Time Series Analysis and Forecasting

Chapter 5: GARCH and Volatility

Daniel Traian PELE

Bucharest University of Economic Studies
IDA Institute Digital Assets
Blockchain Research Center
AI4EFin Artificial Intelligence for Energy Finance
Romanian Academy, Institute for Economic Forecasting
MSCA Digital Finance

Outline

Foundations

Applications

Learning Objectives

By the end of this chapter, you will be able to:

1. Understand **volatility clustering** and stylized facts of financial returns
2. Estimate and interpret **ARCH** and **GARCH** models
3. Apply asymmetric models (**EGARCH**, **GJR-GARCH**) for the leverage effect
4. Perform model validation and selection
5. Forecast volatility and calculate **Value at Risk (VaR)**

Practical Skills

- Python implementation with the `arch` package
  - Estimation, forecasting, and automatic diagnostics
- Interpreting parameters and volatility persistence
- VaR calculation for risk management
  - Backtesting and forecast validation

Why Model Volatility?

Empirical Observations in Financial Series

- Financial returns exhibit **volatility clustering** — periods of high volatility tend to be followed by periods of high volatility
- The distribution of returns has **fat tails** (leptokurtosis)
- Return correlation is nearly zero, but correlation of squares is significant
- Volatility responds **asymmetrically** to shocks (leverage effect)

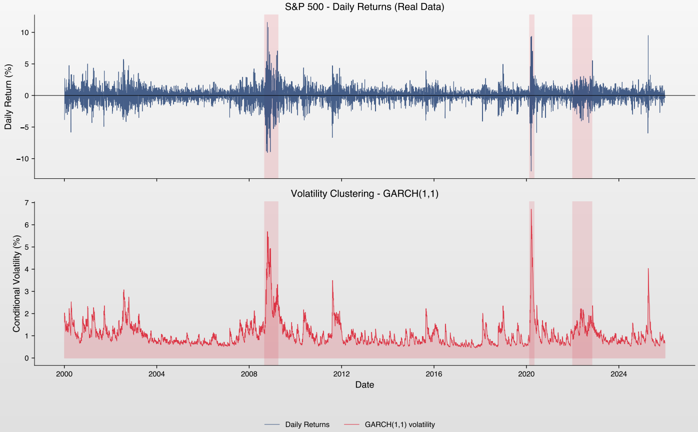
Limitation of ARIMA Models

ARIMA models assume **constant variance** (homoskedasticity), which is not realistic for financial series!

Volatility Clustering

- High volatility periods are followed by high volatility; calm by calm
- This suggests that **conditional variance** is predictable

Volatility Clustering



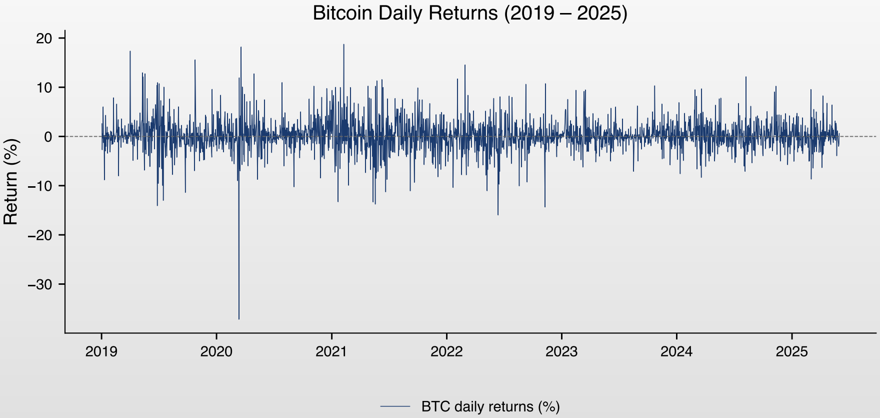
TSA_ch5_clustering

Example: Bitcoin – Volatility Clustering

Observations

- Bitcoin daily returns (2019–2025): extremely pronounced volatility clustering
  - Returns of $\pm 20\%$ during crisis periods (COVID, Terra/Luna)
- Bitcoin volatility is significantly higher than traditional assets
  - Typical $\alpha \approx 0.10$–0.20 (fast reaction to news)

Example: Bitcoin ≻ Volatility Clustering



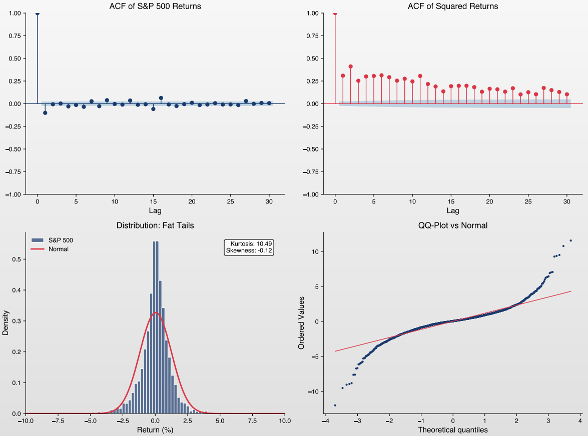
TSA_ch5_btc_returns

Stylized Facts of Financial Returns

Observed Properties

1. **No autocorrelation** in returns
2. **Autocorrelation** in r_t^2, $|r_t|$
3. **Fat tails** (kurtosis > 3)
4. **Leverage effect**
5. **Volatility clustering**

Stylized Facts of Financial Returns



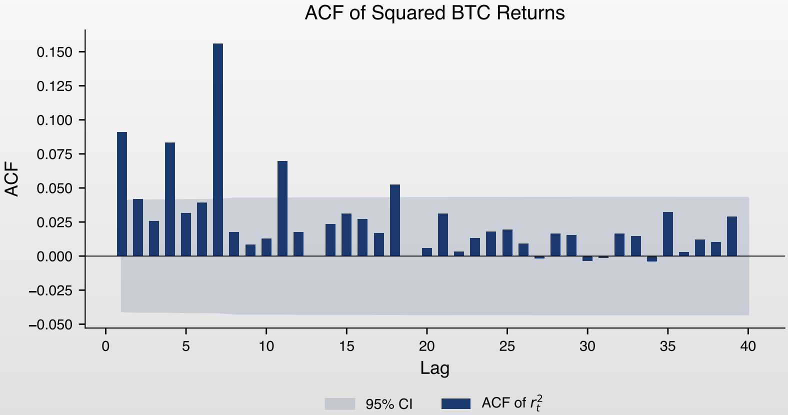
TSA_ch5_stylized

Exam

Interpretation

- **Top**: r_t^2 (volatility proxy) $\succ$ peaks coincide with market crises
- **Bottom**: ACF(r_t^2) significant $\succ$ ARCH effects present, variance is predictable

Example: Bitcoin ≻ Evidence for ARCH Effects



TSA_ch5_btc_arch

Conditional Heteroskedasticity

Definition 1 (Conditional Variance)

For return series $\{r_t\}$, the **conditional variance** at time t is: $\sigma_t^2 = \text{Var}(r_t|\mathcal{F}_{t-1}) = \mathbb{E}[(r_t - \mu_t)^2|\mathcal{F}_{t-1}]$ where $\mathcal{F}_{t-1}$ is the information available up to time $t-1$.

General Model

$r_t = \mu_t + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t, \quad z_t \sim \text{i.i.d.}(0, 1)$

- μ_t = conditional mean (ARMA); σ_t^2 = conditional variance (GARCH)
- z_t = standardized innovations (Normal, Student-t, GED)

Researcher Spotlight: Engle & Bollerslev

Robert Engle (*1942)
Nobel Prize 2003
W Wikipedia

Tim Bollerslev (*1958)
W Wikipedia

Biography

- **Robert Engle**: American economist at NYU Stern. Nobel Prize (2003) "for methods of analyzing economic time series with time-varying volatility (ARCH)"
- **Tim Bollerslev**: Danish-American economist at Duke University, PhD student of Engle

Key Contributions

- **ARCH model** (Engle, 1982) — autoregressive conditional heteroskedasticity
- **GARCH model** (Bollerslev, 1986) — generalized ARCH with persistent volatility
- **Realized volatility** measures and high-frequency econometrics
- Foundation for modern financial risk management (VaR, ES)

The ARCH(q) Model — Engle (1982)

Definition 2 (ARCH(q))

The **Autoregressive Conditional Heteroskedasticity** model of order q:

$$\varepsilon_t = \sigma_t z_t, \quad z_t \sim \text{i.i.d.}(0,1), \quad \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$$

Stationarity Restrictions

- $\omega > 0$ (positive base variance), $\alpha_i \geq 0$ (non-negativity)
- $\sum_{i=1}^{q} \alpha_i < 1$ (stationarity)

Remark 1

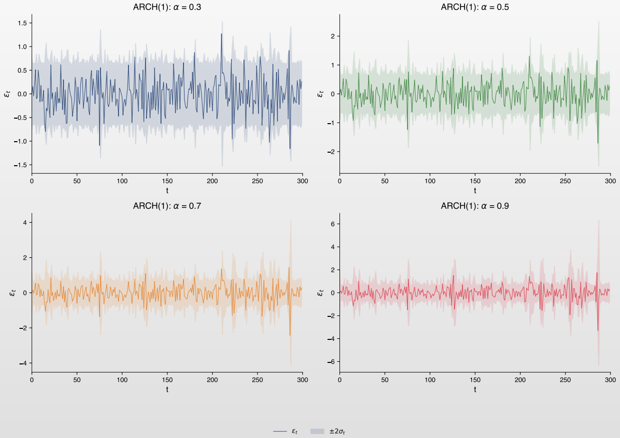
Robert Engle received the **Nobel Prize in Economics** in 2003 for developing the ARCH model!

ARCH(1) Simulation: Effect of α Parameter

Interpretation

- Higher α means volatility reacts more strongly to recent shocks

ARCH(1) Simulation: Effect of α Parameter



TSA_ch5_arch_sim

Properties of the ARCH(1) Model

ARCH(1): $\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2$

- **Unconditional variance**: $\mathbb{E}[\varepsilon_t^2] = \dfrac{\omega}{1-\alpha_1}$ (if $\alpha_1 < 1$)
- **Kurtosis**: $\kappa = 3 \cdot \dfrac{1-\alpha_1^2}{1-3\alpha_1^2}$ (if $\alpha_1^2 < 1/3$)
- Kurtosis > 3 for $\alpha_1 > 0 \Rightarrow$ **fat tails**!

Numerical Example

If $\omega = 0.0001$ and $\alpha_1 = 0.3$:

- Unconditional variance: $\sigma^2 = \frac{0.0001}{1-0.3} = 0.000143$
- Kurtosis: $\kappa = 3 \cdot \frac{1-0.09}{1-0.27} = 3.74 > 3$

Derivation: Unconditional Variance of ARCH(1)

Derivation.

Let $\varepsilon_t = \sigma_t z_t$ with $z_t \sim N(0,1)$ and $\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2$.

Step 1: Take unconditional expectation:

$$\mathbb{E}[\varepsilon_t^2] = \mathbb{E}[\sigma_t^2 z_t^2] = \mathbb{E}[\sigma_t^2] \cdot \mathbb{E}[z_t^2] = \mathbb{E}[\sigma_t^2]$$

Step 2: Apply expectation to variance equation:

$$\mathbb{E}[\sigma_t^2] = \mathbb{E}[\omega + \alpha_1 \varepsilon_{t-1}^2] = \omega + \alpha_1 \mathbb{E}[\varepsilon_{t-1}^2]$$

Step 3: By stationarity, $\mathbb{E}[\varepsilon_t^2] = \mathbb{E}[\varepsilon_{t-1}^2] = \sigma^2$:

$$\sigma^2 = \omega + \alpha_1 \sigma^2 \quad \Rightarrow \quad \sigma^2(1 - \alpha_1) = \omega$$

Result: $\boxed{\sigma^2 = \dfrac{\omega}{1 - \alpha_1}}$ (requires $\alpha_1 < 1$ for stationarity) $\square$

Derivation: Kurtosis of ARCH(1)

For $\varepsilon_t = \sigma_t z_t$ with $z_t \sim N(0,1)$:
Step 1: $\mathbb{E}[\varepsilon_t^4] = \mathbb{E}[\sigma_t^4] \cdot \mathbb{E}[z_t^4] = 3\mathbb{E}[\sigma_t^4]$ (since $\mathbb{E}[z^4] = 3$)
Step 2: Using $\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2$:

$$\mathbb{E}[\sigma_t^4] = \mathbb{E}[(\omega + \alpha_1 \varepsilon_{t-1}^2)^2] = \omega^2 + 2\omega\alpha_1\sigma^2 + \alpha_1^2 \mathbb{E}[\varepsilon_{t-1}^4]$$

Step 3: Solving the recursion yields:

$$\boxed{\kappa = \frac{\mathbb{E}[\varepsilon_t^4]}{(\mathbb{E}[\varepsilon_t^2])^2} = 3 \cdot \frac{1-\alpha_1^2}{1-3\alpha_1^2}}$$

Interpretation

- $\kappa > 3$ for any $\alpha_1 > 0 \Rightarrow$ **fat tails** (leptokurtosis)
- Requires $\alpha_1 < 0.577$ for finite fourth moment
- ARCH naturally generates heavy-tailed distributions!

Testing for ARCH Effects

Engle's Test for ARCH Effects

Procedure:

1. Estimate the mean model and obtain residuals $\hat{\varepsilon}_t$
2. Calculate $\hat{\varepsilon}_t^2$
3. Regress $\hat{\varepsilon}_t^2$ on its lags:
$$\hat{\varepsilon}_t^2 = \beta_0 + \beta_1 \hat{\varepsilon}_{t-1}^2 + \cdots + \beta_q \hat{\varepsilon}_{t-q}^2 + u_t$$
4. Calculate the statistic $LM = T \cdot R^2 \sim \chi^2(q)$

Hypotheses

- H_0: No ARCH effects ($\alpha_1 = \cdots = \alpha_q = 0$)
- H_1: ARCH effects present (at least one $\alpha_i \neq 0$)

Limitations of the ARCH Model

Practical Problems

1. **High order** — many lags are usually needed (large q)
2. **Many parameters** — estimation difficulties
3. **Non-negativity constraints** — difficult to impose for large q
4. **Does not capture persistence** — observed volatility is very persistent

The Solution

The GARCH Model — introduces lags of conditional variance to capture persistence with fewer parameters!

The GARCH(p,q) Model — Bollerslev (1986)

Definition 3 (GARCH(p,q))

The **Generalized ARCH** model:

$$\varepsilon_t = \sigma_t z_t, \quad z_t \sim \text{i.i.d.}(0,1)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

Interpretation

- ω = base level of volatility
- α_i = reaction to recent shocks (news coefficients)
- β_j = volatility persistence (memory)
- $\alpha + \beta$ = total persistence

The GARCH(1,1) Model

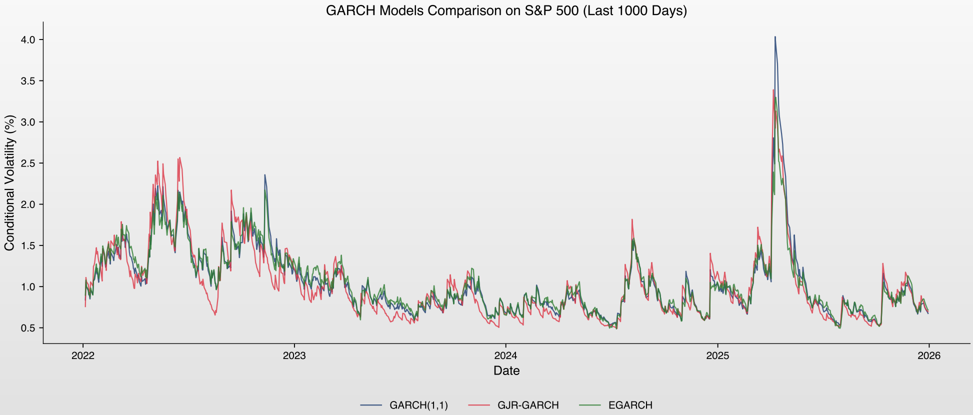
The Most Popular Volatility Model

$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$

Restrictions & Properties

- $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$; $\alpha + \beta < 1$ (stationarity)
- $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$; Half-life: $HL = \frac{\ln(0.5)}{\ln(\alpha+\beta)}$

The GARCH(1,1) Model

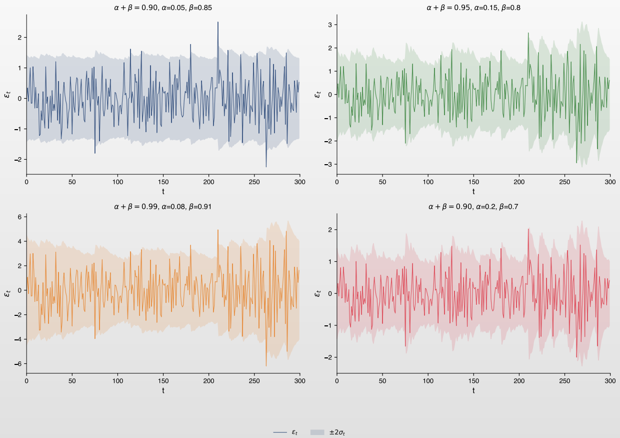


GARCH(1,1) Simulation: Persistence Effect

Interpretation

- α controls reaction to shocks
- β controls persistence
- The sum $\alpha + \beta$ determines mean-reversion speed

GARCH(1,1) Simulation: Persistence Effect



TSA_ch5_garch_sim

Derivation: Unconditional Variance of GARCH(1,1)

Derivation.

For $\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$:

Step 1: Take unconditional expectation: $\mathbb{E}[\sigma_t^2] = \omega + \alpha\mathbb{E}[\varepsilon_{t-1}^2] + \beta\mathbb{E}[\sigma_{t-1}^2]$

Step 2: By stationarity, $\mathbb{E}[\sigma_t^2] = \mathbb{E}[\sigma_{t-1}^2] = \bar{\sigma}^2$ and $\mathbb{E}[\varepsilon_t^2] = \bar{\sigma}^2$: $\bar{\sigma}^2 = \omega + (\alpha+\beta)\bar{\sigma}^2$

Step 3: Solve: $\bar{\sigma}^2(1-\alpha-\beta) = \omega \Rightarrow \boxed{\bar{\sigma}^2 = \dfrac{\omega}{1-\alpha-\beta}}$ □

Stationarity Condition

Requires $\alpha + \beta < 1$ for finite unconditional variance.

GARCH(1,1) as ARMA for ε_t^2

ARMA(1,1) Representation

Define $\nu_t = \varepsilon_t^2 - \sigma_t^2$ (variance shock). Then:

$$\varepsilon_t^2 = \omega + (\alpha + \beta)\varepsilon_{t-1}^2 + \nu_t - \beta\nu_{t-1}$$

This is an **ARMA(1,1)** for ε_t^2!

Implications

- ACF of ε_t^2 decays exponentially (like ARMA)
- Persistence is given by $\alpha + \beta$
- PACF can help identify the order

Derivation: ARMA Representation of GARCH(1,1)

Derivation.

Step 1: Define variance shock: $\nu_t = \varepsilon_t^2 - \sigma_t^2$

- $\mathbb{E}[\nu_t|\mathcal{F}_{t-1}] = \mathbb{E}[\varepsilon_t^2|\mathcal{F}_{t-1}] - \sigma_t^2 = \sigma_t^2 - \sigma_t^2 = 0$
- ν_t is a martingale difference sequence

Step 2: Substitute $\sigma_t^2 = \varepsilon_t^2 - \nu_t$ into GARCH equation:

$$\varepsilon_t^2 - \nu_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta(\varepsilon_{t-1}^2 - \nu_{t-1})$$

Step 3: Rearrange:

$$\varepsilon_t^2 = \omega + (\alpha + \beta)\varepsilon_{t-1}^2 + \nu_t - \beta\nu_{t-1}$$

Result: ARMA(1,1) with AR coefficient $\phi = \alpha + \beta$ and MA coefficient $\theta = -\beta$. □

Derivation: Volatility Persistence and Half-Life

Multi-Step Forecast GARCH(1,1)

- $\mathbb{E}_t[\sigma^2_{t+h}] = \bar{\sigma}^2 + (\alpha + \beta)^{h-1}(\sigma^2_{t+1} - \bar{\sigma}^2)$

Derivation

- **Step 1**: Let $\phi = \alpha + \beta$ and $q_t = \sigma^2_t - \bar{\sigma}^2$ (deviation from mean)
- **Step 2**: From the GARCH equation: $\mathbb{E}_t[q_{t+1}] = \phi \cdot q_t$, so $\mathbb{E}_t[q_{t+h}] = \phi^h \cdot q_t$
- **Step 3**: Half-life = time until deviation halves: $\phi^{HL} = 0.5 \Rightarrow HL = \frac{\ln(0.5)}{\ln(\phi)} = \frac{-0.693}{\ln(\alpha+\beta)}$

Example: S&P 500

- With $\alpha + \beta = 0.988$: $HL = \frac{-0.693}{-0.012} \approx$ **58 days** (shocks persist ∼3 months!)

Estimation of GARCH Models

Maximum Likelihood Estimation (MLE)

Log-likelihood (normal): $\ell(\theta) = -\frac{T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\left[\ln(\sigma_t^2) + \frac{\varepsilon_t^2}{\sigma_t^2}\right]$

Alternative Distributions for z_t

- **Student-t**: captures fat tails — most common choice
- **GED**: flexibility for kurtosis
- **Skewed Student-t**: asymmetry and fat tails

Practical Note

Student-t distribution typically provides better fit for financial returns due to fat tails (kurtosis > 3).

Typical Values for GARCH(1,1)

Series	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\alpha+\beta}$
S&P 500 daily	0.05–0.10	0.85–0.95	0.95–0.99
EUR/USD daily	0.03–0.08	0.90–0.95	0.95–0.99
Bitcoin daily	0.10–0.20	0.75–0.85	0.90–0.98
Bonds	0.02–0.05	0.90–0.97	0.95–0.99

Observations

- $\alpha+\beta$ close to 1 $\Rightarrow$ **very persistent volatility**
- Small α, large β $\Rightarrow$ slow reaction to shocks, long memory
- Bitcoin: larger α $\Rightarrow$ faster reaction to news

IGARCH — Integrated GARCH

Definition 4 (IGARCH(1,1))

When $\alpha + \beta = 1$:

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + (1-\alpha)\sigma_{t-1}^2$$

Properties

- Unconditional variance does not exist (infinite)
- Shocks have **permanent** effect on volatility
- Used for series with extreme persistence
- Useful for **RiskMetrics** (J.P. Morgan): $\alpha = 0.06$, $\beta = 0.94$

Remark 2

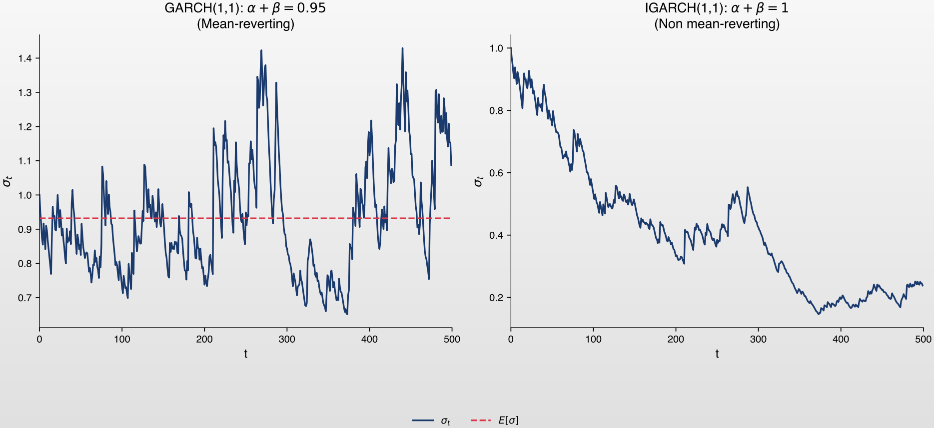
IGARCH is analogous to a unit root in variance!

GARCH vs IGARCH: Persistence Comparison

Interpretation

- Standard GARCH reverts to unconditional mean
- IGARCH has no finite mean $\Rightarrow$ shocks persist indefinitely

GARCH vs IGARCH: Persistence Comparison



TSA_ch5_igarch

Leverage Effect

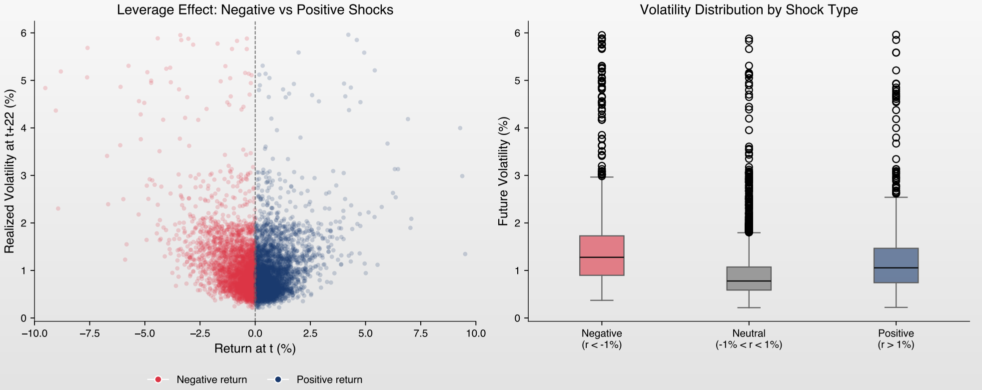
Definition

Leverage effect: Negative shocks increase volatility **more** than positive shocks of the same magnitude.

Problem with GARCH

Standard GARCH: $\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$ — only ε_{t-1}^2 matters, sign is lost! Economic intuition: Bad news $\Rightarrow$ stock price falls $\Rightarrow$ debt/equity ratio rises $\Rightarrow$ volatility increases.

Leverage Effect



The EGARCH Model — Nelson (1991)

Definition 5 (EGARCH(1,1))

Exponential GARCH:

$$\ln(\sigma_t^2) = \omega + \alpha\left(|z_{t-1}| - \mathbb{E}[|z_{t-1}|]\right) + \gamma z_{t-1} + \beta \ln(\sigma_{t-1}^2)$$

where $z_t = \varepsilon_t/\sigma_t$.

EGARCH Advantages

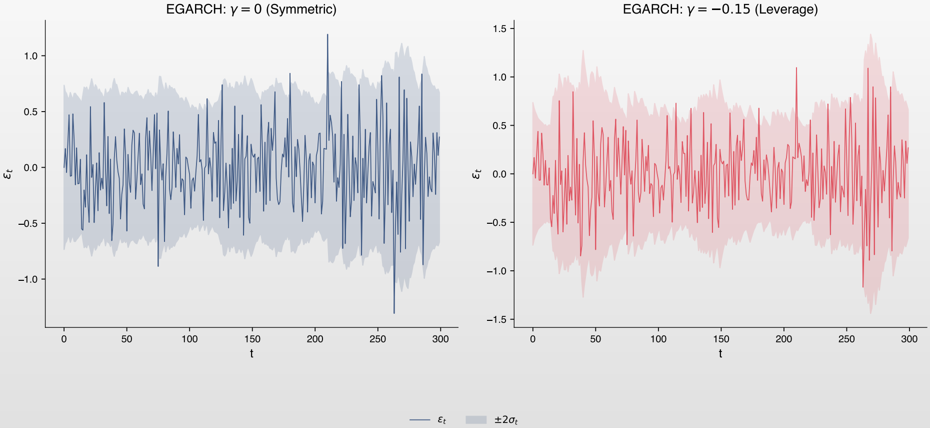
- **No non-negativity constraints required** — models $\ln(\sigma_t^2)$
- **Captures leverage effect** through parameter γ
  - $\gamma < 0$: negative shocks $\Rightarrow$ higher volatility
  - $\gamma = 0$: symmetric effect (like GARCH)
- Persistence is given by β

EGARCH Simulation: Symmetric vs Asymmetric

Interpretation

- When $\gamma < 0$, negative shocks increase volatility more than positive shocks

EGARCH Simulation: Symmetric vs Asymmetric



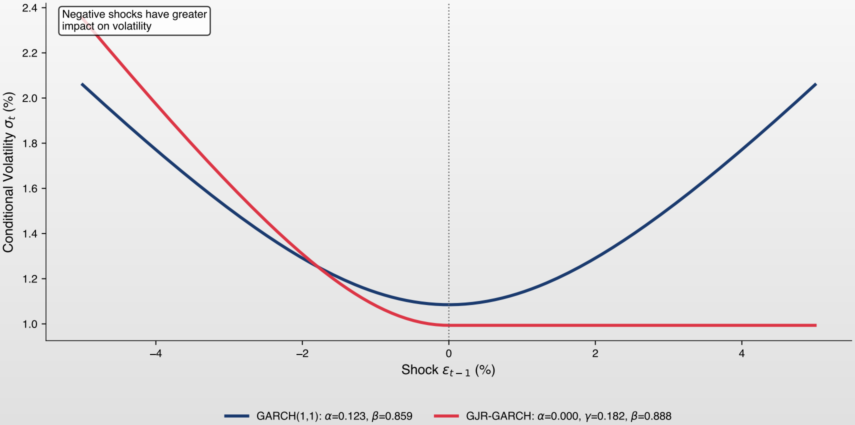
TSA_ch5_egarch_sim

News Impact Curve — EGARCH

Interpretation

- **News Impact Curve**: relationship between ε_t and σ^2_{t+1}
- **GARCH**: symmetric curve (parabola)
  - Positive and negative shocks have the same impact
- **EGARCH**: asymmetric curve
  - Negative shocks have larger impact on volatility

News Impact Curve — EGARCH



TSA_ch5_nic

The GJR-GARCH Model

Definition 6 (GJR-GARCH(1,1))

Glosten, Jagannathan & Runkle (1993): $\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \gamma\varepsilon_{t-1}^2 \cdot I_{t-1} + \beta\sigma_{t-1}^2$ where $I_{t-1} = 1$ if $\varepsilon_{t-1} < 0$, else 0.

Interpretation

- Positive shocks: impact $= \alpha$; Negative shocks: impact $= \alpha + \gamma$
- Leverage effect present if $\gamma > 0$
- Stationarity: $\alpha + \gamma/2 + \beta < 1$

TGARCH — Threshold GARCH

Definition 7 (TGARCH(1,1))

Zakoian (1994) models standard deviation: $\sigma_t = \omega + \alpha^+ \varepsilon^+_{t-1} + \alpha^- \varepsilon^-_{t-1} + \beta \sigma_{t-1}$

Comparison of Asymmetric Models

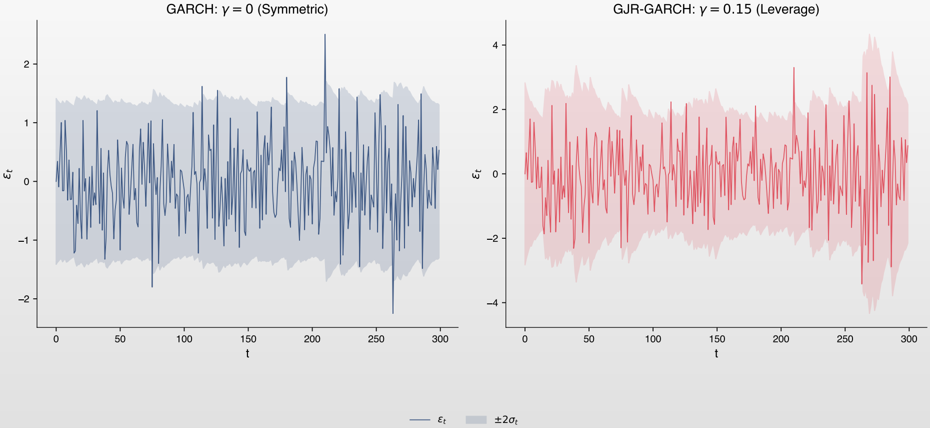
Model	Specification	Leverage
GARCH	σ_t^2	No
EGARCH	$\ln(\sigma_t^2)$	Yes ($\gamma < 0$)
GJR-GARCH	σ_t^2 with indicator	Yes ($\gamma > 0$)
TGARCH	σ_t	Yes ($\alpha^- > \alpha^+$)

GJR-GARCH/TGARCH Simulation

Interpretation

- GJR-GARCH adds an indicator term to capture asymmetric response to negative shocks

GJR-GARCH/TGARCH Simulation



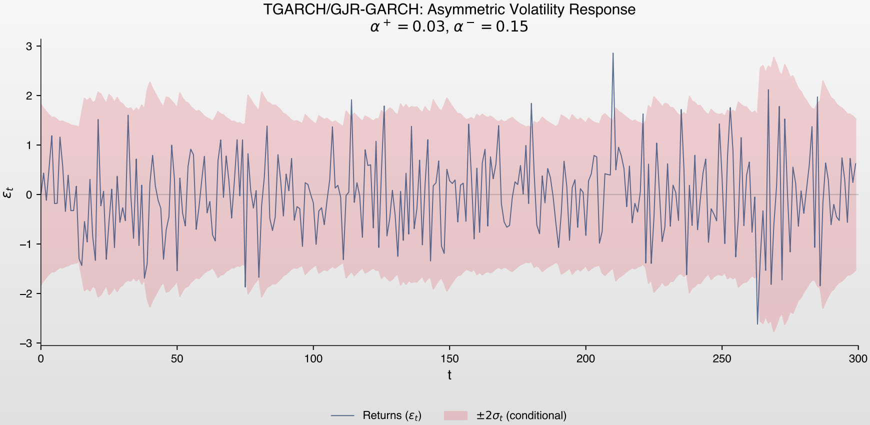
TSA_ch5_gjr_sim

TGARCH Simulation: Asymmetric Volatility Response

Interpretation

- TGARCH with $\alpha^+ = 0.03$ and $\alpha^- = 0.15 \succ$ negative shocks amplify volatility by $5\times$
- Volatility bands $\pm 2\sigma$ widen asymmetrically during crisis periods

TGARCH Simulation: Asymmetric Volatility Response



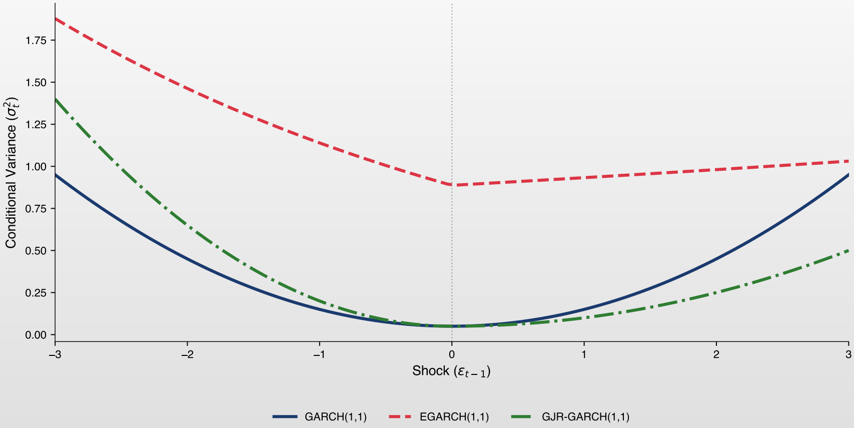
TSA_ch5_tgarch_sim

News Impact Curves Comparison

Interpretation

- **Standard GARCH**: symmetric
  - Treats positive and negative shocks identically
- **EGARCH** and **GJR-GARCH**: capture asymmetry
  - Leverage effect: negative shocks $\Rightarrow$ larger impact

News Impact Curves Comparison



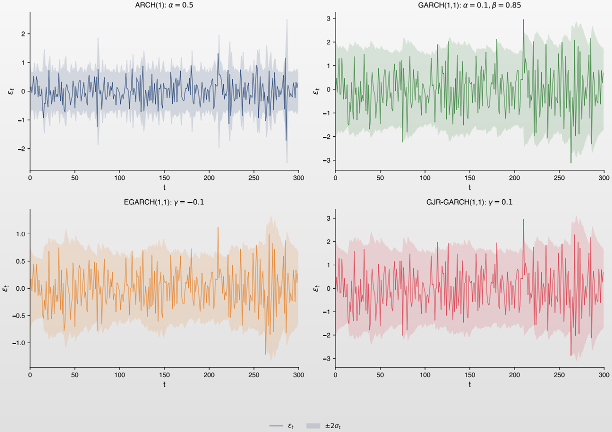
TSA_ch5_nic_comp

GARCH Family Comparison

Interpretation

- All models capture volatility clustering, but differ in how they model asymmetry

GARCH Family Comparison



TSA_ch5_family

GARCH-in-Mean (GARCH-M) — Engle, Lilien & Robins (1987)

Definition 8 (GARCH-M)

- **Model**: Volatility enters directly into the mean equation:

$$r_t = \mu + \delta \cdot g(\sigma_t^2) + \varepsilon_t, \quad \varepsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

- **Function** g: can be σ_t^2, σ_t, or $\ln(\sigma_t^2)$

Economic Interpretation

- $\delta > 0$: **risk premium** $\Rightarrow$ higher returns when volatility is high
- Formalizes the risk-return relationship (CAPM, Merton ICAPM); test $H_0 : \delta = 0$

Typical Example: Equities

- $r_t = 0.02 + \underbrace{0.15}_{\delta} \cdot \sigma_t + \varepsilon_t \quad \Rightarrow$ At $\sigma_t = 2\%$: $\mathbb{E}[r_t] = 0.023$ (0.3% premium)

GARCH-M: Alternative Specifications

Common Specifications

Risk premium can enter in different forms: (1) $r_t = \mu + \lambda\sigma_t + \varepsilon_t$; (2) $r_t = \mu + \lambda\sigma_t^2 + \varepsilon_t$; (3) $r_t = \mu + \lambda \ln(\sigma_t^2) + \varepsilon_t$

Typical Results for Equity Markets

- Estimated λ often positive but small (0.01–0.10)
- Significance varies across markets and periods
- Variance specification yields larger λ estimates

Remark 3

GARCH-M is used in asset pricing, portfolio optimization, and CAPM testing.

Order Selection

Information Criteria

- **AIC** $= -2\ell + 2k$
- **BIC** $= -2\ell + k\ln(T)$
- **HQIC** $= -2\ell + 2k\ln(\ln(T))$

where ℓ = maximized log-likelihood, k = number of parameters.

Practical Recommendations

- GARCH(1,1) is sufficient in **90% of cases**
- Check if asymmetric model significantly improves fit
- Choose innovation distribution that minimizes AIC/BIC

GARCH Model Diagnostics

Standardized Residuals

$$\hat{z}_t = \frac{\hat{\varepsilon}_t}{\hat{\sigma}_t}$$

If the model is correctly specified, $\hat{z}_t$ should be i.i.d.(0,1).

Diagnostic Checks

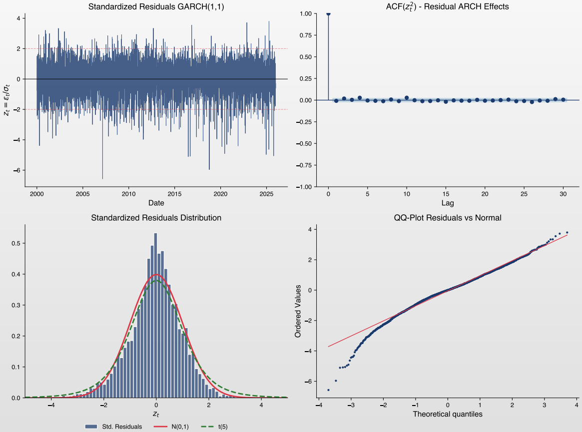
1. **Ljung-Box on** $\hat{z}_t$: check absence of autocorrelation in mean
2. **Ljung-Box on** $\hat{z}_t^2$: check absence of residual ARCH effects
3. **ARCH-LM test on** $\hat{z}_t$: confirm absence of heteroskedasticity
4. **Histogram + QQ-plot**: verify assumed distribution

Diagnostic Example

Verification

- Standardized residuals should be i.i.d. with no residual ARCH effects

Diagnostic Example



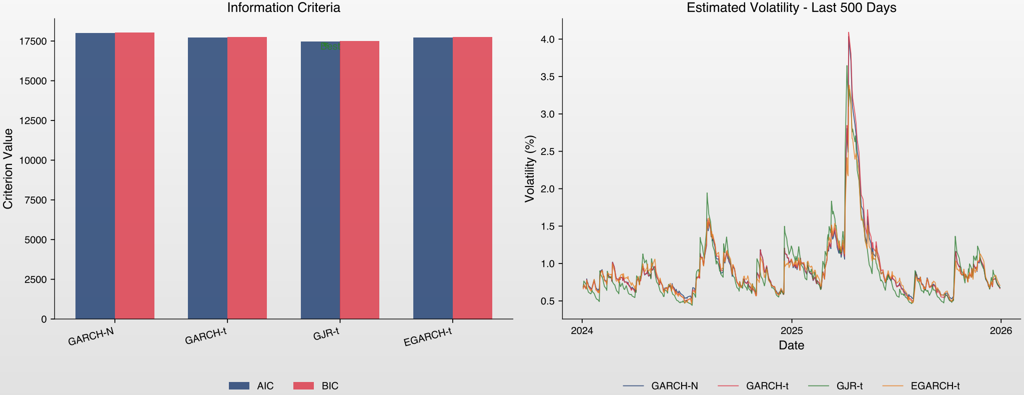
TSA_ch5_diagnostic

GARCH Model Comparison: Validation

Interpretation

- GARCH(1,1) achieves the lowest MAE on the validation set
  - More parsimonious and stable than higher-order models
- GARCH(2,1) and GJR-GARCH: similar performance, but more parameters
- **Conclusion**: simplicity wins $\succ$ GARCH(1,1) is hard to beat

GARCH Model Comparison: Validation



TSA_ch5_comparison

Forecasting with GARCH(1,1)

One-Step-Ahead Forecast

$\hat{\sigma}^2_{T+1} = \omega + \alpha\varepsilon^2_T + \beta\sigma^2_T$

Multi-Step Forecast

For $h > 1$: $\mathbb{E}_T[\sigma^2_{T+h}] = \bar{\sigma}^2 + (\alpha + \beta)^{h-1}(\sigma^2_{T+1} - \bar{\sigma}^2)$ where $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$ = unconditional variance.

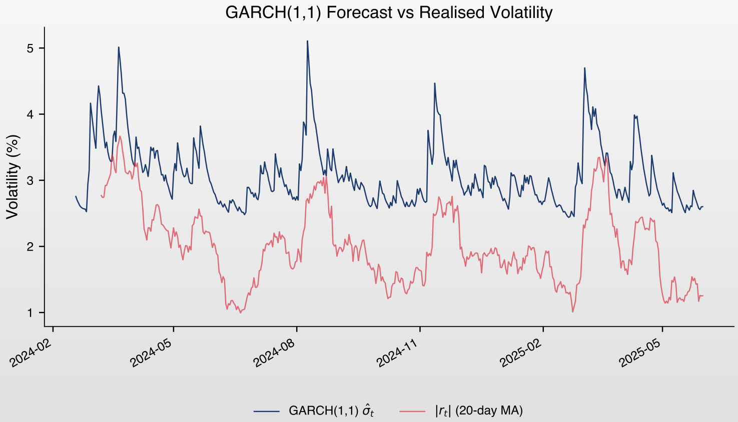
Convergence

$\lim_{h\to\infty} \mathbb{E}_T[\sigma^2_{T+h}] = \bar{\sigma}^2$ — forecast converges to unconditional variance!

Volatility Forecast — Visualization

- Forecast converges exponentially to $\bar{\sigma}^2$; speed depends on $\alpha + \beta$
- The closer $\alpha + \beta$ is to 1, the slower the convergence

Volatility Forecast — Visualization



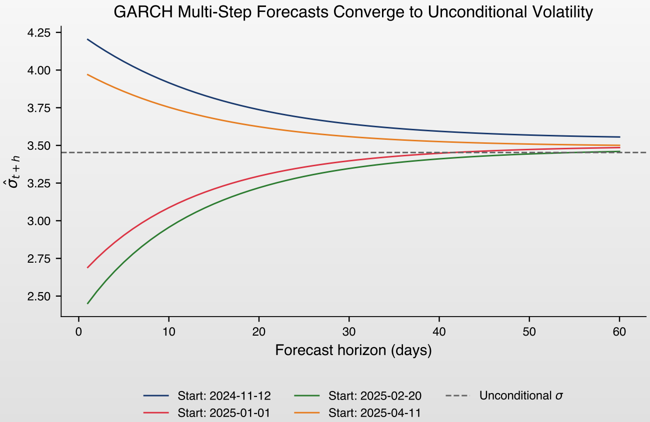
TSA_ch5_vol_forecast

GARCH Forecast Convergence to Unconditional Variance

Interpretation

- Multi-step forecast converges exponentially to $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$
- The closer $\alpha + \beta$ is to 1, the slower the convergence
  - S&P 500: $\alpha + \beta \approx 0.99 \succ$ convergence in $\sim$50 days
  - Bitcoin: $\alpha + \beta \approx 0.95 \succ$ faster convergence

GARCH Forecast Convergence to Unconditional Variance



TSA_ch5_convergence

Applications of Volatility Forecasting

Value at Risk (VaR)

$\text{VaR}_\alpha = z_\alpha \cdot \sigma_{T+1}$
Maximum loss with probability $1 - \alpha$.

Expected Shortfall (ES)

$\text{ES}_\alpha = \mathbb{E}[-r | r < -\text{VaR}_\alpha]$
Average loss when VaR is exceeded.

Other Applications

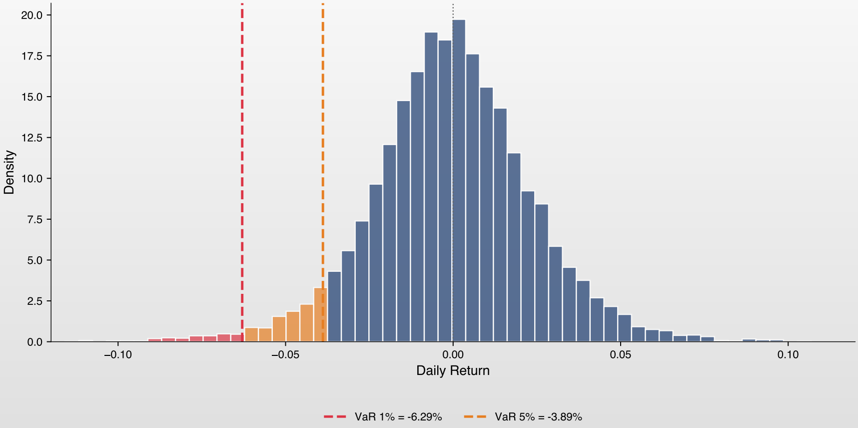
- Option pricing
- Dynamic hedging
- Portfolio allocation
- Stress testing

VaR and ES: Graphical Illustration

Interpretation

- VaR 1% = loss exceeded only in 1% of cases
- Red area = extreme losses (beyond VaR)

VaR and ES: Graphical Illustration



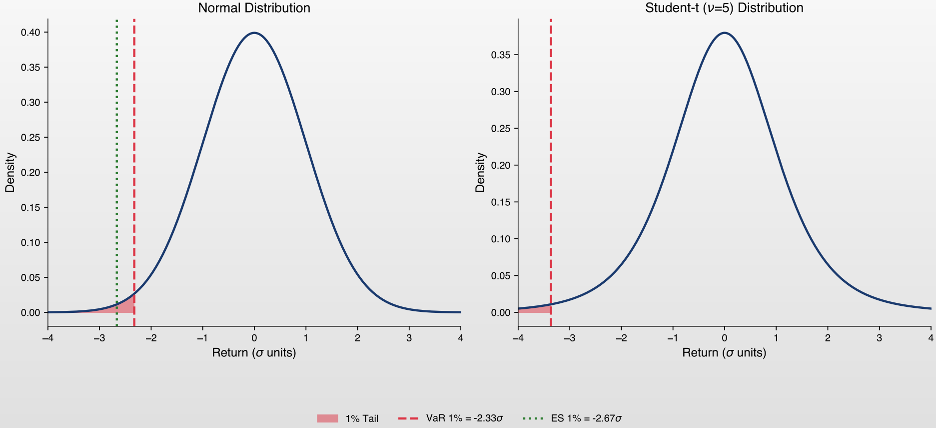
TSA_ch5_var_plot

VaR vs Expected Shortfall: Normal vs Student-t

Interpretation

- ES measures average loss when VaR is exceeded
- Student-t: VaR and ES are larger than under normal distribution

VaR vs Expected Shortfall: Normal vs Student-t



TSA_ch5_var_es

Value at Risk — Numerical Example

VaR Calculation

Portfolio: **1,000,000 EUR**, forecasted volatility $\hat{\sigma}_{T+1} = 1.5\%$

VaR with Normal Distribution

Level	z_α	**VaR (%)**	**VaR (EUR)**
5% (1 day)	1.645	2.47%	24,675
1% (1 day)	2.326	3.49%	34,890

Scaling for Longer Periods

$\text{VaR}_{h \text{ days}} = \text{VaR}_{1 \text{ day}} \cdot \sqrt{h}$ — assumes i.i.d. returns

Value at Risk — Student-t Distribution

Why Student-t?

Normal distribution **underestimates** tail risk. Student-t with ν degrees of freedom better captures fat tails (kurtosis > 3).

VaR 1% (1 day) Comparison: $\sigma = 1.5\%$, Portfolio $=$ 1M EUR

Distribution	Quantile	VaR (EUR)
Normal	2.326	34,890
Student-t ($\nu = 6$)	3.143	47,145
Student-t ($\nu = 4$)	3.747	56,205

Observation

With $\nu = 6$ (typical for stocks), VaR is **35% higher** than normal!

VaR — Complete Example with GARCH

VaR Calculation Procedure

1. Estimate GARCH(1,1) model with Student-t distribution
2. Obtain volatility forecast: $\hat{\sigma}_{T+1}$
3. Calculate VaR: $\text{VaR}_\alpha = t_\alpha(\nu) \cdot \hat{\sigma}_{T+1} \cdot \sqrt{\frac{\nu-2}{\nu}}$

Example: S&P 500

- Estimated parameters: $\alpha = 0.088$, $\beta = 0.900$, $\nu = 6.4$
- Forecasted volatility: $\hat{\sigma}_{T+1} = 1.2\%$
- Portfolio: 10,000,000 EUR

VaR 1% (1 day): $\text{VaR} = 3.05 \times 0.012 \times 10{,}000{,}000 = \mathbf{366{,}000\ EUR}$

What is VaR Backtesting?

Definition

- **Backtesting** = ex-post verification of VaR model quality
- Compares realized losses with the forecasted VaR threshold
  - A **violation** occurs when $r_t < -\text{VaR}_t$

Backtesting Principle

- Violation indicator: $I_t = 1(r_t < -\text{VaR}_{\alpha,t})$
- For a correctly specified model at level α:
  - Frequency: $\hat{p} = \frac{1}{T}\sum I_t \approx \alpha$; violations **independent**
- VaR 1% over 250 days $\Rightarrow$ expect $\sim$2.5 violations/year

Importance

- Regulatory requirement under **Basel III/IV** for banks: backtesting is mandatory

VaR Backtesting: Basel Traffic Light

Basel III/IV Traffic Light Zones

Zone	Violations/250 days	Interpretation	Penalty
Green	0–4	Model acceptable	No penalty
Yellow	5–9	Needs investigation	Factor k increases
Red	≥ 10	Model inadequate	Maximum penalty

Practical Example

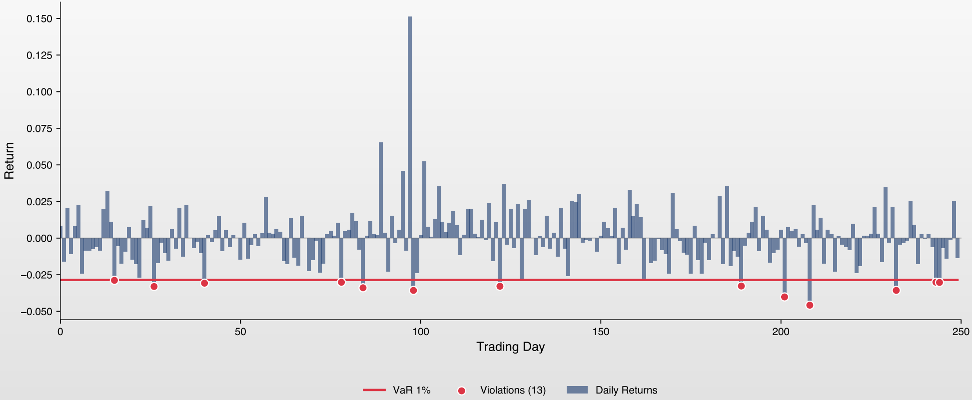
- Portfolio with VaR 1%: 250 days of backtesting
- 3 violations $\Rightarrow$ **Green zone** $\Rightarrow$ model acceptable
- 7 violations $\Rightarrow$ **Yellow zone** $\Rightarrow$ revision needed
- 13 violations $\Rightarrow$ **Red zone** $\Rightarrow$ model rejected

VaR Backtesting: Visual Overview

Interpretation

- Red line: VaR 1% threshold estimated with GARCH(1,1)
- Red dots: 13 violations out of 250 days ($\hat{p} = 5.2\%$)
  - **Basel red zone** $\Rightarrow$ model significantly underestimates risk
  - Solutions: Student-t distribution, EGARCH model, or more conservative VaR level

VaR Backtesting: Visual Overview



TSA_ch5_backtest

Rolling Window VaR Methodology

Rolling Window Concept

- A **rolling window** of fixed size W (e.g., 500 days) moves day by day
- At each step t: re-estimate GARCH on $[t - W,\ t - 1]$, forecast $\hat{\sigma}_{t|t-1}$, compute VaR_t

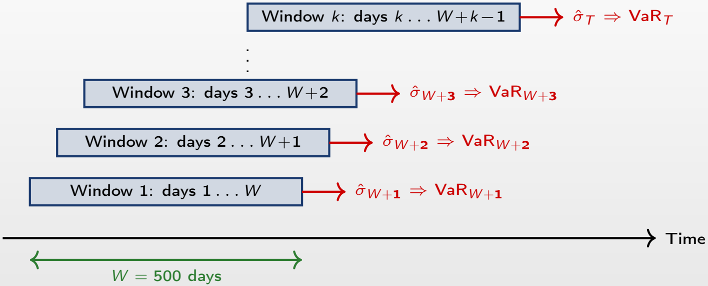
Step-by-Step Procedure (for each day $t = W + 1, \ldots, T$)

1. Estimate GARCH on $\{r_{t-W}, \ldots, r_{t-1}\} \Rightarrow$ parameters $\hat{\omega}, \hat{\alpha}, \hat{\beta}, \hat{\nu}$
2. Forecast: $\hat{\sigma}^2_{t|t-1} = \hat{\omega} + \hat{\alpha} r^2_{t-1} + \hat{\beta}\hat{\sigma}^2_{t-1}$
3. Compute: $\text{VaR}_{\alpha,t} = -t_\alpha(\hat{\nu}) \cdot \sqrt{\frac{\hat{\nu}-2}{\hat{\nu}}} \cdot \hat{\sigma}_{t|t-1}$
4. Check violation: $I_t = 1(r_t < -\text{VaR}_{\alpha,t})$

Why Rolling and Not Expanding?

- Fixed window: parameters reflect the **current regime** of volatility
- Old data ($> W$ days) may be irrelevant (structural changes, crises)

Rolling Window VaR: Procedure Diagram



Result

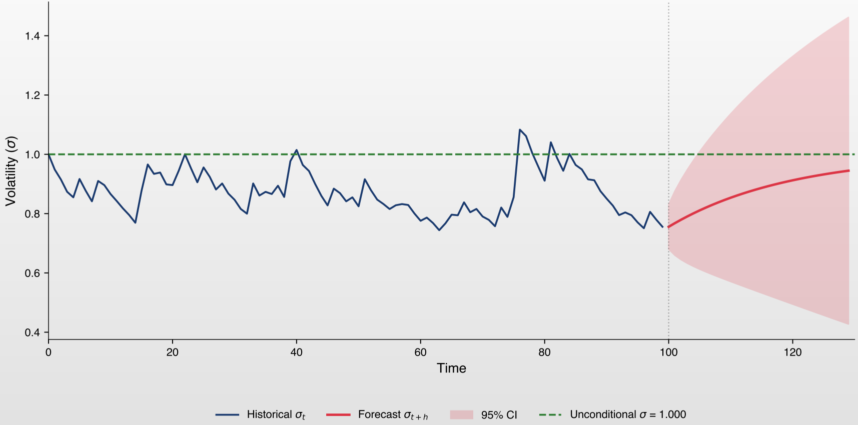
- We obtain the series $\{\text{VaR}_{\alpha,t}\}_{t=W+1}^{T} \Rightarrow$ a **different** threshold every day
- VaR adapts to the current regime: increases during volatile periods, decreases during calm ones
- Compare r_t with $-\text{VaR}_{\alpha,t}$ to identify violations

Volatility Forecast with Confidence Intervals

Interpretation

- Forecast converges to $\bar{\sigma}$
- Uncertainty increases with forecast horizon

Volatility Forecast with Confidence Intervals



TSA_ch5_vol_ci

Rolling Forecast: Step-by-Step Prediction

Procedure

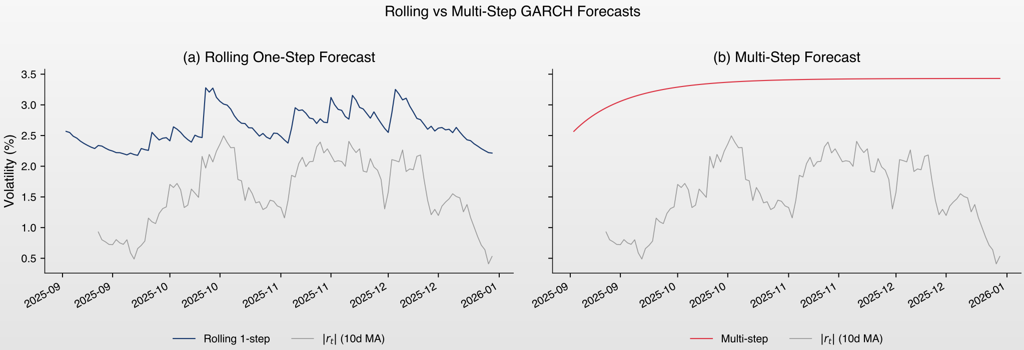
S&P 500, W=500, GARCH(1,1)-t

- Re-estimate GARCH on $[t-W,\ t-1]$; forecast $\hat{\sigma}_{t|t-1}$
- Compare with realized vol. (20-day rolling std.)

Results (2015 days OOS)

- $\rho = 0.938 \succ$ excellent tracking; MAE $= 0.15\%$, RMSE $= 0.24\%$
- COVID-19: temporary under-prediction, rapid adaptation

Rolling Forecast: Step-by-Step Prediction



TSA_ch5_rolling_forecast

GARCH Estimation in Python: arch Package

Python Code

```
pip install arch
from arch import arch_model

model = arch_model(returns,
    vol='Garch', p=1, q=1,
    dist='normal')
results = model.fit(disp='off')
print(results.summary())
```

Key Parameters

- **vol**: model type
  - 'Garch', 'EGARCH'
- **p, q**: GARCH order
  - p=1, q=1 standard
- **dist**: distribution
  - 'normal', 't'

TSA_ch5_garch

Asymmetric Models in Python

EGARCH and GJR-GARCH

```
# EGARCH
model_egarch = arch_model(returns, vol='EGARCH', p=1, q=1)
# GJR-GARCH (o=1 adds the asymmetric term)
model_gjr = arch_model(returns, vol='Garch', p=1, o=1, q=1)
```

Alternative Distributions

```
# Student-t for fat tails
model_t = arch_model(returns, vol='Garch', p=1, q=1, dist='t')
# Skewed Student-t for asymmetry and fat tails
model_skewt = arch_model(returns, vol='Garch', p=1, q=1,
                         dist='skewt')
```

TSA_ch5_egarch

Forecasting and Diagnostics

Volatility Forecast

```
forecasts = results.forecast(horizon=10)
vol_forecast = np.sqrt(forecasts.variance.values[-1,:])
```

Diagnostics and VaR

```
std_resid = results.std_resid
lb_test = acorr_ljungbox(std_resid**2, lags=10)
sigma = np.sqrt(forecasts.variance.values[-1, 0])
VaR_5pct = 1.645 * sigma
```

TSA_ch5_forecast

VaR Backtesting: Kupiec Test

Unconditional Coverage Test

Tests whether the observed violation rate equals the expected rate p (e.g., 1% for VaR 1%).

Let N = number of VaR violations, T = total observations, $\hat{p} = N/T$.

Likelihood Ratio Statistic:

$$LR_{uc} = -2\ln\left[\frac{(1-p)^{T-N}p^N}{(1-\hat{p})^{T-N}\hat{p}^N}\right] \sim \chi^2(1)$$

Hypotheses

- H_0: $\hat{p} = p$ (VaR model is correctly calibrated)
- H_1: $\hat{p} \neq p$ (VaR model under- or over-estimates risk)

VaR Backtesting: Christoffersen Test

Conditional Coverage Test

Tests both **unconditional coverage** and **independence** of violations.
Violations should be independent — no clustering of exceptions!

Test Components

- **Independence test** (LR_{ind}): Tests if violations are serially independent
- **Conditional coverage**: $LR_{cc} = LR_{uc} + LR_{ind} \sim \chi^2(2)$

Interpretation

Reject LR_{uc}: wrong frequency; Reject LR_{ind}: clustered violations; Reject LR_{cc}: model fails

VaR Backtesting: Rolling Window Python

Rolling VaR Estimation with GARCH(1,1)-t

```
from arch import arch_model
import numpy as np
from scipy import stats

returns = ...  # pd.Series with daily returns
window = 500; alpha = 0.01
VaR_series = pd.Series(index=returns.index[window:], dtype=float)

for t in range(window, len(returns)):
    train = returns.iloc[t-window:t]
    model = arch_model(train, vol='Garch', p=1, q=1, dist='t')
    res = model.fit(disp='off', show_warning=False)
    fcast = res.forecast(horizon=1, reindex=False)
    sigma = np.sqrt(fcast.variance.values[-1, 0])
    nu = res.params['nu']   # Student-t degrees of freedom
    q = stats.t.ppf(alpha, nu) * np.sqrt((nu-2)/nu)
    VaR_series.iloc[t - window] = -q * sigma
```

TSA_ch5_backtest_py

VaR Backtesting: Kupiec Test Python

Function kupiec_test()

```
def kupiec_test(violations, T, alpha=0.01):
    """Kupiec (1995) unconditional coverage LR test."""
    x = violations.sum()          # observed violations
    p_hat = x / T                 # empirical rate
    if x == 0 or x == T:
        return np.nan, np.nan
    LR_uc = -2 * (x*np.log(alpha) + (T-x)*np.log(1-alpha)
                  - x*np.log(p_hat) - (T-x)*np.log(1-p_hat))
    p_value = 1 - stats.chi2.cdf(LR_uc, df=1)
    return LR_uc, p_value
```

Usage

```
violations = returns[window:] < -VaR_series
LR, pval = kupiec_test(violations, len(violations))
print(f"Kupiec LR = {LR:.3f}, p-value = {pval:.4f}")
# p-value < 0.05 => reject H0 => inadequate model
```

VaR Backtesting: Christoffersen Test Python

Function christoffersen_test()

```
def christoffersen_test(violations):
    """Christoffersen (1998) conditional coverage test."""
    V = violations.astype(int).values
    # Transition matrix: n_ij
    n00 = ((V[:-1]==0) & (V[1:]==0)).sum()
    n01 = ((V[:-1]==0) & (V[1:]==1)).sum()
    n10 = ((V[:-1]==1) & (V[1:]==0)).sum()
    n11 = ((V[:-1]==1) & (V[1:]==1)).sum()
    # Conditional probabilities
    p01 = n01/(n00+n01) if (n00+n01)>0 else 0
    p11 = n11/(n10+n11) if (n10+n11)>0 else 0
    p = (n01+n11) / (n00+n01+n10+n11)  # global rate
    if p01==0 or p11==0 or p==0 or p==1: return np.nan, np.nan
    LR_ind = -2*(np.log(1-p)*(n00+n10) + np.log(p)*(n01+n11)
        - np.log(1-p01)*n00 - np.log(p01)*n01
        - np.log(1-p11)*n10 - np.log(p11)*n11)
    p_value = 1 - stats.chi2.cdf(LR_ind, df=1)
    return LR_ind, p_value
```

Full Backtesting: Results and Decision

Application S&P 500 (T=500, VaR 1%)

```
violations = returns[window:] < -VaR_series
n_viol = violations.sum()
T = len(violations)
rate = n_viol / T
print(f"Violations: {n_viol}/{T} (rate = {rate:.2%})")
LR_uc, p_uc = kupiec_test(violations, T, alpha=0.01)
LR_ind, p_ind = christoffersen_test(violations)
LR_cc = LR_uc + LR_ind   # combined test ~ chi2(2)
p_cc = 1 - stats.chi2.cdf(LR_cc, df=2)
```

Typical Output

```
Violations: 13/500 (rate = 2.60%)
Kupiec    LR = 5.83,  p-value = 0.0157  => Rejected (p<0.05)
Independ. LR = 0.42,  p-value = 0.5171  => Accepted
Combined  LR = 6.25,  p-value = 0.0439  => Rejected
Basel Zone: RED (>=10 violations) => Inadequate model
```

TSA_ch5_backtest_full

ARMA-GARCH: Joint Mean and Variance Modeling

Why Joint Modeling?

Serial correlation $\Rightarrow$ ARMA for mean; **Volatility clustering** $\Rightarrow$ GARCH for variance.

Definition 9 (ARMA(p,q)-GARCH(r,s))

Mean equation: $r_t = \mu + \sum_{i=1}^{p} \phi_i (r_{t-i} - \mu) + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$

Variance equation: $\varepsilon_t = \sigma_t z_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^{r} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2$

ARMA-GARCH: Model Selection Strategy

Step-by-Step Approach

1. **Identify mean model**: Check ACF/PACF of returns for ARMA structure
2. **Test for ARCH effects**: Apply ARCH-LM test to residuals
3. **Specify variance model**: Usually GARCH(1,1) is sufficient
4. **Joint estimation**: Estimate both equations via MLE
5. **Diagnostic checking**: Standardized residuals should be i.i.d.

Common Specifications

- **Stock returns**: AR(1)-GARCH(1,1) or ARMA(1,1)-GARCH(1,1)
- **Exchange rates**: Often just GARCH(1,1) (no mean dynamics)
- **Interest rates**: AR(1)-EGARCH(1,1) for leverage effects

ARMA-GARCH: Python Implementation

Using the arch Package

```
from arch import arch_model
model = arch_model(returns,
                   mean='ARX',
                   lags=1,
                   vol='Garch',
                   p=1, q=1,
                   dist='t')
result = model.fit(disp='off')
print(result.summary())
```

Parameters

`mean='ARX'`: ARMA mean; `lags=1`: AR(1); `dist='t'`: Student-t

TSA_ch5_armagarch

ARMA-GARCH: Complete Example

```
from arch import arch_model
model = arch_model(returns,
                   mean='ARX', lags=[1],
                   vol='EGARCH', p=1, q=1,
                   dist='skewt')
result = model.fit(update_freq=0)
cond_mean = result.conditional_mean
cond_vol = result.conditional_volatility
forecasts = result.forecast(horizon=5)
```

Note

For MA terms, use `mean='HARX'` or pre-filter with statsmodels ARIMA.

Step 1: Data — S&P 500 Daily Returns

Data Description

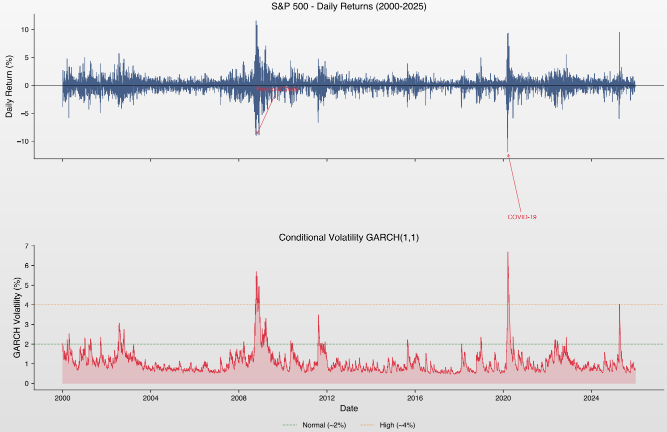
- **Source**: Yahoo Finance, S&P 500, daily data 2000–2024 ($T > 6000$)
- **Returns**: $r_t = \ln(P_t/P_{t-1}) \times 100$

Descriptive Statistics

Mean	Std. Dev.	Skewness	Kurtosis	Min	Max
0.034%	1.21%	-0.29	13.8	-12.8%	$+11.0\%$

- Fat tails (kurtosis $\gg 3$) and negative skewness $\Rightarrow$ ARCH effects

Step 1: Data — S&P 500 Daily Returns



TSA_ch5_sp500

Step 2: Testing for ARCH Effects

Python Code — ARCH-LM and Ljung-Box on r_t^2

```
import yfinance as yf
from statsmodels.stats.diagnostic import het_arch, acorr_ljungbox
sp = yf.download('^GSPC', start='2000-01-01', end='2024-12-31')
returns = np.log(sp['Close']).diff().dropna() * 100
# ARCH-LM test (Engle, 1982)
lm_stat, lm_pval, _, _ = het_arch(returns, nlags=10)
# Ljung-Box on squared returns
lb = acorr_ljungbox(returns**2, lags=20)
```

Results

Test	Statistic	p-value
ARCH-LM (10 lags)	892.4	< 0.0001
Ljung-Box r_t^2 (lag 20)	4217.6	< 0.0001

- **Conclusion**: Strong ARCH effects $\Rightarrow$ significant heteroskedasticity

Step 3: Estimating Multiple GARCH Models

Python Code — Estimating 4 candidate models

```
from arch import arch_model
models = {}
m1 = arch_model(returns, vol='Garch', p=1, q=1, dist='normal')
models['GARCH-N'] = m1.fit(disp='off')
m2 = arch_model(returns, vol='Garch', p=1, q=1, dist='t')
models['GARCH-t'] = m2.fit(disp='off')
m3 = arch_model(returns, vol='EGARCH', p=1, q=1, dist='t')
models['EGARCH-t'] = m3.fit(disp='off')
m4 = arch_model(returns, vol='GARCH', p=1, o=1, q=1, dist='t')
models['GJR-t'] = m4.fit(disp='off')
```

Strategy

- Estimate symmetric **and** asymmetric models, with different distributions
- Final choice: **AIC/BIC** + **diagnostics** + **economic interpretation**

Step 3: Estimated Parameters — Comparison

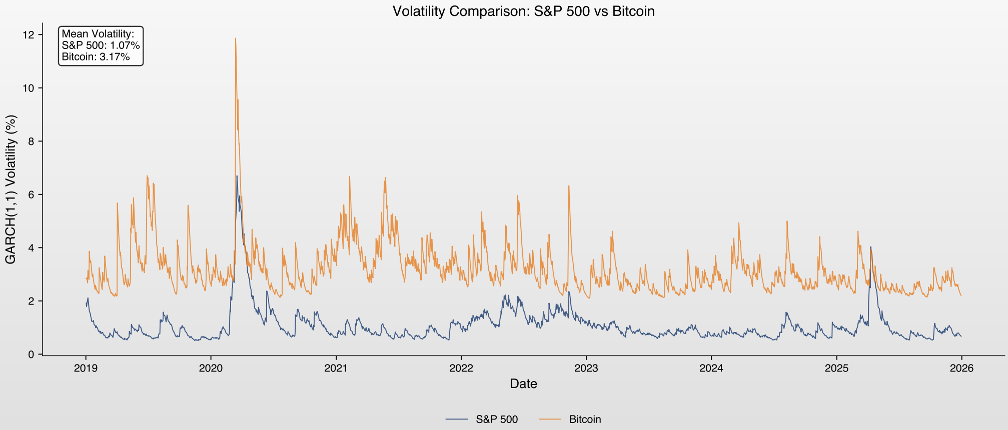
Estimated Parameters Table

Model	ω	α	β	γ	$\alpha+\beta$	ν	HL
GARCH-N	0.011	0.088	0.901	—	0.989	—	60 days
GARCH-t	0.011	0.088	0.900	—	0.989	6.42	60 days
EGARCH-t	0.003	0.103	0.987	−0.120	—	6.38	—
GJR-t	0.010	0.022	0.906	0.126	0.991	6.51	78 days

Interpretation

- EGARCH $\gamma = -0.12$ **significant** $\Rightarrow$ **leverage effect confirmed**
- GJR: $\alpha_{\text{neg}} = \alpha + \gamma = 0.148$ vs $\alpha_{\text{pos}} = 0.022 \Rightarrow$ strong asymmetry

Step 3: Estimated Parameters — Comparison



TSA_ch5_sp500_garch

Step 4: Model Selection — AIC/BIC

Information Criteria

Model	Log-Lik	AIC	BIC	Rank
GARCH(1,1)-N	−8042.3	16090.6	16111.0	4
GARCH(1,1)-t	−7981.5	15971.0	15997.8	3
EGARCH(1,1)-t	−7964.2	**15938.4**	**15971.6**	**1**
GJR-GARCH(1,1)-t	−7968.1	15946.2	15979.4	2

Decision

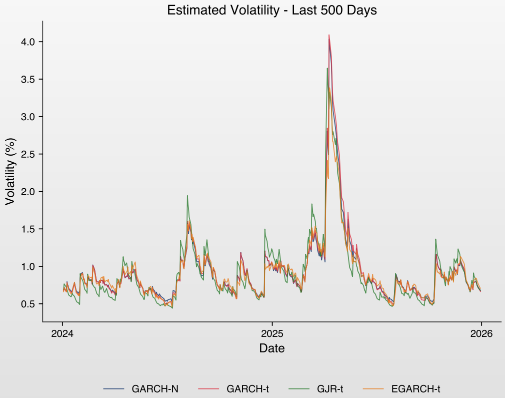
- **EGARCH(1,1)-t wins**: lowest AIC **and** BIC
- Student-t superior to Normal (ΔAIC $\approx$ 120) $\Rightarrow$ fat tails matter!
- Leverage effect justifies asymmetric models (ΔAIC $\approx$ 33 vs GARCH-t)

Step 5: Leverage Effect — Visualization

GARCH vs EGARCH — Volatility Differences

- EGARCH produces **higher** volatility after negative shocks (2008, 2020)
- Symmetric GARCH **underestimates** risk during crisis periods
- Difference: up to 2–3 percentage points in daily volatility

Step 5: Leverage Effect — Visualization



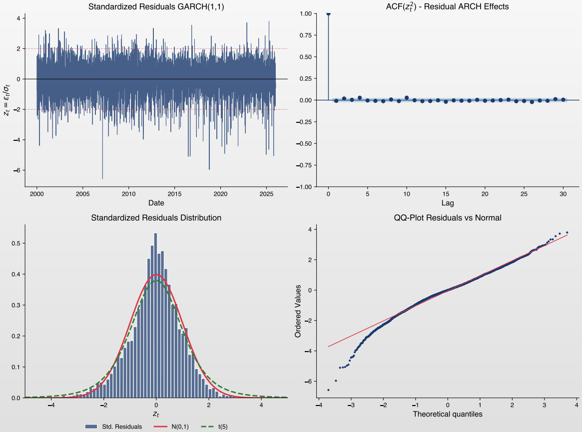
TSA_ch5_sp500_comp

Step 5: Diagnostics — EGARCH(1,1)-t

Checks on Standardized Residuals $z_t = \varepsilon_t / \hat{\sigma}_t$

- **Ljung-Box** on z_t: p-value $= 0.38$ — no residual autocorrelation
- **Ljung-Box** on z_t^2: p-value $= 0.52$ — **ARCH effects eliminated**
- **Q-Q plot**: points follow the theoretical Student-t line
- **Conclusion**: EGARCH(1,1)-t adequately captures volatility dynamics

Step 5: Diagnostics — EGARCH(1,1)-t



TSA_ch5_sp500_diag

Step 6: VaR Rolling Window — S&P 500

Rolling Window Procedure (W=500 days, VaR 1%)

```
from scipy import stats
W = 500; alpha = 0.01
VaR_roll = pd.Series(index=returns.index[W:], dtype=float)
for t in range(W, len(returns)):
    train = returns.iloc[t-W:t]
    m = arch_model(train, vol='Garch', p=1, q=1, dist='t')
    res = m.fit(disp='off', show_warning=False)
    fcast = res.forecast(horizon=1, reindex=False)
    sigma = np.sqrt(fcast.variance.values[-1, 0])
    nu = res.params['nu']
    VaR_roll.iloc[t-W] = -stats.t.ppf(alpha,nu)*np.sqrt((nu-2)/nu)*sigma
```

Rolling VaR Characteristics S&P 500 (2017–2024)

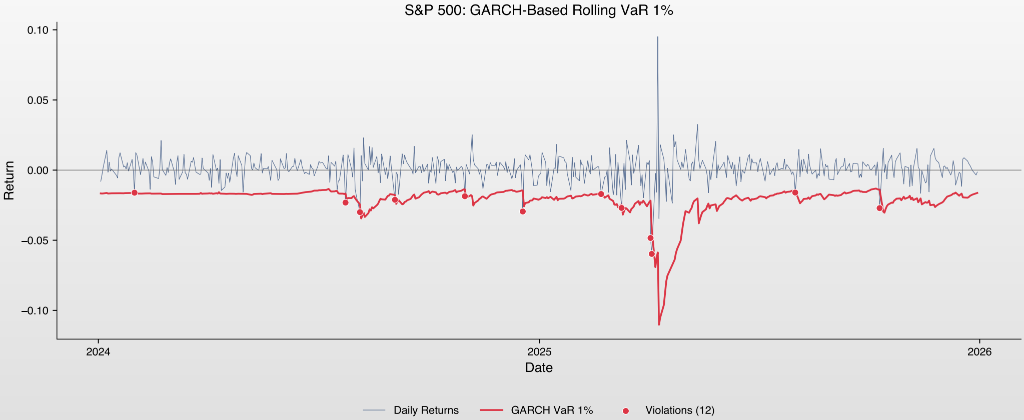
- Mean VaR: 2.53% ($\approx$ EUR 253,000 / 10M EUR)
- Max VaR: 22.02% $\Rightarrow$ COVID-19 crisis ($\approx$ EUR 2,202,000)
- Min VaR: 0.91% $\Rightarrow$ calm period ($\approx$ EUR 91,000)
- **VaR adapts**: automatically increases during crisis periods

Step 6: Backtesting Rolling VaR — S&P 500

Kupiec + Christoffersen Results (2015 days out-of-sample)

Test	Statistic	p-value	Decision
Violations	27/2015 ($\hat{p} = 1.34\%$)	—	**Green** zone
Kupiec (uc)	2.13	0.145	Accepted
Christoffersen (ind)	0.79	0.375	Accepted
Combined (cc)	2.91	0.233	Accepted

Step 6: Backtesting Rolling VaR — S&P 500



TSA_ch5_sp500_var

Step 7: Conclusions — S&P 500 Case Study

Step-by-Step Methodology Summary

1. **Data**: log returns, descriptive statistics $\Rightarrow$ fat tails, skewness
2. **ARCH test**: ARCH-LM + Ljung-Box on r_t^2 $\Rightarrow$ significant ARCH effects
3. **Estimation**: 4 candidate models (symmetric/asymmetric $\times$ Normal/Student-t)
4. **Selection**: AIC/BIC $\Rightarrow$ **EGARCH(1,1)-t** winner
5. **Diagnostics**: standardized residuals $\Rightarrow$ model adequate
6. **VaR**: rolling window + Kupiec/Christoffersen backtesting $\Rightarrow$ model **validated**

Key Lessons

- Student-t distribution is **essential** for financial data
- Leverage effect: asymmetric models **mandatory** for equities
- Systematic backtesting: not just "looks good", but **statistically tested**

Step 1: Data — Bitcoin Daily Returns

Data Description

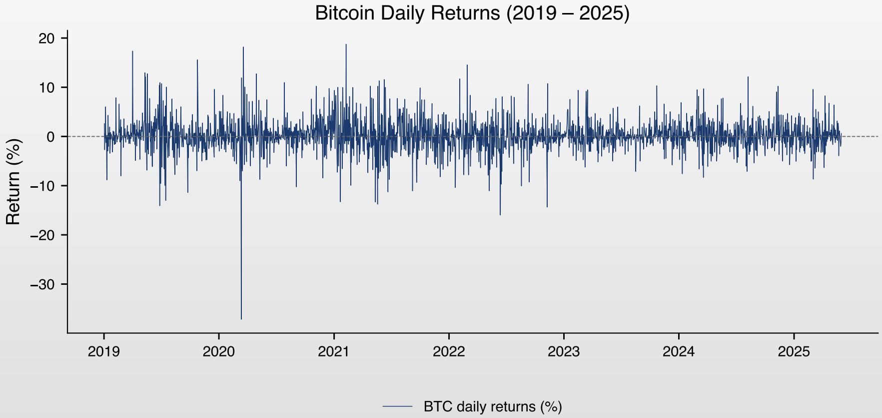
- **Source**: Yahoo Finance (BTC-USD), daily data 2018–2024
- Log returns: mean $\approx 0.05\%$, volatility $\approx 3.5\%$

Descriptive Statistics

Mean	Std. Dev.	Skewness	Kurtosis	Min	Max
0.05%	3.48%	−0.72	12.1	−46.5%	+22.5%

- Volatility $\sim 3\times$ higher than S&P 500
- Extreme kurtosis — high risk of large losses

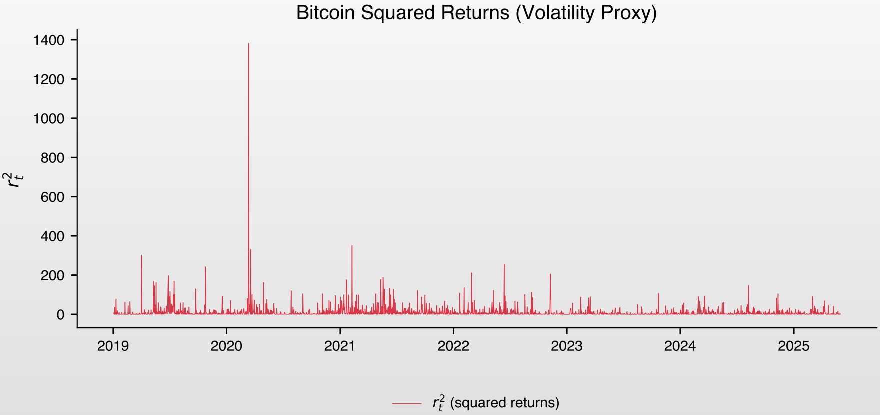
Step 1: Data — Bitcoin Daily Returns



TSA_ch5_btc

Step 2: Testing for ARCH Effects — Bitcoin

Step 2: Testing for ARCH Effects — Bitcoin



TSA_ch5_btc_arch

Steps 3–4: Estimation and Model Selection — Bitcoin

Estimated Parameters

Model	ω	α	β	γ	$\alpha+\beta$	ν	AIC
GARCH-t	0.42	0.131	0.848	—	0.979	4.82	9284
EGARCH-t	0.08	0.184	0.976	-0.061	—	4.79	9276
GJR-t	0.40	0.088	0.854	0.078	0.976	4.85	**9271**

Interpretation

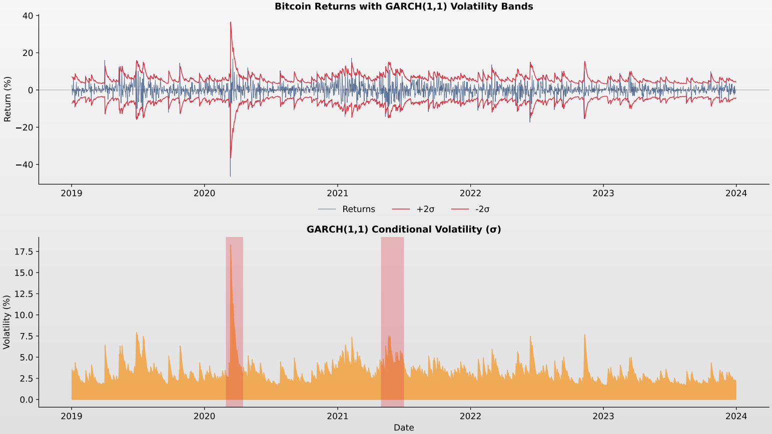
- **GJR-GARCH-t wins** (lowest AIC)
- $\nu \approx 4.8$: **much heavier tails** than S&P 500 ($\nu = 6.4$)
- $\alpha = 0.131$ (BTC) vs 0.088 (S&P) — Bitcoin reacts faster to news
- Leverage effect weaker than for stocks ($\gamma_{\text{BTC}} = 0.078$ vs 0.126)

Step 5: Conditional Volatility — Bitcoin

GJR-GARCH(1,1)-t Diagnostics

- Ljung-Box on z_t^2: p-value $= 0.41$ — **ARCH effects eliminated**
- Volatility peaks: March 2020 (COVID), May 2022 (Terra/Luna)
- Daily volatility: from 1% (calm periods) to $>$15% (crises)

Step 5: Conditional Volatility — Bitcoin



TSA_ch5_btc_garch

Step 6: VaR Rolling Window — Bitcoin

Rolling Window GJR-GARCH-t (W=500 days, VaR 1%)

```
W = 500; alpha = 0.01
VaR_btc = pd.Series(index=btc_returns.index[W:], dtype=float)
for t in range(W, len(btc_returns)):
    train = btc_returns.iloc[t-W:t]
    m = arch_model(train, vol='GARCH', p=1, o=1, q=1, dist='t')
    res = m.fit(disp='off', show_warning=False)
    fcast = res.forecast(horizon=1, reindex=False)
    sigma = np.sqrt(fcast.variance.values[-1, 0])
    nu = res.params['nu']
    VaR_btc.iloc[t-W] = -stats.t.ppf(alpha,nu)*np.sqrt((nu-2)/nu)*sigma
```

Rolling VaR Characteristics Bitcoin (2018–2024)

- Mean VaR: 9.34% ($\approx$ EUR 93,400 / 1M EUR)
- Max VaR: 37.54% $\Rightarrow$ COVID crash March 2020
- Min VaR: 2.90% $\Rightarrow$ calm period
- Bitcoin: rolling VaR $\sim 4\times$ larger than S&P 500 at same exposure

Step

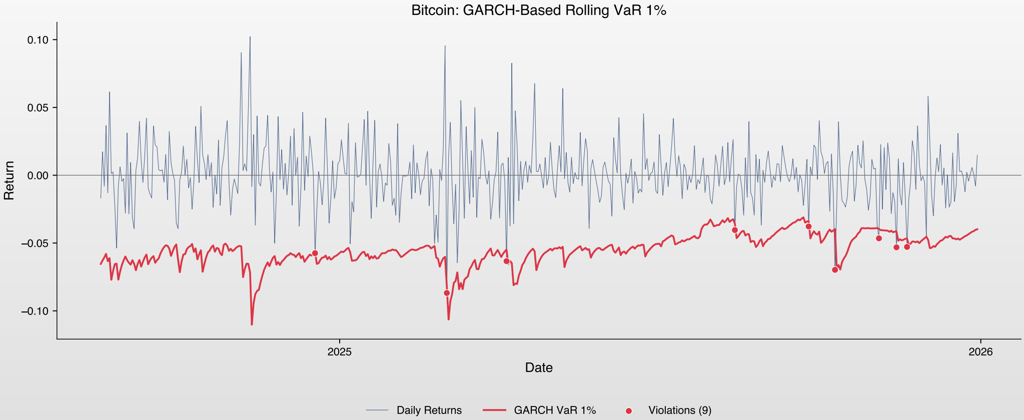
Statistical Tests (2421 days out-of-sample)

Test	Statistic	p-value	Decision
Violations	28/2421 ($\hat{p} = 1.16\%$)	—	Green zone
Kupiec (uc)	0.57	0.450	Accepted
Christoffersen (ind)	0.94	0.333	Accepted
Combined (cc)	1.51	0.471	Accepted

Interpretation

- Volatility ranges from 3% to 38% — rolling window is **essential**
- All tests **accepted**: model valid for risk management

Step 6: Backtesting Rolling VaR — Bitcoin



TSA_ch5_btc_var

Final Comparison: S&P 500 vs Bitcoin

Comparative Summary

	S&P 500	Bitcoin
Average volatility	1.2%	3.5%
Kurtosis	13.8	12.1
Student-t ν	6.42	4.82
Best model	EGARCH(1,1)-t	GJR-GARCH(1,1)-t
Leverage effect	Strong ($\gamma = -0.12$)	Moderate ($\gamma = 0.078$)
Half-life	∼60 days	∼42 days
Rolling VaR 1% mean	2.53%	9.34%
Rolling VaR 1% max	22.02% (COVID)	37.54% (COVID)
Kupiec	Accepted (p=0.145)	Accepted (p=0.450)
Christoffersen (ind)	Accepted (p=0.375)	Accepted (p=0.333)

General Conclusion

- Re-estimating GARCH at each step: Kupiec + Christoffersen **accepted**
- Rolling window VaR: **mandatory** — static VaR is completely inadequate
- Student-t + asymmetric model: **essential** for both markets

Key Formulas

Volatility Models

- **ARCH(q):** $\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$
- **GARCH(1,1):** $\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$
- **EGARCH:** $\ln(\sigma_t^2) = \omega + \alpha(|z_{t-1}| - \mathbb{E}[|z|]) + \gamma z_{t-1} + \beta \ln(\sigma_{t-1}^2)$
- **GJR-GARCH:** $\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \gamma\varepsilon_{t-1}^2 I_{t-1} + \beta\sigma_{t-1}^2$

Properties and Measures

- **Unconditional variance:** $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$ **Half-life:** $HL = \frac{\ln(0.5)}{\ln(\alpha+\beta)}$
- **VaR:** $\text{VaR}_\alpha = z_\alpha \cdot \sigma_{T+1}$ **Stationarity:** $\alpha + \beta < 1$
- **ARCH-LM:** $LM = T \cdot R^2 \sim \chi^2(q)$

AI Exercise: Critical Thinking

Prompt to test in ChatGPT / Claude / Copilot

"Download daily S&P 500 returns using yfinance. Test for ARCH effects, fit a GARCH model, and forecast volatility for the next 20 trading days. Give me complete Python code."

Exercise:

1. Run the prompt in an LLM of your choice and critically analyze the response.
2. Does it compute log returns correctly? Does it remove the mean before fitting GARCH?
3. How does it test for ARCH effects? Does it use Engle's LM test?
4. Does it separate the mean equation from the variance equation?
5. Does it discuss asymmetric effects (GJR-GARCH, EGARCH)? Are VaR estimates computed from the conditional distribution?

Warning: AI-generated code may run without errors and look professional. *That does not mean it is correct.*

Summary → Chapter 5: Volatility Models

Key Concepts

- **ARCH(q)**: conditional variance depends on past squared errors (Nobel 2003)
- **GARCH(p,q)**: adds variance lags for persistence (GARCH(1,1) in 90% of cases)
- **EGARCH**: allows leverage effect, no positivity constraints
- **GJR-GARCH/TGARCH**: captures asymmetry with indicator variables

Applications

- Risk measurement and forecasting (VaR, ES)
- Derivative pricing, dynamic hedging, portfolio management

Practical Tip

- Start with GARCH(1,1), check for leverage, choose distribution minimizing AIC/BIC!
  - Student-t often superior to normal distribution

Quiz Question 1

Question

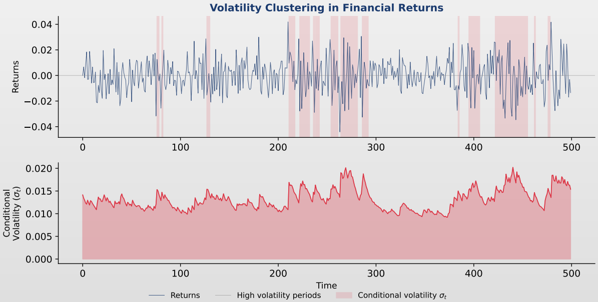
What best describes the phenomenon of *volatility clustering* in financial series?

(A) Financial returns are normally distributed and independent
(B) Periods of high volatility are followed by high volatility, and vice versa
(C) Volatility is constant over time (homoscedasticity)
(D) Correlation between returns is always positive

Quiz Question 1: Answer

Correct Answer: (B) Periods of high volatility are followed by similar periods

Volatility clustering is a fundamental stylized fact of financial series. It implies that conditional variance is **predictable**, motivating ARCH/GARCH models.



TSA_ch5_quiz1_volatility_clustering

Quiz Question 2

Question

What is the main difference between an ARCH(q) and a GARCH(p,q) model?

(A) GARCH models the conditional mean, ARCH models the variance

(B) ARCH includes lags of conditional variance, GARCH does not

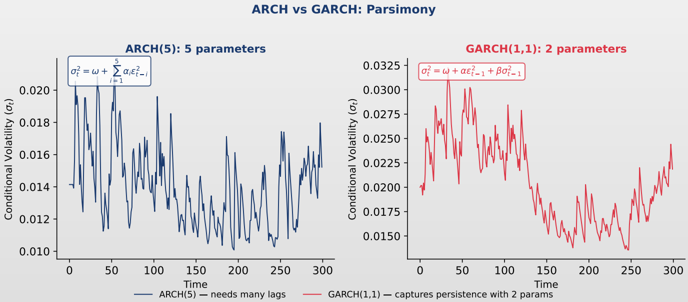
(C) GARCH adds lags of conditional variance (σ^2_{t-j}) beyond squared errors

(D) ARCH is more parsimonious than GARCH

Quiz Question 2: Answer

Correct Answer: (C) GARCH adds lags of conditional variance

GARCH(1,1) captures the same persistence as ARCH(q) with only 2 parameters instead of q. In practice, GARCH(1,1) is sufficient in 90% of cases.



TSA_ch5_quiz2_arch_vs_garch

Quiz Question 3

Question

What is the *leverage effect* and which GARCH models capture it?

(A) Positive shocks increase volatility more; captured by standard GARCH

(B) Negative shocks increase volatility more; captured by EGARCH and GJR-GARCH

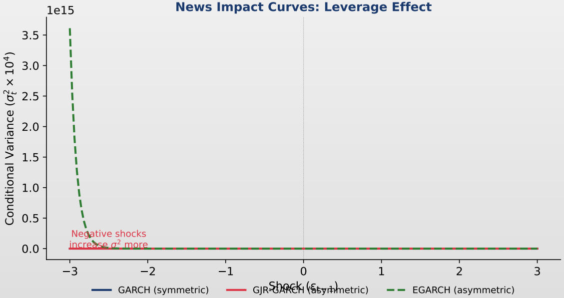
(C) Volatility is symmetric; captured by all GARCH models

(D) Financial leverage effect on stock prices; captured by IGARCH

Quiz Question 3: Answer

Correct Answer: (B) Negative shocks increase volatility more; EGARCH and GJR-GARCH

Price drops increase volatility **more** than equivalent price increases. Standard GARCH uses ε_{t-1}^2, losing the sign information.



TSA_ch5_quiz3_leverage_effect

Quiz Question 4

Question

What is the stationarity condition for a GARCH(1,1) model?

(A) $\alpha + \beta = 1$

(B) $\alpha > 0$ and $\beta > 0$

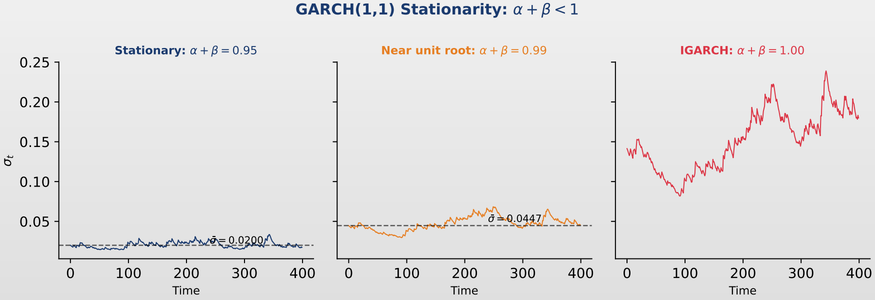
(C) $\alpha + \beta < 1$, with $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$

(D) $\alpha \cdot \beta < 1$

Quiz Question 4: Answer

Correct Answer: (C) $\alpha + \beta < 1$, with $\omega > 0$, $\alpha \geq 0$, $\beta \geq 0$

This ensures a finite unconditional variance $\bar{\sigma}^2 = \frac{\omega}{1-\alpha-\beta}$. When $\alpha + \beta = 1$ (IGARCH), variance is infinite.



TSA_ch5_quiz4_stationarity

Quiz Question 5

Question

What does the *half-life* of volatility represent in a GARCH(1,1) model?

(A) The time needed for the price to return to its mean

(B) The number of periods until volatility becomes zero

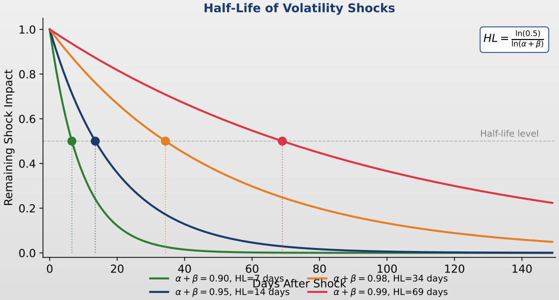
(C) The number of periods for a volatility shock to decay to half its initial impact

(D) The average duration of a high-volatility episode

Quiz Question 5: Answer

Correct Answer: (C) Number of periods for a shock to decay to half

$HL = \frac{\ln(0.5)}{\ln(\alpha+\beta)}$. Example: S&P 500 with $\alpha + \beta = 0.988$ gives $HL \approx 58$ days (shocks persist $\sim$3 months).



TSA_ch5_quiz5_halflife

References I

Fundamental ARCH/GARCH Papers

- Engle, R.F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of UK Inflation, *Econometrica*, 50(4), 987–1007.
- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, 31(3), 307–327.
- Nelson, D.B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach, *Econometrica*, 59(2), 347–370.

Asymmetric Models and Extensions

- Glosten, L.R., Jagannathan, R., & Runkle, D.E. (1993). On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks, *Journal of Finance*, 48(5), 1779–1801.
- Francq, C., & Zakoïan, J.-M. (2019). *GARCH Models: Structure, Statistical Inference and Financial Applications*, 2nd ed., Wiley.

References II

Textbooks and Financial Applications

- Tsay, R.S. (2010). *Analysis of Financial Time Series*, 3rd ed., Wiley.
- Franke, J., Härdle, W.K., & Hafner, C.M. (2019). *Statistics of Financial Markets*, 4th ed., Springer.
- McNeil, A.J., Frey, R., & Embrechts, P. (2015). *Quantitative Risk Management*, 2nd ed., Princeton University Press.

Online Resources and Code

- **Quantlet**: `https://quantlet.com` → Code repository for statistics
- **Quantinar**: `https://quantinar.com` → Learning platform for quantitative methods
- **GitHub TSA**: `https://github.com/QuantLet/TSA` → Python code for this course

Thank You!

Questions?

Course materials available at: https://danpele.github.io/Time-Series-Analysis/

Quantlet   Quantinar