Modeling heteroskedasticity: GARCH modeling

Hedibert Freitas Lopes

5/28/2018

Glossary of ARCH models

Bollerslev wrote the article *Glossary to ARCH* (2010) which lists several families of ARCH models. You can find a technical report version of the paper here:

https://pdfs.semanticscholar.org/ea8e/a53721fbc28efec73e509259b00a9193ba12.pdf.

I reproduce here the first paragraph of his paper:

> Rob Engle's seminal Nobel Prize winning 1982 Econometrica article on the AutoRegressive Conditional Heteroskedastic (ARCH) class of models spurred a virtual "arms race" into the development of new and better procedures for modeling and forecasting timevarying financial market volatility. Some of the most influential of these early papers were collected in Engle (1995). Numerous surveys of the burgeoning ARCH literature also exist; e.g., Andersen and Bollerslev (1998), Andersen, Bollerslev, Christoffersen and Diebold (2006a), Bauwens, Laurent and Rombouts (2006), Bera and Higgins (1993), Bollerslev, Chou and Kroner (1992), Bollerslev, Engle and Nelson (1994), Degiannakis and Xekalaki (2004), Diebold (2004), Diebold and Lopez (1995), Engle (2001, 2004), Engle and Patton (2001), Pagan (1996), Palm (1996), and Shephard (1996). Moreover, ARCH models have now become standard textbook material in econometrics and finance as exemplified by, e.g., Alexander (2001, 2008), Brooks (2002), Campbell, Lo and MacKinlay (1997), Chan (2002), Christoffersen (2003), Enders (2004), Franses and van Dijk (2000), Gourieroux and Jasiak (2001), Hamilton (1994), Mills (1993), Poon (2005), Singleton (2006), Stock and Watson (2007), Tsay (2002), and Taylor (2004). So, why another survey type chapter?

Installing packages and creating functions

```r
#install.packages("fGarch")
#install.packages("rugarch")
library("fGarch")
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
## Loading required package: fBasics
```

```r
library("rugarch")
```

```
## Loading required package: parallel
```

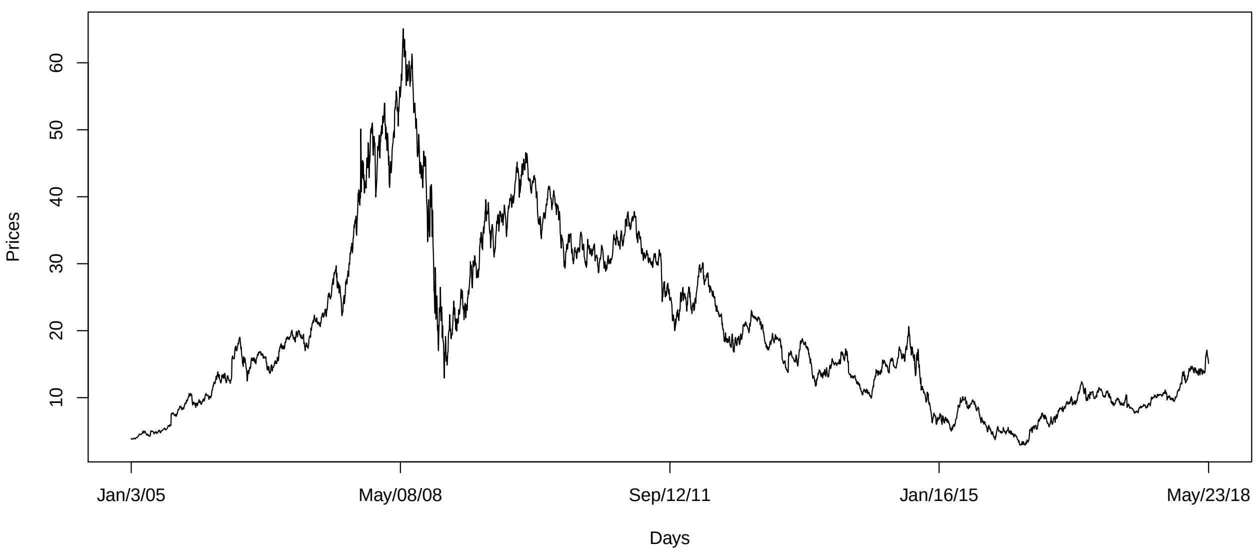
```
##
## Attaching package: 'rugarch'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

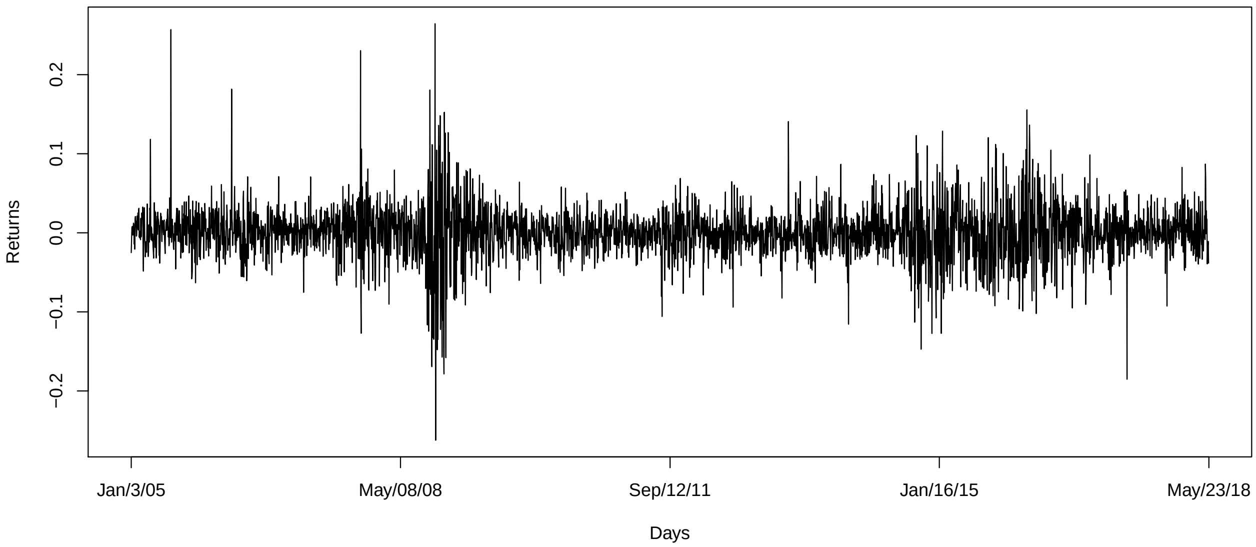
```
plot.sigt = function(y,sigt,model){
  limy = range(abs(y),-abs(y))
  par(mfrow=c(1,1))
  plot(abs(y),xlab="Days",ylab="Log-returns (-/+)",main="",type="h",axes=FALSE,ylim=limy)
  lines(-abs(y),type="h")
  axis(2);box();axis(1,at=ind,lab=date)
  lines(sigt,col=2)
  lines(-sigt,col=2)
  title(model)
}
```

Using Petrobras data as illustration

```
data = read.table("pbr.txt",header=TRUE)
n = nrow(data)
attach(data)
n = nrow(data)
y = diff(log(pbr))

ind = trunc(seq(1,n,length=5))
date = c("Jan/3/05","May/08/08","Sep/12/11","Jan/16/15","May/23/18")

par(mfrow=c(1,1))
plot(pbr,xlab="Days",ylab="Prices",axes=FALSE,type="l")
axis(2);axis(1,at=ind,lab=date);box()
```




```
par(mfrow=c(1,1))
plot(y,xlab="Days",ylab="Returns",axes=FALSE,type="l")
axis(2);axis(1,at=ind,lab=date);box()
```



AutoRegressive Conditional Heteroskedastic (ARCH)

For all models considered in this set of notes, we assume that

$$y_t = \sigma_t \varepsilon_t$$

where ε_t are iid D (Gaussian, Student's t, GED, etc), and the time-varying variances (or standard deviations) are modeled via one of the members of the large GARCH family of volatility models. The ARCH(1), for example, assumes that

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2,$$

with ε_0^2 either estimated or fixed. See Engle (1992).

```
fit.arch  = garchFit(~garch(1,0),data=y,trace=F,include.mean=FALSE)
fit.arch
```

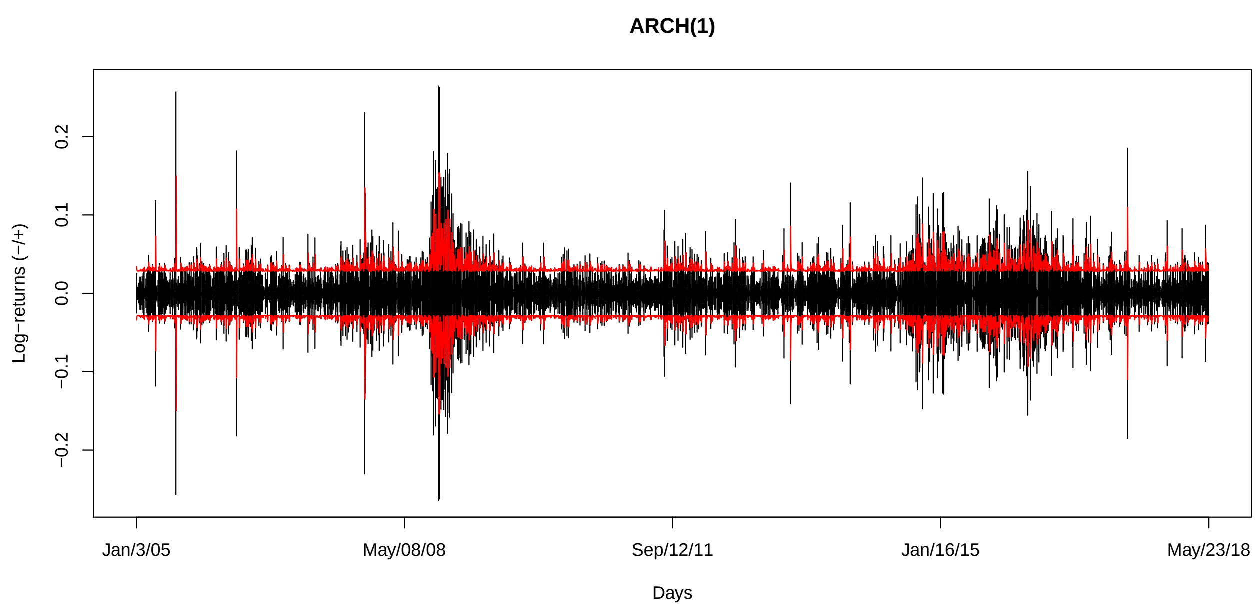
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 0), data = y, include.mean = FALSE,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 0)
## <environment: 0x7fafa0d83150>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##     omega     alpha1
## 0.0007969  0.3281214
##
```

```
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  7.969e-04   2.662e-05   29.935   <2e-16 ***
## alpha1 3.281e-01   3.425e-02    9.581   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  6805.826    normalized:  2.019533
##
## Description:
##  Mon May 28 18:12:29 2018 by user:
```

```
fit.arch@fit$matcoef
```

```
##            Estimate   Std. Error   t value Pr(>|t|)
## omega  0.0007968971 2.662075e-05 29.935186        0
## alpha1 0.3281214246 3.424883e-02  9.580516        0
```

```
plot.sigt(y,fit.arch@sigma.t,"ARCH(1)")
```



Generalized ARCH (GARCH)

The GARCH(1,1) model extends the ARCH(1) model:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

See Bollerslev (1986).

```
fit.garch = garchFit(~garch(1,1),data=y,trace=F,include.mean=F)
fit.garch
```

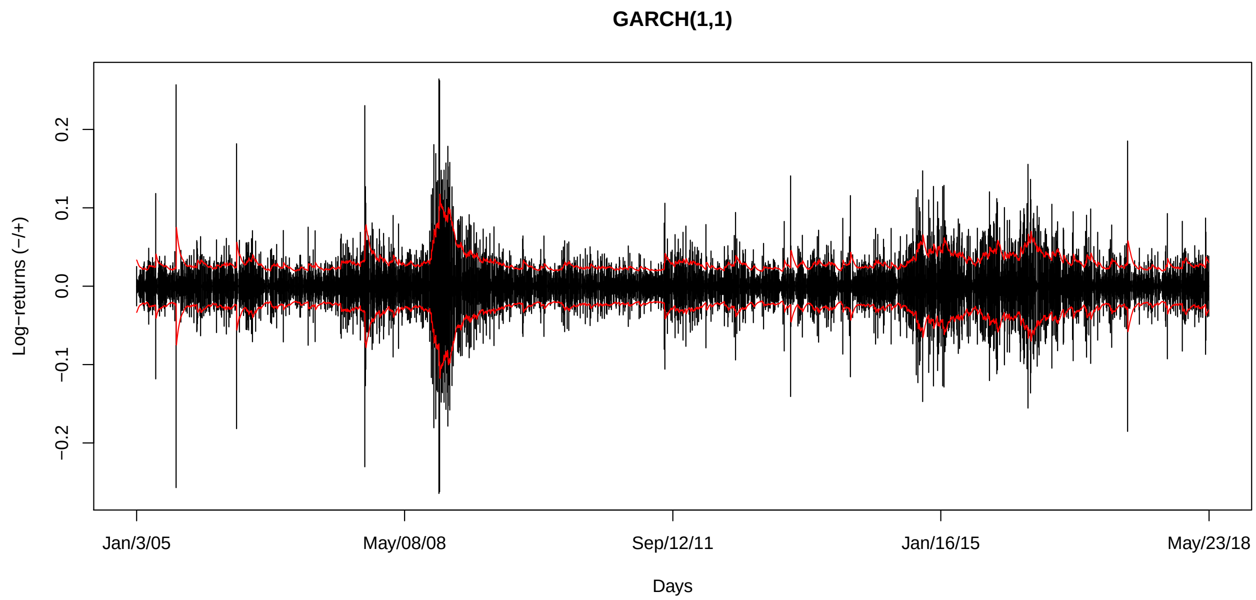
```
##
```

```
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, include.mean = F,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa3a22928>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1       beta1
## 2.7885e-05  7.7781e-02  8.9591e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  2.789e-05   5.282e-06    5.279 1.30e-07 ***
## alpha1 7.778e-02   1.131e-02    6.878 6.07e-12 ***
## beta1  8.959e-01   1.418e-02   63.190  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7106.437    normalized:  2.108735
##
## Description:
##  Mon May 28 18:12:30 2018 by user:
```

```
fit.garch@fit$matcoef
```

```
##            Estimate   Std. Error   t value     Pr(>|t|)
## omega  2.788546e-05 5.281898e-06  5.279439 1.295803e-07
## alpha1 7.778057e-02 1.130860e-02  6.878003 6.069811e-12
## beta1  8.959095e-01 1.417793e-02 63.190414 0.000000e+00
```

```
plot.sigt(y,fit.garch@sigma.t,"GARCH(1,1)")
```



Taylor-Schwert GARCH (TS-GARCH)

The TS-GARCH(1,1) models the time-varying standard deviation:

$$\sigma_t = \omega + \alpha_1 |\varepsilon_{t-1}| + \beta_1 \sigma_{t-1}.$$

See Taylor (1986) and Schwert (1989).

```
fit.tsgarch = garchFit(~garch(1,1),delta=1,data=y,trace=F,include.mean=F)
fit.tsgarch
```

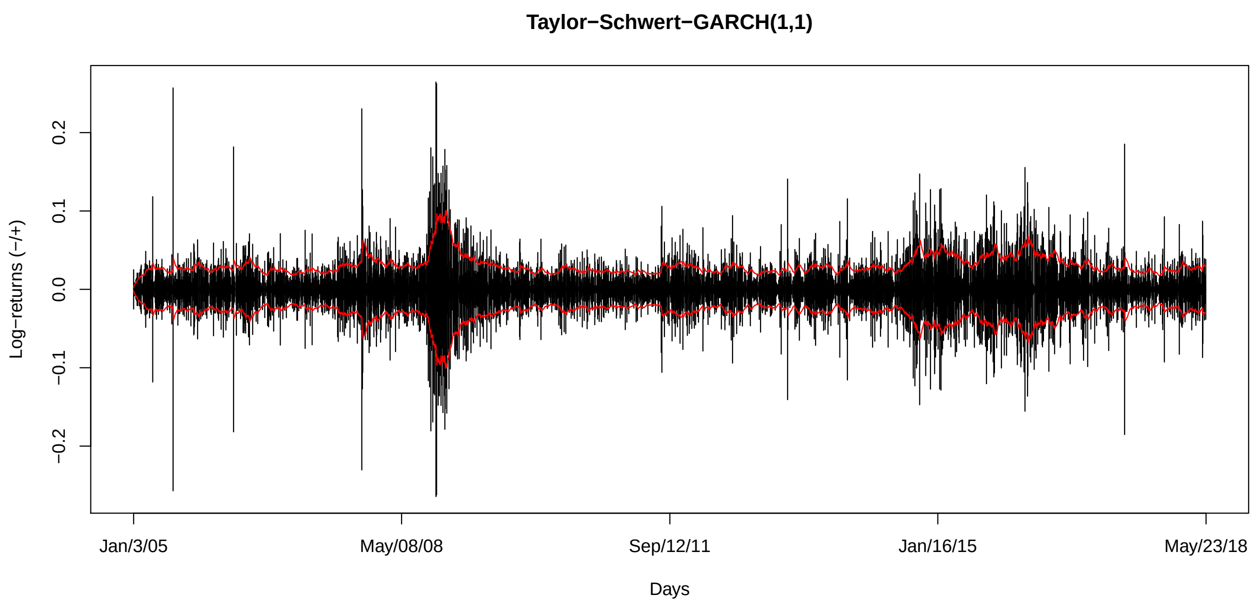
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, delta = 1, include.mean = F,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa19be0b0>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1       beta1
## 0.00062616  0.07315065  0.92469768
##
## Std. Errors:
##  based on Hessian
##
```

```
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  0.0006262   0.0001235    5.071 3.95e-07 ***
## alpha1 0.0731507   0.0083211    8.791  < 2e-16 ***
## beta1  0.9246977   0.0091703  100.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7032.705    normalized:  2.086856
##
## Description:
##  Mon May 28 18:12:31 2018 by user:
```

```
fit.tsgarch@fit$matcoef
```

```
##            Estimate   Std. Error    t value     Pr(>|t|)
## omega  0.0006261552 0.0001234745   5.071130 3.954603e-07
## alpha1 0.0731506516 0.0083210867   8.790997 0.000000e+00
## beta1  0.9246976786 0.0091703468 100.835628 0.000000e+00
```

```
plot.sigt(y,fit.tsgarch@sigma.t,"Taylor-Schwert-GARCH(1,1)")
```



Threshold GARCH (T-GARCH)

The T-GARCH(1,1) also models the time-varying standard deviation:

$$\sigma_t = \omega + \alpha_1|\varepsilon_{t-1}| + \gamma_1|\varepsilon_{t-1}|I(\varepsilon_{t-1} < 0) + \beta_1\sigma_{t-1}.$$

See Zakoian (1993).

```
fit.tgarch = garchFit(~garch(1,1),delta=1,leverage=T,data=y,trace=F,include.mean=F)
fit.tgarch
```

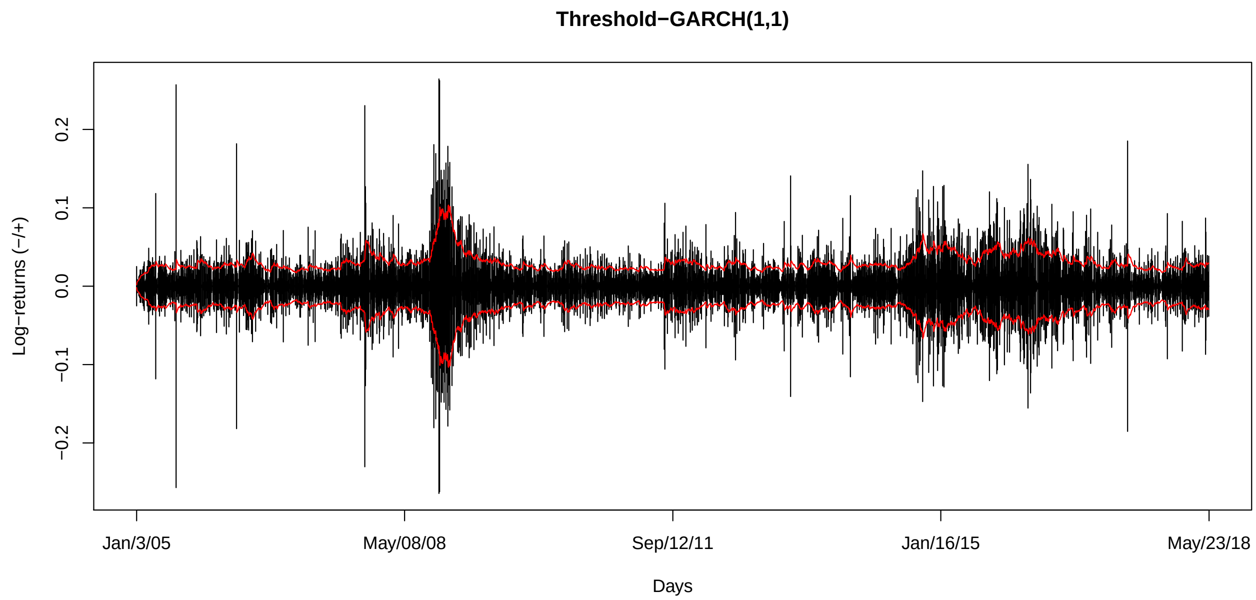
```
##
## Title:
```

```
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, delta = 1, include.mean = F,
##     leverage = T, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa2c032b8>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1      gamma1       beta1
## 0.00066195  0.06996082  0.23759067  0.92600900
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  0.0006620   0.0001261    5.249 1.53e-07 ***
## alpha1 0.0699608   0.0082676    8.462  < 2e-16 ***
## gamma1 0.2375907   0.0699703    3.396 0.000685 ***
## beta1  0.9260090   0.0092035  100.614  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7043.112    normalized:  2.089944
##
## Description:
##  Mon May 28 18:12:32 2018 by user:
```

```
fit.tgarch@fit$matcoef
```

```
##            Estimate   Std. Error    t value     Pr(>|t|)
## omega  0.0006619548 0.0001261216   5.248546 1.533045e-07
## alpha1 0.0699608160 0.0082676465   8.461999 0.000000e+00
## gamma1 0.2375906663 0.0699702622   3.395595 6.847963e-04
## beta1  0.9260089994 0.0092035347 100.614495 0.000000e+00
```

```
plot.sigt(y,fit.tgarch@sigma.t,"Threshold-GARCH(1,1)")
```



Glosten-Jaganathan-Runkle GARCH (GJR-GARCH)

The GJR-GARCH(1,1) model is similar to the T-GARCH(1,1), but it models time-varying variances instead:

$$\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \gamma_1\varepsilon_{t-1}^2 I(\varepsilon_{t-1} < 0) + \beta_1\sigma_{t-1}^2.$$

See Glosten, Jaganathan and Runkle (1993).

```
fit.gjrgarch = garchFit(~garch(1,1),delta=2,leverage=T,data=y,trace=F,include.mean=F)
fit.gjrgarch
```

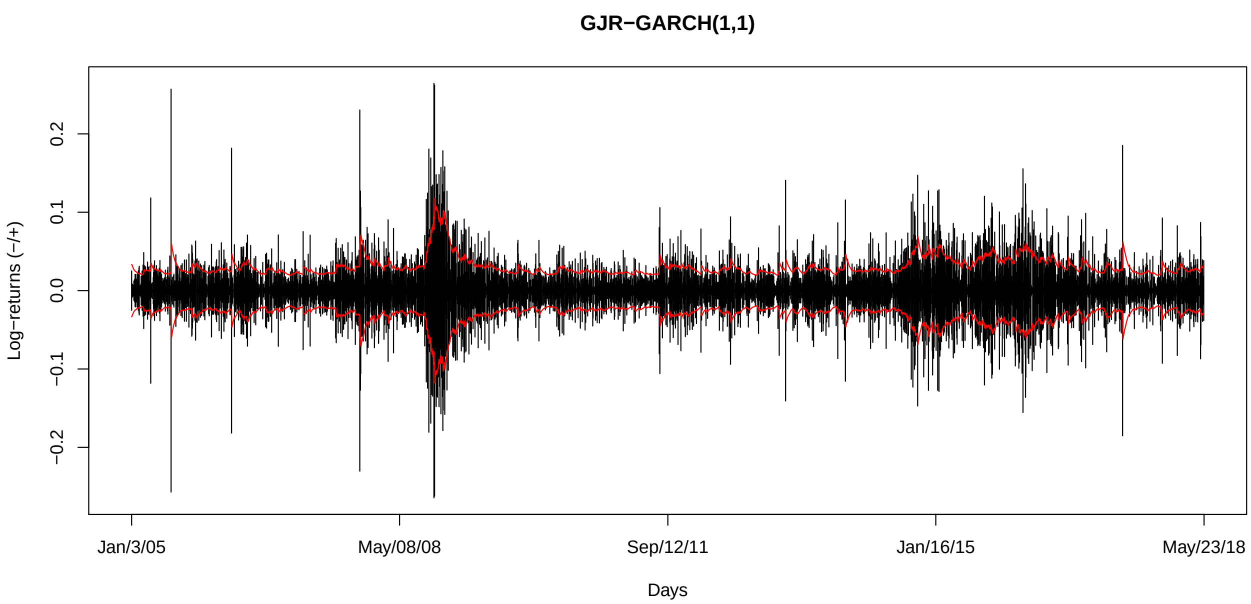
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, delta = 2, include.mean = F,
##     leverage = T, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa34c27e8>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##      omega      alpha1      gamma1       beta1
## 2.7905e-05  6.9168e-02  1.6071e-01  9.0225e-01
##
## Std. Errors:
##  based on Hessian
##
```

```
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  2.791e-05   5.218e-06    5.348 8.90e-08 ***
## alpha1 6.917e-02   1.112e-02    6.223 4.88e-10 ***
## gamma1 1.607e-01   5.858e-02    2.743  0.00608 **
## beta1  9.022e-01   1.398e-02   64.535  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7111.035    normalized:  2.110099
##
## Description:
##  Mon May 28 18:12:33 2018 by user:
```

```
fit.gjrgarch@fit$matcoef
```

```
##            Estimate   Std. Error   t value     Pr(>|t|)
## omega  2.790486e-05 5.217874e-06  5.347936 8.896313e-08
## alpha1 6.916832e-02 1.111524e-02  6.222836 4.882488e-10
## gamma1 1.607094e-01 5.858235e-02  2.743307 6.082376e-03
## beta1  9.022469e-01 1.398071e-02 64.535108 0.000000e+00
```

```
plot.sigt(y,fit.gjrgarch@sigma.t,"GJR-GARCH(1,1)")
```



Asymmetric Power GARCH (AP-GARCH)

The AP-GARCH(1,1) (aka APARCH(1,1)) models

$$\sigma_t^\delta = \omega + \alpha_1(|\varepsilon_{t-1}| - \gamma_1\varepsilon_{t-1})^\delta + \beta_1\sigma_{t-1}^\delta,$$

for $\delta > 0$ and $\gamma_1 \in (-1, 1)$. Th AP-GARCH model includes several models as special cases.

- ARCH - $\delta = 2$, $\gamma_1 = 0$ and $\beta_1 = 0$
- GARCH - $\delta = 2$ and $\gamma_1 = 0$

- TS-GARCH - $\delta = 1$ and $\gamma_1 = 0$
- T-GARCH - $\delta = 1$ and $\gamma_1 \in (0, 1)$
- GJR-GARCH - $\delta = 2$ and $\gamma_1 \in (0, 1)$

See Ding, Granger and Engle (1993).

```
fit.aparch = garchFit(~aparch(1,1),data=y,trace=F,include.mean=F)
fit.aparch
```

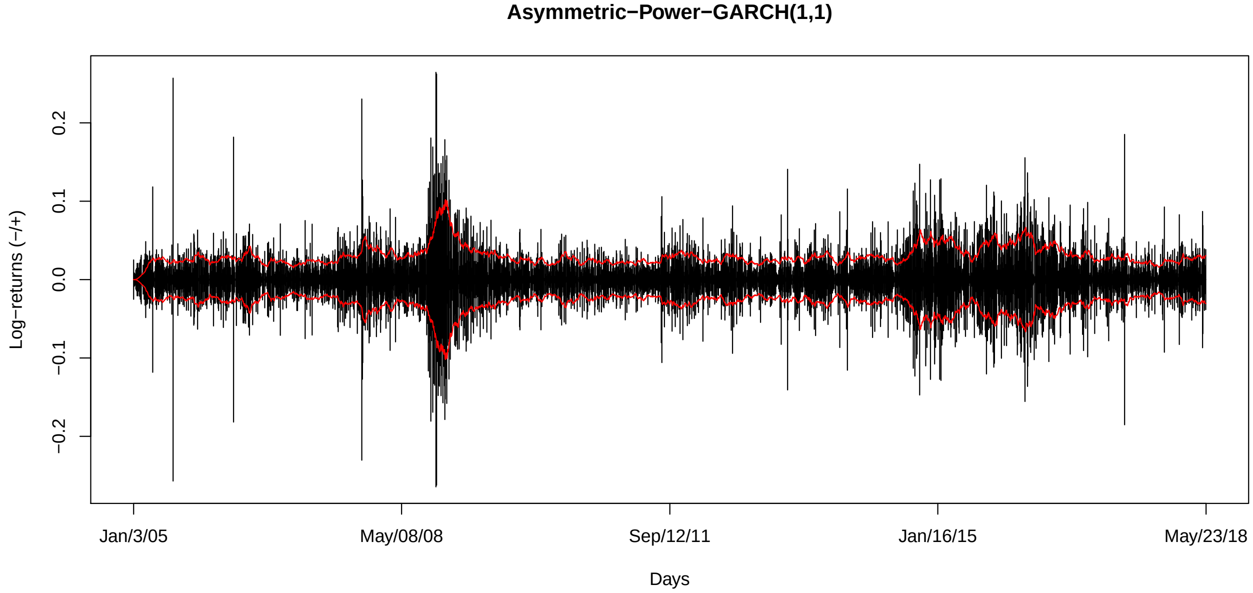
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~aparch(1, 1), data = y, include.mean = F,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ aparch(1, 1)
## <environment: 0x7fafa2bd09f8>
##  [data = y]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##     omega     alpha1     gamma1      beta1      delta
## 0.0050347  0.0465366  0.4426739  0.9482047  0.2815305
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  0.0050347   0.0007813    6.444 1.16e-10 ***
## alpha1 0.0465366   0.0034658   13.427  < 2e-16 ***
## gamma1 0.4426739   0.0866101    5.111 3.20e-07 ***
## beta1  0.9482047   0.0036798  257.680  < 2e-16 ***
## delta  0.2815305   0.0478599    5.882 4.04e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -4747.941    normalized:  -1.408885
##
## Description:
##  Mon May 28 18:12:34 2018 by user:
```

```
fit.aparch@fit$matcoef
```

```
##           Estimate  Std. Error    t value     Pr(>|t|)
## omega  0.005034733 0.000781298   6.444063 1.163172e-10
## alpha1 0.046536630 0.003465810  13.427344 0.000000e+00
## gamma1 0.442673896 0.086610079   5.111113 3.202664e-07
## beta1  0.948204748 0.003679782 257.679594 0.000000e+00
```

```
## delta  0.281530520 0.047859870   5.882392 4.043789e-09
```

```
plot.sigt(y,fit.aparch@sigma.t,"Asymmetric-Power-GARCH(1,1)")
```



Exponential GARCH model

The EGARCH(1,1) models

$$\log \sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1} + \gamma_1 |\varepsilon_{t-1}| + \beta_1 \log \sigma_{t-1}^2.$$

See Nelson (1991).

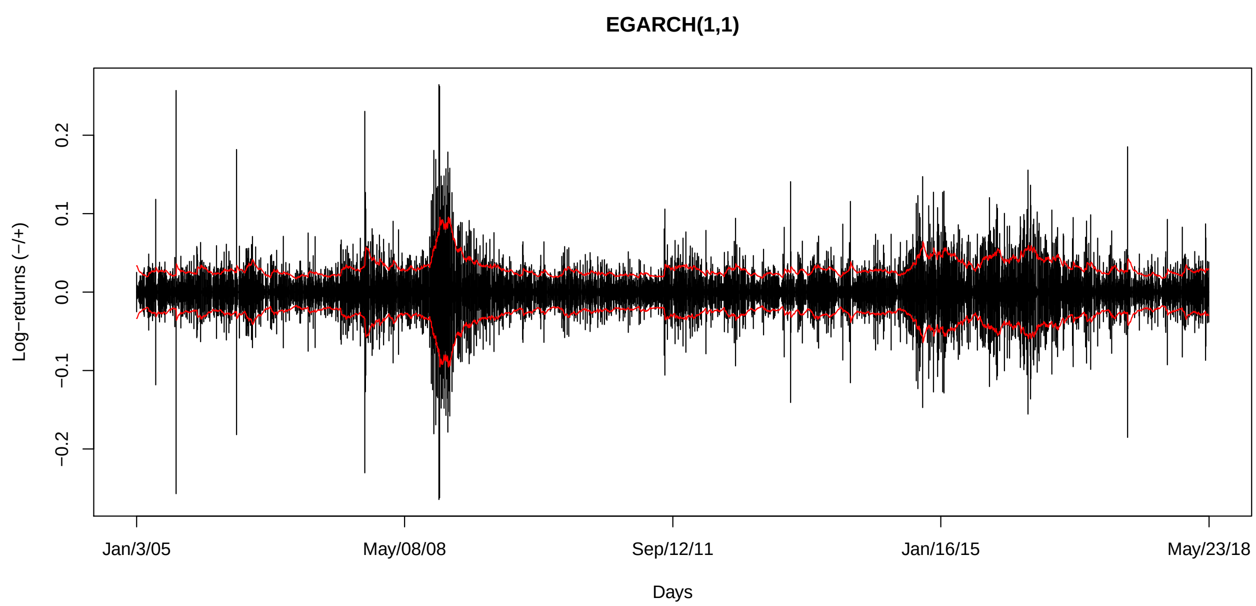
```
fit=ugarchfit(ugarchspec(mean.model=list(armaOrder=c(0,0),include.mean=TRUE,archm=FALSE,archpow=1,arfima

fit
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000461    0.000453    1.0183 0.308519
## omega  -0.113294    0.003542  -31.9850 0.000000
## alpha1 -0.025910    0.007523   -3.4441 0.000573
## beta1   0.982955    0.000439 2239.2495 0.000000
## gamma1  0.120714    0.002965   40.7194 0.000000
```

```
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.000461    0.000565   0.81655  0.41419
## omega  -0.113294    0.010166 -11.14394  0.00000
## alpha1 -0.025910    0.017729  -1.46148  0.14389
## beta1   0.982955    0.001151 853.88078  0.00000
## gamma1  0.120714    0.004767  25.32042  0.00000
##
## LogLikelihood : 7127.306
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.2269
## Bayes        -4.2178
## Shibata      -4.2269
## Hannan-Quinn -4.2236
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      3.022 0.08214
## Lag[2*(p+q)+(p+q)-1][2]     3.211 0.12305
## Lag[4*(p+q)+(p+q)-1][5]     3.686 0.29593
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.2086  0.6479
## Lag[2*(p+q)+(p+q)-1][5]    1.1270  0.8303
## Lag[4*(p+q)+(p+q)-1][9]    1.7846  0.9291
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.4398 0.500 2.000  0.5072
## ARCH Lag[5]    1.4683 1.440 1.667  0.6007
## ARCH Lag[7]    1.7202 2.315 1.543  0.7763
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.2439
## Individual Statistics:
## mu     0.8085
## omega  0.1656
## alpha1 0.0620
## beta1  0.1704
## gamma1 0.2118
##
## Asymptotic Critical Values (10% 5% 1%)
```

```
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias          1.14900 0.2506
## Negative Sign Bias 0.51278 0.6081
## Positive Sign Bias 0.01187 0.9905
## Joint Effect       1.76786 0.6220
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     85.15    2.375e-10
## 2    30     94.71    6.754e-09
## 3    40    105.16    5.489e-08
## 4    50    110.45    1.225e-06
##
##
## Elapsed time : 0.393569
```

```
egarch.sigma.t = as.vector(sigma(fit)[,1])

plot.sigt(y,egarch.sigma.t,"EGARCH(1,1)")
```



Non-Gaussian GARCH models

The R function garchFit has the "cond.dist" option for the specification of conditional distributions. The alternative erro distributions are "dnorm", "dged", "dstd", "dsnorm", "dsged" and "dsstd". Skewed ones: "dsnorm", "dsged", "dsstd".

GARCH(1,1) with Student-t with degrees of freedom (shape) estimated

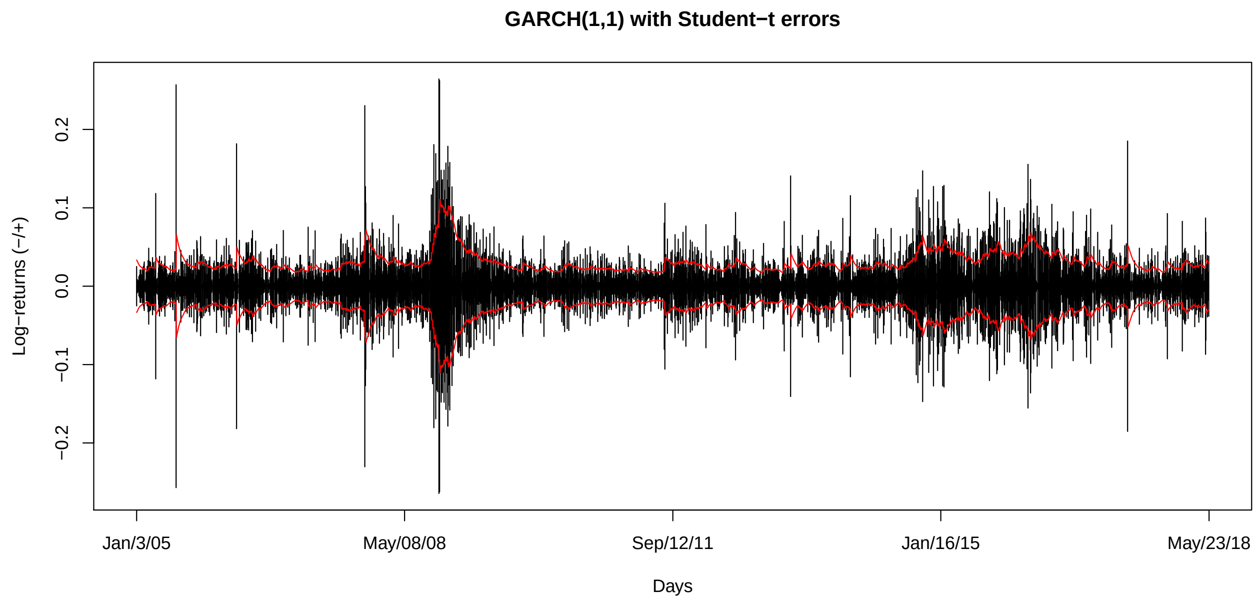
```
fit.garch.std = garchFit(~garch(1,1),cond.dist="std",data=y,trace=F,include.mean=F)
fit.garch.std
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, cond.dist = "std",
##     include.mean = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa2be4b48>
##  [data = y]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##      omega      alpha1       beta1       shape
## 0.00001096  0.05908638  0.93058069  6.16186223
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  1.096e-05   3.548e-06    3.089  0.00201 **
## alpha1 5.909e-02   1.071e-02    5.518 3.44e-08 ***
## beta1  9.306e-01   1.239e-02   75.122  < 2e-16 ***
## shape  6.162e+00   5.980e-01   10.304  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7266.331    normalized:  2.156181
##
## Description:
##  Mon May 28 18:12:37 2018 by user:
```

```
fit.garch.std@fit$matcoef
```

```
##            Estimate   Std. Error   t value     Pr(>|t|)
## omega  1.096023e-05 3.547863e-06  3.089249 2.006629e-03
## alpha1 5.908638e-02 1.070876e-02  5.517574 3.437113e-08
## beta1  9.305807e-01 1.238753e-02 75.122403 0.000000e+00
## shape  6.161862e+00 5.979790e-01 10.304479 0.000000e+00
```

```
plot.sigt(y,fit.garch.std@sigma.t,"GARCH(1,1) with Student-t errors")
```



GARCH(1,1) with skewed Student-t with degrees of freedom (shape) and skewness estimated

```
fit.garch.sstd = garchFit(~garch(1,1),cond.dist="sstd",data=y,trace=F,include.mean=F)
fit.garch.sstd
```

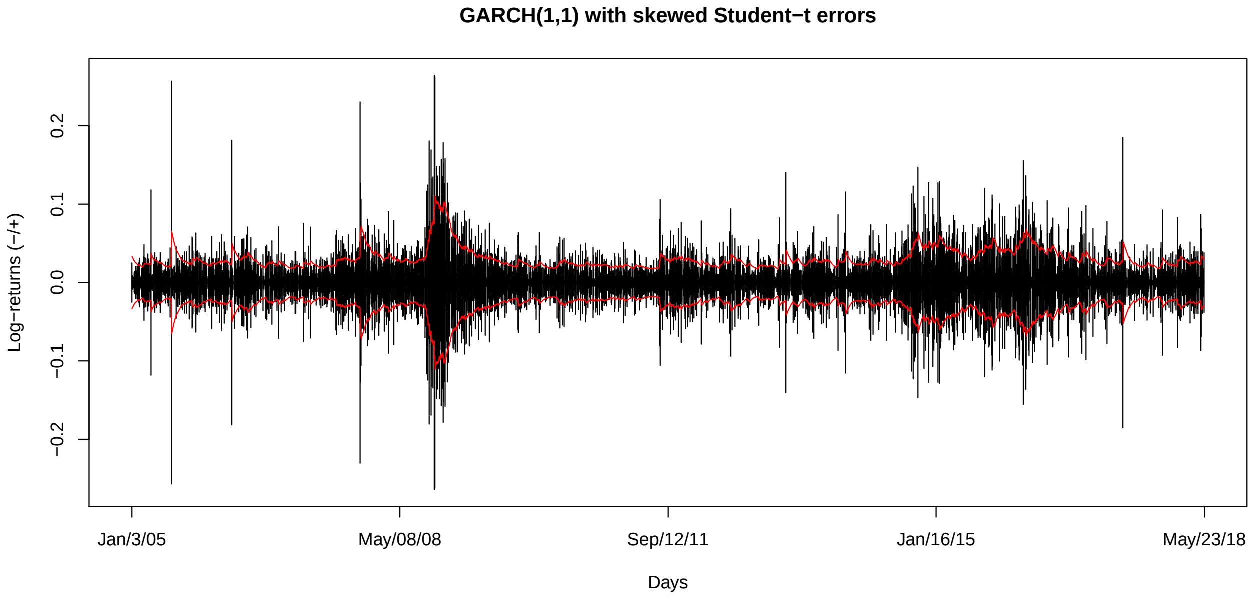
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, cond.dist = "sstd",
##     include.mean = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa34a90a8>
##  [data = y]
##
## Conditional Distribution:
##  sstd
##
## Coefficient(s):
##      omega      alpha1       beta1        skew       shape
## 1.0788e-05  5.8637e-02  9.3140e-01  9.5893e-01  6.0765e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  1.079e-05   3.518e-06    3.066  0.00217 **
```

```
## alpha1 5.864e-02   1.063e-02    5.514 3.52e-08 ***
## beta1  9.314e-01   1.226e-02   75.978  < 2e-16 ***
## skew   9.589e-01   2.161e-02   44.377  < 2e-16 ***
## shape  6.076e+00   5.863e-01   10.364  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7268.05    normalized:  2.156691
##
## Description:
##  Mon May 28 18:12:39 2018 by user:
```

```
fit.garch.sstd@fit$matcoef
```

```
##           Estimate   Std. Error   t value     Pr(>|t|)
## omega  0.000010788 3.518380e-06  3.066183 2.168102e-03
## alpha1 0.058636640 1.063494e-02  5.513586 3.515958e-08
## beta1  0.931395242 1.225873e-02 75.978147 0.000000e+00
## skew   0.958925907 2.160849e-02 44.377270 0.000000e+00
## shape  6.076498694 5.862928e-01 10.364272 0.000000e+00
```

```
plot.sigt(y,fit.garch.sstd@sigma.t,"GARCH(1,1) with skewed Student-t errors")
```



GARCH(1,1) with Student-t with degrees of freedom (shape) fixed at 3

```
fit.garch.std3 = garchFit(~garch(1,1),cond.dist="std",shape=3,include.shape=F,data=y,trace=F,include.mea
fit.garch.std3
```

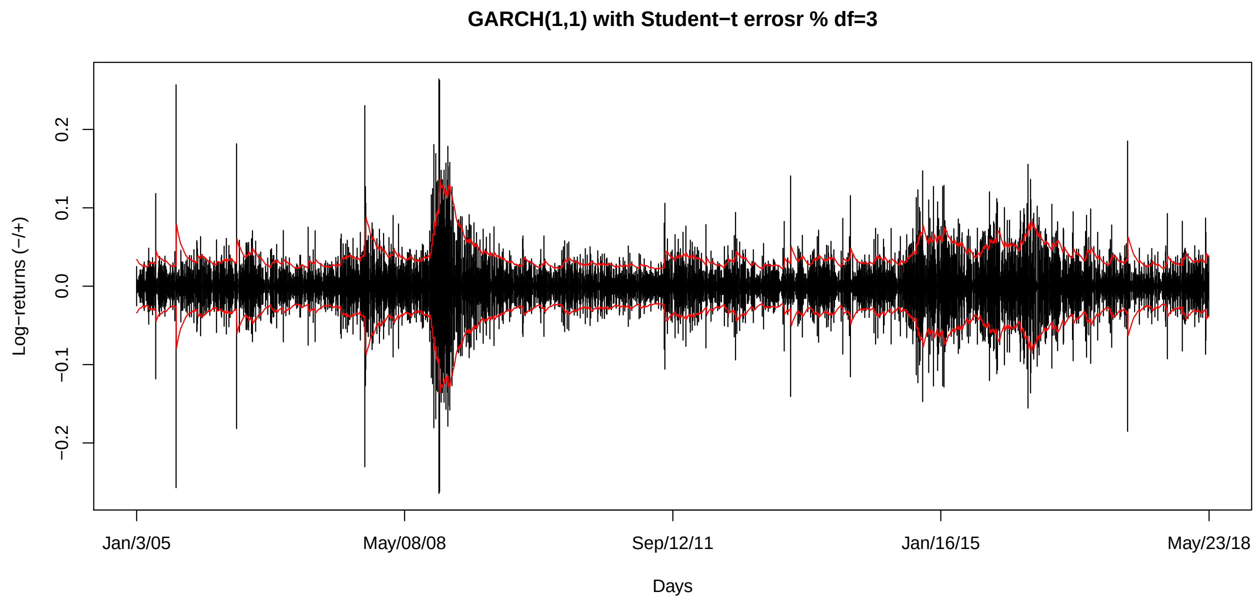
```
##
## Title:
##  GARCH Modelling
##
```

```
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, shape = 3, cond.dist = "std",
##     include.mean = F, include.shape = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa04f2c08>
##  [data = y]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##      omega      alpha1       beta1
## 1.4511e-05  8.5175e-02  9.3790e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  1.451e-05   5.610e-06    2.587  0.00969 **
## alpha1 8.518e-02   1.783e-02    4.778 1.77e-06 ***
## beta1  9.379e-01   1.276e-02   73.506  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7226.447    normalized:  2.144346
##
## Description:
##  Mon May 28 18:12:39 2018 by user:
```

```
fit.garch.std3@fit$matcoef
```

```
##            Estimate   Std. Error   t value     Pr(>|t|)
## omega  0.0000145112 5.610119e-06  2.586611 9.692489e-03
## alpha1 0.0851754798 1.782509e-02  4.778404 1.766919e-06
## beta1  0.9378982856 1.275944e-02 73.506210 0.000000e+00
```

```
plot.sigt(y,fit.garch.std3@sigma.t,"GARCH(1,1) with Student-t errosr % df=3")
```



GARCH(1,1) with Laplace (a GED with shape fixed at 1)

```
fit.garch.ged = garchFit(~garch(1,1),cond.dist="ged",shape=1,include.shape=F,data=y,trace=F,include.mea
fit.garch.ged
```

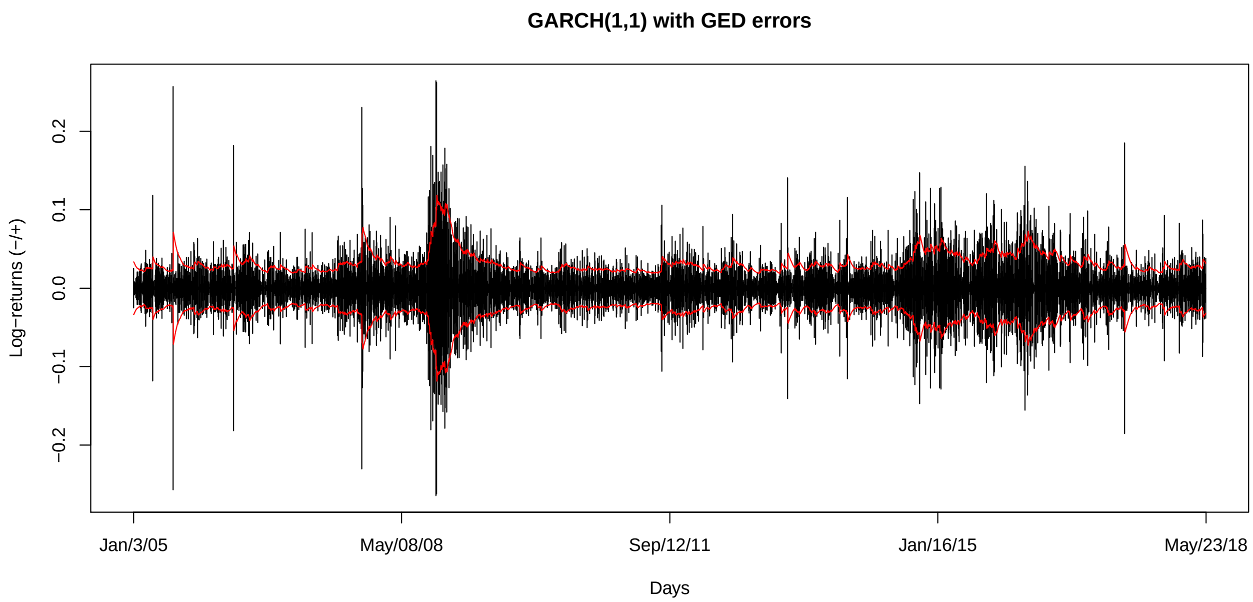
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = y, shape = 1, cond.dist = "ged",
##     include.mean = F, include.shape = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fafa26409a8>
##  [data = y]
##
## Conditional Distribution:
##  ged
##
## Coefficient(s):
##      omega      alpha1       beta1
## 1.5714e-05  6.9774e-02  9.2461e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega  1.571e-05   5.404e-06    2.908  0.00364 **
## alpha1 6.977e-02   1.479e-02    4.718 2.39e-06 ***
## beta1  9.246e-01   1.564e-02   59.110  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  7202.951    normalized:  2.137374
##
## Description:
##  Mon May 28 18:12:40 2018 by user:
```

```
fit.garch.ged@fit$matcoef
```

```
##             Estimate   Std. Error   t value     Pr(>|t|)
## omega  1.571374e-05 5.403615e-06  2.908004 3.637434e-03
## alpha1 6.977373e-02 1.479031e-02  4.717532 2.387234e-06
## beta1  9.246080e-01 1.564207e-02 59.110324 0.000000e+00
```

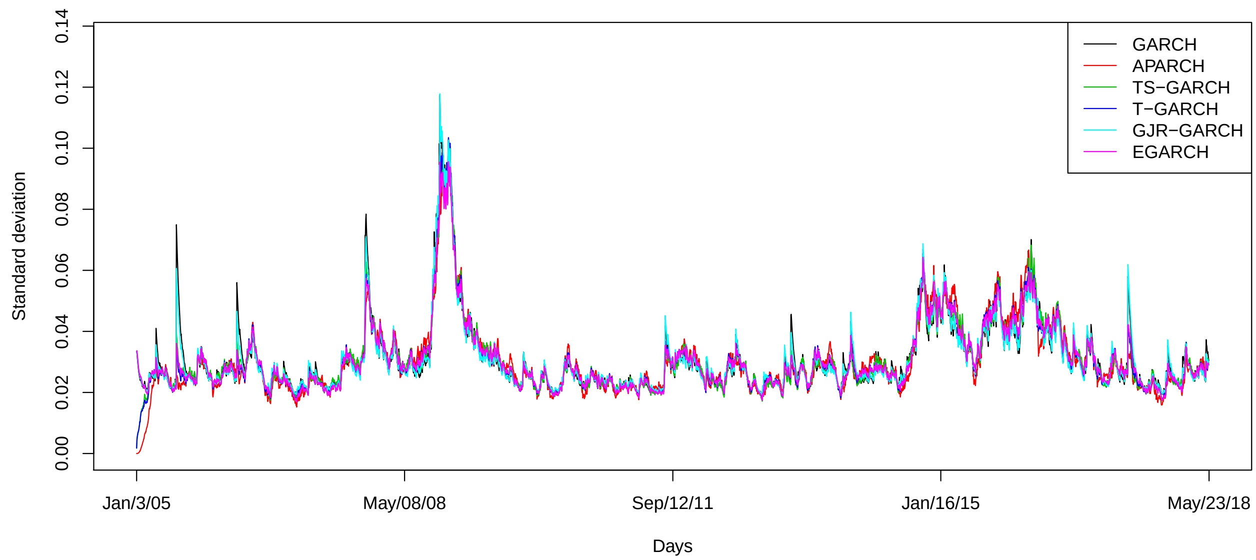
```
plot.sigt(y,fit.garch.ged@sigma.t,"GARCH(1,1) with GED errors")
```



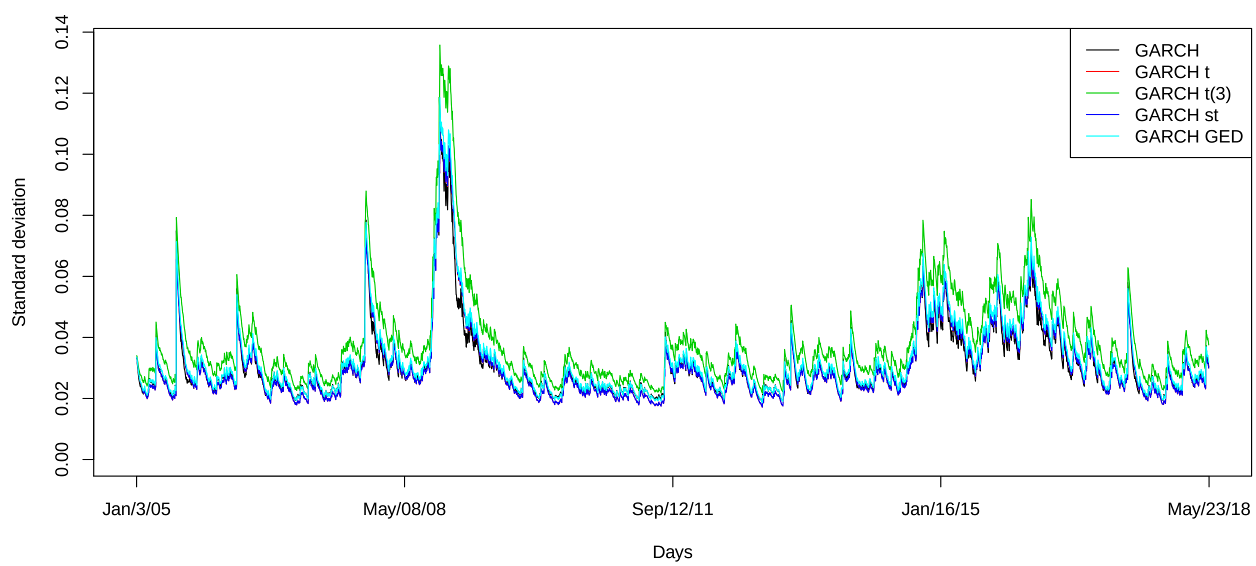
Comparing the estimates of σ_t

```
sigma.t = cbind(fit.garch@sigma.t,
                fit.aparch@sigma.t,
                fit.tsgarch@sigma.t,
                fit.tgarch@sigma.t,
                fit.gjrgarch@sigma.t,
                egarch.sigma.t,
                fit.garch.std@sigma.t,
                fit.garch.std3@sigma.t,
                fit.garch.sstd@sigma.t,
                fit.garch.ged@sigma.t)

limy=range(sigma.t)

par(mfrow=c(1,1))
```

```
plot(sigma.t[,1],xlab="Days",ylab="Standard deviation",main="",type="l",ylim=limy,axes=FALSE)
axis(2);box();axis(1,at=ind,lab=date)
for (i in 2:6)
  lines(sigma.t[,i],col=i)
  legend("topright",col=1:6,lty=1,
         legend=c("GARCH","APARCH","TS-GARCH","T-GARCH","GJR-GARCH","EGARCH"))
```




```
par(mfrow=c(1,1))
plot(sigma.t[,1],xlab="Days",ylab="Standard deviation",main="",type="l",ylim=limy,axes=FALSE)
axis(2);box();axis(1,at=ind,lab=date)
lines(sigma.t[,7],col=2)
lines(sigma.t[,8],col=3)
lines(sigma.t[,9],col=4)
lines(sigma.t[,10],col=5)
legend("topright",legend=c("GARCH","GARCH t","GARCH t(3)","GARCH st","GARCH GED"),col=1:5,lty=1)
```



Cited references

- Bollerslev (1986) Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, 307-327.
- Bollerslev (2010) Glossary to ARCH (GARCH) In *Volatility and Time Series Econometrics: Essays in Honor of Robert F. Engle*. Edited by Bollerslev, Russell and Watson. Chapter 8, pages 137-163.
- Ding, Granger and Engle (1993) A Long Memory Property of Stock Market Returns and a New Model. *Journal of Empirical Finance*, 1, 83-106.
- Engle (1982) Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation. *Econometrica*, 50, 987-1008.
- Glosten, Jagannathan and Runkle (1993) On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks, *Journal of Finance*, 48, 1779-1801.
- Nelson (1991) Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, 59, 347-370.
- Schwert (1989) Why Does Stock Market Volatility Change Over Time? *Journal of Finance*, 44, 1115-1153.
- Taylor (1986) *Modeling Financial Time Series*. Chichester, UK: John Wiley and Sons.
- Zakoian (1994) Threshold Heteroskedastic Models. *Journal of Economic Dynamics and Control*, 18, 931-955.