Arch-garch-modelling

Jai Vrat Singh

11/09/2018

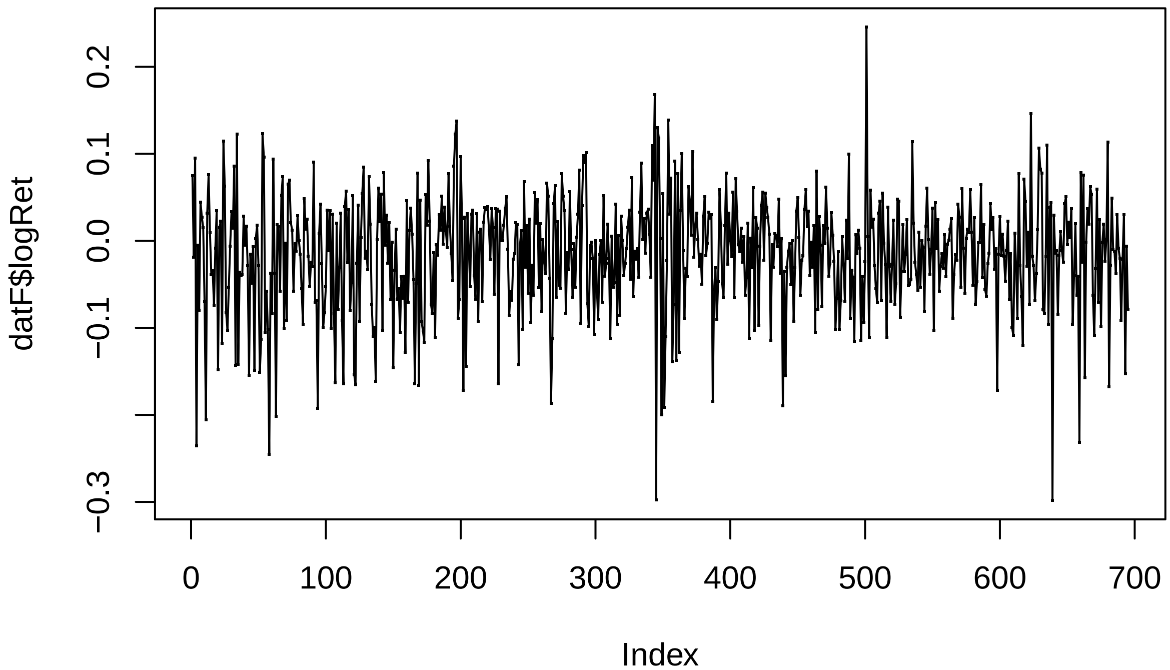
R Markdown

```
ret.what  <- list(date=numeric(), returns=numeric())
ret.widths <- c(12, 8)
strip.white <- c(TRUE, TRUE)

datF = scan(file = "http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts2/m-3m4603.txt",
             what = ret.what, strip.white = strip.white)
datF = as.data.frame(datF)


#Above are simple returns, Convert returns to logReturns
datF[["logRet"]] = log(1- datF$returns)

head(datF)
```

```
##       date  returns       logRet
## 1 19460228 -0.07792  0.075033258
## 2 19460330  0.01859 -0.018764966
## 3 19460430 -0.10000  0.095310180
## 4 19460531  0.20988 -0.235570446
## 5 19460628  0.00513 -0.005143204
## 6 19460731  0.07653 -0.079616965
```

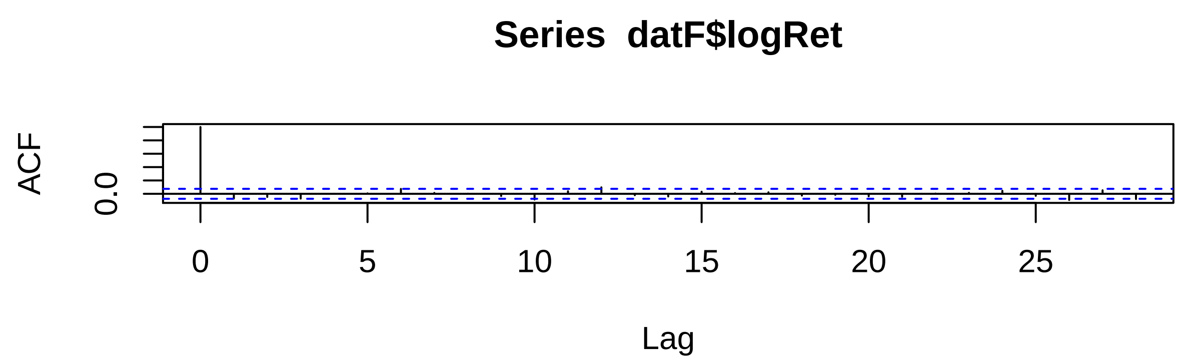
Evidence of ARCH effect

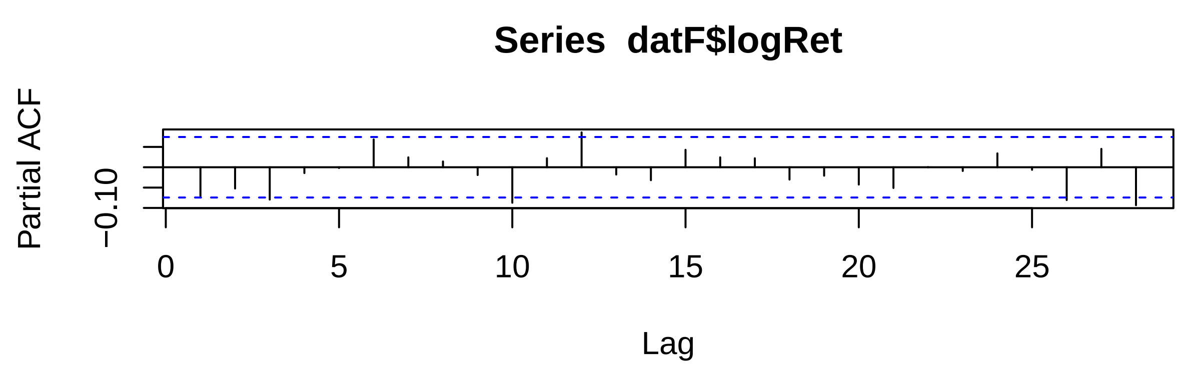
```
plot(datF$logRet, type = "o", pch = ".")
```



This plot reflects a pattern around the line logRet = 0, We need to look into ACF and PACF to get lagged correlations and its exact nature before the ARCH effect can be detected.

```
par(mfrow=c(2,1))
acf(datF$logRet)
acf(datF$logRet, type = "partial")
```





ACF till lags 12 => MA(12) PACF till 3, 12 => AR(3), AR(12)

Try these

```
#AR(12)
logret.fit = arima(x = datF$logRet, order = c(12,0,0))
logret.fit
```

```
##
## Call:
## arima(x = datF$logRet, order = c(12, 0, 0))
##
## Coefficients:
##           ar1      ar2      ar3      ar4     ar5     ar6     ar7     ar8
##       -0.0838  -0.0500  -0.0708  -0.0027  0.0041  0.0649  0.0181  0.0088
## s.e.   0.0378   0.0379   0.0380   0.0383  0.0383  0.0383  0.0383  0.0383
##           ar9     ar10    ar11    ar12  intercept
##       -0.0197  -0.0816  0.0307  0.0882    -0.0161
## s.e.   0.0383   0.0382  0.0384  0.0383     0.0023
##
## sigma^2 estimated as 0.004346:  log likelihood = 903.57,  aic = -1779.14
```

```
#aic = -1779.14

#AR(3)
logret.fit = arima(x = datF$logRet, order = c(3,0,0))
logret.fit
```

```
##
## Call:
## arima(x = datF$logRet, order = c(3, 0, 0))
##
## Coefficients:
##           ar1      ar2      ar3  intercept
##       -0.0794  -0.0584  -0.0808    -0.0161
## s.e.   0.0379   0.0379   0.0380     0.0021
##
## sigma^2 estimated as 0.004445:  log likelihood = 895.85,  aic = -1781.7
```

```
#aic = -1781.7

#MA(12)
logret.fit = arima(x = datF$logRet, order = c(0,0,12))
logret.fit
```

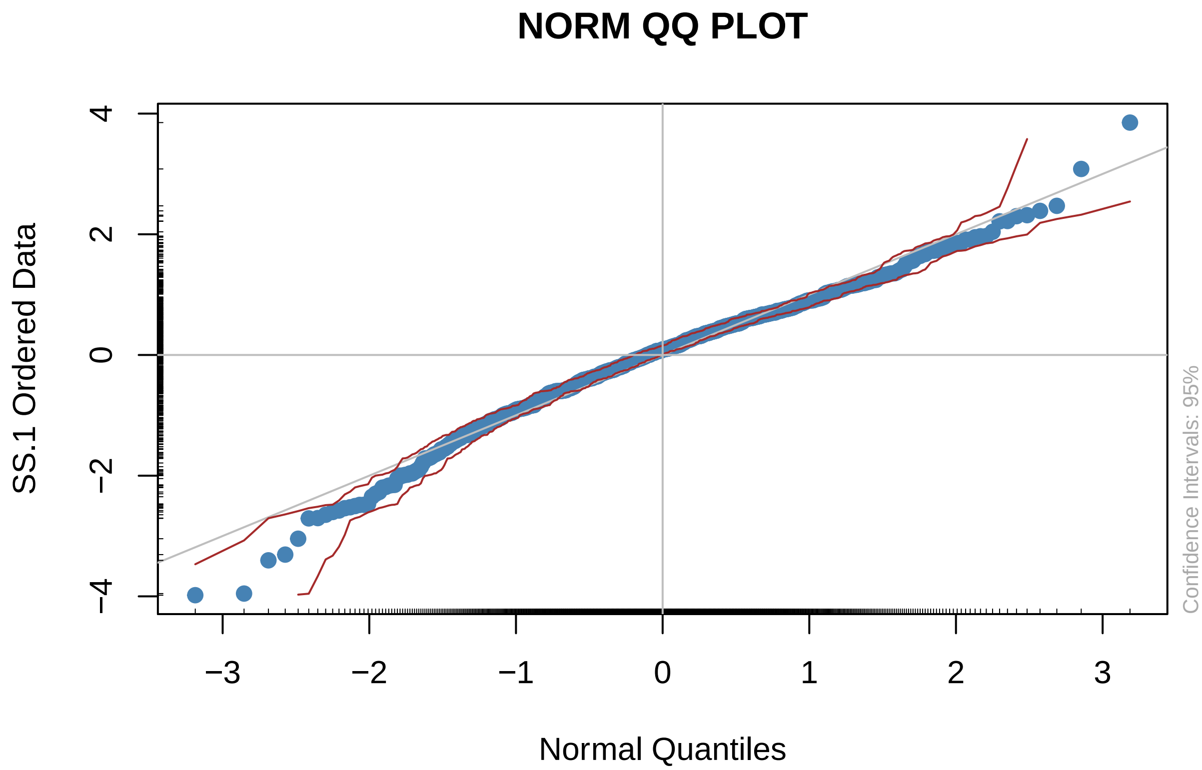
```
##
## Call:
## arima(x = datF$logRet, order = c(0, 0, 12))
##
## Coefficients:
##           ma1      ma2      ma3     ma4     ma5     ma6     ma7      ma8
##       -0.0821  -0.0402  -0.0691  0.0066  0.0121  0.0800  0.0165  -0.0001
## s.e.   0.0378   0.0378   0.0377  0.0382  0.0383  0.0386  0.0380   0.0390
##           ma9     ma10    ma11    ma12  intercept
##       -0.0316  -0.0909  0.0377  0.1035    -0.0161
## s.e.   0.0405   0.0396  0.0382  0.0376     0.0024
##
## sigma^2 estimated as 0.004338:  log likelihood = 904.24,  aic = -1780.48
```

```
#aic = -1780.48
```

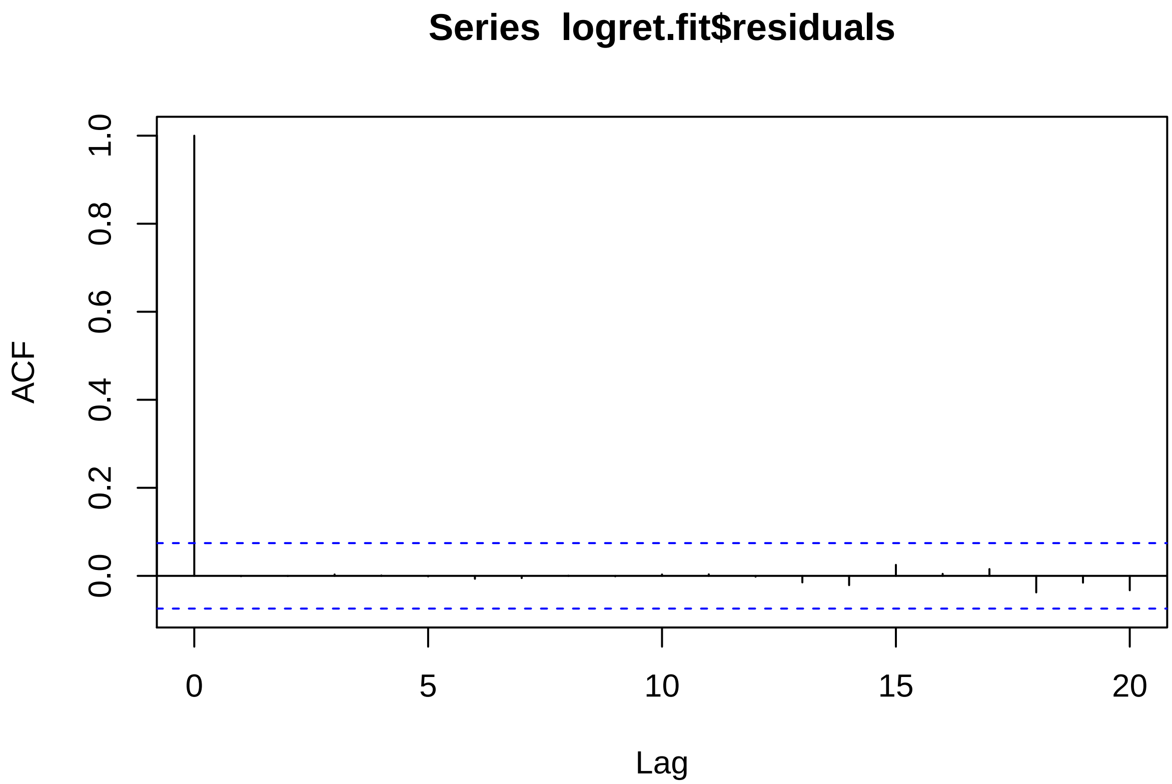
Lowest AIC is for AR(3) => seems it is best fit

Residue analysis of above fit

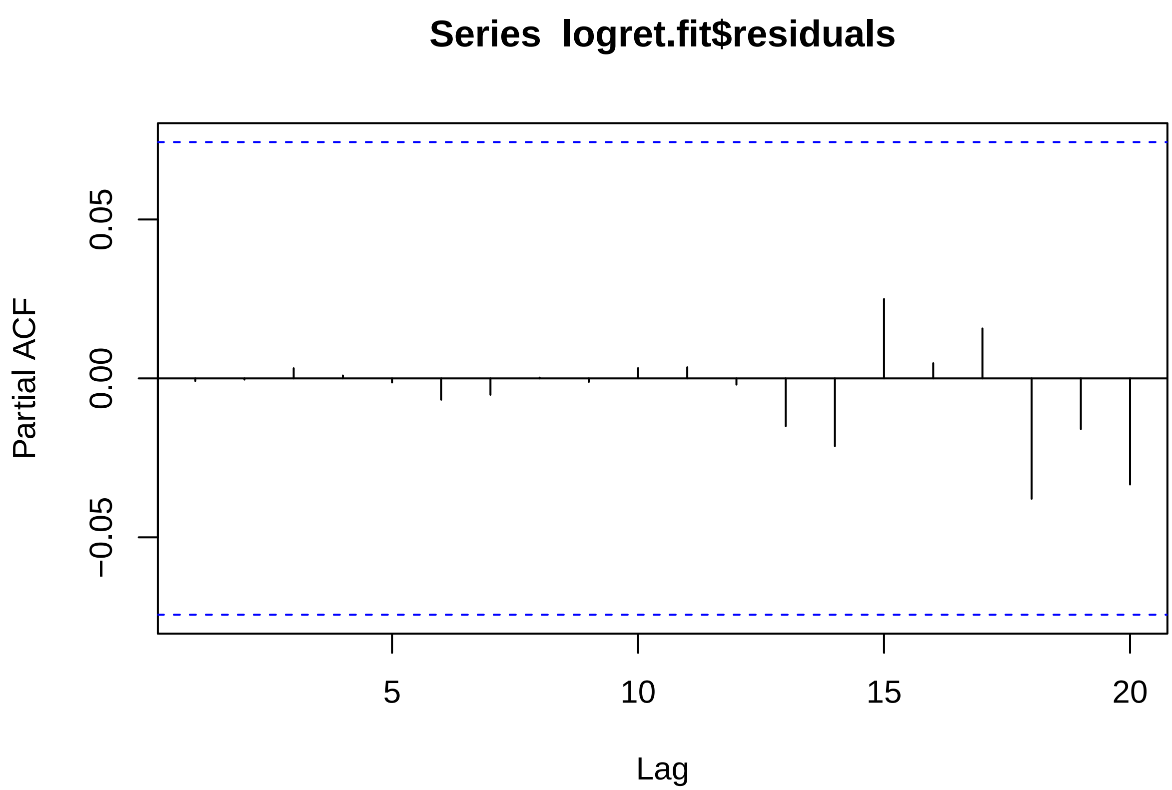
```
#logret.fit$residuals
qqnormPlot(logret.fit$residuals)
```




```
acf(logret.fit$residuals, lag.max = 20)
```




```r
acf(logret.fit$residuals, type = "partial", lag.max = 20)
```



Trying various models and Residual analysis, seems that MA(12) is the best fit model.

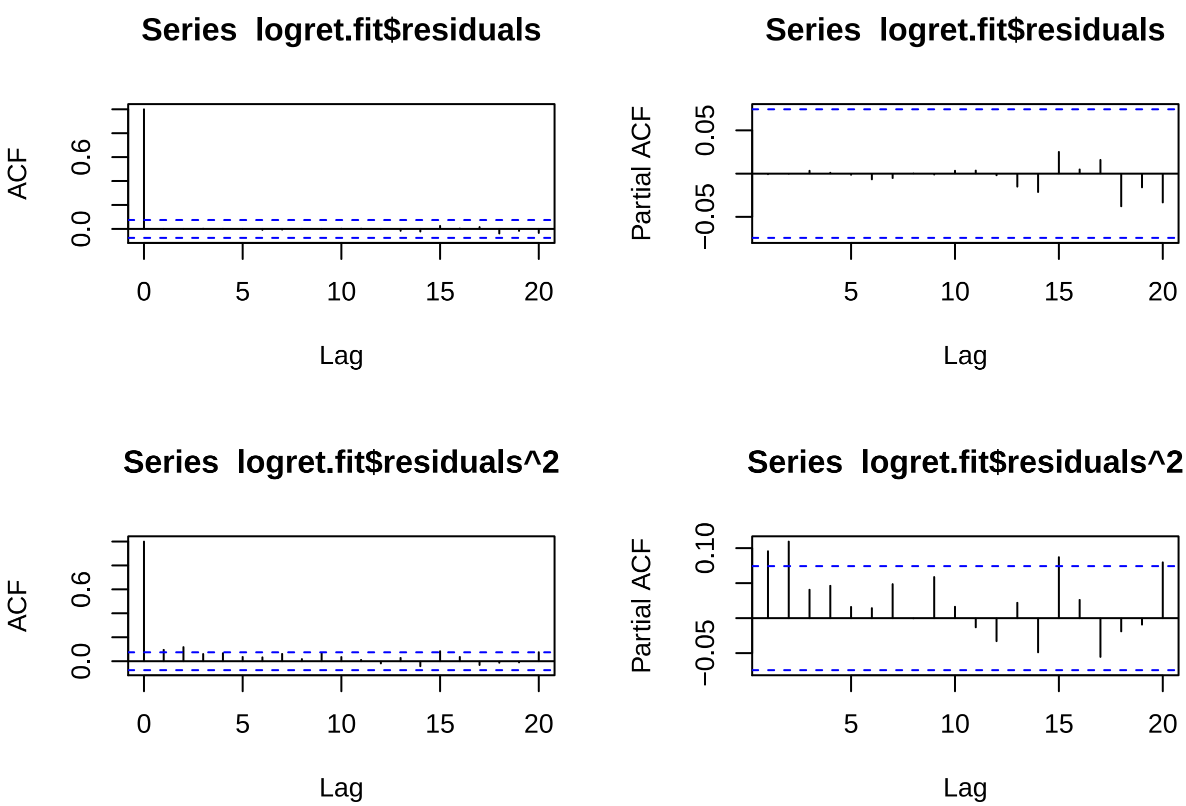
```
lags <- c(1:20)
sapply(lags, FUN = function(lag) {
                     test <- Box.test(logret.fit$residuals, lag = lag, type = c("Box-Pierce", "Ljung-Box
                     test$p.value
})
```

```
##  [1] 0.9833876 0.9997333 0.9998227 0.9999916 0.9999995 0.9999986 0.9999996
##  [8] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
## [15] 0.9999998 0.9999999 1.0000000 0.9999980 0.9999988 0.9999952
```

P-values are very high => the residual is white noise.

ARCH effect on residuals

```
par(mfrow=c(2,2))
acf(logret.fit$residuals, lag.max = 20)
pacf(logret.fit$residuals, lag.max = 20)
acf(logret.fit$residuals^2, lag.max = 20)
pacf(logret.fit$residuals^2, lag.max = 20)
```



ACF and PACF of residuals clearly show conditional heteroscedasticity

Let us do LB test on residuals^2

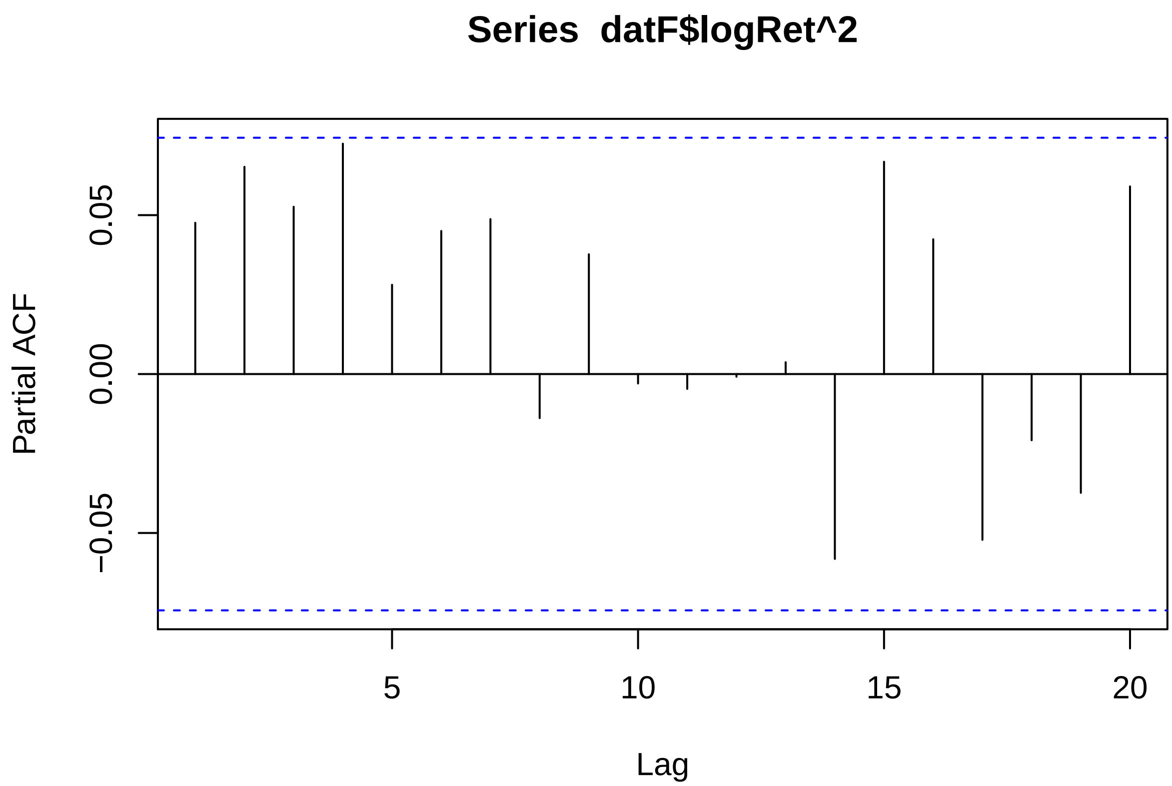
```
lags <- c(1:20)
sapply(lags, FUN = function(lag) {
                     test <- Box.test(logret.fit$residuals^2, lag = lag, type = c("Box-Pierce", "Ljung-B
                     test$p.value
})
```

```
##  [1] 0.0117694631 0.0003430990 0.0003504318 0.0002464178 0.0004326448
##  [6] 0.0007452725 0.0005524597 0.0010513383 0.0004738406 0.0006691216
## [11] 0.0012090605 0.0019856451 0.0028282093 0.0031847927 0.0010630363
## [16] 0.0013212098 0.0017427494 0.0026919060 0.0041035567 0.0019481185
```

P-values are less than 0.05 => means that shocks are not independent. There is ARCH effect in almost all the lags.

PACF of squared log returns

```r
pacf(datF$logRet^2, lag.max = 20)
```



Seems and ARCH(2) model can be fitted

$r_t = \mu + a_t$

$a_t = \sigma_t \epsilon_t$

$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2$

Fit the Model

```r
library(fGarch)
garch2.fit <- garchFit(~garch(2,0), data = datF$logRet, trace = FALSE)
garch2.fit
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
```

```
##  garchFit(formula = ~garch(2, 0), data = datF$logRet, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(2, 0)
## <environment: 0x7ff3ba2d6f08>
##  [data = datF$logRet]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1      alpha2
## -0.0169129   0.0035752   0.1125097   0.0985089
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -0.0169129   0.0024141   -7.006 2.46e-12 ***
## omega   0.0035752   0.0003063   11.672  < 2e-16 ***
## alpha1  0.1125097   0.0533135    2.110   0.0348 *
## alpha2  0.0985089   0.0511087    1.927   0.0539 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  898.8208    normalized:  1.293267
##
## Description:
##  Tue Sep 11 17:40:32 2018 by user:
```

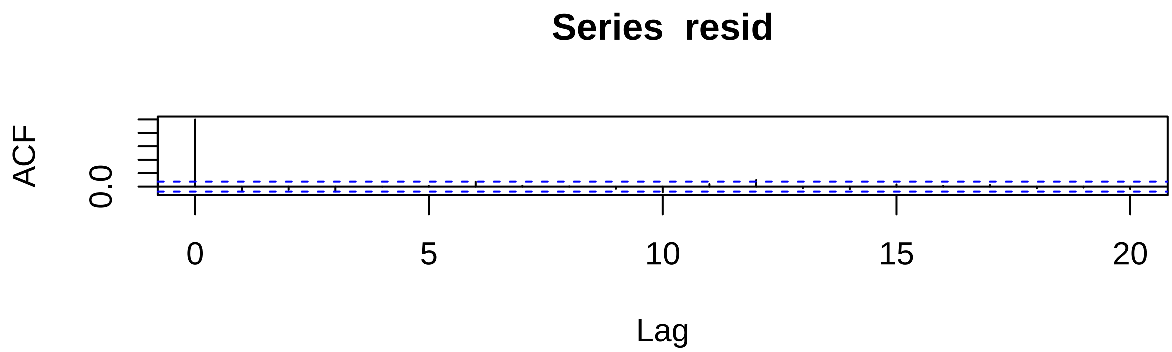
$\mu = -0.0169129$ $\alpha_0 = \omega = 0.0035752$ $\alpha_1 = 0.1125097$ $\alpha_2 = 0.0985089$

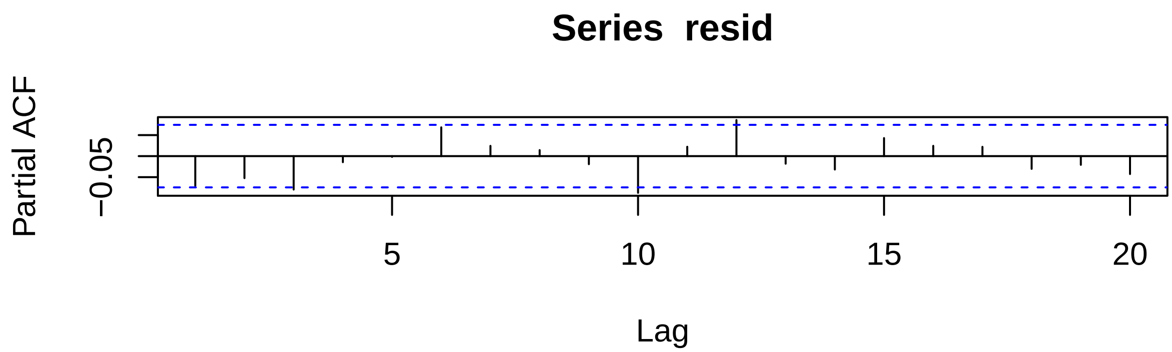
$r_t = -0.0169129 + a_t$

$\sigma_t^2 = 0.0035752 + 0.1125097 a_{t-1}^2 + 0.0985089 a_{t-2}^2$

$a_t = \sigma_t \epsilon_t$

```r
resid <- residuals(garch2.fit)
par(mfrow=c(2,1))
acf(resid, lag.max = 20)
pacf(resid, lag.max = 20)
```






```
qqnormPlot(resid)
```