Problem Set 6

Philip Zhou

March 7, 2016

1

Statistic	S0	S1	S2	S3	S5	S6
Mean	10.718	14.378	10.440	18.899	10.408	87.662
Median	7.325	9.779	8.505	15.278	8.566	83.809
Minimum	0.000	0.000	0.858	1.505	1.004	28.671
Maximum	144.027	110.996	80.355	140.914	94.005	277.526

```
set.seed(10)
require(quantmod)
require(fBasics)
source("yz.R")
getSymbols("AAPL", from = "2007-01-03", to = "2015-04-30")


## [1] "AAPL"


AAPL["/2014-06-06", ] <- AAPL["/2014-06-06", ]/7

open <- AAPL$AAPL.Open
close <- AAPL$AAPL.Close
high <- AAPL$AAPL.High
low <- AAPL$AAPL.Low
N <- length(open)
f <- (24 - 6.5)/24

S0 <- sqrt(252 * (diff(close)^2))
S0 <- S0[-1, ]
S1 <- sqrt(252 * ((open[2:N] - close[1:N - 1])^2/(2 * f) + (close[2:N] - open[2:N])^2/(2 *
    (1 - f))))
S2 <- sqrt(252 * ((high - low)^2/(4 * log(2))))
S3 <- sqrt(252 * (0.17 * (open[2:N] - close[1:N - 1])^2/f + 0.83 * (high[2:N] - low[2:N])^2/((1 -
    f) * 4 * log(2))))
S5 <- sqrt(252 * (0.5 * (high[2:N] - low[2:N])^2 - (2 * log(2) - 1) * (close[2:N] - open[2:N])^2))
S6 <- sqrt(252 * (0.12 * (open[2:N] - close[1:N - 1])^2/f + 0.88 * S5/(1 - f)))

tableS <- cbind(basicStats(S0), basicStats(S1), basicStats(S2), basicStats(S3), basicStats(S5),
    basicStats(S6))
colnames(tableS) <- c("S0", "S1", "S2", "S3", "S5", "S6")
print(tableS)
```

```
##                     S0           S1           S2           S3           S5           S6
## nobs          2095.000000  2094.000000  2096.000000  2094.000000  2095.000000 2.094000e+03
## NAs              0.000000     0.000000     0.000000     0.000000     0.000000 0.000000e+00
## Minimum          0.000000     0.000000     0.858017     1.504601     1.003925 2.867098e+01
## Maximum        144.027199   110.996486    80.354746   140.914021    94.005303 2.775261e+02
## 1. Quartile      3.106850     4.185822     5.362676     9.664994     5.402076 6.651257e+01
## 3. Quartile     13.992262    19.449740    13.333409    24.252554    13.304786 1.044581e+02
## Mean            10.717603    14.378020    10.447929    18.913012    10.419036 8.770139e+01
## Median           7.324906     9.778930     8.505349    15.277813     8.567015 8.382304e+01
## Sum          22453.378565 30107.573182 21898.859360 39603.846988 21827.879834 1.836467e+05
## SE Mean          0.260665     0.326925     0.165545     0.301810     0.163883 6.410150e-01
## LCL Mean        10.206413    13.736887    10.123280    18.321133    10.097646 8.644430e+01
## UCL Mean        11.228793    15.019152    10.772578    19.504891    10.740426 8.895849e+01
## Variance       142.347778   223.806983    57.440878   190.740992    56.266506 8.604254e+02
## Stdev           11.930959    14.960180     7.578976    13.810901     7.501100 2.933301e+01
## Skewness         3.163403     2.231391     2.307092     2.263734     2.731196 9.658460e-01
## Kurtosis        17.739489     7.059283    10.114393     9.380953    16.535375 2.097950e+00
```

2

The time plot of volatility is shown below. The model is: $(1 - B)x_t = a_t - 0.0802a_{t-1} - 0.0635a_{t-2}$. Both of the coefficients are significant. The 1-step to 5-step predictions are 0.1950, 0.1952, 0.1953, 0.1954, and 0.1955. It's important to know that controlling for outliers in the residuals in the model leads to a much lower AIC. I was, however, unable to predict with the new model that controlled for outliers.

```
open=unclass(open)
close=unclass(close)
high=unclass(high)
low=unclass(low)
m1=yz(open,high,low,close)
varyz=sqrt(252*m1$yzsq)
ts.plot(varyz,main="Time Plot of Estimated Volatility")
yy=log(varyz[64:length(varyz)])
t.test(yy)
```

```
##
##  One Sample t-test
##
## data:  yy
## t = -146.5223, df = 2031, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -1.178686 -1.147550
## sample estimates:
## mean of x
## -1.163118
```

```
m2=arima(yy,order=c(0,1,2))
m2
```

```
##
## Call:
```

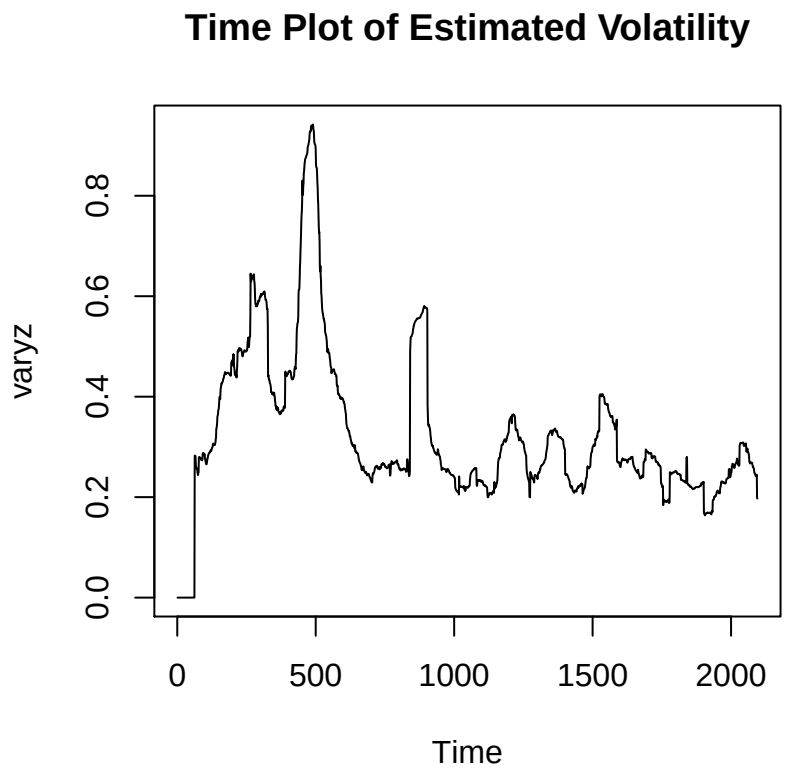
```
## arima(x = yy, order = c(0, 1, 2))
##
## Coefficients:
##          ma1     ma2
##       0.0805  0.0635
## s.e.  0.0222  0.0221
##
## sigma^2 estimated as 0.0008174:  log likelihood = 4337.7,  aic = -8669.41
```

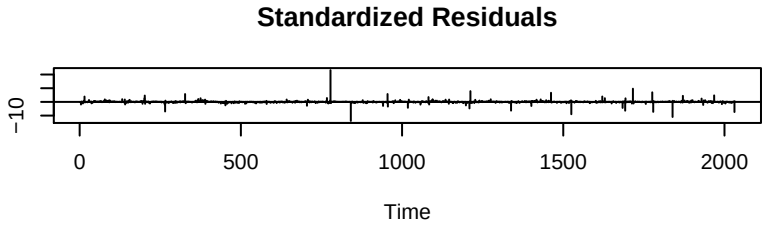
```
tsdiag(m2)

predictm2=predict(m2,5)
predictm2
```

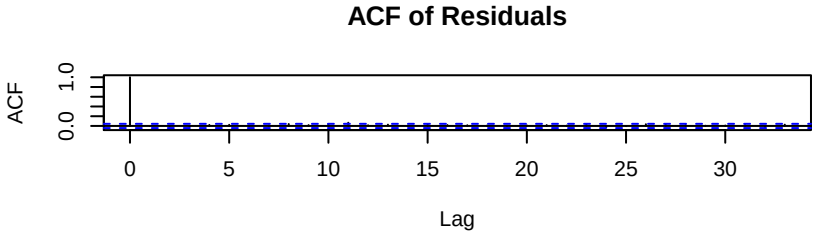
```
## $pred
## Time Series:
## Start = 2033
## End = 2037
## Frequency = 1
## [1] -1.635306 -1.634573 -1.634573 -1.634573 -1.634573
##
## $se
## Time Series:
## Start = 2033
## End = 2037
## Frequency = 1
## [1] 0.02859024 0.04209214 0.05330715 0.06254244 0.07057946
```

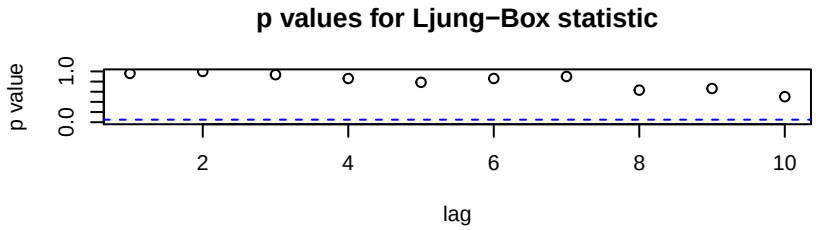
```
pp=exp(predictm2$pred+0.5*predictm2$se^2)
pp
```

```
## Time Series:
## Start = 2033
## End = 2037
## Frequency = 1
## [1] 0.1949724 0.1952085 0.1953130 0.1954175 0.1955221
```









Here is the code to remove the outlier. The AIC is much lower (-8962.11 << -8667.76)

```
Ioutlier=rep(0,length(m2$resid))
Ioutlier[which.max(m2$resid)]=1
m3=arima(yy,order=c(0,1,2),xreg=Ioutlier)
m3


##
```

```
## Call:
## arima(x = yy, order = c(0, 1, 2), xreg = Ioutlier)
##
## Coefficients:
##          ma1     ma2  Ioutlier
##       0.1583  0.0754    0.3185
## s.e.  0.0223  0.0222    0.0173
##
## sigma^2 estimated as 0.0007051:  log likelihood = 4486.07,  aic = -8964.14
```

3

a. The logit model is:
$logit(p_t) = 0.26288 + 0.01638M_{t\text{-}1}$ - $0.16057S_{t\text{-}1}$ - $0.29438M_{t\text{-}2}$ - $0.03684S_{t\text{-}2}$, $p_t = P(M_t = 1)$.
Only the p-value of the $M_{t\text{-}2}$ coefficient is significant. It seems like the model isn't particularly helpful, although $M_{t\text{-}2}$ is informative.

```
cokedata=read.table("m-kosp-4114.txt",header=T)
coke=log(cokedata$ko+1)
sandp=log(cokedata$sprtrn+1)
Mt=ifelse(coke>0,1,0)
St=ifelse(sandp>0,1,0)
Mtnolags=Mt[3:length(Mt)]
Mt1lag=Mt[2:(length(Mt)-1)]
Mt2lag=Mt[1:(length(Mt)-2)]
St1lag=St[2:(length(St)-1)]
St2lag=St[1:(length(St)-2)]
m1=glm(Mtnolags~Mt1lag+St1lag+Mt2lag+St2lag,family=binomial)
summary(m1)


##
## Call:
## glm(formula = Mtnolags ~ Mt1lag + St1lag + Mt2lag + St2lag, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4287  -1.2909   0.9658   1.0289   1.1497
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26288    0.15423   1.704   0.0883 .
## Mt1lag       0.01638    0.14844   0.110   0.9121
## St1lag      -0.16057    0.14973  -1.072   0.2836
## Mt2lag       0.29438    0.14790   1.990   0.0465 *
## St2lag      -0.03684    0.14908  -0.247   0.8048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1205.4  on 885  degrees of freedom
## Residual deviance: 1200.2  on 881  degrees of freedom
## AIC: 1210.2
```

```
##
## Number of Fisher Scoring iterations: 4
```

b. The network model is:

$$h(o_t) = \begin{cases} 1 & : o_t > 0 \\ 0 & : o_t \leq 0 \end{cases}$$

$o_t = -2.75 + 2.12h_{t1} + 1.50h_{2t} - 0.16M_{t\text{-}1} - 0.06S_{t\text{-}1} + 2.98M_{t\text{-}2} + 0.35S_{t\text{-}2}$, and

$$h_{1t} = \frac{exp(2.06 + 1.89M_{t\text{-}1} + 2.65S_{t\text{-}1} - 8.75M_{t\text{-}2} - 2.07S_{t\text{-}2})}{1 + exp(2.06 + 1.89M_{t\text{-}1} + 2.65S_{t\text{-}1} - 8.75M_{t\text{-}2} - 2.07S_{t\text{-}2})}$$

$$h_{2t} = \frac{exp(2.03 + 0.58M_{t\text{-}1} - 1.94S_{t\text{-}1} - 4.35M_{t\text{-}2} - 1.18S_{t\text{-}2})}{1 + exp(2.03 + 0.58M_{t\text{-}1} - 1.94S_{t\text{-}1} - 4.35M_{t\text{-}2} - 1.18S_{t\text{-}2})}$$

```
X=cbind(Mt1lag,St1lag,Mt2lag,St2lag)
require(nnet)
m2=nnet(X,Mtnolags,size=2,skip=T)


## # weights:  17
## initial  value 250.823711
## iter  10 value 214.384024
## iter  20 value 213.555473
## iter  30 value 213.503848
## iter  40 value 213.201620
## iter  50 value 213.043039
## iter  60 value 212.980807
## iter  70 value 212.970705
## iter  80 value 212.970283
## iter  90 value 212.969859
## iter 100 value 212.968872
## final  value 212.968872
## stopped after 100 iterations


summary(m2)


## a 4-2-1 network with 17 weights
## options were - skip-layer connections
##  b->h1 i1->h1 i2->h1 i3->h1 i4->h1
##   2.06   1.89   2.65  -8.75  -2.07
##  b->h2 i1->h2 i2->h2 i3->h2 i4->h2
##   2.03   0.58  -1.94  -4.35  -1.18
##  b->o h1->o h2->o i1->o i2->o i3->o i4->o
## -2.75  2.12  1.50 -0.16 -0.06  2.98  0.35
```

c. If we are looking at out-of-sample predictions, the sum of square of forecast errors for the logistic regression is 37 and those for the neural regressions is 38. They are essentially equivalent.

```
yf=Mtnolags[803:886]
Xf=X[803:886,]
yfit=Mtnolags[1:802]
```

```
Xfit=X[1:802,]
m4=glm(yfit~Xfit,family=binomial)

coef=m4$coefficients
Xfbind=cbind(rep(1,84),Xf)
m4p=exp(Xfbind%*%as.matrix(coef,5,1))/(1+exp(Xfbind%*%as.matrix(coef,5,1)))
logi=c(1:84)[m4p >= 0.5]
yhat=rep(0,84)
yhat[logi]=1
sum((yf-yhat)^2)


## [1] 37


m5=nnet(Xfit,yfit,size=2,skip=T)


## # weights:  17
## initial  value 234.857254
## iter  10 value 194.309008
## iter  20 value 193.822846
## iter  30 value 193.673409
## iter  40 value 193.549602
## iter  50 value 193.509605
## iter  60 value 193.420751
## iter  70 value 193.381665
## iter  80 value 193.380238
## iter  90 value 193.378717
## iter 100 value 193.376068
## final  value 193.376068
## stopped after 100 iterations


predictm5=predict(m5,Xf)
ypredict=ifelse(predictm5 > 0.5,1,0)
sum((yf-ypredict)^2)


## [1] 38
```

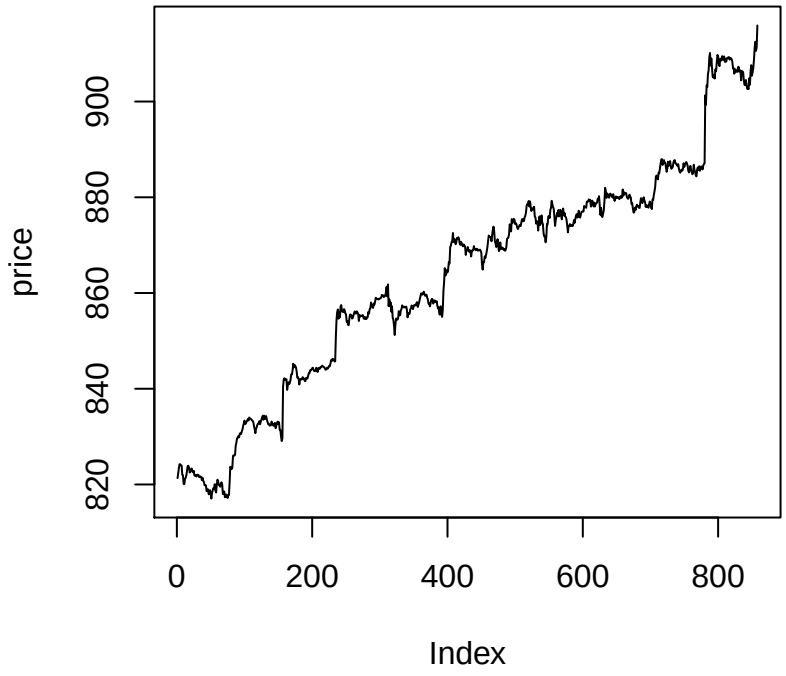
4

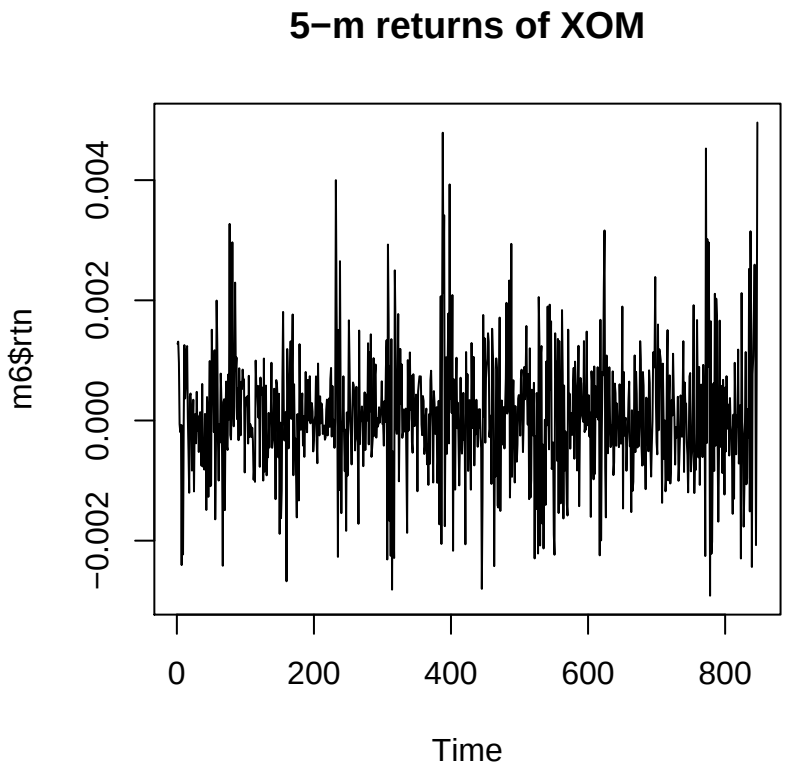
The plot of the five minute log returns can be found below. We found that there are no serial correlations in the five minute log returns, since the Ljung box test returned a p-value of .34.

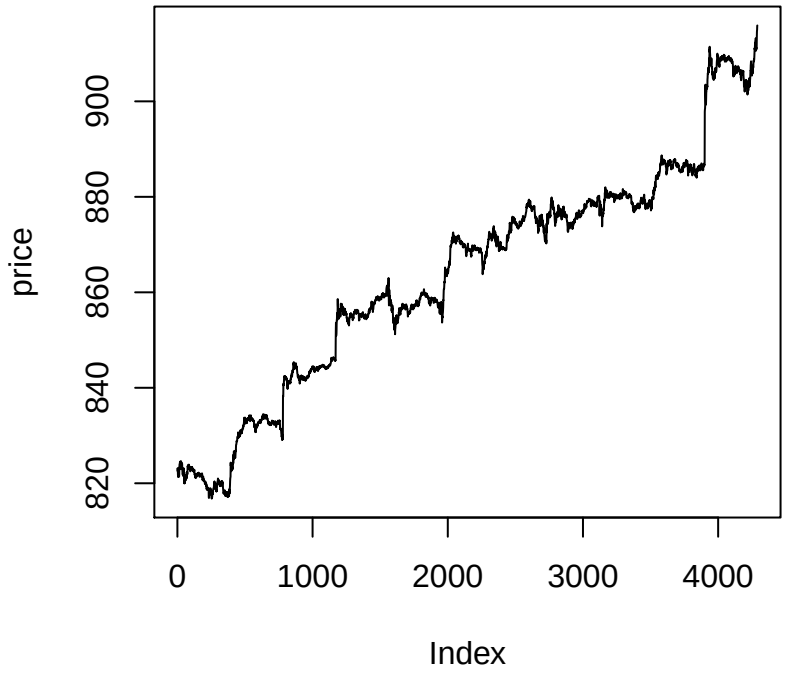
```
db=read.table("taq-goog-may1t152013.txt",header=T)
source("hfrtn.R")
m6=hfrtn(db,5)

ts.plot(m6$rtn,main="5-m returns of XOM")
Box.test(m6$rtn,lag=10,type='Ljung')


##
```

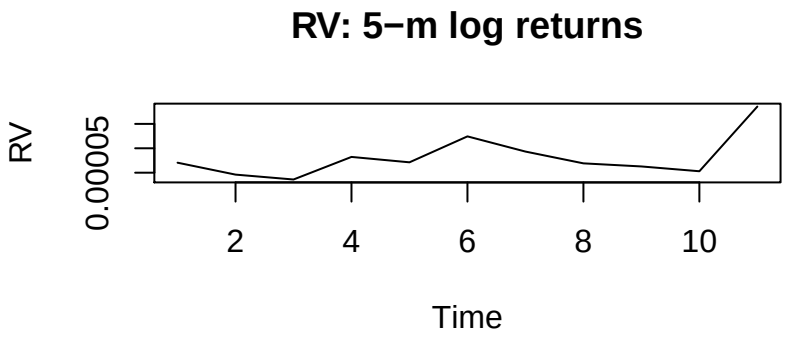
	RV	RV1
1	0.00007049	0.00010308
2	0.00004610	0.00008604
3	0.00003610	0.00005022
4	0.00008232	0.00010841
5	0.00007128	0.00009309
6	0.00012442	0.00009939
7	0.00009331	0.00009237
8	0.00006913	0.00008672
9	0.00006280	0.00008677
10	0.00005295	0.00006664
11	0.00018549	0.00015375

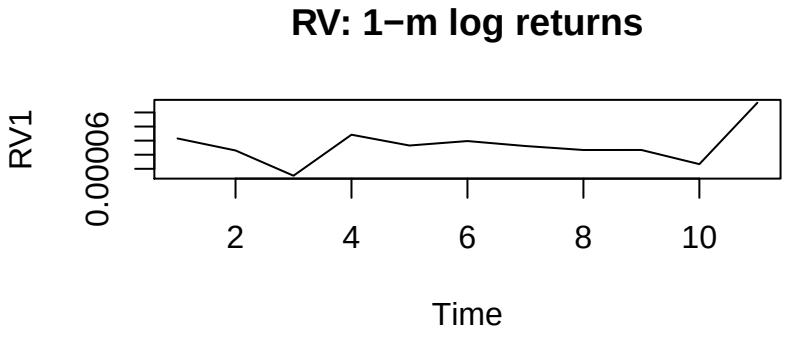
```
##  Box-Ljung test
##
## data:  m6$rtn
## X-squared = 11.2291, df = 10, p-value = 0.34


RV=NULL
for (i in 1:11){
  daycount=(i-1)*77
  x=sum(m6$rtn[(daycount+1):(daycount+77)]^2)
  RV=c(RV,x)
}

m7=hfrtn(db,1)
RV1=NULL
for (i in 1:11){
  daycount=(i-1)*389
  x=sum(m7$rtn[(daycount+1):(daycount+389)]^2)
  RV1=c(RV1,x)
}

tableRV=cbind(RV,RV1)
colnames(tableRV)<-c("RV","RV1")

par(mfcol=c(2,1))
ts.plot(RV,main="RV: 5-m log returns")
ts.plot(RV1,main="RV: 1-m log returns")
```


price
820
840
860
880
900
0
200
400
600
800
Index


5−m returns of XOM
m6$rtn
−0.002
0.000
0.002
0.004
0
200
400
600
800
Time



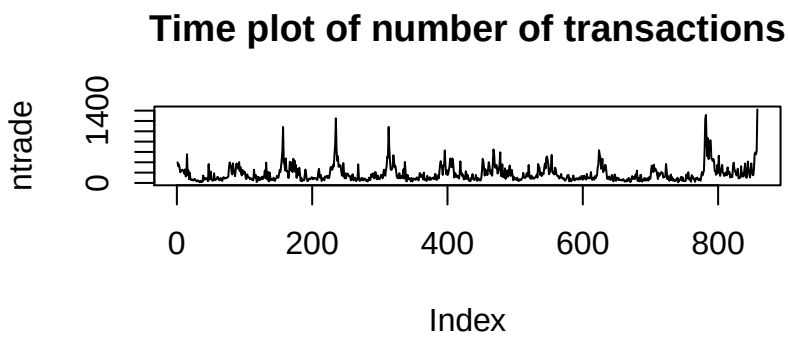


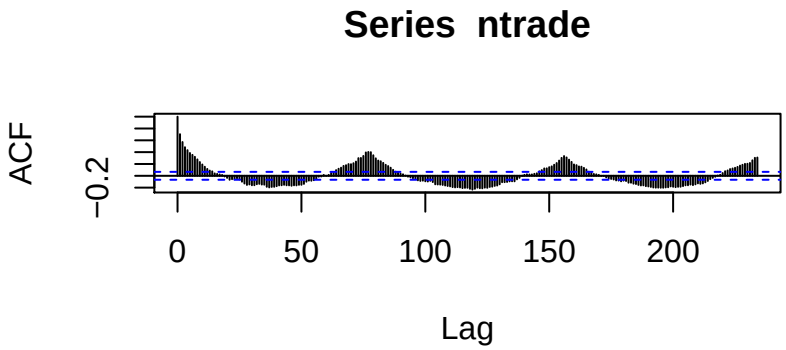


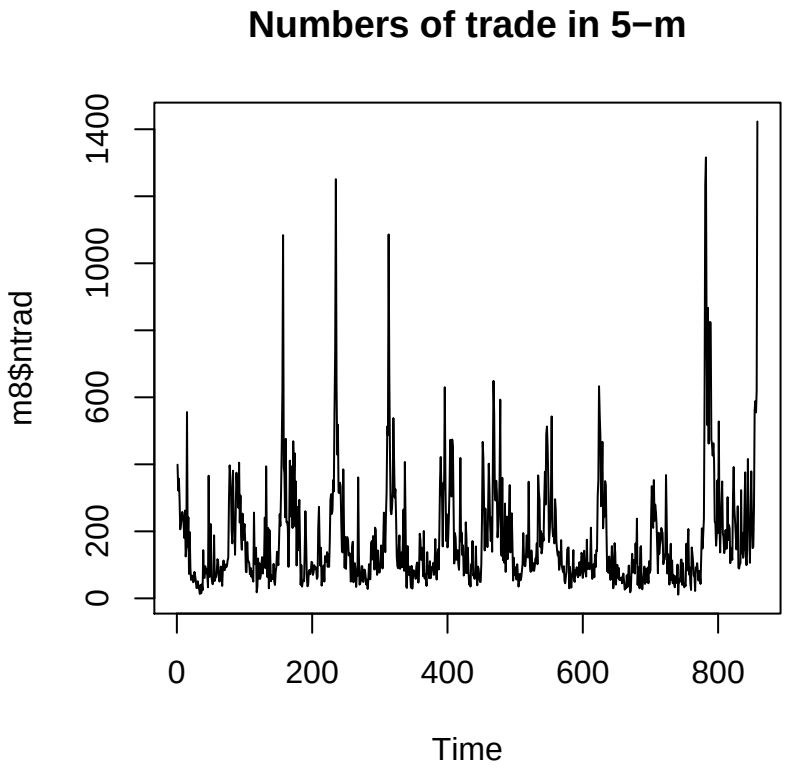
5

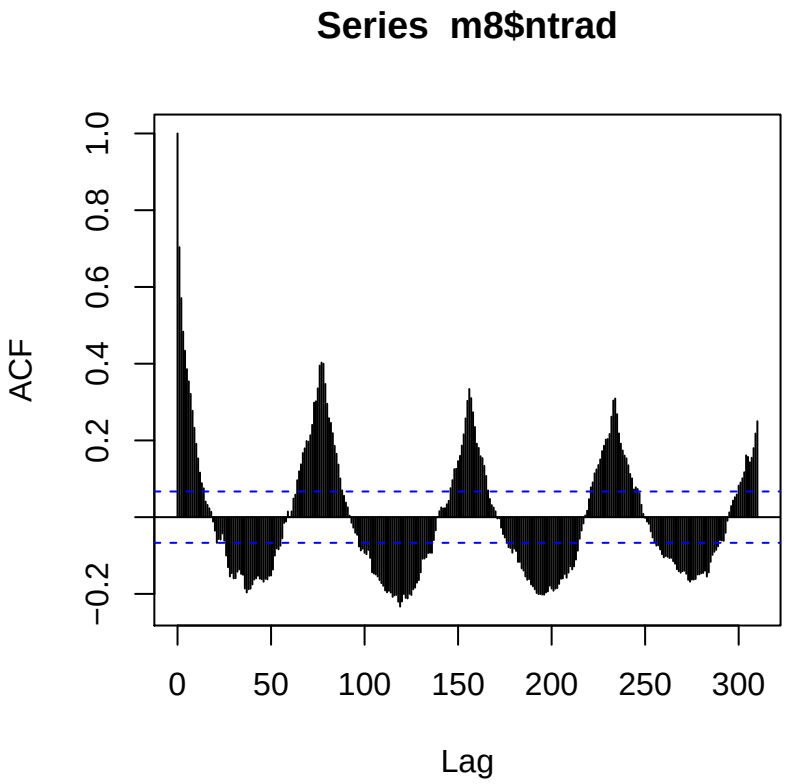
The plot of the intensity of trades is shown below. The ACF is also diplayed below, and we see a clear diurnal pattern, as evidenced by the cyclical nature of the ACF.

```r
source("hfntra.R")
m8=hfntra(db,5)
par(mfcol=c(1,1))
ts.plot(m8$ntrad,main="Numbers of trade in 5-m")
acf(m8$ntrad,lag.max=310)
```








Series m8$ntrad
ACF
1.0
0.8
0.6
0.4
0.2
−0.2
0
50
100
150
200
250
300
Lag