leverage Microsoft's stable volatility** regime ($\beta = 0.86$) as a foundation for conservative, long-term allocations within diversified portfolios.

3. Portfolio Diversification

- **Diversify into** low-correlation assets or sectors (e.g., healthcare, utilities, commodities) to offset concentration risks tied to tech-sector volatility.
- **Monitor GARCH** diagnostics continuously to identify regime shifts and optimize timing of entries/exits, particularly when persistence levels remain elevated (>0.9).

Limitations:

While the findings of this report offer valuable insights, several limitations should be acknowledged:

1. **Model Assumptions**: The GARCH model assumes conditional heteroskedasticity and specific distributional characteristics (Student's t-distribution), which may not capture all real-world dynamics, especially during structural breaks or black swan events.

2. **Limited Forecast Horizon**: The forecast was restricted to short-term horizons (e.g., 10-day volatility), which may not fully reflect longer-term risk trends relevant to strategic investors.

3. **Market Data Scope**: Only Microsoft and Amazon stocks were analyzed, excluding other cloud market players such as Google Cloud or emerging regional competitors, which may limit comparative depth.

4. **External Factors Ignored**: Macroeconomic indicators (e.g., interest rates, inflation) and geopolitical events were not integrated into the model, yet they could significantly impact volatility and strategic investment decisions.

5. **Historical Bias**: The models are based on historical data, which may not fully capture future disruptions, innovations, or market paradigm shifts in the cloud computing space.

Investors should interpret these findings cautiously, supplementing them with real-time data, broader market analysis, and professional financial advice to address these constraints effectively.

Strategic Investment Action Plan Matrix (with Legend):

Task	Company	Short-Term	Long-Term	Legend	Milestone / Trigger
1. Exposure Management	Amazon	☑ Aggressively reduce >12%	🔁 Reassess quarterly	☑	Volatility > 1.9%, α+β > 0.95
	Microsoft	☑ Trim position (5–7%)	🔁 Maintain with periodic review	☑	α+β > 0.9, Volatility > 1.2%
2. Tail-Risk Hedging	Amazon	☑ Use Out-of-The-Money Put Options puts, variance swaps	🔁 Maintain until ν > 5	☑	ν = 4.21
	Microsoft	☑ Light put hedging	🔁 Roll options quarterly	☑	ν = 4.84
3. Stop-Loss Calibration	Amazon	☑ Tight stop-loss (0.02% VaR)	🔁 Monthly recalibration	☑	VaR breach or Volatility > 2.0%
	Microsoft	☑ Conservative stop-loss (~0.09%)	🔁 Quarterly update	☑	Sustained volatility > 1.3%
4. Diversification	Amazon	☑ Shift into defensive sectors	🔁 Maintain balanced risk	☑	Tech sector allocation > 30%
	Microsoft	☑ Portfolio balance check	🔁 Reinvest in multi-sector (Exchange-Traded Funds)	☑	Correlation > 0.6 with NASDAQ
5. Volatility Monitoring	Amazon	☑ Weekly rolling 50-day check	🔁 Volatility alerts	☑	Rolling vol > 1.5× average
	Microsoft	☑ Biweekly tracking	🔁 Trendband creation	☑	Sudden variance spikes above historical
6. GARCH Diagnostics	Amazon	☑ Weekly re-estimation	🔁 Semi-annual refinement	☑	ν < 4.0, ARCH-LM < 0.05
	Microsoft	☑ Model tracking	🔁 Parameter validation	☑	α+β > 0.98
7. Strategy Reassessment	Amazon	☑ Monthly risk adjustment	🔁 Rebalance semi-annually	☑	Sharpe < 0.5, high drawdowns
	Microsoft	☑ Quarterly snapshot	🔁 Full annual review	☑	α > 0.25 consistently
8. Macro Factor Integration	Amazon	✗ Not included yet	☑ Future enhancement	✗ / ☑	External shocks (Fed policy, geopolitics)
	Microsoft	✗ Not included yet	☑ Explore macro overlay	✗ / ☑	Interest rate or regulatory shocks

Legend

- ☑ = Perform task
- 🔁 = Repeat/update task
- ✗ = Not yet applicable / outside current scope

Strategic Investment Action Plan Matrix (Heatmap view):

Task	Company	Short-Term	Long-Term	Trigger	Risk
Exposure Management	Amazon	Aggressively reduce >12%	Reassess quarterly	Vol > 1.9%, α+β > 0.95	High
Exposure Management	Microsoft	Trim position (5–7%)	Maintain with periodic review	α+β > 0.9, Vol > 1.2%	Medium
Tail-Risk Hedging	Amazon	Use Out-of-The-Money Put Options, variance swaps	Maintain until ν > 5	ν = 4.21	High
Tail-Risk Hedging	Microsoft	Light put hedging	Roll options quarterly	ν = 4.84	Low
Stop-Loss Calibration	Amazon	Tight stop-loss (0.02% VaR)	Monthly recalibration	VaR breach or Vol > 2.0%	High
Stop-Loss Calibration	Microsoft	Conservative stop-loss (~0.09%)	Quarterly update	Sustained Vol > 1.3%	Medium
Diversification	Amazon	Shift into defensive sectors	Maintain balanced risk	Tech sector > 30%	Medium
Diversification	Microsoft	Portfolio balance check	Reinvest in multi-sector (Exchange-Traded Funds)	Correlation > 0.6	Low
Volatility Monitoring	Amazon	Weekly rolling 50-day check	Volatility alerts	Rolling vol > 1.5× avg	High
Volatility Monitoring	Microsoft	Biweekly tracking	Trendband creation	Variance spikes	Medium
GARCH Diagnostics	Amazon	Weekly re-estimation	Semi-annual refinement	ν < 4.0, ARCH-LM < 0.05	High
GARCH Diagnostics	Microsoft	Model tracking	Parameter validation	α+β > 0.98	Medium
Strategy Reassessment	Amazon	Monthly risk adjustment	Rebalance semi-annually	Sharpe < 0.5, drawdowns	High
Strategy Reassessment	Microsoft	Quarterly snapshot	Full annual review	α > 0.25 consistently	Medium
Macro Factor Integration	Amazon	Not included yet	Future enhancement	External shocks	Medium
Macro Factor Integration	Microsoft	Not included yet	Explore macro overlay	Interest rate/geopolitical risk	Medium

- 🔴 **High risk** = Amazon (most tasks)
- 🟠 **Medium risk** = Microsoft (most tasks)
- 🟢 **Low risk** = only applies to certain hedging/diversification tasks

References:

1- Cloud Infrastructure and Enterprise IT Environment, Emmanuel Adeoye* & Babasola Osibo, 2023, https://doi.org/10.51583/IJLTEMAS.2023.12903

2- Merits and Demerits of Cloud Computing for Business, Maduabuchukwu Augustine Onwuzurike, 2024, https://www.ijisrt.com/merits-and-demerits-of-cloud-computing-for-business

3- Cloud Computing and IT Infrastructure Outsourcing: A Comparative Study,Praveen K. Choudhary,Monika Mital,Rajeev Sharma,Ashis K. Pani,2015, https://www.igi-global.com/gateway/article/137718

4- Software-Defined Cloud Infrastructure, R. Mohanasundaram (VIT University, India), A. Jayanthiladevi (Jain University, India), and Keerthana G. (VIT University, India),2018, https://www.igi-global.com/gateway/chapter/204267

5- Cloud Infrastructure, Dan C. Marinescu,2013, https://www.sciencedirect.com/science/article/abs/pii/B9780124046276000038?via%3Dihub

6- Cloud continuum testbeds and next-generation ICTs: Trends, challenges, and perspectives, Fran Casino,Peio Lopez-Iturri,Constantinos Patsakis,2024, https://www.sciencedirect.com/science/article/pii/S1574013724000790?dgcid=rss_sd_all&

7- Securing data and preserving privacy in cloud IoT-based technologies an analysis of assessing threats and developing effective safeguard, Mayank Pathak,Kamta Nath Mishra,Satya Prakash Singh,2024 , https://link.springer.com/article/10.1007/s10462-024-10908-x

8- Hybrid energy-efficient algorithm for efficient Internet of Things deployment, Abdul Razaque,Yaser Jararweh,Bandar Alotaibi,Munif Alotaibi,Muder Almiani,2022, https://www.sciencedirect.com/science/article/abs/pii/S221053792200052X?via%3Dihub

9- Cloud Infrastructure Security, Mohammad GhasemiGol,2019, https://onlinelibrary.wiley.com/doi/10.1002/9781119053385.ch2

10- How Does Cloud Infrastructure Work? Karu Lal -Integration Engineer, Ohio National Financial Services, USA,2015, https://i-proclaim.my/journals/index.php/apjee/article/view/697

11- Provisioning Infrastructure Supporting Cloud Operations, Kenichi Sato†, Hideki Hayashi, and Ken Ojiri,2011, https://ntt-review.jp/archive/ntttechnical.php?contents=ntr201112fa2.html

12- Cloud Infrastructure Service Management - A Review, A. Anasuya Threse Innocent,2012, https://arxiv.org/abs/1206.6016

13- Evaluating Benefits and Challenges of Cloud Computing Adoption in It Industry, Dr.T. Vara Lakshmi,2024, https://goldncloudpublications.com/index.php/irjaem/article/view/261

14- UPGRADING CYBERSECURITY: INVESTIGATING THE SECURITY BENEFITS OF CLOUD COMPUTING, Rakhi Shriwas, Rahul Kumar Sen, Vishnu Agrawal, Jitendra Singh Chouhan, Saurbh Tege,2024, https://ijtrs.com/uploaded_paper/UPGRADING%20CYBERSECURITY%20INVESTIGATING%20THE%20SECURITY%20BENEFITS%20OF%20CLOUD%20COMPUTING.pdf,

15- Hybrid IT and Multi Cloud an Emerging Trend and Improved Performance in Cloud Computing, Srinivasa Rao Gundu, Charan Arur Panem & Anuradha Thimmapuram,2020, https://link.springer.com/article/10.1007/s42979-020-00277-x

16- Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey, Amrita Jyoti, Manish Shrimali, Shailesh Tiwari & Harivans Pratap Singh ,2020, https://link.springer.com/article/10.1007/s12652-020-01747-z

17- **Real-Time Cloud-Based Load Balance Algorithms and an Analysis, Srinivasa Rao Gundu, Charan Arur Panem & Anuradha Thimmapuram,2020,** https://link.springer.com/article/10.1007/s42979-020-00199-8

18- **MrLBA: multi-resource load balancing algorithm for cloud computing using ant colony optimization, Arfa Muteeh, Muhammad Sardaraz & Muhammad Tahir,2021,** https://link.springer.com/article/10.1007/s10586-021-03322-3

19- **Energy and carbon-aware initial VM placement in geographically distributed cloud data centers,2023, Ehsan Khodayarseresht ,Alireza Shameli-Sendi ,Quentin Fournier,Michel Dagenais ,** https://www.sciencedirect.com/science/article/abs/pii/S2210537923000434?&dgcid=rss_sd_all

20- **Integrating big data and cloud computing into the existing system and performance impact: A case study in manufacturing, Jeetendra Kumar Saraswat,Sanjay Choudhari, 2024,** https://www.sciencedirect.com/science/article/abs/pii/S0040162524006814?&dgcid=rss_sd_all

21- **An effective service-oriented networking management architecture for 5G-enabled internet of things, Mingfeng Huang , Anfeng Liu ,Neal N. Xiong ,Tian Wang ,Athanasios V. Vasilakos ,2020,** https://www.sciencedirect.com/science/article/abs/pii/S1389128619311442?via%3Dihub

22- **Cloud of Things: architecture, applications and challenges, Fahd Alhaidari, Atta Rahman & Rachid Zagrouba,2020,** https://link.springer.com/article/10.1007/s12652-020-02448-3

23- IoT systems modeling and performance evaluation, Alem Čolaković,2023, https://www.sciencedirect.com/science/article/abs/pii/S1574013723000655?via%3Dihub

24- Cloud-Based IoT Applications and Their Roles in Smart Cities, Tanweer Alam,2021, https://www.mdpi.com/2624-6511/4/3/64

25- A survey on integrated computing, caching, and communication in the cloud-to-edge continuum, Adyson Maia,2024, https://www.sciencedirect.com/science/article/pii/S0140366424000847?dgcid=rss_sd_all&

26- Empirical Research on the Relationship Between Ambidextrous Innovation Value of Cloud Computing and Environmental Characteristics, Zhenghua Li, Nianxin Wang, Zhiying Wang & Shilun Ge,2025, https://link.springer.com/article/10.1007/s13132-024-01973-2?

27- Business Impacts of Cloud Computing, Cameron Deed (Yellowfin, Australia) and Paul Cragg (University of Canterbury, New Zealand),2013 ,https://www.igi-global.com/gateway/chapter/70146

28- Electronic finance – recent developments, Krishnan Dandapani,2017, https://www.emerald.com/insight/content/doi/10.1108/mf-02-2017-0028/full/html

29- Cloud computing and security issues in the cloud, Santosh Kumar Singh,2018, https://journals.acspublisher.com/index.php/tjmitm/article/view/443

30- Addressing Security Concerns for Infrastructure of Cloud Computing, Shweta Gaur Sharma & Lakshmi Ahuja,2017, https://link.springer.com/chapter/10.1007/978-981-10-5699-4_73

31- The Role of Cloud Computing in Modern Business, Chirag Sharma,Hitain Kakkar, Dr. Ashima Mehta,2023, https://ijarsct.co.in/Paper9364.pdf

32- Comparison of Different Cloud Providers, Nishant Bhatte,Vivek Prajapati,Shreekunj Varia, Jyotsna More4,2022, https://www.ijraset.com/best-journal/comaprision-of-different-cloud-providers

33- Decoding the Cloud Giants: A Comparison of AWS, Azure and GCP ,Niket Bharat Patil , Rutik Nilesh Sankapal,2024, https://ijarsct.co.in/Paper18904.pdf

34- Interpretation on the Google Cloud Platform and Its Wide Cloud Services, Rafat Ul Aman Sajid (Lovely Professional University, India), Sirajul Islam (Lovely

Professional University, India), Abul Bashar Khan Rakib (Lovely Professional University, India), and Amandeep Kaur (Lovely Professional University, India),2022, https://www.igi-global.com/gateway/article/313586

35- Amazon and Microsoft Stay Ahead in Global Cloud Market, Felix Richter,2025, https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

36- An In-Depth Analysis of Amazon Web Services (AWS) and Its Position in the Cloud Computing Landscape, Harika Sanugommula,2024, https://ijsrem.com/download/an-in-depth-analysis-of-amazon-web-services-aws-and-its-position-in-the-cloud-computing-landscape/

37- Usability Evaluation of Cloud for HPC Applications, Vanessa Sochat, Daniel Milroy, Abhik Sarkar, Aniruddha Marathe,2025, https://arxiv.org/abs/2506.02709v1

38- Amazon Web Service–Google Cross-Cloud Platform for Machine Learning-Based Satellite Image Detection, David Pacios and others,2025, https://www.mdpi.com/2078-2489/16/5/381

39- Performance Variability in Public Clouds: An Empirical Assessment, Sanjay Ahuja and others,2025, https://www.mdpi.com/2078-2489/16/5/402

40- A Comparative Analysis of Security Services in Major Cloud Service Providers, Ashwin Guptha and others,2021, https://ieeexplore.ieee.org/document/9432189

41- Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure, Annanda Rath and others , 2019, https://www.mdpi.com/2073-431X/8/2/34

42- Cloud Computing Services and Microsoft Azure. Why Microsoft Azure?, Mehedi Hassan and others, 2022, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4103377

43- Hybrid Cloud Security: Balancing Performance, Cost, and Compliance in Multi-Cloud Deployments, Anjani kumar Polinati,2025, https://arxiv.org/abs/2506.00426v1

44- Cloud-Based Medical Named Entity Recognition: A FIT4NER-Based Approach, Philippe Tamla and others,2025, https://www.mdpi.com/2078-2489/16/5/395

45- A Survey of Google Cloud Platform (GCP): Features, Services, and Applications, Praveen Borra,2024, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4914149

46- Tiered Cloud Routing: Methodology, Latency, and Improvement, Shihan Lin and others ,2025, https://dl.acm.org/doi/10.1145/3711705

47- Novel framework for secured bulk creation of virtual machine in IaaS platform, Karishma Varshney and others, 2022, https://link.springer.com/article/10.1007/s11219-021-09573-y

48- Cost Optimization for Cloud Storage from User Perspectives: Recent Advances, Taxonomy, and Survey, Mingyu Liu and others ,2023, https://dl.acm.org/doi/10.1145/3582883

49- Comparative Review: Top Cloud Service Providers ETL Tools -AWS vs. Azure vs. GCP, Praveen Borra,2024, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4914175

50- **Cloud-Based Interoperability in Residential Energy Systems, Darren Leniston and others, 2025, https://arxiv.org/abs/2506.05076v1**

51- **Practical Comparison Between the CI/CD Platforms Azure DevOps and GitHub, Vladislav Manolov and others, 2025, https://www.mdpi.com/1999-5903/17/4/153**

52- **COMPARATIVE ANALYSIS OF CLOUD SERVICES FOR GEOINFORMATION DATA PROCESSING , Oleksandt Nedosnovanyi and others, 2023, https://itce.vntu.edu.ua/index.php/itce/article/view/953**

53- **Enabling the execution of HPC applications on public clouds with HPC@Cloud toolkit, Vanderlei Munhoz and others,2023, https://onlinelibrary.wiley.com/doi/10.1002/cpe.7976?af=R&sid=researcher&sid=researcher**

54- **Serverless Computing: An Investigation of Deployment Environments for Web APIs, Cosmina Ivan and others ,2019, https://www.mdpi.com/2073-431X/8/2/50**

55- **Containerized Microservices Orchestration and Provisioning in Cloud Computing: A Conceptual Framework and Future Perspectives, Abdul Saboor and others,2022, https://www.mdpi.com/2076-3417/12/12/5793?**

56- **Joint optimization of load balancing and resource allocation in cloud environment using optimal container management strategy, Saravanan Muniswamy and other,2024, https://onlinelibrary.wiley.com/doi/10.1002/cpe.8035?af=R&sid=researcher&&sid=researcher**

57- **Economic analysis of proposed regulations of cloud services in Europe, Joshua Gans and others,2023, https://www.tandfonline.com/doi/full/10.1080/17441056.2023.2228668?af=R&**

58- **FORMATION OF COMPETITIVE ADVANTAGES OF SERVICE COMPANIES IN THE CONDITIONS OF DIGITALIZATION, Olga Prygara and others, 2021, http://www.intellect21.nuft.org.ua/journal/2021/2021_6/8.pdf**

59- **COMPETITIVE ADVANTAGES IN THE MARKET OF CONSULTING SERVICES: MARKETING ASPECT, Natalia Yevtushenko and other, 2023, https://journals.vilniustech.lt/index.php/BTP/article/view/15291**

60- **COMPETITIVE ADVANTAGES OF BUSINESS ENTITIES: FORMATION, MODELLING, DETERMINANTS, Vasil Perebyynis and others,2024, http://market-infr.od.ua/journals/2024/78_2024/7.pdf**

61- **COMPETITIVE ADVANTAGES OF INSURANCE COMPANIES IN THE NEW ECONOMY, Sofiia Kucherivska,2024, https://journals.oa.edu.ua/Economy/article/view/3998**

62- **FEATURES OF THE FORMATION OF COMPETITIVE ADVANTAGES OF MOTOR TRANSPORT ENTERPRISES, Grechan and other,2023, http://publications.ntu.edu.ua/avtodorogi_i_stroitelstvo/114.2/245.pdf**

63- **Formation of Competitive Advantages and Competitive Potential of Life Insurance Companies, Maryna M. Novikova and others,2024, https://www.problecon.com/export_pdf/problems-of-economy-2024-1_0-pages-77_83.pdf**

64- **Incentive Mechanism for Cloud Service Offloading in Edge–Cloud Computing Environment, Chendie Yao and others,2025, https://www.mdpi.com/2227-7390/13/10/1685**

65- **A survey of resource provisioning problem in cloud brokers, Xingjia Li and others,2022, https://www.sciencedirect.com/science/article/abs/pii/S1084804522000479?via%3Dihub**

66- How big data analytics can create competitive advantage in high-stake decision forecasting? The mediating role of organizational innovation, Diana Korayim and others,2023, https://www.sciencedirect.com/science/article/abs/pii/S0040162523007254?&dgcid=rss_sd_all

67- Cosmos: A Cost Model for Serverless Workflows in the 3D Compute Continuum, Cynthia Marcelino and others,2025, https://arxiv.org/abs/2504.20189v1

68- Combined cloud and electricity portfolio optimization for cloud service providers, Caishan Guo and others,2025, https://www.sciencedirect.com/science/article/abs/pii/S0306261924013084?via%3Dihub

69- Tsay, R. S. (2010). "Analysis of Financial Time Series." John Wiley & Sons. (A foundational text for understanding time series models in finance, including GARCH, and their application to volatility analysis).

70- Hull, J. C. (2018). "Options, Futures, and Other Derivatives." Pearson. (Chapter on VaR and market risk provides details on how volatility is used in these calculations).

71- Moreira, A., & Muir, R. (2017). "Volatility-Managed Portfolios." Journal of Finance, 72(4), 1613-1643. (This paper demonstrates how volatility management can improve risk-adjusted returns across various investment strategies).

72- Bodie, Z., Kane, A., & Marcus, A. J. (2020). "Investments." McGraw-Hill Education. (Covers the basics of portfolio theory and the role of risk and return in portfolio construction).

73- Engle, R. (2002). "Dynamic Conditional Correlation: A Simple Class of Multivariate GARCH Models." Journal of Business & Economic Statistics, 20(3), 339-350

74- **Hull, J. C. (2018). "Options, Futures, and Other Derivatives." Pearson. (Provides detailed explanation of how volatility impacts option pricing).**

75- **Bodie, Z., Kane, A., & Marcus, A. J. (2020). "Investments." McGraw-Hill Education. (Discusses various risk-adjusted performance measures).**

76- **Cboe Volatility Index (VIX) or the Fear Index: Explanation and Calculation, By Peter Gratton,2025, https://www.investopedia.com/ask/answers/021015/what-cboe-volatility-index-vix.asp**

77- **Market Models: A Guide to Financial Data Analysis or many standard texts in time-series methods, Alexander, (2001).**

Appendix:

In [ ]:

```
#Importing libraries:
from google.colab import drive
import pandas  as pd
drive.mount('/content/drive')
!pip install pydantic-settings
from pydantic_settings import BaseSettings
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import seaborn as sb
from sklearn import linear_model
from datetime import datetime
from timeit import default_timer as timer
from IPython.display import display
pd.set_option("display.max_columns", 10000)
pd.set_option('display.max_rows', 10000)
pd.set_option('display.width', 10000)
pd.set_option('display.max_colwidth', 1)
import random
! cp /content/drive/MyDrive/Capstone\ project-EMBA-Valar/config.py /content

! cp /content/drive/MyDrive/Capstone\ project-EMBA-Valar/config.py /content
! cp /content/drive/MyDrive/Capstone\ project-EMBA-Valar/data.py /content

! cp "/content/drive/MyDrive/Capstone project-EMBA-Valar/.env" "/content"

import config
from config import settings
import data
from data import SQLRepository
from data import AlphaVantageAPI
import requests
import pandas as pd
import numpy as np
from scipy.stats import norm
import sqlite3

!pip install arch
from arch import arch_model
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, cal
l drive.mount("/content/drive", force_remount=True).
Requirement already satisfied: pydantic-settings in /usr/local/lib/python3.1
2/dist-packages (2.10.1)
Requirement already satisfied: pydantic>=2.7.0 in /usr/local/lib/python3.12/
dist-packages (from pydantic-settings) (2.11.7)
Requirement already satisfied: python-dotenv>=0.21.0 in /usr/local/lib/pytho
n3.12/dist-packages (from pydantic-settings) (1.1.1)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/py
thon3.12/dist-packages (from pydantic-settings) (0.4.1)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/pyth
on3.12/dist-packages (from pydantic>=2.7.0->pydantic-settings) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/pytho
n3.12/dist-packages (from pydantic>=2.7.0->pydantic-settings) (2.33.2)
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/p
ython3.12/dist-packages (from pydantic>=2.7.0->pydantic-settings) (4.15.0)
Collecting arch
  Downloading arch-7.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.22.3 in /usr/local/lib/python3.12/di
st-packages (from arch) (2.0.2)
Requirement already satisfied: scipy>=1.8 in /usr/local/lib/python3.12/dist-
packages (from arch) (1.16.1)
Requirement already satisfied: pandas>=1.4 in /usr/local/lib/python3.12/dist
-packages (from arch) (2.2.2)
Requirement already satisfied: statsmodels>=0.12 in /usr/local/lib/python3.1
2/dist-packages (from arch) (0.14.5)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyth
on3.12/dist-packages (from pandas>=1.4->arch) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dis
t-packages (from pandas>=1.4->arch) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/d
ist-packages (from pandas>=1.4->arch) (2025.2)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dis
t-packages (from statsmodels>=0.12->arch) (1.0.1)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/
dist-packages (from statsmodels>=0.12->arch) (25.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-pa
ckages (from python-dateutil>=2.8.2->pandas>=1.4->arch) (1.17.0)
Downloading arch-7.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (978 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 978.3/978.3 kB 18.1 MB/s eta 0:0
0:00
Installing collected packages: arch
Successfully installed arch-7.2.0
```

Preparing to gathering data:

In [ ]:
```python
#Connecting to SQL to save datasets
connection = sqlite3.connect(database=settings.db_name,check_same_thread=Fal
repo = SQLRepository(connection=connection)
print("repo type:", type(repo))
print("repo.connection type:", type(repo.connection))
```

```
repo type: <class 'data.SQLRepository'>
repo.connection type: <class 'sqlite3.Connection'>
```

In [ ]:
```
#Create a URL to get  recent 2500 observations the stock data for Microsoft
ticker = "MSFT"

av = AlphaVantageAPI()

microsoft_records = av.get_daily(ticker=ticker)

# Insert `microsoft_records` database using `repo`
from data import SQLRepository
connection = sqlite3.connect(database=settings.db_name,check_same_thread=Fal
repo = SQLRepository(connection=connection)
response = repo.insert_table(table_name=ticker, records=microsoft_records, i

df_microsoft = repo.read_table(table_name=ticker,limit=2500)

print("df_microsoft type:", type(df_microsoft))
print("df_microsoft shape:", df_microsoft.shape)
df_microsoft.head()
```

```
df_microsoft type: <class 'pandas.core.frame.DataFrame'>
df_microsoft shape: (2500, 5)
```

Out[ ]:

date	open	high	low	close	volume
2025-09-09	501.430	502.250	497.70	498.41	14410542.0
2025-09-08	498.105	501.195	495.03	498.20	16771015.0
2025-09-05	509.070	511.970	492.37	495.00	31994846.0
2025-09-04	504.300	508.150	503.15	507.97	15509486.0
2025-09-03	503.790	507.790	502.32	505.35	15995154.0

In [ ]:
```
#Create a URL to get recent 2500 observations the stock data for Amazon  fro

ticker = "AMZN"

av = AlphaVantageAPI()

amazon_records = av.get_daily(ticker=ticker)

# Insert `amazon_records` database using `repo`
from data import SQLRepository
connection = sqlite3.connect(database=settings.db_name,check_same_thread=Fal
repo = SQLRepository(connection=connection)
response = repo.insert_table(table_name=ticker, records=amazon_records, if_e

df_amazon = repo.read_table(table_name=ticker,limit=2500)

print("df_amazon type:", type(df_amazon))
```

```
print("df_amazon shape:", df_amazon.shape)
df_amazon.head()
```

```
df_amazon type: <class 'pandas.core.frame.DataFrame'>
df_amazon shape: (2500, 5)
```

Out[ ]:

date	open	high	low	close	volume
2025-09-09	236.355	238.8500	235.08	238.24	27033778.0
2025-09-08	234.940	237.6000	233.75	235.84	33947104.0
2025-09-05	235.190	236.0000	231.93	232.33	36721802.0
2025-09-04	231.185	235.7700	230.78	235.68	59391779.0
2025-09-03	225.210	227.1699	224.36	225.99	26355706.0

In [ ]:

```
#Create function to calculate 2500 most observations returns.
def wrangle_data(ticker,n_observations):

    """Extract table data from database. Calculate returns.

    Parameters
    ----------
    ticker : str
        The ticker symbol of the stock (also table name in database).

    n_observations : int
        Number of observations to return.

    Returns
    -------
    pd.Series
        Name will be `"return"`. There will be no `NaN` values.
    """
    # Get table from database
    df=repo.read_table(table_name=ticker,limit=n_observations+1)

    # Sort DataFrame ascending by date
    df.sort_index(ascending=True,inplace=True)


    # Create "return" column
    df["return"]=df["close"].pct_change()*100

    # Return returns
    return df["return"].dropna()
```

In [ ]:

```
#Create Amazon dataset that contains most returns of 2500 observations.
ticker = "AMZN"

y_amazon=wrangle_data(ticker,n_observations=2500)
print("y_amazon type:", type(y_amazon))
```

```
print("y_amazon shape:", y_amazon.shape)
y_amazon.head()
```

```
y_amazon type: <class 'pandas.core.series.Series'>
y_amazon shape: (2500,)
```

Out[ ]:

date	return
2015-09-30	3.144717
2015-10-01	1.768718
2015-10-02	2.269934
2015-10-05	2.091862
2015-10-06	-1.140377

dtype: float64

In [ ]:
```
#Create Microsoft dataset that contains most recent returns of 2500 observat

ticker = "MSFT"

y_microsoft=wrangle_data(ticker,n_observations=2500)
print("y_microsoft type:", type(y_microsoft))
print("y_microsoft shape:", y_microsoft.shape)
y_microsoft.head()
```

```
y_microsoft type: <class 'pandas.core.series.Series'>
y_microsoft shape: (2500,)
```

Out[ ]:

date	return
2015-09-30	1.749540
2015-10-01	0.927602
2015-10-02	2.151984
2015-10-05	2.326092
2015-10-06	0.257345

dtype: float64

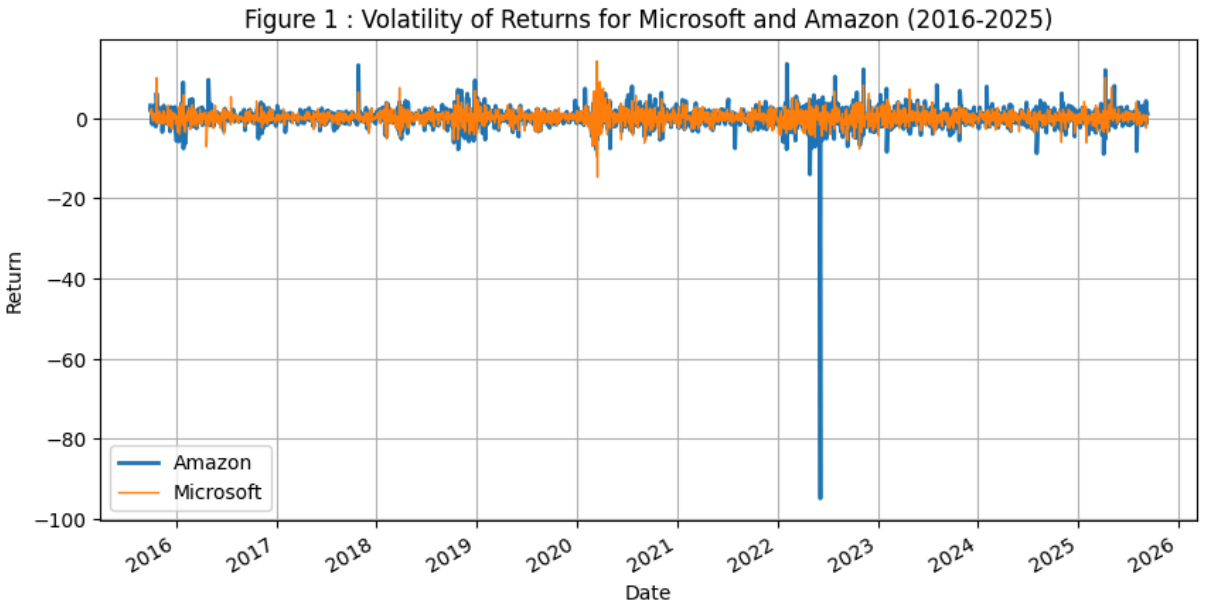
In [ ]:
```

```

Volatility

Exploring

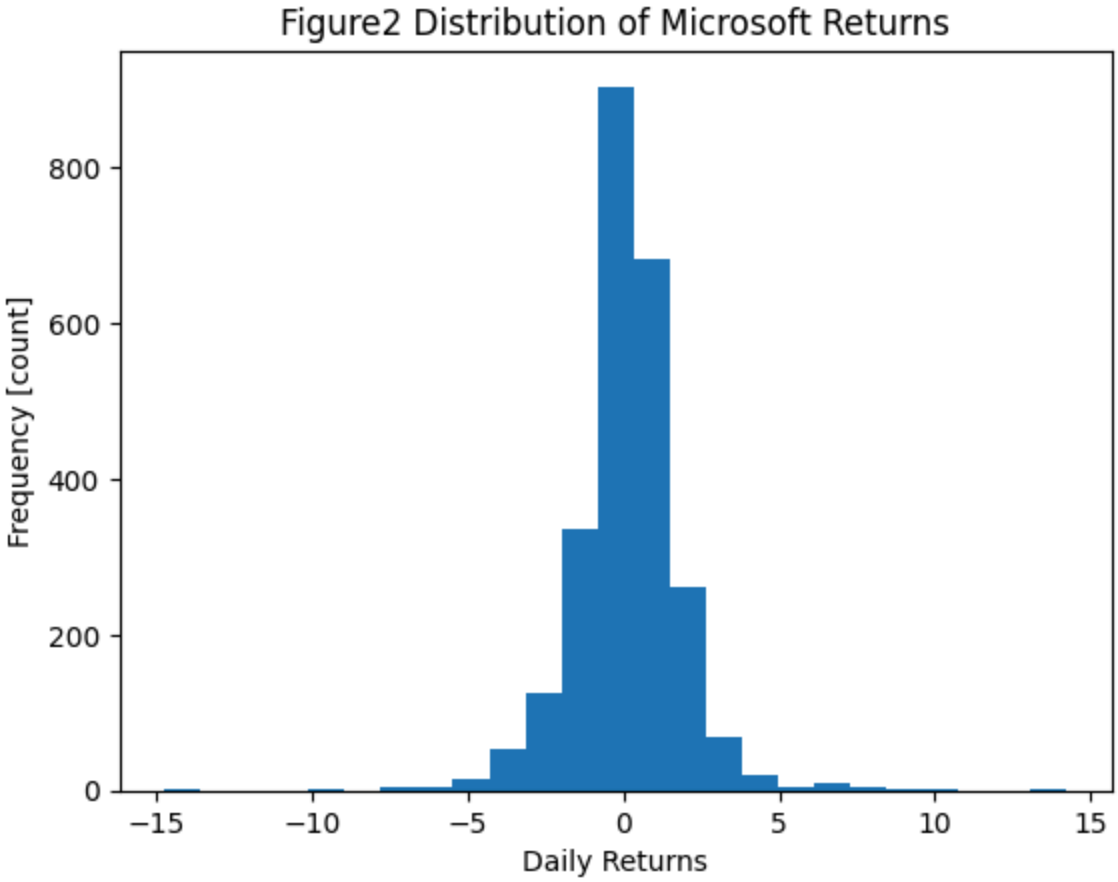
Let's create the volatility time series plot so that we have a visual aid to talk about what volatility is.

In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 5))

# Plot returns for `y_microsoft` and `y_amazon`
y_amazon.plot(ax=ax, label="Amazon",linewidth=2)
y_microsoft.plot(ax=ax, label="Microsoft",linewidth=1)

# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
# Add Title
plt.title("Figure 1 : Volatility of Returns for Microsoft and Amazon (2016-2
# Add grid
plt.grid(True)
# Add legend
plt.legend();
```



Let's create a histogram about Distribution of Microsoft Daily Returns .

In [ ]:
```
plt.hist(y_microsoft,bins=25)
# Add axis labels
plt.xlabel("Daily Returns")
plt.ylabel("Frequency [count]")


# Add title
plt.title("Figure2 Distribution of Microsoft Returns");
```



Create a histogram about Distribution of Amazon Daily Returns .

In [ ]:
```
plt.hist(y_amazon,bins=25)
# Add axis labels
plt.xlabel("Daily Returns")
plt.ylabel("Frequency [count]")


# Add title
plt.title("Figure 3 Distribution of Amazon  Returns");
```

Let's start by measuring the daily volatility of our two stocks. Since our data frequency is also daily, this will be exactly the same as calculating the standard deviation.

In [ ]:
```
microsoft_daily_volatility = y_microsoft.std()
amazon_daily_volatility = y_amazon.std()

print("Microsoft Daily Volatility:", microsoft_daily_volatility)
print("Amazon Daily Volatility:   ", amazon_daily_volatility)
```

```
Microsoft Daily Volatility: 1.7018376833451128
Amazon Daily Volatility:    2.8058045528789792
```

In [ ]:
```
microsoft_annual_volatility = microsoft_daily_volatility*np.sqrt(252)
amazon_annual_volatility = amazon_daily_volatility*np.sqrt(252)

print("Microsoft Annual Volatility:", microsoft_annual_volatility)
print("Amazon Annual Volatility:   ", amazon_annual_volatility)
```

```
Microsoft Annual Volatility: 27.015835691576747
Amazon Annual Volatility:    44.540766446223344
```

In [ ]:
```
amazon_rolling_50d_volatility = y_amazon.rolling(window=50).std().dropna()
```

```
print("rolling_50d_volatility type:", type(amazon_rolling_50d_volatility))
print("rolling_50d_volatility shape:", amazon_rolling_50d_volatility.shape)
amazon_rolling_50d_volatility.head()
```

```
rolling_50d_volatility type: <class 'pandas.core.series.Series'>
rolling_50d_volatility shape: (2451,)
```

Out[ ]:

	return
date	
2015-12-09	1.701234
2015-12-10	1.666265
2015-12-11	1.743974
2015-12-14	1.756259
2015-12-15	1.740478

dtype: float64

In [ ]:
```
microsoft_rolling_50d_volatility = y_microsoft.rolling(window=50).std().drop

print("rolling_50d_volatility type:", type(microsoft_rolling_50d_volatility)
print("rolling_50d_volatility shape:", microsoft_rolling_50d_volatility.shap
microsoft_rolling_50d_volatility.head()
```

```
rolling_50d_volatility type: <class 'pandas.core.series.Series'>
rolling_50d_volatility shape: (2451,)
```

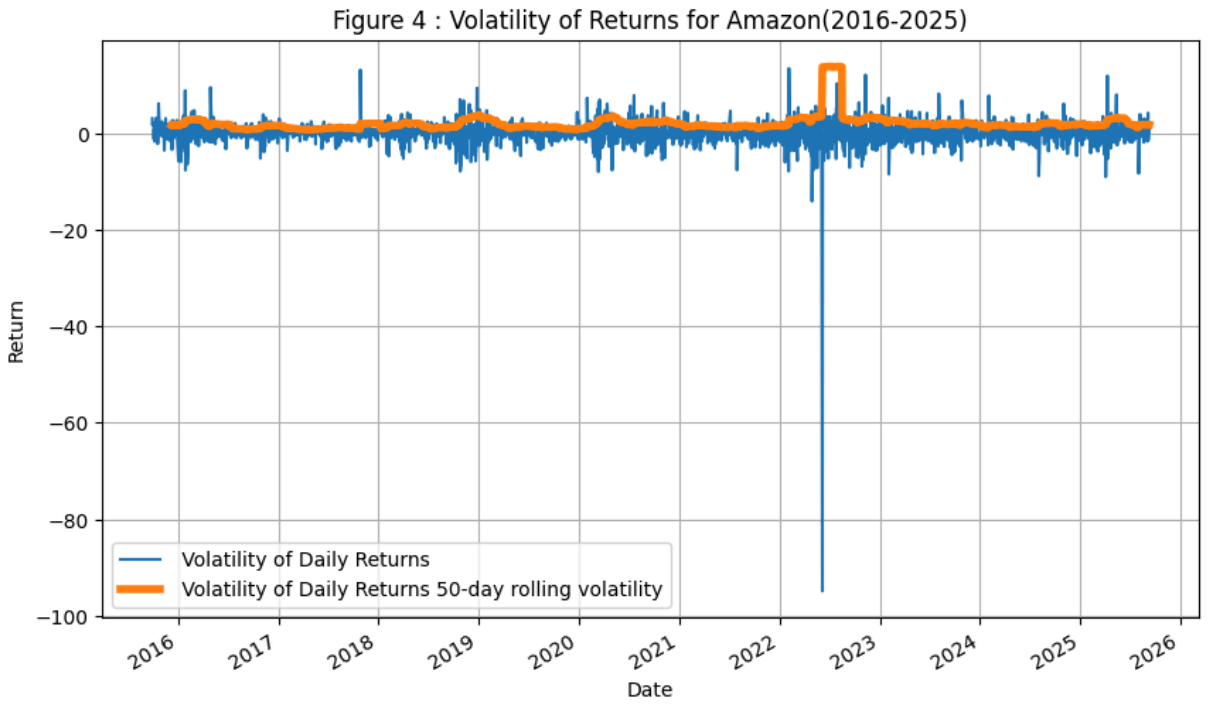
Out[ ]:

	return
date	
2015-12-09	1.819528
2015-12-10	1.810420
2015-12-11	1.847395
2015-12-14	1.844539
2015-12-15	1.823776

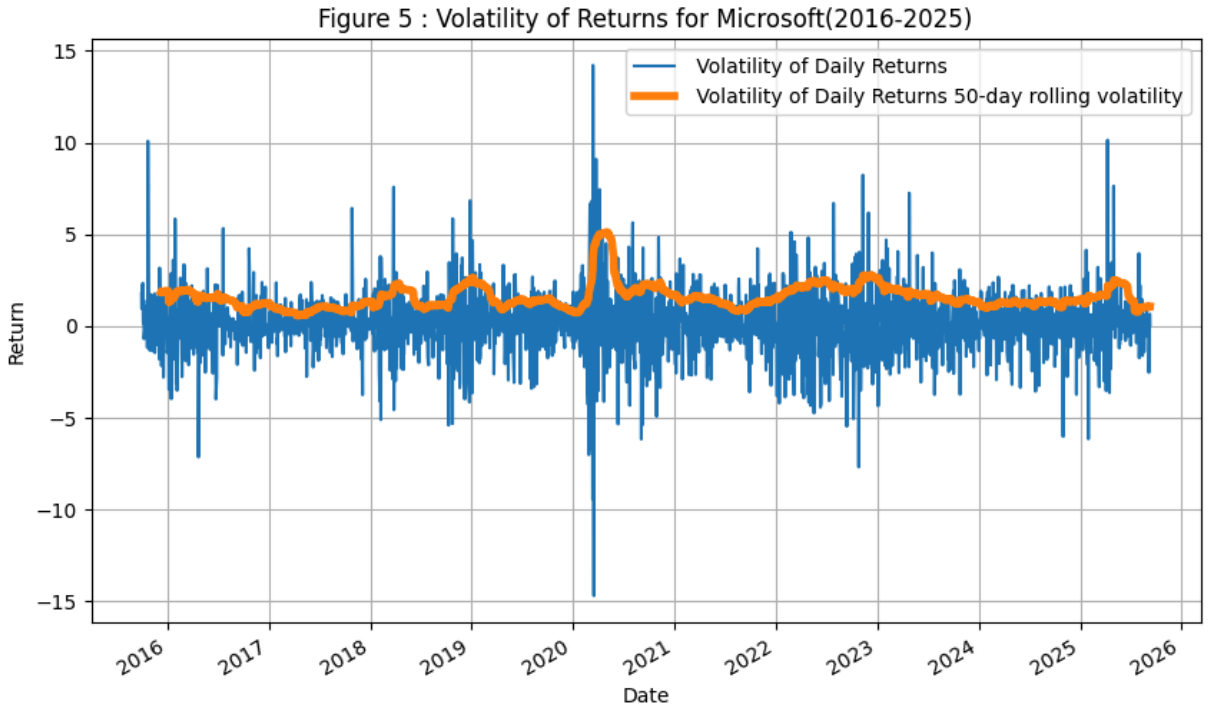
dtype: float64

In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 6))

# Plot daily returns for `y_amazon`
y_amazon.plot(ax=ax, label=" Volatility of Daily Returns")
amazon_rolling_50d_volatility.plot(ax=ax, label=" Volatility of Daily Returr

# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
# Add Title
plt.title("Figure 4 : Volatility of Returns for Amazon(2016-2025)")
```

```
# Add grid
plt.grid(True)
# Add legend
plt.legend();
```

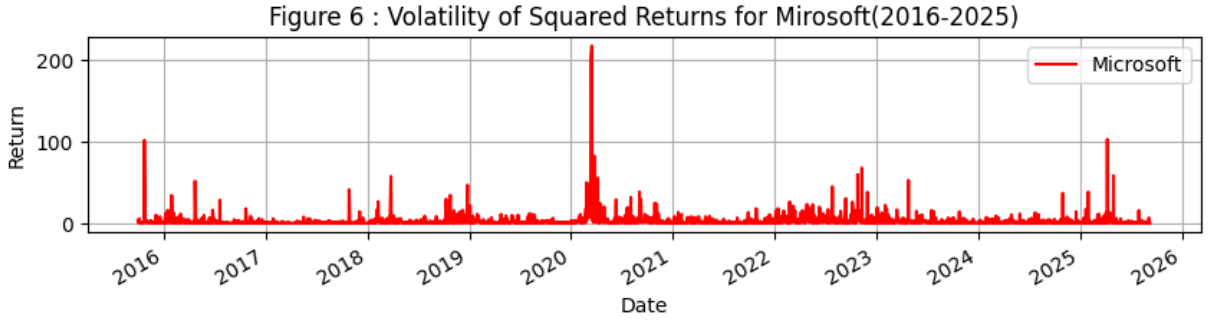


In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 6))

# Plot daily returns for `y_amazon`
y_microsoft.plot(ax=ax, label=" Volatility of Daily Returns")
microsoft_rolling_50d_volatility.plot(ax=ax, label=" Volatility of Daily Ret

# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
# Add Title
plt.title("Figure 5 : Volatility of Returns for Microsoft(2016-2025)")
# Add grid
plt.grid(True)
# Add legend
plt.legend();
```



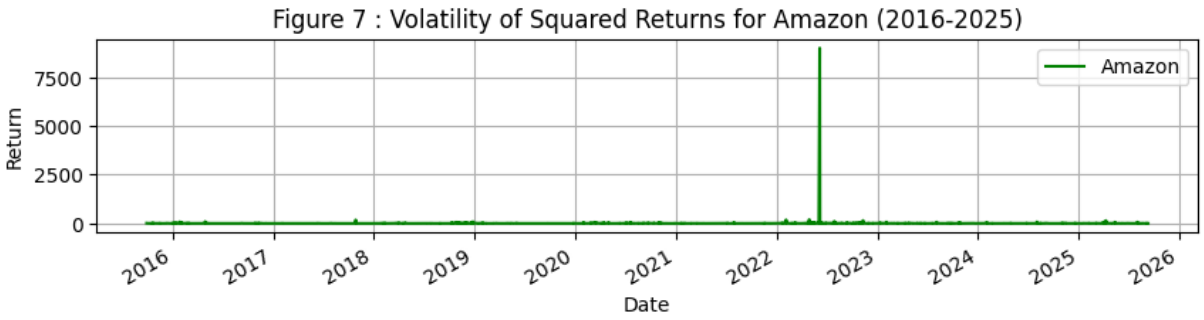
Let's create a time series plot of the squared returns in Amazon and Microsoft returns

In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 2))

(y_microsoft**2).plot(ax=ax, label="Microsoft",linewidth=1.5,color='red')

# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
# Add Title
plt.title("Figure 6 : Volatility of Squared Returns for Mirosoft(2016-2025)"
# Add grid
plt.grid(True)
# Add legend
plt.legend();
```



In [ ]:
```
fig, ax = plt.subplots(figsize=(10, 2))


(y_amazon**2).plot(ax=ax, label="Amazon",linewidth=1.5,color="green")
```

```
# Label axes
plt.xlabel("Date")
plt.ylabel("Return")
# Add Title
plt.title("Figure 7 : Volatility of Squared Returns for Amazon (2016-2025)")
# Add grid
plt.grid(True)
# Add legend
plt.legend();
```



In [ ]:
```
df_amazon.to_csv("/content/drive/MyDrive/Capstone project-EMBA-Valar/df_amaz
df_microsoft.to_csv("/content/drive/MyDrive/Capstone project-EMBA-Valar/df_m
y_amazon.to_csv("/content/drive/MyDrive/Capstone project-EMBA-Valar/y_amazor
y_microsoft.to_csv("/content/drive/MyDrive/Capstone project-EMBA-Valar/y_mic
```

MICROSOFT STOCK VOLATILITY ANALYSIS REPORT

In [ ]:
```
import pandas as pd
import numpy as np
from arch import arch_model
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from scipy import stats
from statsmodels.stats.diagnostic import acorr_ljungbox, het_arch
from tabulate import tabulate

import pandas as pd
import numpy as np
from arch import arch_model
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from scipy import stats
from statsmodels.stats.diagnostic import acorr_ljungbox, het_arch
from tabulate import tabulate

# Updated style configuration with fallback
available_styles = plt.style.available
if 'seaborn-whitegrid' in available_styles:
    plt.style.use('seaborn-whitegrid')
```

```
elif 'seaborn' in available_styles:
    plt.style.use('seaborn')
elif 'ggplot' in available_styles:
    plt.style.use('ggplot')
else:
    print("Using default matplotlib style")

plt.rcParams.update({
    'font.size': 12,
    'figure.figsize': (14, 10),
    'axes.grid': True,
    'grid.linestyle': '--',
    'grid.alpha': 0.6
})

def load_and_prepare_data(file_path):
    """Load and prepare time series data"""
    data = pd.read_csv(file_path, parse_dates=['date'])
    data = data[['date', 'return']].copy()
    data['return'] = pd.to_numeric(data['return'], errors='coerce')
    data = data.dropna(subset=['return']).reset_index(drop=True)
    data.set_index('date', inplace=True)
    return data

def fit_garch_model(returns):
    """Fit GARCH(1,1) model with Student's t-distribution"""
    model = arch_model(returns, vol='Garch', p=1, q=1, dist='StudentsT', res
    return model.fit(disp='off')

def run_diagnostics(residuals, lags=10):
    """Run model diagnostic tests"""
    lb_test = acorr_ljungbox(residuals, lags=lags, return_df=True)
    arch_test = het_arch(residuals)
    return {
        'Ljung-Box': lb_test['lb_pvalue'].values,
        'ARCH-LM': arch_test[1]
    }

def calculate_var(mean_return, volatility, nu, confidence=0.05):
    """Calculate Value at Risk using t-distribution"""
    return mean_return + volatility * stats.t.ppf(confidence, df=nu)

def generate_volatility_forecast(model, horizon=10):
    """Generate volatility forecasts"""
    forecast = model.forecast(horizon=horizon)
    variance = forecast.variance.iloc[-1].values
    return np.sqrt(variance)  # Convert to volatility

def plot_results(returns, train_size, forecasted_vol):
    """Visualize results with comprehensive plots"""
    fig = plt.figure(figsize=(15, 15))
    gs = gridspec.GridSpec(3, 2, figure=fig)

    # Historical returns
    ax1 = fig.add_subplot(gs[0, :])
    returns.plot(ax=ax1, title='Historical Daily Returns', color='royalblue'
```

```
    ax1.axvline(returns.index[train_size], color='r', linestyle='--',
                label='Train/Test Split (80%/20%)')
    ax1.set_ylabel('Return (%)')
    ax1.legend()

    # Recent returns (last 3 months)
    ax2 = fig.add_subplot(gs[1, 0])
    returns.last('90D').plot(ax=ax2, title='Recent Returns (90 Days)', color
    ax2.set_ylabel('Return (%)')

    # Volatility forecast
    ax3 = fig.add_subplot(gs[1, 1])
    horizons = [f'Day {i+1}' for i in range(len(forecasted_vol))]
    ax3.plot(horizons, forecasted_vol, 'o-', color='darkred')
    ax3.set_title('10-Day Volatility Forecast')
    ax3.set_ylabel('Volatility (%)')
    ax3.set_ylim(0, max(forecasted_vol)*1.2)

    # Return distribution
    ax4 = fig.add_subplot(gs[2, 0])
    returns.plot(kind='hist', bins=50, ax=ax4, density=True,
                 color='teal', alpha=0.7, title='Return Distribution')
    returns.plot(kind='kde', ax=ax4, color='darkblue')
    ax4.axvline(returns.mean(), color='red', linestyle='--', label='Mean')

    # QQ-Plot
    ax5 = fig.add_subplot(gs[2, 1])
    stats.probplot(returns, dist="norm", plot=ax5)
    ax5.set_title('Normal Q-Q Plot')

    plt.tight_layout()
    plt.savefig('volatility_analysis.png', dpi=300)
    plt.show()

def generate_investment_implications(params, forecast, var):
    """Generate strategic investment implications"""
    implications = []
    persistence = params['alpha'] + params['beta']

    # Persistence implications
    if persistence > 0.9:
        implications.append({
            'Factor': 'High Volatility Persistence (α+β>0.9)',
            'Implication': 'Long-lasting volatility shocks',
            'Action': 'Reduce exposure, use protective puts'
        })
    else:
        implications.append({
            'Factor': 'Moderate Volatility Persistence',
            'Implication': 'Faster volatility reversion',
            'Action': 'Suitable for tactical trading'
        })

    # ARCH term implications
    if params['alpha'] > 0.3:
        implications.append({
```

```
            'Factor': 'High ARCH Term (α>0.3)',
            'Implication': 'Recent shocks drive volatility',
            'Action': 'Implement stop-loss orders'
        })

    # GARCH term implications
    if params['beta'] > 0.7:
        implications.append({
            'Factor': 'High GARCH Term (β>0.7)',
            'Implication': 'Stable volatility regimes',
            'Action': 'Long-term risk budgeting'
        })

    # Tail risk implications
    if params['nu'] < 5:
        implications.append({
            'Factor': 'Fat Tails (ν<5)',
            'Implication': 'Higher extreme return risk',
            'Action': 'Increase hedging, monitor tail risks'
        })

    # Volatility forecast implications
    hist_vol = returns.abs().mean()
    if forecast[0] > hist_vol:
        implications.append({
            'Factor': f'High Forecasted Volatility ({forecast[0]:.2f}% > {hi
            'Implication': 'Increased near-term risk',
            'Action': 'Reduce position sizes, consider hedging'
        })

    # VaR implications
    implications.append({
        'Factor': f'1-Day VaR ({var:.2f}%)',
        'Implication': 'Potential loss threshold',
        'Action': 'Set stop-loss levels, allocate capital accordingly'
    })

    return pd.DataFrame(implications)

# Main analysis
if __name__ == "__main__":
    # Load and prepare data
    data = load_and_prepare_data("/content/drive/MyDrive/Capstone project-EM
    returns = data['return']

    # Split data (80% train, 20% test)
    train_size = int(0.8 * len(returns))
    train_data = returns[:train_size]
    test_data = returns[train_size:]

    # Fit GARCH model
    results = fit_garch_model(train_data)
    params = {
        'omega': results.params['omega'],
        'alpha': results.params['alpha[1]'],
        'beta': results.params['beta[1]'],
```

```
        'nu': results.params['nu'],
        'persistence': results.params['alpha[1]'] + results.params['beta[1]'
    }

    # Run diagnostics
    residuals = results.std_resid.dropna()
    diagnostics = run_diagnostics(residuals)

    # Generate forecasts
    forecasted_vol = generate_volatility_forecast(results)

    # Calculate VaR
    mean_return = train_data.mean()
    var_95 = calculate_var(mean_return, forecasted_vol[0]/100, params['nu'])

    # Create summary table
    summary_table = [
        ["Parameter", "Value", "Interpretation"],
        ["Omega (ω)", f"{params['omega']:.6f}", "Baseline volatility level"]
        ["Alpha (α)", f"{params['alpha']:.6f}", "Impact of recent shocks"],
        ["Beta (β)", f"{params['beta']:.6f}", "Long-term volatility memory"]
        ["Persistence (α+β)", f"{params['persistence']:.4f}",
         "Duration of volatility shocks (High >0.9)"],
        ["Degrees of Freedom (ν)", f"{params['nu']:.2f}",
         "Tail thickness (Lower = fatter tails)"],
        ["AIC", f"{results.aic:.2f}", "Model fit (Lower = better)"],
        ["BIC", f"{results.bic:.2f}", "Model fit (Lower = better)"],
        ["ARCH-LM p-value", f"{diagnostics['ARCH-LM']:.4f}",
         "Remaining ARCH effects (<0.05 significant)"],
        ["1-Day VaR (95%)", f"{var_95:.2f}%", "Potential daily loss threshol
        ["Day 1 Volatility Forecast", f"{forecasted_vol[0]:.2f}%",
         "Expected near-term volatility"]
    ]

    # Print results
    print("="*80)
    print("MICROSOFT STOCK VOLATILITY ANALYSIS REPORT")
    print("="*80)
    print(f"\n{' Model Parameters and Diagnostics ':-^80}")
    print(tabulate(summary_table, headers="firstrow", tablefmt="grid"))

    print(f"\n{' Strategic Investment Implications ':-^80}")
    implications_df = generate_investment_implications(params, forecasted_vo
    print(tabulate(implications_df, headers='keys', tablefmt='grid', showinc

    print(f"\n{' Forecasted Volatility ':-^80}")
    forecast_df = pd.DataFrame({
        'Horizon': [f'Day {i+1}' for i in range(10)],
        'Volatility (%)': [f"{v:.2f}%" for v in forecasted_vol]
    })
    print(tabulate(forecast_df, headers='keys', tablefmt='grid', showindex=F

    # Generate plots
    plot_results(returns, train_size, forecasted_vol)
```

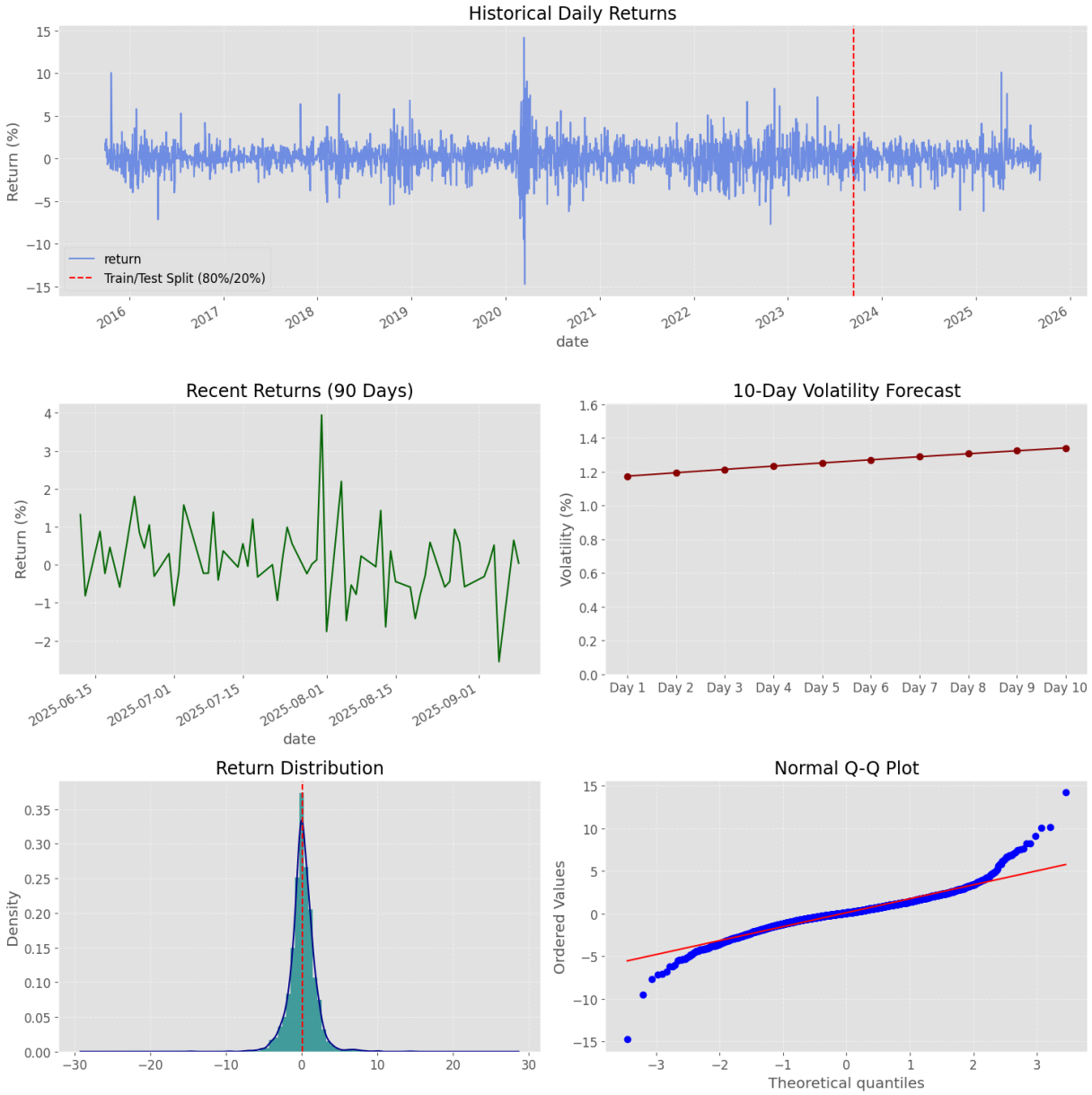
```
================================================================================
MICROSOFT STOCK VOLATILITY ANALYSIS REPORT
================================================================================

---------------------- Model Parameters and Diagnostics ----------------------
+---------------------------+----------+-----------------------------------+
| Parameter                 | Value    | Interpretation                    |
+===========================+==========+===================================+
| Omega (ω)                 | 0.057268 | Baseline volatility level         |
+---------------------------+----------+-----------------------------------+
| Alpha (α)                 | 0.130045 | Impact of recent shocks           |
+---------------------------+----------+-----------------------------------+
| Beta (β)                  | 0.863333 | Long-term volatility memory       |
+---------------------------+----------+-----------------------------------+
| Persistence (α+β)         | 0.9934   | Duration of volatility shocks (High
>0.9)  |
+---------------------------+----------+-----------------------------------+
| Degrees of Freedom (ν)    | 4.84     | Tail thickness (Lower = fatter tail
s)      |
+---------------------------+----------+-----------------------------------+
| AIC                       | 7249.51  | Model fit (Lower = better)        |
+---------------------------+----------+-----------------------------------+
| BIC                       | 7277.51  | Model fit (Lower = better)        |
+---------------------------+----------+-----------------------------------+
| ARCH-LM p-value           | 0.6148   | Remaining ARCH effects (<0.05 signi
ficant) |
+---------------------------+----------+-----------------------------------+
| 1-Day VaR (95%)           | 0.09%    | Potential daily loss threshold    |
+---------------------------+----------+-----------------------------------+
| Day 1 Volatility Forecast | 1.17%    | Expected near-term volatility     |
+---------------------------+----------+-----------------------------------+

---------------------- Strategic Investment Implications ------------------
```

Factor	Implication	Action
High Volatility Persistence (α+β>0.9)	Long-lasting volatility shocks	Reduce exposure, use protective puts
High GARCH Term (β>0.7)	Stable volatility regimes	Long-term risk budgeting
Fat Tails (ν<5)	Higher extreme return risk	Increase hedging, monitor tail risks
High Forecasted Volatility (1.17% > 1.17%)	Increased near-term risk	Reduce position sizes, consider hedging
1-Day VaR (0.09%)	Potential loss threshold	Set stop-loss levels, allocate capital accordingly

Forecasted Volatility

Horizon	Volatility (%)
Day 1	1.17%
Day 2	1.19%
Day 3	1.21%
Day 4	1.23%
Day 5	1.25%
Day 6	1.27%
Day 7	1.29%
Day 8	1.31%
Day 9	1.32%
Day 10	1.34%

```
/tmp/ipython-input-1186087632.py:86: FutureWarning: last is deprecated and w
ill be removed in a future version. Please create a mask and filter using `.
loc` instead
  returns.last('90D').plot(ax=ax2, title='Recent Returns (90 Days)', color
='darkgreen')
```



AMAZON STOCK VOLATILITY ANALYSIS REPORT

In [ ]:
```
import pandas as pd
import numpy as np
from arch import arch_model
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from scipy import stats
from statsmodels.stats.diagnostic import acorr_ljungbox, het_arch
```

```python
from tabulate import tabulate

import pandas as pd
import numpy as np
from arch import arch_model
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from scipy import stats
from statsmodels.stats.diagnostic import acorr_ljungbox, het_arch
from tabulate import tabulate

# Updated style configuration with fallback
available_styles = plt.style.available
if 'seaborn-whitegrid' in available_styles:
    plt.style.use('seaborn-whitegrid')
elif 'seaborn' in available_styles:
    plt.style.use('seaborn')
elif 'ggplot' in available_styles:
    plt.style.use('ggplot')
else:
    print("Using default matplotlib style")

plt.rcParams.update({
    'font.size': 12,
    'figure.figsize': (14, 10),
    'axes.grid': True,
    'grid.linestyle': '--',
    'grid.alpha': 0.6
})

def load_and_prepare_data(file_path):
    """Load and prepare time series data"""
    data = pd.read_csv(file_path, parse_dates=['date'])
    data = data[['date', 'return']].copy()
    data['return'] = pd.to_numeric(data['return'], errors='coerce')
    data = data.dropna(subset=['return']).reset_index(drop=True)
    data.set_index('date', inplace=True)
    return data

def fit_garch_model(returns):
    """Fit GARCH(1,1) model with Student's t-distribution"""
    model = arch_model(returns, vol='Garch', p=1, q=1, dist='StudentsT', res
    return model.fit(disp='off')

def run_diagnostics(residuals, lags=10):
    """Run model diagnostic tests"""
    lb_test = acorr_ljungbox(residuals, lags=lags, return_df=True)
    arch_test = het_arch(residuals)
    return {
        'Ljung-Box': lb_test['lb_pvalue'].values,
        'ARCH-LM': arch_test[1]
    }

def calculate_var(mean_return, volatility, nu, confidence=0.05):
    """Calculate Value at Risk using t-distribution"""
    return mean_return + volatility * stats.t.ppf(confidence, df=nu)
```

```python
def generate_volatility_forecast(model, horizon=10):
    """Generate volatility forecasts"""
    forecast = model.forecast(horizon=horizon)
    variance = forecast.variance.iloc[-1].values
    return np.sqrt(variance)  # Convert to volatility

def plot_results(returns, train_size, forecasted_vol):
    """Visualize results with comprehensive plots"""
    fig = plt.figure(figsize=(15, 15))
    gs = gridspec.GridSpec(3, 2, figure=fig)

    # Historical returns
    ax1 = fig.add_subplot(gs[0, :])
    returns.plot(ax=ax1, title='Historical Daily Returns', color='royalblue'
    ax1.axvline(returns.index[train_size], color='r', linestyle='--',
                label='Train/Test Split (80%/20%)')
    ax1.set_ylabel('Return (%)')
    ax1.legend()

    # Recent returns (last 3 months)
    ax2 = fig.add_subplot(gs[1, 0])
    returns.last('90D').plot(ax=ax2, title='Recent Returns (90 Days)', color
    ax2.set_ylabel('Return (%)')

    # Volatility forecast
    ax3 = fig.add_subplot(gs[1, 1])
    horizons = [f'Day {i+1}' for i in range(len(forecasted_vol))]
    ax3.plot(horizons, forecasted_vol, 'o-', color='darkred')
    ax3.set_title('10-Day Volatility Forecast')
    ax3.set_ylabel('Volatility (%)')
    ax3.set_ylim(0, max(forecasted_vol)*1.2)

    # Return distribution
    ax4 = fig.add_subplot(gs[2, 0])
    returns.plot(kind='hist', bins=50, ax=ax4, density=True,
                 color='teal', alpha=0.7, title='Return Distribution')
    returns.plot(kind='kde', ax=ax4, color='darkblue')
    ax4.axvline(returns.mean(), color='red', linestyle='--', label='Mean')

    # QQ-Plot
    ax5 = fig.add_subplot(gs[2, 1])
    stats.probplot(returns, dist="norm", plot=ax5)
    ax5.set_title('Normal Q-Q Plot')

    plt.tight_layout()
    plt.savefig('volatility_analysis.png', dpi=300)
    plt.show()

def generate_investment_implications(params, forecast, var):
    """Generate strategic investment implications"""
    implications = []
    persistence = params['alpha'] + params['beta']

    # Persistence implications
    if persistence > 0.9:
```

```
        implications.append({
            'Factor': 'High Volatility Persistence (α+β>0.9)',
            'Implication': 'Long-lasting volatility shocks',
            'Action': 'Reduce exposure, use protective puts'
        })
    else:
        implications.append({
            'Factor': 'Moderate Volatility Persistence',
            'Implication': 'Faster volatility reversion',
            'Action': 'Suitable for tactical trading'
        })

    # ARCH term implications
    if params['alpha'] > 0.3:
        implications.append({
            'Factor': 'High ARCH Term (α>0.3)',
            'Implication': 'Recent shocks drive volatility',
            'Action': 'Implement stop-loss orders'
        })

    # GARCH term implications
    if params['beta'] > 0.7:
        implications.append({
            'Factor': 'High GARCH Term (β>0.7)',
            'Implication': 'Stable volatility regimes',
            'Action': 'Long-term risk budgeting'
        })

    # Tail risk implications
    if params['nu'] < 5:
        implications.append({
            'Factor': 'Fat Tails (ν<5)',
            'Implication': 'Higher extreme return risk',
            'Action': 'Increase hedging, monitor tail risks'
        })

    # Volatility forecast implications
    hist_vol = returns.abs().mean()
    if forecast[0] > hist_vol:
        implications.append({
            'Factor': f'High Forecasted Volatility ({forecast[0]:.2f}% > {hi
            'Implication': 'Increased near-term risk',
            'Action': 'Reduce position sizes, consider hedging'
        })

    # VaR implications
    implications.append({
        'Factor': f'1-Day VaR ({var:.2f}%)',
        'Implication': 'Potential loss threshold',
        'Action': 'Set stop-loss levels, allocate capital accordingly'
    })

    return pd.DataFrame(implications)

# Main analysis
if __name__ == "__main__":
```

```
# Load and prepare data
data = load_and_prepare_data("/content/drive/MyDrive/Capstone project-EM
returns = data['return']

# Split data (80% train, 20% test)
train_size = int(0.8 * len(returns))
train_data = returns[:train_size]
test_data = returns[train_size:]

# Fit GARCH model
results = fit_garch_model(train_data)
params = {
    'omega': results.params['omega'],
    'alpha': results.params['alpha[1]'],
    'beta': results.params['beta[1]'],
    'nu': results.params['nu'],
    'persistence': results.params['alpha[1]'] + results.params['beta[1]'
}

# Run diagnostics
residuals = results.std_resid.dropna()
diagnostics = run_diagnostics(residuals)

# Generate forecasts
forecasted_vol = generate_volatility_forecast(results)

# Calculate VaR
mean_return = train_data.mean()
var_95 = calculate_var(mean_return, forecasted_vol[0]/100, params['nu'])

# Create summary table
summary_table = [
    ["Parameter", "Value", "Interpretation"],
    ["Omega (ω)", f"{params['omega']:.6f}", "Baseline volatility level"]
    ["Alpha (α)", f"{params['alpha']:.6f}", "Impact of recent shocks"],
    ["Beta (β)", f"{params['beta']:.6f}", "Long-term volatility memory"]
    ["Persistence (α+β)", f"{params['persistence']:.4f}",
     "Duration of volatility shocks (High >0.9)"],
    ["Degrees of Freedom (ν)", f"{params['nu']:.2f}",
     "Tail thickness (Lower = fatter tails)"],
    ["AIC", f"{results.aic:.2f}", "Model fit (Lower = better)"],
    ["BIC", f"{results.bic:.2f}", "Model fit (Lower = better)"],
    ["ARCH-LM p-value", f"{diagnostics['ARCH-LM']:.4f}",
     "Remaining ARCH effects (<0.05 significant)"],
    ["1-Day VaR (95%)", f"{var_95:.2f}%", "Potential daily loss threshol
    ["Day 1 Volatility Forecast", f"{forecasted_vol[0]:.2f}%",
     "Expected near-term volatility"]
]

# Print results
print("="*80)
print("AMAZON STOCK VOLATILITY ANALYSIS REPORT")
print("="*80)
print(f"\n{' Model Parameters and Diagnostics ':-^80}")
print(tabulate(summary_table, headers="firstrow", tablefmt="grid"))
```

```
    print(f"\n{' Strategic Investment Implications ':-^80}")
    implications_df = generate_investment_implications(params, forecasted_vo
    print(tabulate(implications_df, headers='keys', tablefmt='grid', showinc

    print(f"\n{' Forecasted Volatility ':-^80}")
    forecast_df = pd.DataFrame({
        'Horizon': [f'Day {i+1}' for i in range(10)],
        'Volatility (%)': [f"{v:.2f}%" for v in forecasted_vol]
    })
    print(tabulate(forecast_df, headers='keys', tablefmt='grid', showindex=F

    # Generate plots
    plot_results(returns, train_size, forecasted_vol)
```

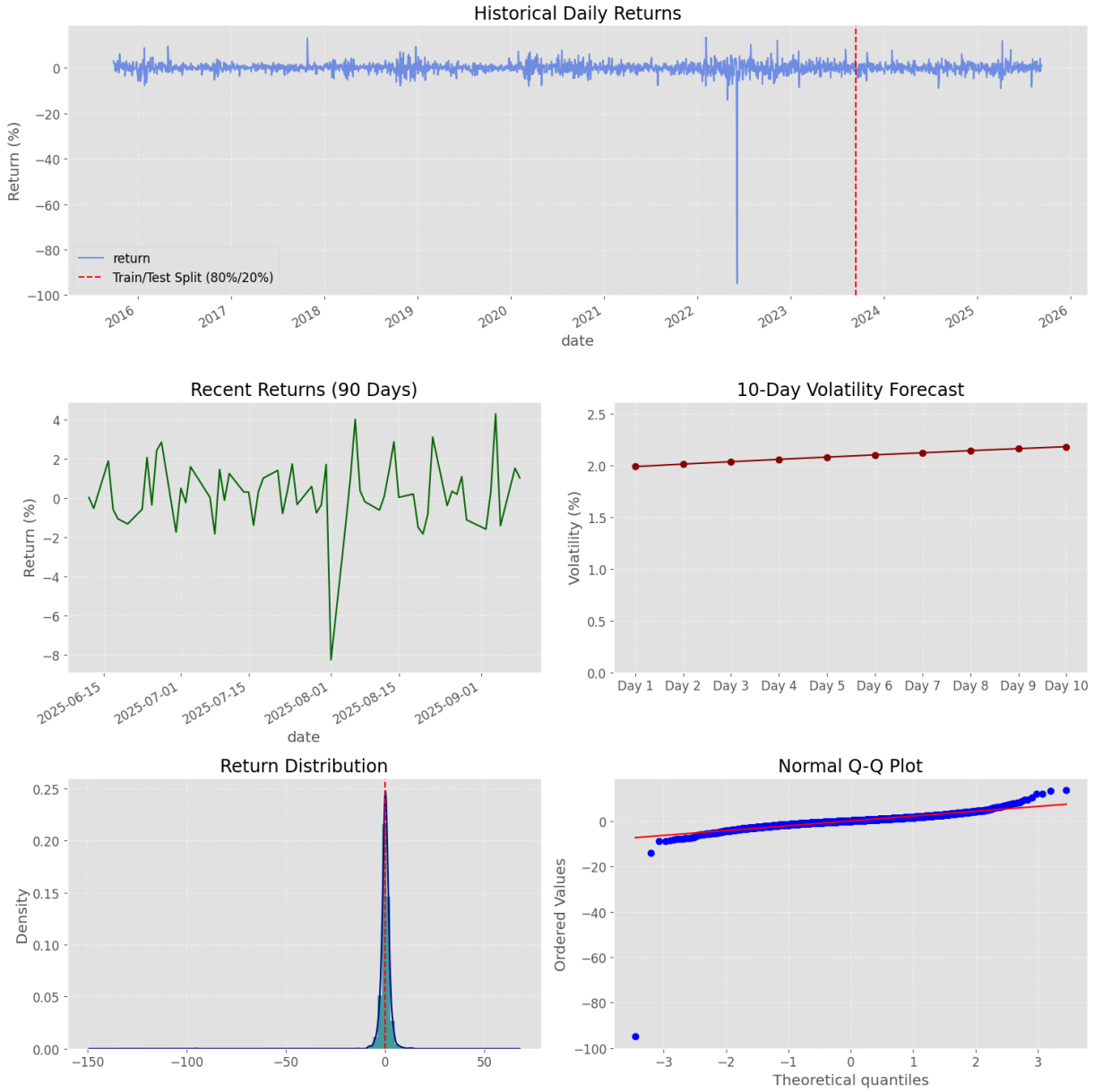
```
================================================================================
AMAZON STOCK VOLATILITY ANALYSIS REPORT
================================================================================

---------------------- Model Parameters and Diagnostics ------------------
+---------------------------+----------+-----------------------------------
| Parameter                 | Value    | Interpretation
|
+===========================+==========+===================================
| Omega (ω)                 | 0.166498 | Baseline volatility level
|
+---------------------------+----------+-----------------------------------
| Alpha (α)                 | 0.165878 | Impact of recent shocks
|
+---------------------------+----------+-----------------------------------
| Beta (β)                  | 0.816288 | Long-term volatility memory
|
+---------------------------+----------+-----------------------------------
| Persistence (α+β)         | 0.9822   | Duration of volatility shocks (High
>0.9)  |
+---------------------------+----------+-----------------------------------
| Degrees of Freedom (ν)    | 4.21     | Tail thickness (Lower = fatter tail
s)      |
+---------------------------+----------+-----------------------------------
| AIC                       | 8051.10  | Model fit (Lower = better)
|
+---------------------------+----------+-----------------------------------
| BIC                       | 8079.10  | Model fit (Lower = better)
|
+---------------------------+----------+-----------------------------------
| ARCH-LM p-value           | 1.0000   | Remaining ARCH effects (<0.05 signi
ficant) |
+---------------------------+----------+-----------------------------------
| 1-Day VaR (95%)           | 0.02%    | Potential daily loss threshold
|
+---------------------------+----------+-----------------------------------
| Day 1 Volatility Forecast | 1.99%    | Expected near-term volatility
|
+---------------------------+----------+-----------------------------------

---------------------- Strategic Investment Implications ------------------
```

Factor	Implication	Action
High Volatility Persistence (α+β>0.9)	Long-lasting volatility shocks	Reduce exposure, use protective puts
High GARCH Term (β>0.7)	Stable volatility regimes	Long-term risk budgeting
Fat Tails (ν<5)	Higher extreme return risk	Increase hedging, monitor tail risks
High Forecasted Volatility (1.99% > 1.48%)	Increased near-term risk	Reduce position sizes, consider hedging
1-Day VaR (0.02%)	Potential loss threshold	Set stop-loss levels, allocate capital accordingly

Forecasted Volatility

Horizon	Volatility (%)
Day 1	1.99%
Day 2	2.01%
Day 3	2.04%
Day 4	2.06%
Day 5	2.08%
Day 6	2.10%
Day 7	2.12%
Day 8	2.14%
Day 9	2.16%
Day 10	2.18%

```
/tmp/ipython-input-1952942018.py:86: FutureWarning: last is deprecated and w
ill be removed in a future version. Please create a mask and filter using `.
loc` instead
  returns.last('90D').plot(ax=ax2, title='Recent Returns (90 Days)', color
='darkgreen')
```



In [ ]:
```
#@title Convert ipynb to HTML in Colab
# Upload ipynb
from google.colab import files
f = files.upload()

# Convert ipynb to html
import subprocess
file0 = list(f.keys())[0]
_ = subprocess.run(["pip", "install", "nbconvert"])
_ = subprocess.run(["jupyter", "nbconvert", file0, "--to", "html"])

# download the html
files.download(file0[:-5]+"html")
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Capstone_project_EMBA_Valar.ipynb to Capstone_project_EMBA_Valar.ipyn
b
```

In [ ]: