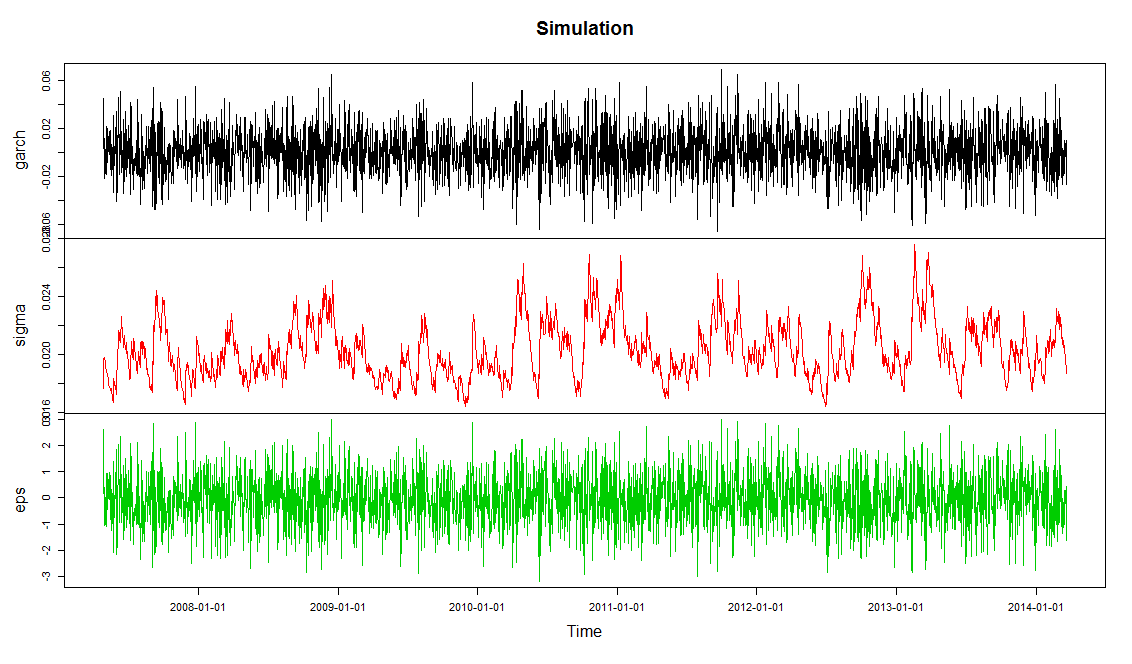
**3(c)**

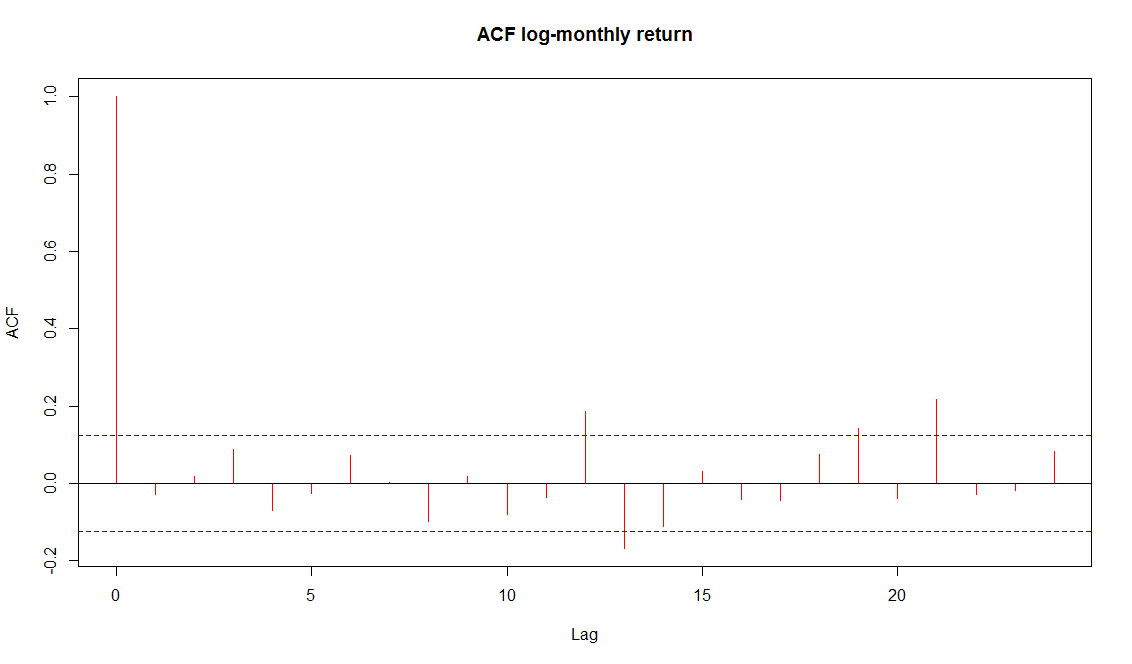


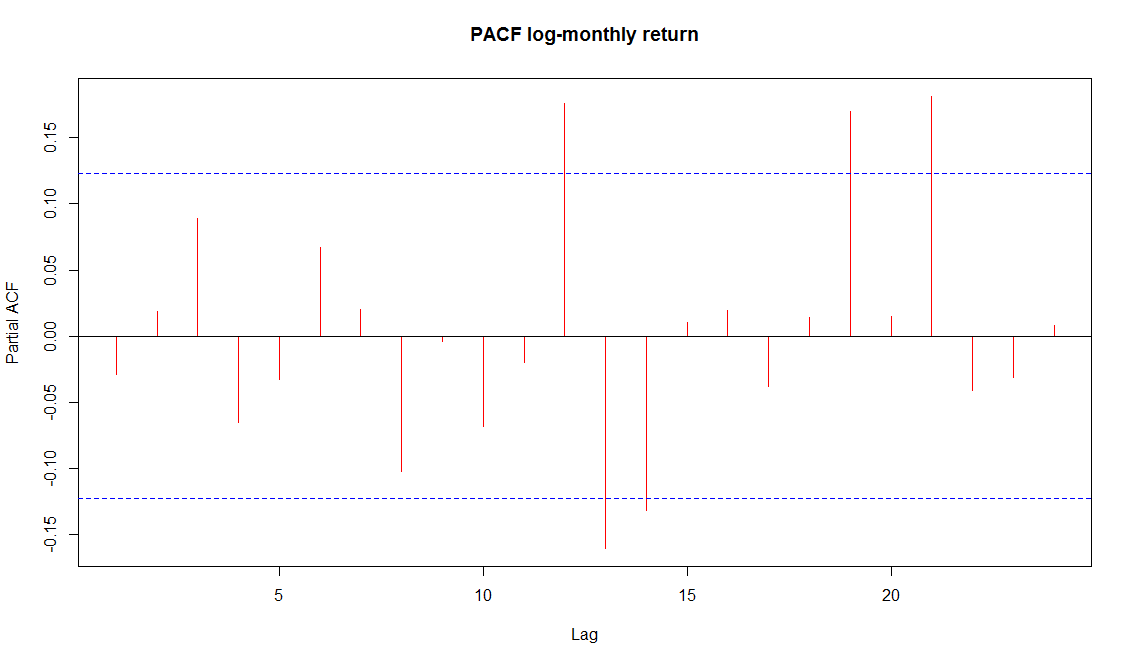
**R-Code:**

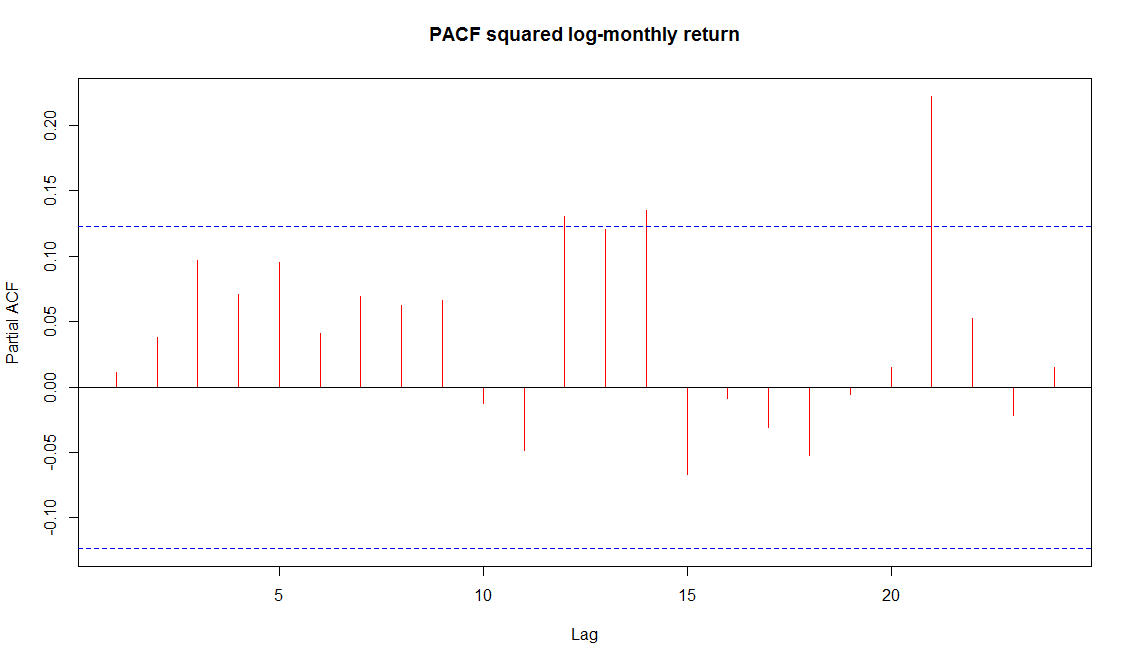
simulate<-garchSim(spec=garchSpec(model=list(mu=0.002,ar=-0.12,omega=0.000015,alpha=0.0414,beta=0.921),cond.dist="norm"),n=2520,extended=TRUE)

plot(simulate,main="Simulation")

**4(a)~(b)**







**R-code:**

setwd("C:/Courses/ORF504Financial Econometrics/HW3")

Temp=read.csv("Intel.csv",header=TRUE)

cp1<-Temp[,7]

cp<-rep(0,length(cp1))

for(i in 1:length(cp1))

{ cp[i]=cp1[length(cp1)-i+1]}

return<-rep(0,length(cp)-1)

for(i in 1:length(cp)-1)

{ return[i]=log(cp[i+1])-log(cp[i])}

acf(return,main="ACF log-monthly return",col=2)

pacf(return,main="PACF log-monthly return",col=2)

sreturn<-return^2

pacf(sreturn,main="PACF squared log-monthly return",col=2)

**4(c)**

Coefficient(s):

mu ar1 ar2 ar3 omega alpha1 beta1

0.00839900 0.00657177 0.01583878 0.04220256 0.00040842 0.04924301 0.91885634

**R-code:**

setwd("C:/Courses/ORF504Financial Econometrics/HW3")

Temp=read.csv("Intel.csv",header=TRUE)

cp1<-Temp[,7]

cp<-rep(0,length(cp1))

for(i in 1:length(cp1))

{

cp[i]=cp1[length(cp1)-i+1]

}

return<-rep(0,length(cp)-1)

for(i in 1:length(cp)-1)

{

return[i]=log(cp[i+1])-log(cp[i])

}

fitmodel<-garchFit(formula=~arma(3,0)+garch(1,1),data=return,cond.dist="norm",trace=FALSE)

fitmodel

**4(d)~(f)**

> alpha/A

[1] 0.00877332

> c/B

[1] 0.01270714

> t(f)%\*%S1%\*%f

[,1]

[1,] 5.501387e-05

> t(g)%\*%S2%\*%g

[,1]

[1,] 1.006961e-05

> sqrt(t(f)%\*%S1%\*%f)

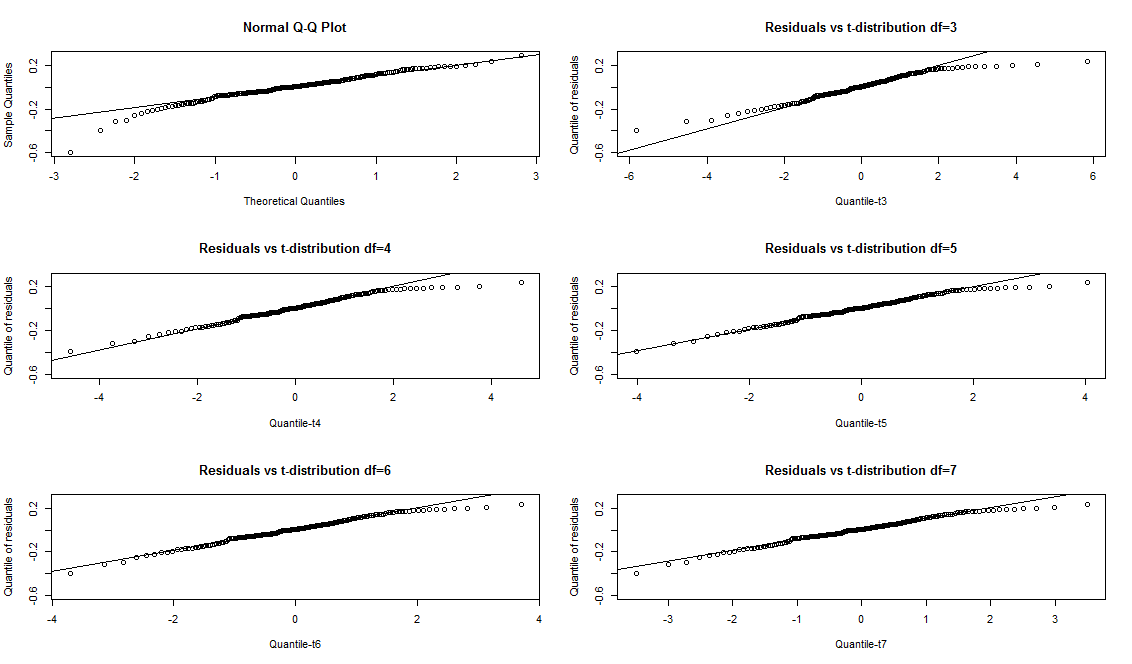
[,1]

[1,] 0.007417134

> sqrt(t(g)%\*%S2%\*%g)

[,1]

[1,] 0.003173265

****

**R-code:**

setwd("C:/Courses/ORF504Financial Econometrics/HW3")

Temp=read.csv("Intel.csv",header=TRUE)

cp1<-Temp[,7]

cp<-rep(0,length(cp1))

for(i in 1:length(cp1))

{

cp[i]=cp1[length(cp1)-i+1]

}

return<-rep(0,length(cp)-1)

for(i in 1:length(cp)-1)

{

return[i]=log(cp[i+1])-log(cp[i])

}

fitmodel<-garchFit(formula=~arma(3,0)+garch(1,1),data=return,cond.dist="norm",trace=FALSE)

S<-fitmodel@fit$cvar

alpha=0.00820701

beta1=0.00692554

beta2=0.01586366

beta3=0.04175989

c=0.00039117

b=0.05012590

a=0.91909062

A=1-beta1-beta2-beta3

f<-c(A,alpha,alpha,alpha)/A^2

S1<-S[1:4,1:4]

B=1-b-a

g<-c(B,c,c)/B^2

S2<-S[5:7,5:7]

alpha/A

c/B

t(f)%\*%S1%\*%f

t(g)%\*%S2%\*%g

sqrt(t(f)%\*%S1%\*%f)

sqrt(t(g)%\*%S2%\*%g)

#######(f)#########

res<-fitmodel@residuals

acf(res, main="ACF for residuals")

Box.test(res,lag=5,type="Ljung-Box")

Box.test(res,lag=10,type="Ljung-Box")

acf(res^2,main="ACF for squared residuals")

Box.test(res^2,lag=5,type="Ljung-Box")

Box.test(res^2,lag=10,type="Ljung-Box")

y<-quantile(res,prob=seq(0,1,0.005))

par(mfrow=c(3,2))

qqnorm(y)

qqline(y)

tdis3<-qt(seq(0,1,0.005),3)

tdis4<-qt(seq(0,1,0.005),4)

tdis5<-qt(seq(0,1,0.005),5)

tdis6<-qt(seq(0,1,0.005),6)

tdis7<-qt(seq(0,1,0.005),7)

qqplot(tdis3,y,xlab="Quantile-t3",ylab="Quantile of residuals", main="Residuals vs t-distribution df=3")

qqline(y)

qqplot(tdis4,y,xlab="Quantile-t4",ylab="Quantile of residuals", main="Residuals vs t-distribution df=4")

qqline(y)

qqplot(tdis5,y,xlab="Quantile-t5",ylab="Quantile of residuals", main="Residuals vs t-distribution df=5")

qqline(y)

qqplot(tdis6,y,xlab="Quantile-t6",ylab="Quantile of residuals", main="Residuals vs t-distribution df=6")

qqline(y)

qqplot(tdis7,y,xlab="Quantile-t7",ylab="Quantile of residuals", main="Residuals vs t-distribution df=7")

qqline(y)

**Results and R-code for Problem5**

**(a)**

Error Analysis:

Estimate Std. Error t value Pr(>|t|)

mu 1.967e-03 2.218e-03 0.887 0.3752

ar1 -7.791e-01 1.674e-01 -4.654 3.26e-06 \*\*\*

ma1 7.463e-01 1.765e-01 4.228 2.36e-05 \*\*\*

omega 7.289e-05 2.855e-05 2.554 0.0107 \*

alpha1 8.126e-02 1.895e-02 4.289 1.80e-05 \*\*\*

beta1 8.943e-01 2.234e-02 40.035 < 2e-16 \*\*\*

shape 5.874e+00 1.050e+00 5.592 2.24e-08 \*\*\*

**(b)**

Error Analysis:

Estimate Std. Error t value Pr(>|t|)

mu 3.086e-03 2.188e-03 1.411 0.1584

ar1 -6.459e-01 2.818e-01 -2.292 0.0219 \*

ma1 5.970e-01 2.993e-01 1.995 0.0461 \*

omega 5.391e-05 2.243e-05 2.404 0.0162 \*

alpha1 1.012e-01 1.799e-02 5.627 1.83e-08 \*\*\*

beta1 8.864e-01 1.995e-02 44.419 < 2e-16 \*\*\*

**(c)**

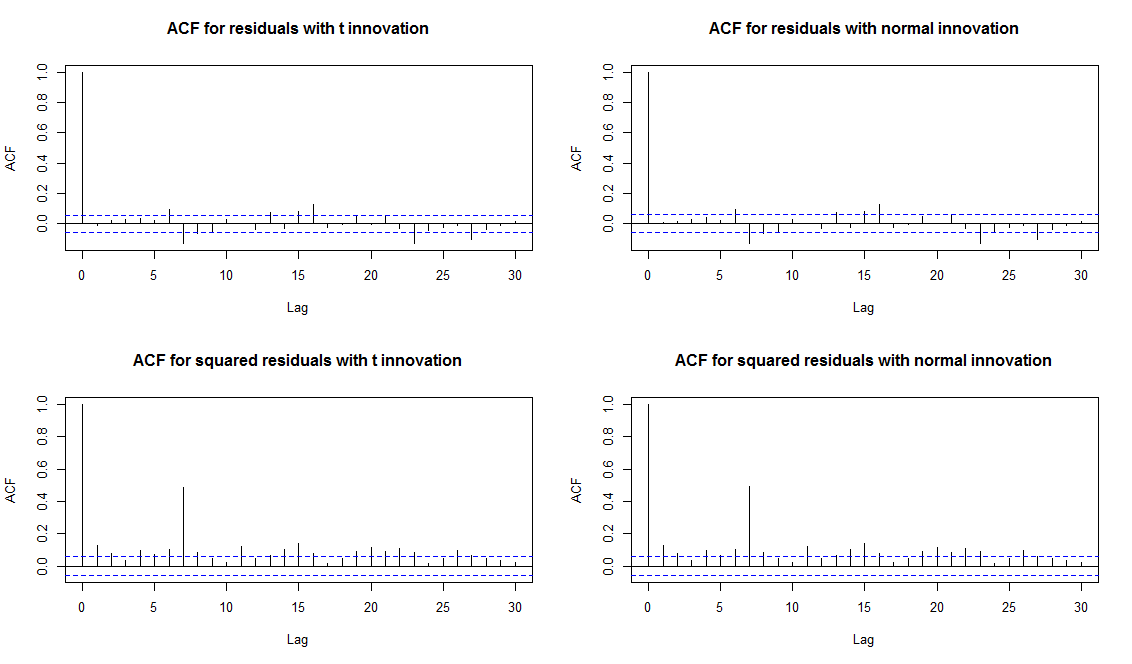
> mean(return)

[1] 0.001170509

> var(return)

[1] 0.003716753

**(d)**



**R code**

setwd("C:/Courses/ORF504Financial Econometrics/HW3")

Temp=read.csv("Ford.csv",header=TRUE)

cp1<-Temp[,7]

cp<-rep(0,length(cp1))

for(i in 1:length(cp1))

{

cp[i]=cp1[length(cp1)-i+1]

}

return<-rep(0,length(cp)-1)

for(i in 1:length(cp)-1)

{

return[i]=log(cp[i+1])-log(cp[i])

}

##################5(a)#######################

fit<-garchFit(formula=~arma(1,1)+garch(1,1),data=return,cond.dist="std",algorithm="nlminb",include.mean=TRUE)

summary(fit)

##################5(b)#######################

fit2<-garchFit(formula=~arma(1,1)+garch(1,1),data=return,cond.dist="norm",algorithm="nlminb",include.mean=TRUE)

summary(fit2)

##################5(c)#######################

mean(return)

var(return)

##################5(d)#######################

par(mfrow=c(2,2))

rt<-fit@residuals

rn<-fit2@residuals

acf(rt,main="ACF for residuals with t innovation")

acf(rn,main="ACF for residuals with normal innovation")

acf(rt^2,main="ACF for squared residuals with t innovation")

acf(rn^2,main="ACF for squared residuals with normal innovation")

summary(fit)

summary(fit2)