

Lecture 9, November 3, 2025

(G)ARCH Models

Marta Boczon

Department of Economics

Copenhagen Business School

mbo.eco@cbs.dk

```
In [31]: #install.packages("quantmod")
#install.packages("fredr")
#install.packages("ggfortify")
#install.packages('urca')
#install.packages("tseries")
#install.packages("forecast")
#install.packages("dynlm")
#install.packages("stargazer")
#install.packages("pracma")
#install.packages("dLagM")
#install.packages("gets")
#install.packages("car")
#install.packages("lmtest")
#install.packages("vars")
#install.packages("tseries")
#install.packages("strucchange")
#install.packages("graphics")
#install.packages("grDevices")
#install.packages("tsDyn")
#install.packages('fGarch')
#install.packages('aTSA')
options(warn=-1)
```

Roadmap

Today we will learn about GARCH models.

Goals for today:

- Test for ARCH effects.
- Estimate AR-GARCH models.

Expected outcomes:

- Understand the difference between conditional and unconditional moments.
- Interpret plots of squared series.
- Interpret ACF plots of squared series.
- Understand why squared residuals are more reliable to investigate than the series in levels.
- Understand the mechanics behind the Engle LM-ARCH test.
- Learn how to estimate an ARMA-GARCH(1,1) model.
- Learn how to forecast using an ARMA-GARCH(1,1) model.

We will practice together using returns on Apple stock. To reinforce each new concept, we will conduct two in-class exercises using returns on BMW stock.

Conditional and Unconditional Moments

Let's start with a definition of volatility.

What is volatility?

Volatility is a measure of the variability of a time series (often a stock return) over a certain period of time. This is often measured by the standard deviation (or variance), but these are not identical. Both terms (variance and volatility) are often understood in financial circles as measures (or approximations) of risk.

*"In conventional econometric models, the variance of the disturbance term is assumed to be constant. However, as shown above, many economic time series exhibit periods of unusually large **volatility** followed by periods of relative **tranquility**. In such circumstances, the assumption of a constant variance (homoskedasticity) is inappropriate."* (Enders, p. 123)

Volatility clusters refer to “bouts” of higher or lower volatility y_t and imply that a time series is conditionally heteroskedastic.

The first step in understanding ARCH and GARCH models is to understand the difference between conditional and unconditional moments. In short:

Unconditional moments describe the overall properties of a variable—its mean or variance on average, without considering any information from the past.

Conditional moments describe how those properties change depending on information available at a given time—typically, past values of the variable. Let us start with mean and variance.

Can the Plot of the Series Tell Us Whether the Unconditional Variance Is Constant?

The plot of the **original series** (e.g., returns) provides only a *visual impression* of volatility stability. However, what we observe in the plot is actually a **mix of conditional and unconditional variance behavior**.

- If the amplitude of fluctuations looks roughly similar over time, this is **consistent** with a constant unconditional variance — the long-run average volatility does not appear to change.
- If there are visible clusters of large and small movements (periods of turbulence followed by calm), this suggests that the **conditional variance** varies over time — even though the **unconditional variance** (its long-run average) may remain constant.

Why visual inspection is limited: The unconditional variance is a long-run, theoretical property of a stationary process, defined as $E[(r_t - \mu)^2]$. The plot of a finite sample reflects both short-run conditional dynamics (temporary bursts of volatility) and this long-run average, so it cannot disentangle the two visually.

Conclusion: The plot of the series shows a combination of *conditional* and *unconditional* variance patterns. It can suggest volatility clustering or stability, but cannot establish whether the *unconditional variance* is constant. Formal modeling (e.g., ARCH-LM test or GARCH estimation) is needed for that distinction.

Can the Plot of the Series Tell Us Whether the Mean Is Constant?

The plot of a time series can show fluctuations that look like the mean is changing over time. However, these apparent shifts may reflect changes in the **conditional mean** rather than a true change in the **unconditional mean**.

- **Unconditional mean:** The long-run expected value of the process, $E(y_t)$, which remains constant for stationary series.
- **Conditional mean:** The expected value of y_t given past information, $E(y_t | \Omega_{t-1})$, which can change over time as past values evolve.

For example, in an AR(1) model $y_t = 0.8y_{t-1} + \epsilon_t$, the conditional mean $E(y_t | y_{t-1}) = 0.8y_{t-1}$ varies with time, but the unconditional mean $E(y_t) = 0$ is constant. The series may appear to drift around zero for long periods, giving the illusion of a changing mean even though it is stationary.

Conclusion: The plot of the series can hint at mean stability, but apparent movements in the mean often reflect *conditional dynamics* rather than non-stationarity. A series can have a constant *unconditional mean* and still show time-varying *conditional means* due to autoregressive structure or other short-run dependencies.

AR(1) Process With Constant Conditional Variance

Mean of AR(1) Process

Model: $y_t = a_0 + a_1 y_{t-1} + \epsilon_t$

Properties of ϵ_t

1. **White noise:** $E(\epsilon_t) = 0, \text{Var}(\epsilon_t) = \sigma^2$.
2. **Independence:** ϵ_t is independent over time and of Ω_{t-1} , so

$$E(\epsilon_t | \Omega_{t-1}) = E(\epsilon_t) = 0.$$

Unconditional Mean

$E(y_t) = E(a_0 + a_1 y_{t-1} + \epsilon_t) = a_0 + a_1 E(y_{t-1})$. Since $E(y_t) = E(y_{t-1})$, we get:

$$E(y_t) = \frac{a_0}{1 - a_1}, \quad |a_1| < 1.$$

Conditional Mean

$$E(y_t | \Omega_{t-1}) = E(a_0 + a_1 y_{t-1} + \epsilon_t | \Omega_{t-1}) = a_0 + a_1 y_{t-1}.$$

Variance of AR(1) Process

Model: $y_t = a_0 + a_1 y_{t-1} + \epsilon_t$

Key Properties of ϵ_t

1. **White noise:** $\mathbb{E}(\epsilon_t) = 0, \text{Var}(\epsilon_t) = \sigma^2$.
2. **Independence:** ϵ_t independent of Ω_{t-1} , so

$$\mathbb{E}(\epsilon_t | \Omega_{t-1}) = 0.$$

Unconditional Variance

$$\text{Var}(y_t) = \text{Var}(a_0 + a_1 y_{t-1} + \epsilon_t) = a_1^2 \text{Var}(y_{t-1}) + \sigma^2.$$

For a covariance-stationary process, $\text{Var}(y_t) = \text{Var}(y_{t-1}) = \gamma_0$, so:

$$\gamma_0 = a_1^2 \gamma_0 + \sigma^2 \Rightarrow \gamma_0 = \frac{\sigma^2}{1 - a_1^2}, \quad |a_1| < 1.$$

$$\text{Var}(y_t) = \frac{\sigma^2}{1 - a_1^2}.$$

Conditional Variance

Given Ω_{t-1} , a_0 and $a_1 y_{t-1}$ are known, so variability comes only from ϵ_t :

$$\text{Var}(y_t | \Omega_{t-1}) = \text{Var}(\epsilon_t | \Omega_{t-1}) = \sigma^2.$$

$$\text{Var}(y_t | \Omega_{t-1}) = \sigma^2.$$

Summary

- **Unconditional variance:** $\text{Var}(y_t) = \frac{\sigma^2}{1 - a_1^2}$ (requires $|a_1| < 1$).
- **Conditional variance:** $\text{Var}(y_t | \Omega_{t-1}) = \sigma^2$ (constant, by assumption).

AR(1) Process With Varying Conditional Variance Modeled by ARCH(1)

Idea: Let us allow the variance of ϵ_t to vary when conditioning on the information set Ω_{t-1} . Specifically, we model ϵ_t as a *multiplicative conditionally heteroskedastic process* given by:

$$y_t = a_0 + a_1 y_{t-1} + \epsilon_t$$

$$\epsilon_t = v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2},$$

where v_t is standardized white noise with $\text{Var}(v_t) = 1$.

Key Properties of ϵ_t

1. **White Noise:** By definition, ϵ_t has $\mathbb{E}(\epsilon_t) = 0$ and $\text{Var}(\epsilon_t) = \sigma^2$.

Unconditional Variance

Step 1:

$$\text{Var}(y_t) = \text{Var}(a_0 + a_1 y_{t-1} + \epsilon_t) = a_1^2 \text{Var}(y_{t-1}) + \underbrace{\text{Var}(\epsilon_t) + 2a_1 \text{Cov}(y_{t-1}, \epsilon_t)}_{=0}.$$

Hence,

$$\text{Var}(y_t) = \frac{\text{Var}(\epsilon_t)}{1 - a_1^2}.$$

Step 2:

$$\text{Var}(\epsilon_t) = \mathbb{E}(\epsilon_t^2) - [\mathbb{E}(\epsilon_t)]^2 = \mathbb{E}(\epsilon_t^2).$$

Step 3:

$$\mathbb{E}(\epsilon_t^2) = \mathbb{E}(v_t^2) \times \mathbb{E}(\omega_0 + \alpha_1 \epsilon_{t-1}^2) = \omega_0 + \alpha_1 \mathbb{E}(\epsilon_{t-1}^2).$$

Solving for the steady-state value gives:

$$\mathbb{E}(\epsilon_t^2) = \frac{\omega_0}{1 - \alpha_1}.$$

Hence,

$$\text{Var}(y_t) = \frac{\omega_0}{(1 - a_1^2)(1 - \alpha_1)}.$$

Conditional Variance

Step 1:

$$y_t = a_0 + a_1 y_{t-1} + \epsilon_t$$

Thus,

$$\text{Var}(y_t | \Omega_{t-1}) = \text{Var}(a_0 + a_1 y_{t-1} + \epsilon_t | \Omega_{t-1}) = \text{Var}(\epsilon_t | \Omega_{t-1}).$$

Step 2:

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \mathbb{E}(\epsilon_t^2 | \Omega_{t-1}) - [\mathbb{E}(\epsilon_t | \Omega_{t-1})]^2.$$

Using $\epsilon_t = v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2}$, we get:

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \mathbb{E}(v_t^2 (\omega_0 + \alpha_1 \epsilon_{t-1}^2) | \Omega_{t-1}) - \mathbb{E}\left(v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2} | \Omega_{t-1}\right)^2.$$

Hence,

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \omega_0 + \alpha_1 \epsilon_{t-1}^2.$$

AR(2)-ARCH(2) Model

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t$$

$$\epsilon_t = v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2},$$

where v_t is standardized white noise with $\text{Var}(v_t) = 1$ and is independent of ϵ_t at all leads and lags.

Unconditional Variance

Step 1:

$$\begin{aligned} \text{Var}(y_t) &= \text{Var}(a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t) \\ &= a_1^2 \text{Var}(y_{t-1}) + a_2^2 \text{Var}(y_{t-2}) + \text{Var}(\epsilon_t) + 2a_1 \underbrace{\text{Cov}(y_{t-1}, \epsilon_t)}_{=0} + 2a_2 \underbrace{\text{Cov}(y_{t-2}, \epsilon_t)}_{=0} + 2a_1 a_2 \text{Cov}(y_{t-1}, y_{t-2}). \end{aligned}$$

Using $\text{Cov}(y_{t-1}, y_{t-2}) = \frac{a_1 \text{Var}(y_t)}{1 - a_2}$, we get:

$$\text{Var}(y_t) = \frac{\text{Var}(\epsilon_t)}{1 - a_1^2 - a_2^2 - \frac{2a_1^2 a_2}{1 - a_2}}.$$

Note: For an AR(2) process, $\text{Cov}(y_{t-1}, y_{t-2}) = \gamma_1$ and it can be shown that $\gamma_1 = a_1 \text{Var}(y_t) + a_2 \gamma_1 \Rightarrow \gamma_1 = \frac{a_1 \text{Var}(y_t)}{1 - a_2}$.

Step 2:

$$\text{Var}(\epsilon_t) = \mathbb{E}(\epsilon_t^2) - [\mathbb{E}(\epsilon_t)]^2 = \mathbb{E}(\epsilon_t^2).$$

Step 3:

$$\mathbb{E}(\epsilon_t^2) = \mathbb{E}(v_t^2) \times \mathbb{E}(\omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2) = \omega_0 + \alpha_1 \mathbb{E}(\epsilon_{t-1}^2) + \alpha_2 \mathbb{E}(\epsilon_{t-2}^2).$$

In steady state:

$$\mathbb{E}(\epsilon_t^2) = \frac{\omega_0}{1 - \alpha_1 - \alpha_2}.$$

Hence,

$$\text{Var}(y_t) = \frac{\omega_0}{(1 - \alpha_1 - \alpha_2) \left(1 - a_1^2 - a_2^2 - \frac{2a_1^2 a_2}{1 - a_2}\right)}.$$

Conditional Variance

Step 1:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t.$$

Therefore,

$$\text{Var}(y_t | \Omega_{t-1}) = \text{Var}(\epsilon_t | \Omega_{t-1}).$$

Step 2:

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \mathbb{E}(\epsilon_t^2 | \Omega_{t-1}) - [\mathbb{E}(\epsilon_t | \Omega_{t-1})]^2.$$

Using $\epsilon_t = v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2}$, we have:

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \mathbb{E}(v_t^2 (\omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2) | \Omega_{t-1}) - \mathbb{E}\left(v_t \sqrt{\omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2} | \Omega_{t-1}\right)^2.$$

Hence,

$$\text{Var}(\epsilon_t | \Omega_{t-1}) = \omega_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2.$$

AR(p)-ARCH(k) Model

The AR(p)-ARCH(k) model combines an autoregressive process for the mean with an autoregressive conditional heteroskedastic process for the variance:

$$y_t = a_0 + \sum_{i=1}^p a_i y_{t-i} + \epsilon_t$$

$$\epsilon_t = v_t \sqrt{\omega_0 + \sum_{j=1}^k \alpha_j \epsilon_{t-j}^2},$$

where v_t is standardized white noise with $\text{Var}(v_t) = 1$ and is independent of ϵ_t at all leads and lags.

Stationarity Condition for ARCH Models

For an ARCH(k) model to have a well-defined (finite) unconditional variance, the sum of the α coefficients—representing the influence of past squared errors—must be less than 1:

$$\sum_{j=1}^k \alpha_j < 1.$$

If this condition holds, the unconditional (long-run) variance exists and is finite:

$$\mathbb{E}(\epsilon_t^2) = \frac{\omega_0}{1 - \sum_{j=1}^k \alpha_j}.$$

However, if $\sum_{j=1}^k \alpha_j \geq 1$, the unconditional variance becomes infinite—implying that shocks to volatility persist indefinitely, and the process is non-stationary in variance.

AR(p)-GARCH(k) Models

When the number of lags in an ARCH model becomes too high, we can reformulate it into a more parsimonious structure that still captures **volatility clustering** effectively. This leads to the **Generalized ARCH (GARCH)** model:

$$\epsilon_t = v_t \sqrt{\omega_0 + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^p \beta_i \mathbb{E}(\epsilon_{t-i}^2 | \Omega_{t-i-1})}$$

where:

- v_t is standardized white noise with $\text{Var}(v_t) = 1$,
- α_i measures the impact of past squared shocks (ARCH terms),
- β_i measures the persistence of past conditional variances (GARCH terms).

Why Use GARCH Instead of High-Order ARCH?

In practice, only lower-order models — particularly **GARCH(1,1)** — are commonly used, as they capture volatility clustering remarkably well with very few parameters.

Interestingly, it can be shown that a **GARCH(1,1)** model is equivalent to an **ARCH($p \rightarrow \infty$)** model. This insight explains why we typically use GARCH when the order p in an ARCH(p) specification becomes large — it provides the same flexibility with far fewer parameters.

Empirical Example

In the following section, we will demonstrate how to test for ARCH effects and estimate ARCH/GARCH models. Before proceeding, we first need to download data on stock returns for Apple and BMW and verify that the series are stationary.

```
In [32]: # Load required libraries
library(quantmod)    # For financial data import and analysis
library(fGarch)      # For GARCH modeling
library(forecast)    # For time series forecasting

# Get stock price data for Apple (AAPL) from Yahoo Finance
# Reference: https://www.rdocumentation.org/packages/quantmod/versions/0.4.26/topics/getSymbols
apple = getSymbols("AAPL",
                   src = "yahoo",          # Data source
                   from = "2000-01-01",    # Start date
                   to = "2025-11-01",      # End date
                   auto.assign = FALSE)    # Return data directly instead of saving to environment

# Extract the closing price column
apple = apple$AAPL.Close

# Display the first few observations
head(apple)

# Display the last few observations
tail(apple)
```

	AAPL.Close
2000-01-03	0.999442
2000-01-04	0.915179
2000-01-05	0.928571
2000-01-06	0.848214
2000-01-07	0.888393
2000-01-10	0.872768
	AAPL.Close
2025-10-24	262.82
2025-10-27	268.81
2025-10-28	269.00
2025-10-29	269.70
2025-10-30	271.40
2025-10-31	270.37

```
In [33]: # Load required libraries
library(quantmod)    # For financial data import and analysis
library(fGarch)      # For GARCH modeling
library(forecast)    # For time series forecasting

# Get stock price data for BMW (BMW.DE) from Yahoo Finance
# Reference: https://www.rdocumentation.org/packages/quantmod/versions/0.4.26/topics/getSymbols
bmw = getSymbols("BMW.DE",
                  src = "yahoo",          # Data source
                  from = "2000-01-01",    # Start date
                  to = "2025-11-01",      # End date
                  auto.assign = FALSE)    # Return data directly instead of saving to environment

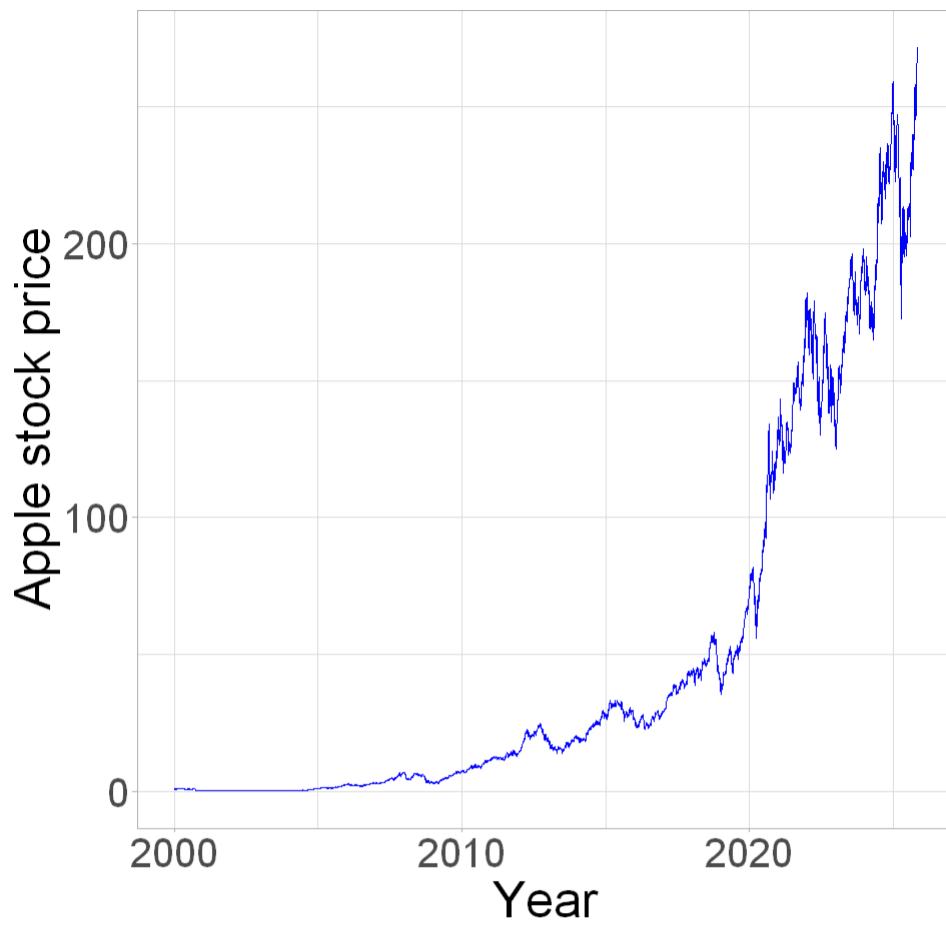
# Extract the closing price column
bmw = bmw$BMW.DE.Close

# Display the first few observations
head(bmw)

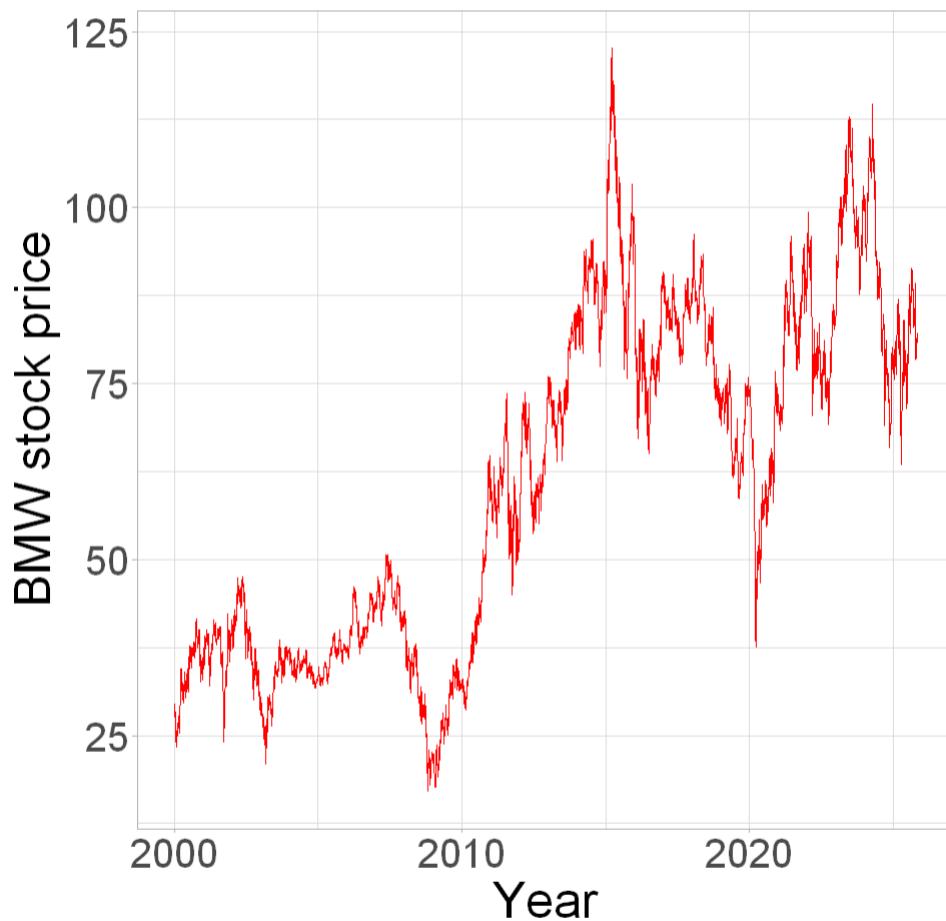
# Display the last few observations
tail(bmw)
```

	BMW.DE.Close
2000-01-03	29.49
2000-01-04	28.30
2000-01-05	27.74
2000-01-06	27.65
2000-01-07	27.60
2000-01-10	28.70
	BMW.DE.Close
2025-10-24	81.16
2025-10-27	81.52
2025-10-28	81.24
2025-10-29	82.10
2025-10-30	81.38
2025-10-31	80.72

```
In [34]: library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(apple, colour = 'blue')
fig = fig + theme(aspect.ratio=1) +
  theme_light() +
  theme(aspect.ratio=1) +
  theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
  theme(text=element_text(size=30)) +
  labs(x = "Year") +
  labs(y = "Apple stock price")
fig
```



```
In [35]: library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(bmw, colour = 'red')
fig = fig + theme(aspect.ratio=1) +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
theme(text=element_text(size=30)) +
labs(x = "Year") +
labs(y = "BMW stock price")
fig
```



```
In [36]: dapple = diff(log(apple)) # Compute Log returns (difference in log prices)
head(dapple) # Show first few log returns
tail(dapple) # Show last few log returns
dapple = na.omit(dapple) # Remove missing values (from first difference)
```

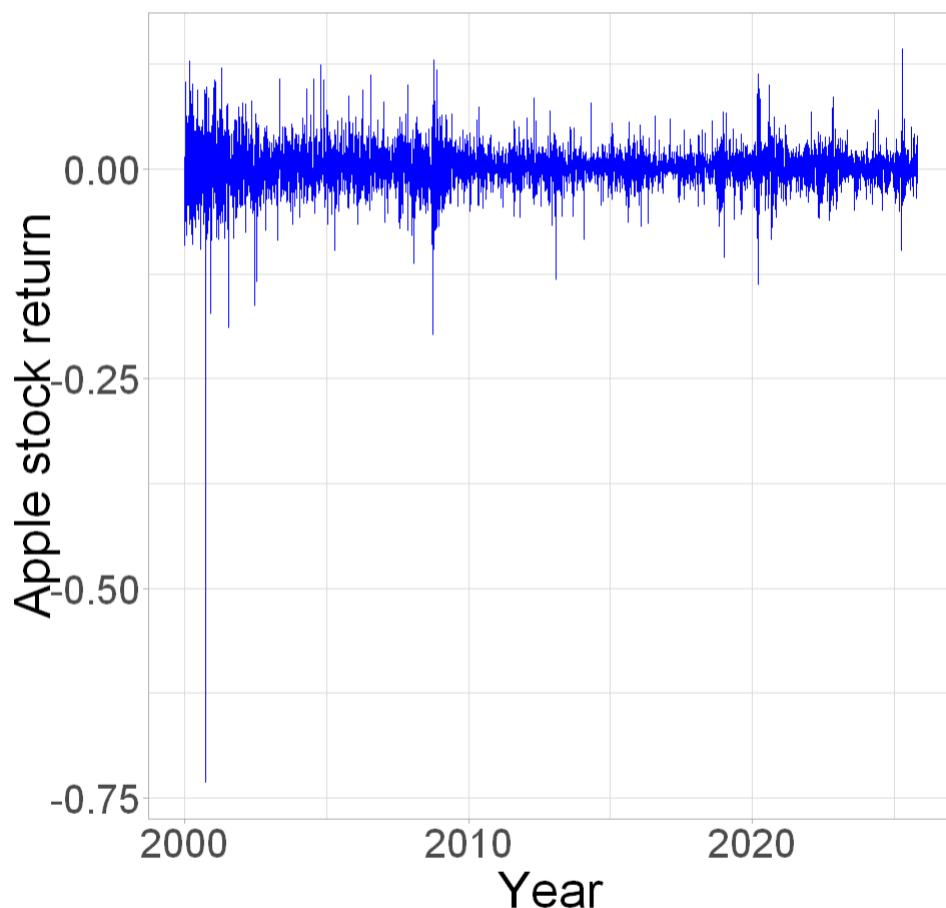
	AAPL.Close
2000-01-03	NA
2000-01-04	-0.08807741
2000-01-05	0.01452714
2000-01-06	-0.09051390
2000-01-07	0.04628127
2000-01-10	-0.01774444

```
AAPL.Close  
2025-10-24  0.0124045266  
2025-10-27  0.0225353862  
2025-10-28  0.0007065783  
2025-10-29  0.0025988958  
2025-10-30  0.0062834495  
2025-10-31 -0.0038023517
```

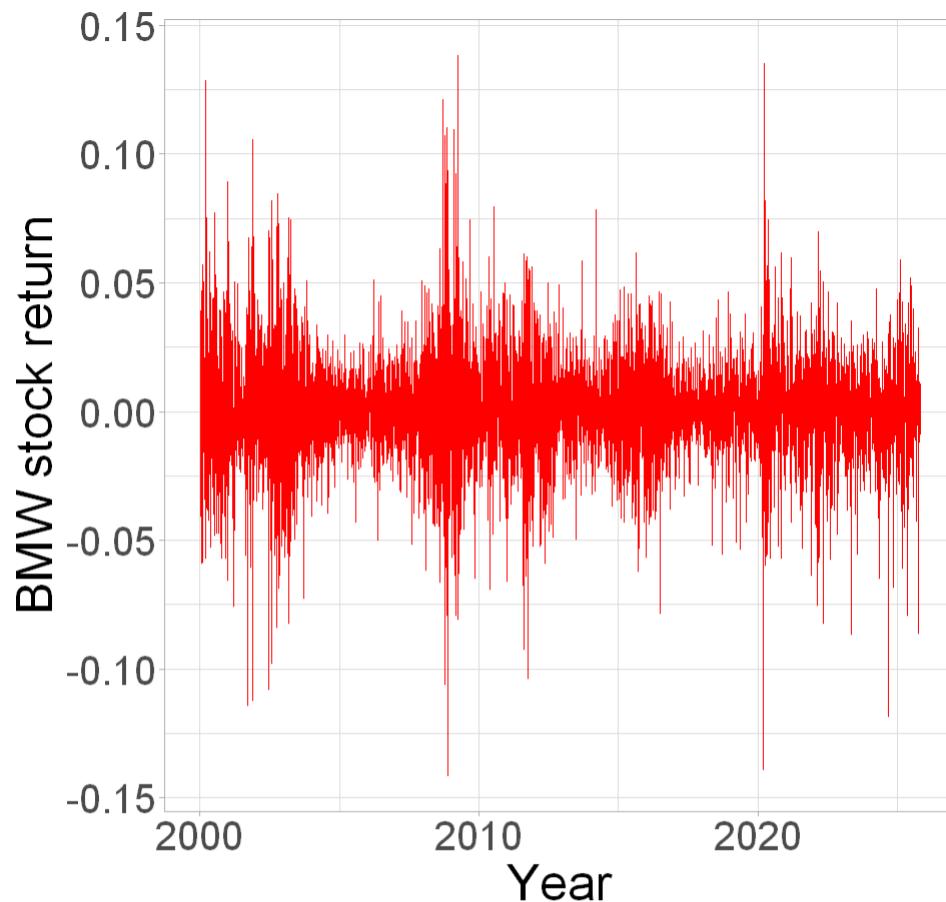
```
In [37]: dbmw = diff(log(bmw)) # Compute log returns (difference in log prices)  
head(dbmw) # Show first few log returns  
tail(dbmw) # Show last few log returns  
dbmw = na.omit(dbmw) # Remove missing values (from first difference)
```

```
BMW.DE.Close  
2000-01-03      NA  
2000-01-04 -0.041189437  
2000-01-05 -0.019986371  
2000-01-06 -0.003249692  
2000-01-07 -0.001809928  
2000-01-10  0.039081363  
BMW.DE.Close  
2025-10-24  0.009657205  
2025-10-27  0.004425788  
2025-10-28 -0.003440637  
2025-10-29  0.010530287  
2025-10-30 -0.008808489  
2025-10-31 -0.008143118
```

```
In [38]: library(ggplot2)  
library(ggfortify)  
options(repr.plot.width=8, repr.plot.height=8)  
fig = autoplot(dapple, colour = 'blue')  
fig = fig + theme(aspect.ratio=1) +  
theme_light() +  
theme(aspect.ratio=1) +  
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +  
theme(text=element_text(size=30)) +  
labs(x = "Year") +  
labs(y = "Apple stock return")  
fig
```



```
In [39]: library(ggplot2)  
library(ggfortify)  
options(repr.plot.width=8, repr.plot.height=8)  
fig = autoplot(dbmw, colour = 'red')  
fig = fig + theme(aspect.ratio=1) +  
theme_light() +  
theme(aspect.ratio=1) +  
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +  
theme(text=element_text(size=30)) +  
labs(x = "Year") +  
labs(y = "BMW stock return")  
fig
```



```
In [40]: library(urca)
summary(ur.df(dapple, type='trend', lags=8, selectlags="AIC"))

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression trend

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.72856 -0.01061 -0.00007  0.01152  0.14123 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.969e-04  6.228e-04  0.958   0.33787  
z.lag.1     -9.754e-01  3.540e-02 -27.552  < 2e-16 *** 
tt          8.049e-08  1.658e-07  0.485   0.62750  
z.diff.lag1 -6.046e-02  3.334e-02 -1.813   0.06982 .  
z.diff.lag2 -6.050e-02  3.102e-02 -1.951   0.05113 .  
z.diff.lag3 -7.391e-02  2.846e-02 -2.597   0.00941 ** 
z.diff.lag4 -3.621e-02  2.556e-02 -1.417   0.15661  
z.diff.lag5 -1.820e-02  2.199e-02 -0.828   0.40785  
z.diff.lag6 -7.792e-03  1.783e-02 -0.437   0.66218  
z.diff.lag7  1.801e-02  1.238e-02  1.455   0.14572  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02502 on 6478 degrees of freedom
Multiple R-squared:  0.5194,    Adjusted R-squared:  0.5188 
F-statistic:  778 on 9 and 6478 DF,  p-value: < 2.2e-16

Value of test-statistic is: -27.5522 253.0426 379.5638

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

Interpretation of the Unit Root Test

Key results: $\tau_3 = -27.5522$; $\varphi_2 = 253.0426$; $\varphi_3 = 379.5638$

- **τ_3 (unit root):** Far below the 1% critical value \Rightarrow **reject** the null of a unit root. The process is not I(1).
- **φ_2 and φ_3 (deterministic terms):** Both exceed their 1% critical values \Rightarrow **reject** that the deterministic components are zero.

Conclusion: The series is **trend-stationary**: stationary *around* a deterministic component (intercept and linear trend).

```
In [41]: t = 1:length(dapple)
model.dapple = lm(dapple ~ t)
```

```

library(stargazer)
stargazer(model.dapple, type = "text")
# The returns are stationary.

=====
Dependent variable:
-----
dapple
-----
t          0.0000
(0.0000)

Constant   0.001
(0.001)

-----
Observations      6,497
R2            0.0001
Adjusted R2     -0.0001
Residual Std. Error    0.025 (df = 6495)
F Statistic     0.352 (df = 1; 6495)
=====
Note:           *p<0.1; **p<0.05; ***p<0.01

```

Interpretation of the Trend Regression

Model: $dapple = \alpha + \beta t + \varepsilon$

- The estimated trend coefficient (β) is approximately **0.00000**.
- The coefficient is **not statistically significant** (p-value > 0.1).
- This means there is **no significant deterministic trend** in the return series.

Conclusion: The stock returns are **stationary** with no deterministic trend to remove. The mean of returns remains constant over time, and any fluctuations are random rather than systematic.

```

In [42]: library(urca)
summary(ur.df(dbmw, type='trend', lags=8, selectlags="AIC"))

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression trend

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q      Max 
-0.141485 -0.009794 -0.000075  0.010168  0.138011 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.272e-04 5.027e-04   0.651   0.5152    
z.lag.1     -1.016e+00 1.732e-02 -58.645   <2e-16 ***  
tt        -4.832e-08 1.318e-07  -0.367   0.7139    
z.diff.lag  2.655e-02 1.231e-02   2.157   0.0311 *    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02037 on 6591 degrees of freedom
Multiple R-squared:  0.4951,    Adjusted R-squared:  0.4949 
F-statistic: 2154 on 3 and 6591 DF,  p-value: < 2.2e-16

Value of test-statistic is: -58.6455 1146.431 1719.646

Critical values for test statistics:
    1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34

```

Interpretation of the Unit Root Test

Key results: $\tau_3 = -58.6455$; $\varphi_2 = 1146.431$; $\varphi_3 = 1719.646$

- τ_3 (unit root):** Much smaller (more negative) than the 1% critical value (-3.96) \Rightarrow **reject** the null of a unit root. The process is not I(1).
- φ_2 and φ_3 (deterministic terms):** Both far exceed their 1% critical values \Rightarrow **reject** the null that the deterministic components (intercept or trend) are zero.

Conclusion: The series is **trend-stationary**. It is stationary around a deterministic component (intercept and linear trend), so no differencing is needed. When modeling, include the trend term to capture the deterministic pattern.

```
In [43]: t = 1:length(dbmw)
model.dbmw = lm(dbmw ~ t)

library(stargazer)
stargazer(model.dbmw, type = "text")
# The returns are stationary.
```

```
=====
Dependent variable:
-----
dbmw
-----
t           -0.00000
(0.00000)

Constant      0.0003
(0.001)

-----
Observations    6,604
R2            0.00001
Adjusted R2     -0.0001
Residual Std. Error   0.020 (df = 6602)
F Statistic     0.076 (df = 1; 6602)
-----
Note:          *p<0.1; **p<0.05; ***p<0.01
```

Interpretation of the Trend Regression

Model: $dbmw = \alpha + \beta t + \epsilon$

- The estimated trend coefficient (β) is approximately **0.00000**.
- The coefficient is **not statistically significant** ($p\text{-value} > 0.1$).
- This means there is **no significant deterministic trend** in the return series.

Conclusion: The stock returns are **stationary** with no deterministic trend to remove. The mean of returns remains constant over time, and any fluctuations are random rather than systematic.

Why ADF Flags Deterministic Components but the Trend Regression Looks Insignificant

Context. With an ADF test that includes a trend, we estimate

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum \delta_i \Delta y_{t-i} + \epsilon_t.$$

- ADF τ_3** tests the unit root ($H_0: \gamma = 0$).
- ADF φ -stats (φ_2, φ_3)** jointly test whether deterministic terms (intercept/trend) belong in the specification when assessing stationarity.

What “significant φ ” means. Rejecting with φ_2/φ_3 says: when testing for a unit root, a model that allows deterministic components fits materially better than one that forces them to zero. It does *not* imply a large economic trend.

OLS trend on returns. Regressing returns on time, $dapple = \alpha + \beta \cdot t + \epsilon$ or $dbmw = \alpha + \beta \cdot t + \epsilon$, directly asks if the slope $\beta \neq 0$. Your estimate is ~ 0 and not significant \Rightarrow **no meaningful linear drift in returns**.

Reconciling the two

- ADF result:** Include deterministic terms in the unit-root testing setup; the series is (trend-)stationary rather than $I(1)$.
- OLS result:** Even if allowed, the *estimated* trend slope β in returns is essentially zero and statistically insignificant.

Conclusion. There is **no contradiction**: the series is stationary (no unit root), and returns have **no deterministic trend to remove**. Model returns with a constant (mean) and stochastic dynamics (e.g., ARMA/GARCH) rather than differencing or imposing a time trend.

Testing for the ARCH Effect

The presence of ARCH (Autoregressive Conditional Heteroskedasticity) can be examined using methods that vary in strictness — from the most visual to the most formal. The steps below are ordered from **least strict** to **most strict**:

1. Plot of the squared series

The squared series (e.g., squared returns) approximates the **conditional variance** because, under a zero-mean assumption, the variance of a series equals the expected value of its square. Therefore, periods where the squared values are large correspond to periods of high

conditional variance (volatility). Plotting the squared series provides a *visual* impression of volatility clustering — periods of high and low volatility can be observed directly. This helps identify whether large shocks tend to cluster together, but it is qualitative and cannot measure how persistent these changes are.

2. ACF of the squared series

The Autocorrelation Function (ACF) of the squared series gives a *quantitative* measure of volatility persistence. If several autocorrelations are significant, it suggests that large (or small) variances tend to follow each other — a hallmark of volatility clustering. Hence, the ACF is **more informative** than the plot alone because it captures the *strength* and *duration* of dependence in volatility rather than just its timing.

3. Plot of the squared residuals

Uses residuals from a correctly specified conditional mean model (e.g., ARMA). By focusing on residuals, this method removes any serial correlation in the mean, ensuring that visible clustering is due to changing variance, not autocorrelation in the level of the series.

4. ACF of the squared residuals

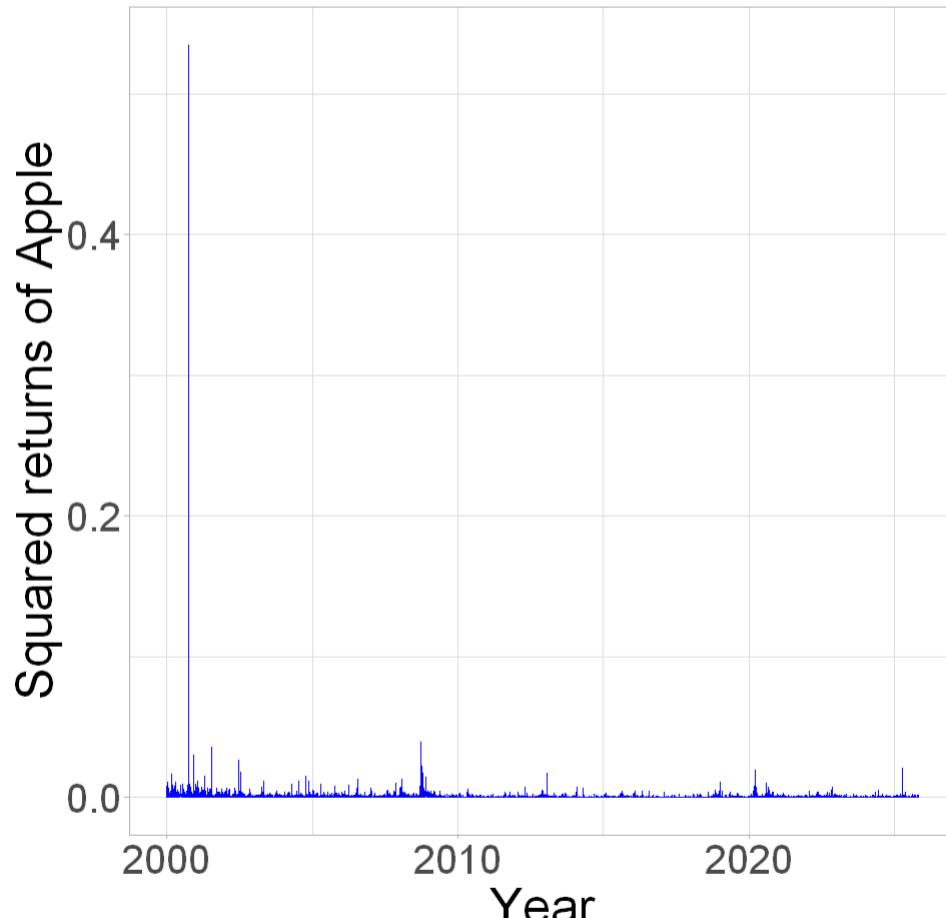
Once the mean equation is properly modeled, the ACF of the squared residuals captures genuine time-varying volatility. Significant autocorrelations here indicate an **ARCH or GARCH effect** — volatility that depends on its own past.

5. Statistical testing

The most formal step — tests such as the ARCH-LM or Ljung–Box test on squared residuals provide quantitative evidence of conditional heteroskedasticity.

Why residuals matter: Residual-based methods are preferred because they ensure that the *conditional mean equation* has been correctly modeled first. Only then can any remaining dependence in the squared residuals be attributed to true volatility dynamics rather than unmodeled mean effects.

```
In [44]:  
library(ggplot2)  
library(ggfortify)  
options(repr.plot.width=8, repr.plot.height=8)  
fig = autoplot(dapple^2, colour = 'blue')  
fig = fig + theme(aspect.ratio=1) +  
theme_light() +  
theme(aspect.ratio=1) +  
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +  
theme(text=element_text(size=30)) +  
labs(x = "Year") +  
labs(y = "Squared returns of Apple")  
fig
```



Interpretation of the Squared Returns Plot

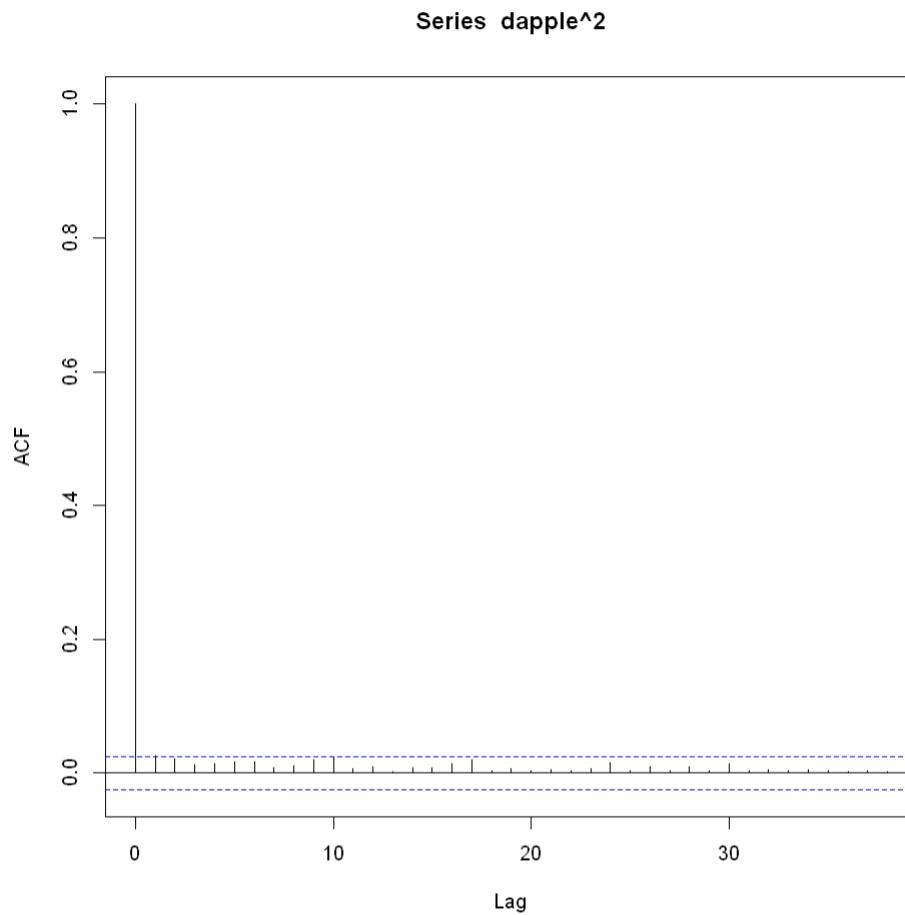
The figure displays the **squared returns of Apple** over time, which approximate the series' conditional variance. This allows us to visually inspect how volatility evolves.

- A few extreme spikes—particularly around the early 2000s—indicate periods of exceptionally high volatility. These could correspond to major market events or data irregularities.

- After these early spikes, the variance appears much lower and more stable, with occasional smaller volatility bursts around 2008–2009 and later years.
- The long stretches of calm followed by short bursts of volatility are indicative of **volatility clustering**, a typical feature of financial returns and consistent with the presence of **ARCH effects**.

Conclusion: The squared returns are mostly small but occasionally exhibit large, isolated spikes, suggesting that Apple's return volatility is not constant over time. This visual evidence points to *time-varying conditional variance*, which can be formally tested using ARCH or GARCH modeling techniques.

```
In [45]: acf(dapple^2)
```



Interpretation of the ACF of Squared Returns

The figure shows the **Autocorrelation Function (ACF)** of the **squared returns** of Apple. This plot helps detect whether volatility (conditional variance) is correlated over time — a key feature of ARCH effects.

- For all lags other than lag=0, autocorrelations drop quickly and remain within the 95% confidence bounds (the blue dashed lines).
- This pattern suggests **no significant serial correlation** in the squared returns — volatility does not appear to depend systematically on past volatility.

Conclusion: Based on the ACF of squared returns, there is **no strong visual evidence of ARCH effects** in Apple's returns. Volatility clustering is not pronounced, meaning that high-volatility periods are not strongly correlated with one another. However, formal tests (such as the ARCH-LM test) would be needed to confirm this result statistically.

```
In [46]: model.dapple = auto.arima(dapple, max.p = 20, max.q = 20)
model.dapple
res.dapple = model.dapple$residuals

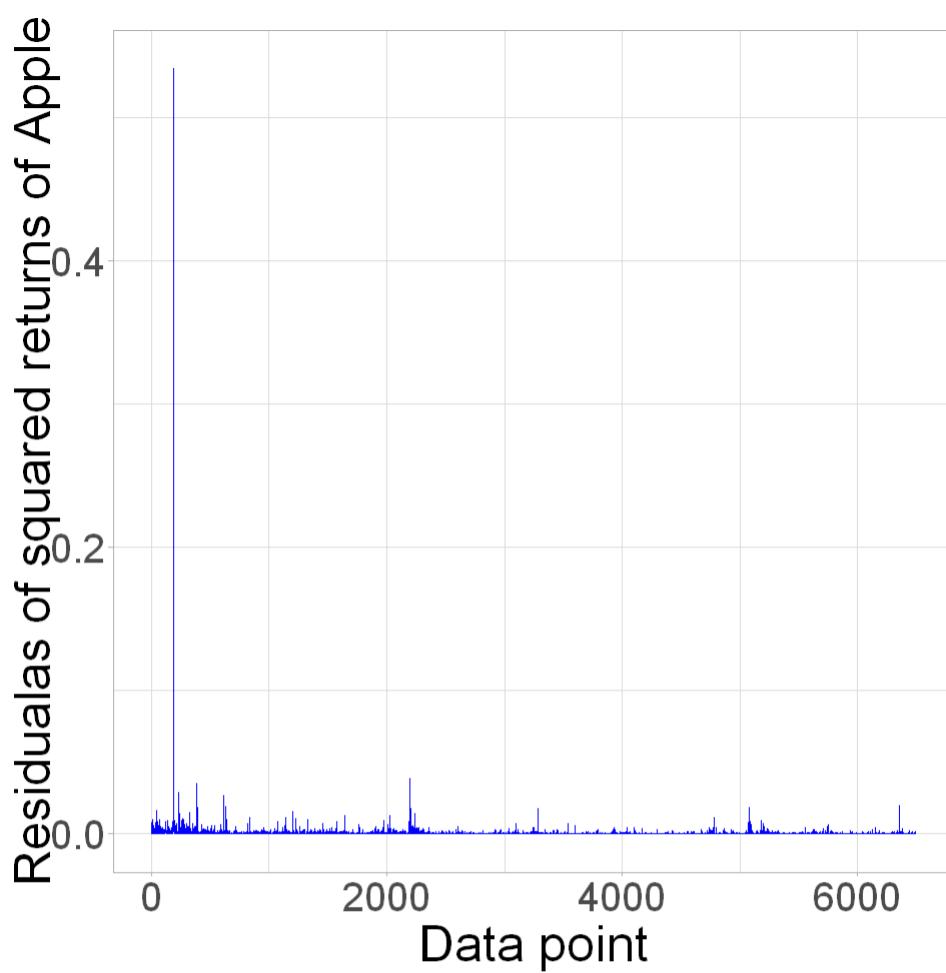
library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(res.dapple^2, colour = 'blue')
fig = fig + theme(aspect.ratio=1) +
  theme_light() +
  theme(aspect.ratio=1) +
  theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
  theme(text=element_text(size=30)) +
  labs(x = "Data point") +
  labs(y = "Residuals of squared returns of Apple")
fig
```

Series: dapple
ARIMA(3,0,3) with non-zero mean

Coefficients:

	ar1	ar2	ar3	ma1	ma2	ma3	mean
ar1	-0.4581	-0.6816	-0.4681	0.4198	0.6663	0.4298	9e-04
s.e.	0.1257	0.0952	0.0588	0.1299	0.0928	0.0581	3e-04

sigma^2 = 0.0006317: log likelihood = 14716.89
AIC=-29417.77 AICc=-29417.75 BIC=-29363.54



Interpretation of the Squared Residuals Plot

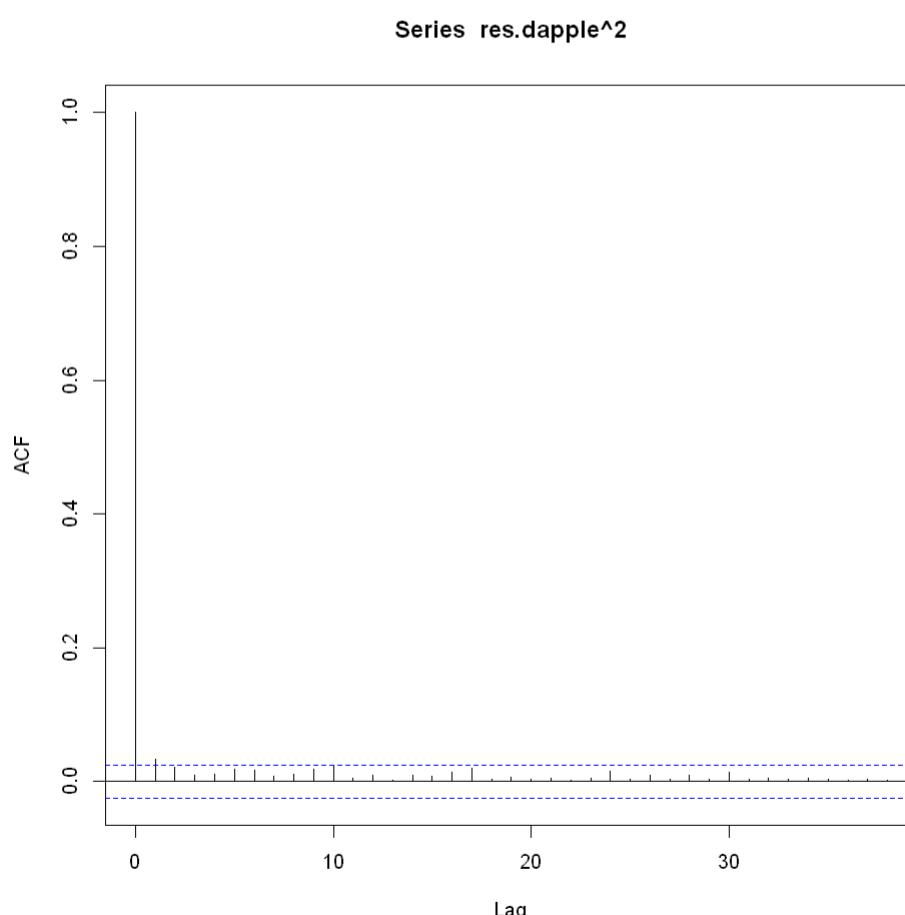
The figure shows the **squared residuals of Apple's returns** over time. Squared residuals approximate the *conditional variance* after accounting for the mean dynamics of the series (e.g., ARMA structure). This plot helps assess whether volatility changes remain after modeling the mean correctly.

Both plots — the squared returns of Apple and the residuals from a fitted model of those squared returns — show very similar patterns.

- The squared returns show strong spikes at certain points (especially around the early 2000s), reflecting periods of large volatility.
- The residuals of the squared returns still display nearly the same spikes and clustering, suggesting that **the mean model (e.g., ARMA)** did not capture much of the volatility dynamics.
- In other words, **the volatility structure remains unexplained** — the conditional variance is not yet fully modeled. This implies that a variance model, such as an ARCH or GARCH specification, is needed to account for time-varying volatility.

Conclusion: Since the residuals of the squared returns look visually similar to the original squared returns, it indicates that *conditional heteroskedasticity remains* even after modeling the mean. The volatility clustering visible in both plots suggests the presence of **ARCH effects** that should be modeled explicitly with an ARCH or GARCH process.

```
In [47]: acf(res.dapple^2)
```



Interpretation of the ACF of Squared Residuals

The plot displays the **Autocorrelation Function (ACF)** of the **squared residuals** from the fitted model. This diagnostic is used to detect any remaining serial dependence in volatility — that is, whether *ARCH effects* persist after modeling the conditional mean.

- The first bar (lag 0) represents the autocorrelation of the series with itself and is always equal to 1, so it is not interpreted.
- The **first-order autocorrelation** is small but statistically significant, while all higher-order autocorrelations lie **within the 95% confidence bounds** (the dashed blue lines).
- This indicates that there is **some short-term dependence** in the conditional variance, but the effect is weak and not persistent across multiple lags.

Conclusion: The presence of a small but significant first-order autocorrelation suggests a *mild ARCH(1) effect* — volatility at time t may depend slightly on volatility from the previous period. However, since higher-order autocorrelations are insignificant, the volatility clustering is **short-lived** and not strongly systematic.

Interpretation of the Squared Residuals Plot and Its ACF

The **plot of squared residuals** shows visible clusters of high volatility, suggesting that volatility is not constant over time. However, the **ACF of squared residuals** indicates that only the first lag is statistically significant, with all higher lags within the confidence bounds.

- This pattern means that volatility changes exist, but they are **short-lived** rather than persistent.
- The clustering seen in the plot likely reflects **random bursts of volatility** rather than systematic, predictable patterns.
- Because serial correlation in the squared residuals is minimal, there is **little evidence of strong ARCH effects**.

Conclusion: Although the squared residuals show occasional volatility clusters, the ACF results imply that these bursts are *mostly random and not predictable*. The volatility does not display strong dependence on its own past, suggesting that a standard constant-variance model or, at most, a weak ARCH(1) specification would be sufficient.

Testing for ARCH Effects: Engle's ARCH-LM Test

Visual tools (plots/ACFs of squared residuals) can suggest volatility clustering, but a **formal test** is preferred. Engle's ARCH-LM test provides this check.

Procedure

1. **Model the conditional mean** and obtain residuals \hat{u}_t .
2. **Auxiliary regression** (ARCH(p)):

$$\hat{u}_t^2 = \alpha_0 + \alpha_1 \hat{u}_{t-1}^2 + \cdots + \alpha_p \hat{u}_{t-p}^2 + v_t$$

3. **Null hypothesis:**

$$\alpha_1 = \alpha_2 = \cdots = \alpha_p = 0 \quad (\text{no ARCH effects})$$

Test statistic (with effective sample)

$$Q = (T - p) \times R^2,$$

where T is the sample size of \hat{u}_t , p is the number of lags in the auxiliary regression, and R^2 is the R^2 in the auxiliary regression. Under the null,

$$Q \sim \chi^2(p).$$

Reject the null if Q exceeds the χ^2 critical value.

How to choose p ?

- **Use several reasonable orders** and check robustness: e.g., $p \in \{1, 4, 8, 12\}$ (monthly/quarterly style horizons) or up to about \sqrt{T} .
- **Report a small table** of p-values across p . If many (or any key) orders are significant, that's evidence of ARCH effects.

Reminder: Ensure the *mean* model is well specified first (residuals uncorrelated). Otherwise, leftover mean dynamics can masquerade as ARCH.

Testing for ARCH Effects: Engle's ARCH-LM Test

Step 1: Estimate conditional mean model

```
In [48]: # Step 1: Fit mean model automatically (allow up to AR/MA(20))
step1.dapple = auto.arima(dapple, max.p = 20, max.q = 20)
```

```
# Inspect selected ARIMA mean model
step1.dapple

Series: dapple
ARIMA(3,0,3) with non-zero mean

Coefficients:
ar1      ar2      ar3      ma1      ma2      ma3    mean
-0.4581 -0.6816 -0.4681  0.4198  0.6663  0.4298  9e-04
s.e.    0.1257  0.0952  0.0588  0.1299  0.0928  0.0581  3e-04

sigma^2 = 0.0006317: log likelihood = 14716.89
AIC=-29417.77  AICc=-29417.75  BIC=-29363.54
```

Testing for ARCH Effects: Engle's ARCH-LM Test

Step 2: Run Engle's ARCH-LM Test

```
In [49]: # Mean model chosen earlier via auto.arima: ARMA(3,3)
# ARCH/GARCH utilities incl. arch.test()
library(aTSA)

# Fit ARMA(3,3) to (stationary) returns
est = arima(dapple, order = c(3,0,3))

# Print coefficients/AIC
est

# Engle's ARCH-LM on residuals (null: no ARCH). Small p-value ⇒ model volatility (ARCH/GARCH)
arch.test(est)
```

Call:
`arima(x = dapple, order = c(3, 0, 3))`

Coefficients:

	ar1	ar2	ar3	ma1	ma2	ma3	intercept
-	-0.4581	-0.6816	-0.4681	0.4198	0.6663	0.4298	9e-04
s.e.	0.1257	0.0952	0.0588	0.1299	0.0928	0.0581	3e-04

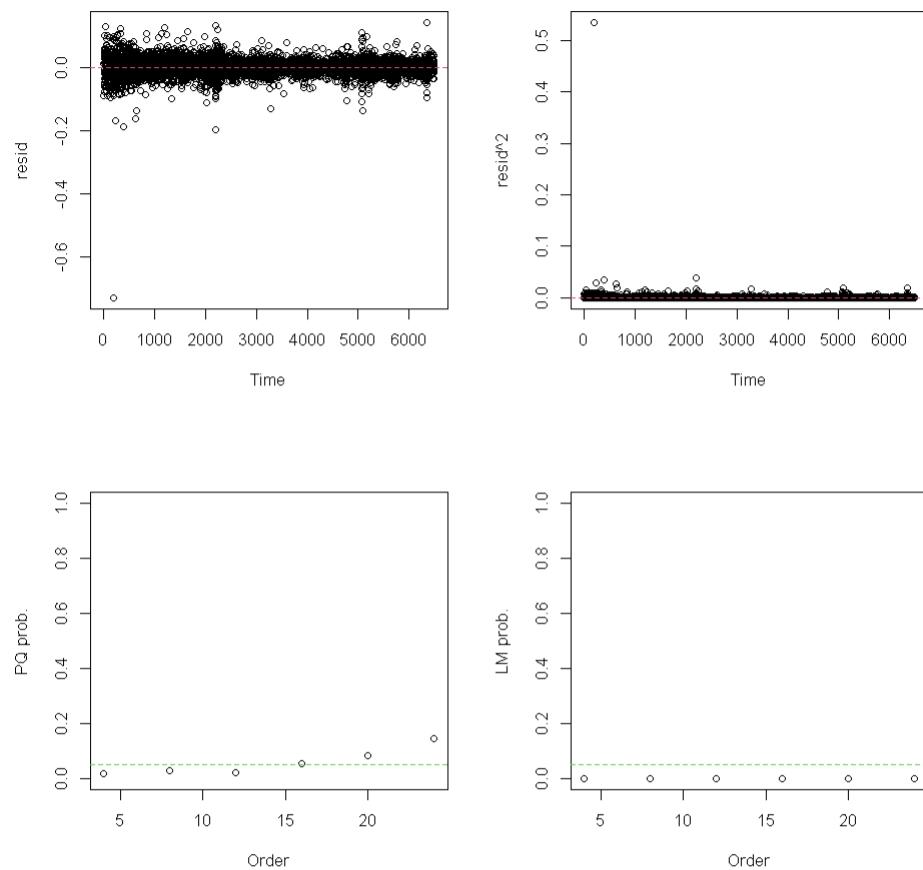
`sigma^2` estimated as 0.000631: log likelihood = 14716.89, aic = -29417.77
 ARCH heteroscedasticity test for residuals
 alternative: heteroscedastic

Portmanteau-Q test:

order	PQ	p.value	
[1,]	4	11.9	0.0179
[2,]	8	17.1	0.0293
[3,]	12	23.8	0.0219
[4,]	16	26.1	0.0530
[5,]	20	29.3	0.0819
[6,]	24	31.3	0.1465

Lagrange-Multiplier test:

order	LM	p.value	
[1,]	4	157123	0
[2,]	8	73970	0
[3,]	12	45968	0
[4,]	16	34083	0
[5,]	20	26199	0
[6,]	24	21798	0



Interpretation of the ARCH-LM Test vs. ACF of Squared Residuals

The **ACF of squared residuals** and the **ARCH-LM test** both assess volatility clustering, but they measure it differently, which explains why the conclusions can diverge.

- **ACF of squared residuals:** This is a *visual and descriptive* tool. It looks for significant autocorrelations in the squared residuals to see if volatility persists over time. Here, only the first lag was slightly above the confidence bounds — suggesting **weak short-term autocorrelation** in volatility.
- **ARCH-LM test (Engle's test):** This is a *formal joint test* of whether several past squared residuals jointly explain current volatility. It is more sensitive and often picks up effects that individual ACF bars do not. Even small individual correlations can jointly produce a **highly significant test statistic** when combined across several lags.

Here, the LM test shows extremely large statistics and **p-values ≈ 0** for all lag orders tested (4, 8, 12, 16, 20, 24). This provides overwhelming evidence of **conditional heteroskedasticity** — the presence of ARCH effects — even though the ACF plot looked almost flat.

Why this happens:

- The ACF is a point-by-point test (each lag individually). It can miss weak but persistent correlations.
- The LM test pools information across multiple lags, giving it much greater statistical power.
- Even small, individually insignificant autocorrelations can add up to a large joint LM statistic.

Conclusion: Although the ACF showed only weak first-order correlation, the LM test confirms that the residual variance still depends strongly on its past values. Therefore, there are clear and significant **ARCH effects**, meaning volatility is not random but clustered and predictable over time — consistent with the need to fit an **ARCH or GARCH model**.

```
In [50]: # ARCH-LM (p = 1): Manual test for first-order autocorrelation in squared residuals
step1.dapple = arima(dapple, order = c(3,0,3))
step1.dapple.res = step1.dapple$residuals
step1.dapple.res2 = step1.dapple.res^2

T.temp = length(step1.dapple.res2)                      # total obs in u_t^2
x.temp = step1.dapple.res2[2:T.temp]                   # u_{t-1}^2 (lag 1)
y.temp = step1.dapple.res2[1:(T.temp - 1)]             # u_t^2 aligned with lag

step2.dapple = lm(y.temp ~ x.temp)                     # auxiliary OLS: u_t^2 ~ u_{t-1}^2
step2.dapple
summary(step2.dapple)$r.squared                         # regression output
                                                       # R^2 from auxiliary regression

# LM statistic Q = (T_eff) * R^2 with df = 1
step2.dapple.res = step2.dapple$residuals
R2 = 1 - var(step2.dapple.res) / var(y.temp)           # same as summary(.)$r.squared
Q = length(y.temp) * R2                                # T_eff = T - 1 after lagging

print("The test statistic is"); Q
print("The critical value is"); qchisq(0.95, df = 1)

# Decision: TRUE => reject H0 (no ARCH(1))
print("We reject the null of homoskedasticity, TRUE or FALSE")
Q > qchisq(0.95, df = 1)
```

```

Call:
lm(formula = y.temp ~ x.temp)

Coefficients:
(Intercept)      x.temp
 0.0006096    0.0341271
0.00116445595473697
[1] "The test statistic is"
7.56430588197096
[1] "The critical value is"
3.84145882069412
[1] "We reject the null of homoskedasticity, TRUE or FALSE"
TRUE

```

Exercise 1

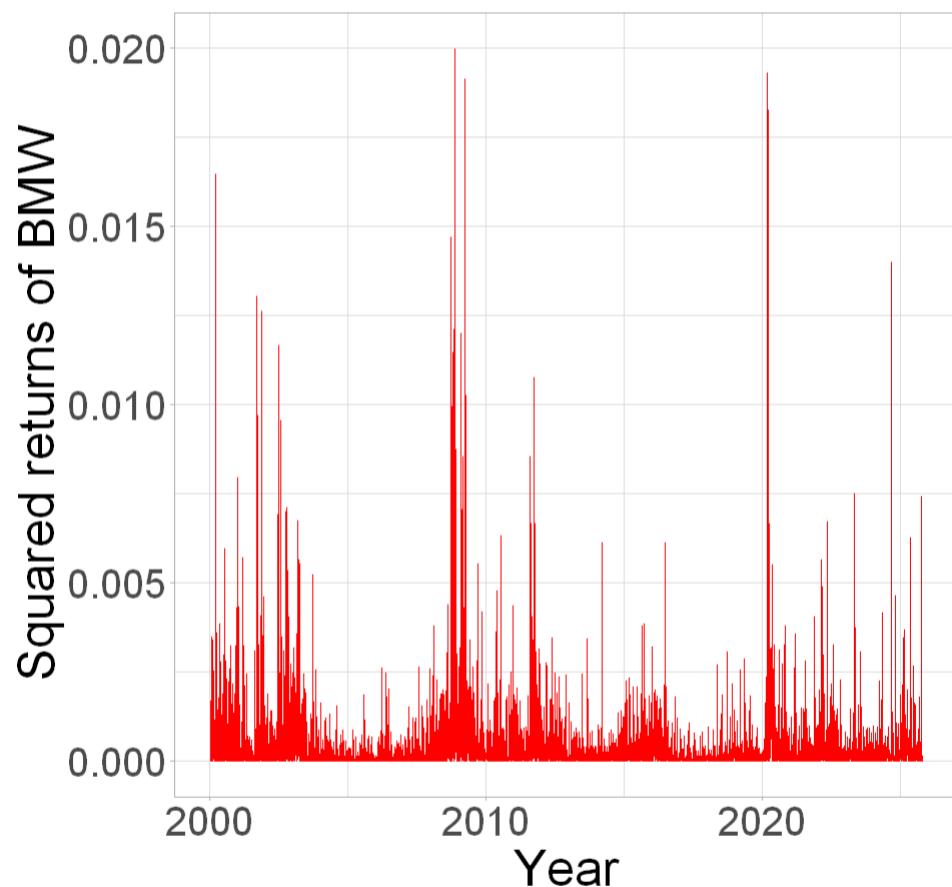
Test for the ARCH effect in the BMW returns.

Solution to Exercise 1

```

In [51]: library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(dbmw^2, colour = 'red')
fig = fig + theme(aspect.ratio=1) +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
theme(text=element_text(size=30)) +
labs(x = "Year") +
labs(y = "Squared returns of BMW")
fig

```



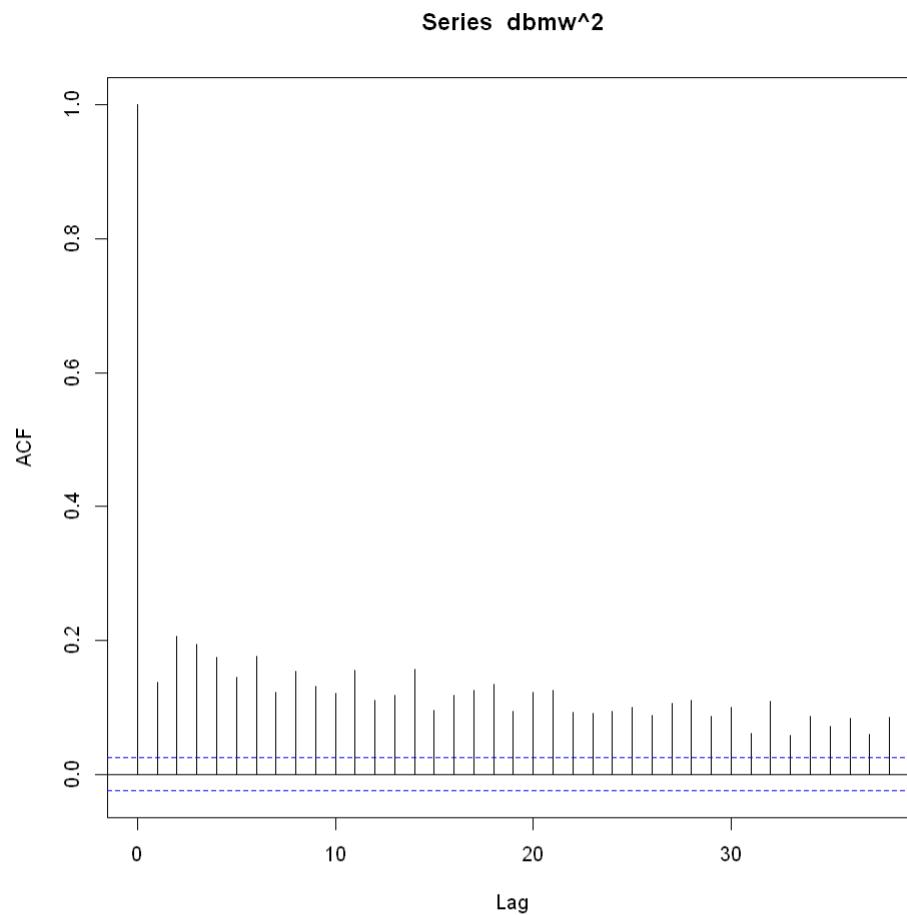
Interpretation of the Squared Returns Plot for BMW

The figure displays the **squared daily returns of BMW** over time. Squared returns approximate the conditional variance of the series and are used to visually inspect volatility patterns and potential ARCH effects.

- The plot shows several distinct **volatility clusters** — periods where large squared returns (high volatility) are followed by more large values, particularly during financial turmoil around **2008–2009** and again around **2020**.
- Between these clusters, there are extended periods of **low volatility** where the squared returns remain close to zero.
- This alternating pattern of calm and turbulent periods indicates that volatility is **time-varying** rather than constant — a hallmark of **conditional heteroskedasticity** or **ARCH effects**.

Conclusion: The squared returns of BMW exhibit clear signs of **volatility clustering**: large shocks tend to be followed by large shocks, and small by small. This suggests that the unconditional variance is not constant over time, and modeling the conditional variance using an **ARCH or GARCH model** would be appropriate to capture this volatility behavior.

```
In [52]: acf(dbmw^2)
```



Interpretation of the ACF of Squared Returns for BMW

The figure presents the **Autocorrelation Function (ACF)** of the **squared returns** of BMW. This diagnostic helps assess whether volatility (conditional variance) is serially correlated — a defining feature of *ARCH effects*.

- All autocorrelations are above the 95% confidence bounds, indicating that the dependence in the squared returns is not limited to short lags but persists across many periods.
- This widespread significance suggests that **volatility shocks are long-lasting** — high-volatility periods are followed by other high-volatility periods, and low-volatility periods by low-volatility ones.
- The slowly decaying pattern implies **strong volatility persistence**, where conditional variance depends heavily on its past values rather than being random or short-lived.

Conclusion: The ACF of BMW's squared returns shows **significant autocorrelation at almost all lags**, providing strong evidence of **persistent ARCH effects**. This means volatility is highly time-dependent and clusters over extended periods. To capture this dynamic structure, a **GARCH-type model** is recommended, as it allows past volatility to influence current variance over many lags.

```
In [53]: model.dbmw = auto.arima(dbmw, max.p = 20, max.q = 20)
model.dbmw
res.dbmw = model.dbmw$residuals
```

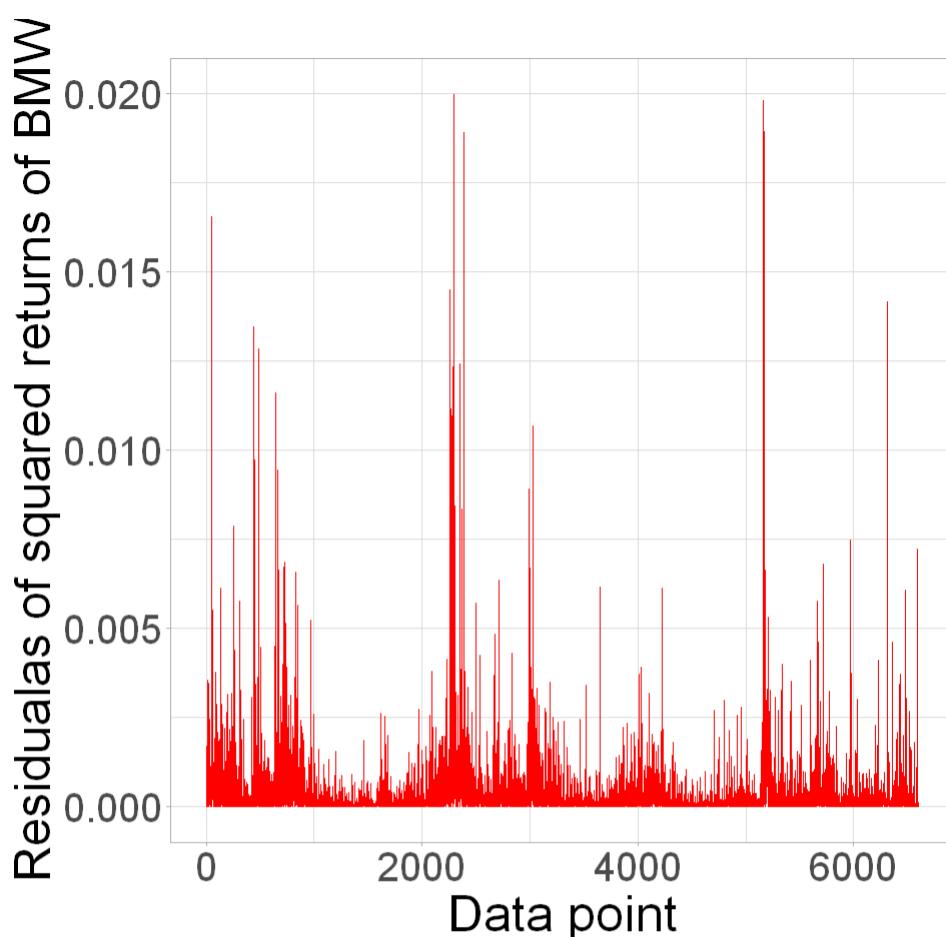
```
library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(res.dbmw^2, colour = 'red')
fig = fig + theme(aspect.ratio=1) +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
theme(text=element_text(size=30)) +
labs(x = "Data point") +
labs(y = "Residuals of squared returns of BMW")
fig
```

Series: dbmw
ARIMA(2,0,1) with zero mean

Coefficients:

	ar1	ar2	ma1
0.4036	-0.0330	-0.3925	
s.e.	2.3884	0.0173	2.3922

sigma^2 = 0.000415: log likelihood = 16343.93
AIC=-32679.87 AICc=-32679.86 BIC=-32652.68



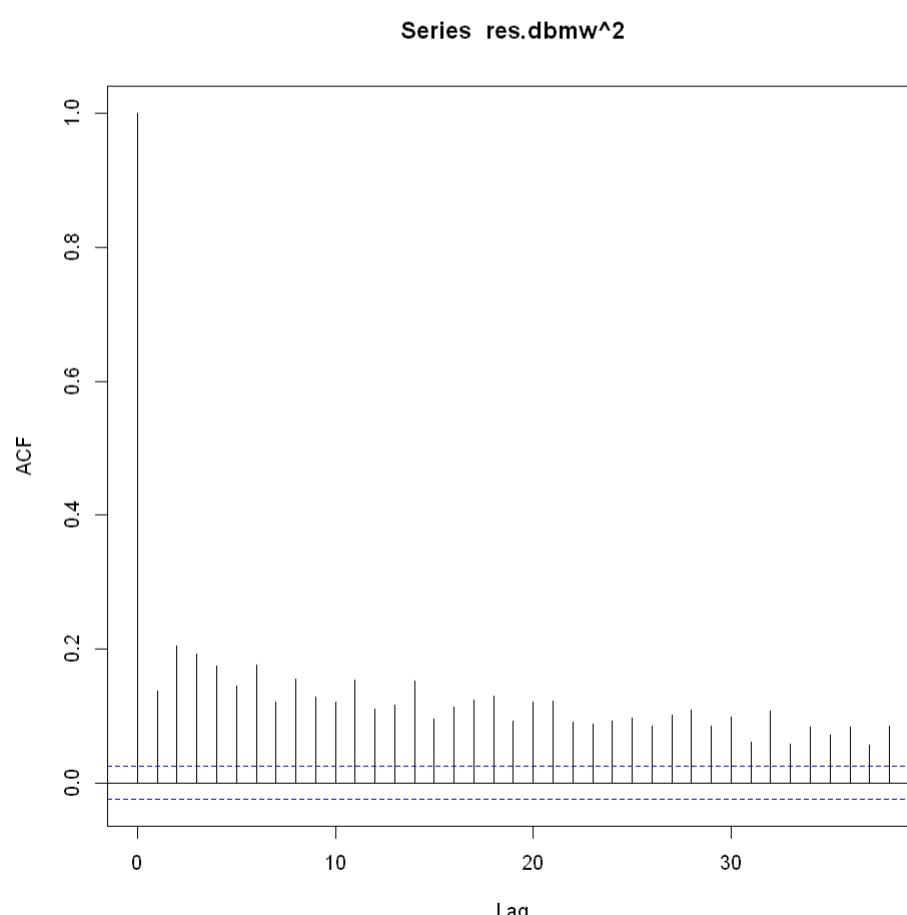
Interpretation of the Squared Residuals Plot for BMW

The figure shows the **squared residuals of BMW's returns** over time. Squared residuals approximate the *conditional variance* after accounting for the mean dynamics (here modeled using an ARMA(2,1) structure). This plot is used to evaluate whether time-varying volatility remains after the mean has been properly specified.

- The plot still displays **pronounced volatility clusters** — periods of high volatility (notably in the early 2000s, around 2008–2009, and near 2020) followed by more tranquil intervals.
- These patterns indicate that **the mean model did not fully capture the dynamics of volatility**. The residuals continue to show large, persistent spikes, implying that the variance is not constant over time.
- The strong clustering behavior in the squared residuals suggests the presence of **conditional heteroskedasticity**, meaning the conditional variance depends on past shocks and remains serially correlated.

Conclusion: The residuals of the squared returns still exhibit clear volatility clustering, similar to what was observed in the original squared returns of BMW. This confirms that the ARMA mean model alone is insufficient to explain the data's time-varying volatility. Therefore, a **GARCH-type model** is recommended to explicitly model and capture these ARCH effects.

```
In [54]: acf(res.dbmw^2)
```



Interpretation of the ACF of Squared Residuals for BMW

The figure presents the **Autocorrelation Function (ACF)** of the **squared residuals** from BMW's return model. This plot is used to check whether any *conditional heteroskedasticity* (ARCH effects) remains after modeling the mean dynamics with an ARMA process.

- All of the autocorrelations are **significantly above the 95% confidence bounds**, extending across many lags. This suggests that serial dependence in the variance persists even after fitting the ARMA mean model.
- The positive autocorrelations decay slowly, indicating that **volatility clustering is strong and long-lasting**. In other words, high-volatility periods tend to be followed by more high-volatility periods.
- Because significant autocorrelation remains in the squared residuals, the conditional variance is still time-dependent — the ARMA model has not captured the heteroskedasticity structure.

Conclusion: The ACF of BMW's squared residuals shows **significant and persistent autocorrelation** across many lags, confirming that *ARCH effects remain* even after modeling the mean. This is strong evidence of **conditional heteroskedasticity**, and it suggests that a **GARCH-type model** should be estimated to properly account for BMW's time-varying volatility.

```
In [55]: # Step 1: Fit mean model automatically (allow up to AR/MA(20))
step1.dbmw = auto.arima(dbmw, max.p = 20, max.q = 20)
```

```
# Inspect selected ARIMA mean model
step1.dbmw
```

```
Series: dbmw
ARIMA(2,0,1) with zero mean
```

```
Coefficients:
```

	ar1	ar2	ma1
ar1	0.4036	-0.0330	-0.3925
s.e.	2.3884	0.0173	2.3922

```
sigma^2 = 0.000415: log likelihood = 16343.93
AIC=-32679.87 AICc=-32679.86 BIC=-32652.68
```

```
In [56]: # Mean model chosen earlier via auto.arima: ARMA(3,3)
# ARCH/GARCH utilities incl. arch.test()
library(aTSA)
```

```
# Fit ARMA(2,1) to (stationary) returns
est = arima(dbmw, order = c(2,0,1))
```

```
# Print coefficients/AIC
est
```

```
# Engle's ARCH-LM on residuals (null: no ARCH). Small p-value ⇒ model volatility (ARCH/GARCH)
arch.test(est)
```

```
Call:
```

```
arima(x = dbmw, order = c(2, 0, 1))
```

```
Coefficients:
```

	ar1	ar2	ma1	intercept
ar1	0.4467	-0.0332	-0.4359	2e-04
s.e.	5.3238	0.0198	5.3330	2e-04

```
sigma^2 estimated as 0.0004148: log likelihood = 16344.2, aic = -32678.4
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic
```

```
Portmanteau-Q test:
```

order	PQ	p.value	
[1,]	4	847	0

[2,]	8	1448	0
------	---	------	---

[3,]	12	1885	0
------	----	------	---

[4,]	16	2269	0
------	----	------	---

[5,]	20	2633	0
------	----	------	---

[6,]	24	2893	0
------	----	------	---

```
Lagrange-Multiplier test:
```

order	LM	p.value	
[1,]	4	5837	0

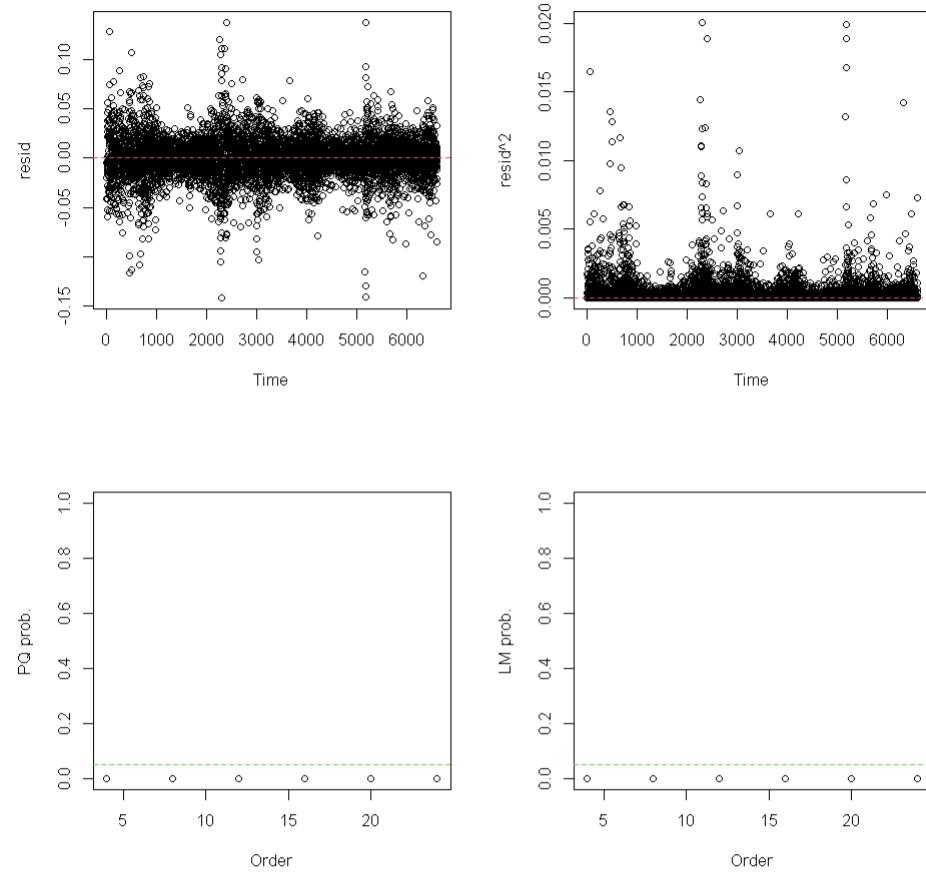
[2,]	8	2543	0
------	---	------	---

[3,]	12	1606	0
------	----	------	---

[4,]	16	1177	0
------	----	------	---

[5,]	20	928	0
------	----	-----	---

[6,]	24	766	0
------	----	-----	---



Interpretation of the Engle's ARCH-LM Test for BMW

The table displays the results of **Engle's Lagrange Multiplier (ARCH-LM) test** for various lag orders. This test examines whether past squared residuals jointly explain current volatility — in other words, whether **ARCH effects** are present in the residuals of BMW's return model.

- For each lag order (4, 8, 12, 16, 20, 24), the LM statistic is **very large** and the corresponding p-value is essentially **0**.
- This means the null hypothesis of *no ARCH effects* is **strongly rejected** at all lag lengths considered.
- The decreasing magnitude of the LM statistic across higher orders (from 5837 at order 4 to 766 at order 24) suggests that most of the ARCH dependence is concentrated in the first few lags, but volatility persistence remains significant throughout.

Conclusion: The ARCH-LM test provides **overwhelming evidence** of conditional heteroskedasticity in BMW's returns. Volatility is not constant but depends on past shocks, confirming the presence of strong **ARCH effects**. Consequently, a **GARCH-type model** is needed to capture the time-varying volatility dynamics of BMW's returns.

Estimating ARMA-GARCH Model

```
In [57]: library(rugarch)
spec = ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model    = list(armaOrder = c(3, 3), include.mean = TRUE),
  distribution.model = "norm"
)
garch.dapple = ugarchfit(spec, dapple)
garch.dapple@fit$matcoef
```

A matrix: 10 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(> t)
mu	1.758733e-03	2.248858e-04	7.820564	5.329071e-15
ar1	-1.228967e+00	3.346934e-04	-3671.918962	0.000000e+00
ar2	-1.219915e+00	2.328223e-04	-5239.684858	0.000000e+00
ar3	-9.888882e-01	1.408235e-03	-702.218145	0.000000e+00
ma1	1.228410e+00	6.956555e-06	176583.166432	0.000000e+00
ma2	1.226750e+00	4.541406e-05	27012.559576	0.000000e+00
ma3	9.938581e-01	3.342210e-05	29736.550066	0.000000e+00
omega	8.882576e-06	1.745264e-06	5.089532	3.589477e-07
alpha1	1.067522e-01	7.226858e-03	14.771589	0.000000e+00
beta1	8.848852e-01	6.107649e-03	144.881468	0.000000e+00

Interpretation of the ARMA(3,3)-GARCH(1,1) Model for Apple Returns

The table reports the estimation results for an **ARMA(3,3)-GARCH(1,1)** model fitted to the series *dapple* (Apple's returns). The ARMA component captures the linear dependence in the mean equation, while the GARCH component models the time-varying volatility (conditional variance).

Mean Equation (ARMA(3,3))

- **μ (constant):** The mean return is estimated at **0.00176** and is highly significant ($p < 0.01$), suggesting that Apple's average daily return is positive and statistically different from zero.
- **AR terms (ar1, ar2, ar3):**
 - **ar1 = -1.229***:** Large, negative, and highly significant, indicating strong short-term mean reversion in returns.
 - **ar2 = -1.219***** and **ar3 = -0.099***:** Both negative and significant, suggesting that past returns up to three days ago have persistent but decaying effects on today's return.
- **MA terms (ma1, ma2, ma3):**
 - **ma1 = 1.228***, ma2 = 1.227***,** and **ma3 = 0.994***:** All large, positive, and significant, indicating that past shocks have strong positive effects on current returns, offsetting the negative AR effects and stabilizing short-run dynamics.

Variance Equation (GARCH(1,1))

- **ω (omega = $8.88 \times 10^{-6}***$):** The constant term is small but highly significant, representing the long-run average variance — a positive baseline level of volatility.
- **α_1 (alpha1 = 0.107***):** Measures the immediate impact of recent shocks (ARCH effect). Its significance indicates that large past shocks increase current volatility — evidence of short-term volatility clustering.
- **β_1 (beta1 = 0.885***):** Measures volatility persistence (GARCH effect). Its large and significant value (close to 1) implies that volatility is highly persistent — once volatility rises, it tends to remain elevated for a long period.

Conclusion: The ARMA(3,3)-GARCH(1,1) model effectively captures both the serial dependence in Apple's mean returns and the strong volatility clustering typical of financial data. The large and significant β_1 (0.885) indicates **high volatility persistence**, while α_1 (0.107) confirms that new shocks have a substantial and immediate impact on volatility. Overall, Apple's return volatility is **time-varying, persistent, and influenced by past shocks**, validating the use of a GARCH-type specification.

Forecasting with an ARMA–GARCH Model

1. Pure AR Model (with Homoscedastic Errors)

In a simple AR(p) model with no ARCH or GARCH components, the errors are assumed to be **conditionally homoscedastic** — meaning their variance is **constant over time**.

- As we forecast further into the future, the forecast error variance increases, but it does so at a **constant rate** determined by the model's **innovation variance** and autoregressive parameters.
- For an AR model, the forecast variance **tends toward a constant long-run value** as the AR effects decay over multiple forecast steps.
- At any given forecast horizon h , the conditional forecast variance is **the same across time** because the model does not adjust to recent volatility.

Conditional vs. Unconditional Variance in Mean-Only Models:

- When we estimate only the mean (e.g., AR, MA, or ARMA models), the innovations ε_t are assumed to have a constant conditional variance:

$$\text{Var}(\varepsilon_t | \Omega_{t-1}) = \sigma^2.$$

This is the **one-step-ahead conditional variance** of the process.

- However, the **unconditional variance** of the process y_t depends on the ARMA parameters and is typically *larger* than σ^2 . For example, for an AR(1) process $y_t = a_0 + a_1 y_{t-1} + \varepsilon_t$:

$$\text{Var}(y_t) = \frac{\sigma^2}{1 - a_1^2}, \quad |a_1| < 1.$$

- Thus, while the **conditional variance** of the innovation is constant, the **unconditional variance** of y_t depends on the persistence of the AR terms.

2. Conditional Heteroscedasticity (GARCH Model)

In a **GARCH** or **AR–GARCH** model, the conditional variance changes dynamically based on past shocks and volatility. This introduces *time-varying conditional heteroscedasticity*.

- The **conditional variance evolves over time**, reflecting periods of high or low volatility.
- Forecast intervals can **expand or contract** depending on recent volatility, unlike the steady pattern in homoscedastic models.
- After large shocks, the conditional variance rises sharply, capturing **volatility clustering** — the tendency for volatility to persist once it increases.

3. Comparing Conditional and Unconditional Variances

- In a **homoscedastic AR/ARMA model**, the innovation variance $\text{Var}(\varepsilon_t | \Omega_{t-1}) = \sigma^2$ is **constant**. The unconditional variance of the process y_t depends on the model parameters (e.g., $\sigma^2/(1 - a_1^2)$ for AR(1)).
- In a **GARCH model**, the conditional variance varies over time:
 - After periods of high volatility (e.g., financial crises), the conditional variance may temporarily exceed the unconditional (long-run) variance.
 - As volatility stabilizes and no new shocks occur, the conditional variance **reverts to the unconditional variance** — its long-run equilibrium level.

Summary

- **AR(p) / ARMA models:** assume constant innovation variance σ^2 ; forecast intervals widen steadily as the horizon increases, approaching the unconditional variance of y_t .
- **GARCH models:** allow the conditional variance to change over time; forecast intervals adapt dynamically to recent shocks and volatility patterns.
- The GARCH framework captures **volatility clustering** — periods of high volatility followed by high volatility, and calm periods followed by calm — a defining feature of financial time series.

```
In [58]: ugarchforecast(garch.dapple, n.ahead = 5)
```

```
*-----*
*      GARCH Model Forecast      *
*-----*
Model: sGARCH
Horizon: 5
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=2025-10-31]:
  Series   Sigma
T+1 0.0013380 0.01556
T+2 0.0009887 0.01578
T+3 0.0016840 0.01599
T+4 0.0032059 0.01620
T+5 0.0008327 0.01641
```

Exercise 2

Estimate ARMA-GARCH(1,1) model for BMW returns and produce 5 periods ahead forecasts

Solution to Exercise 2

```
In [59]: library(rugarch)
spec = ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model     = list(armaOrder = c(2, 1), include.mean = TRUE),
  distribution.model = "norm"
)
garch.dbmw = ugarchfit(spec, dbmw)
garch.dbmw@fit$matcoef
```

A matrix: 7 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(> t)
mu	0.0003354191	7.900173e-05	4.245718	2.178944e-05
ar1	1.0058036073	1.265258e-02	79.493952	0.000000e+00
ar2	-0.0331980801	1.271876e-02	-2.610167	9.049801e-03
ma1	-0.9822924603	2.264151e-05	-43384.575524	0.000000e+00
omega	0.0000033698	1.737948e-06	1.938953	5.250700e-02
alpha1	0.0535447804	8.530577e-03	6.276806	3.455982e-10
beta1	0.9386469657	1.005459e-02	93.355114	0.000000e+00

Interpretation of the ARMA(2,1)-GARCH(1,1) Model for BMW Returns

The table reports the estimation results for an **ARMA(2,1)-GARCH(1,1)** model fitted to the series *dbmw* (BMW's returns). The ARMA component captures the short-run dynamics in the mean equation, while the GARCH component models the time-varying conditional

variance (volatility).

Mean Equation (ARMA(2,1))

- **μ (constant):** The mean return is estimated at **0.00034** and is statistically significant ($p < 0.01$), indicating that BMW's average daily return is positive and different from zero.
- **AR terms:**
 - **ar1 = 1.006***:** Positive and highly significant, suggesting strong persistence in returns — today's return is highly correlated with yesterday's return.
 - **ar2 = -0.033**:** Negative and statistically significant, indicating a mild corrective effect from returns two days ago. This pattern implies that returns tend to slightly revert after a strong positive movement.
- **MA term:**
 - **ma1 = -0.982***:** Large, negative, and highly significant. This value nearly cancels out the AR(1) effect, a common feature in near-integrated ARMA structures. It suggests that shocks to returns are quickly corrected, resulting in short-term mean-reverting dynamics.

Variance Equation (GARCH(1,1))

- **ω (omega = $3.37 \times 10^{-6}**$):** The constant term is small but significant ($p < 0.05$), representing the long-run average variance — a positive baseline level of volatility.
- **α_1 (alpha1 = 0.053***):** Measures the immediate impact of past shocks (ARCH effect). Its significance indicates that large shocks increase current volatility, showing short-term volatility clustering.
- **β_1 (beta1 = 0.938***):** Measures volatility persistence (GARCH effect). The large and significant coefficient (close to 1) implies that volatility is highly persistent — once volatility increases, it tends to remain elevated for a long period.

Conclusion: The ARMA(2,1)-GARCH(1,1) model captures both the persistence in BMW's returns and the pronounced volatility clustering typical of financial time series. The large and significant **β_1 (0.938)** indicates **high volatility persistence**, while the smaller but significant **α_1 (0.053)** confirms that new shocks have an immediate but less dominant impact on volatility. Overall, BMW's return volatility is **time-varying, persistent, and influenced by recent shocks**, making the GARCH(1,1) structure appropriate for modeling this series.

```
In [60]: ugarchforecast(garch.dbmw, n.ahead = 5)

*-----*
*      GARCH Model Forecast      *
*-----*

Model: sGARCH
Horizon: 5
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=2025-10-31]:
  Series   Sigma
T+1 0.0004587 0.01697
T+2 0.0007409 0.01700
T+3 0.0007392 0.01703
T+4 0.0007280 0.01707
T+5 0.0007169 0.01710
```

Relationship Between Conditional and Unconditional Variance in an ARMA–GARCH Model

The *conditional variance* in an ARMA–GARCH model can be **smaller or larger** than the *unconditional variance* at any given time — because it changes dynamically in response to past shocks.

However, over time, as we take the **expectation** of the conditional variance (i.e., average across all periods), it **converges to the unconditional variance**, provided the GARCH model is **covariance-stationary**, meaning:

$$\alpha_1 + \beta_1 < 1$$

Intuition for GARCH(1,1)

- **Conditional variance (σ_t^2):** Varies over time — increases after large shocks and decreases during calm periods.
- **Unconditional variance ($E[\sigma_t^2]$):** The long-run average level of volatility implied by the model, given by:

$$\text{Var}(r_t) = \frac{\omega}{1 - \alpha_1 - \beta_1}$$

Even though at any moment:

$$\sigma_t^2 \neq \frac{\omega}{1 - \alpha_1 - \beta_1},$$

the **expected value** of σ_t^2 equals that unconditional variance in the long run.

