ECON60332 Coursework Template

Group 11

Participant's Student ID	Indicate if the student did not participate
10710007	

Theoretical exercise

Question a

When comparing the ARCH(2) model : $\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2$) to a GARCH(1,1)) : $\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \beta_2\sigma_{t-1}^2$)

The following similarities emerge :

- Both models capture the cluster effect where periods of high (low) volatility are followed by periods of high (low) volatility (probabilistically)
- Both model capture overkurtosis, accounting for a heavier tail distribution than the standard norm
- If z_t is not symmetric, then $\text{Cov}(\varepsilon_t, \sigma_{t+1}^2|\mathcal{F}_{t-1})$ would not be zero, and both model are able to capture leverage (asymmetric) effects

Though with key differences that

- An arch model implies an AR model for the squared residuals whilst a GARCH model implies an ARMA model in both squared residuals and variance
- Fir the GARCH model b_1 can be unidentified if $a_1 = 0$

Question b

Assuming $\alpha_1 + \alpha_2 < 1$ for stationarity

Where $\mathcal{F}_{t-1}$ is the past history of the process, then the conditional expectation of ε_t is given by:

$$E(\varepsilon_t|\mathcal{F}_{t-1}) = E(\sqrt{\sigma_t^2}z_t|\mathcal{F}_{t-1}) = \sigma_t E(z_t|\mathcal{F}_{t-1}) = \sigma_t \cdot 0 = 0 \tag{1}$$

Using LIE, we can obtain the unconditional mean :

$$E(\varepsilon_t) = E(E(\varepsilon_t|\mathcal{F}_{t-1})) = E(0) = 0 \tag{2}$$

To derive the second moment, we can express ε_t^2 as an AR(22)-process:

$$\begin{aligned}\sigma_t^2 + \varepsilon_t^2 &= \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \varepsilon_t^2 \\ \varepsilon_t^2 &= \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \nu_t\end{aligned}$$

with $\nu_t = \varepsilon_t^2 - \sigma_t^2$ being a WN process.

It holds that:

$$E(\varepsilon_t^2|\mathcal{F}_{t-1}) = E(\sigma_t^2 z_t^2|\mathcal{F}_{t-1}) = \sigma_t^2 E(z_t^2|\mathcal{F}_{t-1}) = \sigma_t^2 \cdot 1 = \sigma_t^2 \tag{3}$$

Such that:

$$E(\nu_t|\mathcal{F}_{t-1}) = E(\varepsilon_t^2|\mathcal{F}_{t-1}) - \sigma_t^2 = \sigma_t^2 - \sigma_t^2 = 0 \tag{4}$$

Therefore:

$$E(\nu_t) = E(E(\nu_t|\mathcal{F}_{t-1})) = E(0) = 0 \tag{5}$$

Which allows us to write:

$$\begin{aligned} E(\varepsilon_t^2) = \omega + \alpha_1 E\left[\varepsilon_{t-1}^2\right] + \alpha_2 E\left[\varepsilon_{t-2}^2\right] + E\left[\nu_t\right] \\ = \omega + \alpha_1 E\left[\varepsilon_{t-1}^2\right] + \alpha_2 E\left[\varepsilon_{t-2}^2\right] \end{aligned}$$

Using stationarity, $E(\varepsilon_{t-1}^2) = E(\varepsilon_t^2)$ and hence we conclude that:

$$\mathrm{Var}(\varepsilon_t) = E(\varepsilon_t^2) - (E(\varepsilon_t))^2 = E(\varepsilon_t^2) = \omega + \alpha_1 \mathrm{Var}(\varepsilon_{t-1}) - \alpha_2 \mathrm{Var}(\varepsilon_{t-2}) = \frac{\omega}{1 - \alpha_1 - \alpha_2} = \frac{\omega}{1 - \sum_{i=1}^{2} \alpha_i}$$

Question c

Using the covariance definition $\mathrm{Cov}(X, Y) = E[XY] - E[X]E[Y]$ and given the conditional variance $\sigma_{t+1}^2 = \omega + \alpha_1 \varepsilon_t^2 + \alpha_2 \varepsilon_{t-1}^2$, we derive the covariance as follows:

$$\begin{aligned} \mathrm{Cov}(\sigma_{t+1}^2, \varepsilon_t|\mathcal{F}_{t-1}) &= E[\sigma_{t+1}^2 \varepsilon_t|\mathcal{F}_{t-1}] - E[\sigma_{t+1}^2|\mathcal{F}_{t-1}]E[\varepsilon_t|\mathcal{F}_{t-1}] \\ &= E[\sigma_{t+1}^2 \varepsilon_t|\mathcal{F}_{t-1}] - 0 \\ &= E\left[\left(\omega + \alpha_1 \varepsilon_t^2 + \alpha_2 \varepsilon_{t-1}^2\right) \varepsilon_t|\mathcal{F}_{t-1}\right] \\ &= \omega E[\varepsilon_t|\mathcal{F}_{t-1}] + \alpha_1 E[\varepsilon_t^3|\mathcal{F}_{t-1}] + \alpha_2 E[\varepsilon_{t-1}^2 \varepsilon_t|\mathcal{F}_{t-1}] \quad \text{(since } \omega \text{ and } \varepsilon_{t-1}^2 \text{ don't depend on } \varepsilon_t\text{)} \\ &= 0 + \alpha_1 E[\varepsilon_t^3|\mathcal{F}_{t-1}] + 0 \quad \text{(since } \varepsilon_t \text{ is conditionally zero-mean and independent of } \varepsilon_{t-1}\text{).} \end{aligned}$$

Now, if we consider $\varepsilon_t = \sigma_t z_t$:

$$\begin{aligned} \mathrm{Cov}(\sigma_{t+1}^2, \varepsilon_t|\mathcal{F}_{t-1}) = \alpha_1 E[\varepsilon_t^3|\mathcal{F}_{t-1}] &= \alpha_1 E[(\sigma_t z_t)^3|\mathcal{F}_{t-1}] \\ &= \alpha_1 \sigma_t^3 E[z_t^3|\mathcal{F}_{t-1}] \\ &= \alpha_1 \left(\omega + \alpha_1 \varepsilon_t^2 + \alpha_2 \varepsilon_{t-1}^2\right)^{\frac{3}{2}} E[z_t^3]. \end{aligned}$$

Only with non-zero third moment on z_t, is the model able to capture the leverage (asymmetric) effect. Since z_t $\mathcal{N}(0, 1)$, the third moment is 0 and $\mathrm{Cov}(\sigma_{t+1}^2, \varepsilon_t|\mathcal{F}_{t-1}) = 0$ so the model cannot capture the leverage effect, which is the tendency for negative shocks to have a different impact on volatility than positive shocks.

Question d

With our ARCH(2) model, the 1 step ahead forecast is : $\sigma_{t+1|t}^2 = \omega + \alpha_1 \varepsilon_t^2 + \alpha_2 \varepsilon_{t-1}^2$

Analogously the 2 step ahead forecast, where we take expectation of ε_t^2, and use the unit variance of z_t to simplify

$$\sigma_{t+2|t}^2 = E_t\left[\omega + \alpha_1 \varepsilon_{t+1}^2 + \alpha_2 \varepsilon_t^2\right] = \omega + \alpha_1 \sigma_{t+1|t}^2 + \alpha_2 \sigma_t^2$$

Similarly for the 3-step ahead forecast, following the same logic

$$\sigma_{t+3|t}^2 = E_t\left[\omega + \alpha_1 \varepsilon_{t+2}^2 + \alpha_2 \varepsilon_{t+1}^2\right] = \omega + \alpha_1 \sigma_{t+2|t}^2 + \alpha_2 \sigma_{t+1}^2 \equiv E\left[\sigma_{t+3}^2|F_t\right]$$

Question e

The news impact curve illustrates how a shock to ε_{t-1} affects σ_t^2.

We derive the unconditional long-term variance $\bar{\sigma}^2$ assuming that $E(\varepsilon_t^2)$ at any point is equal to $\bar{\sigma}^2$. From the stationarity condition:

$$E(\varepsilon_t^2) = \omega + \alpha_1 E(\varepsilon_{t-1}^2) + \alpha_2 E(\varepsilon_{t-2}^2). \tag{6}$$

Given $E(\varepsilon_{t-1}^2) = E(\varepsilon_{t-2}^2) = \bar{\sigma}^2$, the equation simplifies to:

$$\bar{\sigma}^2 = \frac{\omega}{1 - \alpha_1 - \alpha_2}. \tag{7}$$

The news impact curve for the ARCH(2) model is then set up as:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \bar{\sigma}^2. \tag{8}$$

With the most recent shock ε_{t-1} highlighted, we rewrite it as:

$$\sigma_t^2 = A + \alpha_1 \varepsilon_{t-1}^2, \tag{9}$$

where $A = \omega + \alpha_2 \bar{\sigma}^2$

Thus with an asymmetric distribution, we can derive the NIC as :

$$\text{NIC}(\varepsilon_{t-1} | \sigma_{t-1}^2 = \sigma_Y^2) = \omega + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \sigma_Y^2. \tag{10}$$

Practical exercise

Question a

	Sample moments		Test statistic	P-value
Mean	0.10	Mean	0.83	0.40
Standard deviation	1.83	Skewness	-2.08	0.04
Skewness	-0.32	Kurtosis	4.06	0.00
Kurtosis	4.26	JB	20.85	0.00

LBQ test results:

	Returns	Squared returns
Test statistic	27.06	25.61
P-value	0.17	0.22

Plot of Daily Log Returns

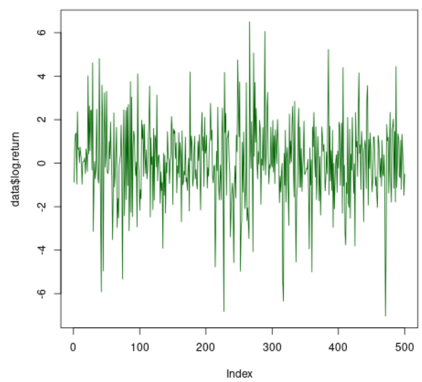

Figure 1: Log Return Plot

SACF and SPACF

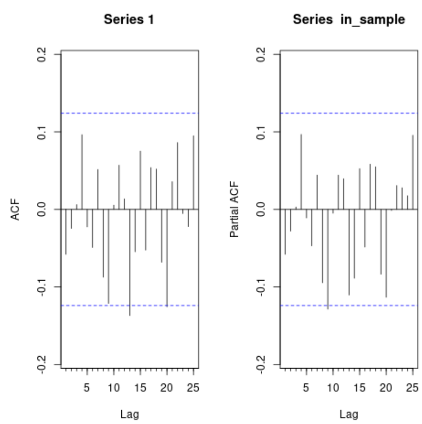

Figure 2: Sample Autocorrelation and Partial Autocorrelation Function Plot

Interpretation

Looking at the Descriptive Statistics, a skewness of -0.32 suggests there are more extreme negative returns than positive returns. Whilst a slightly higher kurtosis of 4.26 suggests the distribution of returns has heavier tails and more extreme values.

Looking at the Test Statistics, testing mean returns are zero ($H_0 : \mu = 0$ against $H_1 : \mu \neq 0$) with a one-sided test, we obtain a p-value of 0.40, meaning we fail to reject H_0 at all conventional significance levels, we do not have enough evidence to conclude the mean return is zero.

Testing the skewness of the return series is zero ($H_1 : \gamma = 0$ against $H_1 : \gamma \neq 0$) with a one-sided test, we obtain a p-value of 0.04 and thus we reject the null hypothesis in favour of the alternative hypothesis that the distribution of returns is significantly skewed, and hence non-symmetric.

Testing for the leptokurtic property in the returns ($H_0 : \kappa = 3$ against $H_1 : \kappa \neq 3$) using a one-sided test, we obtain a p value of 0.00, thus we reject the null hypothesis in favour of the alternative hypothesis that the returns have a significantly different kurtosis from 3, confirming the presence of heavy tails in the distribution.

Using the Jargue-Bera test for normality ($H_0 : \gamma = 0 \& \kappa = 3$ against $H_0 : \gamma \neq 0 \& \kappa \neq 3$) we obtain a p-value of 0.00, thus we reject the null hypothesis in favour of the alternative hypothesis that the distribution of returns differs significantly from normality, as evidenced by the skewness and kurtosis.

Using the LBQ test for the autocorrelation of the returns ($H_0 : \rho_1 = \rho_2 = \ldots = \rho_{21} = 0$ against $H_1 : \rho_i \neq 0$ for some $i \in \{1, 2, \ldots, 21\}$), we obtain a p-value of 0.17 and 0.22 for the squared returns, hence we do not reject the null hypothesis at any conventional significance level, and thus there is no significant evidence of autocorrelation in the returns of the series.

Whilst the Daily Log-returns do not display a clear trend or seasonal pattern, though the volatility appears to be clustered in certain periods, indicative of heteroskedacity, there is no consistent pattern of significant lags in both the SACF and SPACF. Though a significant negative correlation at lag 13 alongside lags approaching significance at 9 and 20 suggests some evidence for a seasonal pattern, however the presence of significant lags mainly inform us the time series is not white noise and exhibits some autocorrelation.

Question b

GARCH			GJR-GARCH		
Parameter	Estimate	P-value	Parameter	Estimate	P-value
ω	0.16	0.22	ω	0.16	0.32
α	0.05	0.09	α	0.05	0.24
β	0.91	0.00	β	0.91	0.00
GARCH-t			γ	-0.00	0.97
ω	0.11	0.35	GJR-GARCH-t		
α	0.07	0.15	ω	0.13	0.42
β	0.91	0.00	α	0.06	0.30
ν	4.88	0.00	β	0.91	0.00
			γ	0.02	0.78
			ν	4.84	0.00

Interpretation:

For the GARCH model, a significant β at the 1% level suggests past volatility is very important to current volatility, whilst a significant α at the 10% level suggests lagged squared returns are important in current volatility under a standard normal distribution. Under a student-t distribution, the relationship on β holds though α loses significance after the inclusion of ν, which suggests heavy tails are important in the GARCH-t model, rather than lagged squared returns which may have been masking the effect under a normal distribution.

For the GJR-GARCH model, β remains significant at the 1% level, suggesting past volatility remains important to current volatility upon the inclusion of the asymmetric impact of shocks. Although a negative γ suggests positive shocks to returns increase future volatility more than negative shocks. Still, the α parameter becomes less certain with a larger p-value Under a student's t-distribution for the GJR-GARCH model, β and ν are significant, indicating the presence of heavy tails in the data is signifiant that could be important in modelling the current variance when accounting for asymmetries. Since they are similar in magnitude to the GARCH-t model a distribution with heavier tails is motivated, whilst asymmetric effects are not in the data.

Question c

Plot of NIC

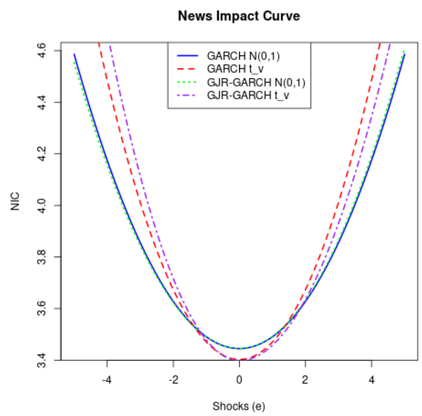

Figure 3: News Impact Curve Plot

Interpretation

In fig. 3 we can see that the GJR news impact curve captures the asymmetry in the effect of news on volatility since it has a steeper slope on its positive side than its negative compared to the GARCH model which inherently does not capture the asymmetry. The student's t-distribution in GJR-GARCH combines both the asymmetry of shocks (leverage effect) and the reduced impact of outliers, highlighting the importance of the leverage effect - properties which may enable better fitting of empirical returns. Thus, in a market uptick like AN AI boom, under a GJR-GARCH-t model the impact of the asymmetry and heavy tails is more pronounced for the GJR-GARCH, so positive shocks to returns will increase future volatility more than negative shocks, which is counter to the usual leverage effect observed in financial markets.

Question d

GARCH	Test statistic	P-value
Z	25.18	0.24
Z^2	20.73	0.48
GARCH-t	Test statistic	P-value
Z	24.65	0.26
Z^2	20.45	0.49

GJR-GARCH	Test statistic	P-value
Z	25.17	0.24
Z^2	20.70	0.48
GJR-GARCH-t	Test statistic	P-value
Z	24.70	0.26
Z^2	20.76	0.47

Interpretation:

Though there are high p-values for the residuals across all models, indicating there is no statistical evidence to reject the null hypothesis that the residuals follow the respective distributions (and are thus white noise (normally distributed with no autocorrelation))

We are mainly interested in the p-values for the squared residuals to identify volatility clustering or ARCH effects Accordingly, the model with the highest p-value for the square residuals is the GARCH-t model, indicating the least amount of autocorrelation is the GJR-GARCH-t. Though we would want to verify this with model selection criteria rather than a box test.

Question e

	GARCH	GARCH-t	GJR-GARCH	GJR-GARCH-t
RMSFE	7.32	7.29	7.32	7.30
DM Test statistic	NA	2.00	-0.03	1.83
P-value	NA	0.98	0.49	0.97

Interpretation:

Interpreting the results, the GARCH-t is the best model with the lowest RMSFE at 7.29. Although the Diebold-Mariano test suggests that the original GARCH model may have better forecast accuracy, with much higher p-values than conventional significance levels (e.g., 0.05 or 0.10), there's no statistical evidence that the forecast accuracy of GARCH-t and GJR-GARCH-t is any different from GARCH.

Empirically, this suggests that a more parsimonious, simpler GARCH model may have the edge over more complex models for forecasting this particular variance, as they do not provide statistically significant improvements in forecast accuracy.

Question f

Interpretation:

The choice of GARCH model involves a tradeoff between model misspecification and estimation noise. On one hand, simpler models like the standard GARCH have fewer parameters and are less prone to estimation noise, but may suffer from misspecification if the true volatility process exhibits features that the model cannot capture (e.g., asymmetry, time-varying volatility).

More flexible models like the GJR-GARCH or GARCH with a student's t-distribution (GARCH+) could potentially fit the data better by accounting for these complexities such as positive leverage effects, and hence reducing misspecification. However, this requires estimating more parameters which inherently introduces greater estimation noise.

While complex models may offer an improved in-sample fit, there is no guarantee for this in out-of-sample forecasting due to the curse of dimensionality resulting in increased estimation noise. The benefits of reduced misspecification may be offset by the noise from estimating additional parameters, especially in small samples.

Ultimately, model selection criteria such as AIC and BIC should be used to inform the tradeoff between model misspecification and estimation noise. By penalising overly complex models, Simple GARCH models are preferred since they are more stable when estimating and thus have less estimation noise.

Appendix

For reproduction of said script, see `https://github.com/oddish3/FE-CW/tree/master`

```r
# ==============================================================================
# FInancaial Economitrics Coursework
# ==============================================================================
#
# Author: 10710007
# Version: 13-03-2024
#
# ==============================================================================

rm(list = ls())
# Packages
library(lubridate)
library(forecast)
library(rugarch)
# Data
setwd("/home/oddish3/Documents/R_folder/MSc/FE/FE-coursework/code")
data = read.csv("../data/group_11.csv")
source("fineco_fun.R")
source("../utils/latex-macro.R")
figures_path <- "../docs/figures/"


# Script
# ==============================================================================
data$date = as.Date(data$date)
# plot(data$log.return, type = "l", col = "darkgreen")
# abline(h = 0, v = 250, col = "red")
# dev.off()


# ------------------------------------------
#          Practical Exercise
# ------------------------------------------

# Use the first 250 observations as an in-sample (estimation) period and the last 250 observations as out of
     sample forecasting
in_sample = as.matrix(data[1:250, 2])
out_sample = as.matrix(data[(251:nrow(data)), 2])

# a) investigating statistical properties of the in-sample data ----
# i) descriptive stats
a1_results = dstats(in_sample)

# ii-v) moment tests
a2_results = test_moment(in_sample)

# Assemble data for moments and test statistics into a dataframe
moment_test_df = data.frame(
  # descriptive stats
  amu = a1_results[1,1],
  asigma = a1_results[2,1],
  askew = a1_results[3,1],
  akurt = a1_results[4,1],
  # t stats
  amut = a2_results[1,1],
  askewt = a2_results[2,1],
  akurtt = a2_results[3,1],
  ajbt = a2_results[4,1],
  # p vals
  amup = a2_results[1,2],
  askewp = a2_results[2,2],
  akurtp = a2_results[3,2],
  ajbp = a2_results[4,2]
```

```r
)
# Appending the second section with its title
write_latex("../results/results.tex", moment_test_df, decimal_precision = 2, append = FALSE, section_title = "Moment Test and Descriptive Statistics")

# vi) lbq test
lbq1 = Box.test(in_sample, lag = 21, type = "Ljung-Box", fitdf = 0)
lbq2 = Box.test(in_sample^2, lag = 21, type = "Ljung-Box", fitdf = 0) # fitdf is the number of parameters estimated ???

lbq_df = data.frame(
  aistat = lbq1$statistic,
  aip = lbq1$p.value,
  aiistat = lbq2$statistic,
  aiip = lbq2$p.value
)
# Writing the first section with its title
write_latex("../results/results.tex", lbq_df, decimal_precision = 2, append = TRUE, section_title = "Ljung-Box Test Results")

# vii) Plot SACF and SPACF
# Open a PNG device for ACF and PACF plots
png(paste0(figures_path, "PACF.png"))

# Setting the plotting area to accommodate two plots side by side
par(mfrow = c(1, 2))

# Generate ACF and PACF plots
Acf(in_sample, lag.max = 25)
Pacf(in_sample, lag.max = 25)

# Close the plotting device for ACF/PACF
dev.off()

# Reset par settings to default for subsequent plots
par(mfrow = c(1, 1))

# Now, generate and save another plot separately if needed
png(paste0(figures_path, "log_return_plot.png"))
plot(data$log.return, type = "l", col = "darkgreen")
dev.off()

# b) estimating the conditional variance ----
# Assume that the conditional mean of the return series is constant
#Use the et series to estimate the following conditional variance

# i) GARCH(1,1) with zt ~ N(0,1)
spec1 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                   mean.model = list(armaOrder = c(0, 0), include.mean = FALSE) ,distribution.model = "norm")

fit1 = ugarchfit(spec = spec1, data = in_sample)

estimates1 = fit1@fit$robust.matcoef[,1] # This extracts the "Estimate" column
p_values1 = fit1@fit$robust.matcoef[,4] # This extracts the "Pr(>|t|)" column


# GARCH(1,1) with zt ~ tv
spec2 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
                  distribution.model = "std") # 'std' for Student's t-distribution
fit2 = ugarchfit(spec = spec2, data = in_sample)

estimates2 = fit2@fit$robust.matcoef[,1] # This extracts the "Estimate" column
p_values2 = fit2@fit$robust.matcoef[,4] # This extracts the "Pr(>|t|)" column

# ii) GJR-GARCH (1, 1) with zt ~ N(0,1)
spec3 = ugarchspec(
  variance.model = list(model = "gjrGARCH", garchOrder = c(1,1)),
```

```
  mean.model = list(armaOrder = c(0,0), include.mean = FALSE),
  distribution.model = "norm" # Standard normal distribution for innovations
)

fit3 = ugarchfit(spec = spec3, data = in_sample)
estimates3 = fit3@fit$robust.matcoef[,1] # This extracts the "Estimate" column
p_values3 = fit3@fit$robust.matcoef[,4] # This extracts the "Pr(>|t|)" column

# GJR-MODEL (1, 1) with zt ~ tv
spec4 = ugarchspec(
  variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std" # 'std' for Student's t-distribution
)
fit4 = ugarchfit(spec = spec4, data = in_sample)
estimates4 = fit4@fit$robust.matcoef[,1] # This extracts the "Estimate" column
p_values4 = fit4@fit$robust.matcoef[,4]

garch11_df = data.frame(
  bw = estimates1["omega"],
  ba = estimates1["alpha1"],
  bb = estimates1["beta1"],
  bpi = p_values1["omega"],
  bpii = p_values1["alpha1"],
  bpiii = p_values1["beta1"]
)

write_latex("../results/results.tex", garch11_df, append = TRUE, section_title = "GARCH(1,1) with zt ~ N(0,1)")

garch11_t_df = data.frame(
  bwi = estimates2["omega"],
  bai = estimates2["alpha1"],
  bbi = estimates2["beta1"],
  bvi = estimates2["shape"],
  bpti = p_values2["omega"],
  bptii = p_values2["alpha1"],
  bptiii = p_values2["beta1"],
  bptiv = p_values2["shape"]
)
write_latex("../results/results.tex", garch11_t_df, append = TRUE, section_title = "GARCH(1,1) with zt ~ tv")

garch_gjr_df = data.frame(
  bwii = estimates3["omega"],
  baii = estimates3["alpha1"],
  bbii = estimates3["beta1"],
  bgii = estimates3["gamma1"],
  bpgi = p_values3["omega"],
  bpgii = p_values3["alpha1"],
  bpgiii = p_values3["beta1"],
  bpgiv = p_values3["gamma1"]
)

write_latex("../results/results.tex", garch_gjr_df, append = TRUE, section_title = "GJR-GARCH(1,1) with zt ~
    N(0,1)")

garch_gjr_t_df = data.frame(
  bwiii = estimates4["omega"],
  baiii = estimates4["alpha1"],
  bbiii = estimates4["beta1"],
  bgiii = estimates4["gamma1"],
  bviii = estimates4["shape"],
  bpgtp = p_values4["omega"],
  bpgtpi = p_values4["alpha1"],
  bpgtpii = p_values4["beta1"],
  bpgtpiii = p_values4["gamma1"],
  bpgtpiv = p_values4["shape"]
)

```

```
write_latex("../results/results.tex", garch_gjr_t_df, append = TRUE, section_title = "GJR-GARCH(1,1) with zt ~ tv")

# c) plotting NIC ----
# For GARCH(1,1) with zt ~ N(0,1)
w1 = estimates1["omega"]
a1 = estimates1["alpha1"]
b1 = estimates1["beta1"]

# For GARCH(1,1) with zt ~ tv
w2 = estimates2["omega"]
a2 = estimates2["alpha1"]
b2 = estimates2["beta1"]
v2 = estimates2["shape"]

# For GJR-GARCH(1,1) with zt ~ N(0,1)
w3 = estimates3["omega"]
a3 = estimates3["alpha1"]
b3 = estimates3["beta1"]
g3 = estimates3["gamma1"]

# For GJR-GARCH(1,1) with zt ~ tv
w4 = estimates4["omega"]
a4 = estimates4["alpha1"]
b4 = estimates4["beta1"]
g4 = estimates4["gamma1"]
v4 = estimates4["shape"]

# Unconditional variance from the first GARCH(1,1) model
ve = w1 / (1 - a1 - b1)

# NIC ----
T = 500
e = seq(-5, 5, length.out = T) # Grid of shocks epsilon
nicG1 = nicG2 = nicGJR1 = nicGJR2 = rep(0, T) # Initialize NIC for each model

# Calculate NIC for each model
for (t in 1:T) {
  nicG1[t] = w1 + b1 * ve + a1 * e[t]^2 # GARCH(1,1) with zt ~ N(0,1)
  nicG2[t] = w2 + b2 * ve + a2 * e[t]^2 # GARCH(1,1) with zt ~ tv

  if (e[t] > 0) {
    nicGJR1[t] = w3 + b3 * ve + a3 * e[t]^2 # GJR-GARCH(1,1) with zt ~ N(0,1)
    nicGJR2[t] = w4 + b4 * ve + a4 * e[t]^2 # GJR-GARCH(1,1) with zt ~ tv
  }
  else{
    nicGJR1[t] = w3 + b3 * ve + (a3 + g3) * e[t]^2 # GJR-GARCH(1,1) with zt ~ N(0,1)
    nicGJR2[t] = w4 + b4 * ve + (a4 + g4) * e[t]^2 # GJR-GARCH(1,1) with zt ~ tv
  }
}

# Plot NIC for all four models
par(mfrow = c(1, 1))
png(paste0(figures_path, "NIC.png"))
plot(e, nicG1, type = "l", col = "blue", ylab = 'NIC', xlab = 'Shocks (e)', lwd = 2, lty = 1, main = "News Impact Curve")
lines(e, nicG2, type = "l", col = "red", lwd = 2, lty = 2)
lines(e, nicGJR1, type = "l", col = "green", lwd = 2, lty = 3)
lines(e, nicGJR2, type = "l", col = "purple", lwd = 2, lty = 4)
legend("top", legend = c("GARCH N(0,1)", "GARCH t_v", "GJR-GARCH N(0,1)", "GJR-GARCH t_v"), col = c("blue", "red", "green", "purple"), lty = 1:4, ncol = 1, lwd = 2)
dev.off()

# d) best model by analysing standarised residuals ----

# Extract standardized residuals from each model
# Conduct LBQ test for each model (21 lags) residuals
# Conduct LBQ test for each model (21 lags) squared residuals
```

```r

# For GARCH(1,1) with zt ~ N(0,1)
std_resid_norm = fit1@fit[["residuals"]] / sqrt(fit1@fit[["sigma"]] )
lbq_z_norm = Box.test(std_resid_norm, lag = 21, type = "Ljung-Box")
lbq_z_norm2 = Box.test(std_resid_norm^2, lag = 21, type = "Ljung-Box")

# For GARCH(1,1) with zt ~ tv
std_resid_t = fit2@fit[["residuals"]] / sqrt(fit2@fit[["sigma"]] )
lbq_z_t = Box.test(std_resid_t, lag = 21, type = "Ljung-Box")
lbq_z_t2 = Box.test(std_resid_t^2, lag = 21, type = "Ljung-Box")

# For GJR-GARCH(1,1) with zt ~ N(0,1)
std_resid_gjr_norm = fit3@fit[["residuals"]] / sqrt(fit3@fit[["sigma"]] )
lbq_z_gjr_norm = Box.test(std_resid_gjr_norm, lag = 21, type = "Ljung-Box")
lbq_z_gjr_norm2 = Box.test(std_resid_gjr_norm^2, lag = 21, type = "Ljung-Box")

# For GJR-GARCH(1,1) with zt ~ tv
std_resid_gjr_t = fit4@fit[["residuals"]] / sqrt(fit4@fit[["sigma"]] )
lbq_z_gjr_t = Box.test(std_resid_gjr_t, lag = 21, type = "Ljung-Box")
lbq_z_gjr_t2 = Box.test(std_resid_gjr_t^2, lag = 21, type = "Ljung-Box")


df_test = data.frame(
  # garch N(0,1)
  zone = lbq_z_norm$statistic,
  pone = lbq_z_norm$p.value,
  zfive = lbq_z_norm2$statistic,
  pfive = lbq_z_norm2$p.value,
  # garch t
  ztwo = lbq_z_t$statistic,
  ptwo = lbq_z_t$p.value,
  zsix = lbq_z_t2$statistic,
  psix = lbq_z_t2$p.value,
  # gjr N(0,1)
  zthree = lbq_z_gjr_norm$statistic,
  pthree = lbq_z_gjr_norm$p.value,
  zseven = lbq_z_gjr_norm2$statistic,
  pseven = lbq_z_gjr_norm2$p.value,
  # gjr t
  zfour = lbq_z_gjr_t$statistic,
  pfour = lbq_z_gjr_t$p.value,
  zeight = lbq_z_gjr_t2$statistic,
  peight = lbq_z_gjr_t2$p.value
)

write_latex("../results/results.tex", df_test, append = TRUE, section_title = "residuals and squared lbq")

# e) 1 step 2ahead forecasting the conditional variance ----

H = 250
T = length(data$log.return) - H
f1 = f2 = f3 = f4 = matrix(0, H, 1) # Initialize forecast for each model

for (i in 1:H) {
  window = data$log.return[i:(T+i-1)]

  fit.g11 = ugarchfit(spec = spec1, data = window, solver = 'hybrid')
  fit.tg11 = ugarchfit(spec = spec2, data = window, solver = 'hybrid')
  fit.gj11 = ugarchfit(spec = spec3, data = window, solver = 'hybrid')
  fit.tgj11 = ugarchfit(spec = spec4, data = window, solver = 'hybrid')

  # forecast
  xx = ugarchforecast(fit.g11, data = window, n.ahead = 1)
  f1[i] = xx@forecast$sigmaFor

  xx = ugarchforecast(fit.tg11, data = window, n.ahead = 1)
  f2[i] = xx@forecast$sigmaFor

```

```
  xx = ugarchforecast(fit.gj11, data = window, n.ahead = 1)
  f3[i] = xx@forecast$sigmaFor

  xx = ugarchforecast(fit.tgj11, data = window, n.ahead = 1)
  f4[i] = xx@forecast$sigmaFor

  print(i)
}

# Forecast errors
e1 = f1 - data$log.return[(T+1):(T+H)]^2
e2 = f2 - data$log.return[(T+1):(T+H)]^2
e3 = f3 - data$log.return[(T+1):(T+H)]^2
e4 = f4 - data$log.return[(T+1):(T+H)]^2

# RMSFE
rmsfe = function(e) {
  sse = sum(e^2) / length(e)
  r = sqrt(sse)
  return(r)
}

# dm test
dm_test_12 = dm.test(e1, e2, alternative = "less", h = 1, power = 2, varestimator = "acf")
dm_test_13 = dm.test(e1, e3, alternative = "less", h = 1, power = 2, varestimator = "acf")
dm_test_14 = dm.test(e1, e4, alternative = "less", h = 1, power = 2, varestimator = "acf")

# Extract DM test statistics and p-values
dm_stat_12 = dm_test_12$statistic
p_value_12 = dm_test_12$p.value

dm_stat_13 = dm_test_13$statistic
p_value_13 = dm_test_13$p.value

dm_stat_14 = dm_test_14$statistic
p_value_14 = dm_test_14$p.value


rmsfe_dm_df = data.frame(
  #rmsfe
  rmsfei = rmsfe(e1),
  rmsfeii = rmsfe(e2),
  rmsfeiii = rmsfe(e3),
  rmsfeiv = rmsfe(e4),
  #dm test
  dm = dm_stat_12,
  dmpii = p_value_12,
  dmi = dm_stat_13,
  dmpiv = p_value_13,
  dmii = dm_stat_14,
  dmpv = p_value_14

)
write_latex("../results/results.tex", rmsfe_dm_df, append = TRUE, section_title = "root mean square forecast error and dm test")



# End of Script
# ==============================================================================
```