ECON60332 Coursework Template

Group 11

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

Theoretical exercise

Question a

Comparing the extended GARCH model, given by:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2, \tag{1}$$

with the canonical GARCH(1,1) model:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \quad 0 \leq \alpha_1, \beta_1 \leq 1, \quad (\alpha_1 + \beta_1) < 1, \tag{2}$$

we elucidate both foundational similarities and pivotal differences. Both frameworks integrate the premise that current volatility, σ_t^2, is a function of past squared residuals (ε^2 and ε^2), addressing the volatility clustering phenomenon observed in financial markets.

The augmented GARCH(2,1) model introduces an additional term, $\alpha_2 \varepsilon_{t-2}^2$, enhancing its capability to encapsulate longer-term volatility dependencies relative to the GARCH(1,1) model. This inclusion is aimed at capturing a broader spectrum of volatility dynamics, thereby potentially increasing forecasting precision for series exhibiting complex volatility patterns.

The parameter constraints in the GARCH(1,1) model, specifically the conditions $0 \leq \alpha_1, \beta_1 \leq 1$ and $(\alpha_1 + \beta_1) < 1$, are crucial for ensuring the model's stability and preventing the persistence of shocks from generating unbounded variance. These constraints necessitate careful consideration in the extended model to maintain its predictive integrity and prevent overfitting.

Empirically, selecting between these models is contingent on the observed dynamics within the financial time series of interest. While the extended GARCH(2,1) model may offer superior forecast accuracy for series with nuanced volatility patterns, this comes at the expense of increased model complexity and the risk of overfitting. This dichotomy accentuates the significance of empirical validation and judicious model selection in the realm of financial econometrics.

Question b

The unconditional mean of the error term, $E[\varepsilon_t]$ is given by

$$E[\varepsilon_t] = E[z_t] = E\left[\sqrt{\sigma_t^2}\right] = 0 \cdot E[\sigma_t] = 0$$

Since by definition of $z_t \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, $E[z_t] = 0$ follows a standard normal distribution.

To derive the unconditional variance, we express ε_t^2 as an AR(2) process

$$
\begin{aligned}
\sigma_t^2 + \varepsilon_t^2 &= \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \varepsilon_t^2 \\
\varepsilon_t^2 &= \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + (\varepsilon_t^2 - \sigma_t^2) \\
\Rightarrow \varepsilon_t^2 &= \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2 + \nu_t \quad \text{where } \nu_t = \varepsilon_t^2 - \sigma_t^2 \text{ is a white noise process}
\end{aligned}
$$

Thus, we can rewrite

$$
\begin{aligned}
&V(\varepsilon_t) \\
E[\varepsilon_t] &= E\left[\frac{\varepsilon_t^2}{\sigma_t^2}\right] = 0 \quad \text{given } |\alpha| < 1 \\
V(\varepsilon_t) &= E[\varepsilon_t^2] \\
&= E[\omega + \alpha\varepsilon_{t-1}^2 + \nu_t] \\
&= \omega + \alpha E[\varepsilon_{t-1}^2] + E[\nu_t] \\
&= \omega + \alpha E[\varepsilon_{t-1}^2] \\
E[\varepsilon_{t-1}^2] - \alpha E[\varepsilon_{t-1}^2] &= \omega \\
E[\varepsilon_{t-1}^2](1-\alpha) &= \omega \\
E[\varepsilon_{t-1}^2] &= \frac{\omega}{1-\alpha}
\end{aligned}
$$

Question c

Using $\text{Cov}(X,Y) = E[XY] - E[X]E[Y]$ we derive the covariance, where $\sigma_{t+1}^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \alpha_2\varepsilon_{t-2}^2)\varepsilon_t$

$$
\text{Cov}(\sigma_{t+1}^2, \varepsilon_t | F_{t-1}) = E[\sigma_{t+1}^2\varepsilon_t | F_{t-1}] - E[\sigma_{t+1}^2 | F_{t-1}]E[\varepsilon_{t-1} | F_{t-1})
$$

Given that ε_t has a conditional expectation of 0, the second term becomes 0, leaving

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

Then, since ω is a constant and the conditional mean of $\varepsilon_t = 0$ (previously shown), the first term disappears. Then, simplifying further, since σ_t^2 is $\mathcal{F}_{t-1}$deterministic We can use the property $E[XY \mid \mathcal{F}_{t-1}] = E[X \mid \mathcal{F}_{t-1}]E[Y \mid \mathcal{F}_{t-1}]$ for independent random variables X and Y:

$$
E[\alpha_2\sigma_t^2\varepsilon_t \mid \mathcal{F}_{t-1}] = \alpha_2\sigma_t^2 E[\varepsilon_t \mid \mathcal{F}_{t-1}] = 0 \tag{3}
$$

Therefore, we obtain:

$$
\alpha_1 E[\varepsilon_t^3 \mid \mathcal{F}_{t-1}] \tag{4}
$$

Which we can rewrite as

$$
\begin{aligned}
a_1 E\left[zt^3\right] E\left[(\omega + \alpha_1\varepsilon_t^2 + \alpha_2\varepsilon_{t-1}^2)^{\frac{3}{2}} \mid \mathcal{F}_{t-1}\right] \\
= \alpha_1\left(\omega + \alpha_1\varepsilon_t^2 + \alpha_2\varepsilon_{t-1}^2)\right)^{\frac{3}{2}} E\left[z_t^3\right]
\end{aligned}
$$

35:52 GARCH

Question d

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

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

$$\sigma^2_{t+2|t} = E_t\left[\omega + \alpha_1\varepsilon^2_{t+1} + \alpha_2\varepsilon^2_t\right] = \omega + \alpha_1\sigma^2_{t+1|t} + \alpha_2\sigma^2_t$$

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

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

Question e

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

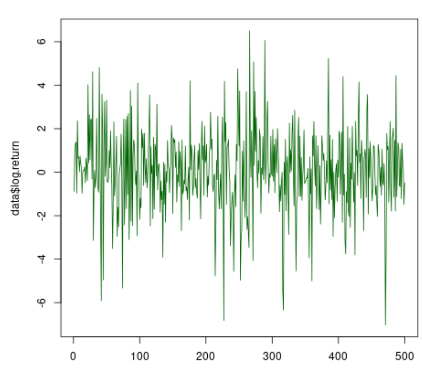

Figure 1: Log Return Plot

SACF and SPACF

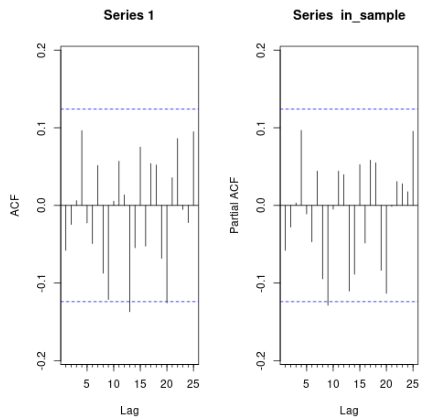

Figure 2: Sample Autocorrelation and Partial Autocorrelation Function Plot

Interpretation

Descriptive Statistics

Mean : the sample mean is 0.10

Standard Deviation : the standard deviation is 1.83, indicating a relatively high volatility of the data.

Skewness : the skewness value of -0.32 suggests that the distribution of returns is slightly negatively skewed, meaning there are more extreme negative returns than positive returns.

Kurtosis : the kurtosis value of 4.26 is higher than the normal distribution value of 3, indicating that the distribution of returns has heavier tails and more extreme values when compared to a normal distribution.

Test Statistics

Testing the significance of the mean return, we set up the following hypothesis and use a one sided test :

- The null hypothesis, the mean return is equal to zero ($H_0 : \mu = 0$) against
- The alternative hypothesis that the mean is not equal to zero $H_1 : \mu \neq= 0$.

The decision rule is that we reject H_0 if the p-value is less than the significance level α (EG, 0.05). The test statistic obtained is 0.83 with a p-value of 0.40, thus we fail to reject H_0 at all conventional significance levels. We do not have enough evidence to conclude the mean return is significantly different from zero, and cannot reject the possibility that the true mean return is equal to zero.

Testing the significance of skewness in the returns, we set up the following hypothesis and one sided test :

- The null hypothesis, the skewness of the return series is zero ($H_1 : \gamma = 0$), indicating a symmetric distribution. Against
- The alternative hypothesis, the skewness of the financial time series returns is not zero ($\gamma \neq= 0$), indicating a non-symmetric distribution

The decision rule is to reject H_0 if the p value is less than the significance level α (EG, 0.05). The test statistic obtained is -2.08 with a p-value of 0.04. Since the p-value is less than 0.05, we reject the null hypothesis (H_0) in favour of the alternative hypothesis. Thus, the distribution of returns is significantly skewed, and hence not symmetric. Testing for kurtosis (leptokurtic property) in the returns, we set up the following hypothesis and one sided test:

- The null hypothesis, the kurtosis of the returns equals the normal distribution value of 3 ($H_0 : \kappa = 3$), indicating a normal distribution in terms of tail thickness. Against,
- The alternative hypothesis, the kurtosis of the financial time series returns is significantly different from 3 ($H_1 : \kappa \neq 3$), indicating a non-normal distribution in terms of tail thickness

The decision rule is to reject H_0 if the p value is less than the significance level α (EG, 0.05). The test statistic obtained is 4.26 and p value of 0.00, thus we reject H_0, in favour of the alternative hypothesis. Thus, the returns have significantly different kurtosis from 3, confirming the presence of heavy tails in the distribution.

Using the Jargue-Bera test for normality, we set up the following hypothesis and 1 sided test :

- The null hypothesis, the returns follow a normal distribution, implying that both the skewness and kurtosis of the series equal those of a normal distribution ($H_0 : \gamma = 0 \& \kappa = 3$). Against,
- The alternative hypothesis, that the returns do not follow a normal distribution, meaning either the or both the skewness and kurtosis significantly differ from those of a normal distribution ($H_0 : \gamma \neq 0 \& \kappa \neq 3$).

The decision rule is to reject H_0 if the p-value is less than the significance level α (EG, 0.05). The outcome of the test since the JB test statistic is 20.85 with a p-value of 0.00. Therefore, we reject H_0 in favour of H_1, that the distribution of returns differs significantly from normality, evidence by its skewness and kurtosis values.

Ljung-Box Test

Testing the autocorrelation of the financial time series returns, we use the Ljung-Box Q-test with the following hypotheses:

- The null hypothesis, which states that there is no autocorrelation in the series up to a certain number of lags ($H_0 : \rho_1 = \rho_2 = \ldots = \rho_{21} = 0$), where ρ represents autocorrelation at different lags.
- The alternative hypothesis, which suggests that there is some autocorrelation in the series at least at one lag ($H_1 : \rho_i \neq 0$ for some $i \in \{1, 2, \ldots, 21\}$).

The decision rule is to reject H_0 if the p-value is less than the significance level α (e.g., 0.05).

For the returns, the test statistic obtained is 27.06 with a p-value of 0.17.

Since the p-value is greater than all conventional significance levels, we do not reject the null hypothesis. Thus, there is no significant evidence of autocorrelation in the returns of the series.

For squared returns, the test statistic obtained is 25.61 with a p-value of 0.22.

Similarly, since the p-value is greater than 0.05, we do not reject the null hypothesis for squared returns at all conventional significance levels either. This indicates no significant evidence of autocorrelation in the volatility (squared returns) of the series.

SACF SPACF plots

The SACF plot measures the correlation between different points in the time series separated by various lags. Whilst most autocorrelations are within the confidence intervals, a significant negative correlation at lag 13 suggests there is a season pattern that repeats every 13 periods, so if a series is above average at one point, it tends to be below average 13 periods later and vice versa. Furthermore, a lag on the border of significance at 20 suggests a possible longer cynical effect, although this is not as pronounced. This is also evident for lag 9 since it is just below significance in the SACF but is significant in the SPACF, indicating a direct negative influence from the observation 9 periods ago on the current observation, after accounting for the influences of all observations in between. In summary, there is no consistent pattern of significant lags, which would typically be used to identify AR or MA components. The presence of significant lags informs us the time series is not white noise and exhibits autocorrelation.

Daily Log-returns

The plot indicates considerable fluctuation around the mean of 0.10, whilst the returns do not display a clear trend or seasonal pattern. The volatility appears to be clustered in certain periods, indicative of heteroskedacity where periods of high volatility are followed by high volatility and vice versa.

Question b

GARCH			GJR-GARCH		
Parameter	Estimate	P-value	Parameter	Estimate	P-value
ω	2.53	0.54	ω	3.06	0.02
α	0.13	0.03	α	0.00	1.00
β	0.15	0.90	β	0.00	1.00
GARCH-t			γ	0.29	0.19
ω	0.24	0.58	GJR-GARCH-t		
α	0.07	0.22	ω	0.22	0.26
β	0.87	0.00	α	0.02	0.57
ν	4.62	0.00	β	0.89	0.00
			γ	0.07	0.13
			ν	4.57	0.00

Interpretation:

Where ω is the constant term of the model, representing the long run average variance when all other terms are zero, α is the coefficient representing the contribution of past squared innovations (lagged error terms) to the current variance, indicating how much past volatility affects current volatility. β is the coefficient representing the contribution of past conditional variance to the current variance, capturing the persistence of volatility shocks. γ is the coefficient specific to GJR-GARCH models, capturing the asymmetric effect of negative shocks (leverage effect), where negative shocks have a different impact on volatility than positive shocks of the same magnitude. ν is the degrees of freedom parameter in the t-distribution and is related to the kurtosis of the distribution, with lower values indicating heavier tails.

For the GARCH model with a normal distribution, the estimates for α and β are 2.53 and 0.15, respectively with p values indicating that only α is statistically significant at a conventional level ($p < 0.05$). Thus the model suggests that past shocks have a significant impact on current volatility, but the effect is not persistent.

For the GARCH-t model, β is signifiant, indicating persistence in volatility and ν is also significant, suggesting that the distribution of innovations has heavier tails than the normal distribution. Thus, the presence of heavy tails in the data is signifiant, which could be important fore forecasting ...

For the GJR-GARCH model, ω is significant, but α and β are very small with correspondingly very large p-values. Thus, negative shocks might have a different impact on volatility, although this effect is not statistically significant at the 5% level

For the GJR-GARCH-t model, both α and β are significant, indicating that past shocks and volatility are important for current volatility, and ν is significant, indicating heavy tails. Although, γ is not significant, suggesting the asymmetric effects of shocks is not statistically significant. Thus, both past shocks and heavy tails are significant in modelling volatility but asymmetric effects of shocks are not significant.

Overall, a garch-t or GJR-GARCH-t model might be preferred

Question c

Plot of NIC

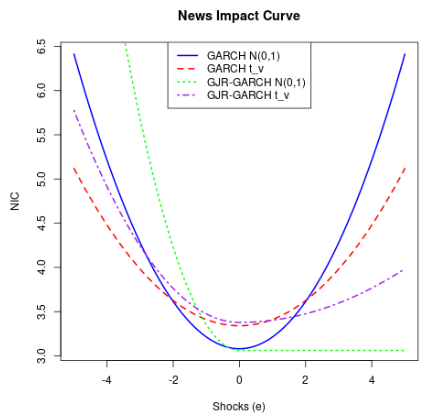

Figure 3: News Impact Curve Plot

Interpretation:

Drawbacks of the GARCH models are obvious here, since the conditional variance is unable to respond asymmetrically to shocks in y_t, that is, a positive return has the same effect as a negative return upon variance. Since it is argued that negative innovations to shock returns tend to increase volatility more than positive innovations of the same magnitude.

This plot offers a visual representation of the effect that new information has on the volatility predicted by different GARCH models. The standard GARCH model with normal innovations shows that the impact on conditional variance is symmetrical, both positive and negative shocks of the same magnitude have the same effect on predicted volatility, leaving the leverage effect unaccounted for.

The GARCH model with a student's t-distribution is also symmetric but flatter compared to the standard GARCH model, indicating less sensitivity to shocks in general, due to the heavier tails in the t-distribution, reducing the impact of outliers.

The GJR-GARCH model with normal distribution shows a pronounced asymmetry where negative shocks increase the conditional variance more than positive shocks. The model accounts for the leverage effect, where bad news influences volatility more than good news.

The GJR-GARCH model with Student's t-distribution combines the properties of the GJR-GARCH with the heavier tails of the t-distribution. Showing an asymmetrical impact of shocks on volatility, similar to the GJR-GARCH but with the additional effect of heavier tails in the distribution of shocks. Heavier tails may imply that extreme values are more probable than in a normal distortion, and the impact curve is less steep for large magnitude shocks, indicating large shocks increase volatility less than what a normal distribution would suggest. Overall, both models with Student's t-distribution suggests that the models account for heavy tails in the distribution of shocks, whilst the GJR-GRACH models reveal an asymmetric response to shocks, highlighting the importance of the leverage effect.

Question d

GARCH	Test statistic	P-value
Z	12.80	0.92
Z^2	16.87	0.72
GARCH-t	Test statistic	P-value
Z	12.39	0.93
Z^2	16.21	0.76

GJR-GARCH	Test statistic	P-value
Z	13.09	0.91
Z^2	18.23	0.63
GJR-GARCH-t	Test statistic	P-value
Z	13.04	0.91
Z^2	17.85	0.66

Interpretation:

High p-values for the residuals indicate there is no statistical evidence to reject the null hypothesis that the residuals follow the respective distributions. Suggesting the residuals are white noise, meaning they are normally distributed with no autocorrelation.

For the squared residuals, the p-values are high but slightly less so, again suggesting that there is no statistical evidence to reject the null hypothesis of no autocorrelation in the squared residuals. Indicating there is no ARCH effect and the conditional variance is well captured by the model.

Whilst the LBQ p-values for the residuals are very high across all models, suggesting that none of the models leaves unexplained autocorrelation in the returns, which means that all models are adequate in this respect.

The p-values for the squared residuals, which help to identify volatility clustering or ARCH effects, are also high across all models. However, in this context, the model with the highest p-value (corresponding to lowest test stat) for the square residuals is the GARCH-t model, indicating the least amount of autocorrelation and possibly the best fit among the compared models.

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:

The RMSFE values indicate that GARCH+ has the smallest forecast error at 7.29, followed by GJR-GARCH-t at 7.30, GARCH at 7.32, and GJR-GARCH at 7.32. Lower RMSFE values suggest better forecast accuracy.

Diebold-Mariano (DM) Test results are only meaningful for GARCH+ and GJR-GARCH-t since GARCH is the benchmark. For GARCH+, the DM Test statistic is 2.00 with a p-value of 0.98, and for GJR-GARCH-t, the statistic is 1.83 with a p-value of 0.97. Since the p-values are much higher than the typical significance levels (e.g., 0.05 or 0.10), there's no statistical evidence that the forecast accuracy of GARCH+ and GJR-GARCH-t is different from GARCH.

Thus, although GARCH+ has a slightly lower RMSFE, the Diebold-Mariano test does not confirm its superiority over the benchmark GARCH model in terms of predictive accuracy. Economically, this suggests that there might be no practical benefit from using more complex models over the simpler GARCH model for forecasting this particular variance, as they do not provide statistically significant improvements in forecast accuracy.

Question f

Interpretation:

In terms of the trade-off between model misspecification and estimation noise, models with more parameters, like GARCH+ and GJR-GARCH-t, can potentially fit the data better and capture more complex structures in volatility, thus reducing model misspecification. However, they also introduce more estimation noise due to the increased number of parameters that need to be estimated.

The simpler GARCH model, with fewer parameters, may be more robust to estimation noise but might suffer from model misspecification if the true variance process has features like asymmetry or time-varying volatility that it cannot capture.

Therefore, while more complex models can offer a better fit, they might not necessarily result in better out-of-sample forecasting performance due to the additional estimation noise. The choice of model should balance the risks of misspecification against the potential for increased noise from estimating more parameters.

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 = data$log.return)

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 = data$log.return)

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 = data$log.return)
estimates3 = fit3@fit$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 = data$log.return)
estimates4 = fit4@fit$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"]
)

```

```r
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 from tutorial ----
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
```

```
# 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

```

```r
  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
# ==============================================================================
```