Time Series Modelling

Lab Coursework

MA5629/MA5676: Time Series

Module Leader: *Janeeta Maunthrooa*
Student: *Vasileios Diplas*
Student Number: *2273633*

Janeeta Maunthrooa
Brunel University, London
April 26, 2023

1 Data Visualization

The plot of the Prices of the FTSE 250 stock is shown in Figure 1.

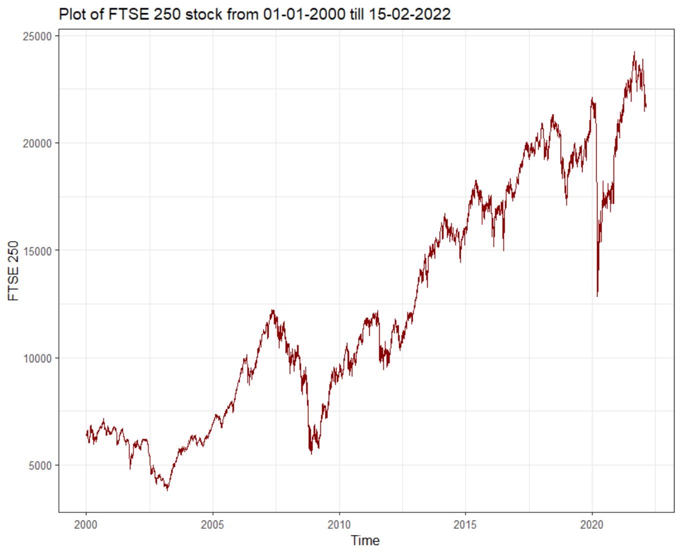

Figure 1: Stock Prices of FTSE 250

Prices of FTSE 250 stock from $\mathscr{T} = 01/01/2000 - 14/02/2022$ have a minimum value of **3802**. The 25% of the observations are below **6954**. The 50% of the observations are below **11181** and the 50% are above **11181**. The mean value of the stock during $\mathscr{T}$ is **12379**. The 75% of the stock Prices are below **17409**. In the end, we found that the maximum value is **24251**.

2 Examination of Stationarity

We observe from Figure 1 that the stock during $\mathscr{T}$ is non-stationary because it is not symmetric around zero (0). Particularly, we observe an upward trend, because prices become higher and higher. We can identify the non-stationarity by applying an Augmented Dickey-Fuller Test (ADF) test. The null hypothesis for the ADF test is that the data is non-stationary. We choose our significance level to be 5%. We found that the **p.value** with lag order 10 for the test is **0.06832056**. We accept the Null Hypothesis and we assume that the data is non-stationary. Assuming that the stock prices are non-stationary, our goal is to transform the Series into Stationary. From theory, we know that a Stationary time series does not depend on the time, at which the series is observed. A time series could be made stationary by a **differencing** method. To achieve stationarity on data, there are transformation techniques that are listed as follows:

- Taking square and cubic root

  We found that for the square and cubic root, the p.values for the Augmented Dickey-Fuller (ADF) test is 0.01. We are confident to claim that the data during $\mathscr{T}$ is stationary in the 5% significance level. Also, we can check the stationarity from Figure 2.

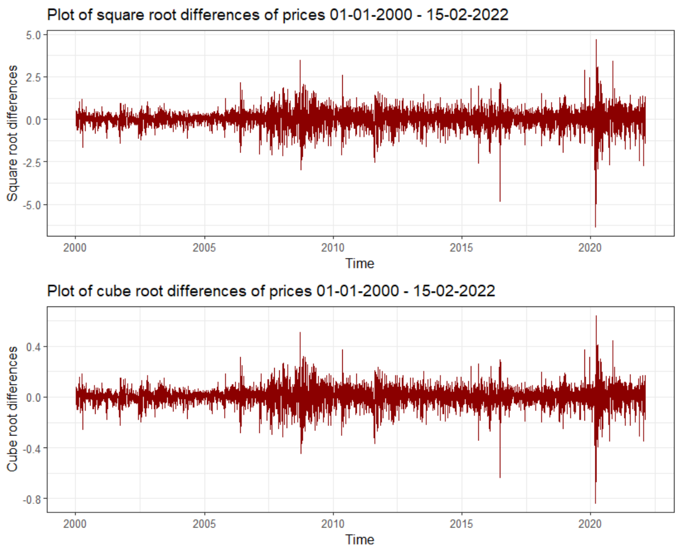

Figure 2: First plot: Squared root differences, Second plot: Cube root differences

- Log transforming data

From theory, we know that the log returns are equal to the differences in log-price data. We found that the p.value for the Augmented Dickey-Fuller (ADF) test is 0.01. We are confident to claim that the data during $\mathscr{T}$ is stationary in the 5% significance level. Also, we can check the stationarity from Figure 3. Along with the plot of differences in log-transformed data, we plotted the squared log returns and the absolute log returns.

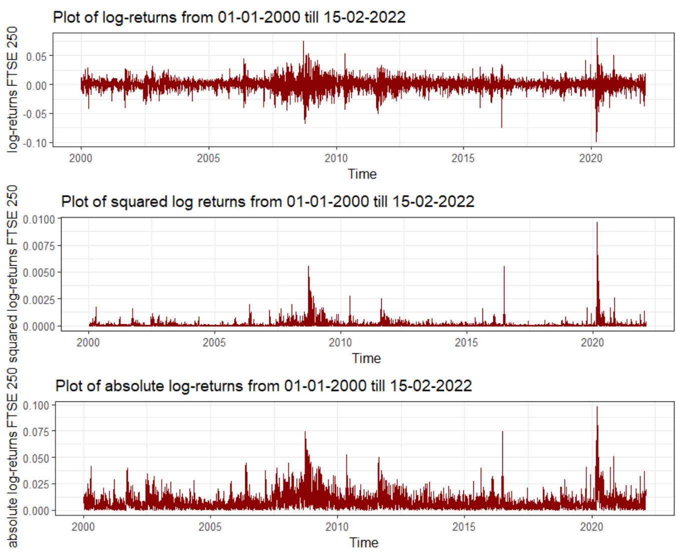

Figure 3: First plot: Log-returns, Second plot: Square Log-returns, Third plot: Absolute log-returns

3 Fitting ARMA-GARCH model

3.1 Summary of log return data

We will do our analysis using log return data. The minimum value of data is **-0.0982014**, the maximum is **0.0803862**, the median is **0.0007883**, and the mean is **0.0002169**. The histogram of log-return data during $\mathscr{T}$ along with the approximation of a Normal curve ($N(0.0002169, 0.0001155)$) is shown in Figure 4.

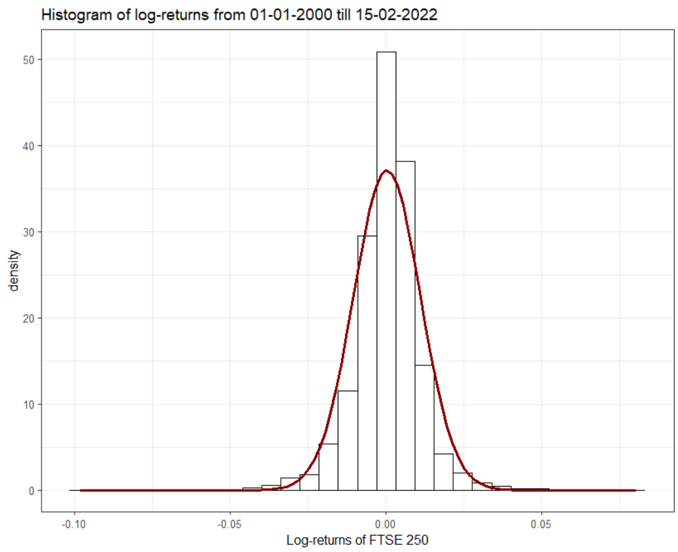

Figure 4: Histogram of log return data and the approximation of Normal curve

We observe that the Normal approximation is good, but there is a better approximation because the normal curve does not approach the peak of empirical data.

Plots of Auto-correlation and Partial Auto-correlation for log returns, squared log returns, and absolute log returns are shown below in Figure 5.

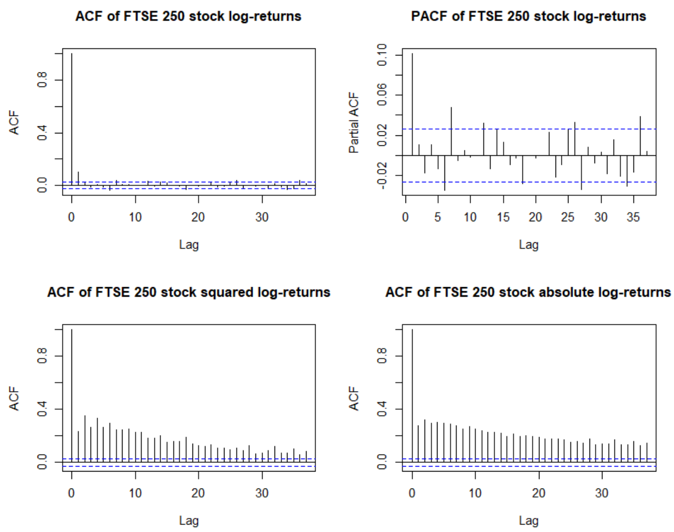

Figure 5: Auto-correlations and Partial Auto-correlation of log-returns, Auto-correlations of squared log returns and absolute log returns

We observe that the Auto-correlation and Partial Auto-correlation plot for log returns has the same pattern with an AR(p) (ARMA(p,0)) model because the auto-correlation curve goes to zero quickly and in partial auto-correlation plot we observe that there are few significance partial auto-correlations. In lag = 1, we observe a strong partial auto-correlation. A possible ARMA model would be the AR(1).

On the other hand, it is clear that auto-correlations for squared and absolute log returns are significant. This means that the squared and absolute log returns do not follow a white noise distribution, so we can claim that there is a GARCH(p,q) model.

3.2 Strategy for fitting an ARMA-GARCH model

We chose to fit 12×12 ARMA-GARCH models. The orders we chose for each process are $\{(k,m), k = 1,2,3 \quad \text{and} \quad m = 0,1,2,3\}$
We decided to run for each GARCH process 12 ARMA process. The best model is the model with the lowest Bayesian Information Criterion (**BIC**) value. From Figure 6 it is clear that the lowest BIC value is for the model ARMA(1,0)-GARCH(1,1).

BIC	GARCH(1,0)	GARCH(1,1)	GARCH(1,2)	GARCH(1,3)	GARCH(2,0)	GARCH(2,1)	GARCH(2,2)	GARCH(2,3)	GARCH(3,0)	GARCH(3,1)	GARCH(3,2)	GARCH(3,3)
ARMA(1,0)	-6.341585	-6.621627	-6.620088	-6.618528	-4.025798	-6.620425	-6.618881	-6.617328	-6.519345	-6.618867	-6.617323	-6.61578
ARMA(1,1)	-6.343271	-6.620314	-6.618774	-6.617214	-6.446477	-6.619083	-6.617539	-6.615984	-6.519386	-6.617526	-6.615983	-6.614445
ARMA(1,2)	-0.517076	-6.618784	-6.61724	-6.61568	-6.450457	-6.617548	-6.616003	-6.614448	-6.517843	-6.61599	-6.614447	-6.612908
ARMA(1,3)	-0.676134	-6.617262	-6.615718	-6.614173	-6.448044	-6.616067	-6.614477	-6.612935	-6.517541	-6.614477	-6.612934	-6.611397
ARMA(2,0)	-6.340049	-6.620344	-6.6188	-6.617237	-6.4503	-6.61911	-6.617566	-6.616007	-6.518953	-6.61755	-6.616006	-6.614484
ARMA(2,1)	0.907207	-6.618809	-6.617264	-6.615701	-6.450447	-6.617572	-6.616029	-6.61447	-6.516875	-6.616013	-6.614469	-6.612937
ARMA(2,2)	-0.328195	-6.617601	-6.616057	-6.614491	-6.44221	-6.616482	-6.614455	-6.613262	-6.51253	-6.614806	-6.613262	-6.611373
ARMA(2,3)	-6.339077	-6.61857	-6.617027	-6.612989	-3.554307	-6.614858	-6.613742	-6.611444	-6.516067	-6.615977	-6.611374	-6.610312
ARMA(3,0)	-1.531122	-6.618777	-6.617233	-6.615689	-6.449609	-6.617542	-6.615998	-6.614456	-6.518384	-6.615998	-6.614456	-6.612931
ARMA(3,1)	-6.339792	-6.617471	-6.615701	-6.614157	-6.446959	-6.61608	-6.614464	-6.612921	-6.517515	-6.614535	-6.612919	-6.611459
ARMA(3,2)	-6.33858	-6.616386	-6.617106	-6.612612	-6.444609	-6.614533	-6.616048	-6.614522	-6.513934	-6.616046	-6.61148	-6.610517
ARMA(3,3)	2.616555	-6.620092	-6.612998	-6.611454	-6.446706	-6.616103	-6.611759	-6.610144	-6.51515	-6.613154	-6.610142	-6.608634

Figure 6: Table of BIC values

With the help of the `rugarch` package in R, we fitted ARMA(1,0)-GARCH(1,1) for the log return data of the FTSE 250 stock and we found that the estimation of coefficients under Normal distribution on residuals is (Table 1):

μ	ϕ_1	ω	a_1	β_1
6.107509e-04	8.463548e-02	2.407180e-06	1.277767e-01	8.508701e-01

Table 1: Coefficients of AR(1)-GARCH(1,1). All coefficients are significant because the p.value is near 0.

3.3 Testing for auto correlation between residuals and square residuals

We would like to examine if the residuals and squared residuals contain auto-correlations. We performed `Box.test` using the Ljung-Box test statistic for different values of lags (`lag=10,lag=20,lag=40`) as in Table 2 and concluded that there are no auto-correlations in residuals. So we are 95% confident of claiming that there is independence in residuals and in squared residuals. It is essential to specify that for residuals the number of degrees of freedom is 1+0 = 1 (ARMA(1,0)) and for squared residuals 1+1 = 2 (GARCH(1,1)).

Data	`lag = 10`	`lag = 20`	`lag = 40`
Residuals	0.6344104	0.8146683	0.09493672
Square Residuals	0.5569829	0.8413216	0.6454828

Table 2: `Box.test` for residuals and square residuals

3.4 Testing normality in residuals

We performed Normality tests in order to examine if the residuals come from a standard normal distribution. Firstly, we provide a histogram of standardized residuals (Figure 7).

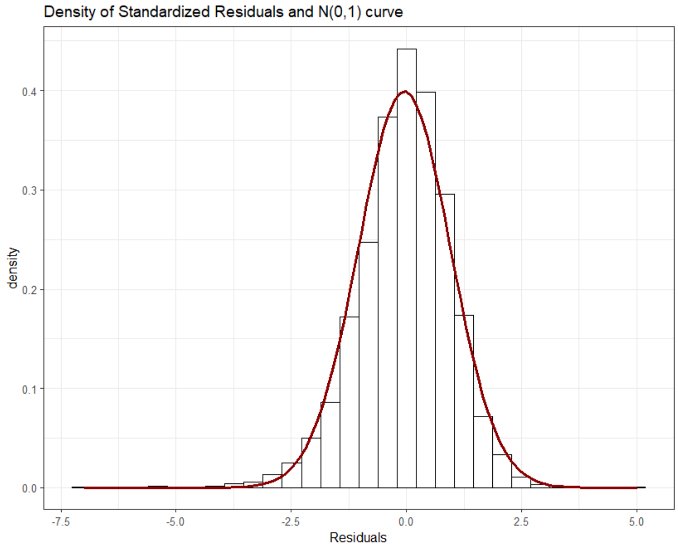

Figure 7: Histogram of Standardized Residuals

Next, we performed Kolmogorov-Smirnov and Jarque-Bera tests and found that the p.values were near zero (0). We reject the Null Hypothesis and there is strong evidence that the residuals do not come from Normal distribution ($N(0,1)$).

Ultimately, we drew a QQ plot as shown in Figure 8. It is clear that the residuals in tails show departures from the Normal line.

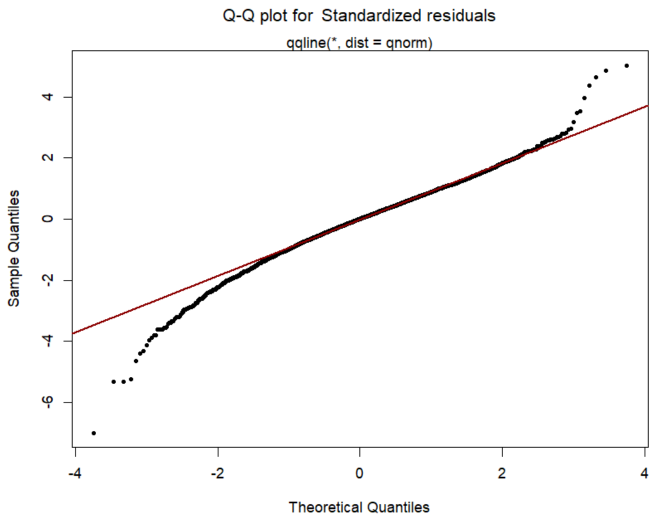

Figure 8: QQ-plot for standardized residuals

We conclude that the residuals do not follow a normal distribution. We will examine if t-distribution or skew-t distribution would be a better fit.

3.5 Fitting Student and Skew Student distribution for residuals

We checked the fitness of t and skew t distribution with the help of the QQ-plot. Figure 9 is shown that the skew-t distribution is a better fit than the t distribution.

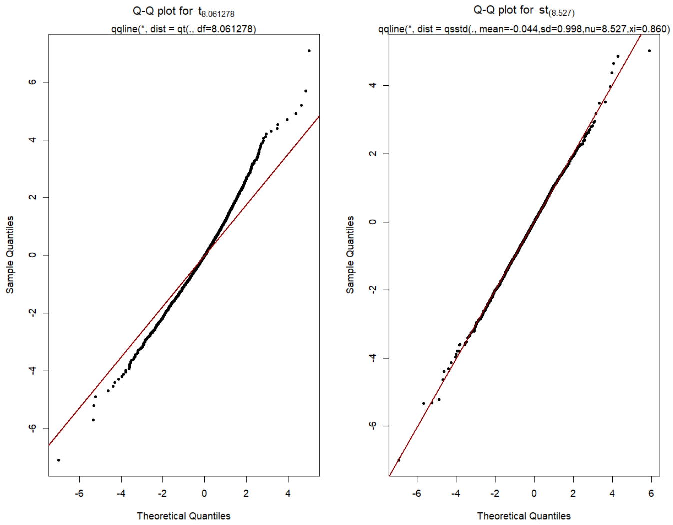

Figure 9: QQ-plot for residuals under t distribution (left) and skew-t distribution (right)

3.6 Results for the final ARMA-GARCH model

Given all the above, we decided to fit the data with ARMA(1,0)-GARCH(1,1) using skew-t distribution for residuals. Figure 10 confirms our claim. Also, the Bayesian Information Criterion (BIC) value of ARMA(1,0)-GARCH(1,1) has a lower value under skew-t distribution (-6.664157) than a normal distribution (-6.621627).

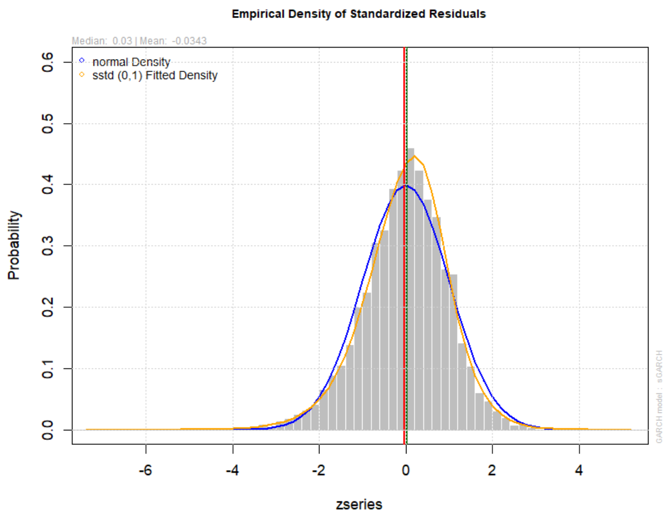

Figure 10: Empirical density of residuals with normal(blue) and skew-t curve(orange)

The new coefficients are shown in Table 3. According to `ugarchfit` function from `rugarch` package all the coefficients are significant except for ω. So $\omega = 0$:

μ	ϕ_1	ω	α_1	β_1	skew	shape
5.074629e-04	6.274257e-02	0	1.199763e-01	8.660391e-01	8.629333e-01	8.443010e+00

Table 3: Coefficinets of ARMA(1,0)-GARCH(1,1) under skew-t distribution

4 Extensions of Garch processes

Extensions of the Garch process will probably lead to a bit of improvement. We chose to compare our first model with exponential GARCH (EGARCH), integrated GARCH (IGARCH), and asymptotic power (APARCH). We use the Bayesian Information Criterion (BIC) criterion in order to examine which of them is better. The BIC values are shown in Table 4.

iGARCH	eGARCH	apARCH
-6.664479	-6.683906	-6.684926

Table 4: BIC values

The lowest Bayesian Information Criterion (BIC) value is -6.684926. So, we chose to compare ARMA(1,0)-apARCH(1,1) with the standard ARMA(1,0)-GARCH(1,1).
Comparing ARMA(1,0)-apARCH(1,1) and standard ARMA(1,0)-GARCH(1,1) we conducted that Bayesian Information Criterion (BIC) value for ARMA(1,0)-apARCH(1,1) is lower than ARMA(1,0)-GARCH(1,1) (Table 5).

sGARCH	apARCH
-6.664157	-6.684926

Table 5: BIC values

We concluded that the best model to fit the log return data of FTSE 250 stock is the ARMA(1,0)-APARCH(1,1).With the help of `ugarchfit` for `rugarch` package in R, the estimated coefficients of the model are significant except for ω. The output is shown below:

Listing 1: R output

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model    : apARCH(1,1)
Mean Model     : ARFIMA(1,0,0)
Distribution   : sstd

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000244    0.000109   2.2323 0.025597
ar1     0.083518    0.014092   5.9267 0.000000
omega   0.000072    0.000044   1.6208 0.105070
alpha1  0.100499    0.009269  10.8426 0.000000
beta1   0.891452    0.009316  95.6942 0.000000
gamma1  0.565933    0.067179   8.4243 0.000000
delta   1.246305    0.125127   9.9603 0.000000
skew    0.859221    0.017097  50.2553 0.000000
shape   9.609707    1.053636   9.1205 0.000000

LogLikelihood : 18719.85

Information Criteria
------------------------------------

Akaike       -6.6956
```

```
Bayes          -6.6849
Shibata        -6.6956
Hannan-Quinn   -6.6919

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                          statistic  p-value
Lag[1]                        3.423  0.06429
Lag[2*(p+q)+(p+q)-1][2]       3.479  0.01323
Lag[4*(p+q)+(p+q)-1][5]       4.155  0.20182
d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                          statistic  p-value
Lag[1]                       0.1765   0.6744
Lag[2*(p+q)+(p+q)-1][5]      0.4225   0.9690
Lag[4*(p+q)+(p+q)-1][9]      1.0129   0.9857
d.o.f=2
```

5 Leverage effect

Figure 11 shows a plot of the relationship between positive and absolute negative log returns and the Leverage effect plot(QQ plot between positive and absolute negative log returns).

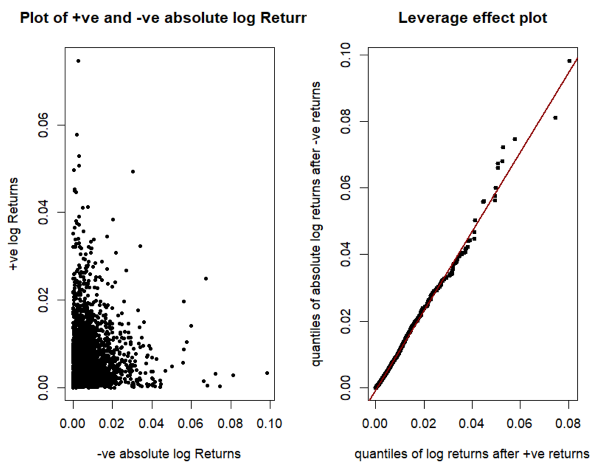

Figure 11: Left plot: plot of positive and negative log returns, Right plot: Leverage effect

The right plot shows that there are no departures from the diagonal, except in the tail.

Another check for leverage effect is the correlation between the square log return in t and the log return in t-1. We found that the correlation is -0.1017784, which is an indication of the leverage effect.

6 Conditional Volatility over time with Skew Student distributed noise

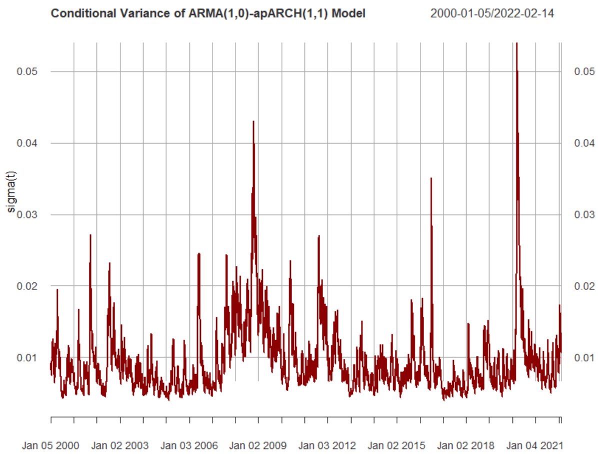

Figure 12: Conditional Volatility over time for log return data of FTSE 250 stock

Below, is a list of the first 20 sigma (conditional volatility) values.

Listing 2: 20 first values of conditional volatility under Skew Student distributed noise

```
> head(sigma(best.fit),20)
                  [,1]
2000-01-05 0.008144418
2000-01-06 0.009255127
2000-01-07 0.008827537
2000-01-10 0.008584171
2000-01-11 0.008244069
2000-01-12 0.007835234
2000-01-13 0.008557454
2000-01-14 0.008000384
2000-01-17 0.007581004
2000-01-18 0.007429376
2000-01-19 0.008483396
2000-01-20 0.009475865
2000-01-21 0.009451366
2000-01-24 0.011073242
2000-01-25 0.010353096
2000-01-26 0.012161972
2000-01-27 0.011455409
2000-01-28 0.010779698
2000-01-31 0.010823758
2000-02-01 0.012183252
```

7 Code in R

```
# Load the necessary packages:
library(tseries);library(dplyr);library(ggplot2);library(fGarch)
library(zoo);library(rugarch);library(forecast);library(Ecdat)
library(ggpubr); library(PerformanceAnalytics)
# Load the data
df <- get.hist.quote(instrument="^FTMC", start="2000-01-01",end="2022-02-15",
                     quote="Adj",retclass = 'zoo')
# Save the dates
Date <- index(df)
# Convert data into a data frame
if (is.data.frame(df) == FALSE) {
  df = as.data.frame(df)
}
# Create a data frame with Dates and Prices
df <- data.frame('Date'=Date,'Price'=df$Adjusted)
head(df,5)
# Omit the NA values from the data
df   <- df %>% filter(!is.na(Price) & !is.na(Date))
Date <- df %>% select(Date)
# Summary of the data:
df %>% select(Price) %>% summary()
#Examination of Stationarity
# Plot of the FTSE 250 stock
g1 <- ggplot(df,aes(x=Date,y=Price)) +
       geom_line(color='darkred') +
        labs(x='Time',y='FTSE 250',
          title='Plot of FTSE 250 stock during T = 01-01-2000 - 15-02-2022')+
            theme_bw();g1 # This is like a trend.
# Convert to stationarity data
adf.test(df$Price,alternative = 'stationary',k=10)$p.val
#Transformations for Stationarity
# Square root
sq <- apply(df %>% select(Price),2,sqrt) #sqroo
sq <- diff(sq); adf.test(sq,k=15)$p.val
g2 <- ggplot(data.frame('Date'=Date[-1,],'Price'=sq),aes(x=Date,y=Price)) +
    geom_line(color='darkred') +
    labs(x='Time',y='Square root differences',
    title='Plot of square root differences of prices during T')+
    theme_bw() # This is like a trend.

# Cube root
cb <- apply(df %>% select(Price),2,function(x){x^(1/3)})
cb <- diff(cb); adf.test(cb,k=15)$p.val
g3 <- ggplot(data.frame('Date'=Date[-1,],'Price'=cb),aes(x=Date,y=Price)) +
```

```r
    geom_line(color='darkred') +
    labs(x='Time',y='Cube root differences',
    title='Plot of cube root differences of prices during T')+
    theme_bw() # This is like a trend.
ggarrange(g2,g3,nrow=2,ncol=1)

# Log transformation
rt <- rep(NA,dim(df)[1]-1)
for(i in 1:(dim(df)[1]-1)){
  rt[i] <- log(df[i+1,2]) - log(df[i,2])
}
rt <- data.frame('Return'=rt, 'Square.return' = (rt)^2)
rownames(rt) <- Date[-1,];summary(rt %>% select(1)) # Summary of log-returns
# Plot of the log returns, squared returns and absolute returns of FTSE 250:
g4 <-  ggplot(data.frame('Date' = Date[-1,],'Return'=rt$Return),
      aes(x=Date,y=Return)) +
      geom_line(color='darkred') +
      labs(x='Time',y='log-returns FTSE 250',
      title='Plot of log-returns during T')+
      theme_bw()
g5 <-  ggplot(data.frame('Date' = Date[-1,],'Square.return'=rt$Square.return),
      aes(x=Date,y=Square.return)) +
      geom_line(color='darkred') +
      labs(x='Time',y='squared log-returns FTSE 250',
      title='Plot of squared log returns during T')+
      theme_bw()
g6 <-  ggplot(data.frame('Date' = Date[-1,],'abs.return'=abs(rt$Return)),
      aes(x=Date,y=abs.return)) +
      geom_line(color='darkred') +
      labs(x='Time',y='absolute log-returns FTSE 250',
      title='Plot of absolute log-returns during T')+
      theme_bw()
ggarrange(g4,g5,g6,nrow = 3,ncol = 1) # Combination of ggplots
# Histogram of the Log-returns with the N(0,1) density curve:
g7 <- rt %>%
      ggplot() +
      geom_histogram(aes(x=Return,y=..density..),color='black',fill='white') +
      stat_function(fun=dnorm, args = list(mean = mean(rt$Return),
      sd = sd(rt$Return)), size=1,col='darkred')+ labs(
      x='Log-returns of FTSE 250',
      title='Histogram of log-returns during T')+
      theme_bw();g7
#Fitting ARMA-GARCH MODEL
# log returns,square log returns, and absolute log returns:
par (mfrow=c(2,2))
acf (rt  %>% select(1),main='ACF of FTSE 250 stock log-returns')
```

```
pacf(rt %>% select(1),main='PACF of FTSE 250 stock log-returns')
acf (rt  %>% select(2),main='ACF of FTSE 250 stock squared log-returns')
acf (abs(rt %>% select(1)),main='ACF of FTSE 250 stock absolute log-returns')
par (mfrow=c(1,1))
# MODEL SELECTION: (We suppose 12*12 models)
garch_order <- matrix(NA,12,2)
for(i in 1:3){
  garch_order[(4*i-3):(4*i),1] <- rep(i,4)
  garch_order[(4*i-3):(4*i),2] <- seq(0,3,by=1)
             }
garch_order  ;arma_order <- garch_order

bayes_crit <- matrix(NA,12,12)
for(j in 1:dim(garch_order)[1]) {
  for(i in 1:dim(arma_order)[1])  {
    fit.model <- ugarchfit(spec = ugarchspec(variance.model=list(
                 model="sGARCH", garchOrder=garch_order[j,]),
                 mean.model=list(armaOrder=arma_order[i,]),
                 distribution.model="norm"),data = rt$Return,solver = 'hybrid')
     bayes_crit[i,j] <- infocriteria(fit.model)[2] # With the BIC criterion
                                  }
                                }
index.garch_order <-
  which(apply(bayes_crit,2,min)==min(apply(bayes_crit,2,min)))
index.arma_order  <-
  which(bayes_crit[,index.garch_order]==min(apply(bayes_crit,2,min)))
# Best model:
garch.fit <- garch_order[index.garch_order,]
arma.fit  <- arma_order[index.arma_order,]
# We conclude that according to BIC criterion the model is:
# GARCH(1,1)-ARMA(1,0)
ftse.model <- ugarchfit(spec = ugarchspec(variance.model=list(
                 model="sGARCH",garchOrder=garch.fit),
                 mean.model=list(armaOrder=arma.fit),distribution.model="norm"),
                 data = rt$Return,solver = 'hybrid')
ftse.model;coef(ftse.model) # Coefficients of the ARMA(1,0)-GARCH(1,1)

#Testing for auto correlations of the residuals and square residuals
### Residuals
# Box.test(*,type='Ljung',fitdf=1+0):
res <- ftse.model@fit$z
Box.test(res,type = 'Ljung',lag = 10,fitdf = 1)$p.val
Box.test(res,type = 'Ljung',lag = 20,fitdf = 1)$p.val
Box.test(res,type = 'Ljung',lag = 40,fitdf = 1)$p.val
### Square Residuals:
# Box.test(*,type='Ljung',fitdf=1+1):
```

```r
Box.test(res^2,type = 'Ljung',lag = 10,fitdf = 2)$p.val
Box.test(res^2,type = 'Ljung',lag = 20,fitdf = 2)$p.val
Box.test(res^2,type = 'Ljung',lag = 40,fitdf = 2)$p.val

#Testing the Normality in residuals
### Histogram of standardized residuals
g8 <- ggplot(data.frame('Residuals' = res)) +
    geom_histogram(aes(x=Residuals,y=..density..),
    color='black',fill='white') +
    stat_function(fun=dnorm, args = list(mean = mean(res),sd = sd(res)),
    size=1,col='darkred') + labs(x='Residuals')+
    ggtitle('Density of Standardized Residuals and N(0,1) curve') +
    theme_bw();g8
### QQ-plot of standardized residuals:
qqnorm(res,pch=20,
main = expression("Q-Q plot for" ~~ {'Standardized residuals'}))
qqline(res,col='darkred',lwd=2);mtext("qqline(*, dist = qnorm)")

### Tests for normality:
# kolmogorov-smirnov, Jarque-bera test
fun <- function(x){
  c(round(ks.test(x,'pnorm',mean=mean(res),sd = sd(res))$p.val,4),# ks.test
    round(jarque.bera.test(x)$p.val,4))# jarque-bera test
};res %>% fun

#Fitting t-distribution for residuals
par(mfrow=c(1,2)) # We will use the function stdFit:
res.t.fit <- stdFit(res)
dof <- as.numeric(res.t.fit$par[3])
qqplot(res,qt(ppoints(10000),df=dof), ylab = 'Sample Quantiles',
  xlab = 'Theoretical Quantiles',
  main = expression("Q-Q plot for" ~~ {t}[nu = 8.061278 ]),pch=20)
qqline(res,distribution = function(y,df=dof) qt(y,df=dof),
  col = 'darkred',lwd=2);mtext("qqline(*, dist = qt(., df=8.061278)")

#Fitting skew t-distribution for residuals
res.st.fit <- sstdFit(res)
param <- round(res.st.fit$estimate,3)
set.seed(1235)
qqplot(x=rsstd(10000,mean = param[1],
  sd = param[2],nu = param[3],xi = param[4]),
  y = res, main = expression("Q-Q plot for" ~~ {st}[nu = (8.527)]),pch=20,
  ylab = 'Sample Quantiles', xlab = 'Theoretical Quantiles')
qqline(res,
 distribution = function(p) qsstd(p,mean=param[1],sd=param[2],nu=param[3],
                                  xi=param[4]),lwd=2,col='darkred')
```

```
mtext("qqline(*, dist = qsstd(., mean=-0.044,sd=0.998,nu=8.527,xi=0.860)")
par(mfrow=c(1,1))

# Skew t distribution is a better fit than normal and t distribution.
ftse.model <- ugarchfit(spec = ugarchspec(variance.model=list(
              model="sGARCH", garchOrder=garch.fit),
              mean.model=list(armaOrder=arma.fit),distribution.model="sstd"),
              data = rt %>% select(1),solver = 'hybrid')
ftse.model
BIC.sgarch <- infocriteria(ftse.model)[2]
# Normal Vs Skew-Student for Residuals.
plot(ftse.model,which=8)
coef(ftse.model)
# Extensions of Garch processes
BIC.egarch <- infocriteria(ugarchfit(spec = ugarchspec(variance.model=list(
              model="eGARCH",garchOrder=garch.fit),
              mean.model=list(armaOrder=arma.fit),distribution.model="sstd"),
              data = rt %>% select(1),solver = 'hybrid'))[2]
BIC.aparch <- infocriteria(ugarchfit(spec = ugarchspec(variance.model=list(
              model="apARCH",garchOrder=garch.fit),
              mean.model=list(armaOrder=arma.fit),distribution.model="sstd"),
              data = rt %>% select(1),solver = 'hybrid'))[2]
BIC.igarch <- infocriteria(ugarchfit(spec = ugarchspec(variance.model=list(
              model="iGARCH",garchOrder=garch.fit),
              mean.model=list(armaOrder=arma.fit),distribution.model="sstd"),
              data = rt %>% select(1),solver = 'hybrid'))[2]
name_model <- c('sGARCH','iGARCH','eGARCH','apARCH')
method     <- c(BIC.sgarch,BIC.igarch,BIC.egarch,BIC.aparch)
# Create a matrix with the BIC values according to different Garch processes:
MODEL_SELECTION <- matrix(method,nrow=1,ncol=4)
colnames(MODEL_SELECTION)  <- name_model
row.names(MODEL_SELECTION) <- 'BIC'
MODEL_SELECTION
# Find the model with the lowest BIC value:
min(MODEL_SELECTION)
# So, we have a better fit for the problem which is the apARCH(1,1):
name_model[which(MODEL_SELECTION == min(MODEL_SELECTION))]
# This suggests that the AR(1)-APARCH(1,1) model is the one that
# fits the FTSE 250 time series best.
best.fit <- ugarchfit(spec = ugarchspec(variance.model=list(
            model="apARCH",garchOrder=garch.fit),
            mean.model=list(armaOrder=arma.fit),distribution.model="sstd"),
            data = rt %>% select(1),solver = 'hybrid');best.fit
# Leverage effect
# Method:
col_1 <- rep(NA,dim(rt)[1]); col_2 <- rep(NA,dim(rt)[1])
```

```r
for(i in 1:dim(rt)[1]) {
  col_1[i] <- rt$Return[i]
  col_2[i] <- rt$Square.return[i+1] }
leverage <- data.frame('ret'=col_1[-dim(rt)[1]],'s.ret'=col_2[-dim(rt)[1]])
cor(leverage$ret,leverage$s.ret)

# Another method:
r.neg <- abs(rt$Return[which(rt$Return < 0)]) # Absolute negative log-returns
r.pos <-     rt$Return[which(rt$Return > 0)]  # Positive log-returns
# Plot of the relationship between +ve and absolute -ve log-returns and
# Leverage effect plot(QQ-plots between +ve and absolute -ve log-returns).
set.seed(565)
par(mfrow=c(1,2))
plot(x = r.neg, y = sample(r.pos,length(r.neg)), pch = 20,
    xlab = '-ve absolute log Returns',ylab='+ve log Returns',
    main='Plot of +ve and -ve absolute log Returns')
qqplot(x = sample(r.pos,length(r.neg)), y = r.neg,
       xlab = 'quantiles of log returns after +ve returns',
       ylab = 'quantiles of absolute log returns after -ve returns',
      pch = 20, lwd = 2, col = 'black', main = 'Leverage effect plot')
abline(lm(sort(r.neg) ~ sort(sample(r.pos,length(r.neg)))),
       col = 'darkred', lwd = 2) ;par(mfrow=c(1,1))
# Conditional Volatility plot
sigma(best.fit) # Sigma values
head(sigma(best.fit),20)
# Plot of the Volatility
plot(sigma(best.fit), ylab="sigma(t)", col="darkred",
        main = "Conditional Variance of ARMA(1,0)-apARCH(1,1) Model")
# -------------------------------------------
# END: Vasileios Diplas
# -------------------------------------------
```