Forecasting Homework 8

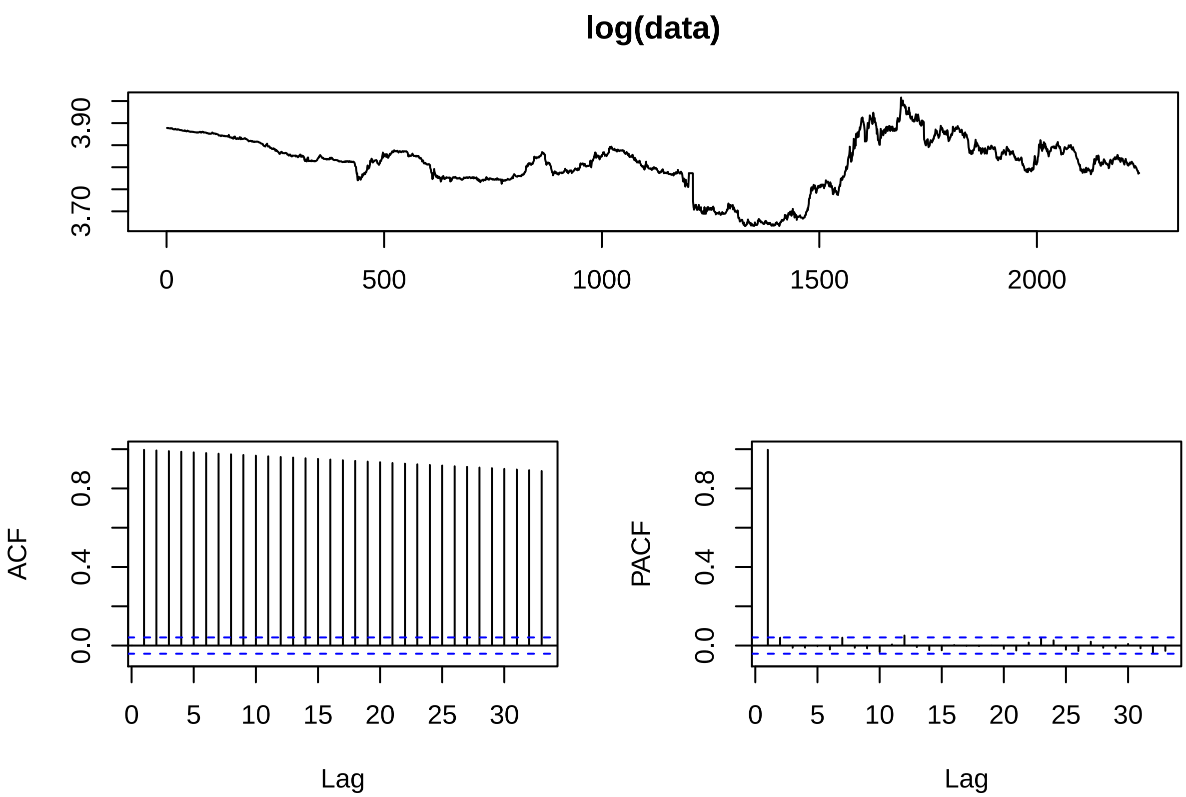
Yuhao Zhao, Yz3085

December 8, 2015

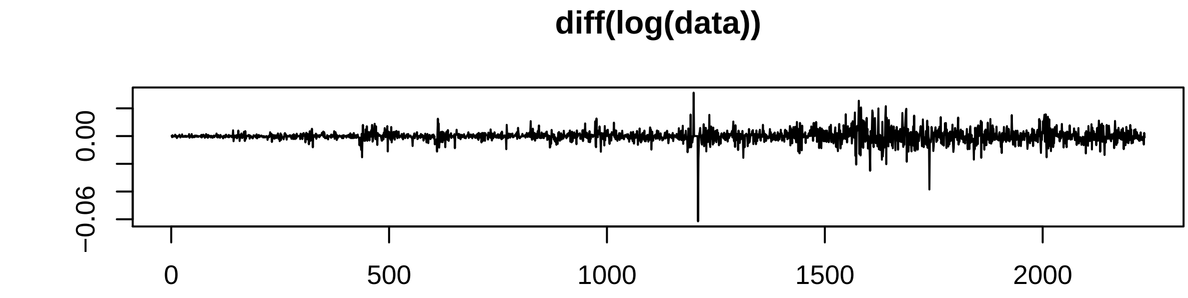
```r
library('forecast')
library('itsmr')
library('tseries')
data = read.csv("http://people.stern.nyu.edu/churvich/Forecasting/Data/Rupee.CSV")[,2]
```

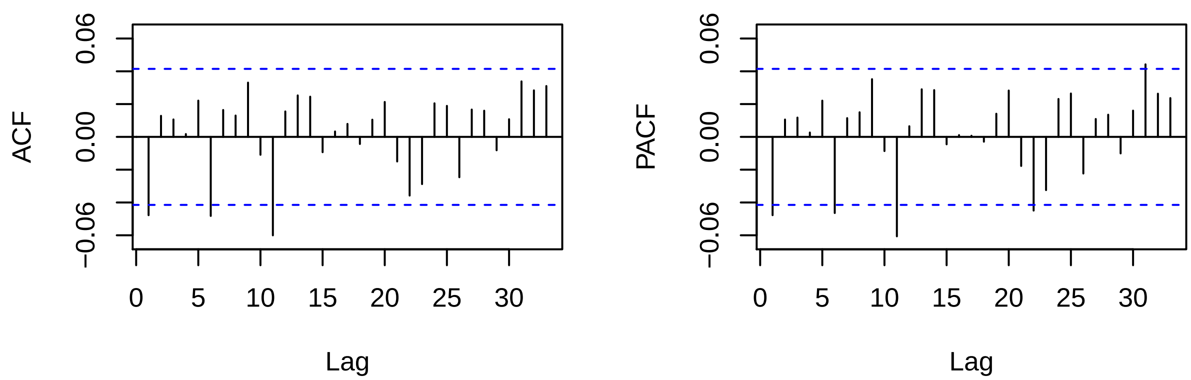
1)

```r
tsdisplay(log(data),points = F)
```




```r
tsdisplay(diff(log(data)),points = F)
```





The plot of logs of Rupee shows that it's non-stationary, since it is not clearly mean-reverting. The differenced logs of Rupee shows strong stationary behavior. From the ACF and PACF of differenced logs of Rupee an MA(1) model is may be adequant. Therefore, a possible model to logs of Rupee can be ARIMA(0,1,1).

2)

```
model= auto.arima(data,d = 1,parallel = F,trace = T,ic = "aicc",stepwise = T
                  ,allowmean = TRUE,allowdrift = F,seasonal =F,max.p = 2,max.q=2 )
```

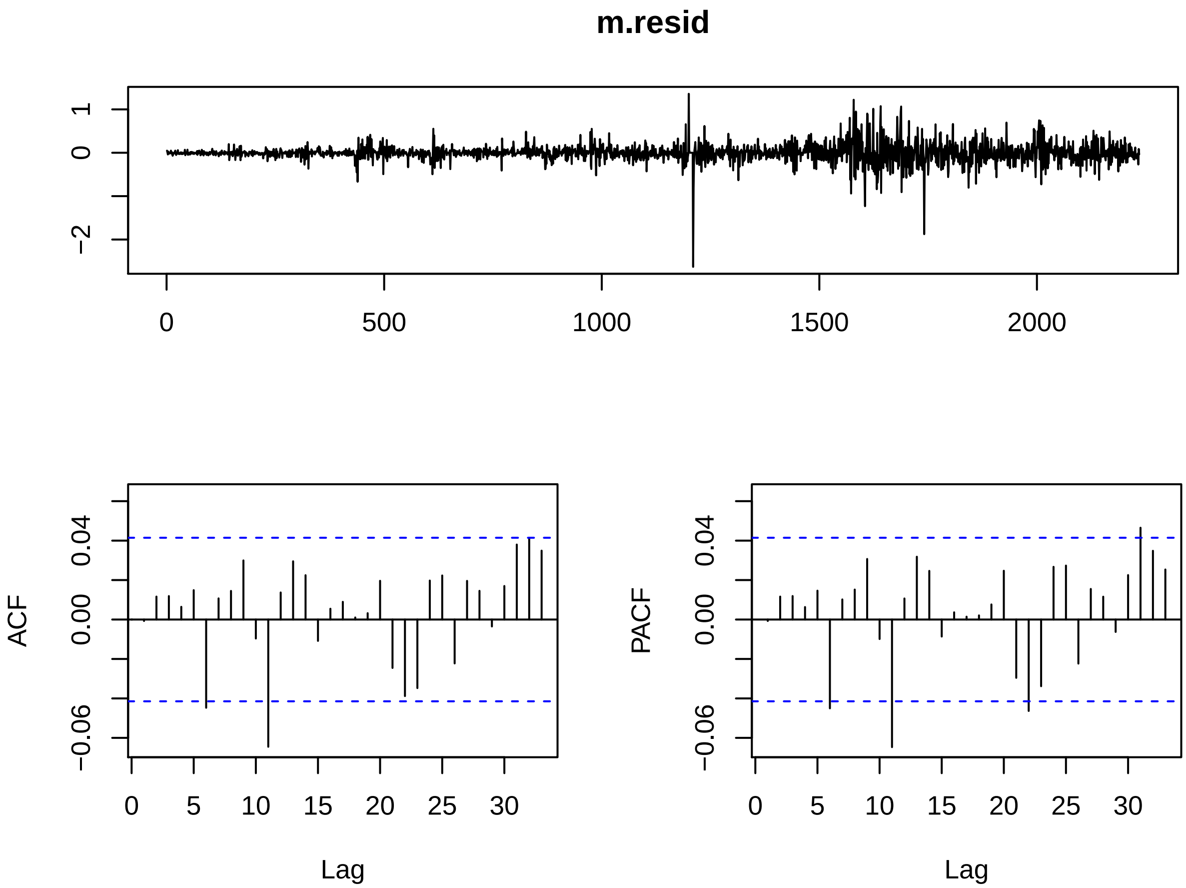
```
##
##  ARIMA(2,1,2)                    : -663.257
##  ARIMA(0,1,0)                    : -667.2863
##  ARIMA(1,1,0)                    : -669.9146
##  ARIMA(0,1,1)                    : -670.7962
##  ARIMA(1,1,1)                    : -668.0208
##  ARIMA(0,1,2)                    : -669.1281
##  ARIMA(1,1,2)                    : -666.1072
##
##  Best model: ARIMA(0,1,1)
```

```
m.resid = model$residuals
m.fore = forecast::forecast(model,h=1,level = 95)
m.fit = fitted(model)
```

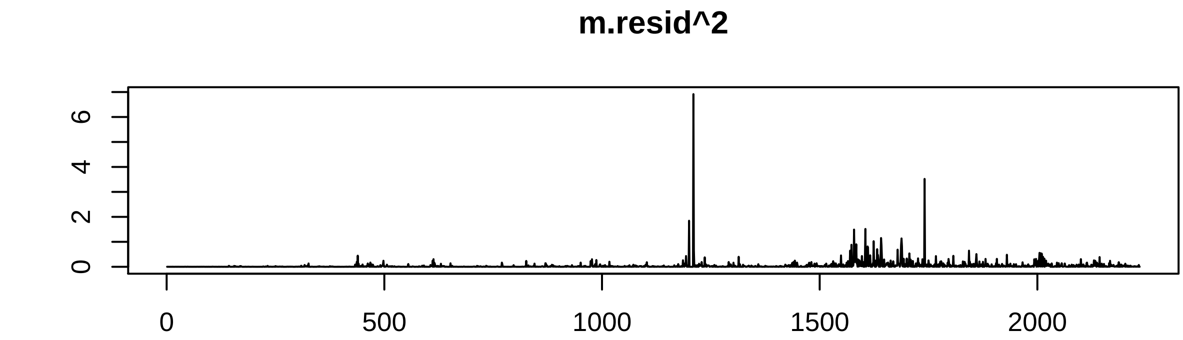
The model we fit with lowest aicc is ARIMA(0,1,1)

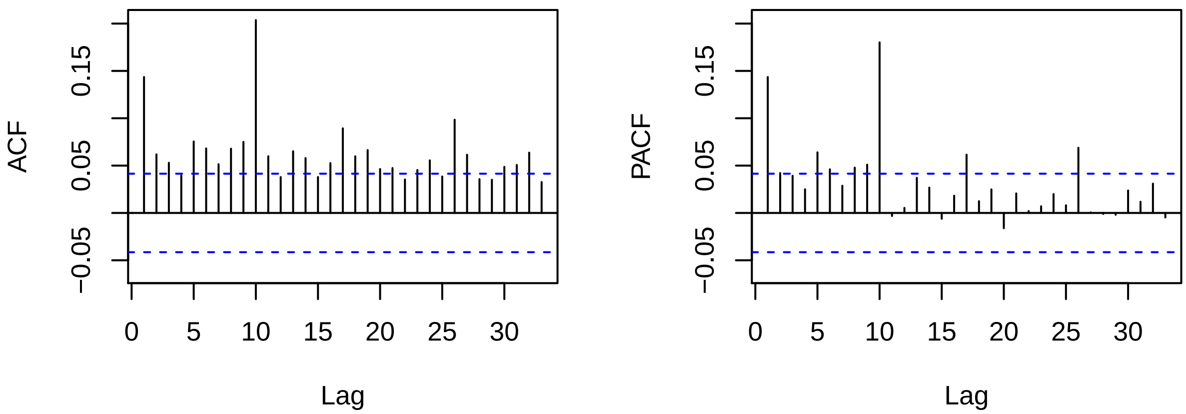
3)

```
tsdisplay(m.resid,points = F)
```




```
tsdisplay(m.resid^2,points = F)
```





The ACF and PACF plot of the residuals show that there is some evidence of autocorralation with some certain lags. Therefore the residuals are not uncorrelated thus not independent.

4)

```
AICc_arch <- function(data,order)
{
  N = length(data)-1
  res = c()
  for (i in order[1]:order[2])
  {
    if (i == 0) {
        loglik = -0.5*N*(1 + log(2*pi*mean(data^2)))
        aicc = -2*loglik+2*(i +1)*N/(N-i-2)
        res = rbind(res,c(i,loglik,aicc) )}
    else{
      loglik= logLik(garch(data,c(0,i),control = garch.control(trace = F)))
      aicc = -2*loglik+2*(i +1)*N/(N-i-2)
      res = rbind(res,c(i,loglik,aicc ))
    }
  }
  colnames(res) = c('order q','log lik','AICc')
  res
```

```r
}
res = AICc_arch(m.resid,c(0,10));res
```

```
## Warning in sqrt(pred$e): NaNs produced
```

```
##       order q   log lik       AICc
##  [1,]       0  339.4450  -676.8883
##  [2,]       1  488.1378  -972.2703
##  [3,]       2  540.8305 -1075.6502
##  [4,]       3  606.5893 -1205.1607
##  [5,]       4  622.0544 -1234.0819
##  [6,]       5  636.4470 -1260.8562
##  [7,]       6  648.4877 -1282.9250
##  [8,]       7  666.9233 -1317.7819
##  [9,]       8  669.4783 -1320.8756
## [10,]       9  669.3576 -1318.6162
## [11,]      10 1062.7164 -2103.3140
```

```r
m.garch = garch(m.resid,c(1,1),control =garch.control(trace = F))
```

```
## Warning in sqrt(pred$e): NaNs produced
```

```r
N = length(m.resid)-1
m.garch.loglik = logLik(m.garch)
m.garch.aicc = -2*m.garch.loglik+2*(2 +1)*N/(N-2-2);m.garch.aicc[1]
```

```
## [1] -2147.886
```

```r
summary(m.garch);m.garch.loglik
```

```
##
## Call:
## garch(x = m.resid, order = c(1, 1), control = garch.control(trace = F))
##
## Model:
## GARCH(1,1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2593 -0.5678 -0.0177  0.4713  5.6857
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 1.334e-04   1.544e-05    8.637   <2e-16 ***
## a1 1.326e-01   7.737e-03   17.144   <2e-16 ***
## b1 8.843e-01   5.096e-03  173.511   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
```

```
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 3604.5, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.020074, df = 1, p-value = 0.8873
```

```
## 'log Lik.' 1076.949 (df=3)
```

From the outout we see that the lowest aicc provided by arch model is -2103.3140, and the garch(1,1) model reterns a -2147.886 aicc. Therefore we use garch(1,1) to model the residuals. The p-values for the parameter values of garch(1,1) are all <2e-16. Therefore they are all significant. The complete form of the model is

$$\epsilon_t|\psi_{t-1} \sim N(0, h_t), h_t = 1.334 \times 10^{-4} + 1.326 \times 10^{-1}\epsilon_{t-1}^2 + 8.843 \times 10^{-1}h_{t-1}$$

5)

```
cond_var = tail(m.garch$fit[,1],1)
h_f1 = m.garch$coef[1]+m.garch$coef[2]* tail(m.resid,1)^2 + m.garch$coef[3]* tail(m.garch$fit[,1]^2,1);
m.garch.fore= m.fore$mean[1] + c(0,-qnorm(0.975)*sqrt(h_f1),qnorm(0.975)*sqrt(h_f1));
colnames(m.garch.fore) = colnames(m.fore)
m.fore;m.garch.fore
```

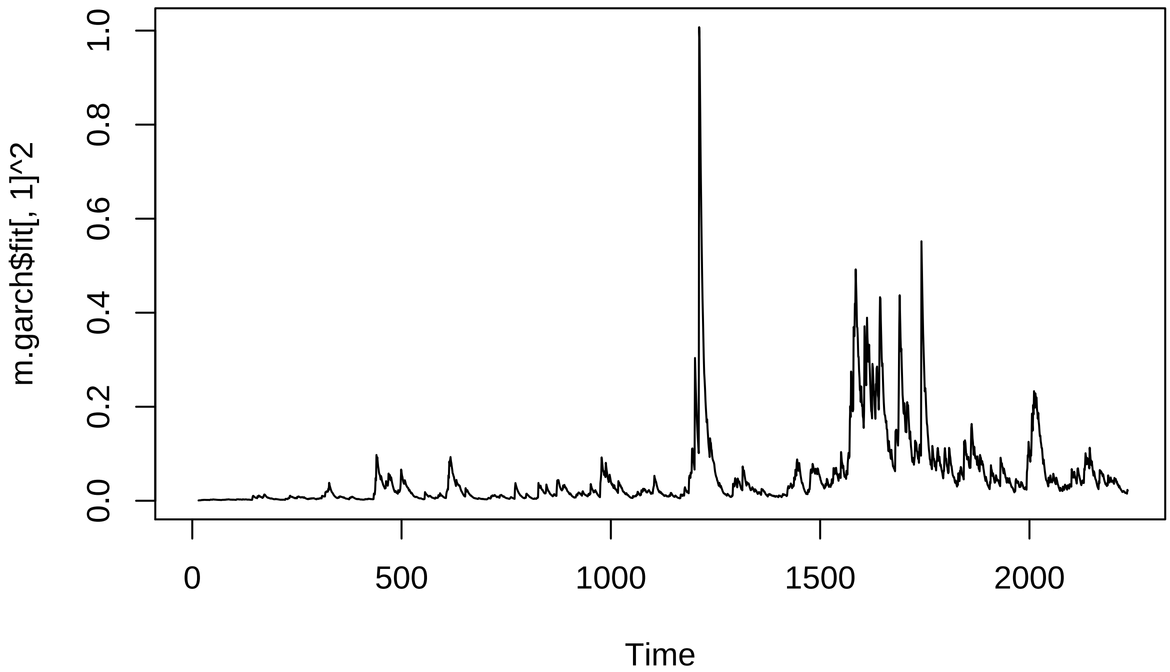
```
##      Point Forecast    Lo 95    Hi 95
## 2236       44.11175 43.70427 44.51924
```

```
##                a0       a0
## 44.11175 43.83846 44.38505
```

The forecast interval for ARIMA-GARCH is 44.07765, 44.14586 which is narrower than that for the ARIMA model.

6)

```
plot(m.garch$fit[,1]^2)
```

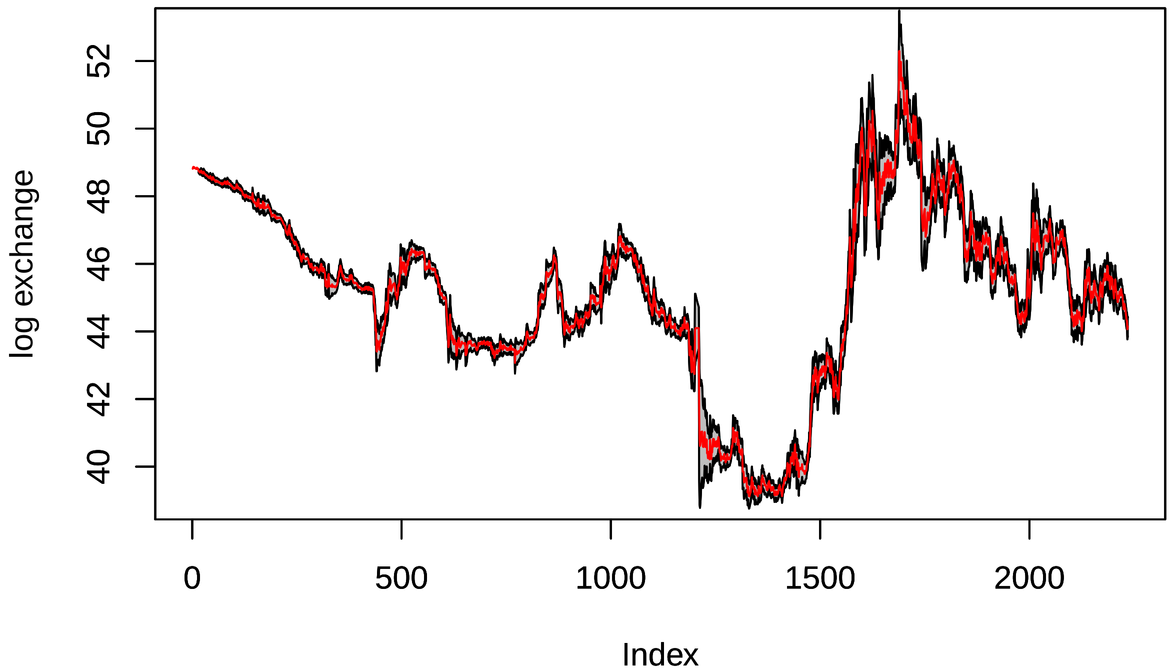



```
ht = m.garch$fit[,1]^2
```

The plot shows high volatility during time index 1100 to 1300, and index 1500 to 1800. This is agree with the log exchange itself.

7)

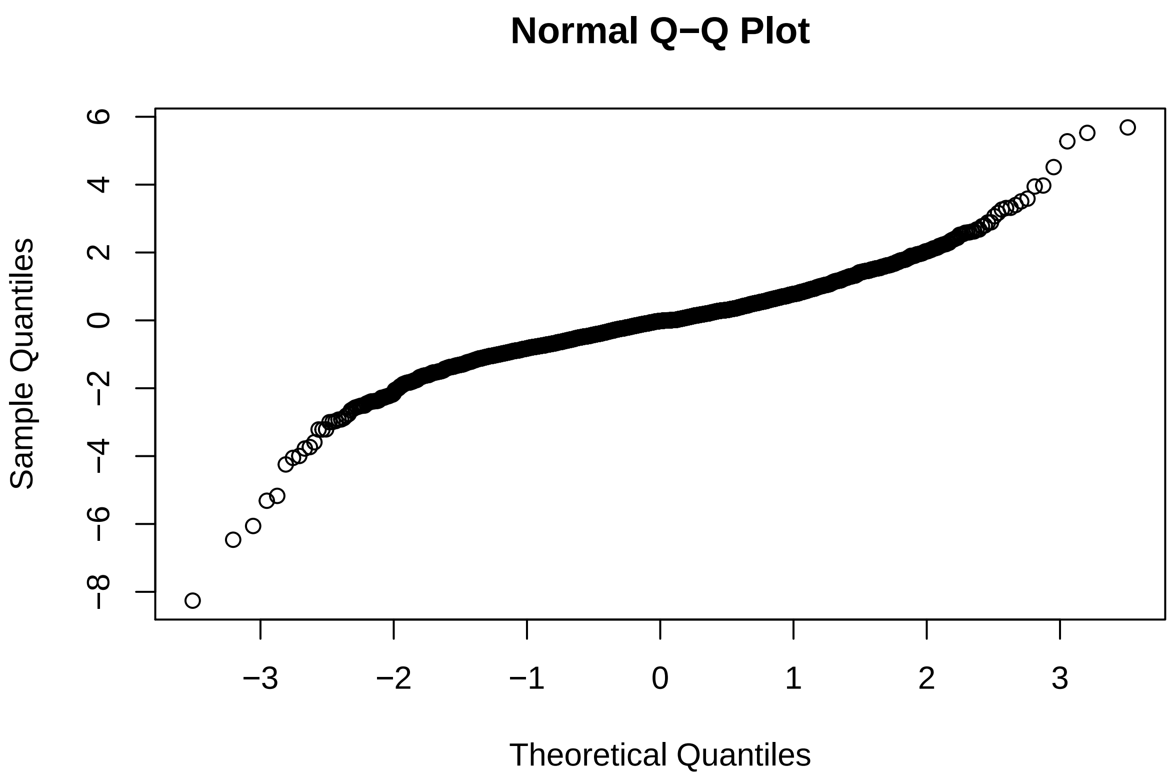
```
conf_int = matrix(0, 2235, 3)
conf_int[,1] = m.fit
conf_int[,2:3] = m.fit + cbind(-qnorm(0.975)*sqrt(ht),qnorm(0.975)*sqrt(ht))

plot(conf_int[,2],type = 'l',ylim = c(39,53),ylab = c(''));par(new = T)
plot(conf_int[,3],type = 'l',ylim = c(39,53),ylab =c(''));par(new = T)
polygon(c(1:2235,rev(1:2235)),c(conf_int[,3],rev(conf_int[,2])),col="grey");par(new = T)
plot(conf_int[,1],type = 'l',ylim = c(39,53),col = 'red',ylab = c('log exchange'))
```



The one-step-ahead forecast covers the true log exchange rate most of the times. The confidence interval is wide when the previous log exchange volatility is high and narrow when it's low. The model seems to have great predictive power in terms of very high accuracy. However we are using the model fitted by the whole data and explaining the same data. This forecast interval may be too optimistic.

8)

```
m.garch.resid = m.garch$residuals
qqnorm(na.omit(m.garch.resid))
```



From the Normal Q-Q plot, it shows clear long-tailedness. Thus the garch model is adequante to describe the leptokurtosis.

9)

```
cur = na.omit(m.garch.resid)
sum(abs(cur)>1.96)/length(cur)
```

```
## [1] 0.05087798
```

The interval fails 5.087798 of time