Lab 7. Discrete volatility models

1

Let us first play around the simulated GARCH series. You need to install and load the libraries "fGarch" and "lattice".

```
> library(fGarch)   # for garch model simulation and estimation
> library(lattice)  # for qq-plot
>
> ?garchSim    # simulate Garch model
> ?garchSpec   # Garch model specification
>
> spec = garchSpec(model = list(omega = 1.5, alpha = c(0.4, 0.1), beta = 0.3))
> ## specify a GARCH(2,1) model
> garch21 = garchSim(spec, n.start = 500, n = 1000)
>
> par(mfrow=c(3,1))
> plot(garch21, main="Series garch21")
> acf(garch21, lag=30)
> acf(garch21^2, lag=30)
> ?qqmath       # qqplot against a non-Gaussian distribution
> qqmath(~ garch21, distribution = function(p) qt(p, df = 6.2), xlab="t(6.2)")
> ## Theoretical distribution is t(6.2); don't forget the ~
```

2

Now, let us fit a GARCH(1,1) model to the daily log-returns of the SP500 index.

```
> return500 <- 100*diff(sp500)
> par(mfrow=c(1,3))
> plot(return500)
> acf(return500)
> acf(return500^2)
> qqmath(~ return500, distribution = function(p) qt(p, df = 5), xlab="t(5)")
> qqmath(~ return500, distribution = function(p) qt(p, df = 6), xlab="t(6)")
> ?garchfit
```

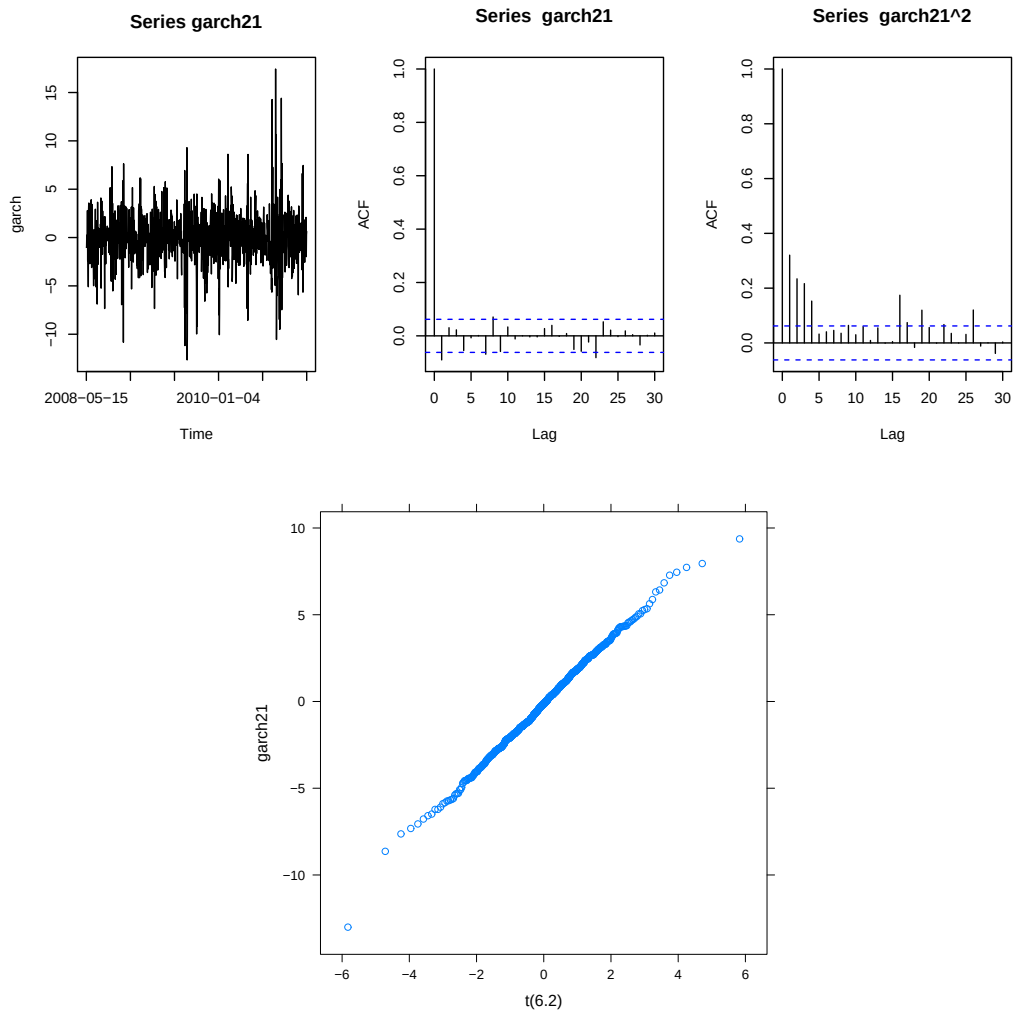

Figure 1: Garch simulation, the simulated series and $t_{6.2}$ show similar tail behavior.

```
> fit1 <- garchFit(formula = ~ garch(1, 1), data = return500, dist.est=T, cond.dist = "std" )
> summary(fit1)

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = return500, cond.dist = "std",
    dist.est = T)
```

```
Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 06c97cd8>
 [data = return500]

Conditional Distribution:
 std

Coefficient(s):
       mu      omega     alpha1      beta1      shape
0.0508101  0.0059345  0.0571273  0.9369567  7.2770316

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.050810    0.021033    2.416 0.015703 *
omega   0.005935    0.004777    1.242 0.214151
alpha1  0.057127    0.015569    3.669 0.000243 ***
beta1   0.936957    0.019473   48.115  < 2e-16 ***
shape   7.277032    1.698177    4.285 1.83e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Log Likelihood:
 -1122.209    normalized:  -1.123332

Description:
 Wed Feb 09 17:02:32 2011 by user: Lei Qi


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  348.7529  0
 Shapiro-Wilk Test  R    W      0.9770631 1.870651e-11
 Ljung-Box Test     R    Q(10)  12.74649  0.2381933
 Ljung-Box Test     R    Q(15)  15.23198  0.4348398
 Ljung-Box Test     R    Q(20)  18.22832  0.5723709
 Ljung-Box Test     R^2  Q(10)  7.267727  0.6999512
 Ljung-Box Test     R^2  Q(15)  11.13938  0.7426528
 Ljung-Box Test     R^2  Q(20)  11.95324  0.917676
 LM Arch Test       R    TR^2   7.890712  0.7936115
```

```
Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
2.256675 2.281233 2.256625 2.266009
```

3

You can fit other Garch-type models such as TGARCH and EGARCH, etc as follows.

```
> fitT <- garchFit(formula = ~ aparch(1, 1), data = return500, dist.est=T,
cond.dist = "std" )
> ## to fit TGarch, change the formula to "aparch"
>
> library(egarch)
> # egarch package that depends on fGarch
> fitE <- egarch(x=return500, order = c(1,1))
```