# 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 $\varepsilon_t$ 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 $\varepsilon_t^2$ as an AR(22)-process:

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

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

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:

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

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

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 $\varepsilon_t^2$, 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 $\varepsilon_{t-1}$ affects $\sigma_t^2$.

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 $\varepsilon_{t-1}$ 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 :

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

# 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:

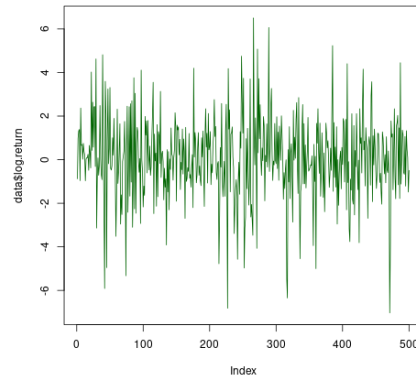|  | Returns | Squared returns |
|---|---|---|
| Test statistic | 27.06 | 25.61 |
| P-value | 0.17 | 0.22 |

**Plot of Daily Log Returns**



Figure 1: Log Return Plot
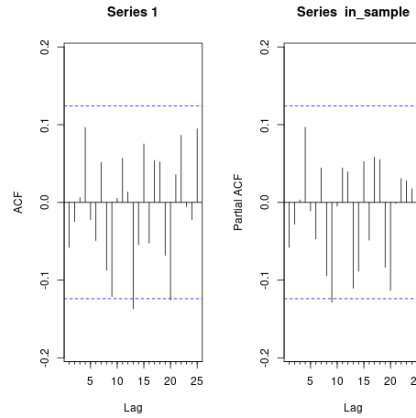
**SACF and SPACF**



Figure 2: Sample Autocorrelation and Partial Autocorrelation Function Plot

## Interpretation

Looking at the Descriptive Statistics, a skewness of -0.32 suggests that the distribution of returns negatively skewed, meaning there are more extreme negative returns than positive returns. Whilst a kurtosis of 4.26 is slightly higher than a normal distribution (3), suggesting 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. Obtaining 0.83 with a p-value of 0.40, we fail to reject $H_0$ at all conventional significance levels, since we do not have enough evidence to conclude the mean return is significantly different from zero.

Testing the skewness of the return series is zero ($H_1 : \gamma = 0$ against $H_1 : \gamma \neq= 0$) We obtain -2.08 with a p-value of 0.04, since the p-value is less than 0.05, we reject the null hypothesis in favour of the alternative hypothesis. Thus, the distribution of returns is significantly skewed, and hence non-symmetric. Testing for leptokurtic property in the returns ($H_0 : \kappa = 3$ against $H_1 : \kappa \neq 3$) We obtain 4.26 and p value of 0.00, thus we reject $H_0$, in favour of the alternative hypothesis that the returns have 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 $H_0 : \gamma \neq 0 \& \kappa \neq 3$)

Obtaining 20.85 with a p-value of 0.00, 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.

Testing the autocorrelation of the financial time series returns ($H_0 : \rho_1 = \rho_2 = \ldots = \rho_{21} = 0$ against $H_1 : \rho_i \neq 0$ for some $i \in \{1, 2, \ldots, 21\}$)

Obtaining a p-value of 0.17, we do not reject the null hypothesis at all conventional significance levels, thus, there is no significant evidence of autocorrelation in the returns of the series. We proceed analogously for the squared returns, arriving at the same conclusion, indicating no significant evidence of autocorrelation in the volatility (squared returns) of the series.

For the plots, there is no consistent pattern of significant lags in both plots, however 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 informs us the time series is not white noise and exhibits some autocorrelation.

The Daily Log-returns plot does not display a clear trend or seasonal pattern, although the volatility appears to be clustered in certain periods, indicative of heteroskedacity.

## Question b

| GARCH | | | GJR-GARCH | | |
|---|---|---|---|---|---|
| Parameter | Estimate | P-value | Parameter | Estimate | P-value |
| $\omega$ | 2.53 | 0.54 | $\omega$ | 3.06 | 0.02 |
| $\alpha$ | 0.13 | 0.03 | $\alpha$ | 0.00 | 1.00 |
| $\beta$ | 0.15 | 0.90 | $\beta$ | 0.00 | 1.00 |
| GARCH-t | | | $\gamma$ | 0.29 | 0.19 |
| $\omega$ | 0.24 | 0.58 | GJR-GARCH-t | | |
| $\alpha$ | 0.07 | 0.22 | $\omega$ | 0.22 | 0.26 |
| $\beta$ | 0.87 | 0.00 | $\alpha$ | 0.02 | 0.57 |
| $\nu$ | 4.62 | 0.00 | $\beta$ | 0.89 | 0.00 |
| | | | $\gamma$ | 0.07 | 0.13 |
| | | | $\nu$ | 4.57 | 0.00 |

**Interpretation:**

Where $\omega$ is the constant term of the model, representing the long run average variance when all other terms are zero, $\alpha$ 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. $\beta$ is the coefficient representing the contribution of past conditional variance to the current variance, capturing the persistence of volatility shocks. $\gamma$ 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. $\nu$ 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 $\alpha$ and $\beta$ are 2.53 and 0.15, respectively with p values indicating that only $\alpha$ 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, $\beta$ is signifiant, indicating persistence in volatility and $\nu$ 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, $\omega$ is significant, but $\alpha$ and $\beta$ 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 $\alpha$ and $\beta$ are significant, indicating that past shocks and volatility are important for current volatility, and $\nu$ is significant, indicating heavy tails. Although, $\gamma$ 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

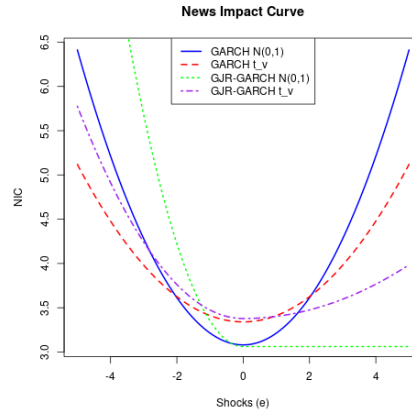## Question c

**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:**

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 additional components (GARCH+) can potentially better fit the data by accounting for these complexities, reducing misspecification. However, they require estimating more parameters, introducing greater estimation noise.

While complex models offer an improved in-sample fit, they do not necessarily outperform simpler models in out-of-sample forecasting due to the increased estimation noise. The benefits of reduced misspecification may be offset by the noise from estimating additional parameters, especially in small samples.

Ultimately, the optimal model balances the competing goals of parsimony and flexibility. Simple models are preferred when estimation noise is a greater concern, while more complex specifications are justified if the true data-generating process exhibits features that warrant additional modeling components, even at the cost of some estimation noise.

Model selection criteria like information criteria aim to navigate this tradeoff by penalizing overly complex models. However, the final choice depends on the relative importance of in-sample fit versus out-of-sample forecasting performance for the specific application.

# Appendix

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

```r
1   # ==============================================================================
2   # FInancaial Economitrics Coursework
3   # ==============================================================================
4   #
5   # Author: 10710007
6   # Version: 13-03-2024
7   #
8   # ==============================================================================
9
10  rm(list = ls())
11  # Packages
12  library(lubridate)
13  library(forecast)
14  library(rugarch)
15  # Data
16  setwd("/home/oddish3/Documents/R_folder/MSc/FE/FE-coursework/code")
17  data = read.csv("../data/group_11.csv")
18  source("fineco_fun.R")
19  source("../utils/latex-macro.R")
20  figures_path <- "../docs/figures/"
21
22
23  # Script
24  # ==============================================================================
25  data$date = as.Date(data$date)
26  # plot(data$log.return, type = "l", col = "darkgreen")
27  # abline(h = 0, v = 250, col = "red")
28  # dev.off()
29
30
31  # ------------------------------------------
32  #            Practical Exercise
33  # ------------------------------------------
34
35  # Use the first 250 observations as an in-sample (estimation) period and the last 250 observations as out of
          sample forecasting
36  in_sample = as.matrix(data[1:250, 2])
37  out_sample = as.matrix(data[(251:nrow(data)), 2])
38
39  # a) investigating statistical properties of the in-sample data ----
40  # i) descriptive stats
41  a1_results = dstats(in_sample)
42
43  # ii-v) moment tests
44  a2_results = test_moment(in_sample)
45
46  # Assemble data for moments and test statistics into a dataframe
47  moment_test_df = data.frame(
48    # descriptive stats
49    amu = a1_results[1,1],
50    asigma = a1_results[2,1],
51    askew = a1_results[3,1],
52    akurt = a1_results[4,1],
53    # t stats
54    amut = a2_results[1,1],
55    askewt = a2_results[2,1],
56    akurtt = a2_results[3,1],
57    ajbt = a2_results[4,1],
58    # p vals
59    amup = a2_results[1,2],
60    askewp = a2_results[2,2],
61    akurtp = a2_results[3,2],
62    ajbp = a2_results[4,2]
```

```r
63  )
64  # Appending the second section with its title
65  write_latex("../results/results.tex", moment_test_df, decimal_precision = 2, append = FALSE, section_title =
        "Moment Test and Descriptive Statistics")
66
67  # vi) lbq test
68  lbq1 = Box.test(in_sample, lag = 21, type = "Ljung-Box", fitdf = 0)
69  lbq2 = Box.test(in_sample^2, lag = 21, type = "Ljung-Box", fitdf = 0) # fitdf is the number of parameters
        estimated ???
70
71  lbq_df = data.frame(
72    aistat = lbq1$statistic,
73    aip = lbq1$p.value,
74    aiistat = lbq2$statistic,
75    aiip = lbq2$p.value
76  )
77  # Writing the first section with its title
78  write_latex("../results/results.tex", lbq_df, decimal_precision = 2, append = TRUE, section_title = "Ljung-Box
        Test Results")
79
80  # vii) Plot SACF and SPACF
81  # Open a PNG device for ACF and PACF plots
82  png(paste0(figures_path, "PACF.png"))
83
84  # Setting the plotting area to accommodate two plots side by side
85  par(mfrow = c(1, 2))
86
87  # Generate ACF and PACF plots
88  Acf(in_sample, lag.max = 25)
89  Pacf(in_sample, lag.max = 25)
90
91  # Close the plotting device for ACF/PACF
92  dev.off()
93
94  # Reset par settings to default for subsequent plots
95  par(mfrow = c(1, 1))
96
97  # Now, generate and save another plot separately if needed
98  png(paste0(figures_path, "log_return_plot.png"))
99  plot(data$log.return, type = "l", col = "darkgreen")
100 dev.off()
101
102 # b) estimating the conditional variance ----
103 # Assume that the conditional mean of the return series is constant
104 #Use the et series to estimate the following conditional variance
105
106 # i) GARCH(1,1) with zt ~ N(0,1)
107 spec1 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
108                    mean.model = list(armaOrder = c(0, 0), include.mean = FALSE) ,distribution.model = "norm")
109
110 fit1 = ugarchfit(spec = spec1, data = data$log.return)
111
112 estimates1 = fit1@fit$robust.matcoef[,1] # This extracts the "Estimate" column
113 p_values1 = fit1@fit$robust.matcoef[,4] # This extracts the "Pr(>|t|)" column
114
115
116 # GARCH(1,1) with zt ~ tv
117 spec2 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
118                    mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
119                    distribution.model = "std") # 'std' for Student's t-distribution
120 fit2 = ugarchfit(spec = spec2, data = data$log.return)
121
122 estimates2 = fit2@fit$robust.matcoef[,1] # This extracts the "Estimate" column
123 p_values2 = fit2@fit$robust.matcoef[,4] # This extracts the "Pr(>|t|)" column
124
125 # ii) GJR-GARCH (1, 1) with zt ~ N(0,1)
126 spec3 = ugarchspec(
127   variance.model = list(model = "gjrGARCH", garchOrder = c(1,1)),
```

```r
    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"]
)
```

```
195  write_latex("../results/results.tex", garch_gjr_t_df, append = TRUE, section_title = "GJR-GARCH(1,1) with zt ~
         tv")
196
197  # c) plotting NIC ----
198  # For GARCH(1,1) with zt ~ N(0,1)
199  w1 = estimates1["omega"]
200  a1 = estimates1["alpha1"]
201  b1 = estimates1["beta1"]
202
203  # For GARCH(1,1) with zt ~ tv
204  w2 = estimates2["omega"]
205  a2 = estimates2["alpha1"]
206  b2 = estimates2["beta1"]
207  v2 = estimates2["shape"]
208
209  # For GJR-GARCH(1,1) with zt ~ N(0,1)
210  w3 = estimates3["omega"]
211  a3 = estimates3["alpha1"]
212  b3 = estimates3["beta1"]
213  g3 = estimates3["gamma1"]
214
215  # For GJR-GARCH(1,1) with zt ~ tv
216  w4 = estimates4["omega"]
217  a4 = estimates4["alpha1"]
218  b4 = estimates4["beta1"]
219  g4 = estimates4["gamma1"]
220  v4 = estimates4["shape"]
221
222  # Unconditional variance from the first GARCH(1,1) model
223  ve = w1 / (1 - a1 - b1)
224
225  # NIC from tutorial ----
226  T = 500
227  e = seq(-5, 5, length.out = T) # Grid of shocks epsilon
228  nicG1 = nicG2 = nicGJR1 = nicGJR2 = rep(0, T) # Initialize NIC for each model
229
230  # Calculate NIC for each model
231  for (t in 1:T) {
232    nicG1[t] = w1 + b1 * ve + a1 * e[t]^2 # GARCH(1,1) with zt ~ N(0,1)
233    nicG2[t] = w2 + b2 * ve + a2 * e[t]^2 # GARCH(1,1) with zt ~ tv
234
235    if (e[t] > 0) {
236      nicGJR1[t] = w3 + b3 * ve + a3 * e[t]^2 # GJR-GARCH(1,1) with zt ~ N(0,1)
237      nicGJR2[t] = w4 + b4 * ve + a4 * e[t]^2 # GJR-GARCH(1,1) with zt ~ tv
238    }
239    else{
240      nicGJR1[t] = w3 + b3 * ve + (a3 + g3) * e[t]^2 # GJR-GARCH(1,1) with zt ~ N(0,1)
241      nicGJR2[t] = w4 + b4 * ve + (a4 + g4) * e[t]^2 # GJR-GARCH(1,1) with zt ~ tv
242    }
243  }
244
245  # Plot NIC for all four models
246  par(mfrow = c(1, 1))
247  png(paste0(figures_path, "NIC.png"))
248  plot(e, nicG1, type = "l", col = "blue", ylab = 'NIC', xlab = 'Shocks (e)', lwd = 2, lty = 1, main = "News
         Impact Curve")
249  lines(e, nicG2, type = "l", col = "red", lwd = 2, lty = 2)
250  lines(e, nicGJR1, type = "l", col = "green", lwd = 2, lty = 3)
251  lines(e, nicGJR2, type = "l", col = "purple", lwd = 2, lty = 4)
252  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)
253  dev.off()
254
255  # d) best model by analysing standarised residuals ----
256
257  # Extract standardized residuals from each model
258  # Conduct LBQ test for each model (21 lags) residuals
259  # Conduct LBQ test for each model (21 lags) squared residuals
```

```r
260
261  # For GARCH(1,1) with zt ~ N(0,1)
262  std_resid_norm = fit1@fit[["residuals"]] / sqrt(fit1@fit[["sigma"]] )
263  lbq_z_norm = Box.test(std_resid_norm, lag = 21, type = "Ljung-Box")
264  lbq_z_norm2 = Box.test(std_resid_norm^2, lag = 21, type = "Ljung-Box")
265
266  # For GARCH(1,1) with zt ~ tv
267  std_resid_t = fit2@fit[["residuals"]] / sqrt(fit2@fit[["sigma"]] )
268  lbq_z_t = Box.test(std_resid_t, lag = 21, type = "Ljung-Box")
269  lbq_z_t2 = Box.test(std_resid_t^2, lag = 21, type = "Ljung-Box")
270
271  # For GJR-GARCH(1,1) with zt ~ N(0,1)
272  std_resid_gjr_norm = fit3@fit[["residuals"]] / sqrt(fit3@fit[["sigma"]] )
273  lbq_z_gjr_norm = Box.test(std_resid_gjr_norm, lag = 21, type = "Ljung-Box")
274  lbq_z_gjr_norm2 = Box.test(std_resid_gjr_norm^2, lag = 21, type = "Ljung-Box")
275
276  # For GJR-GARCH(1,1) with zt ~ tv
277  std_resid_gjr_t = fit4@fit[["residuals"]] / sqrt(fit4@fit[["sigma"]] )
278  lbq_z_gjr_t = Box.test(std_resid_gjr_t, lag = 21, type = "Ljung-Box")
279  lbq_z_gjr_t2 = Box.test(std_resid_gjr_t^2, lag = 21, type = "Ljung-Box")
280
281
282  df_test = data.frame(
283    # garch N(0,1)
284    zone = lbq_z_norm$statistic,
285    pone = lbq_z_norm$p.value,
286    zfive = lbq_z_norm2$statistic,
287    pfive = lbq_z_norm2$p.value,
288    # garch t
289    ztwo = lbq_z_t$statistic,
290    ptwo = lbq_z_t$p.value,
291    zsix = lbq_z_t2$statistic,
292    psix = lbq_z_t2$p.value,
293    # gjr N(0,1)
294    zthree = lbq_z_gjr_norm$statistic,
295    pthree = lbq_z_gjr_norm$p.value,
296    zseven = lbq_z_gjr_norm2$statistic,
297    pseven = lbq_z_gjr_norm2$p.value,
298    # gjr t
299    zfour = lbq_z_gjr_t$statistic,
300    pfour = lbq_z_gjr_t$p.value,
301    zeight = lbq_z_gjr_t2$statistic,
302    peight = lbq_z_gjr_t2$p.value
303  )
304
305  write_latex("../results/results.tex", df_test, append = TRUE, section_title = "residuals and squared lbq")
306
307  # e) 1 step 2ahead forecasting the conditional variance ----
308
309  H = 250
310  T = length(data$log.return) - H
311  f1 = f2 = f3 = f4 = matrix(0, H, 1) # Initialize forecast for each model
312
313  for (i in 1:H) {
314    window = data$log.return[i:(T+i-1)]
315
316    fit.g11 = ugarchfit(spec = spec1, data = window, solver = 'hybrid')
317    fit.tg11 = ugarchfit(spec = spec2, data = window, solver = 'hybrid')
318    fit.gj11 = ugarchfit(spec = spec3, data = window, solver = 'hybrid')
319    fit.tgj11 = ugarchfit(spec = spec4, data = window, solver = 'hybrid')
320
321    # forecast
322    xx = ugarchforecast(fit.g11, data = window, n.ahead = 1)
323    f1[i] = xx@forecast$sigmaFor
324
325    xx = ugarchforecast(fit.tg11, data = window, n.ahead = 1)
326    f2[i] = xx@forecast$sigmaFor
327
```

```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
# ============================================================================
```