

Volatility Models - JPMorgan
stock

Code ▼

CW3

11 October, 2022

Volatility Models - JPMorgan stock

EWMA volatility model

Running the EWMA volatility model for JPM stock with two lambda values.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan stock

```
# Comparing EWMA models with lambda = 0.94 and lambda = 0.99
lambda <- c(0.94, 0.99)
sigmaEWMA <- xts(matrix(nrow = dim(log_returns_demean)[1], ncol = 1
  length(lambda)), order.by = index(log_returns_demean)) # Pre-
  allocation of estimated conditional volatility matrix
hEWMA <- xts(matrix(nrow = dim(log_returns_demean)[1], ncol = length
  h(lambda)), order.by = index(log_returns_demean)) # Pre-al
  location of estimated conditional variance matrix

hEWMA[1, ] <- var(x = log_returns_demean[1:30]) # Initialize condit
  ional variance by taking sample variance of the first 30 d
  ays

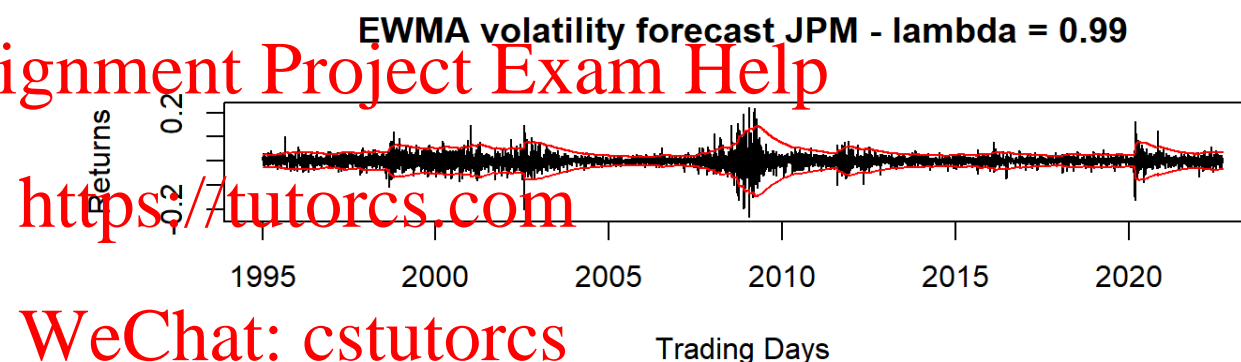
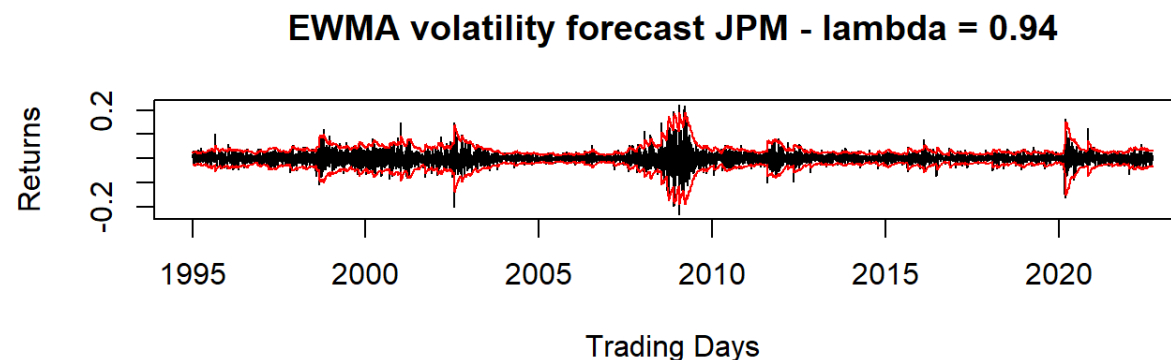
par(mfrow=c(2,1))
for (i in 1:2) {
  for (t in 2:dim(log_returns_demean)[1]) {
    hEWMA[t, i] <- lambda[i] * hEWMA[t-1, i] + (1-lambda[i]) * log_
      returns_demean[t-1]^2
    sigmaEWMA[t, i] <- sqrt(hEWMA[t, i])
  }
}
plot(x = index(sigmaEWMA), y = log_returns_demean, type = 'l', ma
  in = paste('EWMA volatility forecast JPM - lambda =', lamb
    da[i]), xlab = 'Trading Days', ylab = 'Returns')
lines(x = index(sigmaEWMA), y = 2 * sigmaEWMA[, i], col = 'red')
lines(x = index(sigmaEWMA), y = -2 * sigmaEWMA[, i], col = 'red')
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan stock



GARCH(1,1) Model

Estimate Garch(1,1) parameters and fit the conditional variance to returns.

We can specify the type of GARCH model we want to fit using the `ugarchspec()` function. For example, we can specify the number of lags, if the model includes leverage or power effects, whether the mean should be included or not, and the conditional distribution of the returns. The common syntax we will use inside `ugarchspec()` is:

Type of model: `variance.model = list(model = "type of model")`,

Volatility Models - JPMorgan stock

Whether to include or not a mean: `mean.model = list(armaOrder = c(0,0), include.mean = FALSE)`,

Type of conditional distribution: `distribution.model = "std"`

To get a detailed explanation on how to use all the options of `ugarchspec()`, you can type `?ugarchspec` in the console.

```
GARCH_1_1 <- ugarchspec(variance.model = list(model = "sGARCH", gar
  chOrder = c(1, 1)), # garchOrder = c(q,p) where q = alpha
    (ARCH) and p = beta (GARCH)
    mean.model = list(armaOrder = c(0, 0), incl
      ude.mean = FALSE)) # the default setting is to consider a
    non-zero mean which follows an ARMA(1,1) process
GARCH_1_1_fit <- ugarchfit(spec = GARCH_1_1, data = log_returns_dem
  ean, solver = 'hybrid') # estimate the model
sigmaGARCH_1_1 <- xts(x = GARCH_1_1_fit@fit$var, order.by = index(log_r
  eturns_demean))
sigmaGARCH_1_1 <- xts(x = GARCH_1_1_fit@fit$sigma, order.by = index
  (log_returns_demean))
```

Assignment Project Exam Help

<https://tutors.com>

WeChat: cstutorcs

Print full report provided rugarch package

```
GARCH_1_1_fit # print full report
```

Volatility Models - JPMorgan stock

```
##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : norm
##
## Optimal Parameters
## -----
##              Estimate  Std. Error  t value  Pr(>|t|)
## omega         0.000003    0.000002    1.9194  0.054939
## alpha1        0.076285    0.010920    6.9856  0.000000
## beta1         0.919316    0.011419   80.5101  0.000000
##
## Robust Standard Errors:
##              Estimate  Std. Error  t value  Pr(>|t|)
## omega         0.000003    0.000009    0.31891  0.74979
## alpha1        0.076285    0.061704    1.23630  0.21635
## beta1         0.919316    0.066037   13.92131  0.00000
##
## LogLikelihood : 18173.24
##
## Information Criteria
## -----
##
## Akaike          -5.2086
## Bayes           -5.2057
## Shibata         -5.2086
## Hannan-Quinn   -5.2076
```

Volatility Models - JPMorgan stock

```
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##               statistic p-value
## Lag[1]                0.001112  0.9734
## Lag[2*(p+q)+(p+q)-1][2]  0.294543  0.7988
## Lag[4*(p+q)+(p+q)-1][5]  1.319539  0.7841
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##               statistic p-value
## Lag[1]                5.169 0.02300
## Lag[2*(p+q)+(p+q)-1][5]  8.644 0.02035
## Lag[4*(p+q)+(p+q)-1][9]  10.009 0.04995
## d.o.f=1
##
## Weighted ARCH LM Tests
## -----
##               Statistic Shape Scale P-Value
## ARCH Lag[3]      0.5261 0.500 2.000  0.4683
## ARCH Lag[5]      0.6877 1.440 1.667  0.8272
## ARCH Lag[7]      1.7000 2.315 1.543  0.7805
##
## Nyblom stability test
## -----
## Joint Statistic:  27.9437
## Individual Statistics:
## omega  4.5371
## alpha1 0.2336
## beta1  0.3832
##
```

Assignment Project Exam Help
<https://tutorcs.com>
 WeChat: cstutorcs

Volatility Models - JPMorgan stock

```
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          0.846 1.01 1.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## -----
##              t-value      prob sig
## Sign Bias      0.5487 5.832e-01
## Negative Sign Bias 3.9847 6.824e-05 ***
## Positive Sign Bias 0.9095 3.631e-01
## Joint Effect      21.5506 8.090e-05 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##      group statistic p-value(g-1)
## 1      10      148.4      6.273e-21
## 2      30      156.3      2.128e-19
## 3      40      171.2      1.715e-18
## 4      50      193.8      4.116e-19
##
##
## Elapsed time : 0.202493
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Print only a selection of results

The function provides comprehensive information on the model fit. It includes the optimal parameters estimates, standard errors, tests for significance, likelihood, information criteria, and several other tests.

Since this is too much information in the same place, we can tell R what particular information we are interested in.

Volatility Models - JPMorgan stock

```
# Summarize parameter coefficients and t-statistic
param <- as.table(t(rbind(c(coef(GARCH_1_1_fit)["omega"],coef(GARCH_1_1_fit)["alpha1"], coef(GARCH_1_1_fit)["beta1"]),c(GARCH_1_1_fit@fit$tval['omega'],GARCH_1_1_fit@fit$tval['alpha1'], GARCH_1_1_fit@fit$tval['beta1']))))
      colnames(param) = c('Coefficient','t-Value')
knitr::kable(param, digits=6)
```

	Coefficient	t-Value
omega	0.000003	1.919358
alpha1	0.076285	6.985617
beta1	0.919316	80.510089

Assignment Project Exam Help

<https://tutores.com>

Extra code on how to access information

By using the @ symbol we can extract the contents from an object.

WeChat: cstutorcs

```
# Check the content of GARCH_1_1_fit
names(GARCH_1_1_fit@fit)
```


Volatility Models - JPMorgan stock

```
## [1] "hessian"          "cvar"              "var"              "sigm
a"
## [5] "condH"            "z"                 "LLH"              "log.
likelihoods"
## [9] "residuals"        "coef"              "robust.cvar"      "A"
## [13] "B"                "scores"            "se.coef"          "tva
l"
## [17] "matcoef"          "robust.se.coef"    "robust.tval"      "robu
st.matcoef"
## [21] "fitted.values"    "convergence"       "kappa"            "pers
istence"
## [25] "timer"            "ipars"             "solver"
```

Assignment Project Exam Help

```
# Extract a particular element
GARCH_1_1_fit@fit$coef
```

<https://tutorcs.com>

```
##          omega          alpha1          beta1
## 2.978548e-06 7.528451e-02 9.193160e-01
```

```
likelihood(GARCH_1_1_fit)
```

```
## [1] 18173.24
```

Plotting the GARCH output

A very useful feature of rugarch package is plotting. If we call the function `plot()` on a fitted GARCH model like `default_garch`, the following interactive menu will appear:

Make a plot selection (or 0 to exit):

1. Series with 2 Conditional SD Superimposed

Volatility Models - JPMorgan stock

2. Series with 1% VaR Limits
3. Conditional SD (vs |returns|)
4. ACF of Observations
5. ACF of Squared Observations
6. ACF of Absolute Observations
7. Cross Correlation
8. Empirical Density of Standardized Residuals
9. QQ-Plot of Standardized Residuals
10. ACF of Standardized Residuals
11. ACF of Squared Standardized Residuals
12. News-Impact Curve

We can visualize any of the plots by selecting the corresponding number.

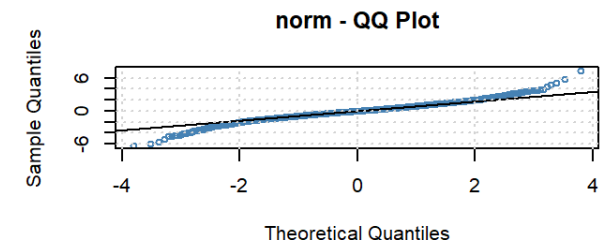
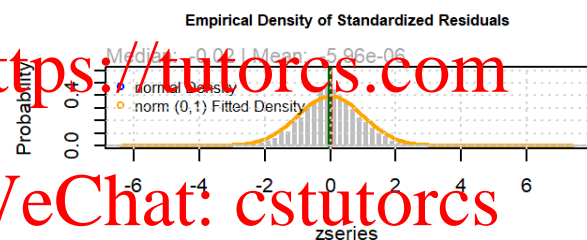
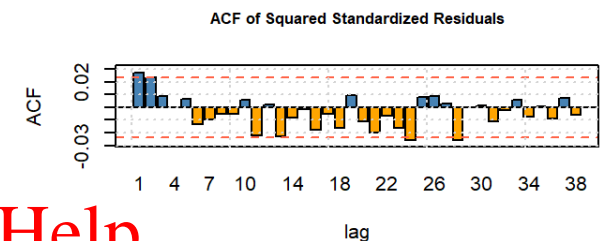
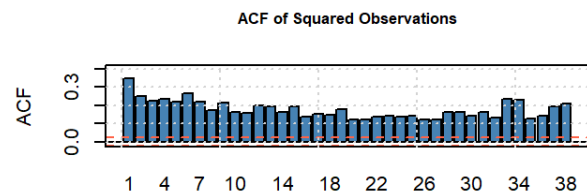
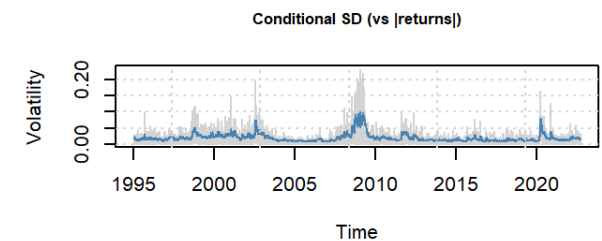
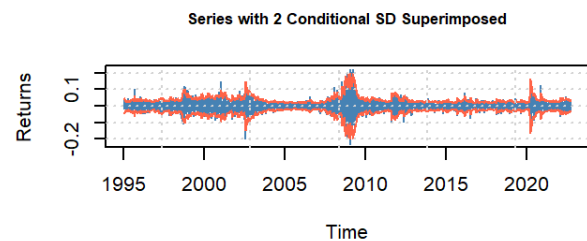
Assignment Project Exam Help

<https://tutores.com>

WeChat: estutores

```
par(mfrow=c(3,2))
plot(GARCH_1_1_fit, which = 1)
plot(GARCH_1_1_fit, which = 3)
plot(GARCH_1_1_fit, which = 5)
plot(GARCH_1_1_fit, which = 11)
plot(GARCH_1_1_fit, which = 8)
plot(GARCH_1_1_fit, which = 9)
```

Volatility Models - JPMorgan stock



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

ARCH(1) Model

Estimate ARCH(1) parameters and fit the conditional variance to returns

Volatility Models - JPMorgan stock

```
ARCH_1 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 0)), # garchOrder = c(q,p) where q = alpha (ARCH) and p = beta (GARCH)
                    mean.model = list(armaOrder = c(0, 0), include.mean = FALSE)) # armaOrder default is c(1, 1); but it is unstable with ARCH; hence c(0, 0) and include.mean = FALSE
ARCH_1_fit <- ugarchfit(spec = ARCH_1, data = log_returns_demean, solver = 'hybrid')
hARCH1 <- xts(x = ARCH_1_fit@fit$var, order.by = index(log_returns_demean))
sigmaARCH1 <- xts(x = ARCH_1_fit@fit$sigma, order.by = index(log_returns_demean))

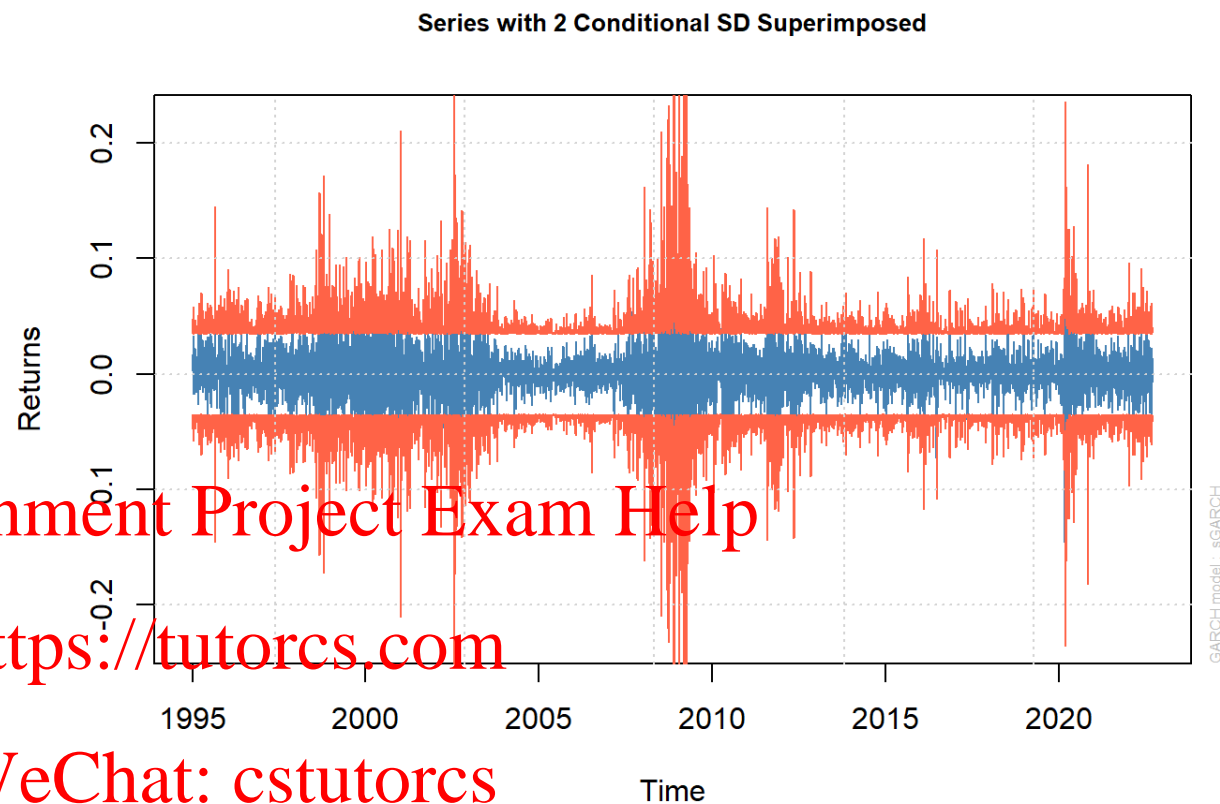
par(mfrow=c(1,1))
plot(ARCH_1_fit, which = 1)
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan
stock



LogLikelihoodRatio Test

ARCH(1) - Restricted model is nested in GARCH(1,1) - Unrestricted model

$$LR = 2 * (\text{LogU} - \text{LogR}) \sim \text{Chi-squared}(1)$$

```
LR = 2 * (likelihood(GARCH_1_1_fit) - likelihood(ARCH_1_fit))
LR
```

Volatility Models - JPMorgan stock

```
## [1] 2288.291
```

```
CV <- qchisq(p = 0.95, df = 1, lower.tail=TRUE)
CV
```

```
## [1] 3.841459
```

```
if (LR > CV) {
  print('H0: LR = 0 is rejected.')
} else {
  print('We cannot reject H0')
}
```

```
## [1] "H0: LR = 0 is rejected."
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Residual Analysis GARCH(1,1)

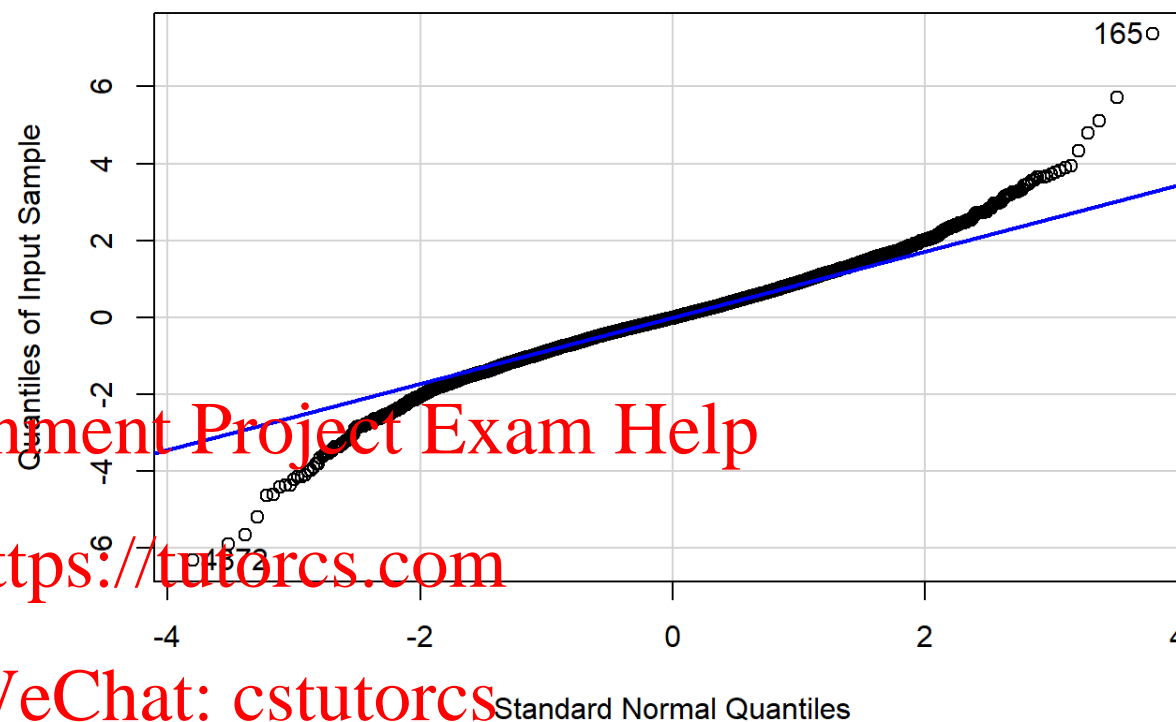
Check for Normality and Serial Correlation of the standardized residuals:

```
eps <- log_returns_demean / GARCH_1_1_fit@fit$sigma # Standardized
residuals
eps2 <- eps^2 # Standardized residuals squared

par(mfrow=c(1,1))
qqPlot(as.vector(eps), xlab = 'Standard Normal Quantiles',
       ylab = 'Quantiles of Input Sample', main = 'QQplot of sample
data vs Standard Normal',
       envelope = FALSE)
```

Volatility Models - JPMorgan
stock

QQplot of sample data vs Standard Normal



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

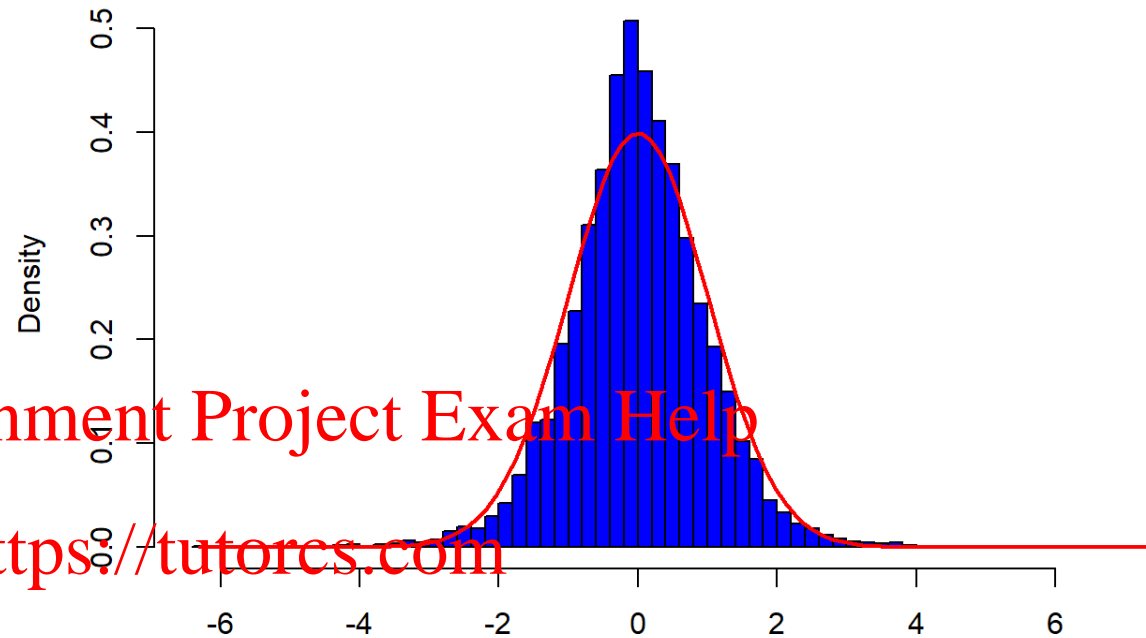
```
## [1] 165 4372
```

Histogram of residuals fitted with standard normal

```
par(mfrow=c(1,1))
seq_curve <- seq(min(eps), max(eps), length = 100)
normal_density <- dnorm(x = seq_curve, mean = mean(eps), sd = sd(eps))
hist(x = eps, prob = TRUE, breaks = 50, main = 'GARCH(1,1) Residuals', col = 'blue', xlab = '')
lines(seq_curve, normal_density, lwd = 2, col = 'red')
```

Volatility Models - JPMorgan
stock

GARCH(1,1) Residuals



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

JB Test

```
JB=jarque.bera.test(eps)
CV <- qchisq(p = 0.95, df = 2, lower.tail=TRUE)

if (JB$statistic > CV) {
  print('H0: LR = 0 is rejected.')
} else {
  print('We cannot reject H0')
}
```

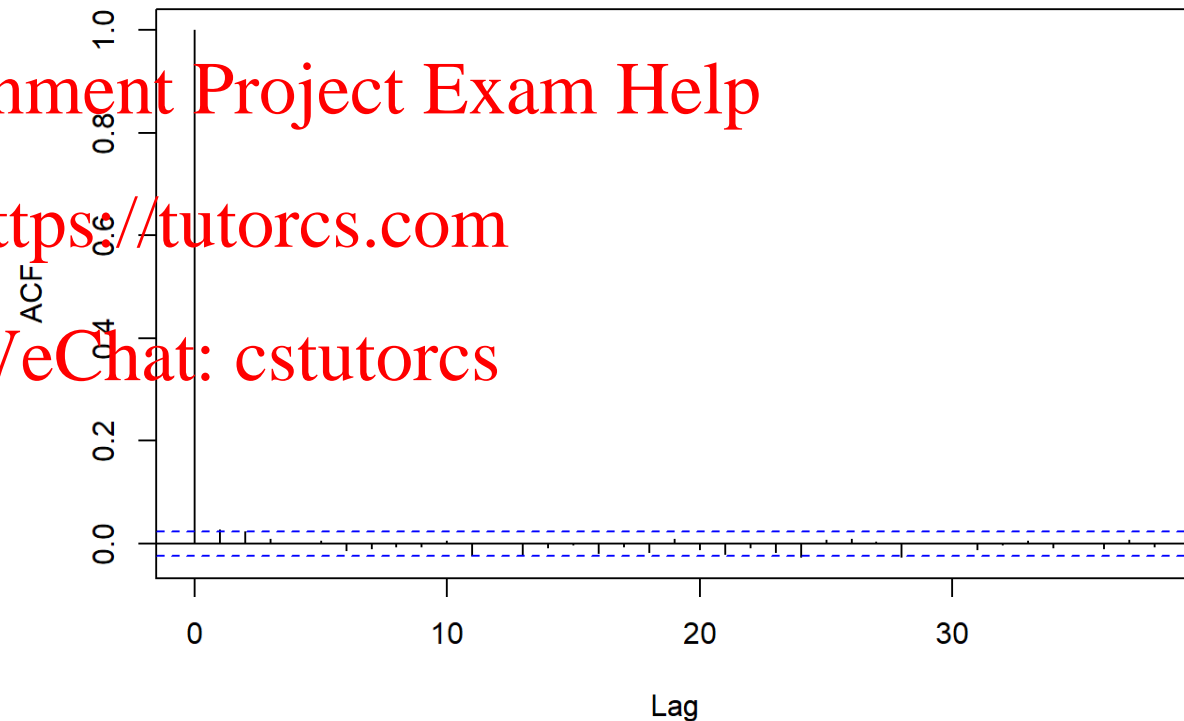

Volatility Models - JPMorgan stock

```
## [1] "H0: LR = 0 is rejected."
```

ACF's

```
acf(x = eps2, main = 'Sample Autocorrelation Function - Residuals squared')
```

Sample Autocorrelation Function - Residuals squared



LB Test

Volatility Models - JPMorgan stock

```
LB=Box.test(eps2, type = "Ljung-Box", lag = 1)
CV <- qchisq(p = 0.95, df = 1, lower.tail=TRUE)

if (LB$statistic > CV) {
  print('H0: LR = 0 is rejected.')
} else {
  print('We cannot reject H0')
}
```

```
## [1] "H0: LR = 0 is rejected."
```

Assignment Project Exam Help

Model Extensions Additional Code

ARCH(4)

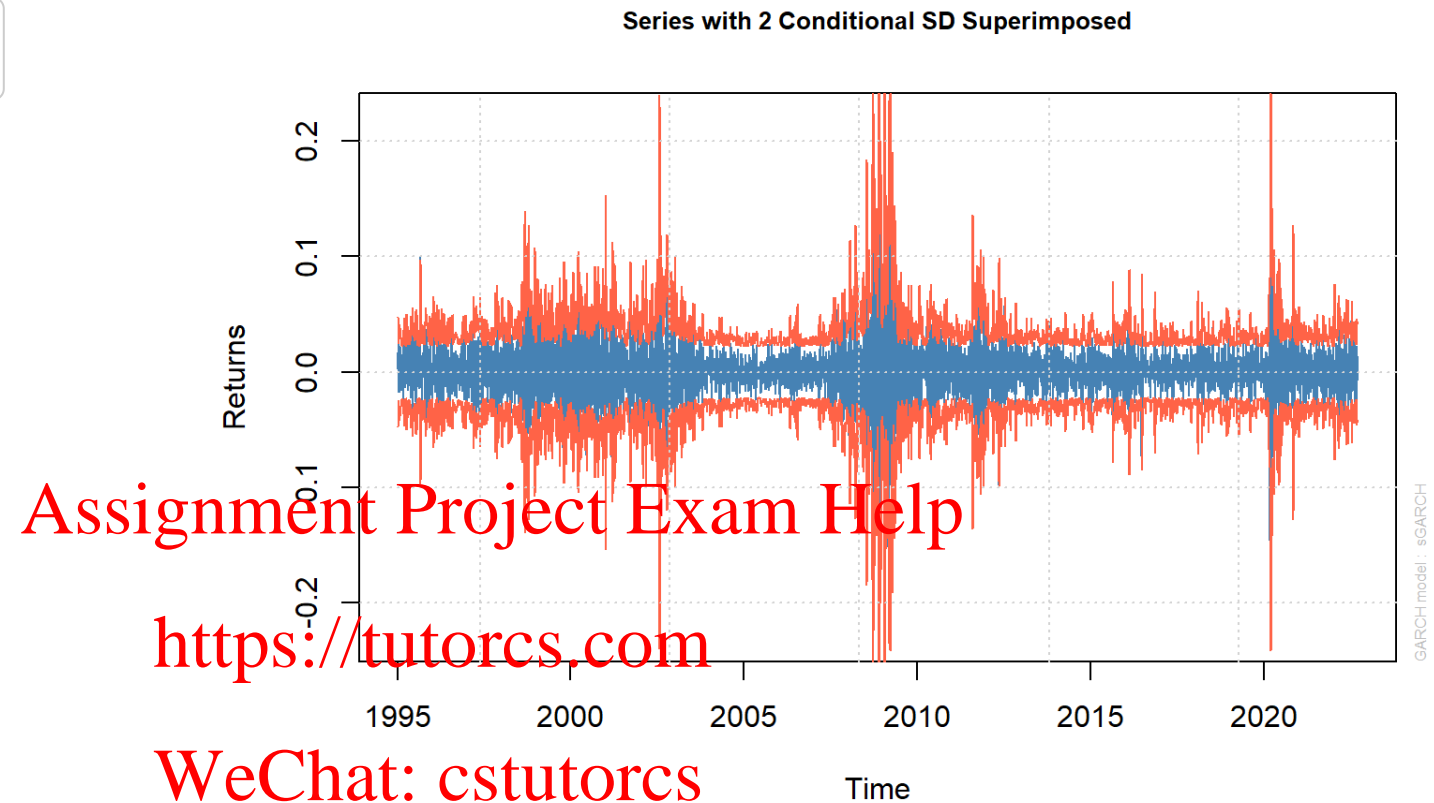
<https://tutorcs.com>

WeChat: cstutorcs

```
ARCH_4 <- ugarchspec(variance.model = list(model = "sGARCH", garchO
  rder = c(4, 0)), # garchOrder = c(q,p) where q = alpha (A
  RCH) and p = beta (GARCH)
  mean.model = list(armaOrder = c(0, 0), includ
  e.mean = FALSE))
ARCH_4_fit <- ugarchfit(spec = ARCH_4, data = log_returns_demean, s
  olver = 'hybrid')
hARCH_4 <- xts(x = ARCH_4_fit@fit$var, order.by = index(log_returns
  _demean))
sigmaARCH_4 <- xts(x = ARCH_4_fit@fit$sigma, order.by = index(log_r
  eturns_demean))

par(mfrow=c(1,1))
plot(ARCH_4_fit, which = 1)
```

Volatility Models - JPMorgan
stock



t-GARCH(1,1)

Estimate Garch(1,1) parameters and fit the conditional variance to returns. Assumes that innovations are drawn from a Student-t distribution.

Volatility Models - JPMorgan
stock

```
tGARCH_1_1 <- ugarchspec(variance.model = list(model = 'sGARCH', ga  
rchOrder = c(1,1)), distribution.model = 'std',  
mean.model = list(armaOrder = c(0, 0), inc  
lude.mean = FALSE))  
tGARCH_1_1_fit <- ugarchfit(spec = tGARCH_1_1, data = log_returns_d  
emean, solver = 'hybrid')  
  
tGARCH_1_1_fit
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan stock

```
##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(0,0,0)
## Distribution   : std
##
## Optimal Parameters
## -----
##              Estimate  Std. Error  t value  Pr(>|t|)
## omega         0.000002    0.000001    1.6888  0.091253
## alpha1         0.079110    0.012680    6.2391  0.000000
## beta1          0.919127    0.012668   72.5554  0.000000
## shape          5.944203    0.374325   15.8798  0.000000
##
## Robust Standard Errors:
##              Estimate  Std. Error  t value  Pr(>|t|)
## omega         0.000002    0.000006    0.43854  0.66099
## alpha1         0.079110    0.051211    1.54480  0.12240
## beta1          0.919127    0.050369   18.24783  0.00000
## shape          5.944203    0.746023    7.96786  0.00000
##
## LogLikelihood : 18376.88
##
## Information Criteria
## -----
##
## Akaike          -5.2667
## Bayes           -5.2628
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan
stock

```
## Shibata      -5.2667
## Hannan-Quinn -5.2653
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##
##                      statistic p-value
## Lag[1]                0.000203  0.9886
## Lag[2*(p+q)+(p+q)-1][2] 0.310814  0.7897
## Lag[4*(p+q)+(p+q)-1][5] 1.351332  0.7764
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##
##                      statistic p-value
## Lag[1]                4.389 0.03617
## Lag[2*(p+q)+(p+q)-1][5] 7.413 0.04113
## Lag[4*(p+q)+(p+q)-1][9] 8.902 0.08514
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##
##          Statistic Shape Scale P-Value
## ARCH Lag[3]    0.3524 0.500 2.000  0.5528
## ARCH Lag[5]    0.4630 1.440 1.667  0.8945
## ARCH Lag[7]    1.7158 2.315 1.543  0.7772
##
## Nyblom stability test
## -----
## Joint Statistic:  51.7181
## Individual Statistics:
## omega  10.5539
## alpha1  0.7017
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan stock

```
## beta1    0.8375
## shape    1.0502
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.07 1.24 1.6
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## -----
##                               t-value      prob sig
## Sign Bias                   0.503 0.6150142
## Negative Sign Bias         3.594 0.0003283 ***
## Positive Sign Bias         1.149 0.2507064
## Joint Effect               18.614 0.0002815 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##   group statistic p-value(g-1)
## 1    20      22.06      0.2814
## 2     30      36.44      0.1610
## 3     40      41.20      0.3747
## 4     50      59.96      0.1357
##
##
## Elapsed time : 0.3823621
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

When calling tGARCH, we will see that there is a new parameter called shape, which is the estimation of the degrees of freedom of the distribution:

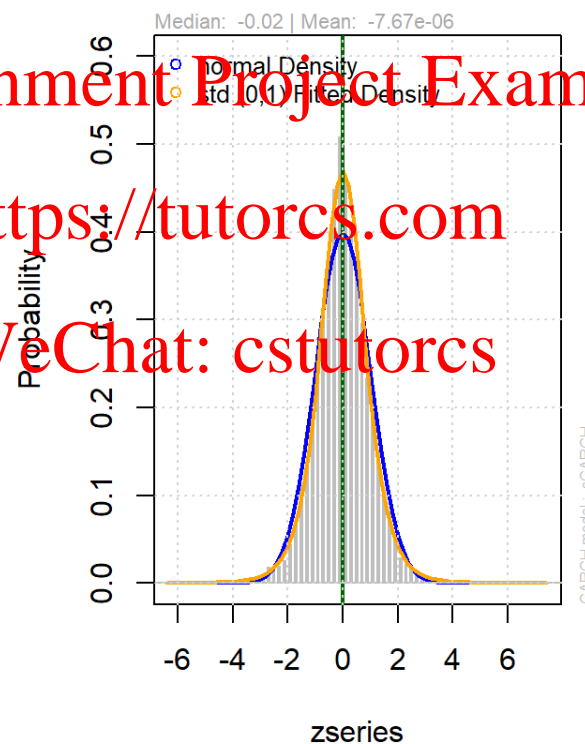
```
coef(tGARCH_1_1_fit)
```

Volatility Models - JPMorgan stock

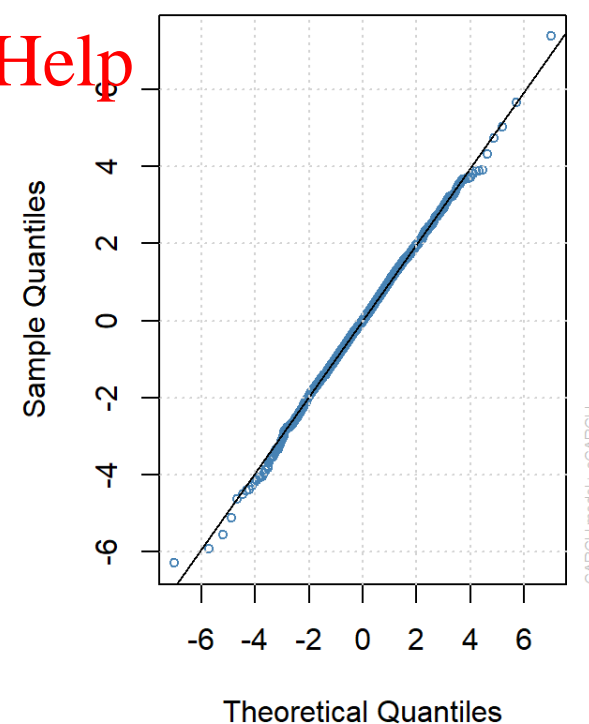
##	omega	alpha1	beta1	shape
##	2.439861e-06	7.911032e-02	9.191270e-01	5.944203e+00

```
par(mfrow=c(1,2))
plot(tGARCH_1_1_fit, which = 8)
plot(tGARCH_1_1_fit, which = 9)
```

Empirical Density of Standardized Residuals



std - QQ Plot



LogLikelihood Ratio Test

GARCH(1,1) - Restricted model is nested in t-GARCH(1,1) - Unrestricted model

Volatility Models - JPMorgan stock

$$LR = 2 * (\text{LogU} - \text{LogR}) \sim \text{Chi-squared}(1)$$

```
LR = 2 * (likelihood(tGARCH_1_1_fit) - likelihood(GARCH_1_1_fit))

CV <- qchisq(p = 0.95, df = 1, lower.tail=TRUE)

if (LR > CV) {
  print('H0: LR = 0 is rejected.')
} else {
  print('We cannot reject H0')
}
```

Assignment Project Exam Help

```
## [1] "H0: LR = 0 is rejected."
```

GJR - GARCH(1,1)

<https://tutorcs.com>

WeChat: cstutorcs

```
GJR_GARCH_1_1 <- ugarchspec(variance.model = list(model = 'gjrGARCH',
  H', garchOrder = c(1,1)),
  mean.model = list(armaOrder = c(0, 0),
    include.mean = FALSE))
GJR_GARCH_1_1_fit <- ugarchfit(spec = GJR_GARCH_1_1, data = log_returns_demean, solver = 'hybrid')

coef(GJR_GARCH_1_1_fit)
```

```
##          omega          alpha1          beta1          gamma1
## 3.365275e-06 2.598761e-02 9.222511e-01 9.340704e-02
```

Volatility Models - JPMorgan
stock

Comparing volatility models visually

```
plot(x = index(sigmaEWMA), y = log_returns_demean, type = 'l', main = paste('Volatility forecast JPM'), xlab = 'Trading Days', ylab = 'Returns', col = 'grey')  
lines(x = index(sigmaEWMA), y = 2 * sigmaEWMA[, 1], col = 'red')  
lines(x = index(sigmaEWMA), y = -2 * sigmaEWMA[, 1], col = 'red')  
lines(x = index(sigmaEWMA), y = 2 * sigmaGARCH_1_1, col = 'blue')  
lines(x = index(sigmaEWMA), y = -2 * sigmaGARCH_1_1, col = 'blue')  
legend('topleft', legend = c('EWMA', 'GARCH'), col = c('red', 'blue'), lty=1)
```

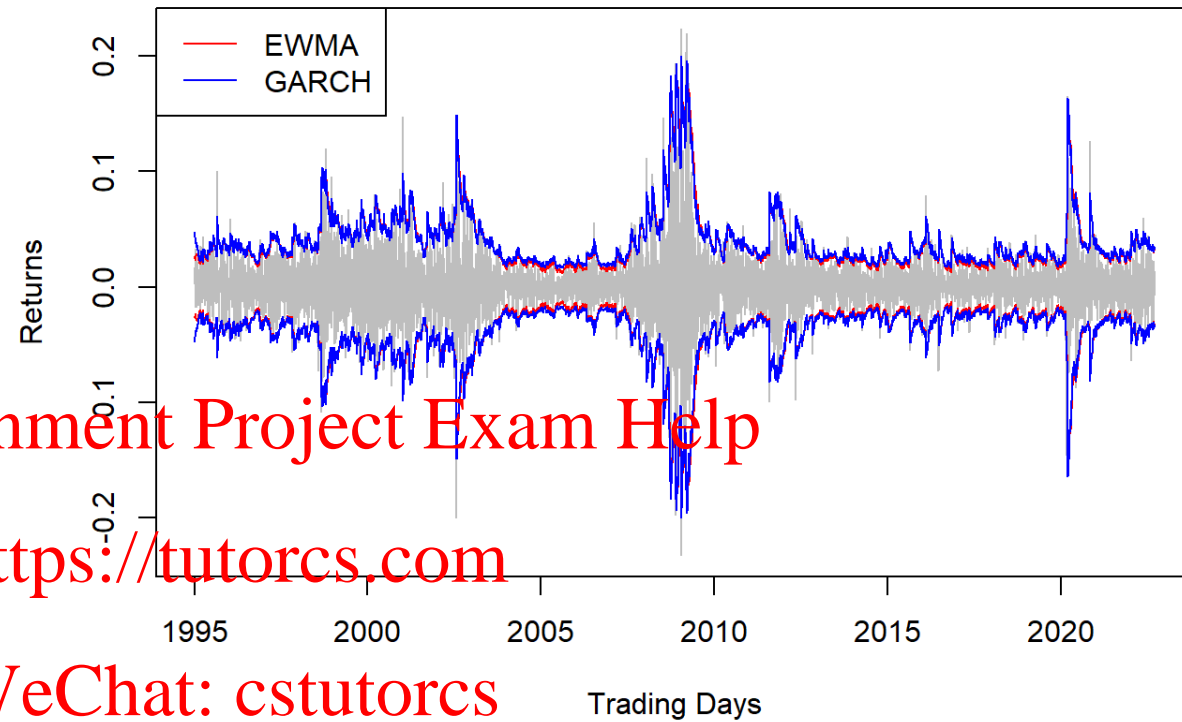
Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Volatility Models - JPMorgan
stock

Volatility forecast JPM



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs