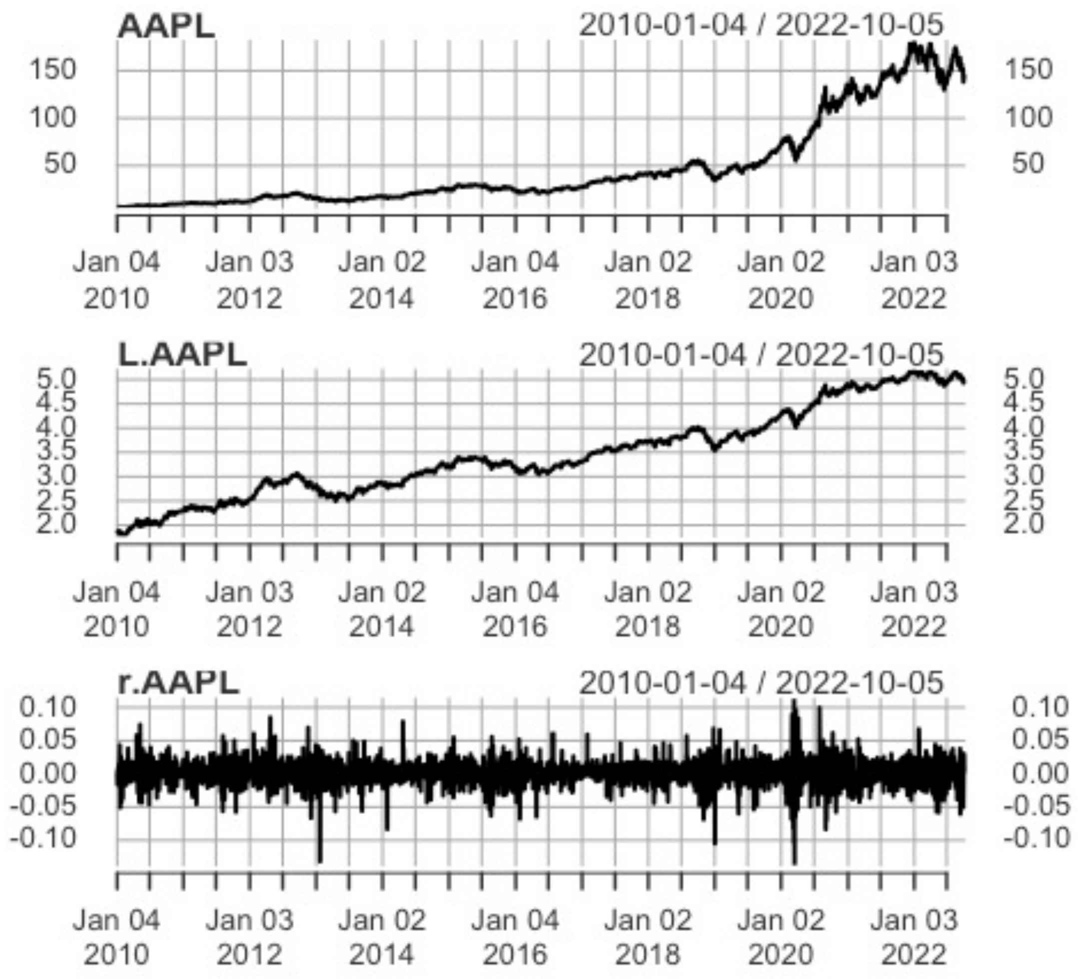
Applied Financial Econometrics

Class 6: Reviewing the course contents and assignments + GARCH modelling

Lecturer: Axel A. Araneda, Ph.D.

Reviewing the previous class




```
###### Dickey-Fuller test ###########
#library(tseries)
adf.test(AAPL,k=1) # k: number of lags
adf.test(L.AAPL,k=1)
adf.test(r.AAPL[2:length(r.AAPL)],k=1)
```

```
        Augmented Dickey-Fuller Test

data:  AAPL
Dickey-Fuller = -1.4038, Lag order = 1, p-value = 0.8307
alternative hypothesis: stationary
```

```
data:  L.AAPL
Dickey-Fuller = -2.3349, Lag order = 1, p-value = 0.4365
alternative hypothesis: stationary
```

```
data:  r.AAPL[2:length(r.AAPL)]
Dickey-Fuller = -40.737, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```


MUNI ECON

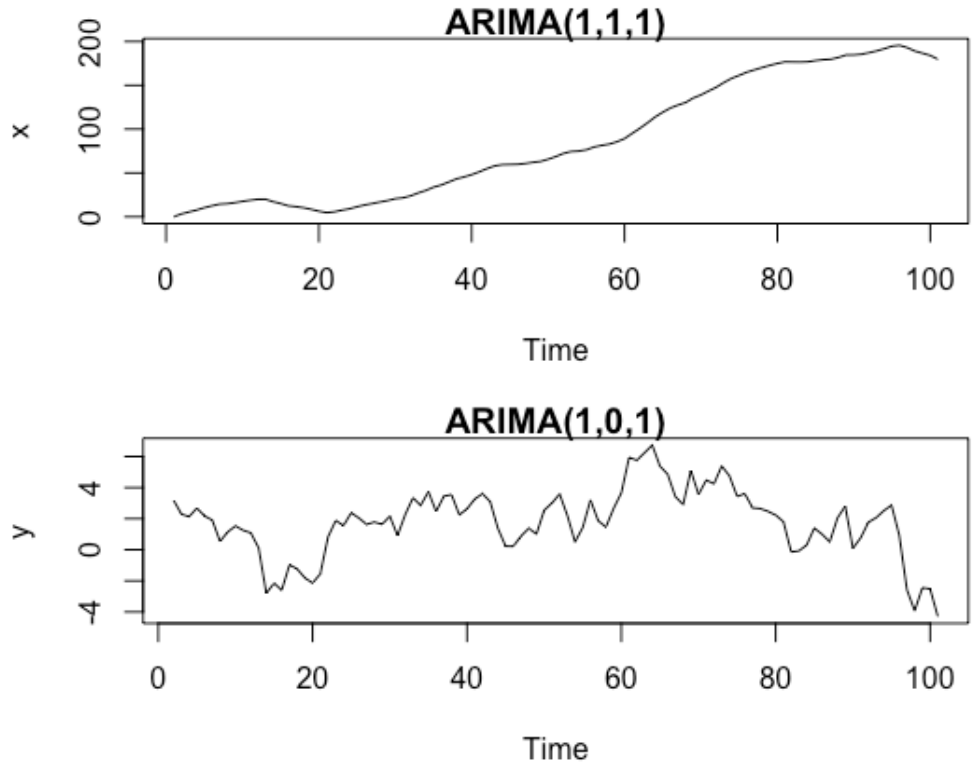
ARIMA modelling

ARIMA(p,d,q):

$$x_t \sim I(d), \quad y_t = \Delta^d x_t, \quad \varepsilon_t \sim N(0,1)$$

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-i} + \varepsilon_t$$

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) y_t = \left(1 + \sum_{j=1}^{q} \theta_i L^i\right) \varepsilon_t$$




```
par(mar=c(5,4,1,1),mfrow=c(2,1))
x<-arima.sim(list(order=c(1,1,1),ar=.9,ma=.2),n=100)
plot(x,ylab='x',main='ARIMA(1,1,1)')
plot(diff(x),ylab='y',main='ARIMA(1,0,1)')
```

ARIMA modelling

```
library(lmtest)
arima_1_1_1=arima(L.AAPL, order=c(1,1,1))
coeftest(arima_1_1_1)
```

```
z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ar1 -0.36729    0.31992 -1.1481   0.2509
ma1  0.32451    0.32331  1.0037   0.3155
```

```
arima_1_1_0=arima(L.AAPL, order=c(1,1,0))
coeftest(arima_1_1_0)
```

```
     Estimate Std. Error z value Pr(>|z|)
ar1 -0.043228   0.017622 -2.4531  0.01416 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’
```

```
arima_0_1_1=arima(L.AAPL, order=c(1,1,0))
coeftest(arima_0_1_1)
```

```
     Estimate Std. Error z value Pr(>|z|)
ma1 -0.042605   0.017506 -2.4337  0.01494 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’
```

```
arima_0_0_0=arima(L.AAPL, order=c(0,1,0))
coeftest(arima_0_0_0)
```

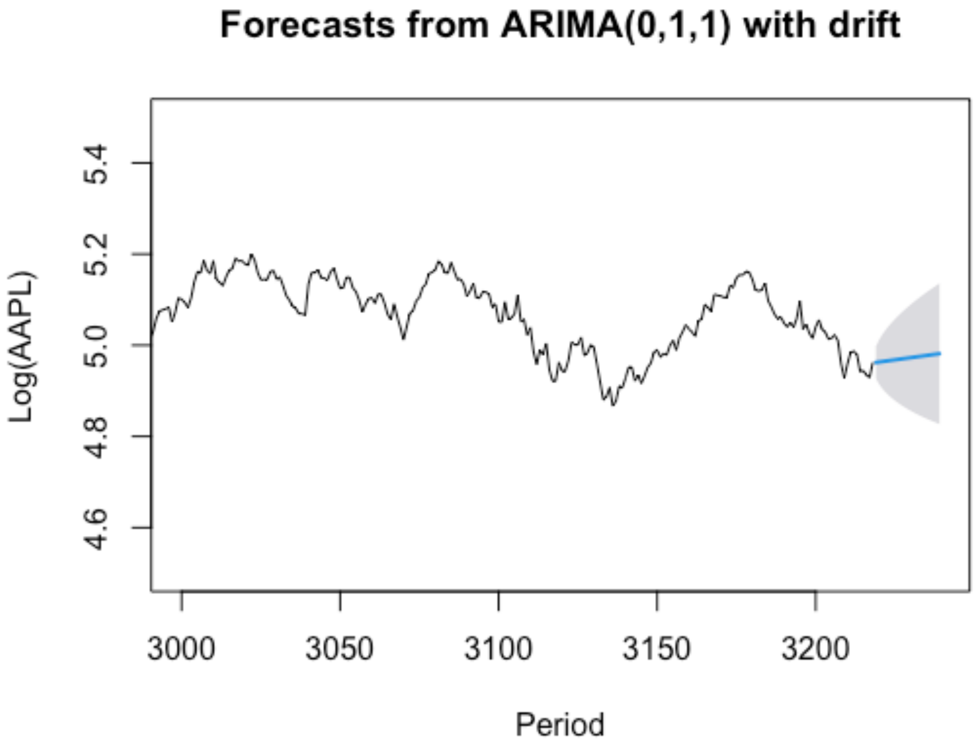
ARIMA modelling

```
##### Model selection #####

#comparing AIC (lower wins)
arima_1_1_1$aic
arima_1_1_0$aic
arima_0_1_1$aic
arima_0_1_0$aic

#comparing BIC (lower wins)
BIC(arima_1_1_1)
BIC(arima_1_1_0)
BIC(arima_0_1_1)
BIC(arima_0_1_0)
```

```
auto_arima <- auto.arima(L.AAPL)
```




```
##################Forecasting ####################
forecasting<-forecast(auto_arima,21,level=95)
# Forecast future 21 values and 95% CI
plot(autoarima_forecasting, xlim=c(3000,3239),
ylim=c(4.5,5.5),ylab=('Log(AAPL)'), xlab=('Period'))
```


MUNI ECON

Assignment 3

1. Download the daily historical price for the last ten years for two stocks of your interest. These stocks must fulfill the same sector (i.e., industrials, consumer staples, financials, utilities, etc.). Repeat the procedure for a new pair of stocks belonging to a different sector (Four stocks in total). For example, KO & PEP (Consumer Staples) and HP & IBM (informatics).

   (a) Are the stock prices stationary? What about of the log-prices? Are booth an Integrated process? Which oder?

2. For the whole log-prices used in the previous point computes an ARMA (1,1), ARMA (1,0), and ARMA (0,1) and evaluates which one is better.

3. Using the VIX index and the ARIMA modelling (the best one) predict the 1-month implied volatility for the following week.

MUNI
ECON

Assignment 3 (3)

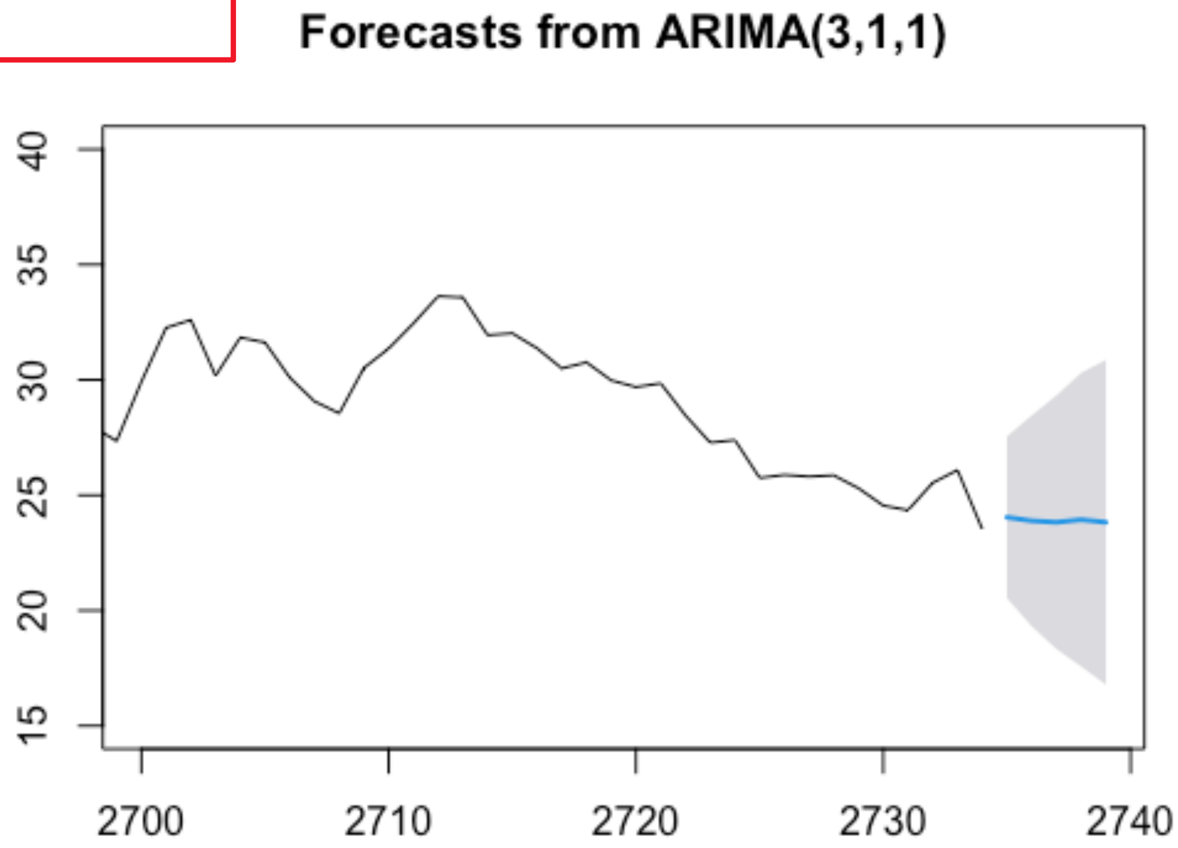
```
getSymbols('^VIX',src='yahoo', from="2012-01-01",periodicity = 'daily')
# VIX since 2012
VIX<-VIX[,6]
```

```
best<-auto.arima(VIX)
coeftest(best)
```

```
z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ar1 -0.997221   0.052294 -19.0695 < 2.2e-16 ***
ar2 -0.126861   0.028534  -4.4459 8.751e-06 ***
ar3  0.059174   0.020317   2.9125  0.003586 **
ma1  0.834010   0.049158  16.9658 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
```

```
oneweek<-forecast(best,5,level = 95)
plot(oneweek,ylim=c(15,40),xlim=c(2700,2739))
```



MUNI
ECON

DL and ARDL models

- If the regression model includes current and lagged (past) values of the explanatory variables (the X's), it is called a distributed lagged model.

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + \beta_k X_{t-k} + u_t$$

- The coecient β_0 is known as the short-run or impact multiplier

- An autoregressive distributed lag model (ARDL) is a model that contains both independent variables and their lagged values as well as the lagged values of the dependent variable

$$Y_t = \delta + \theta_1 Y_{t-1} + \ldots + \theta_p Y_{t-p} + \delta_1 X_{t-1} + \ldots + \delta_q X_{t-q} + v_t$$

MUNI ECON

Selection model criteria and forecasting performance

$$AIC = log\hat{\sigma}_k^2 + \frac{n+2k}{n}$$

$$BIC = log\hat{\sigma}_k^2 + \frac{klogn}{n}$$

where $\hat{\sigma}_k^2 = \frac{SSE_k}{n}$, and k is the number of model parameters, n the sample size, and SSE_k is equal to the sum of the squared residuals under the model k ($SSE_k = \sum_{t=1}^{n}(x_t - \bar{x})^2$).

1. Mean Error,
2. MAE - Mean Absolute Error,
3. MSE - Mean Squared Error,
4. MIS - Mean Interval Score
5. MPE - Mean Percentage Error
6. MAPE - Mean Absolute Percentage Error

MUNI ECON

ARCH and GARCH models

$$\epsilon_t = \sigma_t w_t$$

$$w_t \sim N(0,1)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta \epsilon_{t-1}^2$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

Assignment 4

1. Select two assets from two different sectors and download the last-year historical daily prices. Using the first 11 months (train data), calibrate 2-3 ARIMA models and examine the mean forecasting performance for each one of the approaches using the last-month data (out-of-sample or test data) by means of the mean squared error.

2. Consider that the random variable x_t is described by following process: $x_t = \varepsilon_t$ with:

   (a) $\varepsilon_t \sim N(0,1)$

   (b) $\varepsilon_t = \sigma_t w_t, \quad w_t \sim N(0,1), \quad \sigma_t^2 = \alpha + \beta\varepsilon_{t-1}^2$

   (c) $\varepsilon_t = \sigma_t w_t, \quad w_t \sim N(0,1), \quad \sigma_t^2 = \alpha + \beta\varepsilon_{t-1}^2 + \gamma\sigma_{t-1}^2$

   - Comment on each specification (main features).
   - Simulate a path (1000 values) for x_t using each specification and compare the models empirically.

Assignment 4 (1)

```
getSymbols('^AAPL',src='yahoo', from="2021-11-05"
           ,periodicity = 'daily')
AAPL<-AAPL[,6]
L<-length(AAPL)
m<-21
AAPL_train=AAPL[1:(L-m)]
AAPL_test=AAPL[(L-m+1):L]
```

```
m1<-arima(AAPL_train,order = c(1,1,0))
m2<-arima(AAPL_train,order = c(0,1,1))
m3<-arima(AAPL_train,order = c(1,1,1))
m4<-arima(AAPL_train,order = c(0,1,0))
```

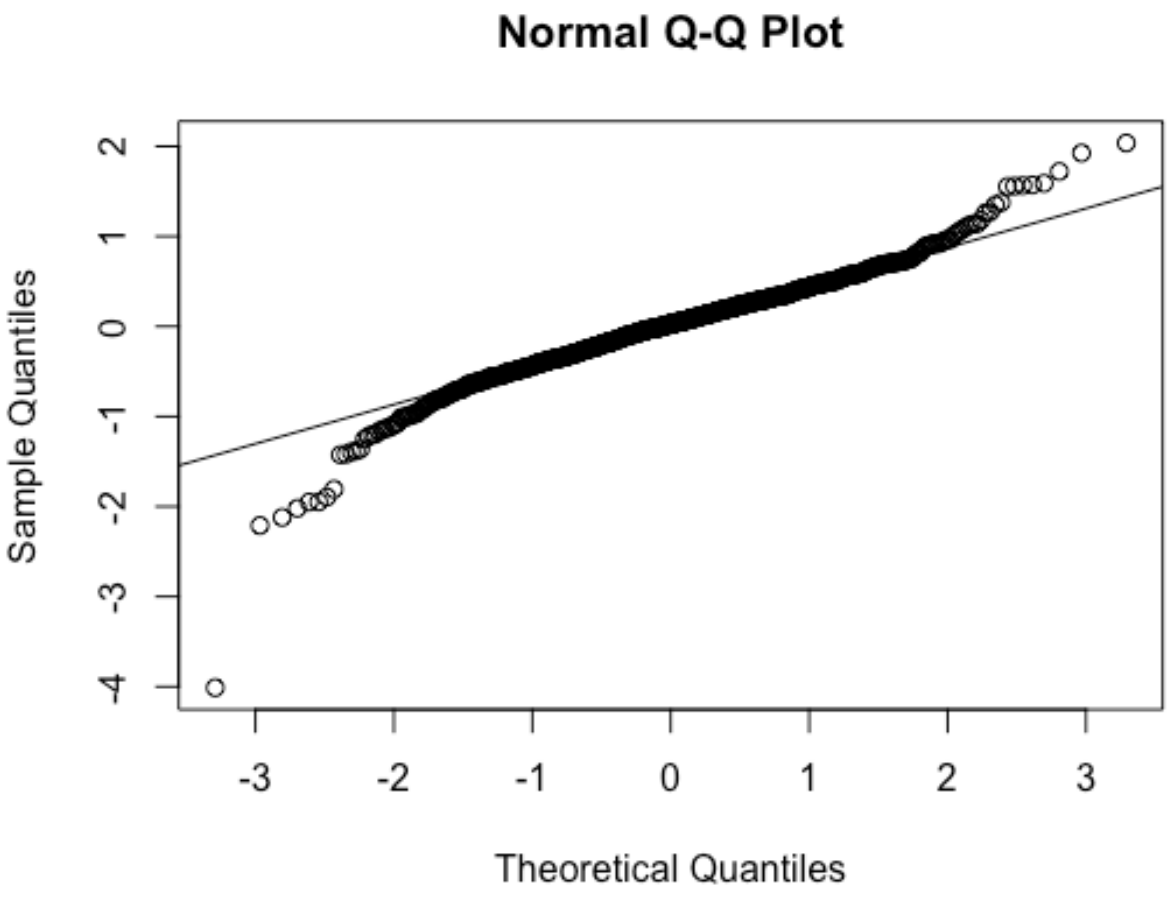
```
m1_test<-arima(AAPL_train, order=c(1,1,0),fixed=m1$coef)
m2_test<-arima(AAPL_train, order=c(0,1,1),fixed=m2$coef)
m3_test<-arima(AAPL_train, order=c(1,1,1),fixed=m3$coef)
m4_test<-arima(AAPL_train, order=c(0,1,0),fixed=m4$coef)
```

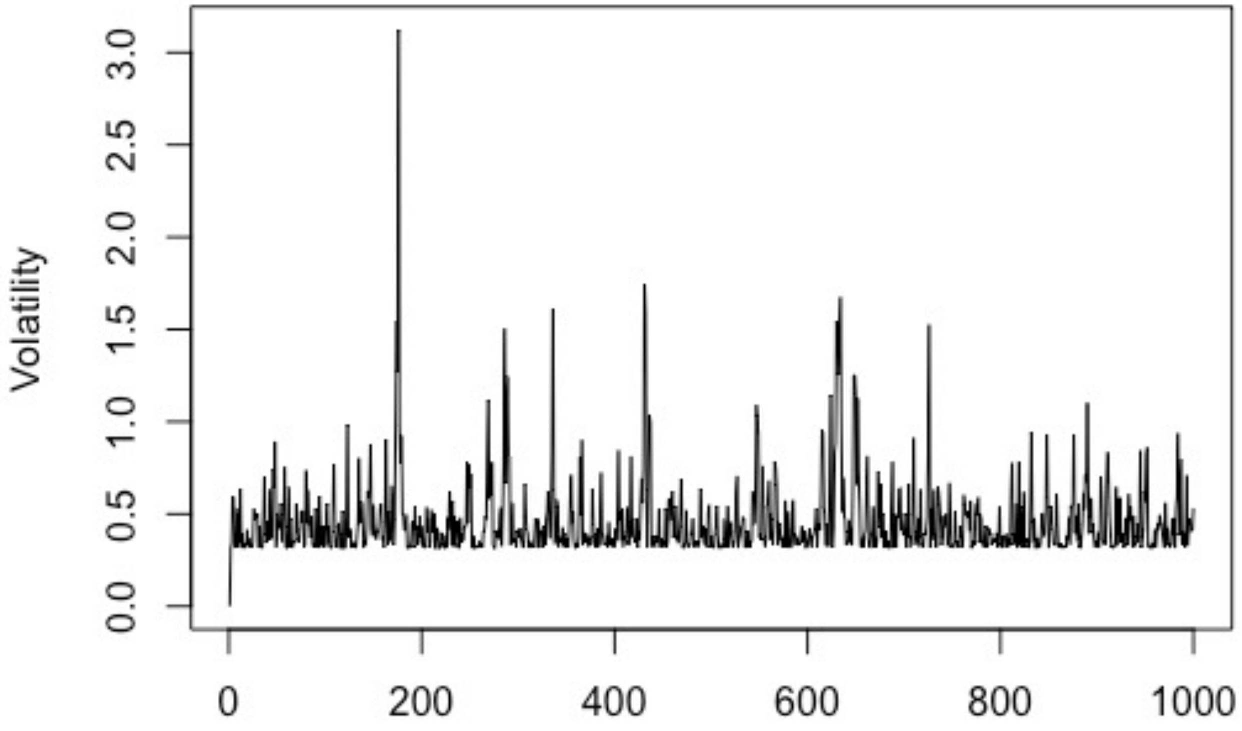
```
A1<-accuracy(m1_test)
A2<-accuracy(m2_test)
A3<-accuracy(m3_test)
A4<-accuracy(m4_test)
RMSE<-c(A1[,2],A2[,2],A3[,2],A4[,2])
```

```
> RMSE
[1] 3.259902 3.259898 3.221729 3.259925
```

Assignment 4 (2b)

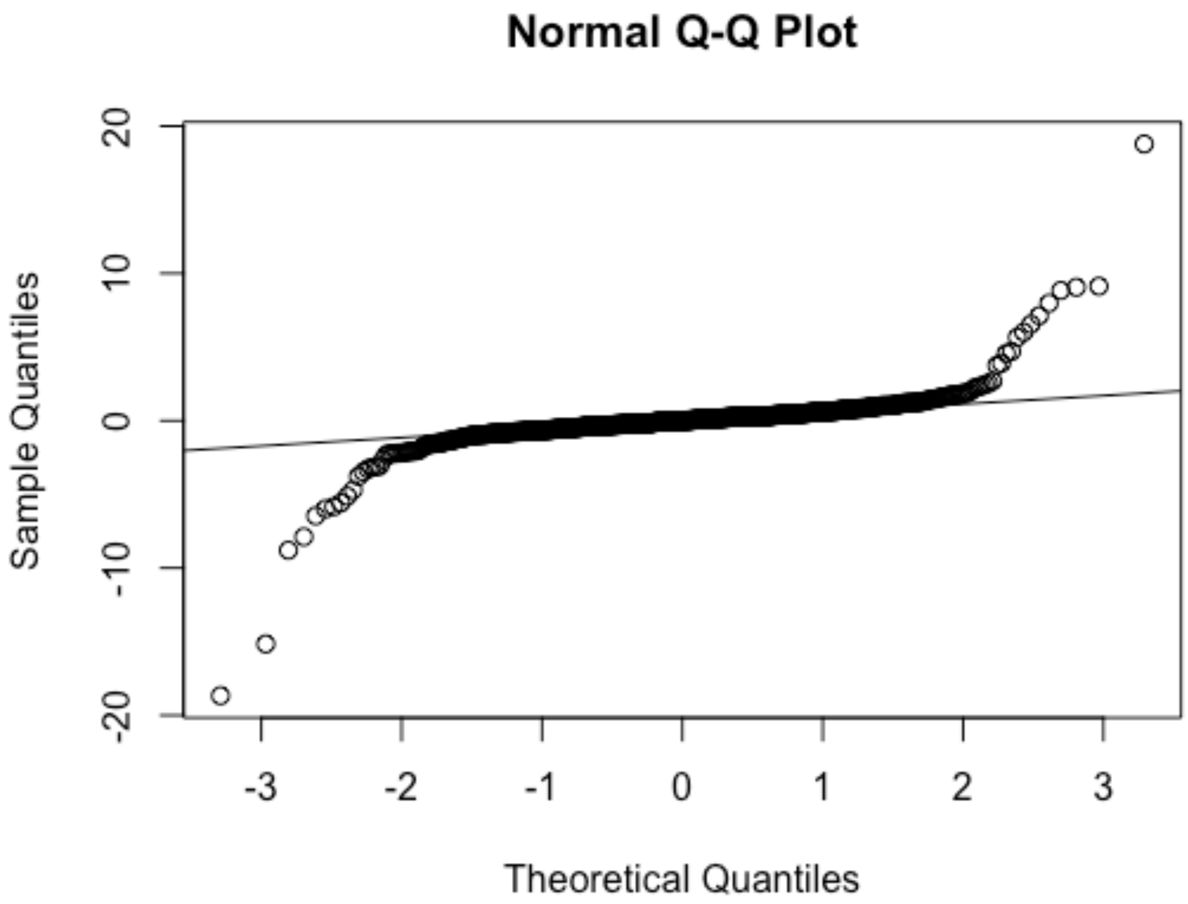
```r
s2<-rep(0,1000); e<-rep(0,1000)
alpha0<-0.1; alpha1<-0.6
for (i in 2:1000) {
  s2[i]<-alpha0+alpha1*(e[i-1])**2
  e[i]<-sqrt(s2[i])*rnorm(1)
}
plot(sqrt(s2),type='l',ylab = 'Volatility')
qqnorm(e); qqline(e)
```

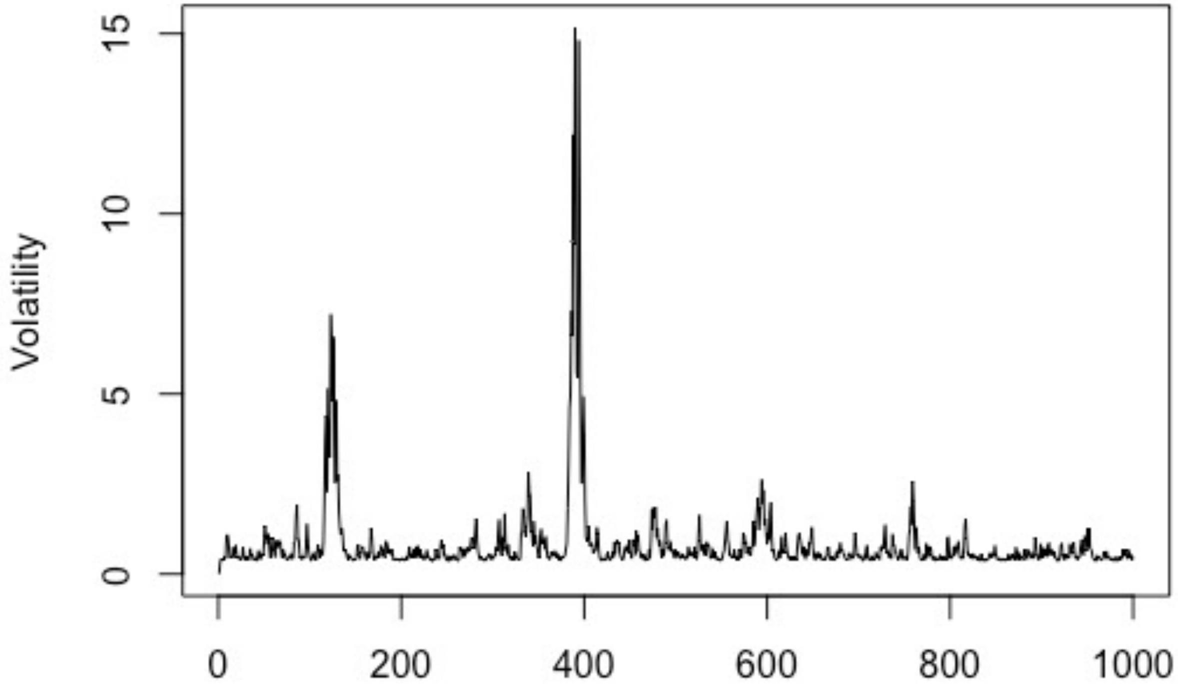




Assignment 4 (2c)

```
s2<-rep(0,1000); e<-rep(0,1000)
alpha0<-0.1; alpha1<-0.6; beta1<-0.25
for (i in 2:1000) {
  s2[i]<-alpha0+alpha1*(e[i-1])**2+beta1*(s2[i-1])
  e[i]<-sqrt(s2[i])*rnorm(1)
}
plot(sqrt(s2),type='l',ylab='Volatility')
qqnorm(e); qqline(e)
```





MUNI
ECON

ARCH test

```r
install.packages('aTSA')
library('aTSA')
mod <- arima(e,order = c(0,0,0))
arch.test(mod)
```

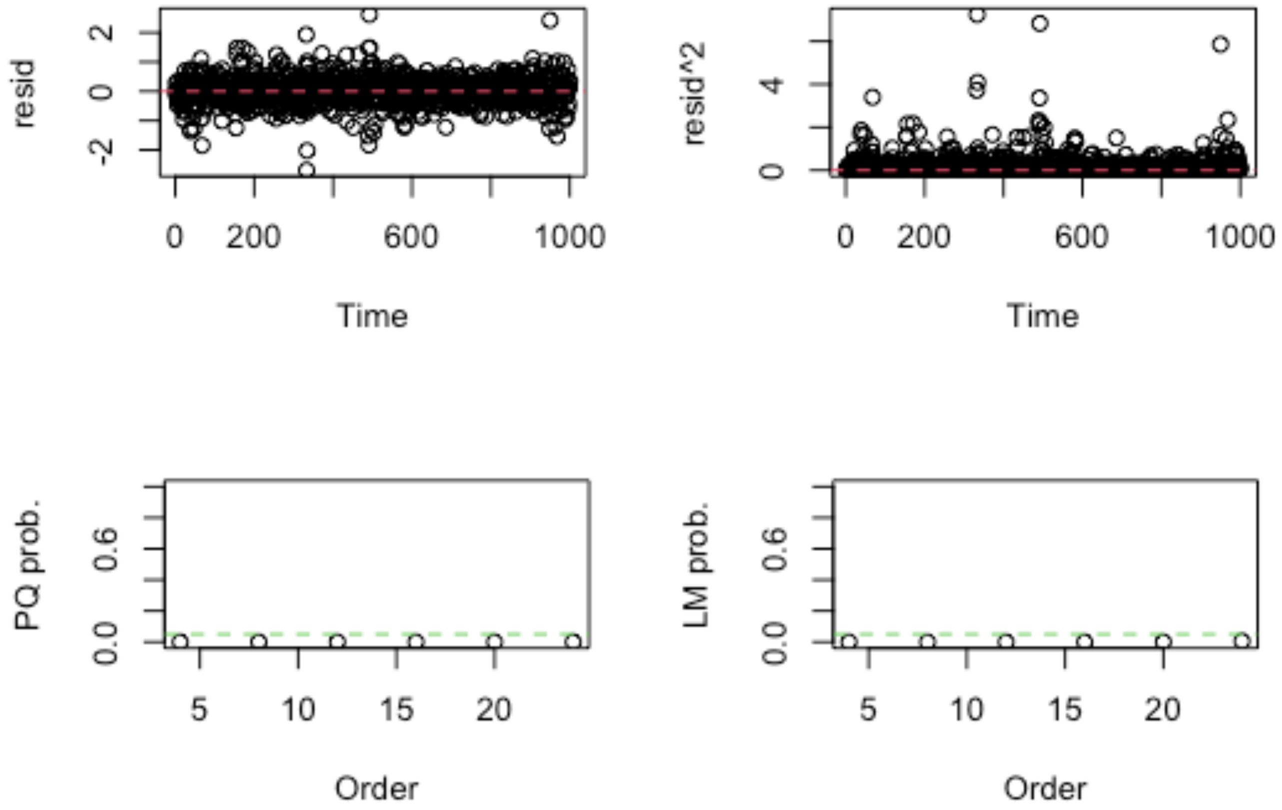
```
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic
```

Portmanteau-Q test:

	order	PQ	p.value
[1,]	4	356	0
[2,]	8	359	0
[3,]	12	360	0
[4,]	16	366	0
[5,]	20	371	0
[6,]	24	374	0

Lagrange-Multiplier test:

	order	LM	p.value
[1,]	4	290.2	0.00e+00
[2,]	8	143.3	0.00e+00
[3,]	12	94.3	2.44e-15
[4,]	16	68.7	7.53e-09
[5,]	20	53.3	4.28e-05
[6,]	24	44.1	5.15e-03

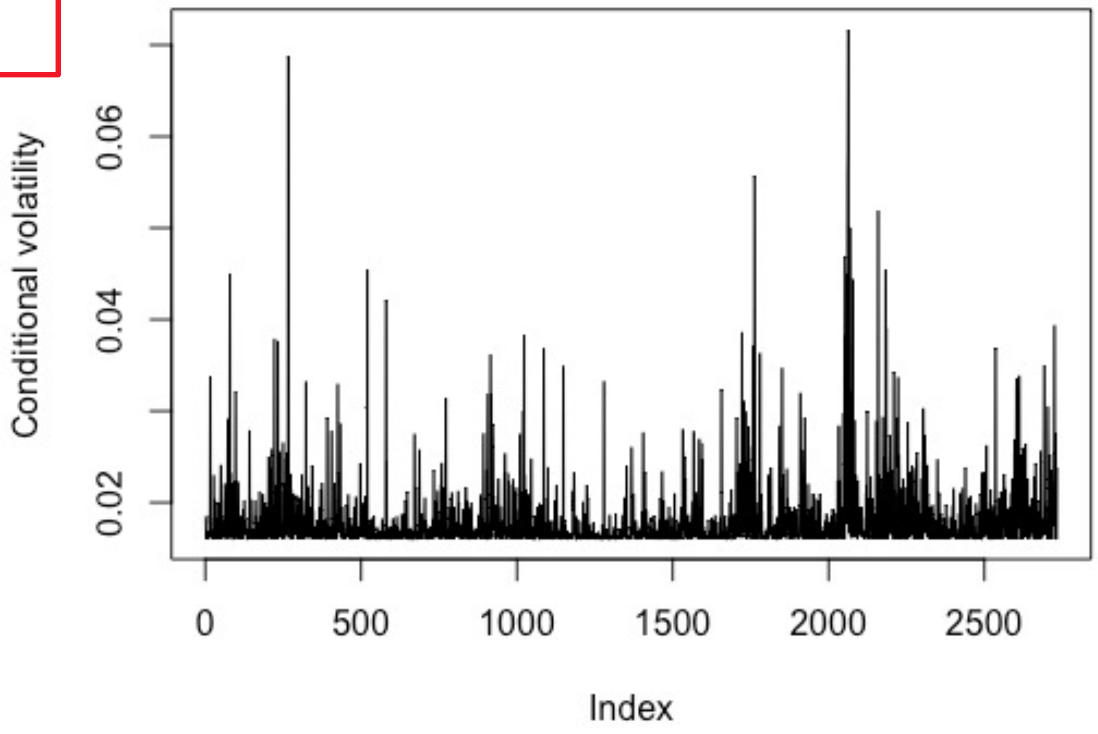


Fitting an ARCH model

```r
getSymbols('AAPL',src='yahoo', from="2012-01-01",periodicity = 'daily')
AAPL<-AAPL[,6]
r.AAPL<-diff(log(AAPL))
r.AAPL<-r.AAPL[2:length(AAPL)]
```

```r
install.packages('fGarch')
library(fGarch)
arch.fit <- garchFit(~garch(1,0), data = r.AAPL)
plot(arch.fit@sigma.t,type='l',ylab='Conditional volatility')
#plot(arch.fit@h.t) # conditional variance
```

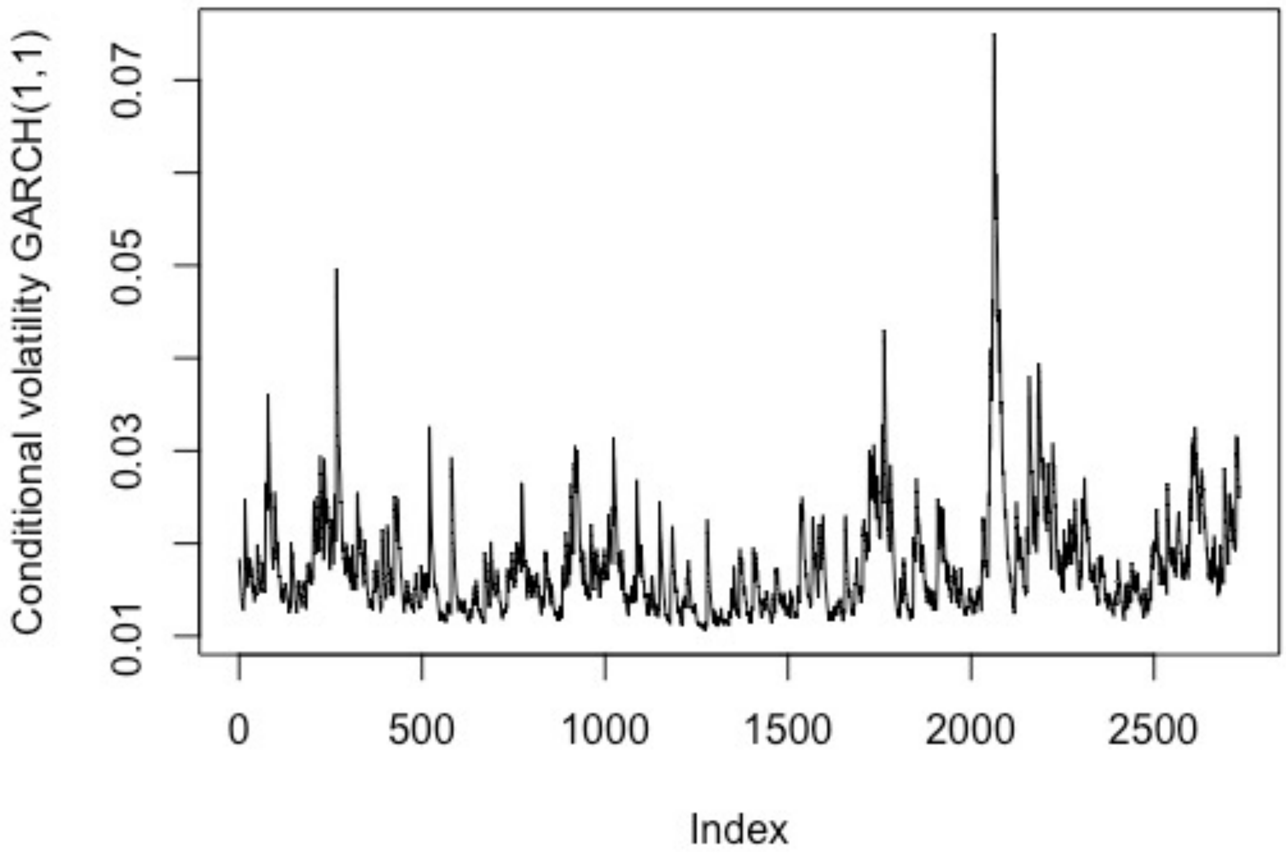
```r
#alternatively
garchfit<-garch(r.AAPL, order = c(0, 1), itmax = 200)
plot(garchfit$fitted.values[,1],type='l')
summary(garchfit)
```



Fitting a GARCH model

```
Garch.fit <- garchFit(~garch(1,1), data = r.AAPL)
plot(arch.fit@sigma.t,type='l',
     ylab='Conditional volatility GARCH(1,1)')
#plot(arch.fit@h.t) # conditional variance
```

```
#alternatively
garchfit2<-garch(r.AAPL, order = c(1, 1), itmax = 200)
plot(garchfit$fitted.values[,1],type='l')
summary(garchfit)
```



MUNI
ECON

GARCH extensions

- GARCH-in-Mean
- GJR GARCH
- EGARCH

Fitting GARCH extensions

– Selecting GARCH-type, coefficients and mean-equation:

```
install.packages('rugarch')
library(rugarch)

garchSpec <- ugarchspec(
  variance.model=list(model="sGARCH",
   garchOrder=c(1,1)),
  mean.model=list(armaOrder=c(0,0)),
  distribution.model="std")
```

– Fit the model:

```
garchFit <- ugarchfit(spec=garchSpec, data=r.AAPL)
coef(garchFit)
r_hat <- garchFit@fit$fitted.values
plot.ts(r_hat)
vol_hat <- ts(garchFit@fit$sigma)
plot.ts(hhat)
```


MUNI
ECON

Fitting GARCH extensions

- EGARCH: model='eGARCH'
- GJR-GARCH: model=''gjrGARCH'

```
# GARCH-in-mean
garchMod <- ugarchspec(
          variance.model=list(model="fGARCH",
                              garchOrder=c(1,1),
                              submodel="APARCH")
           mean.model=list(armaOrder=c(0,0),
                           include.mean=TRUE,
                           archm=TRUE,
                           archpow=2
                           ),
           distribution.model="std"
                           )
```


MUNI ECON