ECON7350: Applied Econometrics
for Macroeconomics and Finance

Tutorial 8: Modelling Volatility - I

At the end of this tutorial you should be able to:

- use R to infer the presence of heteroscedasticity using the Breusch-Pagan test;

- construct an adequate set of models with possible GARCH errors;

- use R to estimate volatilities based on models with GARCH errors.

Problems with Solutions

Consider the daily share prices of Commonwealth Bank (CWB) for the period 5
September 1996—30 August 2006 ($T = 2605$) in the data file `cwb.csv`. Let $\{y_t\}$
denote the process of share prices.

1. Plot the *share prices* (y_t) and comment on the possible features of the DGP.

**Solution**   For this tutorial, we load the following useful packages.

```
library(forecast)
library(dplyr)
library(rugarch)
```

The package `rugarch` provides functions for working with conditional heteroscedas-
ticity specifications. In this tutorial we will use the functions `ugarchspec` and
`ugarchfit` to estimate volatilities based on standard GARCH models.

Next, load the data and and extract the variables.

```
mydata <- read.delim("cwb.csv", header = TRUE,  sep = ",")


date <- as.Date(mydata$date, format = "%m/%d/%Y")
y <- mydata$y
r <- diff(log(y))
plot(date, y, type = "l", xlab = "", ylab = "share prices")
```
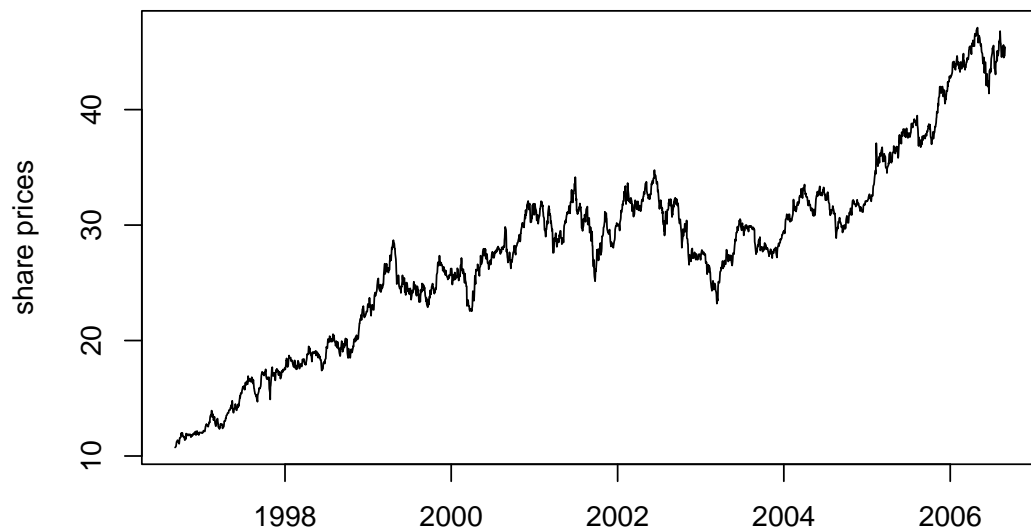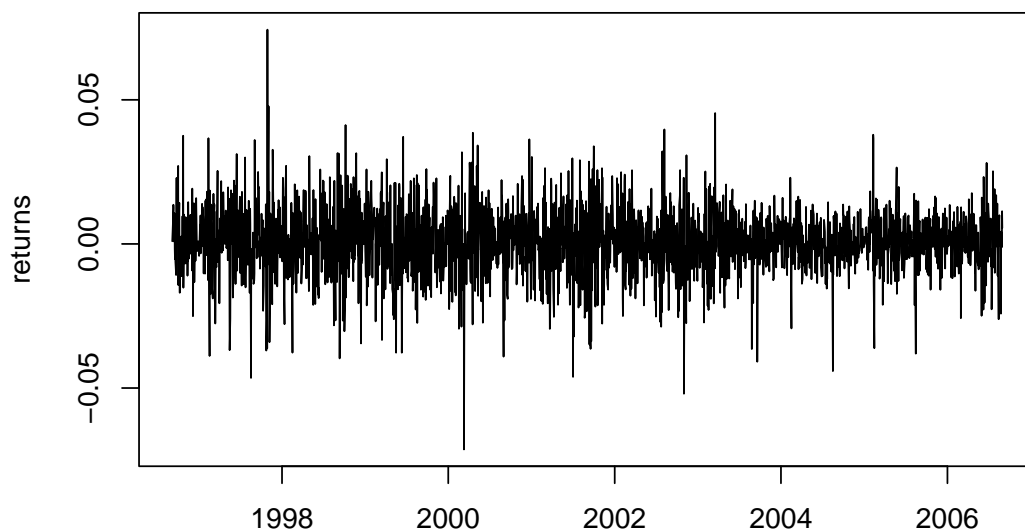
Share prices are clearly trending upwards, being around $10 in the beginning of the sample and reaching about $45 towards the end.

---

2. Plot the *returns* ($r_t = \ln y_t - \ln y_{t-1}$) and comment on the possible features of the DGP.

```
plot(date[-1], r, type = "l", xlab = "", ylab = "returns")
```

Solution

Returns look stable with possible heteroscedasticity. Observe that if returns are heteroscedastic then so must be the share prices. However, evidence of heteroscedasticity may be less obvious in plots of share prices than in plots of returns.

---

3. Assuming homoscedasticity, identify an adequate set of ARMA models for r_t.

Solution   We use the familiar approach to building an adequate set of homoscedastic ARMA models.

```
ARMA_est <- list()
ic_arma <- matrix( nrow = 4 * 4, ncol = 4 )
colnames(ic_arma) <- c("p", "q", "aic", "bic")
for (p in 0:3)
{
  for (q in 0:3)
  {
    i <- p * 4 + q + 1
    ARMA_est[[i]] <- Arima(r, order = c(p, 0, q))
    ic_arma[i,] <- c(p, q, ARMA_est[[i]]$aic, ARMA_est[[i]]$bic)
  }
}
```

```
ic_aic_arma <- ic_arma[order(ic_arma[,3]),][1:10,]
ic_bic_arma <- ic_arma[order(ic_arma[,4]),][1:10,]

ic_int_arma <- intersect(as.data.frame(ic_aic_arma),
                         as.data.frame(ic_bic_arma))

adq_set_arma <- as.matrix(arrange(as.data.frame(
                                  rbind(ic_int_arma[c(1:3, 6),],
                                   ic_bic_arma[2,])), p, q))
adq_idx_arma <- match(data.frame(t(adq_set_arma[, 1:2])),
                      data.frame(t(ic_arma[, 1:2])))

nmods <- length(adq_idx_arma)
for (i in 1:nmods)
{
  checkresiduals(ARMA_est[[adq_idx_arma[i]]])
}
```


Residuals from ARIMA(0,0,1) with non-zero mean
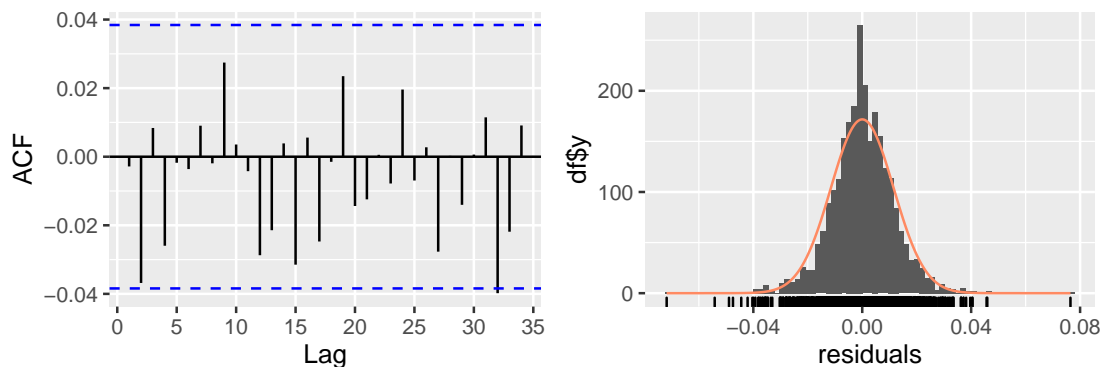
```
##
##   Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with non-zero mean
## Q* = 7.7683, df = 8, p-value = 0.4564
##
## Model df: 2.   Total lags used: 10
```
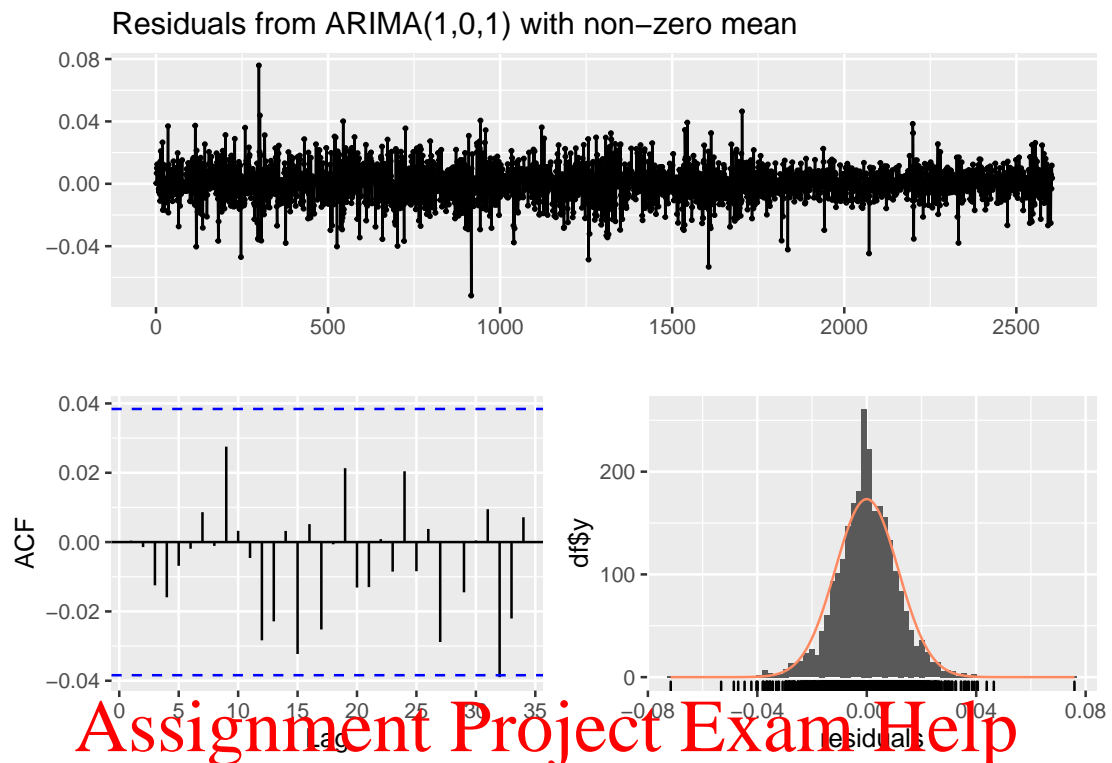
Residuals from ARIMA(0,0,2) with non−zero mean
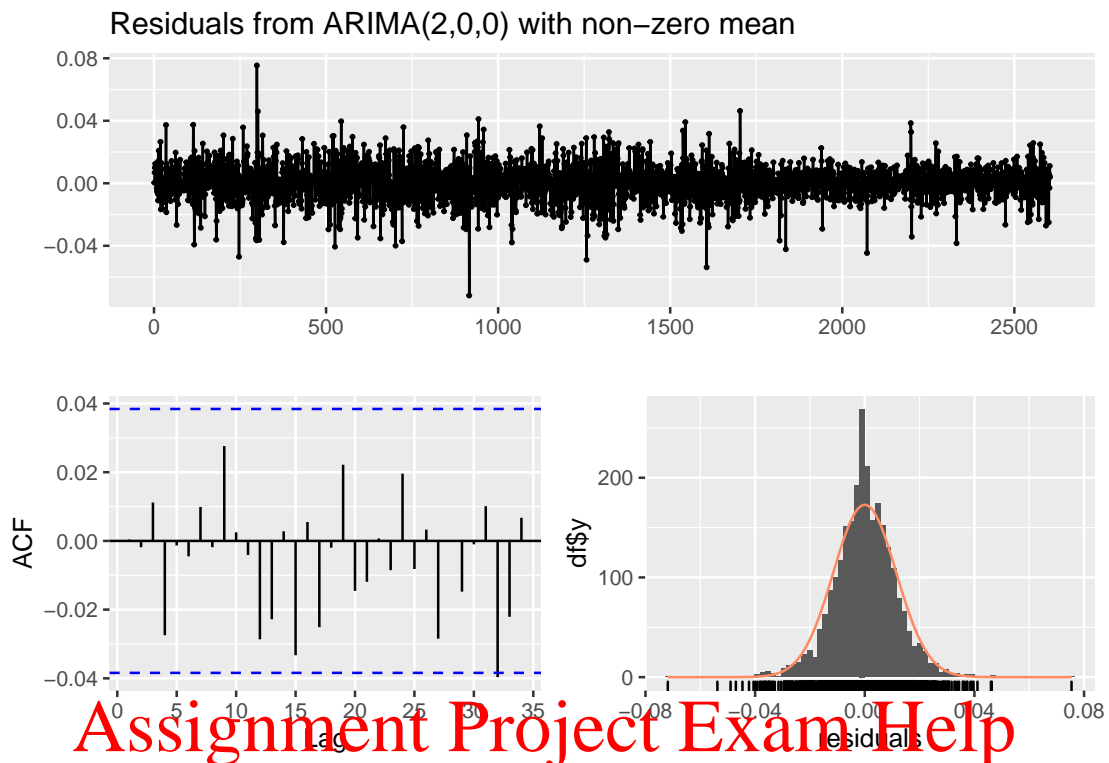
```
## 
##   Ljung-Box test
## 
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 4.2044, df = 7, p-value = 0.756
## 
## Model df: 3.   Total lags used: 10
```

Residuals from ARIMA(1,0,0) with non-zero mean
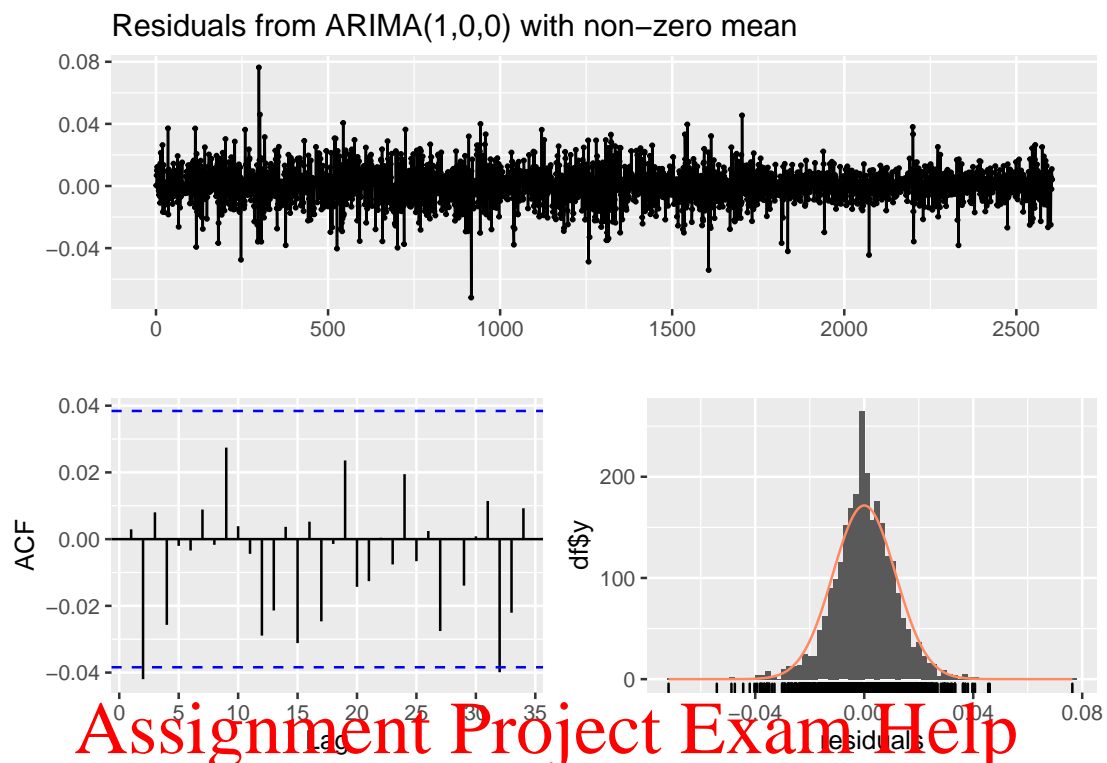
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 8.7674, df = 8, p-value = 0.3623
##
## Model df: 2.   Total lags used: 10
```

Residuals from ARIMA(1,0,1) with non−zero mean

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

```
## 
##  Ljung-Box test
## 
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 3.4143, df = 7, p-value = 0.8442
## 
## Model df: 3.   Total lags used: 10
```

Residuals from ARIMA(2,0,0) with non-zero mean

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with non-zero mean
## Q* = 4.632, df = 7, p-value = 0.7048
##
## Model df: 3.   Total lags used: 10
```

The adequate set consists of five models: MA(1), MA(2), AR(1), AR(2) and ARMA(1, 1).

---

4. Generate estimated squared residuals for the set of models chosen in Question 3. Plot the squared residuals along with the sample ACFs. Interpret your findings.

Solution For each model in the adequate set, extract the estimated residuals using the `resid` function. Then, use `plot` and `acf` to obtain the desired graphs.

```
e2_arma <- list()
for (i in 1:nmods)
{
  e2_arma[[i]] <- resid(ARMA_est[[adq_idx_arma[i]]]) ^ 2
```

```r
title_p_q <- paste("ARMA(",
                    as.character(adq_set_arma[i, 1]), ", ",
                    as.character(adq_set_arma[i, 2]), ")",
                    sep = "")
plot(date[-1], e2_arma[[i]], type = "l",
                xlab = "", ylab = "squared resid",
                main = paste("Plot: ", title_p_q))

acf(e2_arma[[i]], xlab = "", ylab = "",
                main = paste("SACF: ", title_p_q))
}
```
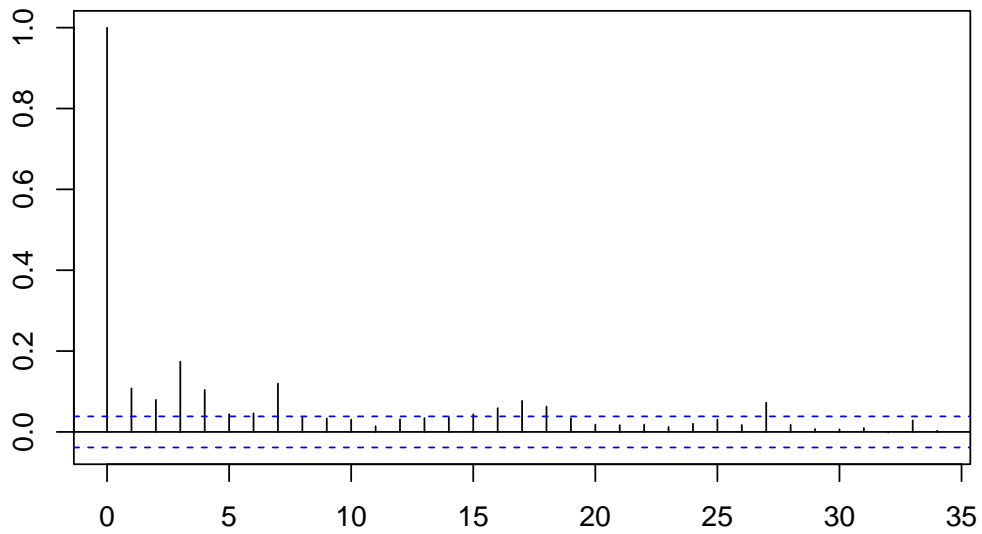
Plot: ARMA(0, 1)

SACF: ARMA(0, 1)

Plot: ARMA(0, 2)

SACF: ARMA(0, 2)

Plot: ARMA(1, 0)

SACF: ARMA(1, 0)
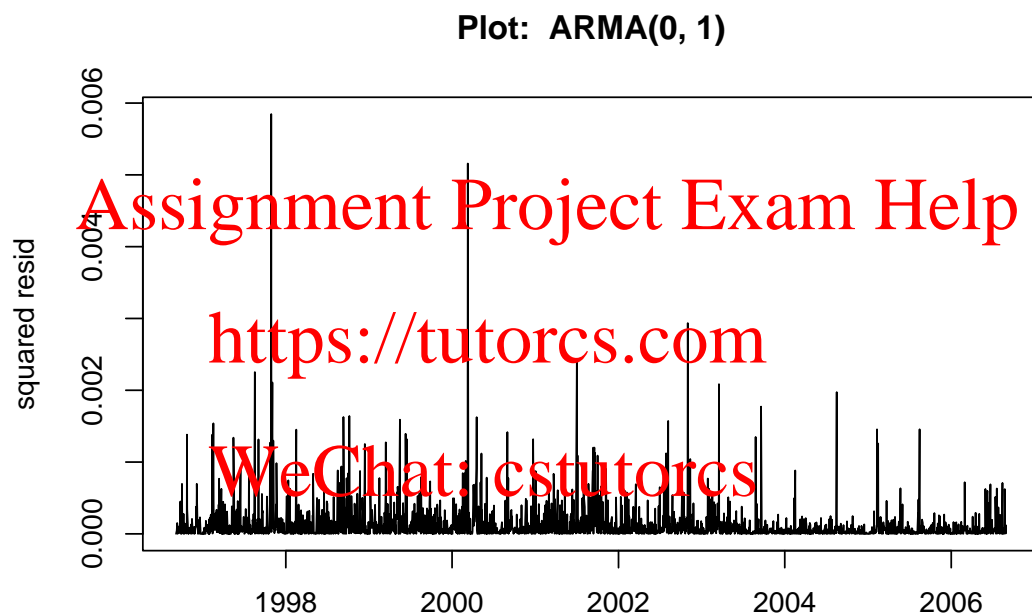
Plot: ARMA(1, 1)
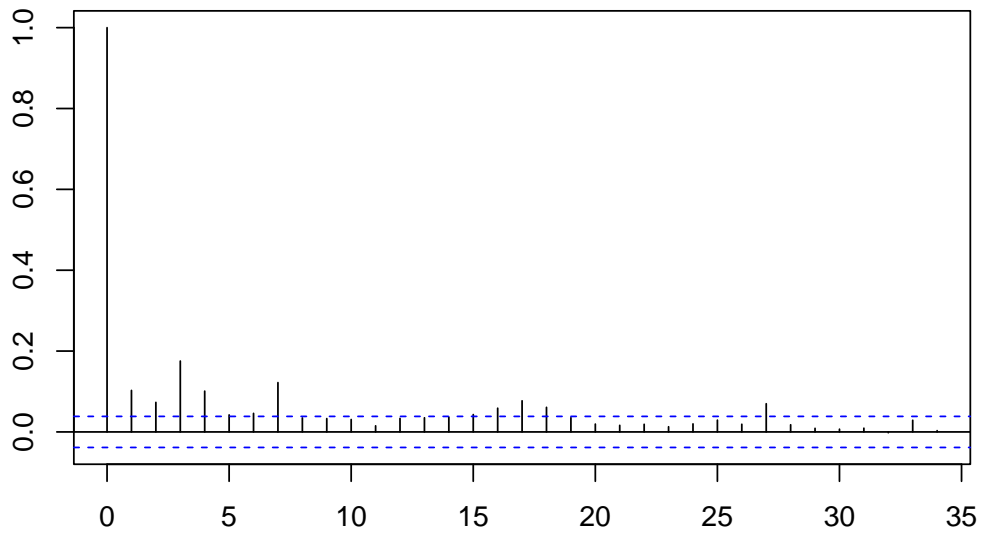
SACF: ARMA(1, 1)

Assignment Project Exam Help

Plot: ARMA(2, 0)

https://tutorcs.com
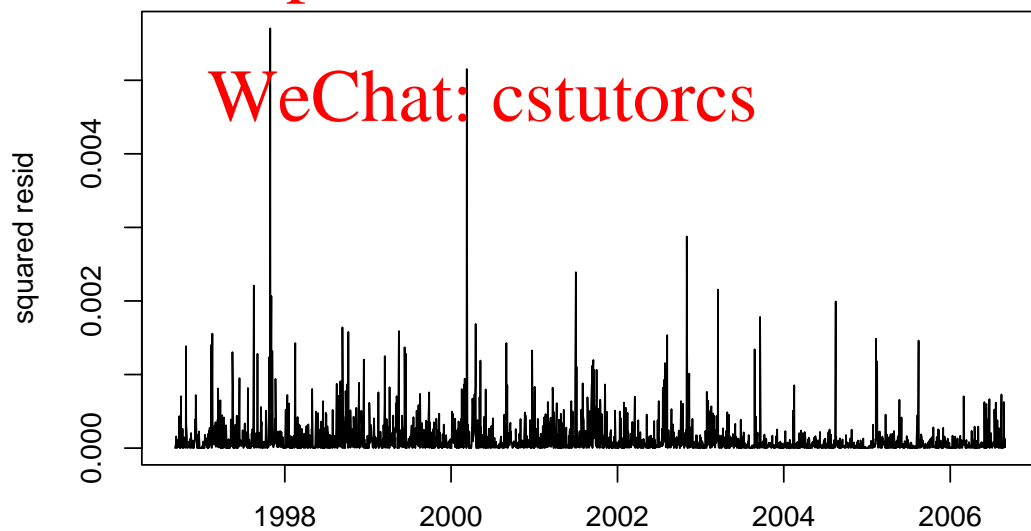
WeChat: cstutorcs

SACF: ARMA(2, 0)
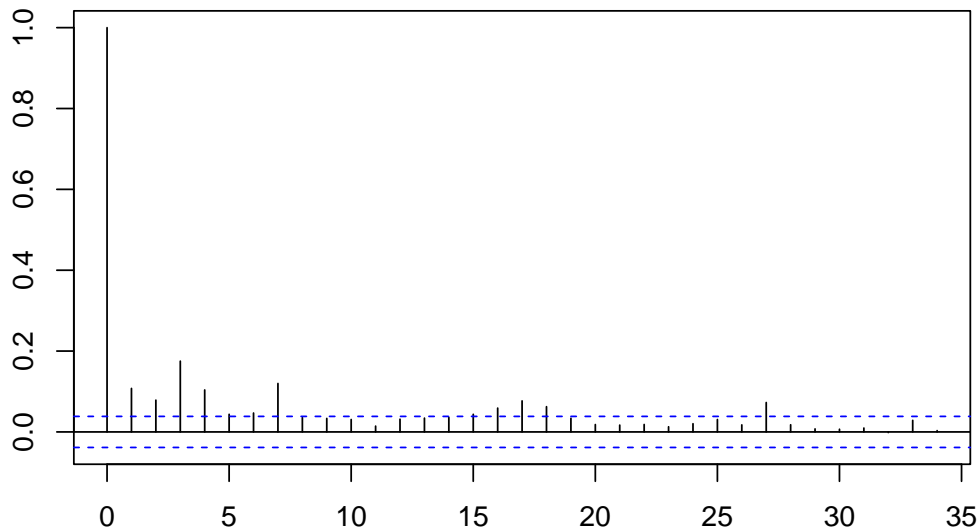
Squared residuals appear to be strongly autocorrelated even though autocorrelation in the residual levels appears to be quite low. We interpret this as evidence of conditional heteroscedasticity possibly being present in the DGP.

---

5. Test if the errors in your set of models contain ARCH or GARCH effects.

Solution   We use the Breush-Pagan test, which is a type of Lagrange-Multiplier (LM) test. The test is implemented for every model in the adequate set constructed in Question 3. It also requires a specification of j lags in the regression of squared residuals on lags. We will try lags ranging from 1 to 10. If there are discrepancies, we will need to delve deeper and construct an adequate set of B-P regressions for the test using the same approach we have learned based on information criteria and residuals analysis.

Note that the residual in the B-P regression involving squared residuals from the ARMA specification is assumed to be mean-independent. This is necessary to justify the χ^2 distribution of the LM test statistic. You should notice the connection to the corresponding role of mean-independence in the ADF test.

Also, the B-P regression is in fact an AR(j). However, because we need the R^2 from the B-P regression, it is easier to use the `lm` function in R rather than `Arima`.

```
bptest <- matrix(nrow = 10 * nmods, ncol = 5)
colnames(bptest) <- c("p", "q", "j", "LM-stat", "p-value")
for (i in 1:nmods)
{
  e2_i <- as.vector(e2_arma[[i]])
  f <- formula(e2_i ~ 1)
  for (j in 1:10)
  {
    # lag lengths in the auto-regression of squared residuals
    k <- 10 * (i - 1) + j
    f <- update.formula(f, paste("~ . + lag(e2_i, n = ", j, ")"));
    bp_reg_j <- lm(f)
    LM_j <- length(e2_i) * summary(bp_reg_j)$r.squared
    p_val_j <- 1 - pchisq(LM_j, df = j)
    bptest[k,] <- c(adq_set_arma[i, 1:2], j, LM_j, p_val_j)
  }
}
```

In specifications, the null hypothesis of homostedastic errors is rejected in favor of ARCH/GARCH errors at very low significance levels. This suggests we should consider expanding our adequate set of models to include conditional heteroscedasticity.

---

6. Expand the adequate set of models to specifications with heteroscedasticity. To this end, only consider conditional variances modelled with ARCH/GARCH residuals. Hint: Use the `rugarch` package with `ugarchspec` and `ugarchfit` functions.

Solution   Based on the adquate set constructed with homodcedastic ARMAs, we begin by considering ARMA variants with $p_m = 0, \cdots, 2$ and $q_m = 0, ..., 2$ in the conditional mean along with GARCH lags $p_h = 0, \cdots, 2$ and $q_h = 0, ..., 2$ in the conditional variance.

There is a total of $3^4 = 81$ models altogether—it will take a few minutes to estimate them all! Also, some specifications will be so bad that numerical optimisation will fail, so it is helpful to use the `try` function to avoid interruptions.

```
ARMA_GARCH_est <- list()
ic_arma_garch <- matrix( nrow = 3 ^ 4, ncol = 6 )
colnames(ic_arma_garch) <- c("pm", "qm", "ph", "qh", "aic", "bic")
i <- 0
for (pm in 0:2)
{
```

```r
  for (qm in 0:2)
  {
    for (ph in 0:2)
    {
      for (qh in 0:2)
      {
        i <- i + 1
        ic_arma_garch[i, 1:4] <- c(pm, qm, ph, qh)

        if (ph == 0 && qh == 0)
        {
          # for models with constant variance, the ugarchspec and
          # ugarchfit functions do not work well; instead, the
          # documentation advises to use arfimaspec and arfimafit
          ARMA_GARCH_mod <- arfimaspec(
            mean.model = list(armaOrder = c(pm, qm)))

          ARMA_GARCH_est[[i]] <- arfimafit(ARMA_GARCH_mod, r)
          ic_arma_garch[i,5:6] <- infocriteria(
                            ARMA_GARCH_est[[i]])[1:2]
        }
        else
        {
          try(silent = T, expr =
          {
            ARMA_GARCH_mod <- ugarchspec(
                mean.model = list(armaOrder = c(pm, qm)),
                variance.model = list(garchOrder = c(ph, qh)))

            ARMA_GARCH_est[[i]] <- ugarchfit(ARMA_GARCH_mod, r,
                                      solver = 'hybrid')

            ic_arma_garch[i,5:6] <- infocriteria(
                            ARMA_GARCH_est[[i]])[1:2]
          })
        }
      }
    }
  }
}

ic_aic_arma_garch <- ic_arma_garch[
                order(ic_arma_garch[,5]),][1:40,]
ic_bic_arma_garch <- ic_arma_garch[
```

```
                            order(ic_arma_garch[,6]),][1:40,]

ic_int_arma_garch <- intersect(as.data.frame(ic_aic_arma_garch),
                                as.data.frame(ic_bic_arma_garch))
```

There are *a lot* of models that have comparable AIC and BIC measures! We select
the first 36 in the intersecting set.

```
adq_set_arma_garch <- as.matrix(arrange(as.data.frame(
                        ic_int_arma_garch[1:36,]), pm, qm, ph, qh))
adq_idx_arma_garch <- match(
                        data.frame(t(adq_set_arma_garch[, 1:4])),
                        data.frame(t(ic_arma_garch[, 1:4])))
```

Next, we check the residuals for specifications in the adequate set. Note that the
standard Ljung-Box test is not well-defined for heteroscedastic models because the
derivation of the sampling distribution assumes homoscedasticity. We will therefore
avoid relying on the hypothesis test, but still examine the SACF for obvious for
evidence of substantial autocorrelation.

```
nmods <- length(adq_idx_arma_garch)
sacf_garch <- matrix(nrow = nmods, ncol = 14)
colnames(sacf_garch) <- c("pm", "qm", "ph", "qh", 1:10)
for (i in 1:nmods)
{
  sacf_garch[i,1:4] <- adq_set_arma_garch[i,1:4]
  sacf_garch[i,5:14] <-
                    acf(ARMA_GARCH_est[[adq_idx_arma_garch[i]]]@fit$z,
                lag = 10, plot = F)$acf[2:11]
}
```

Residual autocorrelations appear to be relatively small for all models in the adquate
set.

---

7. Estimate each model in the set identified in Question 6 and plot the estimated
   volatilities ($\hat{h}_t$). Interpret the results.

Solution   We have already estimated all the models in the adequate set, so the
only thing left to do is plot the estimated volatilities. Unfortunately, the `rugarch`
package *does not* provide standard errors or confidence intervals for estimated
volatilities. We would need to write a customised function to compute these (left
as practice...or for an advanced econometrics course)!

17

```
for (i in 1:nmods)
{
  title_p_q <- paste("ARMA(",
                      as.character(adq_set_arma_garch[i, 1]), ", ",
                      as.character(adq_set_arma_garch[i, 2]),
                      ")-GARCH(",
                      as.character(adq_set_arma_garch[i, 3]), ", ",
                      as.character(adq_set_arma_garch[i, 4]), ")",
                      sep = "")
  plot(date[-1], ARMA_GARCH_est[[adq_idx_arma_garch[i]]]@fit$var,
       type = "l", xlab = "", ylab = "volatilities",
       main = title_p_q)
}
```

ARMA(0, 0)–GARCH(1, 1)
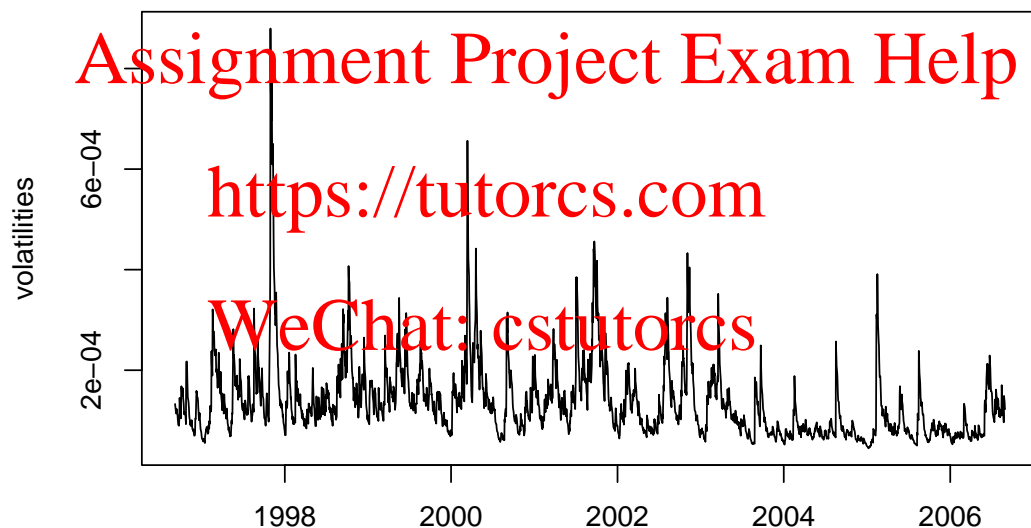
ARMA(0, 0)−GARCH(1, 2)
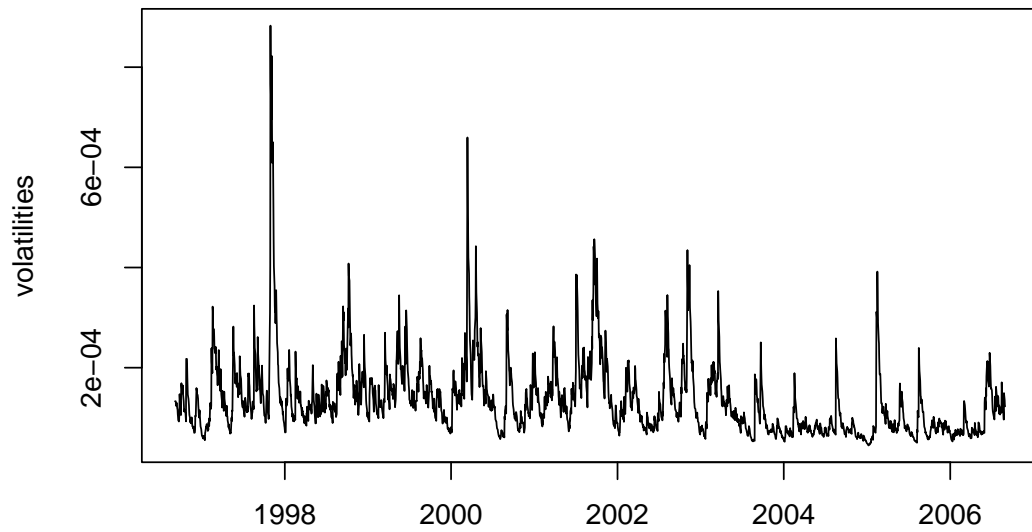
Assignment Project Exam Help

ARMA(0, 0)−GARCH(2, 1)

https://tutorcs.com
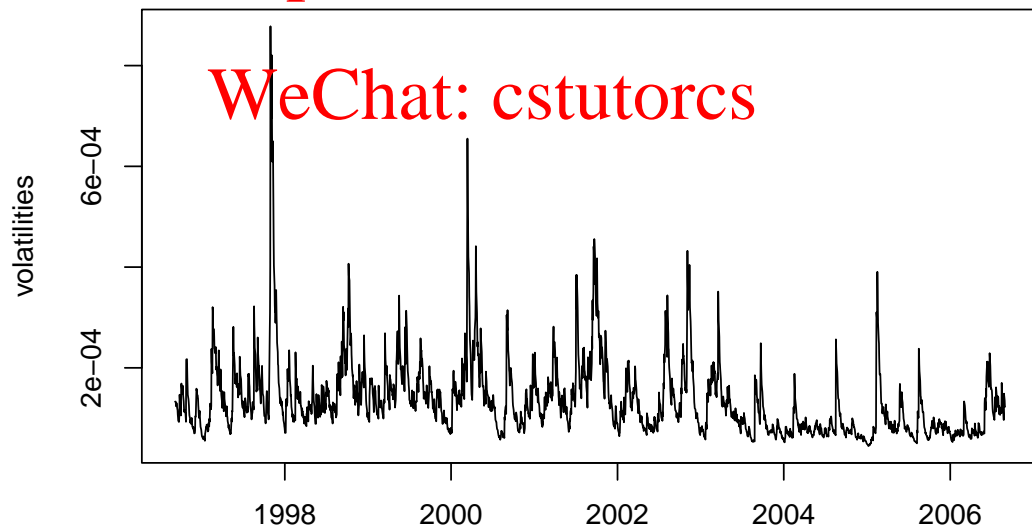
WeChat: cstutorcs

ARMA(0, 0)–GARCH(2, 2)
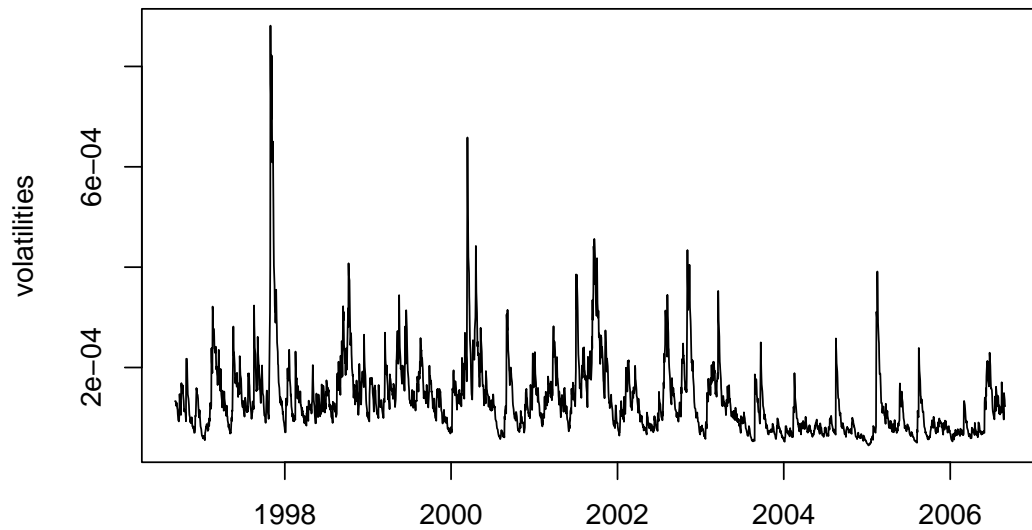
Assignment Project Exam Help

ARMA(0, 1)–GARCH(1, 1)

https://tutorcs.com
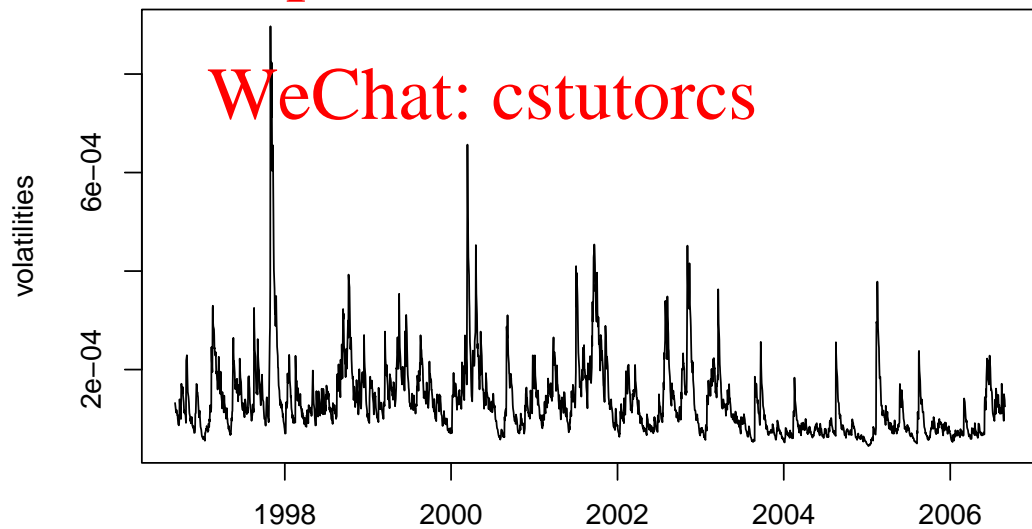
WeChat: cstutorcs

ARMA(0, 1)−GARCH(1, 2)

ARMA(0, 1)−GARCH(2, 1)

ARMA(0, 1)−GARCH(2, 2)
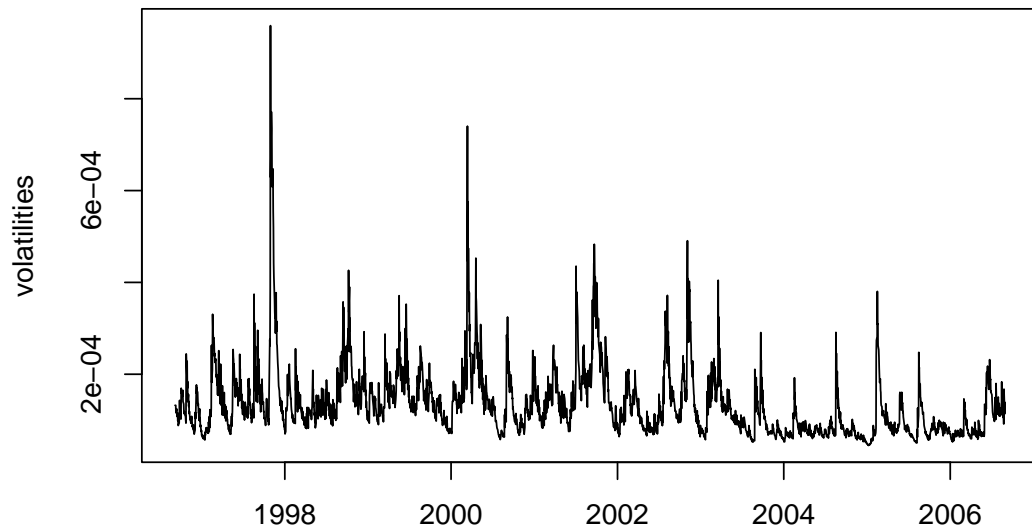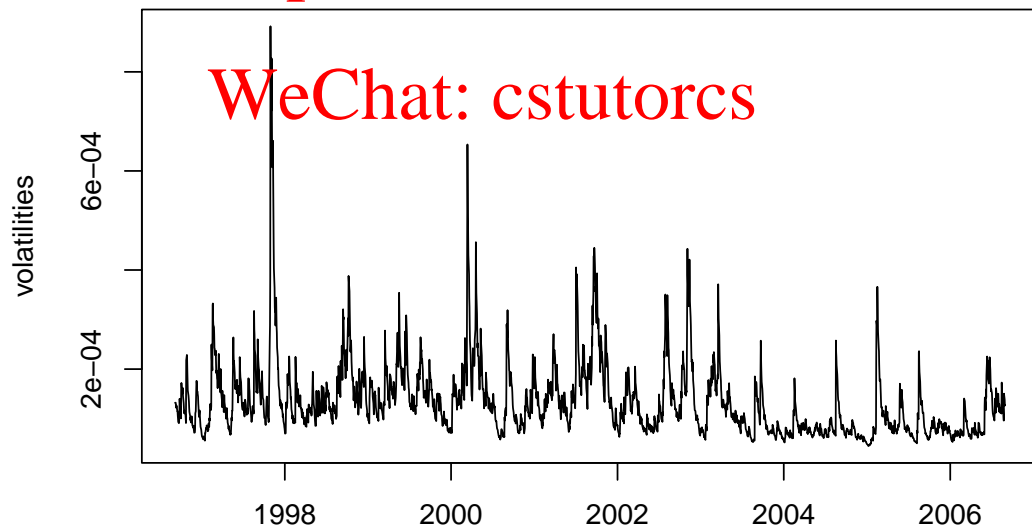
volatilities

2e−04   6e−04

1998   2000   2002   2004   2006

ARMA(0, 2)−GARCH(1, 1)

volatilities

2e−04   6e−04

1998   2000   2002   2004   2006
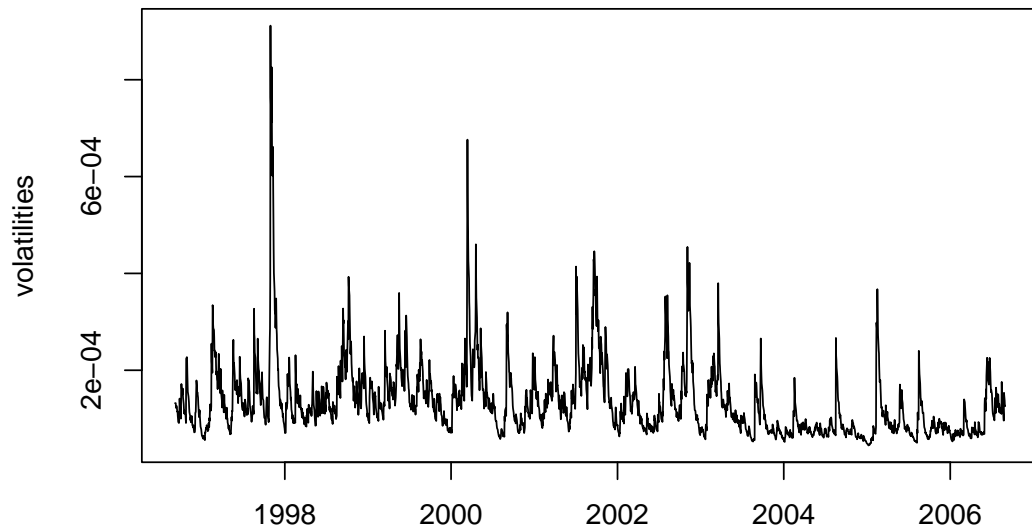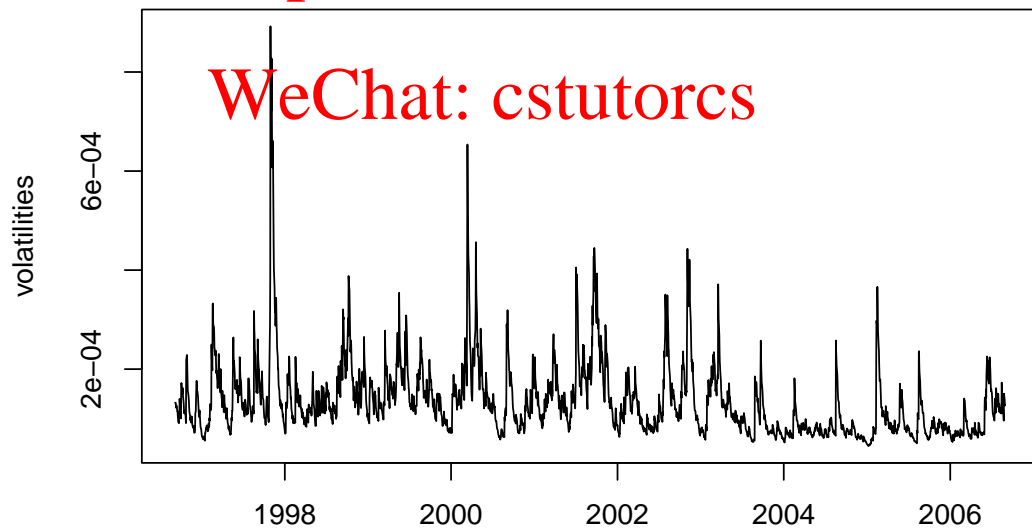
ARMA(0, 2)−GARCH(1, 2)

ARMA(0, 2)−GARCH(2, 1)

ARMA(0, 2)−GARCH(2, 2)

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

ARMA(1, 0)−GARCH(1, 1)
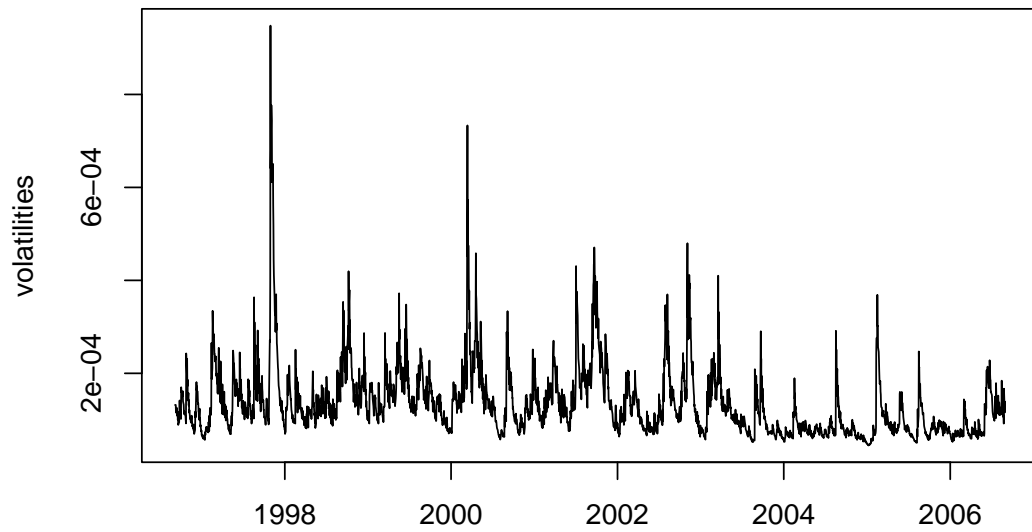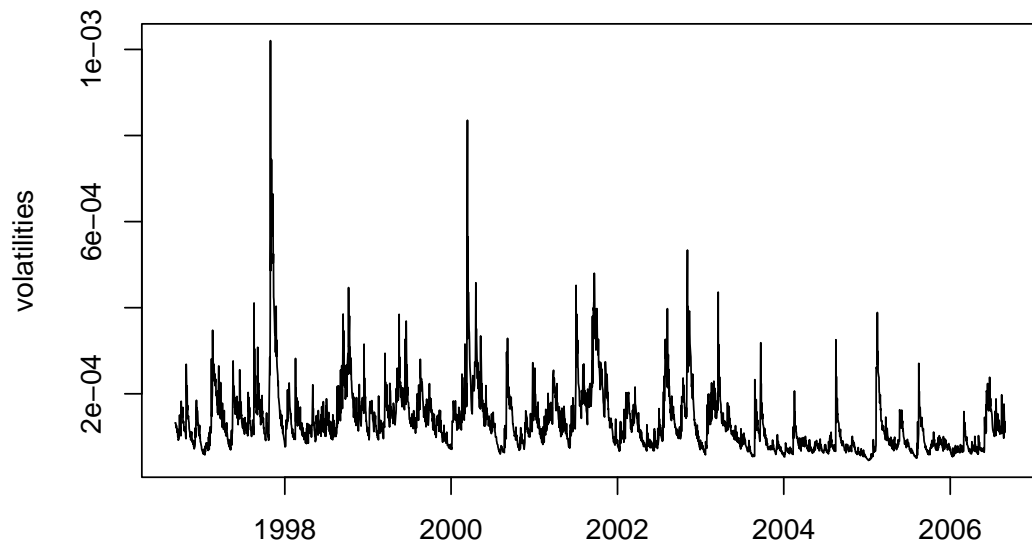
ARMA(1, 0)−GARCH(1, 2)

Assignment Project Exam Help

ARMA(1, 0)−GARCH(2, 1)

https://tutorcs.com
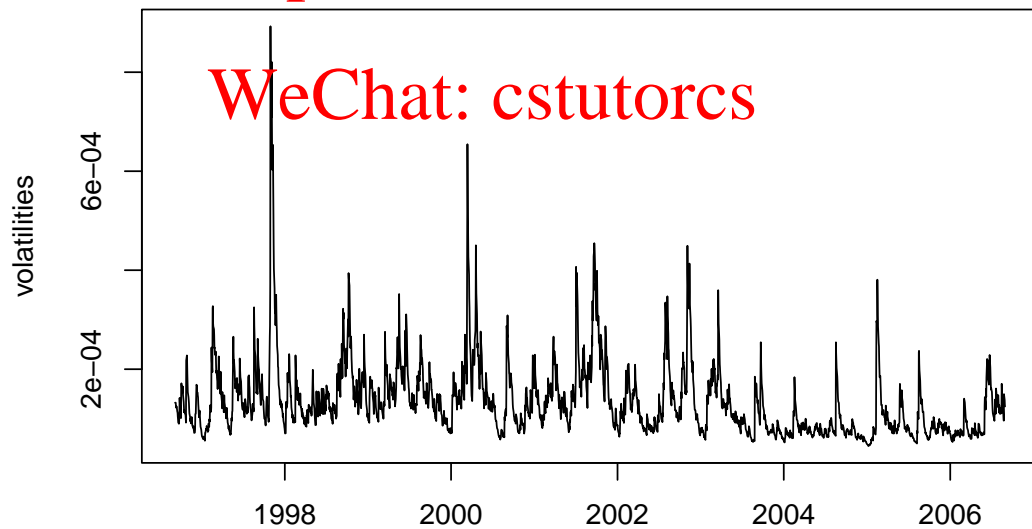
WeChat: cstutorcs

ARMA(1, 0)−GARCH(2, 2)

Assignment Project Exam Help

https://tutorcs.com
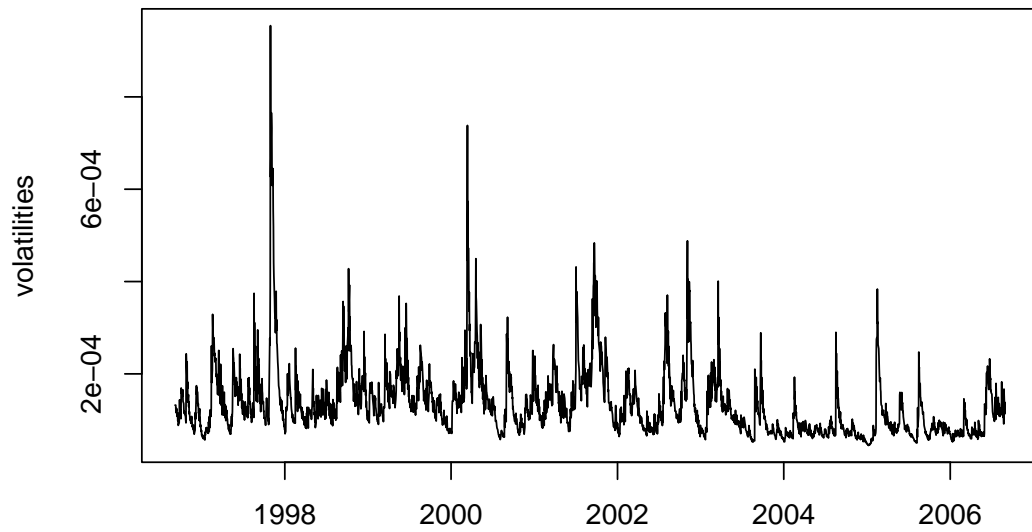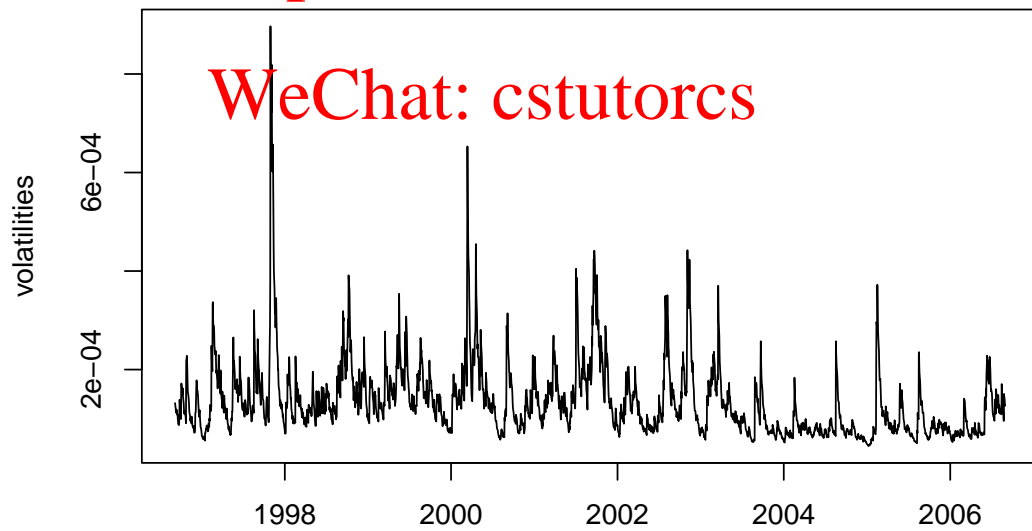
ARMA(1, 1)−GARCH(1, 1)

WeChat: cstutorcs

ARMA(1, 1)−GARCH(1, 2)

Assignment Project Exam Help

https://tutorcs.com
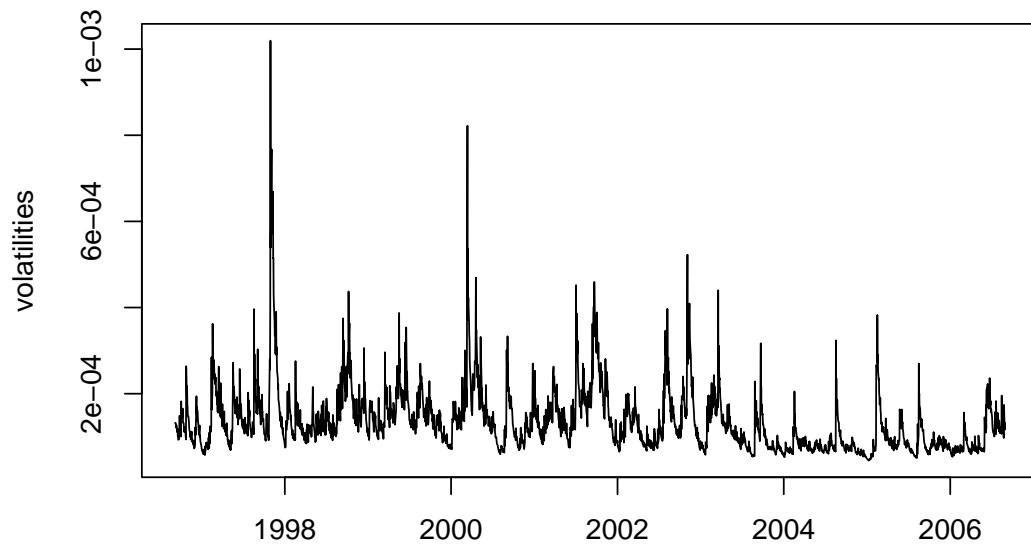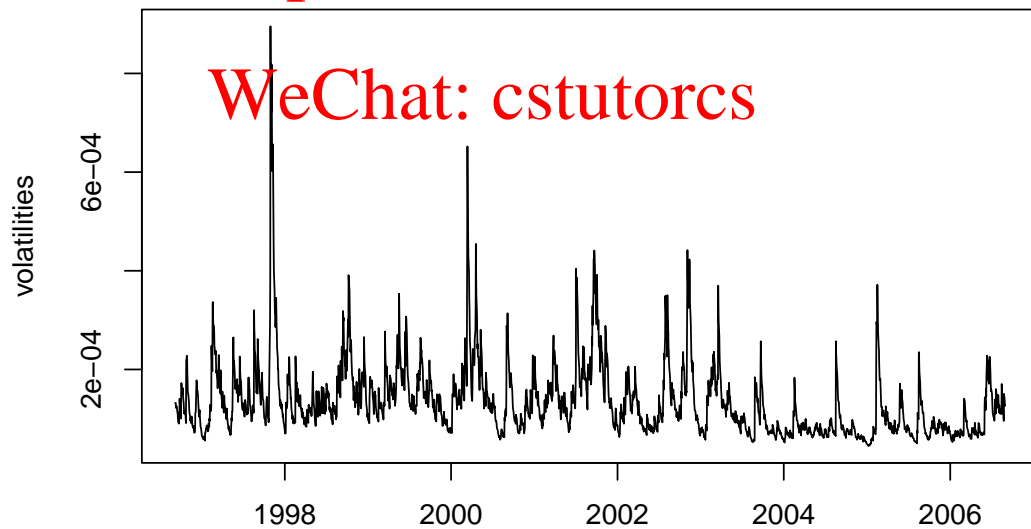
ARMA(1, 1)−GARCH(2, 1)

WeChat: cstutorcs

ARMA(1, 1)−GARCH(2, 2)

Assignment Project Exam Help

https://tutorcs.com
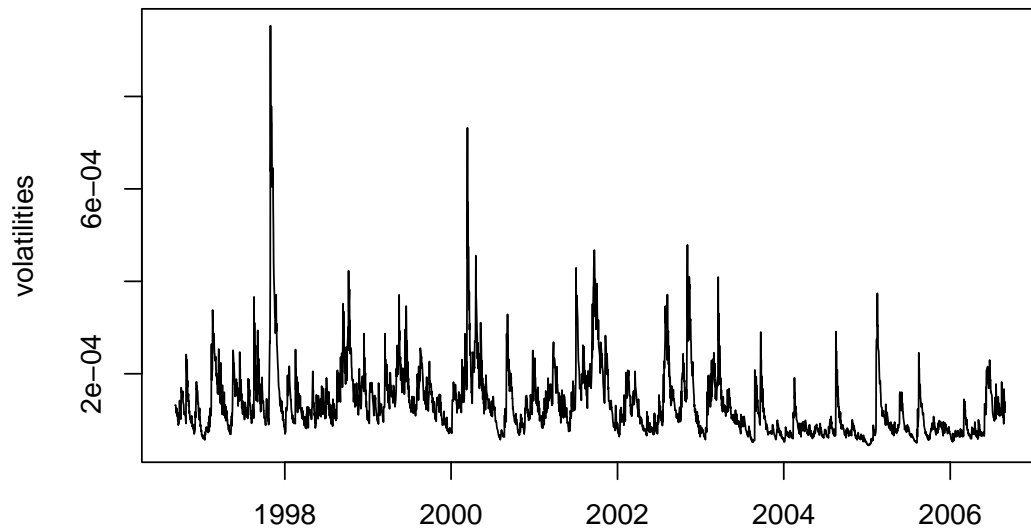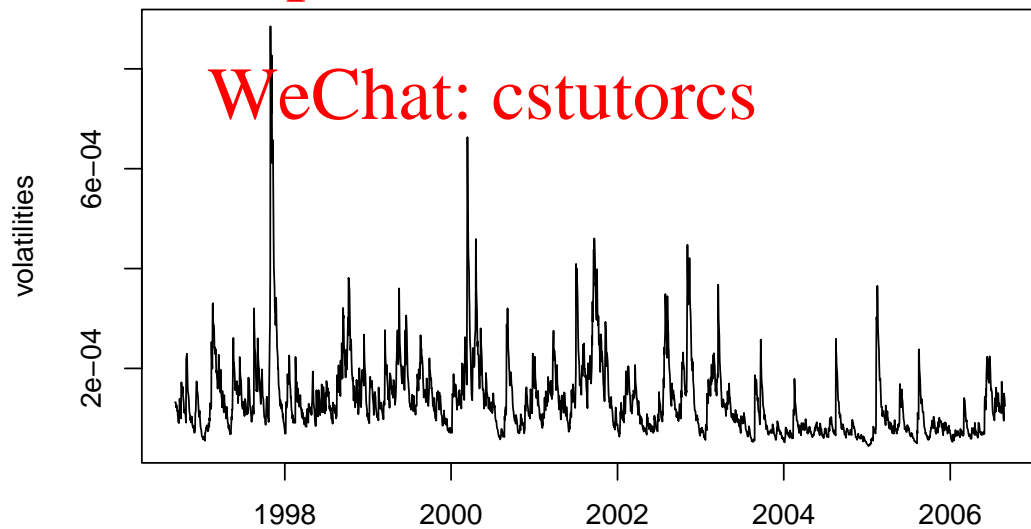
ARMA(1, 2)−GARCH(1, 1)

WeChat: cstutorcs

ARMA(1, 2)−GARCH(1, 2)

Assignment Project Exam Help

https://tutorcs.com
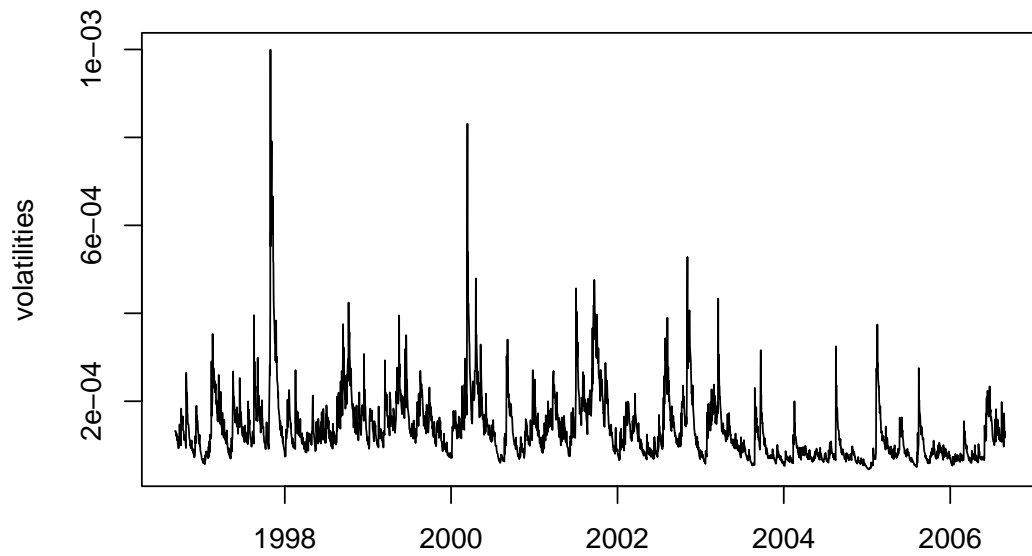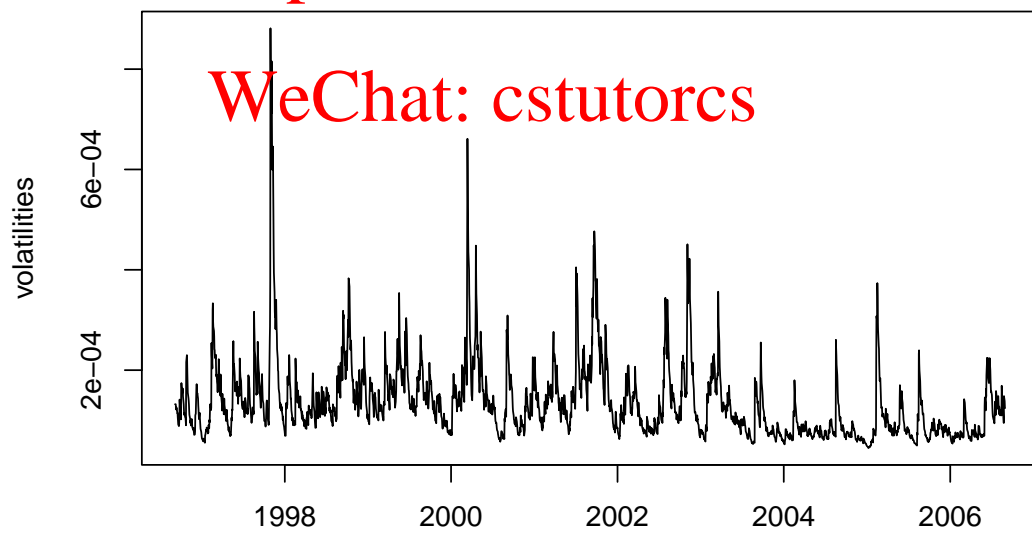
ARMA(1, 2)−GARCH(2, 1)

WeChat: cstutorcs

ARMA(1, 2)−GARCH(2, 2)

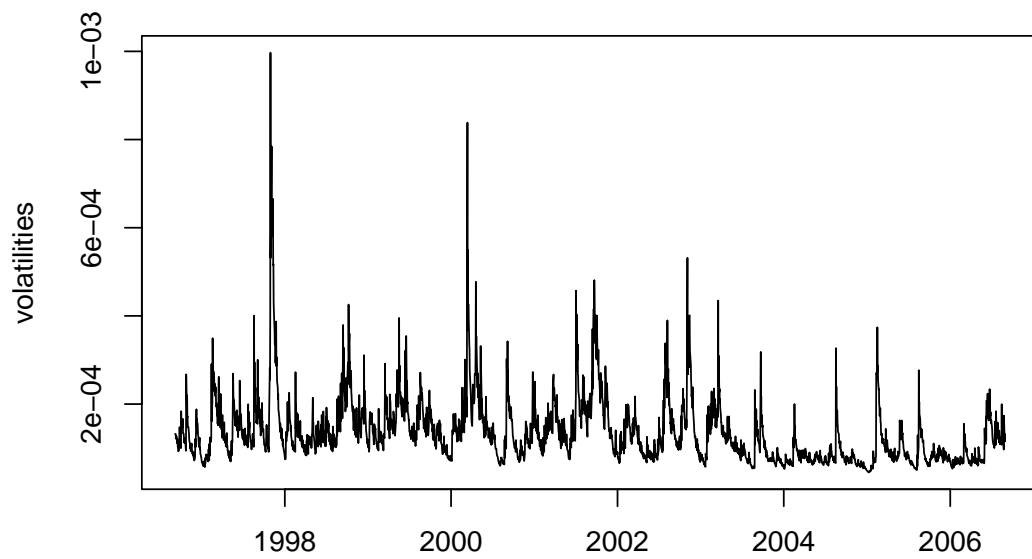
Assignment Project Exam Help

https://tutorcs.com

ARMA(2, 0)−GARCH(1, 1)

WeChat: cstutorcs
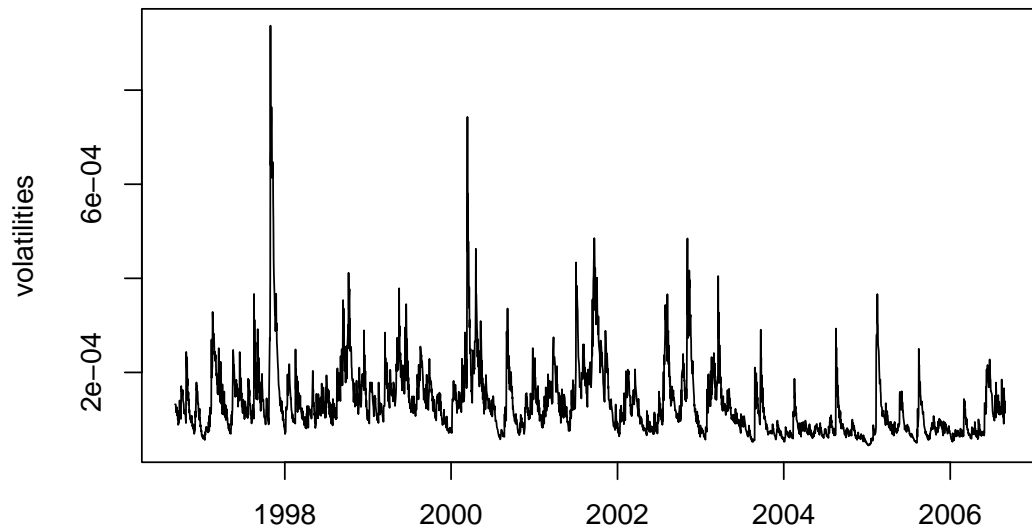
ARMA(2, 0)−GARCH(1, 2)
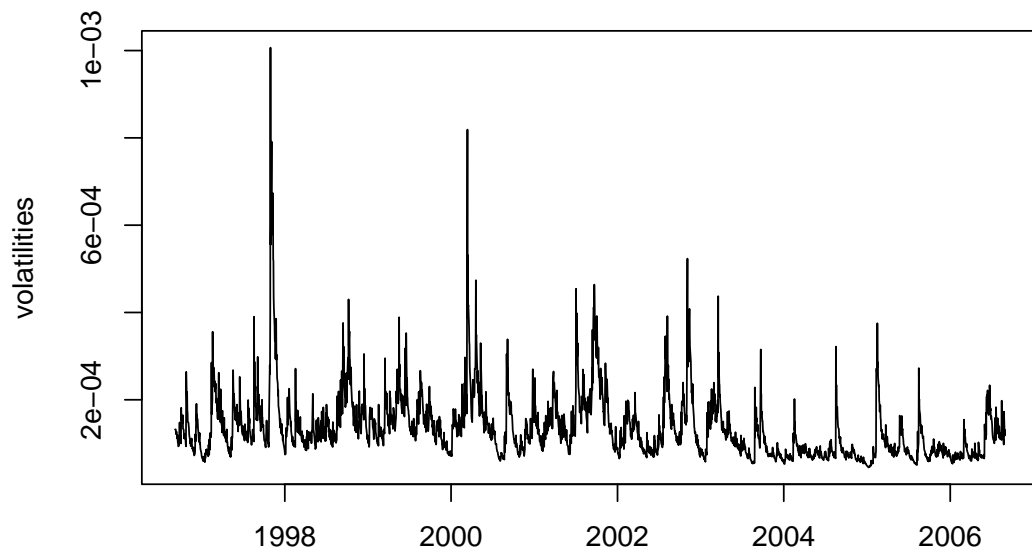
Assignment Project Exam Help

ARMA(2, 0)−GARCH(2, 1)

https://tutorcs.com
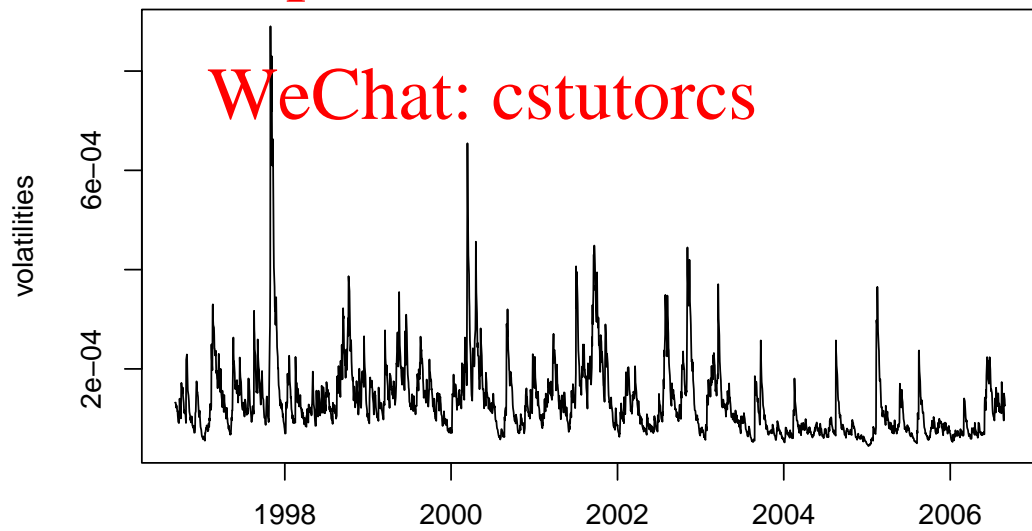
WeChat: cstutorcs

ARMA(2, 0)−GARCH(2, 2)

Assignment Project Exam Help

https://tutorcs.com
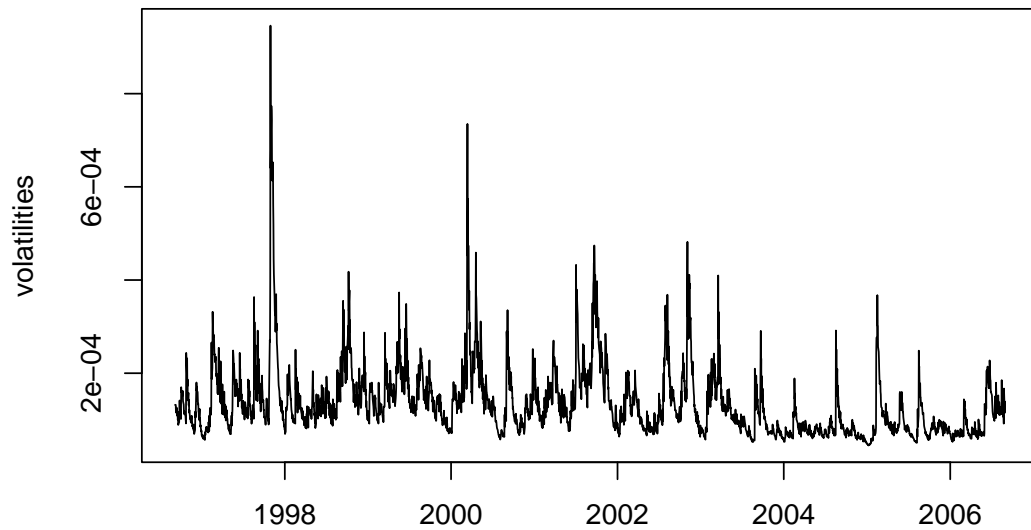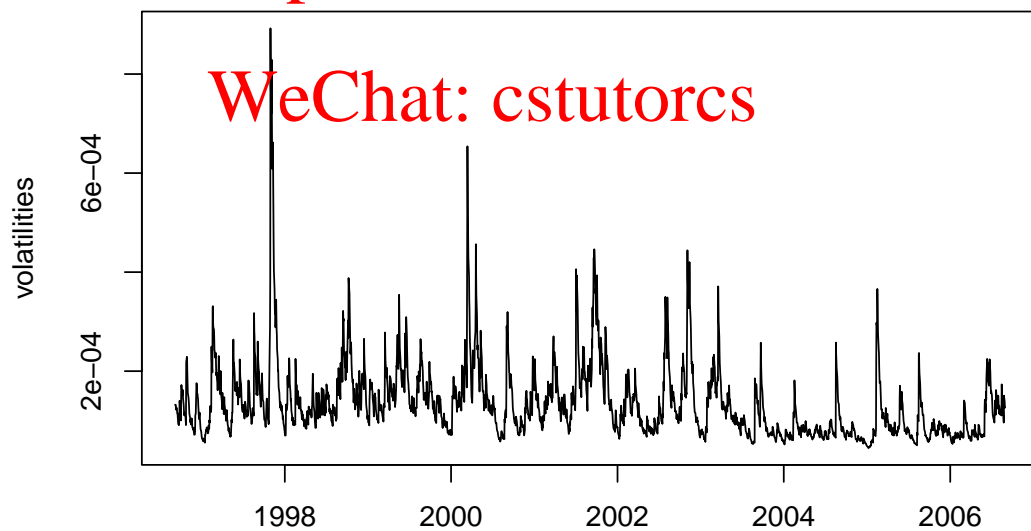
ARMA(2, 1)−GARCH(1, 1)

WeChat: cstutorcs

ARMA(2, 1)–GARCH(1, 2)

Assignment Project Exam Help

https://tutorcs.com

ARMA(2, 1)–GARCH(2, 1)

WeChat: cstutorcs
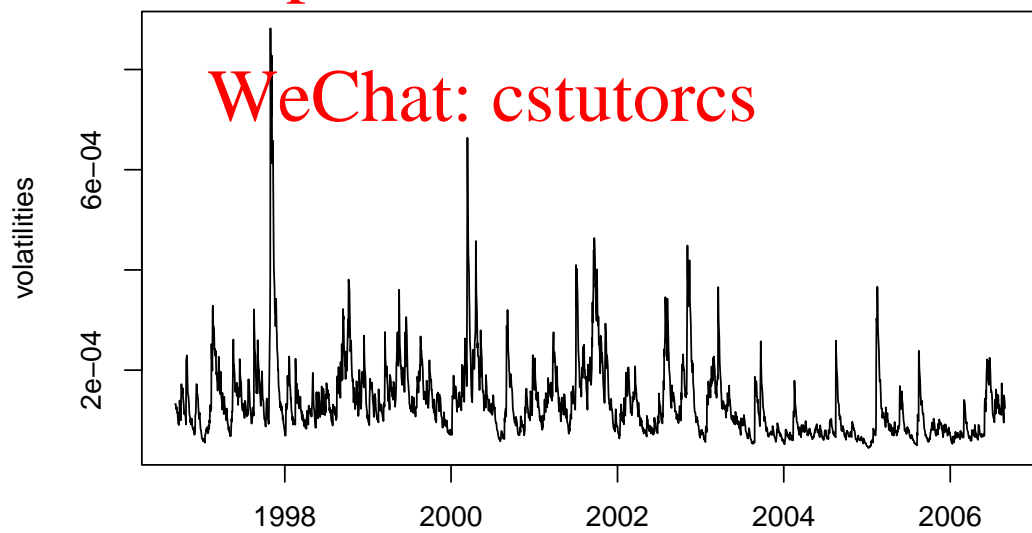
ARMA(2, 1)−GARCH(2, 2)

ARMA(2, 2)−GARCH(1, 1)
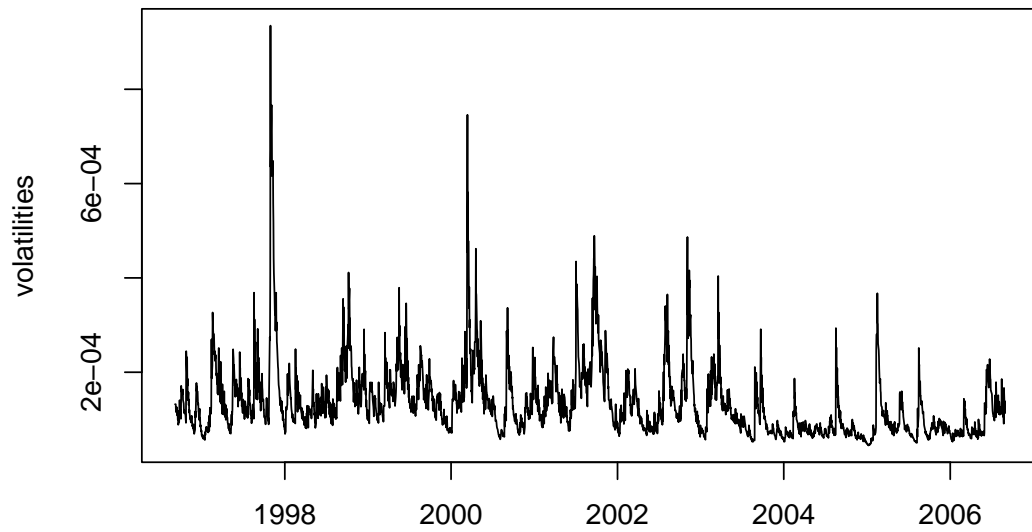
ARMA(2, 2)−GARCH(1, 2)

Assignment Project Exam Help

ARMA(2, 2)−GARCH(2, 1)

https://tutorcs.com
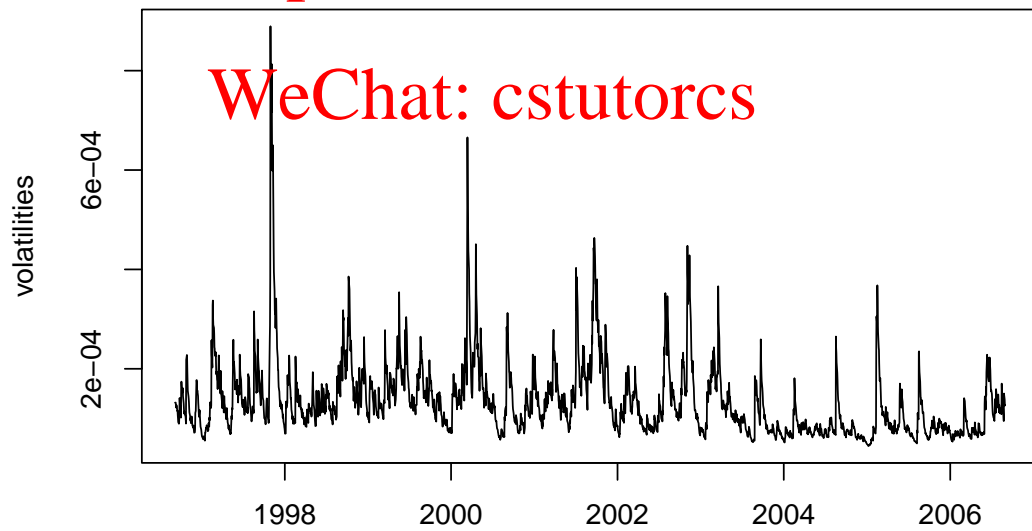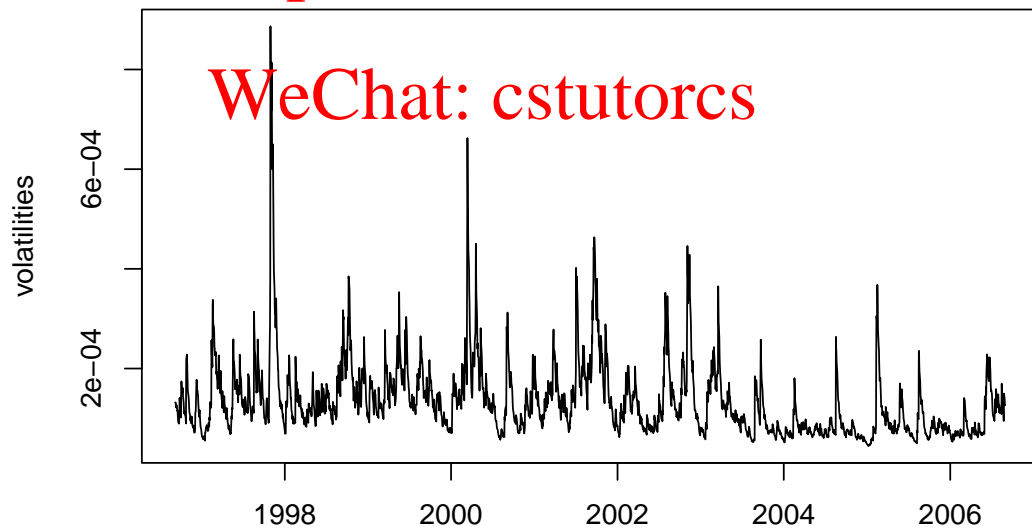
WeChat: cstutorcs

ARMA(2, 2)–GARCH(2, 2)
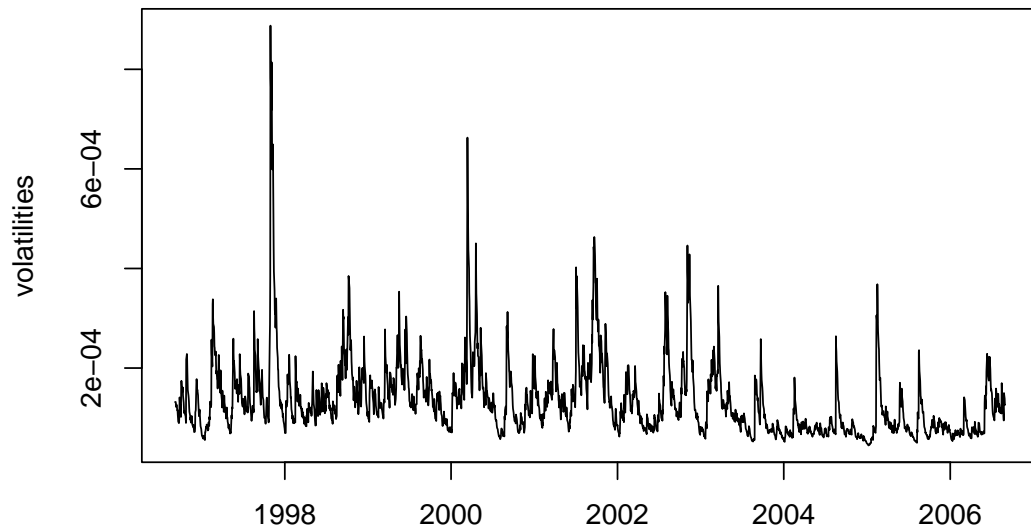
We notice some subtle differences across these specifications, but overall the estimated volatilities over the sample period follow a similar pattern. Hence, even though our adequate set contains many models, the implied high degree of model specification uncertainty is of little practical consequence for inference on volatilities. Put in a positive light: the inference we obtain regarding volatilities is robust to model specification uncertainty.