

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑΤΑ ΔΙΕΘΝΩΝ & ΕΥΡΩΠΑΪΚΩΝ ΟΙΚΟΝΟΜΙΚΩΝ
ΣΠΟΥΔΩΝ ΚΑΙ ΟΙΚΟΝΟΜΙΚΗΣ ΕΠΙΣΤΗΜΗΣ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ ΚΑΙ ΤΡΑΠΕΖΙΚΗ

**One-Day-Ahead Tail Risk
Forecasting: VaR–ES
Calibration and
Backtesting Across Stress
Events**

ΓΡΙΒΑΣ ΓΡΗΓΟΡΙΟΣ

**Διπλωματική εργασία υποβληθείσα προς μερική εκπλήρωση
των απαραίτητων προϋποθέσεων**

για την απόκτηση του

Διπλώματος Μεταπτυχιακών Σπουδών

Αθήνα

Ιανουάριος, 2026



Επιβλέπων καθηγητής:

ΗΛΙΑΣ ΤΖΑΒΑΛΗΣ

Τμήμα Οικονομικής Επιστήμης

Εξεταστής καθηγητής:

ΝΙΚΟΛΑΟΣ ΤΟΠΑΛΟΓΛΟΥ

Τμήμα Διεθνών και Ευρωπαϊκών Οικονομικών σπουδών

Εξεταστής καθηγητής:

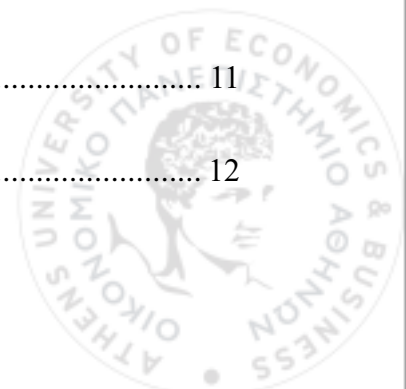
ΣΠΥΡΙΔΩΝ ΣΚΟΥΡΑΣ

Τμήμα Διεθνών και Ευρωπαϊκών Οικονομικών σπουδών

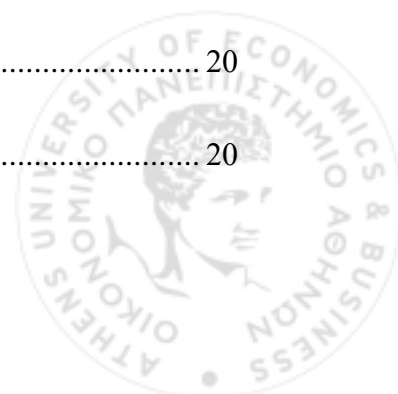


Contents

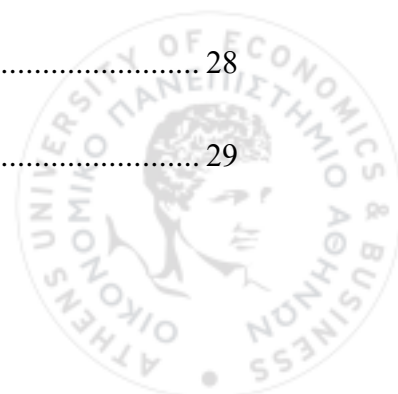
Contents.....	1
Abstract	6
Chapter 2 – Introduction.....	7
2.1 Research objective.....	8
2.2 Research Questions.....	8
2.3 Data and empirical design.....	9
Chapter 3 – Literature Review.....	9
3.1 Market risk and the role of tail risk.....	9
3.2 Value at Risk as a risk benchmark and its limitations	10
3.3 Regulatory shift from VaR to ES under the Basel market risk framework.....	10
3.4 VaR and ES forecasting approaches: HS, Gaussian, Student-t, and EWMA	11
3.4.1 Historical Simulation (HS).....	11
3.4.2 Gaussian parametric VaR/ES.....	12



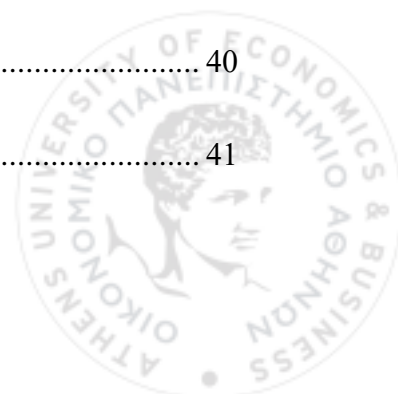
3.4.3 Student-t parametric VaR/ES.....	12
3.4.4 EWMA volatility VaR/ES	12
3.5 Forecast evaluation and backtesting philosophy	13
3.6 VaR Backtesting: exceedances, coverage, and independence.....	13
3.6.1 Unconditional coverage (Kupiec test)	13
3.6.2 Independence and conditional coverage	14
3.6.3 Regulatory backtesting culture	14
3.6.4 Beyond exceptions: density and distributional evaluation.....	14
3.7 ES Backtesting	15
3.7.1 Problems of backtesting ES like VaR	15
3.7.2 ES evaluation framework.....	15
3.7.3 Recent ES backtesting methods: calibration, severity, and estimation error	15
3.8 Stress periods and the motivation for event-window analysis.....	16
3.9 Summary.....	17
4. Hypothesis Development and Formulation	18
5. Research Design and Methodology	19
5.1 Research objective and motivation.....	19
5.2 Methodology Questions.....	20
5.3 Data description and sample construction	20
5.3.1 Asset selection	20



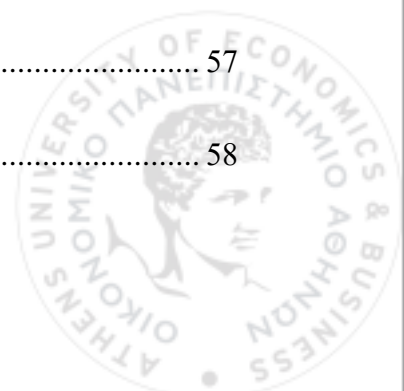
5.3.2 Data source, frequency, and sample period.....	21
5.3.3 Return construction.....	21
5.4 Empirical forecasting framework.....	22
5.4.1 One-step-ahead forecasting setup.....	22
5.4.2 Confidence levels used	22
5.5 Risk measure definitions (VaR and ES).....	23
5.5.1 Value-at-Risk (VaR).....	23
5.5.2 Expected Shortfall (ES / CVaR)	23
5.6 Risk forecasting models implemented.....	23
5.6.1 Historical Simulation (HS).....	24
5.6.2 Parametric Gaussian (Normal) VaR and ES.....	24
5.6.3 Student- <i>t</i> VaR and ES	25
5.6.4 EWMA VaR and ES.....	25
5.7 VaR backtesting methodology.....	26
5.7.1 VaR breach indicator.....	26
5.7.2 Kupiec unconditional coverage test.....	27
5.7.3 Christoffersen independence test.....	27
5.7.4 Conditional coverage test.....	28
5.7.8 ES backtesting approach	28
5.8 Event-window (stress-period) analysis.....	29



5.8.1 Motivation	29
5.8.2 Event window definition	30
5.8.3 Events discussion	30
5.8.4 Event-based evaluation outputs	32
5.9 Implementation details and reproducibility	32
5.9.1 Step-by-step empirical pipeline	32
5.9.2 Software tools	32
5.10 Summary	33
Chapter6 — Data Description	33
6.1 Chapter overview	33
6.2 Describe Out-of-Sample data	34
6.3 VaR and ES forecast levels	35
6.3.1 Full-sample forecast distribution (VaR)	35
6.3.2 Full-sample forecast distribution (ES)	37
6.3.3 VaR–ES gap behaviour	37
6.4 Event Analysis	39
Chapter7 — Empirical Results	40
7.1 Chapter overview	40
7.2 VaR backtesting results	40
7.2.1 Breach counts and breach rates	41



7.2.2 Kupiec unconditional coverage test.....	43
7.2.3 Christoffersen independence test.....	44
7.2.4 Conditional coverage test.....	45
7.2.5 Ranking summary.....	46
7.3 Expected Shortfall evaluation	48
7.3.1 Full-sample ES calibration results.....	48
7.4 Event-window results	49
7.4.1 Event windows included	49
7.4.2 Forecast level behaviour during stress.....	50
7.4.3 The best model during the backtest period	52
7.4.4 The best model during the event analysis	53
7.5 Summary.....	53
Chapter 8 - Conclusion.....	54
8.1 Building the dataset.....	54
8.2 Data Description.....	55
8.3 Backtesting Results	55
8.4 Event Period Analysis.....	56
8.5 Hypotheses Evaluation	57
8.6 Limitations	57
8.7 Future Research.....	58



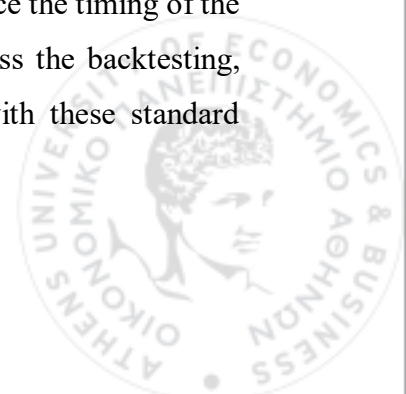
8.8 Summary.....	58
9. References.....	59
10. Appendix.....	64
.....	65

Abstract

This dissertation assesses the accuracy of various market risk models in forecasting the one-day-ahead downside risk of an equity index, specifically the S&P 500. This analysis will compare Value at Risk and expected Shortfall, produced by four models: Historical Simulation (HS), Gaussian parametric, Student-t parametric, and EWMA volatility-based. The dataset contains daily S&P 500 prices downloaded from Refinitiv DataStream and covers the period from 1/1/2014 to 31/12/2025. The study uses daily log returns and implements a rolling-window out-of-sample framework with a 250-day trading estimation window, producing forecasts at $\alpha = 95\%$ and $\alpha = 99\%$.

The performance of the models is calculated using standard VaR backtesting tools, including breach-rate analysis, the Kupiec test, the Christoffersen independence test, and the conditional coverage test, which jointly evaluate not only the frequency of breaches but also their timing. Because ES cannot be validated with similar tools, it is assessed using a joint VaR-ES calibration framework, consistent with the model's idea that VaR and ES should always be evaluated together. The dissertation is also doing an event-based analysis, where it examines which of the models performed well under shock scenarios such as COVID or the war in the Middle East in 2025.

The empirical results at $\alpha = 95\%$ suggest that all models produce a breach rate that is close to the theoretical one. However, only EWMA consistently avoids rejection once the timing of the breaches is also taken into account. At $\alpha = 99\%$, none of the models pass the backtesting, highlighting the difficulty of forecasting in extreme shock scenarios with these standard



models. Overall, the results suggest that model issues become more pronounced during crisis periods and that calibrating the model in the extreme tails is challenging.

Chapter 2 – Introduction

This dissertation will focus on risk management. Risk management involves quantifying losses from movements in market prices over a specific time period. In practice, the focus is on measuring results under extreme scenarios rather than just calculating average performance. Two measures are the most common in the literature and practice: Value-at-Risk and Expected Shortfall. VaR provides a single number for losses at a confidence level, while ES measures average losses above the VaR level, capturing not only how often losses occur but also how severe they are.

Financial returns are usually different from what normal distributions assume, they exhibit fat tails, skewness, and volatility clusters. These empirical results suggest that models built on constant variance and thin tail assumptions can underestimate risk when risk matters most, which is during market shocks.

Also, in recent years, there has been an increase in discussions about model frameworks and governance. Every day, there is a higher need to assess the tail risk more comprehensively, and VaR and ES are often treated as a joint forecasting and validation problem. In particular, VaR backtesting includes many tests that evaluate accuracy in terms of success ratio and the timing of the risk. At the same time, ES is trickier because it is not just a simple threshold and cannot be validated by frequencies alone. For that reason, it is common to use joint VaR-ES tests that aim to combine VaR/ES assessment.

For this reason, this dissertation will evaluate how different VaR/ES models estimate the one-day-ahead downside risk for the well-established S&P 500 equity index. The analysis is designed to provide out-of-sample results using a rolling forecasting framework with a fixed window of 250 days, and to evaluate the performance of the models under both normal and stressed conditions.



2.1 Research objective

The focus of this dissertation is to compare alternative VaR/ES models to measure which of these is the most robust for forecasting the one-day-ahead tail risk. Robustness is assessed in two ways:

1. VaR backtest accuracy: Here, we focus on whether the expected breaches of the models match the realised results, and if these breaches are time independent
2. VaR/ES joint calibration: Here, we evaluate the prediction of ES combined with VaR to measure which model better accounts for the loss.

To achieve this, this thesis will use the following models:

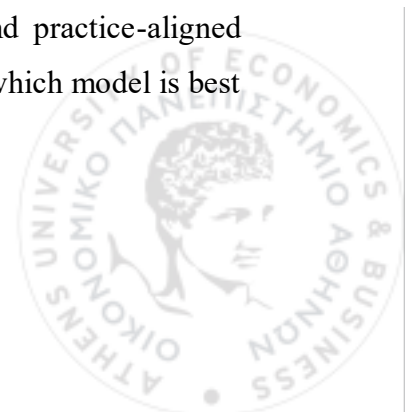
- Historic Simulation: a non-parametric model, which is used as a benchmark and is calculated on the rolling empirical returns
- Gaussian parametric VaR/ES: A model that uses the normal distribution assumption
- Student-t parametric VaR ES: This model extends the normal assumption by allowing for heavier tails
- EWMA-based VaR/ES: Introduces volatility that is time-dependent in order to capture the volatility clustering.

2.2 Research Questions

The dissertation will try to answer the following research questions:

1. How different are the VaR and ES forecasted products from the models discussed in the 2.1 chapter?
2. Which model performs better in backtesting results for each scenario and event?
3. Do fat tails and volatility approaches behave more robustly than the Gaussian and HS during volatility periods?

The contribution of this dissertation will be to provide a full, clear, and practice-aligned comparison of common VaR/ES models and to help practitioners identify which model is best and under which scenario.



2.3 Data and empirical design

The empirical analysis uses S&P 500 daily prices from Refinitiv DataStream and covers the period from 2014 to the end of 2025. Returns are computed as daily log returns, and all models are estimated using a 250-trading-day rolling window to produce one-day-ahead VaR and ES forecasts. The first 250 observations are for training, and the out-of-sample periods contain 2,766 forecasts. The results are produced at $\alpha = 95\%$ and $\alpha = 99\%$. The dissertation then focuses on evaluating the selected VaR and ES models. This evaluation will be performed by:

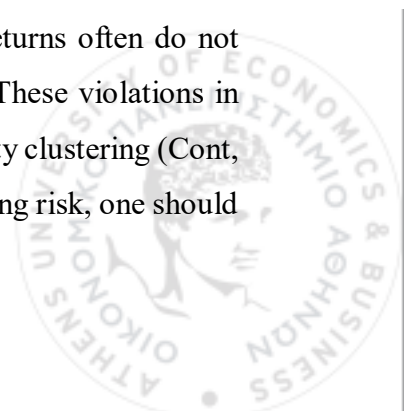
1. Var Backtesting: using breach rate, and a test to measure the coverage and the impedance of breaches
2. ES Evaluation: using VaR-ES joint calibration approach because it cannot be validated alone with simple hypothesis tests
3. Event period analysis: here we re-evaluate the model behaviour, but we focus only on pre-defined stress periods such as the COVID crash or the recent tariff crash

Chapter 3 – Literature Review

3.1 Market risk and the role of tail risk

Efforts to measure and quantify losses that are coming from movements in market prices are called market risk. An example of market risk is the movement in an equity index, such as the S&P 500, over a specific horizon (Basel Committee on Banking Supervision, 2019). Practitioners and regulators usually use two metrics to capture the market risk: Value-at-Risk (VaR) and Expected Shortfall (ES). For VaR, it is defined as the α -quantile of the loss distribution over a time period t . At the same time, ES measures the conditional average loss given that the VaR threshold has been defined and therefore aims to measure not only how often we have losses but also how severe they can be (Meng et al., 2020; Rockafellar & Uryasev, 2000).

An important reason to focus on the tails of return distributions is that returns often do not conform to the Gaussian assumptions underlying many financial models. These violations in assumptions are mainly concentrated around fat tails, skewness, and volatility clustering (Cont, 2001; Ratliff-Crain et al., 2025). Empirical results imply that when forecasting risk, one should



always allow for complex volatility patterns, especially during stress periods (Engle, 1982; Bollerslev, 1986).

A known volatility pattern in finance is called volatility clustering. This phenomenon suggests that high volatility is followed by high volatility and vice versa. That empirical evidence motivates the creation of a model that captures this pattern and does not assume constant volatility over time (Engle, 1982; Bollerslev, 1986).

3.2 Value at Risk as a risk benchmark and its limitations

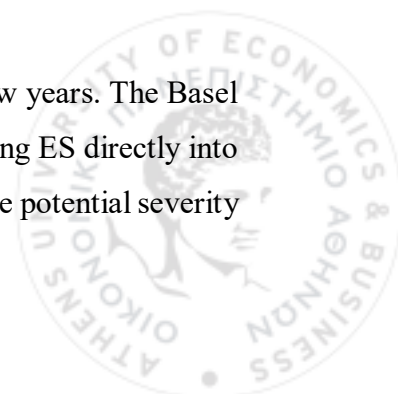
VaR is a popular metric because it can be calculated as a single number and it is easy to interpret and communicate (Jorion, 2007). This simplicity makes it an attractive metric for risk reports, risk frameworks, and capital rules. However, a significant problem with the model is that it provides no information on the magnitude of losses if the VaR threshold is breached (Yamai & Yoshida, 2005). For example, two portfolios may have the same VaR but differ in tail risk and expected losses. That motivates the adoption of ES as a metric that could provide more information on retail risk in both academic work and regulations (Artzner et al., 1999; Rockafellar & Uryasev, 2000).

The critique of the simple VaR models is quite extensive in the literature. Artzner, Delbaen, Eber, and Heath (1999) formalise the rules defining a risk measure and show that VaR can violate them. On the other hand, ES type models align better with the conditions of a proper market risk framework. While coherence is not the only criterion in the paper, they explain why VaR alone cannot properly measure a portfolio's risk.

In addition, Rockafellar and Uryasev provide their own optimization framework for ES, showing that ES-type measures are helpful not only for monitoring risk but also for controlling it and hedging portfolio exposure problems, making ES as applicable, if not more, than VaR.

3.3 Regulatory shift from VaR to ES under the Basel market risk framework

The global regulatory framework has evolved significantly over the past few years. The Basel Committee has advanced its VaR-based model approach by also incorporating ES directly into the internal model framework, reflecting the view that ES is needed to capture potential severity



and to provide more detailed information to be used in addition to VaR (Basel Committee on Banking Supervision, 2019).

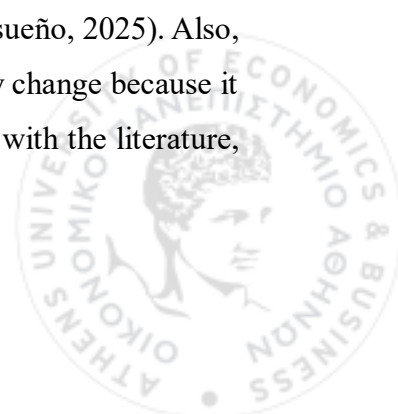
Here, we need to mention that none of the regulatory frameworks eliminate the need for VaR; they focus on combining with ES to capture risk across the whole distribution and make it easier to quantify not only the point of the distribution, but also the loss in its tails. This is why in most modern validation systems, VaR and ES are usually treated as a joint forecasting problem (Nolde & Ziegel, 2017; Fissler et al., 2015). The literature, therefore, motivates the development of a dissemination that forecasts both VaR and ES, evaluates their outcomes and behaviours using established tests for model calibration, and assesses them under different conditions (Kupiec, 1995; Christoffersen, 1998; Nolde & Ziegel, 2017).

3.4 VaR and ES forecasting approaches: HS, Gaussian, Student-t, and EWMA

In the literature, there is a broad range of various models, from simple non-parametric approaches to fully complex stochastic volatility models and methods. This dissertation will focus on a small number of widely used approaches that represent distinct modelling philosophies. The models that this dissertation will focus on for VaR and ES are: Historical Simulation, Gaussian parametric, Student-t parametric, and EWMA-based. They have been chosen because they offer different perspectives on how these metrics can be used, and each aims to solve a problem empirically observed in the returns.

3.4.1 Historical Simulation (HS)

Historical simulation estimates VaR and ES by using empirical quantiles and tail averages from a rolling window of historical returns (Manganelli, 2001). This model is helpful because of its simplicity: it makes no assumptions and does not rely on Gaussian distributions. However, because of this simplicity, it has some serious drawbacks (Pritsker, 2006). It is well known that it reacts too slowly to market shifts, especially when the estimation window does not have enough data similar to current market conditions (Pritsker, 2006; García-Risueño, 2025). Also, this model will most likely perform quite poorly during periods of volatility change because it does not account for conditional variance. These limitations are consistent with the literature,



which suggests that using only historical observations is not enough to capture returns that change over time.

3.4.2 Gaussian parametric VaR/ES

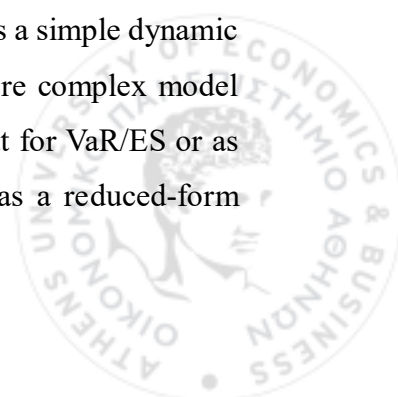
The normal parametric approach assumes that returns follow a normal distribution, with a mean and variance that can be estimated from a selected period (for this dissertation, the rolling window) (Morkūnaitė et al., 2024). Because of this assumption, VaR and ES have closed-form solutions, making them easy to calculate and interpret (Michaelides & Poudyal, 2024). This approach, as far as I can tell, is computationally efficient and easy to work with. However, the literature is usually criticised for its fat-tailed assumptions, even though we can empirically see that real returns are heavy-tailed and clustered (Ratliff-Crain et al., 2023). As a result, Gaussian VaR/ES will most of the time perform poorly during economic distress, and it should be used only as a baseline model.

3.4.3 Student-t parametric VaR/ES

The Student-t distribution extends the Gaussian model by allowing heavier tails through the degrees of freedom parameter. This is quite a relevant extension, as empirical evidence shows that extreme returns occur more often than the simple Gaussian model predicts. Under the Student-t assumptions, both VaR and ES can be analytically tractable, which makes them still efficient models. For that reason, this methodology can be considered a middle ground between pure non-parametric models and dynamic ones. For that reason, this distribution is widely used by practitioners as a robust alternative to the Gaussian model, especially for modelling its fat tails.

3.4.4 EWMA volatility VaR/ES

The EWMA model captures volatility by using exponentially decaying weights, allowing recent observations to influence the variance more strongly than older ones. In simple terms, the volatility of the previous day is more important than the volatility of one month ago (Longerstaey & Spencer, 1996). The approach is popular because it provides a simple dynamic tool that captures volatility clustering without using a full parametric, more complex model like GARCH. In practice, this model is either used as direct volatility input for VaR/ES or as part of the simulation approach. In the literature, EWMA is often seen as a reduced-form



response to GARCH models (Engle, 1982; Bollerslev, 1986; Alexander & Dakos, 2023). If volatility is persistent and clustered, the risk forecasts should be conditioned on a variance process rather than assume that it is constant (Engle, 1982; Bollerslev, 1986).

3.5 Forecast evaluation and backtesting philosophy

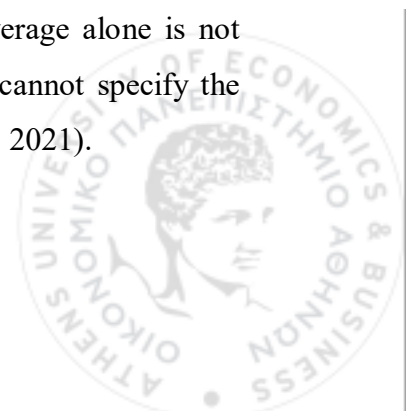
In the literature, the main approach to model and assess a risk forecasting model is to train it and then evaluate it on data it has not seen during calibration (West, 1996; Giacomini & White, 2006). Rolling-window forecasting is therefore used in the literature to mimic real-time risk modelling and to avoid data leakage (Giacomini & White, 2006). The dissertation's one-day-ahead rolling design aligns with that idea. To evaluate model accuracy, evaluation frameworks have been developed to assess VaR and ES model performance, focusing on the models' main objectives. This dissertation will focus on some of these.

3.6 VaR Backtesting: exceedances, coverage, and independence

Backtesting VaR is intuitive; the main objective is to ensure that the model's expected number of breaches is similar to those that will occur in reality. That said, various backtesting methods aim to measure it properly (Lopez, 1999; Zhang & Nadarajah, 2017). Some of the most common in the literature are the Kupiec test (1995), Christoffersen (1998), and the conditional variable. These tests focus on evaluating VaR based on both the number of breaches and their timing (Christoffersen, 1998; Zhang & Nadarajah, 2017).

3.6.1 Unconditional coverage (Kupiec test)

Kupiec (1995) proposed a likelihood ratio test to determine whether the observed breach frequency matches the VaR confidence level (Kupiec, 1995; Leung, 2021). This unconditional coverage test is attractive to practitioners and the literature because it is simple and aligns with the regulatory framework, which focuses on the number of breaches (either too many or too few) (Lopez, 1999; Ben Ayed et al., 2024). However, unconditional coverage alone is not sufficient, as it may have the correct coverage expectation rate but still cannot specify the correct timing of the breaches (Christoffersen, 1998; Leung, 2021; Tsafack, 2021).



3.6.2 Independence and conditional coverage

In 1998, Christoffersen expanded the VaR evaluation by testing whether the exceptions are also independent over time. He did that by using a Markov chain structure. His logic is that clustered exceptions suggest the model fails to respond enough to changing volatility in the markets, which is especially relevant given volatility clustering (Christoffersen, 1998; Berger, 2021). The conditional coverage test combines the frequency test from Kupiec with the independence tests and performs a strict test of VaR accuracy.

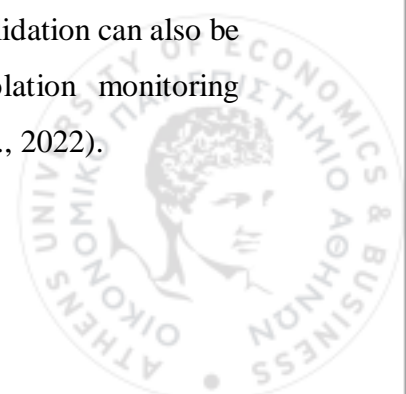
These tests are quite important because they connect VaR backtesting with empirical market dynamics, suggesting that volatility is time-varying and persistent, and, for that reason, a good VaR model should adapt quickly (Christoffersen, 1998; Leung et al., 2021).

3.6.3 Regulatory backtesting culture

The regulatory use of backtesting helped with the idea that the internal VaR model should continuously monitor realised P&L and that systematic underestimation of risk should be followed by capital adjustments (Lopez, 1999; Berkowitz & O'Brien, 2002). The Basel Committee's earlier supervisory framework on the use of backtesting and model monitoring framed exception counting and the model overfitting process, and remains today one of the most important methodologies for how banks structure VaR validation reporting (Lopez, 1999; McCullagh, 2023).

3.6.4 Beyond exceptions: density and distributional evaluation

A further critique of pure expectation-based backtesting models is that they usually use limited information. They focus on whether returns exceed a single quantile, for example, and they completely ignore the rest of the predictive distribution. Berkowitz (2001) proposed a density of casting evaluation method that is based on transforming the probability distribution, enabling a more distribution-sensitive adjustment (Abboud et al., 2021; Cuoco & Liu, 2006; Frésard et al., 201). While density tests are not always included in empirical papers, they provide an important advancement in VaR modelling by suggesting that risk model validation can also be framed as distribution calibration rather than only as threshold violation monitoring (Christoffersen & Pelletier, 2004; Lopez, 1999; Sizova, 2023; Duncan et al., 2022).



At the regulatory level, Lopez (1996) also reviews approaches for evaluating VaR models, and emphasized that VaR is a forecast that must be tested and examined using statistical approaches with supervision and internal model governance

3.7 ES Backtesting

3.7.1 Problems of backtesting ES like VaR

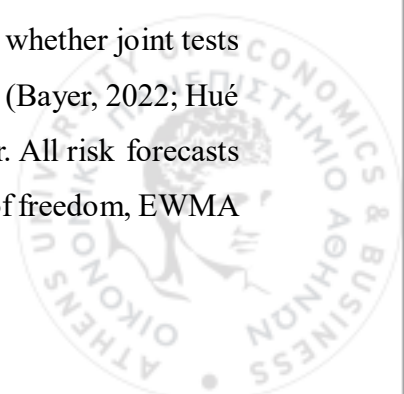
Unlike VaR, ES is not a threshold; it represents the average loss in the tail after the VaR threshold (Du & Escanciano, 2017; Nolde & Ziegel, 2017). For that reason, a simple breach indicator cannot determine whether ES is correct (Du & Escanciano, 2017; Fissler & Ziegel, 2016). This creates a practical issue: a model can produce ES that are not conservative enough but still have a really healthy VaR expectation rate, which is close to reality (Barendse et al., 2023; Hué et al., 2024). In other words, ES validation needs its own frameworks that assess both the frequency of tail events and also the severity of loss in these events.

3.7.2 ES evaluation framework

A major development in the ES backtesting literature is the recognition that ES, which cannot be quantified easily on its own, can be quantified jointly with VaR (Gneiting, 2011; Nolde & Ziegel, 2017). This observation enables the construction of loss functions for the pair of VaR and ES, rather than ES alone, which then allows forecasting comparisons and backtesting using similar statistical tools to those used for VaR (Fissler, Ziegel, & Gneiting, 2015; Nolde & Ziegel, 2017). Fissler and Ziegel (2016) have provided a theoretical framework for combining these two metrics and discussed how the practical implementation of ES can be evaluated when paired with VaR. This joint model supported several models that treat VaR and ES forecasts as joint objects (Patton, Ziegel, & Chen, 2019). VaR identifies the tail region, while ES summarises the severity within that region, and both must agree with the realised data.

3.7.3 Recent ES backtesting methods: calibration, severity, and estimation error

Modern ES backtesting research has moved beyond the early debates about whether joint tests address issues such as severity, time dependence, and estimation uncertainty (Bayer, 2022; Hué et al., 2024; Wang et al., 2025). A key practical issue is the estimation error. All risk forecasts are calculated from a set of parameters, such as the mean, variance, degrees of freedom, EWMA



components, and others, and a simple backtest can misinterpret these results as a model failure (Barendse et al., 2023; Dimitriadis & Schnaitmann, 2021). Barendse, Kole, and van Dijk (2023) show that estimation errors can affect backtest results and propose a new robust version of the test that accounts for these effects, which are relevant for the rolling-window implementation and cannot be ignored in relatively small samples (Barendse et al., 2023).

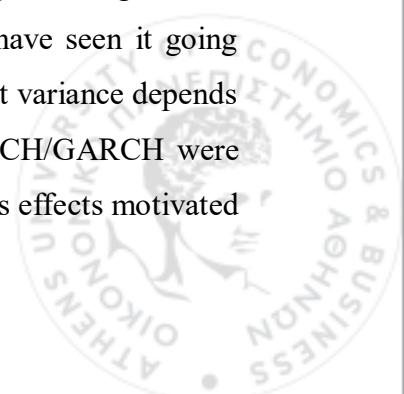
Another framework focuses on separating the frequency of the breaches from their severity. Hué, Hurlin, and Lu (2024) proposed a duration-severity approach to ES backtesting that does not split frequency and severity components, even though ES, as a measure itself, focuses mostly on the severity of tails.

An even more recent innovation in this type of backtesting is e-backtesting. This framework proposes model-free backtesting methods for risk measures such as VaR and ES, based solely on e-values and processes. Wang and Ziegel in 2025 developed this approach and emphasised its usefulness for risk backtesting and a few modelling assumptions over the traditional parametric frameworks.

Overall, these contributions focus on two practical points. Firstly, the ES evaluation can be done effectively with a joint VaR-ES framework rather than ES alone (Fissler, Ziegel, & Gneiting, 2015; Nolde & Ziegel, 2017). Secondly, the model test addresses some of the issues with the basic backtesting tool, such as volatility clustering and estimation error, which align with the goal of producing results that are meaningful and can be used for real-world monitoring scenarios (Barendse, Kole, & van Dijk, 2023; Cont, 2001; Engle, 1982; Bollerslev, 1986; Wang et al., 2025).

3.8 Stress periods and the motivation for event-window analysis

A common issue in financial markets, especially in risk management, is that models can appear to work well during healthy economic conditions but break down under market volatility and stress. This is not only a practical observation from the literature but also a statistical one. We know that asset returns have heavy tails, skewness, and volatility clustering, meaning that the distribution of losses is not stable across time, and on the contrary, we have seen it going extreme during crisis events (Cont, 2001). Volatility clustering suggests that variance depends on time and is not constant, which is why volatility models such as ARCH/GARCH were developed in the first place (Engle, 1982; Bollerslev, 1986). The previous effects motivated



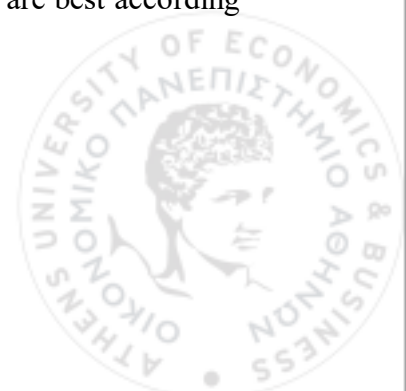
the literature to examine behaviour within specific event windows, mostly during shocks, rather than relying solely on the full-sample average, which, of course, includes mostly normal market days.

Since VaR and ES are tail measures, the usefulness of the models can be mostly seen during stressed market scenarios; for that reason, comparing model results during turbulent windows can reveal differences in responsiveness and tail fit that might be muted at the aggregated level. In particular, historical simulation can be slow to adapt when the world confronts too few situations similar to current market conditions, and its weakness becomes apparent when volatility changes rapidly (Pritsker, 2006). Generally, empirical research indicates that, regarding model risk, most models produce similar risk estimates in calm periods but diverge significantly when uncertainty rises (Boucher et al., 2014; Danielsson et al., 2014).

Another motivation is the methodology that is followed. The standard backtest can have limited power in sample sizes, which becomes more problematic when the data-generating process shifts across regimes. Time-dependent backtests show that measuring beyond standard breach counts can improve the ability to detect clusters, an issue that is more observable when volatility increases (Christoffersen & Pelletier, 2004). Similarly, much of the literature argues that focusing solely on a single tail point can miss important aspects of the predictive distribution; for this reason, it suggests using more tools for diagnostics (Berkowitz, 2001). Empirical evidence from banks' VaR systems also shows that models may appear either conservative or acceptable on average, while still failing to respond to market shifts and increases in volatility, which again supports targeting the analysis to specific events (Berkowitz & O'Brien, 2002).

3.9 Summary

The literature review examines two things. Firstly, why is it important to measure and analyse the tail risk? Why is VaR useful but incomplete as a measure by itself? Thirdly, it focuses on why ES is a good measure, why it should always be considered together with VaR, and why both VaR and ES should be validated through backtesting, and which tools are best according to the literature.



4. Hypothesis Development and Formulation

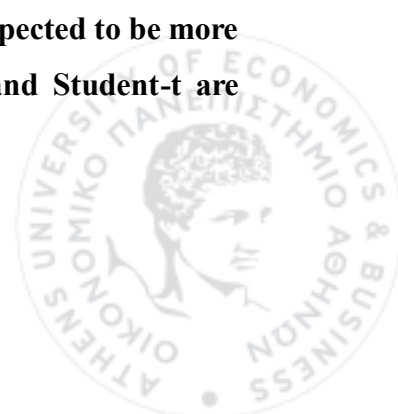
As discussed in Chapter 3, the returns are known to show fat tails and volatility clustering, meaning that extreme losses happen way more often than what a Gaussian model suggests and risk changes over time (Cont, 2001; Engle, 1982; Bollerslev, 1986). This evidence from returns suggests that models that assume thin tails and constant variance may appear acceptable on average, but they fail when volatility shifts sharply, when risk measurement matters most. This motivates us to compare four different VaR/ES modelling approaches that represent different assumptions.

- Historical Simulation: A non-parametric, simple but slow-to-react model.
- Gaussian parametric: An analytical and simple model, but with strong assumptions regarding the tails of the distributions.
- Student-t: A similar model to Gaussian, but allows the tails to be heavier
- EWMA: A time-varying volatility model that allows the volatility to change over time.

VaR can be tested using backtests that examine both frequency and timing of exceptions. However, ES cannot be validated by a simple breach rate, so as described in the literature, it can be validated in combination with VaR as a joint test (Fissler & Ziegel, 2016; Nolde & Ziegel, 2017). This dissertation's contribution is to assess whether the same model ranking holds under a single forecasting model (S&P 500, one-day-ahead rolling forecasts, 250-day window, $\alpha = 95\%$ and $\alpha = 99\%$).

This dissertation will focus on checking the following hypotheses:

- H1. In the full out-of-sample period, EWMA will be more robust than Student, Gaussian, and HS for VaR/ES forecasting at $\alpha = 95\%$ and $\alpha = 99\%$. Robustness is defined as (i) fewer rejections in VaR backtests (coverage and independence) and (ii) better joining VaR-ES calibration.**
- H2. Model performance is expected to be worse during stress event windows relative to the full sample. This deterioration of the model accuracy is expected to be more visible for the Gaussian and HS models, while the EWMA and Student-t are expected to be more stable.**



H3. Across all the tested models, calibration is expected to be more difficult at $\alpha = 99\%$ than at $\alpha = 95\%$, because the far tails of the distribution are rarer. For that reason, rejections in VaR/ES backtesting should be more common at the 99% level.

5. Research Design and Methodology

5.1 Research objective and motivation

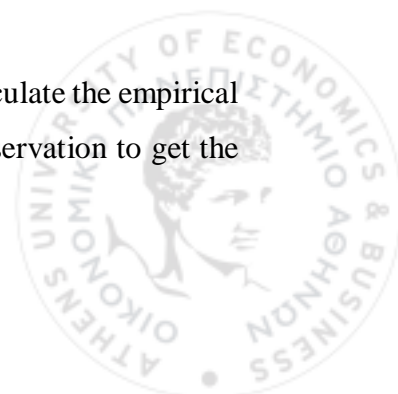
In this dissertation, the primary focus is on comparing alternative market risk forecasting methods for measuring one-day-ahead downside risk. The metrics we will use are the Value at Risk (VaR) and expected Shortfall (ES). VaR, or Value at Risk, has become quite popular in the financial industry. It is a simple statistic that practitioners can easily use. It is a single number that can be used to calculate traders' risk limits and easily fit into any dashboard, making it practical for everyday use (Jorion, 2007; Hull, 2021).

Another helpful tool among practitioners is the ES, a metric that measures the average loss in the tail. This metric addresses the main issue with VaR: it specifies a cutoff point but does not explain how significant losses can occur beyond that point (Rockafellar & Uryasev, 2000; Yamai & Yoshida, 2002; McNeil, Frey, and Embrechts, 2015).

In this dissertation, the main reason for comparing different VaR/ES methods is that the returns we observe in practice do not always conform to the assumptions of the most famous parametric models. In the literature, many studies suggest that returns do not follow a normal distribution but may exhibit fatter tails than the normal distribution would suggest, higher skewness, and, most importantly, volatility clusters. (Mandelbrot, 1963; Fama, 1965; Cont, 2001; McNeil, Frey, and Embrechts, 2015).

These patterns suggest that all models that rely on normal-distribution assumptions likely underestimate risk, particularly tail risk. This has been observed frequently during periods of market volatility (Cont, 2001; McNeil, Frey, and Embrechts, 2015). This is why, in this dissertation, we examine the following models.

1. **Historical Simulation (HS):** This method uses historical data to calculate the empirical quantiles as a benchmark, which means that we rely a lot on historical observation to get the distributions and not on any theoretical distribution



2. **Student-t:** This is a parametric style model, which is helpful because it can handle the issue of fat tails, and it is well-suited for modelling returns with outliers and extreme values. that may show extreme negative returns.

3. **EWMA:** This model tries to handle the central issue of HS VaR by constantly updating the volatility over time in order to solve the problem of clustering, which was discussed earlier.

Finally, an important feature of this analysis is that the model can perform well across different market conditions; some models perform best in calm markets, while others perform well during turbulence (Cont, 2001). For this reason, this dissertation will evaluate models both i) for the full sample and also ii) during recent tremors in the financial market, like the Russian/Ukrainian war, or the most recent tariffs from the US government. This is consistent with the idea of examining model behavior.

5.2 Methodology Questions

This dissertation answers three practical questions.

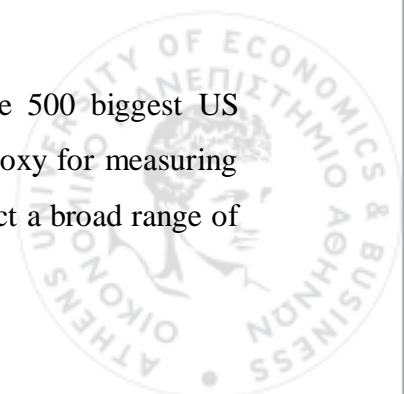
- Q1. How different are the VaR and ES forecasts produced by Historical Simulation, Gaussian, Student- t , and EWMA models at the 95% and 99% confidence levels?
- Q2. Which VaR model performs best in standard backtesting, with the breach rate close to the target level and breaches not clustered over time?
- Q3. Do Student- t (fat-tail) And do EWMA approaches behave more robustly than Gaussian VaR and HS during volatile periods?

To answer the previous questions, we will calculate the metrics discussed in the previous chapter and apply a back-testing methodology similar to that followed in the literature, for example, Kupiec (1995) and Christoffersen (1998).

5.3 Data description and sample construction

5.3.1 Asset selection

The baseline analysis uses the S&P 500 index. This index includes the 500 biggest US companies in the stock market and is generally considered an excellent proxy for measuring the performance of the US economy. A board index like this one can reflect a broad range of



market conditions and multiple market crashes over the past few years, making it useful for stress testing (Jorion, 2007; McNeil, Frey, and Embrechts, 2015).

5.3.2 Data source, frequency, and sample period

The daily price data for the S&P 500 come from the Refinitiv DataStream database, and the raw data are then cleaned and edited for use in the models we are building. The dataset covers the period from the beginning of 2014 until the end of 2025. We select this period because it offers a wide range of interesting financial events that we will examine for the accuracy of the VaR models, the events we will be discussing in the next chapter.

For this raw data, we will use daily data and calculate the one-day VaR, an approach commonly used in the industry. This fits well with how VaR is calculated and applied (Jorion, 2007; Hull, 2021). Daily observation is necessary to achieve a large enough sample size to calculate returns using the rolling method, especially for EWMA, and for our backtesting.

After we get the raw data, we use Python to prepare them for analysis. Specifically, we check the time indexes, ensure the dates are correct and fall within the intervals we want to work with, handle outliers and missing values, and, in the end, compute the VaR and ES returns (Aroussi, 2025).

5.3.3 Return construction

From the raw data, we have the daily price. Let P_t denote the raw price on day t . The daily log

returns can be calculated as:

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

The benefit of using log returns instead of simple returns is that they provide a stable score without outliers and can be added over time, making them a stable variable for modelling VaR and ES (Jorion, 2007).



5.4 Empirical forecasting framework

5.4.1 One-step-ahead forecasting setup

For this methodology, we will apply the one-step-ahead forecast: each model produces a risk forecast for the day t using information available up to day $t - 1$. The available information set is denoted by \mathcal{F}_{t-1} .

A rolling-window method is applied. For each day t , parameters (or empirical quantities) are estimated using the most recent W returns:

$$\{r_{t-W}, r_{t-W+1}, \dots, r_{t-1}\}.$$

The baseline window length in the code is:

$$W = 250,$$

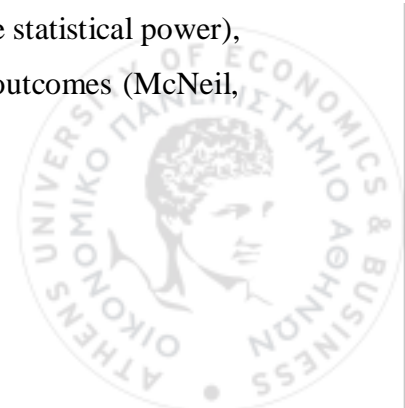
This number can be used to approximate the number of trading days in a year and has been used in many similar papers, such as those by Jorion in 2007 and McNeil, Frey, and Embrechts in 2015. The rolling setup helps us avoid creating look-ahead issues and biases and allows the model to maintain continuity over time. (Engle, 1982; Cont, 2001).

5.4.2 Confidence levels used

The analysis considers two confidence levels:

$$\alpha \in \{0.95, 0.99\}.$$

These are standard in risk management: 95% provides more breaches (more statistical power), while 99% focuses on rarer tail events and is more sensitive to extreme outcomes (McNeil, Frey, and Embrechts, 2015; Hull, 2021).



5.5 Risk measure definitions (VaR and ES)

5.5.1 Value-at-Risk (VaR)

Define $q_{t,\alpha}$ as the conditional $(1 - \alpha)$ -quantile of returns given \mathcal{F}_{t-1} , i.e.:

$$\Pr(r_t \leq q_{t,\alpha} | \mathcal{F}_{t-1}) = 1 - \alpha.$$

VaR is reported as a positive loss number:

$$\text{VaR}_{t,\alpha} = -q_{t,\alpha}.$$

This makes interpretation consistent: larger $\text{VaR}_{t,\alpha}$ means higher predicted downside risk (Jorion, 2007).

5.5.2 Expected Shortfall (ES / CVaR)

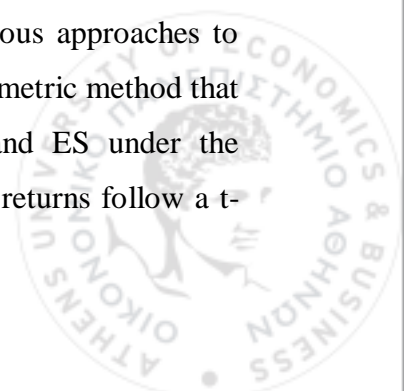
Expected Shortfall is defined as the average loss in the tail.

$$\text{ES}_{t,\alpha} = -\mathbb{E}[r_t | r_t \leq q_{t,\alpha}, \mathcal{F}_{t-1}].$$

It is considered more useful than VaR because it captures not only a threshold but also the estimated loss in extreme scenarios (Rockafellar & Uryasev, 2000; Yamai & Yoshida, 2002; Basel Committee on Banking Supervision, 2019).

5.6 Risk forecasting models implemented

As discussed in the previous chapter, this dissertation will focus on various approaches to capturing VaR. These include VAR with Historical Simulations, a nonparametric method that captures VaR using historical returns. Secondly, we calculate VaR and ES under the assumption that returns follow a normal distribution. Thirdly, we assume returns follow a t-



distribution, and finally, we approximate volatility and account for volatility clusters using EWMA.

5.6.1 Historical Simulation (HS)

Historical Simulation calculates the quantiles from the historical distribution in the following window that we have defined.

$$q_{t,\alpha}^{HS} = \text{Quantile}_{1-\alpha}(r_{t-W}, \dots, r_{t-1}),$$

So

$$\text{VaR}_{t,\alpha}^{HS} = -q_{t,\alpha}^{HS}.$$

The ES will be the average loss from the historical quantiles, as the following equation suggests.:

$$\text{ES}_{t,\alpha}^{HS} = -\frac{1}{N_{\text{tail}}} \sum_{i: r_i \leq q_{t,\alpha}^{HS}} r_i,$$

where N_{tail} is the number of observations in the window. HS is straightforward because it does not assume any distribution, but it cannot react to events because all its weights are based on past observations (McNeil, Frey, and Embrechts, 2015; Cont, 2001).

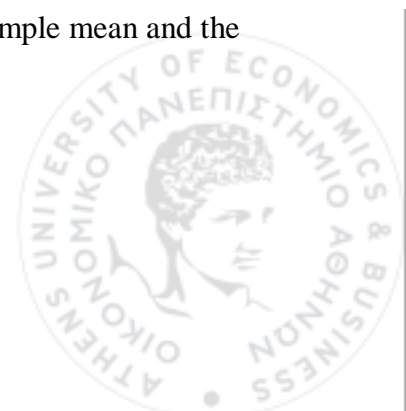
5.6.2 Parametric Gaussian (Normal) VaR and ES

If we assume the returns follow a normal distribution, we assume they do.

$$r_t | \mathcal{F}_{t-1} \sim \mathcal{N}(\mu_t, \sigma_t^2),$$

where μ_t and σ_t are estimated from the rolling window dataset using the sample mean and the sample standard deviation.

Let $z_{1-\alpha} = \Phi^{-1}(1 - \alpha)$. Then:



$$\text{VaR}_{t,\alpha}^N = -(\mu_t + z_{1-\alpha}\sigma_t).$$

Because we assume that returns follow a normal distribution, it turns out that EL will have an analytical solution, which is:

$$\text{ES}_{t,\alpha}^N = -\left(\mu_t - \sigma_t \frac{\varphi(z_{1-\alpha})}{1-\alpha}\right),$$

where $\varphi(\cdot)$ is the standard normal density.

If we use VaR/ES calculated from a normal distribution, it is widely used and easy to interpret, but it can underestimate tail risk under fat-tailed distributions (Jorion, 2007; Cont, 2001).

5.6.3 Student-*t* VaR and ES

To model heavier tails and address one issue with HS VaR, we assume returns follow a *t*-distribution.

$$r_t \mid \mathcal{F}_{t-1} \sim t_\nu(\text{loc}_t, \text{scale}_t),$$

where ν , loc_t , and scale_t are estimated by maximum likelihood on each period.

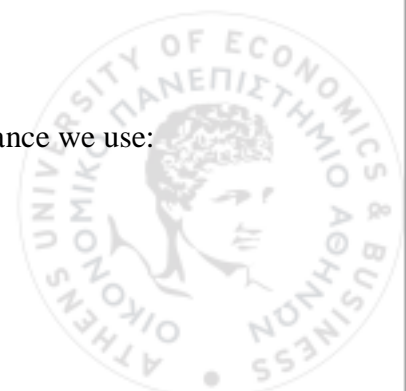
We assume that $q_{t,\alpha}^t$ be the fitted $(1 - \alpha)$ -quantile from the Student-*t* distribution; then:

$$\text{VaR}_{t,\alpha}^t = -q_{t,\alpha}^t.$$

ES under the Student-*t* distribution can be computed using an analytical approximation, yielding a tail-mean measure consistent with heavy-tailed distributions (McNeil, Frey, and Embrechts, 2015). The motivation is that the student-*t* allocates more probability mass to the tails than Gaussian models, which may improve performance during turbulent periods (Cont, 2001).

5.6.4 EWMA VaR and ES

With EWMA, we capture the volatility using a recursive update in the variance we use:



$$\sigma_t^2 = \lambda \sigma_{t-1}^2 + (1 - \lambda) r_{t-1}^2,$$

Where the daily decay value:

$$\lambda = 0.94.$$

These specific values are widely used in practice for daily risk because they increase responsiveness to recent volatility (Longerstaey & Spencer, 1996).

EWMA VaR assumes conditional normality with time-varying σ_t and $\mu = 0$:

$$\text{VaR}_{t,\alpha}^{\text{EWMA}} = -(z_{1-\alpha} \sigma_t).$$

Similarly, EWMA ES uses the normal ES formula with $\mu = 0$:

$$\text{ES}_{t,\alpha}^{\text{EWMA}} = \sigma_t \frac{\varphi(z_{1-\alpha})}{1 - \alpha}.$$

We decided to go with these models because they target volatility clustering and respond more quickly than classic HS when volatility changes (Engle, 1982; Cont, 2001; Longerstaey & Spencer, 1996).

5.7 VaR backtesting methodology

5.7.1 VaR breach indicator

We define a VaR breach as the moment when the return falls below the VaR threshold.

$$I_t = \mathbf{1}(r_t < -\text{VaR}_{t,\alpha}),$$

where $I_t = 1$ indicates a breach and $I_t = 0$ otherwise.

If the VaR model is correctly calibrated at the level α , then:

$$\mathbb{E}[I_t] = 1 - \alpha.$$



For T out-of-sample forecasts, the expected number of breaches is:

$$(1 - \alpha)T.$$

5.7.2 Kupiec unconditional coverage test

In this dissertation, to evaluate whether VaR models produce the correct number of breaches, we will apply the Kupiec coverage Test (Kupiec, 1995). This test assesses whether VaR is estimated at a confidence level α ; for example, 95%. Then the breaches should be rare and occur with this expected probability. That means that the probability that we will have a VaR breach at any point should be: $p = 1 - \alpha$

For example, if we have a 99% VaR, we expect about 1% of days to be breached. The test checks whether the observed breach rate in the backtest is statistically consistent with our expectations.

The null hypothesis is that the model has the correct unconditional coverage, meaning that the breach probability equals the theoretical one:

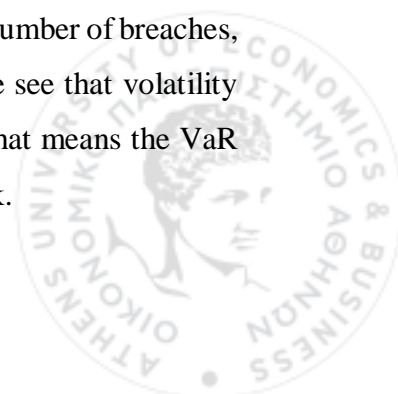
$$H_0: \Pr(\text{breach}) = 1 - \alpha$$

The Kupic test is a calibration test; its job is not to measure losses beyond VaR, and it does not check for a cluster in these breaches. It only checks that the frequency of the breaches matches the theoretical levels.

If the null hypothesis is rejected, it suggests that the VaR model does not correctly capture the risk implied by its assumptions. That practically means that this Var is either too conservative, meaning it predicts more breaches than they actually are, or too aggressive, which means that the model does not cover all the breaches that we see in the market.

5.7.3 Christoffersen independence test

Even if he has ensured, using the Kubic test, that the model has the correct number of breaches, it can still be unreliable if those breaches occur in clusters. In markets, we see that volatility most often occurs in clusters, and models do not adjust quickly enough. That means the VaR models may look fine on average, but they fail during periods of higher risk.



For that issue, we will use Christoffersen's (1998) independence test. This test now focuses more on the timing of the breach. In simple terms, it measures whether the exceptions occur independently of one another using the variable, because under a well-specified VaR model, observing a breach today should not make a breach tomorrow more likely. In statistical terms, the breach should behave like a sequence of Bernoulli trials with a constant probability of success. $1 - \alpha$

In the test, we model $\{I_t\}$ with a first-order Markov structure, which allows the probability of a breach tomorrow to depend only on today's breach probability. This leads to two types of scenarios:

- a) the case of not having a breach first and then having one
- b) the case of not having a breach in 2 consecutive days

If breaches are independent, the probability of a) should be the same as the probability of b). If the test rejects the null of independence, it indicates a cluster (Christoffersen, 1998).

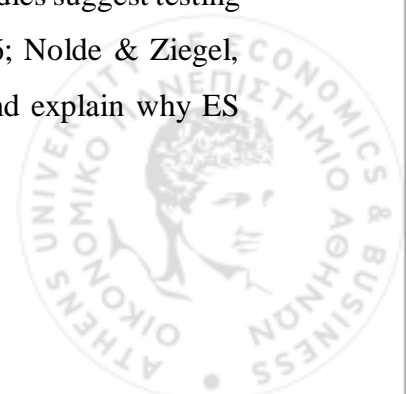
5.7.4 Conditional coverage test

The conditional coverage test combines the unconditional coverage and independence requirements into one joint test (Christoffersen, 1998). In this dissertation, a VaR model is considered empirically stronger if it shows:

- a breach rate close to $1 - \alpha$, and
- higher backtest p-values (i.e., fewer rejections), and
- stable behavior across standard and event windows.

5.7.8 ES backtesting approach

To measure the accuracy of the ES, we did not have a specific breach count like VaR, because ES is not simply a cutoff value like VaR but rather the average loss in the worst tail of the distribution. Backtesting ES by itself is pretty tricky, which is why many studies suggest testing ES jointly with VaR rather than testing ES alone (Fissler & Ziegel, 2016; Nolde & Ziegel, 2017). Some practical ES backtesting ideas also address this difficulty and explain why ES requires different tools than VaR (Acerbi & Székely, 2014).



In this methodology, VaR and ES forecasts are always positive-loss numbers, while the realized return is negative. r_t can be positive or negative. To apply the ES backtest, we can convert the forecasts to return sign by multiplying by -1 (so they are negative numbers, like bad returns):

$$q_t = -VaR_t^\alpha, \quad e_t = -ES_t^\alpha,$$

where q_t is the predicted quantile for the (VaR threshold) and e_t is the predicted ES with the same sign as the return. Let $\tau = 1 - \alpha$ be the tail probability (e.g. $\tau = 0.01$ for $\alpha = 0.99$). Then we define a tail indicator:

$$1(r_t \leq q_t),$$

which equals 1 when the realized return is in the left tail (below the VaR threshold). Using these, we compute a daily calibration statistic (Acerbi & Székely):

$$Z_t = \frac{1}{\tau}(q_t - r_t)1(r_t \leq q_t) - (q_t - e_t)$$

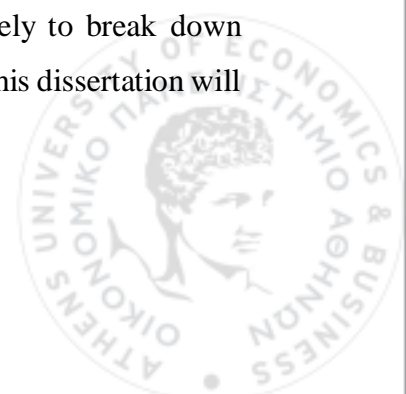
The basic logic is: the first term measures how far the realized return falls below the VaR threshold on tail days, scaled by the probability of tail days. The second term ($q_t - e_t$) measures the difference between VaR and ES. If the VaR–ES forecasts are well calibrated, it means that this statistic should be close to 0, so the test should have:

$$E[Z_t] = 0$$

5.8 Event-window (stress-period) analysis

5.8.1 Motivation

It is well documented that risk models like VaR or ES are most applicable during market stress and economic crises; however, these periods are when they are most likely to break down (Cont, 2001). For that reason, in addition to presenting full-sample results, this dissertation will also evaluate whether VaR/ES behaves differently during volatile periods.



5.8.2 Event window definition

For this methodology, we define several event periods as fixed intervals, such as the COVID crash or the Ukraine shock. For that reason, we label the day using the following method:

- 0, if it falls outside all windows, or
- 1, if it falls within a defined interval.

This approach is similar to what is usually followed in the literature in these types of scenarios, where it is important to analyze shocks also during specific time events rather than their unconditional averages (MacKinlay, 1997). The reason is that VaR and ES models are usually more stable during good economic conditions but become unstable when volatility is high (Cont, 2001).

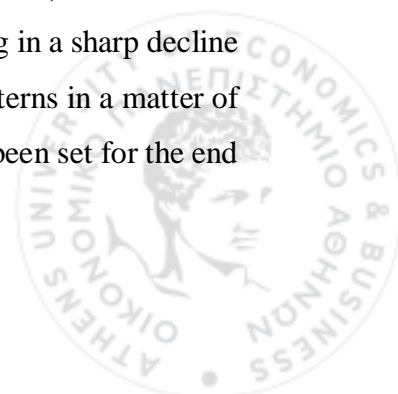
5.8.3 Events discussion

A key component of this dissertation is not just to measure the performance of the VaR and ES models, but also to examine their behavior when it matters most: during periods of economic turbulence in the financial system. For that reason, a list of events has been created in order to focus the analysis on this side. These events have been selected because they represent different types of tail risk, such as a global health shock, a major war, a new policy, or a banking-sector stress event.

Having multiple events is helpful for this analysis because, for example, if a VaR method works well in one event, it may not work well in others, especially when volatility clusters and correlations change significantly across time. In practice, these warnings help indicate whether model performance is stable over time and across scenarios.

Covid Crash (2020-02-20 to 2020-04-30)

We will discuss first the crash during the COVID period. During this period, the measures imposed by governments have created significant market volatility, resulting in a sharp decline in stock prices. At this time, markets move suddenly and in unexpected patterns in a matter of days, making this window important to evaluate. The end of this event has been set for the end



of April to include the aftershock phase, when markets began to absorb the significant policy changes that took place during this period.

Ukraine / Russian War (2022-02-24 to 2022-03-31)

In this event, we try to cover how geopolitical risk, and specifically the biggest war in Europe in the past few decades, affected the market, and specifically the assumptions of parametric models. It is included because it represents a shock that can spread across the whole industry, especially the energy sector and commodities. That makes it helpful to check whether VaR/ES methods respond to volatility jumps.

Q4 2018 Sell off (2018-10-01 to 2018-12-31)

In a slightly different type of event, we also include the Q4 2018 sale. This is a different type of event because it did not happen suddenly, but it was built over weeks and months, and it did not come from a single headline event. It is a period of persistent negative returns and high volatility, which will help us evaluate the models from a slightly different perspective. We need this time period to evaluate whether the VAR/ES metrics behave as assumed during prolonged periods of high risk and volatility.

US Debt ceiling 2023 (2023-05-01 to 2023-06-15)

This event introduces a different kind of uncertainty, coming from policy uncertainty rather than an event. At this time, there was significant discussion about US debt and the possibility of a US default, which affected risk sentiment and investor confidence. This period begins in May and extends till mid-June. We think this type of policy risk can create volatility clusters that could affect VaR model assumptions.

Tariff Shocks 2025 (2025-04-02 to 2025-04-10)

This dissertation will also focus on more recent events that have affected the financial markets



and examine which model performs better on the more recent data. Specifically, in April 2025, there was significant market volatility due to the tariffs the US president imposed on major trading partners. This situation has created high volatility, and the markets have been affected significantly.

War in the Middle East 2025 (2025-06-13 to 2025-06-30)

During the summer of 2025, the situation in the Middle East escalated, driving a period of high volatility amid rising commodity prices. This event has been included because such geopolitical events can stress models in many ways and reveal weaknesses in the assumptions underlying them.

5.8.4 Event-based evaluation outputs

Within each period (Normal vs event windows), the dissertation compares:

- the distribution of $\text{VaR}_{t,\alpha}$ and $\text{ES}_{t,\alpha}$ levels (boxplots),
- breach rates (bar charts),
- backtest statistics where the event sample is large enough

5.9 Implementation details and reproducibility

5.9.1 Step-by-step empirical pipeline

Firstly, we need to get the daily prices for the S&P 500 (from DataStream) and then calculate the daily (log) returns. Next, calculate all the VaR and ES methods using the rolling window method and produce one-day-ahead forecasts of VaR and ES for $\alpha=0.95$ and $\alpha=0.99$. After that, compute VaR breaches for each VaR model, run full-sample VaR and ES backtest, and finally repeat the descriptive evaluation and breach-rate summaries by event window. This design shows how risk models can be used in practice: forecasts are produced using only information available at the time (McNeil, Frey, and Embrechts, 2015).

5.9.2 Software tools

For this analysis, the following tools from Python will be used:



- a) **Pandas** and **NumPy**: these are widespread libraries to work with raw data
- b) **SciPy**: This library will be used for fitting the distributions and calculating the quantiles
- c) **Matplotlib**: This library will be used for calculating the graphs

5.10 Summary

For this dissertation, we are using a framework to compare different approaches to calculating VaR and ES. We specifically predict using Historical Simulation, Gaussian parametric, Student-*t*, and EWMA. VaR and ES are computed at $\alpha = 0.95$ and $\alpha = 0.99$ and a rolling window of 250 days. The VaR models are evaluated and then backtested.

Overall, the methodology is designed to test whether:

- a) Relaxing the standard assumptions via Student-*t* improves the VaR predictions
- b) Adapting volatility with EWMA would improve the adaptation of models and reduce the clustering of breaches, especially in stressed periods where tail forecasting is important (Cont, 2001; Kupiec, 1995; Christoffersen, 1998; Basel Committee on Banking Supervision, 2019).

Chapter6 — Data Description

6.1 Chapter overview

This chapter will describe the dataset. The main focus will be to discuss the basic characteristics of the dataset's normal distribution and to analyze the distribution of predictions from the different VaR models across the two confidence intervals evaluated in this dissertation. Additionally, we will try to visualize the impact of ES by comparing it with VaR and measuring its behavior in tails. Finally, we will demonstrate the returns for the event periods discussed in the dissertation's methodology.



6.2 Describe Out-of-Sample data

This dissertation uses the daily prices from the S&P 500. For that analysis, daily log returns are computed and used to generate one-day-ahead rolling forecasts of VaR and ES. Because we used the first 250 observations to make the first prediction, the evaluation period begins after that.

- The length of the whole series in our sample is 3016 days.
- Out-of-sample evaluation sample for backtesting: $T = 2,766$ observations.

In the following table, we present summary statistics for the out-of-sample daily log returns over the 2,766 trading days of our dataset. The numbers suggest that the returns align with the recent studies in the literature. We can see that the average daily returns are close to 0; they are volatile, negatively skewed, and exhibit significantly strong tails, as indicated by the Kurtosis (Belhachemi, 2024; Ratliff-Crain et al., 2025; Larralde, 2025).

These distributional features motivate tail-focused risk measures, and models will work better, since extreme losses occur more frequently than what normal distribution assumptions suggest (Lazar, Pan, & Wang, 2024).



Figure 6.1 — Distribution of Log Returns (Across the whole sample)

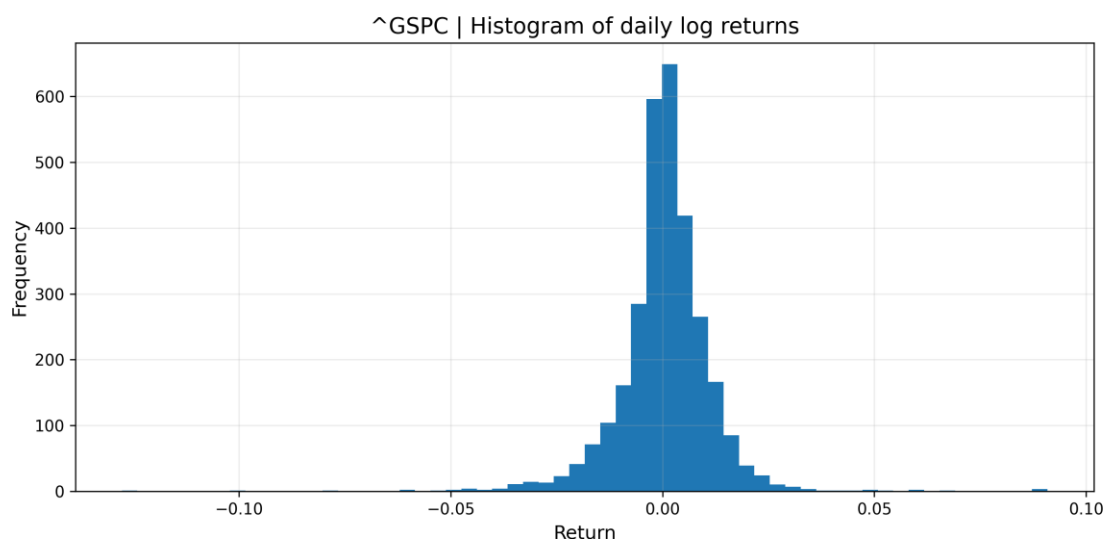


Table 6-1 — Out-of-sample return summary statistics (daily log returns)

Mean	Std	Min	Max	Skewness	Kurtosis
0.04%	1.10%	-12.77%	9.09%	-0.66	19.29

6.3 VaR and ES forecast levels

In this section, we will evaluate the distribution of one-day-ahead VaR and ES forecasts across the models chosen in the methodology. The purpose of this section is to understand how each model's modelling assumption relates to the different risk levels before we compare the test results with the back-test results.

6.3.1 Full-sample forecast distribution (VaR)

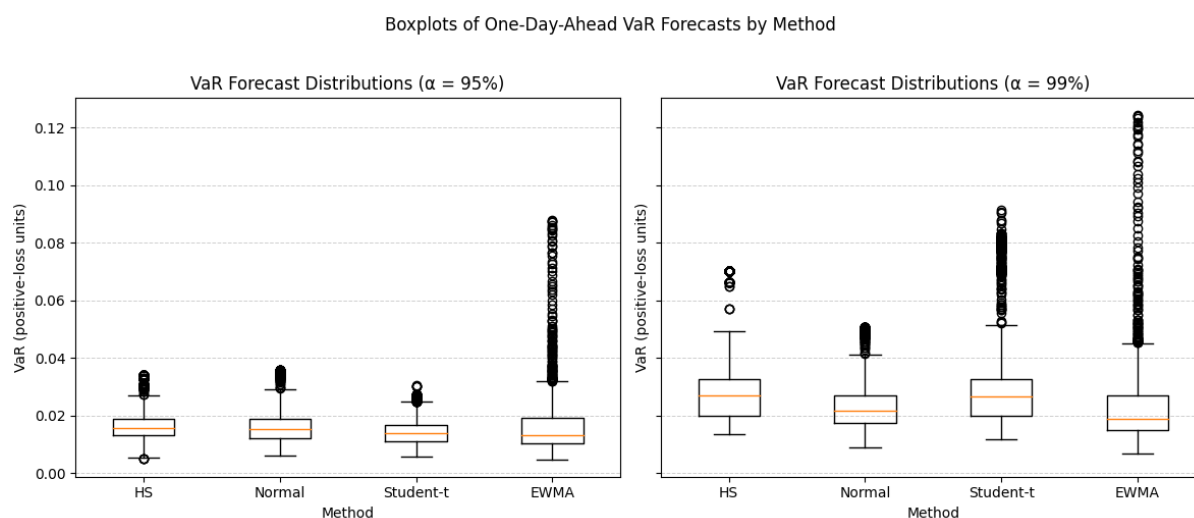
The following graph summarizes VaR forecast distributions at $\alpha = 95\%$ and $\alpha = 99\%$ for the four methodologies discussed earlier. We calculate VaR and rank them, with higher VaR indicating a more conservative forecast.

In the graph, it can be seen that:



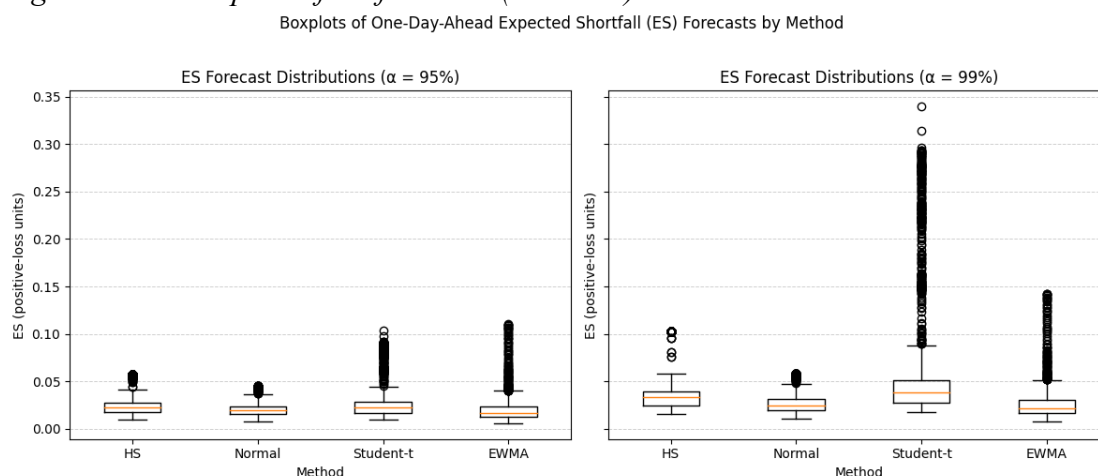
- EWMA VaR typically reacts fastest to volatility changes, and that can be seen from the many extreme values that the model produces.
- Historical Simulation VaR depends on the observations of the previous 250 days, and is strongly affected by the volatility of the past.
- Normal VaR tends to be smoother and can understate extreme tail risk.
- Student-t VaR has fatter tails, which means that it will have more extreme values as α increases, which can also be seen from the graph. There is a big difference in the outliers between $\alpha = 95\%$ and $\alpha = 99\%$

Figure 6.2 — Boxplots of VaR forecasts



6.3.2 Full-sample forecast distribution (ES)

Figure 6.3 — Boxplots of ES forecasts ($\alpha = 95\%$)



In the previous graph, it can be seen that across all methodologies, the model's ES forecast is always higher than the corresponding VaR forecast, as shown in the previous section. This is absolutely expected because, by nature, ES measures the average in the tail of the return's distribution using quantiles defined by VaR. Based on the graph, this result is more evident at $\alpha = 99\%$, where the tail is longer, and extreme losses are more easily seen. Also, from the box plots, we can see an apparent increase in the ES value from $\alpha = 95\%$ to $\alpha = 99\%$, with broader spreads and more extreme outliers, especially for the t-distribution.

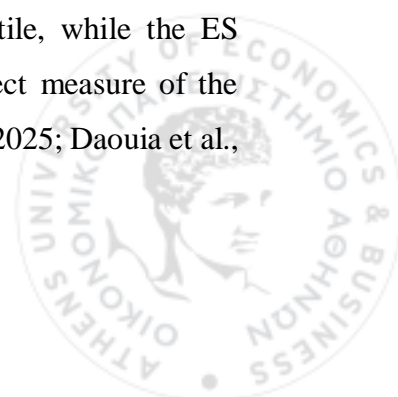
Overall, this figure supports that ES magnifies the model differences when it matters more at $\alpha = 99\%$. This effect would be more pronounced in extreme scenarios, under heavy-tailed distributions (e.g., Student-t), and when volatility clusters (e.g., EWMA).

6.3.3 VaR–ES gap behaviour

In this section, we will focus on the difference between VaR and ES. We define the difference as:

$$Gap_{t,\alpha} = ES_{t,\alpha} - VaR_{t,\alpha}.$$

Because, as discussed in the previous chapter, the VaR is just a quantile, while the ES represents the average loss beyond this quantile, the gap provides a direct measure of the importance of losses once the VaR threshold is breached (García-Risueño, 2025; Daouia et al.,



2025). Here, a significant gap means that VaR is not just more likely to occur, but, more importantly, the expected tail loss is likely to be much worse than VaR itself.

The following figure illustrates the gap using ECDF curves, making it easy to compare both distributions and, most importantly, the behavior of the right tails. We see that at 95%, all models show steep ECDFs, which are concentrated and have small gaps, indicating that ES typically does not exceed VaR by much during normal market conditions, which is similar to the results from Lazar in 2024 and García-Risueño in 2025.

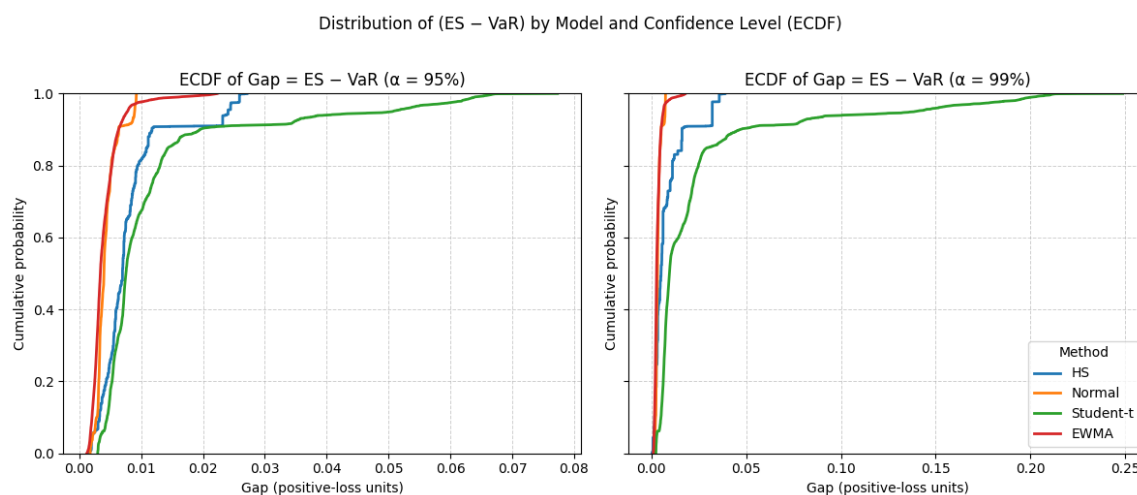
However, at the 1% confidence level, the distance between the curves is significant, indicating that the gap widens further as the α level increases. An example from the graph is the Student-t type VaR, which shows a slower rise in the ECDF but significantly longer tails, consistent with the model's assumptions. By contrast, the model which assumes normality in its parameters remains concentrated around small gaps even at $\alpha=99\%$. Historical Simulation lies between these two cases because it is wider than the model that uses parameters from a normal distribution. However, it does not generate the same heavy tails as the model that uses the t-Distribution.

Finally, EWMA produces relatively small gaps between VaR and ES, consistent with volatility scaling, which tends to raise both VaR and ES jointly. That means absolute risk levels will likely rise in crises, but the difference between ES and VaR will remain relatively small.

Overall, the graph suggests the $ES_{t,\alpha} - VaR_{t,\alpha}$. The gap tends to widen at $\alpha = 99\%$, providing an additional tool for understanding how VaR and ES behave by summarizing not just whether a tail event occurs, but also how severe the damage is expected to be once it does (Lazar et al., 2024; Daouia et al., 2025).

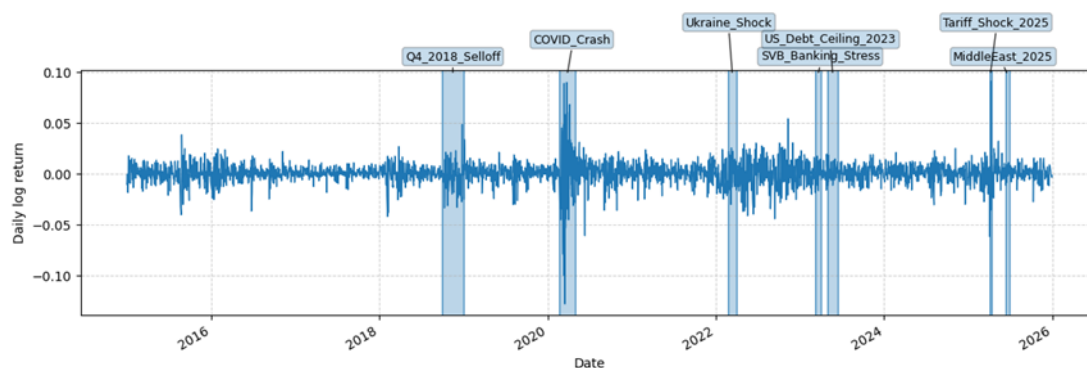


Figure 6.4 — Empirical Cumulative Distribution Function of (ES – VaR)



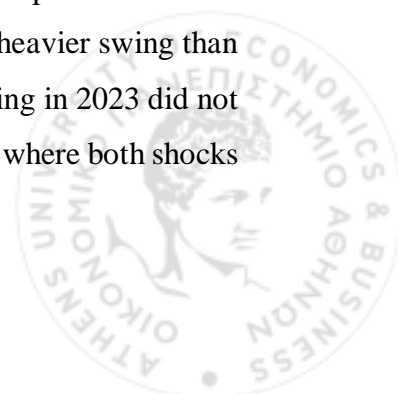
6.4 Event Analysis

Figure 6.5 – Event analysis



The previous graph illustrates the events this dissertation will discuss. From the chart, we can see that the daily log returns are mainly concentrated around zero, but the series includes several extreme periods. We can see that these periods are not spread uniformly across time. However, in many cases, extreme values are clustered, as seen during the COVID period, when the numbers of both positive and negative returns increased significantly. It is also the period with the most significant single-day loss over the past 10+ years.

A smaller but still noticeable increase in the number of shocks in a single period occurred during the 2018 sell-off and the war in Ukraine, when the returns exhibit a heavier swing than in the periods around the. On the other side, events such as the US debt ceiling in 2023 did not have a significant effect on returns. Similar results can also be seen in 2025, where both shocks



from Tariffs and the war did not significantly stress the markets, and their returns did not differ significantly from those in other periods.

Chapter7 — Empirical Results

7.1 Chapter overview

This chapter will focus on the results from the methodology presented in the previous chapter. The main goal of this dissertation is to analytically examine the VaR and EL models this dissertation focuses on, compare them in out-of-sample settings, and explore their specific characteristics. The models discussed in this dissertation are Historical Simulation (HS), Normal parametric VaR/ES, Student-t parametric VaR/ES, and EWMA-based VaR/ES. Forecasts are produced at two confidence levels, $\alpha = 95\%$ and $\alpha = 99\%$, using one-day-ahead rolling forecasts with a 250-trading-day estimation window.

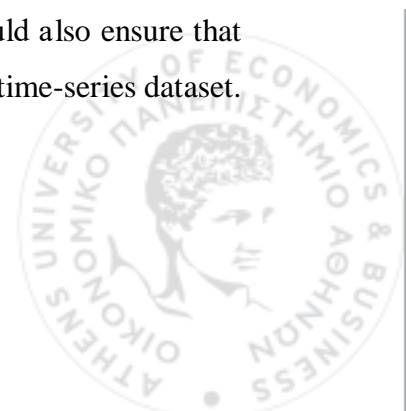
Specifically, the focus will be on producing and interpreting backtesting results for VaR and ES by counting the number of times each model breaches its threshold and performing tests such as the Kupiec unconditional coverage test, the Christoffersen independence test, and the conditional coverage test. Also, it will provide an event-window analysis to compare the different models under extreme scenarios that have significantly affected markets in the past few years.

7.2 VaR backtesting results

In this section, we will evaluate VaR accuracy using the backtesting methodology described in the methodology. The breach is defined as:

$$I_t = \mathbf{1}(r_t < -VaR_{t,\alpha}),$$

where r_t is the realised return. If we assume the models have been accurate, the breach should be expected to be: $(1-\alpha) = 5\%$ for $\alpha = 95\%$ and 1% for $\alpha = 99\%$. We should also ensure that breaches do not cluster over time or occur at random points throughout our time-series dataset.



7.2.1 Breach counts and breach rates

At $\alpha = 95\%$, expected breaches are:

$$(1 - 0.95) \times 2766 \approx 138$$

The following table shows the actual times that each model breached their threshold. From there, we can see the range of breaches is 5.46% to 6.51%, all above 5% but close to it. The most accurate model is HS with a 5.46% breach rate, and the least accurate, with a substantial difference from the target, is the Student t distribution. The normal VaR has 157 breaches (5.68%), which is higher than the expected rate, while the Student-t VaR has 180 breaches (6.51%), the highest breach rate and the one farthest from the actual value. Based on these results, we can see that all the Var models are slightly more conservative for this specific period, but they are all close to the value, except for the VaR model using the t-Student distribution.

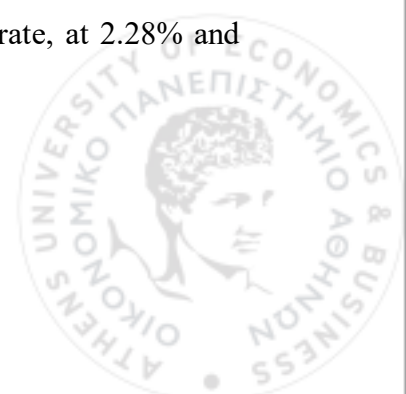
Table 7-1 — VaR breaches ($\alpha = 95\%$): counts and rates

Model	T	Expected	Breaches	Breach rate
EWMA	2,766	138	152	5.50%
HS	2,766	138	151	5.46%
Normal	2,766	138	157	5.68%
Student-t	2,766	138	180	6.51%

At $\alpha = 99\%$, expected breaches are:

$$(1 - 0.99) \times 2766 \approx 28.$$

In the following table, we present the realised breach rate at the 1% confidence level, recheck the number of breaches, and compare them with the expected values. In this case, we can see that the models closest to the target are the Vars that use the Student-t distribution and the historical distribution, with 48 breaches at a 1.74% breach rate. The other two models, EWMA and Normal, suggest that the breach rate is much higher than the actual rate, at 2.28% and 1.74%, respectively. These results suggest two conclusions:



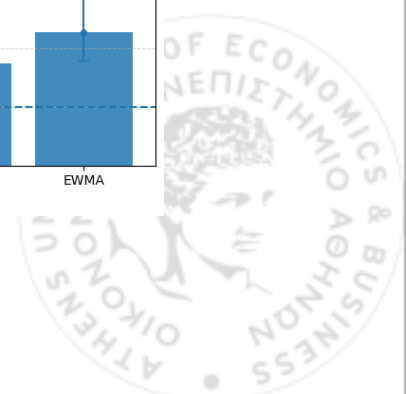
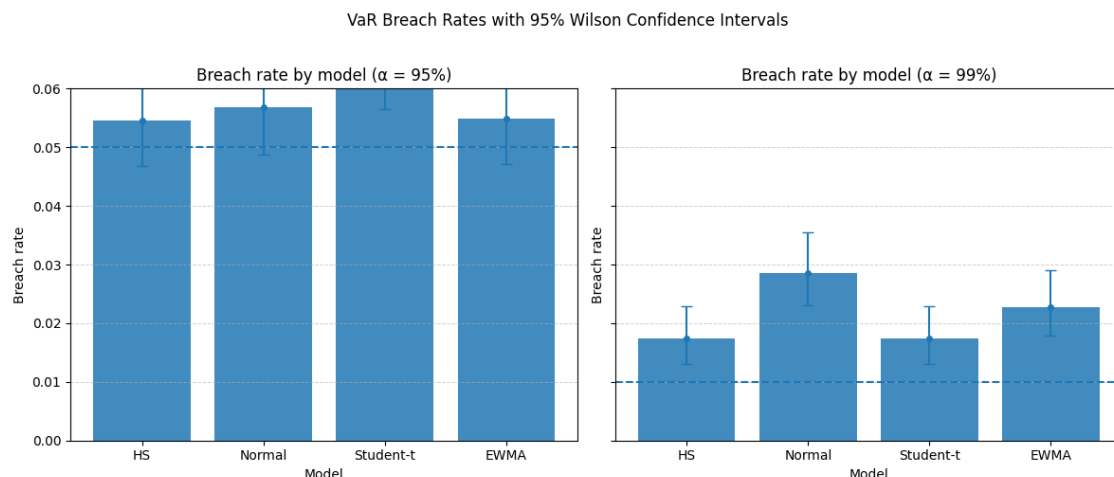
- 1) The 99% tail seems to be really challenging to capture with just one of these models, and models that have been created to be more conservative, like Student-t assumptions, fail to predict the correct number of breaches each time.
- 2) If we rank across the models, it is clear that the assumptions from the normal distribution fail to capture the number of breaches completely, and this is a result that we expected and agree with the literature, like the research of Morkūnaitė, Celov, and Leipus in 2024. The reason is that the normal distribution assumes thin tails, which do not align with the actual returns we observe.

Table 7.2 — VaR breaches ($\alpha = 99\%$): counts and rates

Model	T	Expected	Breaches	Breach rate
EWMA	2,766	28	63	2.28%
HS	2,766	28	48	1.74%
Normal	2,766	28	79	2.86%
Student-t	2,766	28	48	1.74%

To better see the difference between 1% and 5%, we will plot the VaR breaches for these two confidence intervals. The graph basically shows, in an easy-to-compare way, the results from the previous tables, indicating that all the models had pretty much the same performance at the 95% confidence interval. However, at 99%, all the rates are significantly higher, with the Normal distribution having the highest breach rate.

Figure 7.1 — VaR breach rate



Overall, breach rates across models are slightly higher than the target at $\alpha = 95\%$, with the highest breach rate coming from the model using the t-distribution. At $\alpha = 99\%$, all models are significantly above the threshold.

7.2.2 Kupiec unconditional coverage test

To formalise the previous indications from the graphs, we provide a test to compare the performance of the different models in terms of calibration at 5% and 1% confidence intervals.

At an $\alpha = 95\%$, the EWMA, HS, and Normal models remain statistically close to our target expectation of a 5% breach rate, as indicated by the p-values. Specifically, for these models, the p-value is significantly lower than the 5% benchmark, indicating that the null hypothesis can be rejected and that these models could, on average, serve as viable VaR options. However, if we consider the VaR based on the Student-t distribution, it firmly rejects UC ($LRuc = 12.14L$, $p < 0.001$), consistent with its materially higher breach rate (6.51%). In practical terms, the result suggests that the Student t-distribution is not well calibrated in the full sample used for this analysis. That result agrees with a similar result from Cheng in 2025.

Table 7.3 — Kupiec UC test ($\alpha = 95\%$)

Model	LRuc	p-value	Decision (5%)
EWMA	1.39	0.24	Do not reject
HS	1.19	0.28	Do not reject
Normal	2.56	0.11	Do not reject
Student-t	12.14	<0.001	Reject

On the other hand, at $\alpha = 99\%$, the picture is entirely different. Here, the null hypothesis can be rejected across all the models. LRuc across models is relatively high, and that makes the test sure that these models cannot capture the risk at the 1% tail. From the table, we can also see that the normal distribution has the highest score, which aligns with the higher breach rate and, once more, suggests that its assumptions cannot capture real-world scenarios. Overall, when $\alpha = 99\%$, results suggest a common problem: even when models work well in lower confidence levels, the extreme tail calibration is way more complex, and that is also baked into other recent empirical suggestions like the one from Goel, Pasricha, and Kannianen 2025

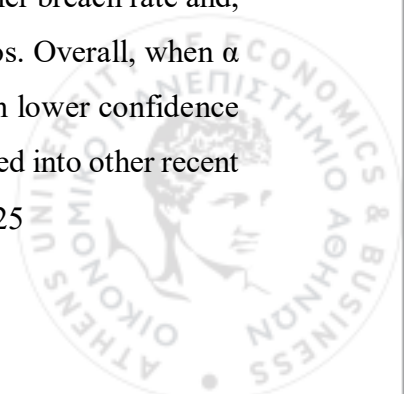


Table 7.4 — Kupiec UC test ($\alpha = 99\%$)

Model	LRuc	p-value	Decision (5%)
EWMA	33.49	<0.001	Reject
HS	12.39	<0.001	Reject
Normal	64.10	<0.001	Reject
Student-t	12.39	<0.001	Reject

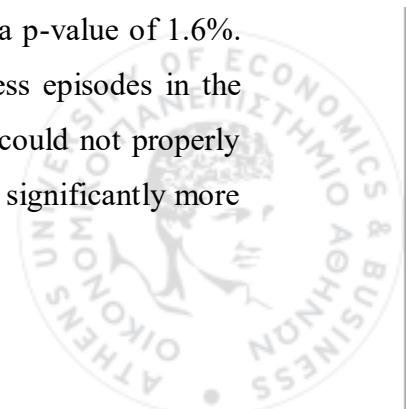
7.2.3 Christoffersen independence test

The Christoffersen is a powerful tool in our analysis, because it adds what is missing from the Kupiec test, which is the timing of the breaches, and shows which model behaves better when the clusters are happening in specific periods. In our dataset, the results are strong and indicate which model better handles the clustering issue at $\alpha = 95\%$. However, in the $\alpha = 99\%$ confidence interval, all models reject the null hypothesis at 5% confidence level.

At $\alpha = 95\%$, the EWMA is the only model that does not reject the null hypothesis. It has a Lrind of 1.59, which gives a p-value of 20.7%. This suggests that this model also tracks well the changing trend in volatility conditions and can better adjust in terms of timing compared to the other models. This is consistent with the intuition that models that are built to adapt to volatility can significantly reduce breach clustering and respond more quickly when variance shifts (Likitratcharoen, 2024).

Contrary to HS, Normal, and Student-t, all strongly reject the independence at the 5% level ($p < 0.001$ for all). This indicates that their breaches are not randomly distributed over time, but instead occur in bursts. That means that even if the models' breach rate is close to the target breach rate, a sudden jump in volatility would still indicate that the models' risk forecasting is not adjusting to the event. This type of clustering, as it has been referred to in the previous chapters, is a well-known empirical feature and points out the emphasis on more diagnostic patterns that focus on the clusters (Balter & Ziegel, 2024)

At $\alpha = 99\%$, all models reject independence, including the EWMA, with a p-value of 1.6%. These results make sense because at 1% tail, breaches mean intense stress episodes in the markets, and market conditions change drastically. The fact that EWMA could not properly adapt to the clusters at 99% suggests that capturing the most extreme tail is significantly more



difficult than the 95%. For the other models, the LRind value suggests that, under their assumptions, capturing the far tail of returns is a difficult challenge.

Table 7.5 — Christoffersen independence ($\alpha = 95\%$)

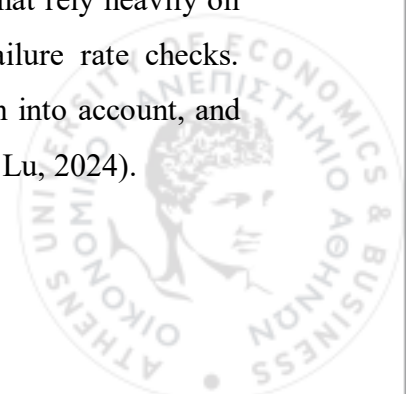
Model	LRind	p-value	Decision (5%)
EWMA	1.59	0.207	Do not reject
HS	26.16	<0.001	Reject
Normal	23.12	<0.001	Reject
Student-t	26.51	<0.001	Reject

Table 7.6 — Christoffersen independence ($\alpha = 99\%$)

Model	LRind	p-value	Decision (5%)
EWMA	5.78	0.016	Reject
HS	19.16	<0.001	Reject
Normal	15.92	<0.001	Reject
Student-t	14.54	<0.001	Reject

7.2.4 Conditional coverage test

The conditional coverage test combines the Kopiec UC and the independence test into a joint assessment. Based on the following tables, we can draw overall conclusions for each VaR model. EWMA is the only model that does not reject the null hypothesis with an LRcc = 2.98, yielding a p-value of 22.6%. In contrast, all the other models strongly reject the null hypothesis at any confidence level. This finding agrees with the results from the Christoffersen and Kupiec tests, which means, once more, that his model is the only methodology that can jointly satisfy the two criteria in the full sample. On the other hand, models like HS or Normal do not reject the null at the 95% level. The rejection suggests that the model's calibration does not hold over time. This is in line with recent empirical evidence showing that models that rely heavily on assumptions from a static distribution may appear acceptable under failure rate checks. However, they will massively fail once the timing of the breaches is taken into account, and that phenomenon is more present during volatility clusters (Hué, Hurlin, & Lu, 2024).



At $\alpha = 99\%$, the results become clear and the same across all models. Here, we reject the null hypothesis across all the VaR methodologies. That implies that, in the extreme tails, none of the four models was sufficient to reproduce the exception that would cover the joint coverage test. The magnitude of the test statistic also provides a ranking that helps us better interpret the results. From that core, we can see that the worst, by far, with more than double the score of any other model, is the VaR that uses Normal distribution parameters. In contrast, the best model is the VaR based on the Student-t distribution, as it naturally has fat tails that better capture extreme scenarios. However, none of these models capture these scenarios well enough.

Table 7.7 — Conditional coverage ($\alpha = 95\%$)

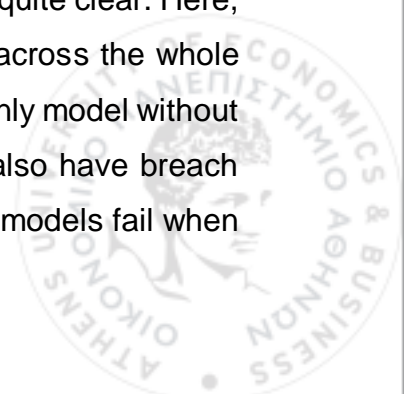
Model	LRcc	p-value	Decision (5%)
EWMA	2.98	0.226	Do not reject
HS	27.35	<0.001	Reject
Normal	25.68	<0.001	Reject
Student-t	38.64	<0.001	Reject

Table 7.8 — Conditional coverage ($\alpha = 99\%$)

Model	LRcc	p-value	Decision (5%)
EWMA	39.27	<0.001	Reject
HS	31.55	<0.001	Reject
Normal	80.03	<0.001	Reject
Student-t	26.93	<0.001	Reject

7.2.5 Ranking summary

The following table summarizes the results from the previous tests and ranks the different models. At $\alpha = 95\%$, the overall ranking across the model is quite clear. Here, the best model by far across all tests is the EWMA, which passes across the whole test. It has a nominal breach rate close to expectations, and it is the only model without significant clustering of expectations at this level. HS and Normal also have breach rates close to the target; that is why they pass the UC test, but both models fail when

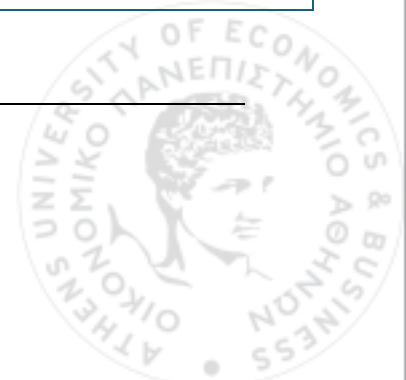


backtesting focuses on independence and conditional coverage. Finally, the Student-t model performs worst: at 95% decay, it fails across all tests, produces too many breaches relative to the actual, and shows strong evidence of clustering issues, leading to clear rejection for this type of analysis. To summarise, at 95%, the results suggest that an EWMA model is the best approach for relatively stable day-to-day VaR performance.

At $\alpha = 99\%$, the results change significantly. When we talk about the extreme tail, no model performs satisfactorily on the full sample. All models reject the null hypothesis across all tests. That means the far-tail violations are too frequent relative to the model's predictions and are concentrated in crisis periods rather than arriving evenly over time. While the breach rate across the models ranges from 1.74% to 2.86%, the conclusion is that none of the selected models is properly calibrated and provides reliable calibration at the 99% level.

Table 7.9 — Summary Table (VaR)

α	Model	Breach rate	UC (Kupiec)	Independence	Conditional coverage	Overall comment
95%	EWMA	5.50%	Pass	Pass	Pass	Best overall
95%	HS	5.46%	Pass	Fail	Fail	Clustering dominates
95%	Normal	5.68%	Pass	Fail	Fail	Clustering dominates
95%	Student-t	6.51%	Fail	Fail	Fail	Too many breaches + clustering
99%	All	1.74%–2.86%	Fail	Fail	Fail	Extreme tail not captured



7.3 Expected Shortfall evaluation

ES is more difficult to backtest than VaR because there is no simple breach percentage to calculate like VaR for ES alone. ES is therefore evaluated using a joint VaR–ES calibration statistic, Z_t , which is defined in the methodology chapter. Under correct calibration of the VaR–ES pair, the key condition is:

$$E[Z_t] = 0.$$

7.3.1 Full-sample ES calibration results

The following two tables summarise the results from the ES calibration, using the sample mean of Z_t . Moreover, a t-test of whether this mean is different from zero. At $\alpha = 95\%$, HS and Student-t have means closer to 0, and the test also suggests that their VaR-ES calibration is correctly calibrated in the full sample at the 5% confidence level. On the other hand, both EWMA and Normal distributions yield a very small p-value, so we can conclude that their means are significantly greater than zero.

At $\alpha = 99\%$, the calibration problem is larger for some models, such as the EWMA and Normal, which both have statistically significant non-zero means, and the t-statistics are larger than at $\alpha = 95\%$. However, the Student-t model performs better at this level, with a p-value of 73%, and, together with the VaR from HS, it better captures the EL dynamics in the models.

Overall, the two tables suggest that ES calibration is more challenging at $\alpha = 99\%$, as expected, because losses are rare but significant there, and the best model to capture these losses is VaR, which accounts for fatter tails.

Table 7.10 — ES calibration summary ($\alpha = 95\%$)

Model	T	Mean Z_t	t-stat	p-value	Interpretation
EWMA	2,766	0.0046	3.33	<0.001	Mean differs from 0
HS	2,766	0.0029	1.10	0.273	Not significantly different
Normal	2,766	0.0070	2.73	0.006	Mean differs from 0
Student-t	2,766	0.0016	0.52	0.603	Not significantly different

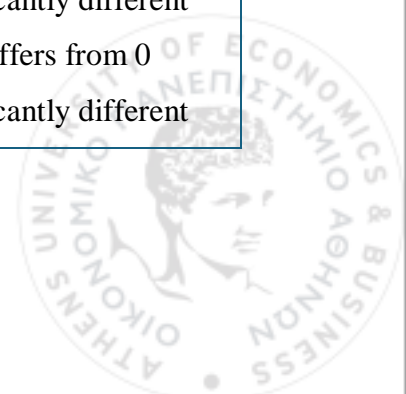


Table 7.11 — ES calibration summary ($\alpha = 99\%$)

Model	T	Mean Z_t	t-stat	p-value	Interpretation
EWMA	2,766	0.0173	4.00	<0.001	Mean differs from 0
HS	2,766	0.0113	1.50	0.133	Not significantly different
Normal	2,766	0.0278	2.74	0.006	Mean differs from 0
Student-t	2,766	-0.0030	-0.34	0.733	Not significantly different

7.4 Event-window results

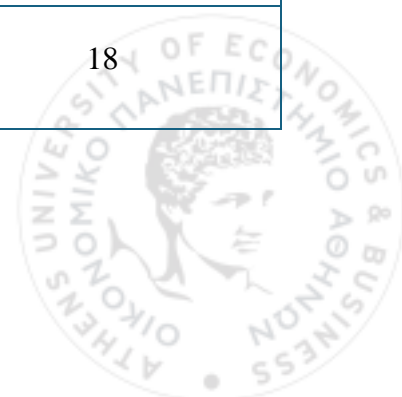
This section shows how various VaR models behave during predefined important events in the stock market. The key objective is to assess whether model performance differs significantly in crises compared to normal conditions, as tail risk estimation is most important during such episodes.

7.4.1 Event windows included

For this analysis, we include the following events, as discussed in the methodology. We are trying to cover the event period and the period immediately after to observe how markets and returns adjust during and after the event.

Table 7.12 — Event windows and number of observations

Period	Start	End	Observations (T)
Q4 2018 Sell-off	01/10/2018	31/12/2018	63
COVID Crash	20/02/2020	30/04/2020	50
Ukraine Shock	24/02/2022	31/03/2022	26
SVB Banking Stress	08/03/2023	31/03/2023	18



US Debt Ceiling	01/05/2023	15/06/2023	33
Tariff Shock	02/04/2025	10/04/2025	7
Middle East Tensions	13/06/2025	30/06/2025	11

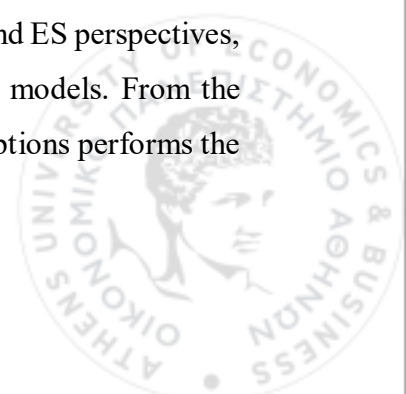
7.4.2 Forecast level behaviour during stress

During periods of stress, VaR and ES levels rise sharply. EWMA typically responds quickly to volatility spikes, whereas HS may show lag due to equal weighting within the historical window. Parametric Normal forecasts can understate tail outcomes in crisis regimes, while Student-t is designed to allow fatter tails, which may increase forecasted tail losses.

At $\alpha = 95\%$, the number of breaches depends on the events; for some events, we observe multiple breaches, as discussed in the previous section and also in the literature. The best model that fit them as expected was EWMA, because it adjusted to volatility more quickly. In some cases, such as the Middle East war, the movement in the returns was not particularly significant, and there was no breach at all.

For $\alpha = 99\%$, we observe that during periods of extreme volatility, such as the COVID crash, the EWMA adjusted much faster to volatility than the other models, both during the crash and after it. Also, the main model generally worked well across all events for both VaR and ES. The Student-t VaR model seems to perform at the same level as the normal distribution and HS, but it is much more conservative regarding the loss rate. Here, the values become extreme as volatility increases, as seen during COVID, when the loss rate reached almost 20%.

Overall, EWMA seems to be the most stable model, better adjusting to risk scenarios and responding faster to market conditions. It works pretty well from both VaR and ES perspectives, and its dynamic volatility addresses the main issue with other parametric models. From the graphs, we can once again conclude that the VaR based on Gaussian assumptions performs the



worst, a result that is not unexpected, as we expected a volatility cluster for this type of event, as discussed in the previous section.

Figure 7.2 — Events zoom-in analysis ($\alpha = 95\%$)

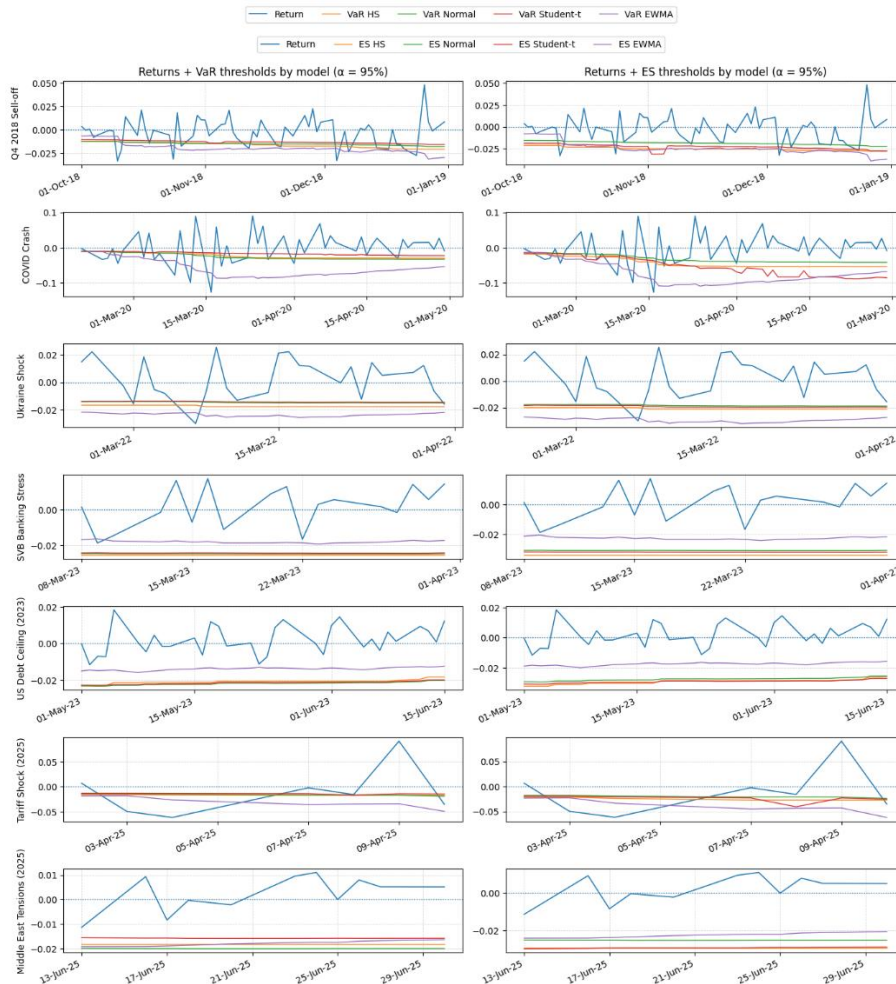
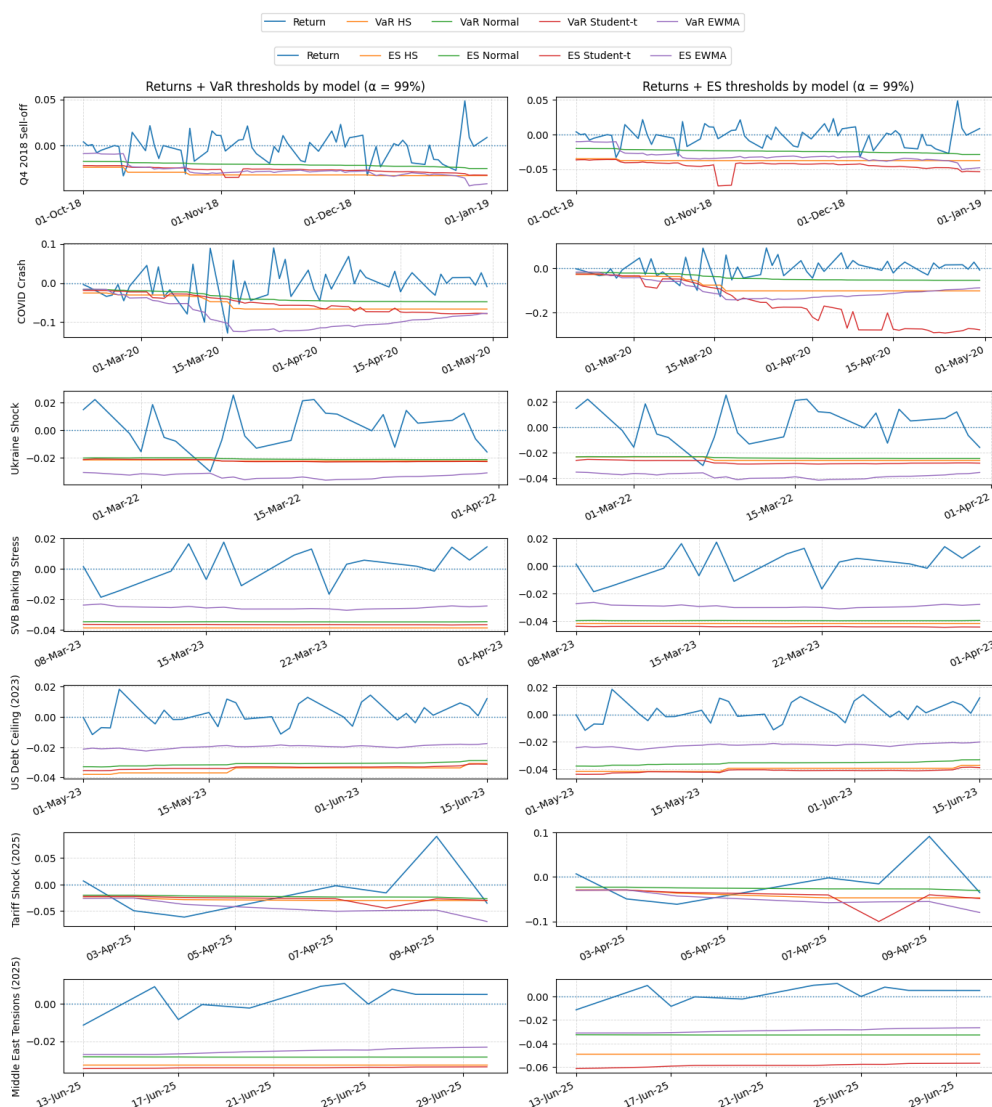
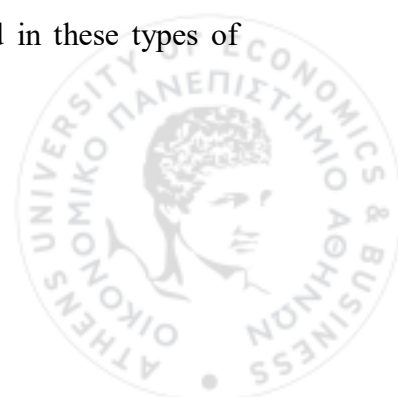


Figure 7.3 — Events zoom-in analysis ($\alpha = 99\%$)



7.4.3 The best model during the backtest period

At $\alpha=95\%$, EWMA is the best-performing model, with an accurate breach rate of 5.5%, and none of the tests in the dissertation rejected the null hypothesis, making it the most reliable across all conditions. HS and Normal, even if they have breach rates close to actual ones, they fail to account for independence, while Subtend fails to pass all the tests. At 99%, no model can pass any test, suggesting that other methodologies should be applied in these types of scenarios.



7.4.4 The best model during the event analysis

For the event-based analysis, the results suggest that calibration is more challenging during market distress, particularly during COVID. However, we can easily see that the model that best adjusts to the new conditions is still the EWMA, making it more robust in these scenarios. The results show extreme deterioration in calibration during crises, particularly during COVID. EWMA tends to be relatively more robust in stress (lower breach rates than alternatives at $\alpha=95\%$), consistent with its volatility. However, at $\alpha=99\%$, all models fail during crisis windows, suggesting that modelling rare-tail outcomes remains highly challenging.

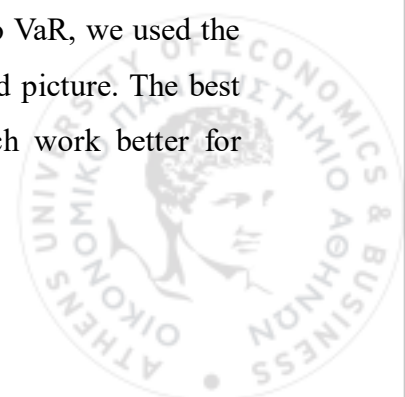
7.5 Summary

This chapter presents the complete set of empirical results for the four VAR/ES forecasting approaches that were examined in this dissertation. Historical Simulation, parametric Normal, parametric Student-t, and EWMA, using one-day ahead forecasts with a 250-day estimation window at confidence levels $\alpha = 95\%$ and $\alpha = 99\%$. In this analysis, we have combined a breach-rate analysis with a proper backtest for both VaR and ES.

For VaR at $\alpha = 95\%$, all models produce rates close to the theoretical rate. However, as it has noticed from the Christoffersen test, the timing of the breaches was wrong. EWMA is the only model that performs consistently well across all the backtests that we used in the analysis. For the remaining models, such as HS and Normal, we can accept them only if the focus is on the acceptance rate. Finally, the Student-t VaR is the worst-performing model at the 95% confidence level, as it fails all backtesting tests.

At $\alpha = 99\%$, the main conclusion of the results is that none of the four VaR models can capture the fat tails of the returns distribution. All models have significantly higher breach rates than the expected 1, ranging from 1.74 to 2.86%, and all fail in all backtesting tests.

For the ES, because simple backtesting tests could not be applied easily to VaR, we used the VaR–ES calibration statistic Z test. The results here showed a more mixed picture. The best model here was the Student-t; it has assumptions for fatter tails, which work better for estimating the loss rate.



Finally, when analysing the event window, we reconfirmed the results we reported for the whole sample. The models' weaknesses become more apparent during periods of stress, when tail risk estimates are most needed. Across large spikes, such as those from COVID and other shocks, EWMA seems to be the clear winner, quickly adjusting to volatility and capturing it more effectively. At the same time, the other models lag due to their strict assumptions and parametric inputs.

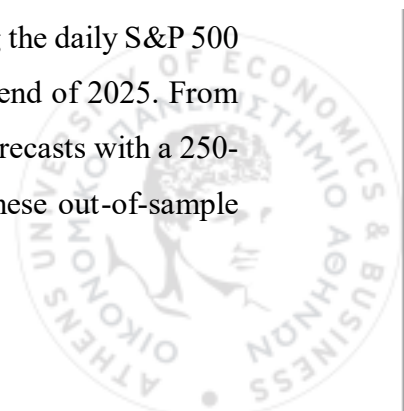
Overall, the evidence supports the idea that, to rank these models, we need to consider the confidence level and the risk measure we want to focus on. For VaR calculation, EWMA appears to be the most robust, especially at the 95% confidence level. However, for $\alpha = 99\%$, no model works well enough under these scenarios. Regarding ES, the best model for capturing fat-tail losses is the Student-t distribution because it assumes fatter tails, which can better capture the actual risk in markets.

Chapter 8 - Conclusion

The primary focus of the dissertation is to evaluate and measure how well different risk models forecast one-day-ahead downside risk for an equity benchmark, specifically the S&P 500. The main risk metrics this thesis will work with are Value-at-Risk and Expected Shortfall. This motivation is coming from sources. Firstly, equity returns are well known to have fat tails and their volatility clusters, meaning that simple assumptions for parametric distributions can underestimate the risk exactly when risk management matters most, during economic distress. Secondly, regulation and current frameworks treat VaR and ES jointly, meaning that a proper risk measure must not only account for the frequency and timing of VaR, but also whether the VaR-ES pair is correctly calibrated in the tail. The overall objective was therefore to compare alternative VaR/ES models and identify which methods are more robust across normal economic times and stressed scenarios.

8.1 Building the dataset

In the dissertation, a rolling out-of-sample framework was constructed using the daily S&P 500 prices from Refinitiv DataStream, and covers the period from 2015 to the end of 2025. From these prices, daily log returns were computed and used to obtain one-day forecasts with a 250-day rolling window, yielding an evaluation over 2766 observations. For these out-of-sample



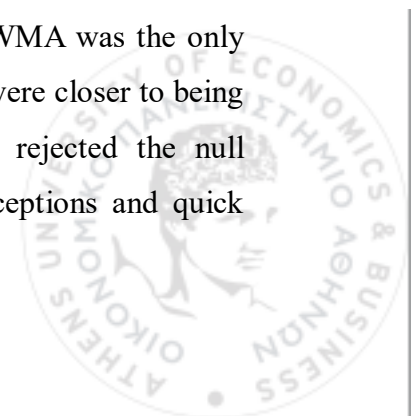
observations, risk for cases is calculated for two confidence intervals, $\alpha = 95\%$ and $\alpha = 99\%$. Four models were calibrated and compared: Historical Simulation (HS) as a non-parametric benchmark; the Gaussian parametric VaR/ES as a classical simple baseline; the Student-t parametric VaR/ES to allow heavier tails; and the EWMA-based VaR/ES to capture time-varying volatility and volatility clustering. These models were chosen because they try to correct different problems that the basic VaR/ES model has.

8.2 Data Description

After examining the returns of S&P over the past decade, we can see that over these 2766 days, the returns had a near-zero mean, high volatility, negative skewness, and significant Kurtosis with heavy tails. These results provide a reason why some pure Gaussian frameworks may fail in the extreme scenario and why potential models that account for the volatility changes may be more accurate. This dissertation formally examines the backtests of the different models. It provides a descriptive analysis that shows the distribution of VaR/ES models differs significantly across the selected models. For example, EWMA tends to react more to volatility spikes, as shown by the wider box plot and the more extreme values it exhibits. Also, we can see that Student-t has heavier tails than the simple normal distribution, especially at $\alpha = 99\%$. The Var-ES gap analysis also demonstrated differences across these models, as this graph makes it more evident that $\alpha = 99\%$ Es can push the model divergence further into the tails.

8.3 Backtesting Results

The main findings focus on the backtesting of VaR and ES models in the full sample periods. There, we can see that at $\alpha = 95\%$, the expected number of breaches is approximately 138. The empirical breach rates for EWMA, HS, and the normal distribution were quite close; for the Student-t distribution, they were slightly higher. When we formalised using the Kupiec coverage test, EWMA, HS, and Gaussian did not reject the null of correct coverage at the 5% significance level. In contrast, the Student-t test was rejected, indicating it was not calibrated correctly. However, the results were not as positive when the timing of the breaches was also taken into account. The Christoffersen independence test showed that EWMA was the only model that did not reject the null hypothesis, meaning that its excursions were closer to being randomly distributed throughout time. HS, normal, and student-t all rejected the null hypothesis, which means that they could handle correctly clustered exceptions and quick



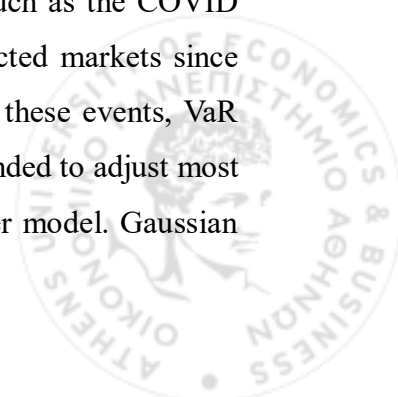
changes in the volatility. The combined coverage test showed that at $\alpha = 95\%$, EWMA was the only model that did not reject the joint requirement of correct frequency and independence. All together, these results suggest that, in practice, most VaR validation models will be acceptable when only the breach count matters, but robustness requires accounting for breach timing. In this area, only the EWMA model, which accounts for validity changes, has a clear advantage.

At $\alpha = 99\%$, the findings were the same across all the models. The expected number of breaches for this period should be 28; however, all the models produced a significantly higher breach rate, indicating that the far-left tails were not captured properly. The Kupiec test rejected the null hypothesis for all the models, and the Christoffersen test with the conditional coverage test also had the same results. Examine these results further, and we could see that Student-t and HS were relatively less poor but still failed. The worst-performing model was the normal distribution, which is consistent with its thin-tail assumptions. From those results, we can see that none of the models captured these extremely rare events, because such events are rare and tend to occur in bursts during a crisis. Therefore, although EWMA was robust at $\alpha = 95\%$, none of the selected models delivered acceptable results.

The second set of findings is related to the ES evaluation, where the dissertations have used the joint-Var ES calibration statistic to measure the breaching count. At a 95% confidence level, HS and Student-t produced calibrations closer to zero and showed no significant deviation from zero; EWMA and Gaussian showed significant deviations from zero. At $\alpha = 99\%$, the Student-t remains the strongest performer in terms of ES calibration, something that is consistent with the intuition that heavier tails usually mean better tail calibration. These results suggest an important result. The best model can depend on the metric we evaluate each time. EWMA performed best for VaR timing and joint Var test at $\alpha = 95\%$, but Student-t performed better for capturing ES tail stability and VaR-ES calibration, especially at $\alpha = 99\%$.

8.4 Event Period Analysis

The event window analysis provided additional evidence that model performance is highly affected by timing. The dissertation examines various stressed periods, such as the COVID crash, the Russia-Ukraine war, and other important events that have affected markets since 2014, and compares the examined models across these scenarios. Across these events, VaR forecasts rose significantly, yet each model responded similarly. EWMA tended to adjust most quickly during volatility spikes, while HS often lags because of its simpler model. Gaussian



forecasts were the most vulnerable during stress, because they understate the tail risk. As expected, we observed that not all events had the same level of stress; some showed extreme volatility and clustered breaches, while others had a limited impact. Overall, the event analysis suggests that model weeks are more visible during a crisis, and that the full-sample analysis can sometimes hide performance issues in the models.

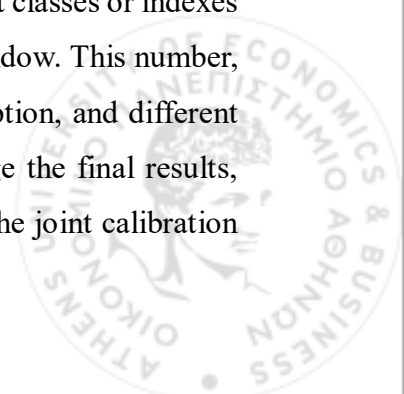
8.5 Hypotheses Evaluation

The empirical results provide clear answers to the hypotheses we set in Chapter 4.

- H1. In our sample period, EWMA is the most robust VaR model at $\alpha = 95\%$, because it is the approach that rejects none of the selected tests, while all the other selected models fail to reject tests related to the timing of breaches. So, for $\alpha = 95\%$, the selected models are indeed the most robust. On the other hand, as described before, at $\alpha = 99\%$, no model was robust enough to withstand the extreme left tail of the distribution, which means the hypothesis is rejected at the 99% confidence interval.
- H2. This hypothesis is supported overall. We can see, on multiple occasions, that the models' performance is worse during stress event windows than in the full sample; in most cases, VaR and ES levels rose significantly.
- H3. This hypothesis is also strongly supported: across all model calibrations, it is much more difficult to achieve a 99% confidence interval than a 95% one; the models exhibit a significantly higher breach rate at 1% than at 5%. All the backtests reject the null hypothesis, and ES calibration becomes more difficult because tail observations tend to cluster during stressed periods.

8.6 Limitations

Despite the contributions of this thesis, several limitations should be taken into account. For example, in this empirical analysis, we use only one asset as a proxy for the US markets, specifically the S&P 500. While this asset is commonly used for risk management analysis, it's important to note that results would not necessarily be the same if other asset classes or indexes were included. Secondly, in the methodology, we use a 250-day rolling window. This number, even if it's a standard form for evaluating risk metrics, remains an assumption, and different event windows could alter the models' responsiveness and, in turn, change the final results, especially at $\alpha = 99\%$. Thirdly, the ES backtesting evaluation depends on the joint calibration



statistic and a mean test; different ES backtesting methodologies could give different results. Finally, the chosen event periods were also a modelling choice; they are fixed data periods and may not perfectly match the true event boundaries, leading to some windows with relatively few observations.

8.7 Future Research

The limitation from Chapter 8.6 could motivate many future research opportunities. A natural extension of this dissemination would be to evaluate the performance of these models on other indices or portfolios and to stress-test them, especially during the event windows discussed. Another extension would be to introduce more sophisticated volatility models or tail dynamics, such as Extreme Value Theory (EVT), which may be a game-changer, especially at $\alpha = 99\%$. Finally, promising future research would be to treat risk forecasting as a solely event-based problem and to take advantage of machine learning techniques, such as Support Vector Machines, to better categorise events and more accurately predict model performance.

8.8 Summary

In summary, this dissertation provides empirical research of VaR and ES models, under an out-of-sample setting, and shows that the selection of the model is really important, especially when the risk metrics matter most, and that is during economic distress markets. The evidence suggests that EWMA is the most robust model at $\alpha = 95\%$, because it accounts for volatility clustering in returns, but at $\alpha = 99\%$, all the examined models fail to properly calibrate to the data. For ES, we find that the most effective model is ES, a result that makes sense given its fat-tailed assumptions. The broader implication for regulators and practitioners is that, when building risk metrics, more than one metric is usually necessary to draw safe conclusions about expected tail risk. Different models can be useful for different scenarios and better capture the risk.



9. References

- Abboud, A., Anderson, C., Game, A. L., Iercosan, D. A., Inanoglu, H., & Lynch, D. (2021). Banks' backtesting exceptions during the COVID-19 crash: Causes and consequences. *FEDS Notes*.
- Acerbi, C., & Székely, B. (2014). Back-testing expected shortfall. *Risk Magazine*.
- Adrian, T., & Shin, H. S. (2014). Procyclical leverage and value-at-risk. *Review of Financial Studies*, 27(2), 373–403.
- Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3), 203–228.
- Barendse, S., Kole, E., & van Dijk, D. (2023). Backtesting value-at-risk and expected shortfall in the presence of estimation error. *Journal of Financial Econometrics*, 21(2), 528–568.
- Basel Committee on Banking Supervision. (2019). *Minimum capital requirements for market risk* (rev. Feb 2019). Bank for International Settlements.
- Bayer, S. (2022). Regression-based expected shortfall backtesting. *Journal of Financial Econometrics*.
- Belhachemi, R. (2024). Hidden truncation model with heteroskedasticity: S&P 500 index returns reexamined. *Studies in Economics and Finance*, 41(5), 1085–1105.
- Ben Ayed, W., Derbali, A., & Lamouchi, A. (2024). The Basel 2.5 capital regulatory framework and the COVID-19 crisis: Evidence from market risk models. *Palestine Review of Risk and Financial Studies*.
- Berger, T. (2021). Value-at-risk backtesting: Beyond the empirical failure rate. *Expert Systems with Applications*, 177, 114893.
- Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics*, 19(4), 465–474.



Berkowitz, J., & O'Brien, J. (2002). How accurate are value-at-risk models at commercial banks? *The Journal of Finance*, 57(3), 1093–1111.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327.

Boucher, C. M., Daniélsson, J., Kouontchou, P. S., & Maillet, B. B. (2014). Risk models-at-risk. *Journal of Banking & Finance*, 44, 72–92.

Cheng, Y. (2025). Monte Carlo-based VaR estimation and backtesting under Basel III. *Risks*, 13(8), 146.

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International Economic Review*, 39(4), 841–862.

Christoffersen, P., & Pelletier, D. (2004). Backtesting value-at-risk: A duration-based approach. *Journal of Financial Econometrics*, 2(1), 84–108.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236.

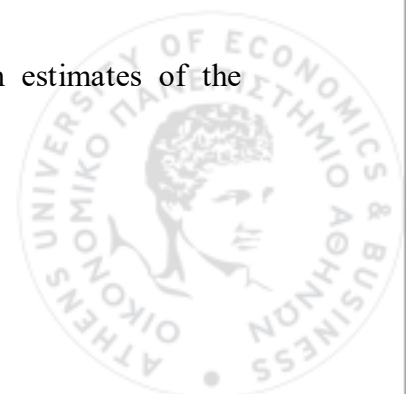
Cuoco, D., & Liu, H. (2006). An analysis of VaR-based capital requirements. *Journal of Financial Intermediation*, 15(3), 362–394.

Daouia, A., Stupfler, G., & Usseglio-Carleve, A. (2025). Corrected inference about the extreme expected shortfall in the general max-domain of attraction. *Information and Inference: A Journal of the IMA*, 14(3).

Dimitriadis, T., & Schnaitmann, J. (2021). Forecast encompassing tests for the expected shortfall. *International Journal of Forecasting*, 37(2), 604–621.

Du, Z., & Escanciano, J. C. (2017). Backtesting expected shortfall: Accounting for tail risk. *Management Science*.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1007.



Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, 38(1), 34–105.

Fissler, T., & Ziegel, J. F. (2016). Higher order elicibility and Osband's principle. *The Annals of Statistics*, 44(4), 1680–1707.

García-Risueño, P. (2025). Historical simulation systematically underestimates the expected shortfall. *Journal of Risk and Financial Management*, 18(1), 34.

Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica*, 74(6), 1545–1578.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494), 746–762.

Hué, S., Hurlin, C., & Lu, Y. (2024). Backtesting expected shortfall: Accounting for both duration and severity with bivariate orthogonal polynomials. *arXiv preprint*.

Hull, J. C. (2021). *Options, futures, and other derivatives* (11th ed.). Pearson.

Jorion, P. (2007). *Value at risk: The new benchmark for managing financial risk* (3rd ed.). McGraw-Hill.

Kratz, M., Lok, Y. H., & McNeil, A. J. (2018). A simple implicit approach to backtesting expected shortfall. *Journal of Banking & Finance*.

Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2), 73–84.

Lazar, E., Pan, J., & Wang, S. (2024). On the estimation of value-at-risk and expected shortfall at extreme levels. *Journal of Commodity Markets*, 34, 100391.

Leung, M. Y., Li, W. K., & Wong, H. Y. (2021). Bayesian value-at-risk backtesting: The case of annuity pricing. *European Journal of Operational Research*.

Lopez, J. A. (1999). Methods for evaluating value-at-risk estimates. *Economic Review (Federal Reserve Bank of San Francisco)*.



Longerstaey, J., & Spencer, M. (1996). *RiskMetrics™ technical document* (4th ed.). J.P. Morgan/Reuters.

MacKinlay, A. C. (1997). Event studies in economics and finance. *Journal of Economic Literature*, 35(1), 13–39.

Mandelbrot, B. (1963). The variation of certain speculative prices. *The Journal of Business*, 36(4), 394–419.

McCullagh, O. (2023). Decoupling VaR and regulatory capital: An examination of the fundamental review of the trading book. *Journal of Banking Regulation*.

McNeil, A. J., Frey, R., & Embrechts, P. (2015). *Quantitative risk management: Concepts, techniques and tools* (2nd ed.). Princeton University Press.

Menganelli, S. (2001). Value at risk models in finance. *European Central Bank Working Paper*.

Meng, X., Taylor, J. W., & Zhang, B. (2020). Estimating value-at-risk and expected shortfall using the asymmetric Laplace distribution. *European Journal of Operational Research*.

Michaelides, M., & Poudyal, N. (2024). Good risk measures, bad statistical assumptions, ugly risk forecasts. *The Financial Review*.

Morkūnaitė, I., Celov, D., & Leipus, R. (2024). Evaluation of value-at-risk (VaR) using the Gaussian mixture models. *Research in Statistics*, 2(1).

Nolde, N., & Ziegel, J. F. (2017). Elicitability and backtesting: Perspectives for banking regulation. *The Annals of Applied Statistics*.

Patton, A. J., Ziegel, J. F., & Chen, R. (2019). Dynamic semiparametric models for expected shortfall (and value-at-risk). *Journal of Econometrics*, 211(2), 388–413.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2(3), 21–41.



Su, Q., Qin, Z., Peng, L., & Qin, G. (2021). Efficiently backtesting conditional value-at-risk and conditional expected shortfall. *Journal of the American Statistical Association*, *116*(536), 2041–2052.

Wang, Q., Wang, R., & Ziegel, J. F. (2025). E-backtesting. *Management Science*.

Yamai, Y., & Yoshida, T. (2002). Comparative analyses of expected shortfall and value-at-risk. *Monetary and Economic Studies*, *20*(1), 87–121.

Zhang, Y., & Nadarajah, S. (2017). A review of backtesting for value at risk. *Communications in Statistics—Theory and Methods*.



10. Appendix



```
import os
import argparse
import warnings
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt

warnings.filterwarnings("ignore")

def set_style():
    plt.rcParams.update({
        "figure.figsize": (12, 6),
        "figure.dpi": 120,
        "savefig.dpi": 300,
        "font.size": 12,
        "axes.titlesize": 14,
        "axes.labelsize": 12,
        "legend.fontsize": 10,
        "xtick.labelsize": 10,
        "ytick.labelsize": 10,
        "lines.linewidth": 2.0,
        "axes.grid": True,
        "grid.alpha": 0.25,
    })

def ensure_dir(path: str) -> None:
    os.makedirs(path, exist_ok=True)

def download_prices(ticker: str, start: str, end: str) -> pd.Series:
    """Download historical price data for a given ticker and date range."""
    """We use this function to make sure the data we downloaded from DATASTREAM is correct."""
    import yfinance as yf
    px = yf.download(ticker, start=start, end=end, auto_adjust=False, progress=False)
    if px is None or px.empty:
        raise ValueError("No data downloaded. Check ticker/date range/internet.")

    if isinstance(px.columns, pd.MultiIndex):
        lv10 = list(px.columns.get_level_values(0).unique())
        lv11 = list(px.columns.get_level_values(1).unique())

        if any(x in ["Close", "Adj Close", "Open", "High", "Low", "Volume"] for x in lv10):
            fields = lv10
            tickers = lv11
            use_ticker = ticker if ticker in tickers else tickers[0]
            for field in ["Adj Close", "Close", "close", "AdjClose"]:
                if field in fields:
                    s = px[(field, use_ticker)].dropna()
                    s.name = "price"
                    return s
            sub = px.xs(use_ticker, axis=1, level=1)
            s = sub.select_dtypes(include=[np.number]).iloc[:, 0].dropna()
            s.name = "price"
            return s

        if any(x in ["Close", "Adj Close", "Open", "High", "Low", "Volume"] for x in lv11):
            tickers = lv10
            fields = lv11
            use_ticker = ticker if ticker in tickers else tickers[0]
            for field in ["Adj Close", "Close", "close", "AdjClose"]:
                if field in fields:
                    s = px[(use_ticker, field)].dropna()
                    s.name = "price"
                    return s
            sub = px.xs(use_ticker, axis=1, level=0)
            s = sub.select_dtypes(include=[np.number]).iloc[:, 0].dropna()
            s.name = "price"
            return s

        px.columns = ["_".join(map(str, c)) for c in px.columns]

    cols = list(px.columns)
    for c in ["Adj Close", "Adj Close", "adjclose", "AdjClose", "Close", "close"]:
        if c in cols:
            s = px[c].dropna()
            if isinstance(s, pd.DataFrame):
                s = s.iloc[:, 0].dropna()
            s.name = "price"
            return s

    candidates = [c for c in cols if "close" in str(c).lower()]
    if candidates:
        s = px[candidates[0]].dropna()
        if isinstance(s, pd.DataFrame):
            s = s.iloc[:, 0].dropna()
        s.name = "price"
        return s

    num = px.select_dtypes(include=[np.number])
    if num.shape[1] == 0:
        raise ValueError(f"Downloaded data has no numeric columns. Columns: {cols}")
    s = num.iloc[:, 0].dropna()
    s.name = "price"
    return s

def log_returns(price: pd.Series) -> pd.Series:
    r = np.log(price / price.shift(1)).dropna()
    r.name = "return"
    return r

def ewma_sigma(returns: pd.Series, lam: float = 0.94) -> pd.Series:
    r = returns.values
    sig2 = np.zeros(len(r), dtype=float)
    sig2[0] = np.nanvar(r)
    for i in range(1, len(r)):
        sig2[i] = lam * sig2[i - 1] + (1.0 - lam) * (r[i - 1] ** 2)
    return pd.Series(np.sqrt(sig2), index=returns.index, name="sigma_ewma")

def var_hs(w: pd.Series, alpha: float) -> float:
    q = np.quantile(w, 1.0 - alpha)
    return float(-q)

def es_hs(w: pd.Series, alpha: float) -> float:
    q = np.quantile(w, 1.0 - alpha)
    tail = w[w <= q]
    return float(-tail.mean()) if len(tail) else float("nan")
```



```

def var_normal(w: pd.Series, alpha: float) -> float:
    mu = w.mean()
    sigma = w.std(ddof=1)
    z = stats.norm.ppf(1.0 - alpha)
    return float(-mu + z * sigma)

def es_normal(w: pd.Series, alpha: float) -> float:
    mu = w.mean()
    sigma = w.std(ddof=1)
    z = stats.norm.ppf(1.0 - alpha)
    phi = stats.norm.pdf(z)
    return float(-mu - sigma * (phi / (1.0 - alpha)))

def fit_t_params(w: pd.Series):
    df, loc, scale = stats.t.fit(w.values)
    return df, loc, scale

def var_t(w: pd.Series, alpha: float) -> float:
    df, loc, scale = fit_t_params(w)
    q = stats.t.ppf(1.0 - alpha, df, loc=loc, scale=scale)
    return float(-q)

def es_t(w: pd.Series, alpha: float) -> float:
    df, loc, scale = fit_t_params(w)
    q = stats.t.ppf(1.0 - alpha, df)
    f = stats.t.pdf(q, df)
    if df <= 1:
        return float("nan")
    es_std = -(((df + q**2) / (df - 1.0)) * (f / (1.0 - alpha)))
    return float(-(loc + scale * es_std))

def var_ewma_sigma(sig: float, alpha: float, mu: float = 0.0) -> float:
    z = stats.norm.ppf(1.0 - alpha)
    return float(-mu + z * sig)

def es_ewma_sigma(sig: float, alpha: float, mu: float = 0.0) -> float:
    z = stats.norm.ppf(1.0 - alpha)
    phi = stats.norm.pdf(z)
    return float(-mu - sig * (phi / (1.0 - alpha)))

def kupiec_uc(breaches: np.ndarray, alpha: float) -> dict:
    breaches = np.asarray(breaches).astype(int)
    T = len(breaches)
    x = int(breaches.sum())
    p = 1.0 - alpha
    phat = x / T if T > 0 else np.nan

    if T == 0 or x == 0 or x == T:
        return {"T": T, "x": x, "p_expected": p, "p_hat": phat, "LR_uc": np.nan, "p_value": np.nan}

    num = (1.0 - p) ** (T - x) * (p ** x)
    den = (1.0 - phat) ** (T - x) * (phat ** x)
    LR = -2.0 * np.log(num / den)
    pval = 1.0 - stats.chi2.cdf(LR, df=1)
    return {"T": T, "x": x, "p_expected": p, "p_hat": phat, "LR_uc": LR, "p_value": pval}

def christoffersen_ind(breaches: np.ndarray) -> dict:
    breaches = np.asarray(breaches).astype(int)
    if len(breaches) < 2:
        return {"LR_ind": np.nan, "p_value": np.nan, "n00": 0, "n01": 0, "n10": 0, "n11": 0}

    b0 = breaches[:-1]
    b1 = breaches[1:]
    n00 = int((b0 == 0) & (b1 == 0)).sum()
    n01 = int((b0 == 0) & (b1 == 1)).sum()
    n10 = int((b0 == 1) & (b1 == 0)).sum()
    n11 = int((b0 == 1) & (b1 == 1)).sum()

    def safe_div(a, b):
        return a / b if b > 0 else 0.0

    pi01 = safe_div(n01, n00 + n01)
    pi11 = safe_div(n11, n10 + n11)
    pi1 = safe_div(n01 + n11, n00 + n01 + n10 + n11)

    eps = 1e-12
    pi01 = min(max(pi01, eps), 1 - eps)
    pi11 = min(max(pi11, eps), 1 - eps)
    pi1 = min(max(pi1, eps), 1 - eps)

    L0 = ((1 - pi1) ** (n00 + n10)) * (pi1 ** (n01 + n11))
    L1 = ((1 - pi01) ** n00) * (pi01 ** n01) * ((1 - pi11) ** n10) * (pi11 ** n11)

    LR = -2.0 * np.log(L0 / L1)
    pval = 1.0 - stats.chi2.cdf(LR, df=1)
    return {"LR_ind": LR, "p_value": pval, "n00": n00, "n01": n01, "n10": n10, "n11": n11}

def conditional_coverage(breaches: np.ndarray, alpha: float) -> dict:
    uc = kupiec_uc(breaches, alpha)
    ind = christoffersen_ind(breaches)
    if np.isnan(uc["LR_uc"]) or np.isnan(ind["LR_ind"]):
        return {"LR_cc": np.nan, "p_value_cc": np.nan, **uc, **ind}
    LR_cc = uc["LR_uc"] + ind["LR_ind"]
    pval = 1.0 - stats.chi2.cdf(LR_cc, df=2)
    return {"LR_cc": LR_cc, "p_value_cc": pval, **uc, **ind}

def hac_se(x: np.ndarray, L: int = 10) -> float:
    x = np.asarray(x, dtype=float)
    x = x[np.isfinite(x)]
    T = len(x)
    if T < 5:
        return np.nan

    x = x - x.mean()
    gamma0 = np.dot(x, x) / T
    s = gamma0

    for l in range(1, min(L, T - 1) + 1):
        w = 1.0 - l / (L + 1.0)
        gam = np.dot(x[1:l], x[1:l]) / T
        s += 2.0 * w * gam

```



```

ax.set_ylabel("ES (positive loss)")
ax.legend(loc="center left", bbox_to_anchor=(1.02, 0.5), frameon=True)
save_fig(fig, outdir, f"es_levels_alpha_{int(alpha*100)}")

def plot_returns_with_var(df: pd.DataFrame, alpha: float, outdir: str, title: str, models):
    sub = df.xs(alpha, level="alpha").reset_index().set_index("date")
    fig, ax = plt.subplots(figsize=(13, 6))
    ax.plot(sub.index, sub["return"], alpha=0.35, label="Return")
    for m in models:
        ax.plot(sub.index, -sub[m], label=f"-{m}")
        breach = sub[f"breach_{m}"].astype(bool)
        ax.scatter(sub.index[breach], sub["return"][breach], s=14, alpha=0.9, label=f"Breaches {m}")
    ax.axhline(0.0, linewidth=1)
    ax.set_title(title)
    ax.set_xlabel("Date")
    ax.set_ylabel("Return")
    ax.legend(loc="center left", bbox_to_anchor=(1.02, 0.5), frameon=True)
    tag = " ".join([m.replace("VaR ", "") for m in models])
    save_fig(fig, outdir, f"returns_var_breaches_{tag}_alpha_{int(alpha*100)}")

def write_report(path: str, headline: str, levels: pd.DataFrame, bt: pd.DataFrame, bt_event: pd.DataFrame,
                es_bt: pd.DataFrame, es_bt_event: pd.DataFrame):
    with open(path, "w", encoding="utf-8") as f:
        f.write(headline + "\n\n")
        f.write("SUMMARY LEVELS\n")
        f.write(levels.to_string(index=False))
        f.write("\n\nBACKTEST VAR FULL\n")
        f.write(bt.to_string(index=False))
        f.write("\n\nBACKTEST VAR BY PERIOD\n")
        f.write("EMPTY\n" if bt_event.empty else bt_event.to_string(index=False))
        f.write("\n\nBACKTEST ES FULL\n")
        f.write(es_bt.to_string(index=False))
        f.write("\n\nBACKTEST ES BY PERIOD\n")
        f.write("EMPTY\n" if es_bt_event.empty else es_bt_event.to_string(index=False))
        f.write("\n\n")

DEFAULT_EVENT_WINDOWS = {
    "COVID_Crash": ("2020-02-20", "2020-04-30"),
    "Ukraine Shock": ("2022-02-24", "2022-03-31"),
    "Q4 2018 Selloff": ("2018-10-01", "2018-12-31"),
    "SVB Banking Stress": ("2023-03-08", "2023-03-31"),
    "US Debt Ceiling 2023": ("2023-05-01", "2023-06-15"),
    "Raffi Shock 2025": ("2025-04-02", "2025-04-10"),
    "MiddleEast 2025": ("2025-06-13", "2025-06-30"),
}

def run_pipeline(args):
    ensure_dir(args.outdir)
    set_style()

    price = download_prices(args.ticker, args.start, args.end)
    ret = log_returns(price)

    raw = pd.concat([price.rename("price"), ret.rename("return")], axis=1)
    raw.to_csv(os.path.join(args.outdir, "data_prices_returns.csv"))

    alphas = [float(x) for x in args.alphas.split(",")]
    df = build_forecasts(
        ret=ret,
        window=args.window,
        alphas=alphas,
        lam=args.lambda_ewma,
        event_windows=DEFAULT_EVENT_WINDOWS,
        min_event_days=args.min_event_days
    )
    df.to_csv(os.path.join(args.outdir, "Forecast_table_full.csv"))

    levels = summarize_levels(df)
    bt = run_backtests(df)
    bt_event = run_backtests_by_period(df, min_event_days=args.min_event_days)
    es_bt = run_es_backtests(df, hac_lags=args.hac_lags)
    es_bt_event = run_es_backtests_by_period(df, min_event_days=args.min_event_days, hac_lags=args.hac_lags)

    levels.to_csv(os.path.join(args.outdir, "summary_levels.csv"), index=False)
    bt.to_csv(os.path.join(args.outdir, "backtest_full_sample.csv"), index=False)
    if bt_event is not None and not bt_event.empty:
        bt_event.to_csv(os.path.join(args.outdir, "backtest_by_period.csv"), index=False)
    es_bt.to_csv(os.path.join(args.outdir, "es_backtest_full_sample.csv"), index=False)
    if es_bt_event is not None and not es_bt_event.empty:
        es_bt_event.to_csv(os.path.join(args.outdir, "es_backtest_by_period.csv"), index=False)

    title_prefix = f"{args.ticker} | "
    plot_hist_returns(ret, args.outdir, title_prefix + "Histogram of daily log returns")
    plot_qq_returns(ret, args.outdir, title_prefix + "QQ-plot: returns vs Normal")
    plot_rolling_vol(ret, args.outdir, title_prefix + "Rolling volatility (30-day std)")

    for a in alphas:
        plot_var_levels(df, a, args.outdir, title_prefix + f"VaR levels over time (alpha={a})")
        plot_es_levels(df, a, args.outdir, title_prefix + f"ES levels over time (alpha={a})")
        plot_returns_with_var(df, a, args.outdir, title_prefix + f>Returns and VaR breaches (alpha={a})', [{"VaR_BS", "VaR_Normal"}])
        plot_returns_with_var(df, a, args.outdir, title_prefix + f>Returns and VaR breaches (alpha={a})', [{"VaR_t", "VaR_EWMA"}])

    headline = [
        f"VaR/ES Report\n",
        f"ticker={args.ticker}\n",
        f"start={args.start}\n",
        f"end={args.end}\n",
        f"window={args.window}\n",
        f"lambda_ewma={args.lambda_ewma}\n",
        f"alphas={alphas}\n",
        f"hac_lags={args.hac_lags}\n",
        f"min_event_days={args.min_event_days}\n",
        f"event_windows={DEFAULT_EVENT_WINDOWS}\n"
    ]

    write_report(
        os.path.join(args.outdir, "report.txt"),
        headline=headline,
        levels=levels,
        bt=bt,
        bt_event=bt_event,
        es_bt=es_bt,
        es_bt_event=es_bt_event
    )

def make_figures_from_csv(args):
    ensure_dir(args.figdir)
    set_style()

```



```

    })
    return pd.DataFrame(rows).sort_values(["alpha", "model"])

def run_bactests_by_period(df: pd.DataFrame, min_event_days: int) -> pd.DataFrame:
    rows = []
    for alpha in sorted(df.index.get_level_values("alpha").unique()):
        suba = df.xs(alpha, level="alpha").copy()
        for period, sub in suba.groupby("period"):
            if len(sub) < min_event_days:
                continue
            for model in ["VaR_HS", "VaR_Normal", "VaR_L", "VaR_EWMA"]:
                breaches = sub["breach_{}".format(model)].values
                uc = kupiec uc(breaches, alpha)
                ind = christoffersen ind(breaches)
                cc = conditional_coverage(breaches, alpha)
                rows.append({
                    "alpha": alpha,
                    "period": period,
                    "model": model,
                    "T": uc["T"],
                    "expected_breaches": (1.0 - alpha) * uc["T"],
                    "breaches": uc["x"],
                    "breach_rate": uc["p_hat"],
                    "kupiec_p": uc["p_value"],
                    "christoffersen_p": ind["p_value"],
                    "CondCov_p": cc["p_value_cc"],
                })
    out = pd.DataFrame(rows)
    return out.sort_values(["alpha", "period", "model"]) if not out.empty else out

def run_es_bactests(df: pd.DataFrame, hac_lags: int = 10) -> pd.DataFrame:
    rows = []
    pairs = [
        ("HS", "VaR_HS", "ES_HS"),
        ("Normal", "VaR_Normal", "ES_Normal"),
        ("L", "VaR_L", "ES_L"),
        ("EWMA", "VaR_EWMA", "ES_EWMA"),
    ]
    for alpha in sorted(df.index.get_level_values("alpha").unique()):
        sub = df.xs(alpha, level="alpha")
        for name, vcol, ecol in pairs:
            res = es_bactest_2_mean(sub, alpha, vcol, ecol, hac_lags=hac_lags)
            rows.append({"alpha": alpha, "model": name, "VaR_col": vcol, "ES_col": ecol, **res})
    return pd.DataFrame(rows).sort_values(["alpha", "model"])

def run_es_bactests_by_period(df: pd.DataFrame, min_event_days: int, hac_lags: int = 10) -> pd.DataFrame:
    rows = []
    pairs = [
        ("HS", "VaR_HS", "ES_HS"),
        ("Normal", "VaR_Normal", "ES_Normal"),
        ("L", "VaR_L", "ES_L"),
        ("EWMA", "VaR_EWMA", "ES_EWMA"),
    ]
    for alpha in sorted(df.index.get_level_values("alpha").unique()):
        suba = df.xs(alpha, level="alpha").copy()
        for period, sub in suba.groupby("period"):
            if len(sub) < min_event_days:
                continue
            for name, vcol, ecol in pairs:
                res = es_bactest_2_mean(sub, alpha, vcol, ecol, hac_lags=hac_lags)
                rows.append({"alpha": alpha, "period": period, "model": name, **res})
    out = pd.DataFrame(rows)
    return out.sort_values(["alpha", "period", "model"]) if not out.empty else out

def save_fig(fig, outdir: str, name: str):
    fig.tight_layout()
    fig.savefig(os.path.join(outdir, name + ".png"))
    fig.savefig(os.path.join(outdir, name + ".pdf"))
    plt.close(fig)

def plot_hist_returns(ret: pd.Series, outdir: str, title: str):
    r = ret.dropna().values
    fig, ax = plt.subplots(figsize=(10, 5))
    ax.hist(r, bins=50)
    ax.set_title(title)
    ax.set_xlabel("Return")
    ax.set_ylabel("Frequency")
    save_fig(fig, outdir, "hist_returns")

def plot_qq_returns(ret: pd.Series, outdir: str, title: str):
    r = ret.dropna().values
    fig, ax = plt.subplots(figsize=(6.5, 6.5))
    stats.probplot(r, dist="norm", plot=ax)
    ax.set_title(title)
    save_fig(fig, outdir, "qqplot_returns_vs_normal")

def plot_rolling_vol(ret: pd.Series, outdir: str, title: str):
    roll = ret.rolling(30).std()
    fig, ax = plt.subplots(figsize=(12, 5))
    ax.plot(roll.index, roll.values, label="30-day rolling volatility")
    ax.set_title(title)
    ax.set_xlabel("Date")
    ax.set_ylabel("Volatility")
    ax.legend(loc="upper right")
    save_fig(fig, outdir, "rolling_vol_30d")

def plot_var_levels(df: pd.DataFrame, alpha: float, outdir: str, title: str):
    sub = df.xs(alpha, level="alpha").reset_index().set_index("date")
    fig, ax = plt.subplots(figsize=(13, 6))
    for col in ["VaR_HS", "VaR_Normal", "VaR_L", "VaR_EWMA"]:
        ax.plot(sub.index, sub[col], label=col)
    ax.set_title(title)
    ax.set_xlabel("Date")
    ax.set_ylabel("VaR (positive loss)")
    ax.legend(loc="center left", bbox_to_anchor=(1.02, 0.5), frameon=True)
    save_fig(fig, outdir, f"var_levels_alpha_{int(alpha*100)}")

def plot_es_levels(df: pd.DataFrame, alpha: float, outdir: str, title: str):
    sub = df.xs(alpha, level="alpha").reset_index().set_index("date")
    fig, ax = plt.subplots(figsize=(13, 6))
    for col in ["ES_HS", "ES_Normal", "ES_L", "ES_EWMA"]:
        ax.plot(sub.index, sub[col], label=col)
    ax.set_title(title)
    ax.set_xlabel("Date")

```



```

var_mean = s / T
if var_mean <= 0 or not np.isfinite(var_mean):
    return np.nan
return float(np.sqrt(var_mean))

def es_backtest_Z_mean(sub: pd.DataFrame, alpha: float, var_col: str, es_col: str, hac_lags: int = 10) -> dict:
    tau = 1.0 - alpha
    r = sub["return"].values.astype(float)
    q = (-sub[var_col]).values.astype(float)
    e = (-sub[es_col]).values.astype(float)

    mask = np.isfinite(r) & np.isfinite(q) & np.isfinite(e)
    r, q, e = r[mask], q[mask], e[mask]
    T = len(r)
    if T < 30:
        return {"T": T, "Z_mean": np.nan, "t_stat": np.nan, "p_value": np.nan}

    I = (r <= q).astype(float)
    Z = (1.0 / tau) * (q - r) * I - (q - e)

    Z_mean = float(np.mean(Z))
    se = hac_se(Z, L=hac_lags)

    if se is None or not np.isfinite(se) or se == 0:
        return {"T": T, "Z_mean": Z_mean, "t_stat": np.nan, "p_value": np.nan}

    t_stat = Z_mean / se
    pval = 2.0 * (1.0 - stats.norm.cdf(abs(t_stat)))
    return {"T": T, "Z_mean": Z_mean, "t_stat": float(t_stat), "p_value": float(pval)}

def build_forecasts(
    ret: pd.Series,
    window: int,
    alphas: list,
    lam: float,
    event_windows: dict,
    min_event_days: int
) -> pd.DataFrame:
    sig_ewma = ewma_sigma(ret, lam=lam)
    out_dates = ret.index[window:]
    rows = []

    def tag_period(dt: pd.Timestamp) -> str:
        for name, (a, b) in event_windows.items():
            if pd.Timestamp(a) <= dt <= pd.Timestamp(b):
                return name
        return "Normal"

    for t in out_dates:
        w = ret.loc[t].iloc[-window-1:]
        r_t = float(ret.loc[t])
        sig_t = float(sig_ewma.loc[t])

        for alpha in alphas:
            rows.append({
                "date": t,
                "alpha": alpha,
                "return": r_t,
                "VaR_HS": var_hs(w, alpha),
                "VaR_Normal": var_normal(w, alpha),
                "VaR_t": var_t(w, alpha),
                "VaR_EWMA": var_ewma_sigma(sig_t, alpha, mu=0.0),
                "ES_HS": es_hs(w, alpha),
                "ES_Normal": es_normal(w, alpha),
                "ES_t": es_t(w, alpha),
                "ES_EWMA": es_ewma_sigma(sig_t, alpha, mu=0.0),
            })

    df = pd.DataFrame(rows)
    df["period"] = df["date"].apply(lambda d: tag_period(pd.Timestamp(d)))
    df = df.set_index(["date", "alpha"]).sort_index()

    for c in ["VaR_HS", "VaR_Normal", "VaR_t", "VaR_EWMA"]:
        df[f"breach_{c}"] = (df["return"] < -df[c]).astype(int)

    return df

def summarize_levels(df: pd.DataFrame) -> pd.DataFrame:
    rows = []
    for alpha in sorted(df.index.get_level_values("alpha").unique()):
        sub = df.xs(alpha, level="alpha")
        cols = ["VaR_HS", "VaR_Normal", "VaR_t", "VaR_EWMA", "ES_HS", "ES_Normal", "ES_t", "ES_EWMA"]
        for col in cols:
            rows.append({
                "alpha": alpha,
                "measure": col,
                "mean": sub[col].mean(),
                "median": sub[col].median(),
                "min": sub[col].min(),
                "max": sub[col].max(),
                "p90": sub[col].quantile(0.90),
                "p95": sub[col].quantile(0.95),
            })
    return pd.DataFrame(rows).sort_values(["alpha", "measure"])

def run_backtests(df: pd.DataFrame) -> pd.DataFrame:
    rows = []
    for alpha in sorted(df.index.get_level_values("alpha").unique()):
        sub = df.xs(alpha, level="alpha")
        for model in ["VaR_HS", "VaR_Normal", "VaR_t", "VaR_EWMA"]:
            breaches = sub[f"breach_{model}"].values
            uc = kupiec_uc(breaches, alpha)
            ind = christoffersen_ind(breaches)
            cc = conditional_coverage(breaches, alpha)
            rows.append({
                "alpha": alpha,
                "model": model,
                "q": uc["q"],
                "expected_breaches": (1.0 - alpha) * uc["T"],
                "breaches": uc["k"],
                "breach_rate": uc["p_hat"],
                "Kupiec_LRuc": uc["LR_uc"],
                "Kupiec_p": uc["p_value"],
                "Christoffersen_LRind": ind["LR_ind"],
                "Christoffersen_p": ind["p_value"],
                "CondCov_LRcc": cc["LR_cc"],
                "CondCov_p": cc["p_value_cc"],
                "n00": ind["n00"], "n01": ind["n01"], "n10": ind["n10"], "n11": ind["n11"]
            })

```



```
df = pd.read_csv(args.input_csv, parse_dates=["date"])
df = df.set_index(["date", "alpha"]).sort_index()
alphas = sorted(df.index.get_level_values("alpha").unique())

for col in ["VaR_HS", "VaR_Normal", "VaR_t", "VaR_EWMA"]:
    bcol = f"breach_{col}"
    if bcol not in df.columns:
        df[bcol] = (df["return"] < -df[col]).astype(int)

for a in alphas:
    plot_var_levels(df, a, args.figdir, f"VaR levels over time (alpha={a})")
    plot_es_levels(df, a, args.figdir, f"ES levels over time (alpha={a})")
    plot_returns_with_var(df, a, args.figdir, f>Returns and VaR breaches (alpha={a})", ["VaR_HS", "VaR_Normal"])
    plot_returns_with_var(df, a, args.figdir, f>Returns and VaR breaches (alpha={a})", ["VaR_t", "VaR_EWMA"])

def parse_args():
    p = argparse.ArgumentParser()
    p.add_argument("--mode", choices=["run", "figures"], default="run")

    p.add_argument("--ticker", default="^GSPC")
    p.add_argument("--start", default="2014-01-01")
    p.add_argument("--end", default="2025-12-31")
    p.add_argument("--window", type=int, default=250)
    p.add_argument("--alpha", default="0.95,0.99")
    p.add_argument("--lambda_ewma", type=float, default=0.94)
    p.add_argument("--hac_lags", type=int, default=10)
    p.add_argument("--min_event_days", type=int, default=40)
    p.add_argument("--outdir", default="outputs")

    p.add_argument("--input_csv", default="outputs/forecast_table_full.csv")
    p.add_argument("--figdir", default="outputs/beautiful_figures")

    return p.parse_args()

def main():
    args = parse_args()
    if args.mode == "run":
        run_pipeline(args)
    else:
        make_figures_from_csv(args)

if __name__ == "__main__":
    main()
```

