10/10 Questions Answered
Saved at 12:27 PM

IE 522 HW13

Q1
3 Points

Consider a GARCH(2,2) process $a_t = \sigma_t \epsilon_t$, $\epsilon_t \sim N(0,1)$, $\sigma_t = \sqrt{w + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2}$ with $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\beta_1 \geq 0$, $\beta_2 \geq 0$, $w > 0$, $\alpha_1 + \alpha_2 + \beta_1 + \beta_2 < 1$.

Q1.1
1 Point

Calculate $E[a_t|\mathcal{F}_{t-1}]$ and $E[a_t]$.

Choose Files No file chosen

Enter your answer here

CURRENTLY UPLOADED FILES

▾ Q1-1.png Download ✖ Remove

Q1

3 Points

Consider a GARCH(2,2) process $a_t = \sigma_t \epsilon_t$, $\epsilon_t \sim N(0,1)$, $\sigma_t = \sqrt{w + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2}$ with $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\beta_1 \geq 0$, $\beta_2 \geq 0$, $w > 0$, $\alpha_1 + \alpha_2 + \beta_1 + \beta_2 < 1$.

Q1.1

1 Point

Calculate $E[a_t | \mathcal{F}_{t-1}]$ and $E[a_t]$.

$E[a_t | \mathcal{F}_{t-1}] = E[\sigma_t \varepsilon_t | \mathcal{F}_{t-1}]$

$= \sigma_t E[\varepsilon_t | \mathcal{F}_{t-1}] = 0$

$E[a_t] = E[E[a_t | \mathcal{F}_{t-1}]] = 0$

Choose Files No file chosen

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q1.2

1 Point

Calculate $var(a_t|\mathcal{F}_{t-1})$.

Enter your answer here

Choose Files No file chosen

CURRENTLY UPLOADED FILES

Q1-2.png Download Remove

Q1.2
1 Point

Calculate $var(a_t|\mathcal{F}_{t-1})$.

$Var(a_t | \mathcal{F}_{t-1})$

$= E[a_t^2 | \mathcal{F}_{t-1}] = E[\sigma_t^2 \varepsilon_t^2 | \mathcal{F}_{t-1}]$

$= \sigma_t^2 E[\varepsilon_t | \mathcal{F}_{t-1}] = \sigma_t^2 Var(\varepsilon_t) = \sigma_t^2$

$= w + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2$

Choose Files No file chosen

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q1.3

1 Point

Calculate $\gamma_a(0) := var(a_t)$.

Enter your answer here

Choose Files No file chosen

CURRENTLY UPLOADED FILES

Q1-3.png Download Remove

Q1.3
1 Point

Calculate $\gamma_a(0) := var(a_t)$.

$var(a_t) = E(a_t^2) = E(\sigma_t^2)$

$= E[w + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + B_1 \sigma_{t-1}^2 + B_2 \sigma_{t-2}^2]$

$= w + \alpha_1 Var(a_{t-1}) + \alpha_2 var(a_{t-2})$

$+ B_1 var(a_{t-1}) + B_2 var(a_{t-2})$

$\gamma_a(0) = var(a_t), Var(a_{t-1}), var(a_{t-2})$

$\Rightarrow \gamma_a(0) = w + (\alpha_1 + B_2)\gamma_a(0) + (\alpha_2 + B_2)\gamma_a(0)$

$\Rightarrow \gamma_a(0) = \dfrac{w}{1 - (\alpha_1 + B_1) - (\alpha_2 + B_2)}$

Choose Files No file chosen

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q2

7 Points

Use the following code to get S&P 500 daily prices and log returns from https://finance.yahoo.com. `head()` and `tail()` show the first 6 and last 6 observations of the time series. The first return should be on 2000/01/03, and the last return should be on 2022/11/11. In the tests below, the significance level is 5%.

Choose Files No file chosen

```
uantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12", auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
```

Q2.1

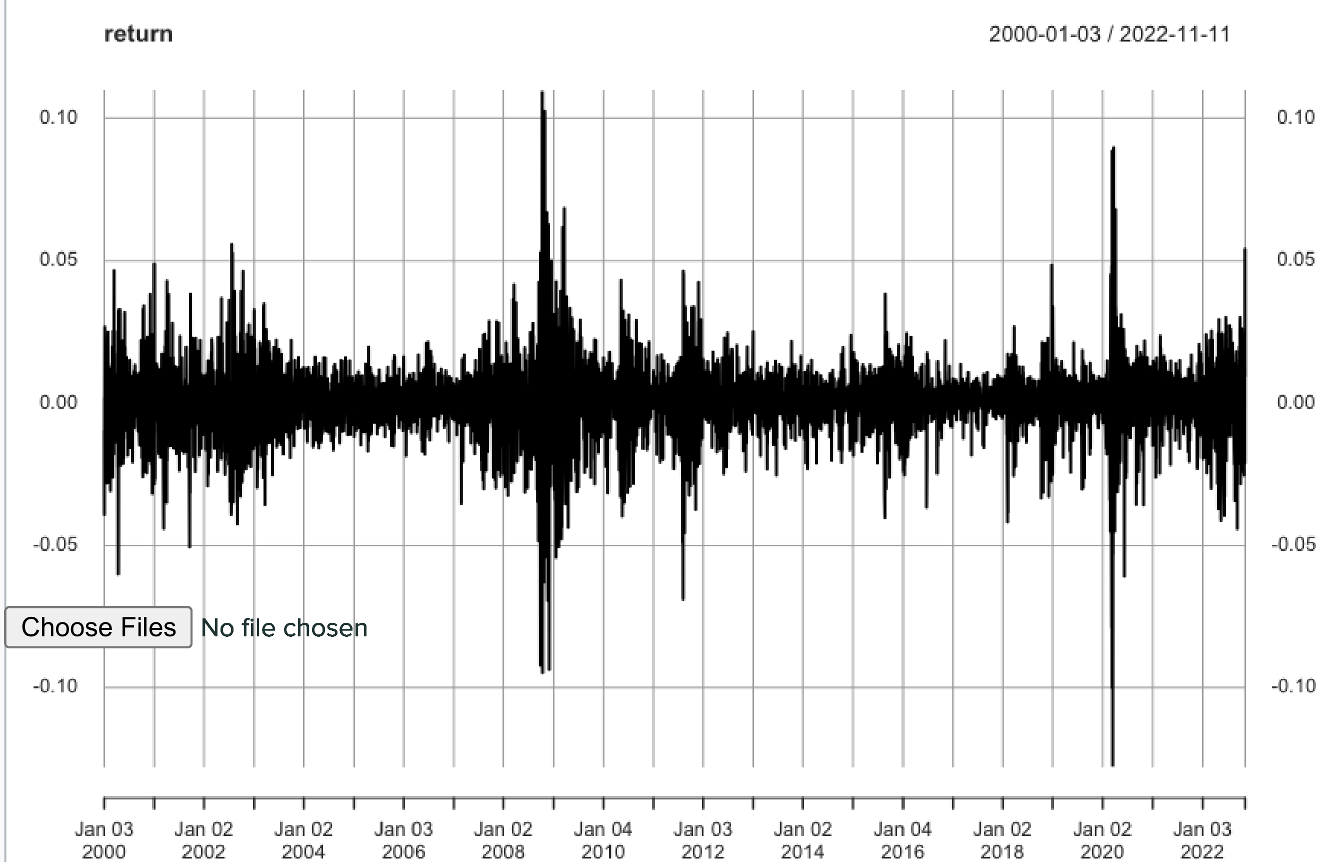
1 Point

Construct a time series plot for the daily log returns. Does it exhibit volatility clustering?

Yes, there is a volatile on Jan 02 2002 that follows Jan 03 2000, and also Jan 03 2012 after around Jan 03 2009.

CURRENTLY UPLOADED FILES

Q2-1_Plot.png   Download   Remove



Choose Files No file chosen

Q2-1.R   Download   Remove

```
#Q2-1
#Data Preparation
library(quantmod)
```

```r
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12",
auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#Construct a time series plot for the daily log returns.
plot(return)

```

Please select file(s) | Select file(s)

Save Answer | Last saved on **Nov 29 at 12:27 PM**

Q2.2

1 Point

We want to test whether the daily log returns are random (independent) or autocorrelated. Does the runs test show evidence that the daily log returns are non-random? The Ljung-Box test is another commonly used test, where H_0 is that data are independent and hence not autocorrelated, H_1 is that data are not independent and there exists autocorrelation. In R, use `Box.test(return,lag=10,type="Ljung-Box")` to conduct the Ljung-Box test for lags up to 10. Does it show evidence that the daily log returns are auto-correlated?

Choose Files | No file chosen

Runs Test

data: x

statistic = 4.6408, runs = 3054, n1 = 2877, n2 = 2877, n = 5754,

p-value = 3.47e-06

alternative hypothesis: nonrandomness

We conclude that return is not random since the p value is small enough for us to reject H0: Randomness.

Box-Ljung test

data: return
X-squared = 96.921, df = 10, p-value = 2.22e-16

We reject the H0: independent since the p-value is small. Hence, we conclude that return is autocorrelated.

CURRENTLY UPLOADED FILES

Q2-2.R Download Remove

```
#Q2-2
#Data Preparation
library(quantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12",
auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
```

Choose Files No file chosen

```
tail(return)
alpha = 0.05

#Run test
library(randtests)
x <- as.vector(return[,"daily.returns"])
runs.test(x,alternative = "two.sided", threshold = median(x), pvalue =
"normal", plot = FALSE)

#Ljung-Box test
Box.test(return,lag=10,type="Ljung-Box")
```

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q2.3

1 Point

The Augmented Dickey-Fuller Test is used to test whether a time series is stationary. H_0 is that there exists a unit root (AR component with coefficient 1) and it's not stationary, H_1 is that there is no unit root and it's stationary. This could be done in R using 'adf.test' function in `library(tseries)`. Is there evidence that there is no unit root in the daily log returns?

Augmented Dickey-Fuller Test

data: return

Dickey-Fuller = -18.101, Lag order = 17, p-value = 0.01

alternative hypothesis: stationary

Choose Files No file chosen --

We conclude that return is stationary since p value is small enough to be rejected.

CURRENTLY UPLOADED FILES

Q2-3.R Download ✖ Remove

```r
#Q2-3
#Data Preparation
library(quantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12",
auto_assign = TRUE)
```

```
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#Augmented Dickey-Fuller Test
library(tseries)
adf.test(return)

```

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q2.4

1 Point

Using `arima()`, find the best ARMA(p,q) model for the daily log returns with $0 \leq p \leq 3, 0 \leq q \leq 3$. Denote the daily log return on day t by Y_t. Write down the equation of the best model, including the distribution for the white noise (don't use `auto.arima` since it might pick a model where mean is set to zero). What's the AIC of the best model?

Choose Files No file chosen

```
[1] "Find the best ARMA Model:"
[1] "p = 0 ,  q = 0 ,  AIC = -34066.4839765114"
[1] "p = 0 ,  q = 1 ,  AIC = -34129.3491511865"
[1] "p = 0 ,  q = 2 ,  AIC = -34127.4834424758"
[1] "p = 0 ,  q = 3 ,  AIC = -34125.6811487401"
[1] "p = 1 ,  q = 0 ,  AIC = -34127.9372075773"
[1] "p = 1 ,  q = 1 ,  AIC = -34127.4617479751"
[1] "p = 1 ,  q = 2 ,  AIC = -34126.6751340284"
[1] "p = 1 ,  q = 3 ,  AIC = -34123.7865976053"
```

```
[1] "p = 2 ,  q = 0 ,  AIC = -34127.6617590531"
[1] "p = 2 ,  q = 1 ,  AIC = -34125.5744935221"
[1] "p = 2 ,  q = 2 ,  AIC = -34123.8282991566"
[1] "p = 2 ,  q = 3 ,  AIC = -34121.7914499761"
[1] "p = 3 ,  q = 0 ,  AIC = -34126.112121182"
[1] "p = 3 ,  q = 1 ,  AIC = -34124.0002462696"
[1] "p = 3 ,  q = 2 ,  AIC = -34122.0164945384"
[1] "p = 3 ,  q = 3 ,  AIC = -34123.7734978826"
[1] "--------------------------------------------------------------------------------"
[1] "The best fit is the one with the minimal AIC value:"
[1] "The best ARMA model is ARMA( 0 , 1 ) Model, AIC =  -34129.3491511865"

Call:
arima(x = x, order = c(smallest_p, 0, smallest_q))

Coefficients:
          ma1  intercept
      -0.1070      2e-04
s.e.   0.0132      1e-04

Choose Files No file chosen
timated as 0.0001553:  log likelihood = 17067.67,  aic = -34129.35
[1] "--------------------------------------------------------------------------------"
```

$$Y_t = 0.0002 + \epsilon_t + -0.1070\epsilon_{t-1}$$

$$AIC = -34129.35$$

CURRENTLY UPLOADED FILES

Q2-4.R

Download ✖ Remove

```
#Q2-4
#Data Preparation
```

```
library(quantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12", auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#find the best arima
find_the_best_ARMA <- function(x){
  smallest_p = 0
  smallest_q = 0
  smallest_AIC = 0
  print("Find the best ARMA Model:")

  for (p in 0:3){
    for (q in 0:3){
      fit = arima(x, order = c(p,0,q))
      print(paste("p =", p, ", ", "q =", q, ", ", "AIC =", AIC(fit)))
      if (smallest_AIC > AIC(fit)){
        smallest_AIC = AIC(fit)
        smallest_p = p
        smallest_q = q
      }
Choose Files No file chosen
  }
  print("---------------------------------------------------------------------------------")
  print("The best fit is the one with the minimal AIC value:")
  print(paste("The best ARMA model is ARMA(", smallest_p, ",", smallest_q,") Model, AIC = ", smallest_AIC))
  fit4 = arima(x, order = c(smallest_p,0,smallest_q))
  print(fit4)
  print("---------------------------------------------------------------------------------")
}
find_the_best_ARMA(return)
```

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q2.5

1 Point

The `garchFit` function in `library(fGarch)` only reports scaled AIC, that is, AIC/length of the series. If `result=garchFit(...)` is the fitting result, AIC can be obtained with `result@fit$ics[1]*nrow(return)`. Using AIC as the criteria, find the best model among ARMA(i,j)/GARCH(p,q), where $0 \leq i \leq 2, 0 \leq j \leq 2, 1 \leq p \leq 2, 0 \leq q \leq 2$ (54 models in total). Write down the equation of the best model. Compare the AIC of the best model in 2.5 and the one in 2.4.

Choose Files No file chosen

```
[1] "Find the best GARCH Model:"
[1] "i, j, p, q = 0 , 0 , 1 , 0 , AIC = -34964.6097855521"
[1] "i, j, p, q = 0 , 0 , 1 , 1 , AIC = -36987.0445508525"
[1] "i, j, p, q = 0 , 0 , 1 , 2 , AIC = -36985.4333670941"
[1] "i, j, p, q = 0 , 0 , 2 , 0 , AIC = -35909.7204044629"
[1] "i, j, p, q = 0 , 0 , 2 , 1 , AIC = -36994.9842070349"
[1] "i, j, p, q = 0 , 0 , 2 , 2 , AIC = -36997.2988781183"
[1] "i, j, p, q = 0 , 1 , 1 , 0 , AIC = -35093.1649378015"
[1] "i, j, p, q = 0 , 1 , 1 , 1 , AIC = -37003.7266029695"
```

```
[1] "i, j, p, q = 0 , 1 , 1 , 2 , AIC = -37003.0005888905"
[1] "i, j, p, q = 0 , 1 , 2 , 0 , AIC = -35915.5292905145"
[1] "i, j, p, q = 0 , 1 , 2 , 1 , AIC = -37013.2400450684"
[1] "i, j, p, q = 0 , 1 , 2 , 2 , AIC = -37014.8620147826"
[1] "i, j, p, q = 0 , 2 , 1 , 0 , AIC = -35113.1765883622"
[1] "i, j, p, q = 0 , 2 , 1 , 1 , AIC = -37013.006642762"
[1] "i, j, p, q = 0 , 2 , 1 , 2 , AIC = -37011.4357192459"
[1] "i, j, p, q = 0 , 2 , 2 , 0 , AIC = -35929.4764288811"
[1] "i, j, p, q = 0 , 2 , 2 , 1 , AIC = -37021.7285688839"
[1] "i, j, p, q = 0 , 2 , 2 , 2 , AIC = -37023.8277883297"
[1] "i, j, p, q = 1 , 0 , 1 , 0 , AIC = -35091.9346436501"
[1] "i, j, p, q = 1 , 0 , 1 , 1 , AIC = -37002.6482205327"
[1] "i, j, p, q = 1 , 0 , 1 , 2 , AIC = -37001.9634564251"
[1] "i, j, p, q = 1 , 0 , 2 , 0 , AIC = -35914.9824551822"
[1] "i, j, p, q = 1 , 0 , 2 , 1 , AIC = -37012.1526433613"
[1] "i, j, p, q = 1 , 0 , 2 , 2 , AIC = -37013.8081397373"
[1] "i, j, p, q = 1 , 1 , 1 , 0 , AIC = -35092.7078837632"
[1] "i, j, p, q = 1 , 1 , 1 , 1 , AIC = -37012.2558461202"
[1] "i, j, p, q = 1 , 1 , 1 , 2 , AIC = -37011.1288380256"
[1] "i, j, p, q = 1 , 1 , 2 , 0 , AIC = -35928.8258286894"
Choose Files No file chosen 1 , 1 , 2 , 1 , AIC = -37021.7789607776"
[1] "i, j, p, q = 1 , 1 , 2 , 2 , AIC = -37023.9449796672"
[1] "i, j, p, q = 1 , 2 , 1 , 0 , AIC = -35111.360221783"
[1] "i, j, p, q = 1 , 2 , 1 , 1 , AIC = -37011.4176094034"
[1] "i, j, p, q = 1 , 2 , 1 , 2 , AIC = -37009.8277964388"
[1] "i, j, p, q = 1 , 2 , 2 , 0 , AIC = -35931.2963068677"
[1] "i, j, p, q = 1 , 2 , 2 , 1 , AIC = -37020.3175263243"
[1] "i, j, p, q = 1 , 2 , 2 , 2 , AIC = -37022.5351040867"
[1] "i, j, p, q = 2 , 0 , 1 , 0 , AIC = -35112.4418924947"
[1] "i, j, p, q = 2 , 0 , 1 , 1 , AIC = -37012.5218927331"
```

```
[1] "i, j, p, q = 2 , 0 , 1 , 2 , AIC = -37010.9507238099"
[1] "i, j, p, q = 2 , 0 , 2 , 0 , AIC = -35928.7349856275"
[1] "i, j, p, q = 2 , 0 , 2 , 1 , AIC = -37021.2053472111"
[1] "i, j, p, q = 2 , 0 , 2 , 2 , AIC = -37023.2959905097"
[1] "i, j, p, q = 2 , 1 , 1 , 0 , AIC = -35111.6040234149"
[1] "i, j, p, q = 2 , 1 , 1 , 1 , AIC = -37011.3521014812"
[1] "i, j, p, q = 2 , 1 , 1 , 2 , AIC = -37009.7578516936"
[1] "i, j, p, q = 2 , 1 , 2 , 0 , AIC = -35927.9366690353"
[1] "i, j, p, q = 2 , 1 , 2 , 1 , AIC = -37020.2536261821"
[1] "i, j, p, q = 2 , 1 , 2 , 2 , AIC = -37022.4701777692"
[1] "i, j, p, q = 2 , 2 , 1 , 0 , AIC = -35110.0248585506"
[1] "i, j, p, q = 2 , 2 , 1 , 1 , AIC = -37013.2780992923"
[1] "i, j, p, q = 2 , 2 , 1 , 2 , AIC = -37011.6373348586"
[1] "i, j, p, q = 2 , 2 , 2 , 0 , AIC = -35932.1497493543"
[1] "i, j, p, q = 2 , 2 , 2 , 1 , AIC = -37021.273683065"
[1] "i, j, p, q = 2 , 2 , 2 , 2 , AIC = -37023.5221536438"
[1] "--------------------------------------------------------------------------------"
[1] "The best fit is the one with the minimal AIC value:"
[1] "The best GARCH model is ARMA( 1 , 1 )| GARCH( 2 , 2 ) Model, AIC =
-37023.9449796672"
```

Choose Files No file chosen

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = Formulax(smallest_i, smallest_j, smallest_p,
    smallest_q), data = return, cond.dist = "norm", trace = FALSE)

Mean and Variance Equation:
 data ~ arma(1, 1) + garch(2, 2)
```

```
<environment: 0x7ff3f9fa7110>
 [data = return]

Conditional Distribution:
 norm

Coefficient(s):
        mu         ar1         ma1       omega      alpha1      alpha2       beta1
 1.7250e-04   7.2105e-01  -7.7157e-01   4.1956e-06   8.7558e-02   1.3365e-01   1.7393e-01
     beta2
 5.7718e-01

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu      1.725e-04   6.940e-05    2.486   0.0129 *
ar1     7.211e-01   1.085e-01    6.648 2.98e-11 ***
ma1    -7.716e-01   9.998e-02   -7.717 1.20e-14 ***
[illegible]96e-06   5.601e-07    7.491 6.84e-14 ***
alpha1  8.756e-02   1.373e-02    6.379 1.79e-10 ***
alpha2  1.336e-01   1.681e-02    7.948 2.00e-15 ***
beta1   1.739e-01   1.401e-01    1.241   0.2144
beta2   5.772e-01   1.262e-01    4.574 4.78e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 18519.97    normalized:  3.218626
```

Description:
Tue Nov 29 01:51:51 2022 by user:

[1] "--"

$$Y_t - 0.00017250 = 0.72105(Y_{t-1} - 0.00017250) + \epsilon_t - 0.77157\epsilon_{t-1}$$

$$a_t = \sigma_t\epsilon_t, \sigma_t =$$

$$\sqrt{0.0000041956 + 0.087558a_{t-1}^2 + 0.13365a_{t-2}^2 + 0.17393\sigma_{t-1}^2 + 0.57718\sigma_{t-2}^2}$$

$AIC = -37023.9449796672$, it is smaller than the one in Q2-4, means it fits better.

CURRENTLY UPLOADED FILES

Q2-5.R   Download   Remove

```
#Q2-5
#Data Preparation
library(quantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12", auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#GARCH fit
library(fGarch)
find_the_best_GARCH <- function(x){
  smallest_p = 0
  smallest_q = 0
  smallest_i = 0
  smallest_j = 0
  smallest_AIC = 0
  print("Find the best GARCH Model:")
```

Choose Files No file chosen

```
  for (i in 0:2){
    for (j in 0:2){
      for (p in 1:2){
        for (q in 0:2){
          Formula<-function(i, j, p, q) as.formula(paste('~arma(',i,',',j,')+garch(',p,',',q,')'))
          fit = garchFit(formula = Formula(i, j, p, q), data = x, cond.dist = "norm", trace = FALSE)
          AIC = fit@fit$ics[1]*nrow(return)
          print(paste("i, j, p, q =", i,",", j,",", p,",", q, ", AIC =", AIC))
          if (smallest_AIC > AIC){
            smallest_AIC = AIC
            smallest_p = p
            smallest_q = q
            smallest_i = i
            smallest_j = j
          }
        }
      }
    }
  }

  print("------------------------------------------------------------------------------------")
  print("The best fit is the one with the minimal AIC value:")
  print(paste("The best GARCH model is ARMA(", smallest_i, ",", smallest_j,") GARCH(",smallest_p, ",", smallest_q,") Model, AIC = ", smallest_AIC))
  Formula5<-function(i, j, p, q) as.formula(paste('~arma(',i,',',j,')+garch(',p,',',q,')'))
  fit5 = garchFit(formula = Formula5(smallest_i, smallest_j, smallest_p, smallest_q), data = return, cond.dist = "norm", trace = FALSE)
  print(fit5)
  print("------------------------------------------------------------------------------------")
}
find_the_best_GARCH(return)
```

Please select file(s) | Select file(s)

Save Answer | Last saved on **Nov 29 at 12:27 PM**

Q2.6

1 Point

For the best model chosen in 2.5, construct a normal probability plot for the residuals. According to the probability plot, is normal distribution good enough for ϵ_t?

No, it does not fit to the normal line.

CURRENTLY UPLOADED FILES

Q2-6.R | Download | ✖ Remove

Choose Files | No file chosen

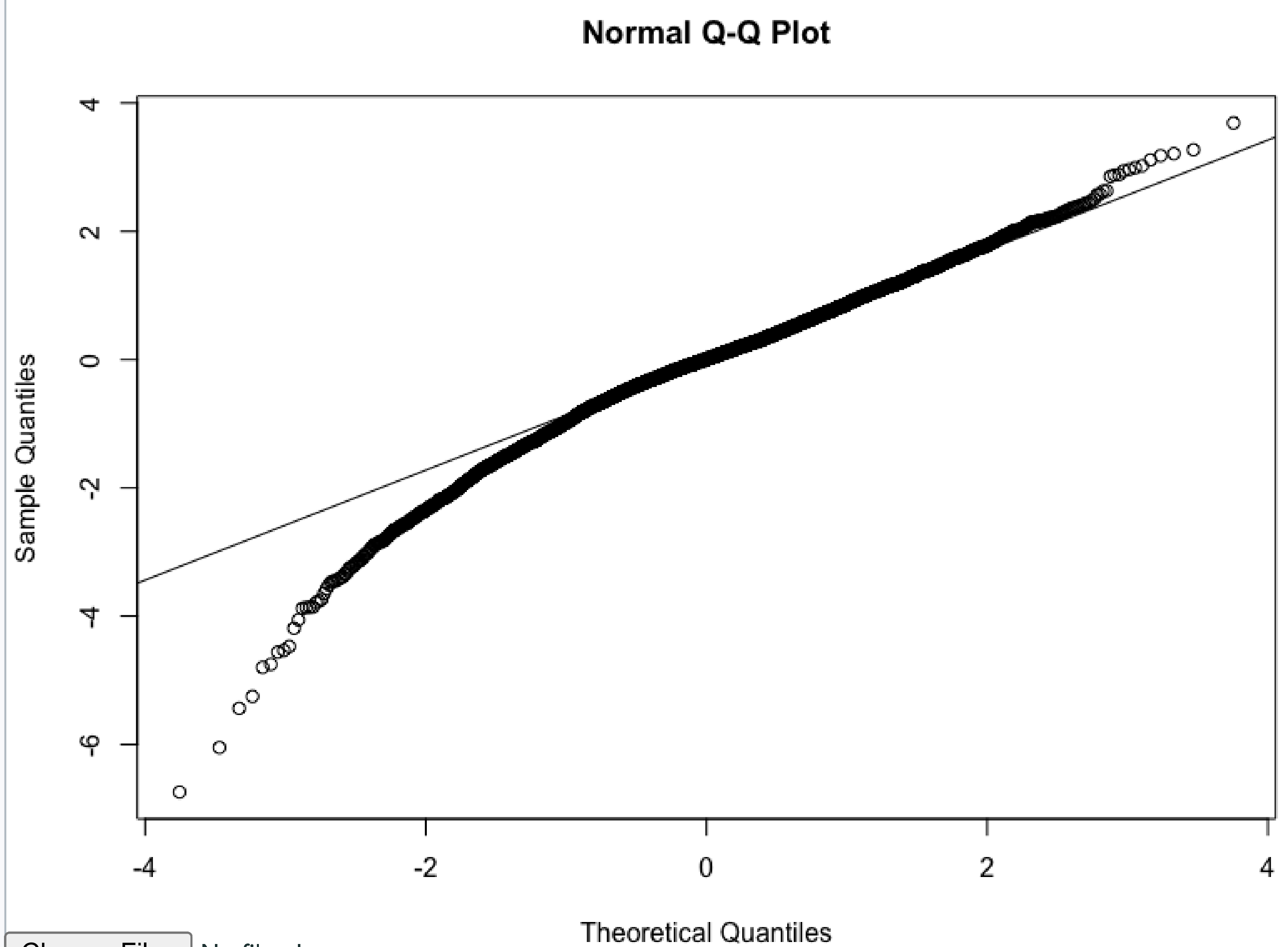
```
#Q2-6
#Data Preparation
library(quantmod)
bols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12",
auto_assign = TRUE)
return=dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#Get GARCH model
smallest_p = 0
smallest_q = 0
smallest_i = 0
smallest_j = 0
smallest_AIC = 0
```

```r
for (i in 0:2){
  for (j in 0:2){
    for (p in 1:2){
      for (q in 0:2){
        Formula<-function(i, j, p, q)
as.formula(paste('~arma(',i,',',j,')+garch(',p,',',q,')'))
        fit = garchFit(formula = Formula(i, j, p, q), data = return, cond.dist
= "norm", trace = FALSE)
        AIC = fit@fit$ics[1]*nrow(return)
        if (smallest_AIC > AIC){
          smallest_AIC = AIC
          smallest_p = p
          smallest_q = q
          smallest_i = i
          smallest_j = j
        }
      }
    }
  }
}

Formula6<-function(i, j, p, q)
as.formula(paste('~arma(',i,',',j,')+garch(',p,',',q,')'))
fit6 = garchFit(formula = Formula6(smallest_i, smallest_j, smallest_p,
smallest_q), data = return, cond.dist = "norm", trace = FALSE)
print(fit6)

#Get residual
res6 = residuals(fit6, standardize = T)
qqnorm(res6)
qqline(res6)
```

Choose Files No file chosen

QQPlot_Q2_6.png

Download ✖ Remove



Choose Files No file chosen

Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Q2.7

1 Point

Replace the normal distribution by `sged` in the model chosen in 2.5 (`cond.dist="sged"`). According to AIC, is sged better than the normal distribution for ϵ_t? Use `qqplot(res,rsged(10000,nu=shape parameter,xi=skew parameter))` to construct a probability plot for residuals. Here shape and skew parameters are the numeric estimates you obtain in `garchFit`. According to the probability plot, is the `sged` distribution a better candidate for ϵ_t?

$$AIC = -37383.07$$

Yes, it is better normal since it has smaller AIC value.

Yes, it fits the normal line better.

CURRENTLY UPLOADED FILES

Q2-7.R

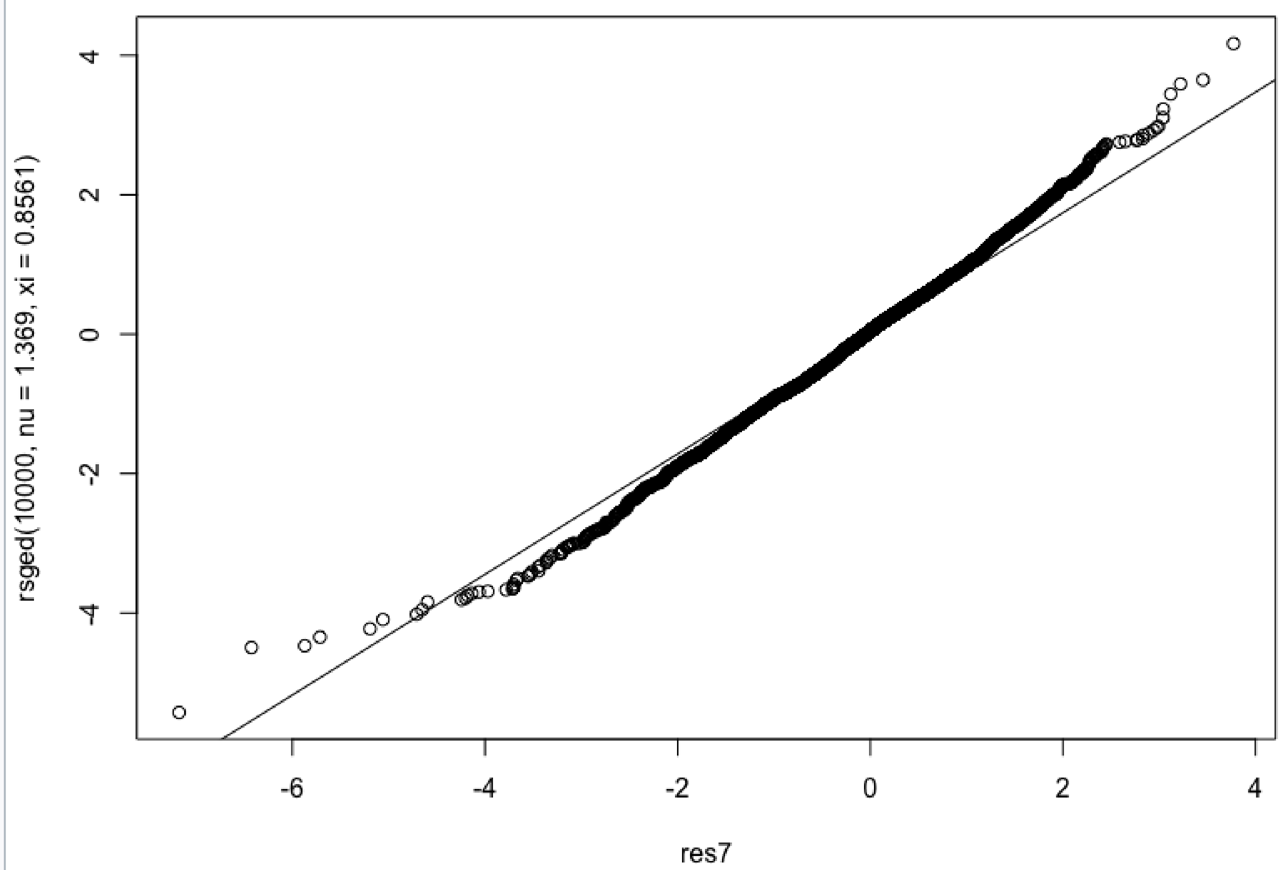
Download ✖ Remove

Choose Files No file chosen

```
#Q2-7
#Data Preparation
library(quantmod)
getSymbols(Symbols = "^GSPC", from = "2000-01-01", to = "2022-11-12",
auto_assign = TRUE)
return = dailyReturn(GSPC,type="log")
head(return)
tail(return)
alpha = 0.05

#Get sged model
Formula7<-function(i, j, p, q)
as.formula(paste('~arma(',i,',',j,')+garch(',p,',',q,')'))
fit7 = garchFit(formula = Formula7(smallest_i, smallest_j, smallest_p,
smallest_q), data = return, cond.dist = "sged", trace = FALSE)
AIC7 = fit7@fit$ics[1]*nrow(return)
print(AIC7)
```

```r

#QQPlot
res7 = residuals(fit7, standardize = T)
qqplot(res7,rsged(10000,nu=1.369,xi=0.8561))
qqline(res7)

```

QQPlot_Q2-7.png Download Remove



Please select file(s) Select file(s)

Save Answer Last saved on **Nov 29 at 12:27 PM**

Save All Answers

Submit & View Submission ›