Lecture 10: Advanced Topics in GARCH Time Series Analysis
Statistics 509 – Winter 2022
Reference: Chapter 14 of Ruppert/Matteson

Brian Thelen
258 West Hall
bjthelen@umich.edu

Overview

- Review of GARCH
- More on ARCH/GARCH
  - Forecasting volatility - variance prediction
  - Half-life on volatility
- ARMA-GARCH Model and Estimation
  - Data examples
  - R-functions for estimation and variance prediction
  - Diagnostics
- APARCH Model and Estimation
  - Data examples
  - R-functions for estimation and variance prediction
  - Diagnostics
- Summary

Review: GARCH

Objective

- Extension of ARCH processes to capture
  - "persistence" of *outliers regions/values* in financial time series

Recall definition of (GARCH) A process $X = \{X_n\}$ is a mean-zero GARCH(p,q) process if

$$X_n = \sigma_n \epsilon_n$$

where $\{\epsilon_n\}$ is an iid white noise process with mean 0 and variance of 1, and the model for σ_n is

$$\sigma_n^2 = \alpha_o + \sum_{i=1}^{p} \beta_i \sigma_{n-i}^2 + \sum_{j=1}^{q} \alpha_j X_{n-j}^2$$

GARCH(1,1) Model

Remark. Most widely used model is GARCH(1,1) –

$$X_n = \sigma_n \epsilon_n$$

where

$$\sigma_n^2 = \alpha_o + \beta_1 \sigma_{n-1}^2 + \alpha_1 X_{n-1}^2$$

Remark. Can add a non-zero mean function μ_n to account for a non-zero mean (e.g., seasonal effects).

Assumption. For rest of this lecture, GARCH processes will be assumed to be zero-mean unless otherwise stated.

GARCH(1,1) Model: Some Properties

Important Results. If $X = \{X_n\}$ is a GARCH(1,1) process, then based on previous lecture (done in class),

$$\begin{aligned} X_n^2 &= \sigma_n^2 + \beta_1 X_{n-1}^2 - \beta_1 \sigma_{n-1}^2 - \beta_1 V_{n-1} + V_n \\ &= \alpha_o + (\alpha_1 + \beta_1) X_{n-1}^2 - \beta_1 V_{n-1} + V_n \end{aligned}$$

白噪音

where

$$V_n = \sigma_n^2(\epsilon_n^2 - 1)$$

- $Y_n = \{X_n^2\}$ is an ARMA(1,1) with a weak white-noise process $V_n = \sigma_n^2(\epsilon_n^2 - 1)$

Remark. As shown in previous lecture (in class), the V_n process is a mean-zero (weak) white noise process – derived from

- σ_n^2 depends on the history up to time
- ϵ_n is independent of the history up to time
- $E(\epsilon_n^2) = 1$ for all n

Long-Run Variance of GARCH(1,1) Processes

Remark. A GARCH(1,1) process $\{X_n\}$ with positive parameters $\alpha_1 + \beta_1 < 1$, is stationary (at least asymptotically). Basic model was $X_n = \sigma_n \epsilon_n$ where

$$\sigma_n^2 = \alpha_o + \beta_1 \sigma_{n-1}^2 + \alpha_1 X_{n-1}^2 \quad (1)$$

Example. Using the above equation, derive the long-term variance corresponding to σ_n^2.

Answer. . Suppose $E[\sigma_n^2] \longrightarrow \sigma_\infty^2$ as $n \to \infty$

long term var

$$E[\sigma_n^2] = \alpha_0 + \beta E[\sigma_n^2] + \alpha_1 E[X_{n-1}^2]$$

$$E[X_{n-1}^2] = E[\sigma_{n-1}^2]\, E[\epsilon_n^2] = E[\sigma_n^2]$$

$$E[\sigma_n^2] = \frac{\alpha_0}{1-\alpha_1-\beta}$$

Forward Prediction of Variance

Notation. We let $\mathcal{H}_n$ denote the history of the process up to time n, i.e., of $\ldots, X_{n-2}, X_{n-1}, X_n$, and conditioning on $\mathcal{H}_n$ is equivalent to conditioning on $\ldots, X_{n-2}, X_{n-1}, X_n$.

H_n只是一个代表过去的符号

Background. $\{X_n\}$ is a GARCH(1,1) process with (positive) parameters $\alpha_o, \alpha_1, \beta_1$ and $\alpha_1 + \beta_1 < 1$. In previous lecture showed that the one-step ahead prediction of variance is given by

$$E(\sigma_{n+1}^2 | \mathcal{H}_n) = \sigma_{n+1}^2 = \alpha_o + \alpha_1 X_n^2 + \beta_1 \sigma_n^2$$

Would like to derive the general k-step ahead variance prediction.

Remark. Result on the next page essentially derives the formula.

Example. For a GARCH(1,1) process X_n, given the historical data up to time n, $\mathcal{H}_n$, show that the k-ahead variance prediction for σ^2_{n+k} is given by

$$E(\sigma^2_{n+k}|\mathcal{H}_n) = \alpha_o \frac{1-\lambda^k}{1-\lambda} + \lambda^{k-1}(\alpha_1 X_n^2 + \beta_1 \sigma_n^2), \quad k = 1, 2, 3, \ldots$$

where $\lambda = (\alpha_1 + \beta_1)$.

Proof. $\sigma_{n+1}^2 = \alpha_0 + \alpha_1 X_n^2 + \beta \sigma_n^2$ $= \lambda(E[\sigma_{n+h-1}^2] - \sigma_\infty^2) = \lambda^{h-1}\sigma_n^2 + (1-\lambda^{h-1})\sigma_\infty^2$

$\sigma_{n+h}^2 = \alpha_0 + \alpha_1 X_{n+h-1}^2 + \beta \sigma_{n+h-1}^2$

$\alpha_0 = E[\sigma_\infty^2](1-\alpha_1-\beta)$

拆分a0

$\sigma_{n+h}^2 = \sigma_\infty^2 + \alpha_1(X_{n+h-1}^2 - \sigma_\infty^2) + \beta(\sigma_{n+h-1}^2 - \sigma_\infty^2)$

两边同时取期望

$E[\sigma_{n+h}^2] = E[\sigma_\infty^2] + \alpha_1(E[X_{n+h-1}^2] - \sigma_\infty^2) + \beta(E[\sigma_{n+h-1}^2] - \sigma_\infty^2)$

X和sigma期望相同

$= \sigma_\infty^2 + \lambda[E(\sigma_{n+h-1}^2) - \sigma_\infty^2] = \lambda^{h-1}\sigma_n^2 + (1-\lambda^{h-1})\sigma_\infty^2$

More on Variance Prediction

Interpretation/Implications. Recall that

$$E(\sigma^2_{n+k}|\mathcal{H}_n) = \lambda^{h-1}\sigma_n^2 + (1-\lambda^{h-1})\sigma_\infty^2$$

where

$$\sigma^2 = \frac{\alpha_0}{1-\lambda} \quad (\lambda = \alpha_1 + \beta_1)$$

- Weighted heavier towards σ^2 λ close to 0 .
- Weighted heavier towards σ^2_{n+1} λ close to 1 .

lambda的大小决定了sigma偏向于期望,还是最近值

Half-Life of Volatility in GARCH

Natural Question. How persistent is the "conditional" variance σ_n^2

- How long does it take for the k-step ahead variance prediction to converge back to σ^2?
- Compare $E(\sigma_{n+k}^2|\mathcal{H}_n) - \sigma^2$ relative to $\sigma_{n+1}^2 - \sigma^2$
- How many time steps, k, will it take before the volatility difference is half of the original volatility difference of $\sigma_{n+1}^2 - \sigma^2$

过多久才会复原?

Half-Life of Volatility in GARCH

Definition For a GARCH(1,1) process, the minimum value of k_* where the volatility difference of

$$\left|E(\sigma^2_{n+k_*}|\mathcal{H}_n) - \sigma^2\right| \leq \frac{1}{2}\left|\sigma^2_{n+1} - \sigma^2\right|$$

is referred to as the half-life of the volatility.

Result. Based on the previous results, with $\lambda = \alpha_1 + \beta_1$, have

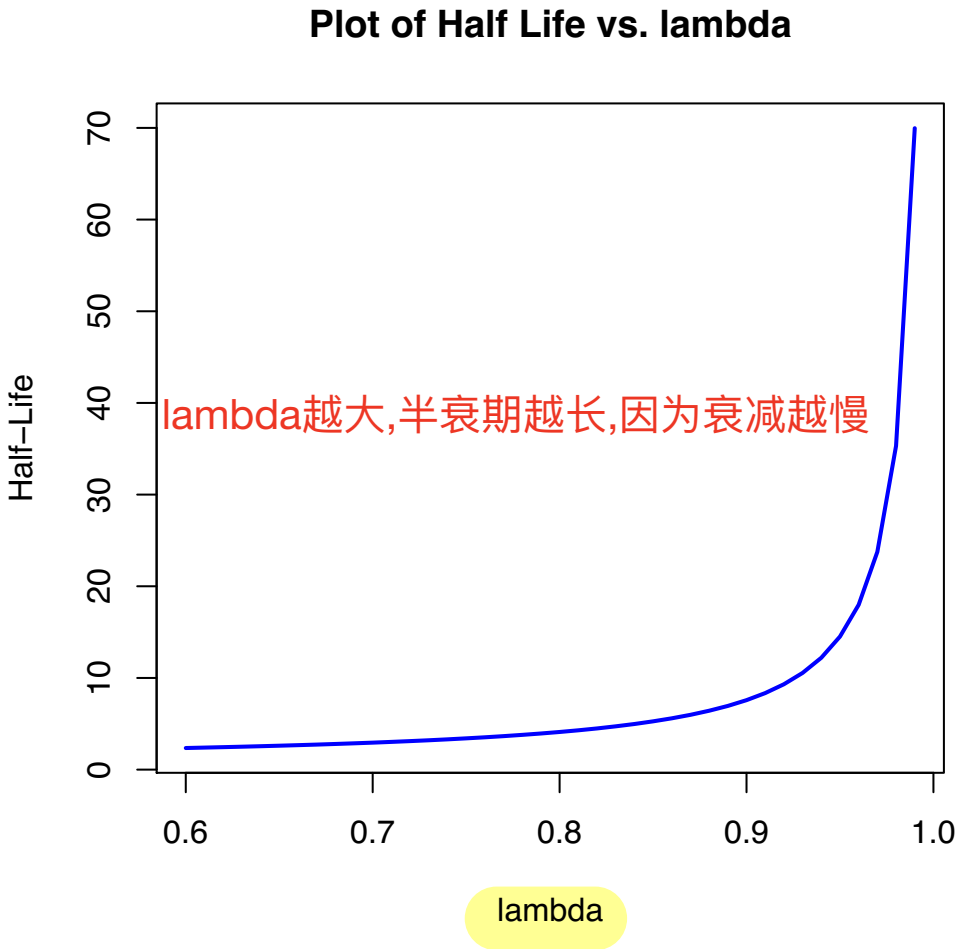
$$\left|E(\sigma^2_{n+k}|\mathcal{H}_n) - \sigma^2\right| = \lambda^{k-1}\left|\sigma^2_{n+1} - \sigma^2\right|.$$

"Volatility" half-life is the smallest positive integer $k = k_*$ such that

$$\lambda_*^{k-1} \leq \frac{1}{2}.$$

波动率半衰期是波动率之差还原到一半的时间

GARCH(1,1): Half-life Plot



Comments

ARMA/GARCH Models

Recall. The ARMA(p, q) model for X_n corresponds to

$$X_n = \sum_{i=1}^{p} \alpha_i X_{n-i} + \sum_{j=1}^{q} \phi_j \epsilon_{n-j} + \epsilon_n$$

where $\{\epsilon_n\}$ is mean-zero white noise process. A more general model is to allow the noise process ϵ_n be a GARCH(p, q) process, i.e.,

$$\epsilon_n = \sigma_n \delta_n$$ new arma process for Garch(p,q)

where δ_n is iid $\mathcal{N}(0, 1)$ (or more generally a t-distribution), and

$$\sigma_n^2 = \alpha_{g,o} + \sum_{j=1}^{p} \beta_{g,j} \sigma_{n-j}^2 + \sum_{i=1}^{q} \alpha_{g,i} \epsilon_{n-i}^2$$

ARMA-GARCH: Example and Comments

Example: For ARMA with GARCH(1,1) errors, have $\epsilon_n = \sigma_n \delta_n$ where δ_n are iid $\mathcal{N}(0,1)$ and

$$\sigma_n^2 = \alpha_{g,o} + \beta_{g,1}\sigma_{n-1}^2 + \alpha_{g,1}\epsilon_{n-1}^2$$

Remark. Note that we only see an estimated error process $\{\epsilon_n\}$

Motivation. ARMA/GARCH model essentially says:

- there is an "ARMA" type relationship
- there are periods of greater and lesser volatility in the ARMA innovations process $\{\epsilon_n\}$.

Popular models in this class are AR/GARCH (i.e., no moving average part).

MA没用

Estimation in ARMA/GARCH: Approach 1

Remark. For an ARMA/GARCH model of a fixed order, there are two approaches.

Approach 1. (i) First estimate coefficients of the ARMA model, (ii) generate residuals $\{\hat{\epsilon}_n\}$ from

$$\hat{\epsilon}_n = X_n - \sum_{i=1}^{p} \hat{\alpha}_i X_{n-i} + \sum_{j=1}^{q} \hat{\phi}_j \hat{\epsilon}_{n-j},$$

and (iii) generate estimates for GARCH model based on residuals $\hat{\epsilon}_n$. 先估计ARMA的参数,再对残差使用GARCH估计

Advantages/Disadvantages

- Theoretically less sound/accurate approach
  - Assumed estimation models for the "ARMA part" are not really right. 理论上不是最优解,但是结构清晰稳定
- More stable optimization for reliable estimation
  - Stable as it would be GARCH and ARMA separately

Estimation in ARMA/GARCH: Approach 2

Approach 2. Jointly estimate coefficients of the ARMA model and GARCH model via MLE/Conditional MLE.

Advantages/Disadvantages

- Theoretically more sound/accurate approach
- Less stable optimization routines for reliable estimation
  - Especially for higher order

一起估计的结果从理论上说结果更准确

R-Code for ARMA-GARCH

Remark. Will use R package of fGarch. For Simulation.

Usage

```
garchSim(spec = garchSpec(), n = 100, n.start = 100, extended = FALSE)
```

Arguments

extended	logical parameter if the output series should be a 3 columns timeSeries object with garch, sigma and eps data (extended = TRUE) or a univariate GARCH\APARCH time series (extended = FALSE
spec	a specification object of class "fGARCHSPEC" as returned by the function garchSpec. The model parameters are taken from the @model slot, a list with the following entries: omega - the constant coefficient of the variance equation, by default 1e-6; alpha - the value or vector of autoregressive coefficients, by default 0.1, specifying a model of order 1; beta - the value or vector of variance coefficients, by default 0.8, specifying a model of order 1; The optional values for the linear part are: mu - the intercept value, by default 0 (it implies that the mean = mu/(1-sum(ar))); ar - the autoregressive ARMA coefficients, by default 0; ma - the moving average ARMA coefficients, by default 0. The optional parameters for the conditional distributions are: skew - the skewness parameter (also named xi), by default 0.9, effective only for the "dsnorm", the "dsged", and the "dsstd" skewed conditional distributions; shape = the shape parameter (also named nu), by default 2 for the "dged" and "dsged", and by default 4 for the "dstd" and "dsstd" conditional distributions. See also below for further details.
n	length of output series, an integer value. An integer value, by default n=100.
n.start	length of "burn-in" period, by default 100.

Note. GARCH parameters are "omega", "alpha1", and "beta1".

R-Code for ARMA-GARCH: Examples

Examples of `garchSpec`: : For example specifying a AR(2)-GARCH(1,1) model.

```
garchSpec(model = list(ar = c(0.5,0.1), omega = 0.2, alpha = 0.2,
+ beta = 0.65), cond.dist = "norm")
```

For an AR(5[1,5])-GARCH(2,1) model with a standardized Student-t distribution with four degrees of freedom will return the following printed output:

```
garchSpec(model = list(ar = c(0.5,0,0,0,0.1), omega = 0.2, alpha = c(0.1, 0.1),
+ beta = 0.75, shape = 4), cond.dist = "std")
```

残差是t分布

Examples of `garchSim`: For simulation of AR(2)-GARCH(1,1) model,

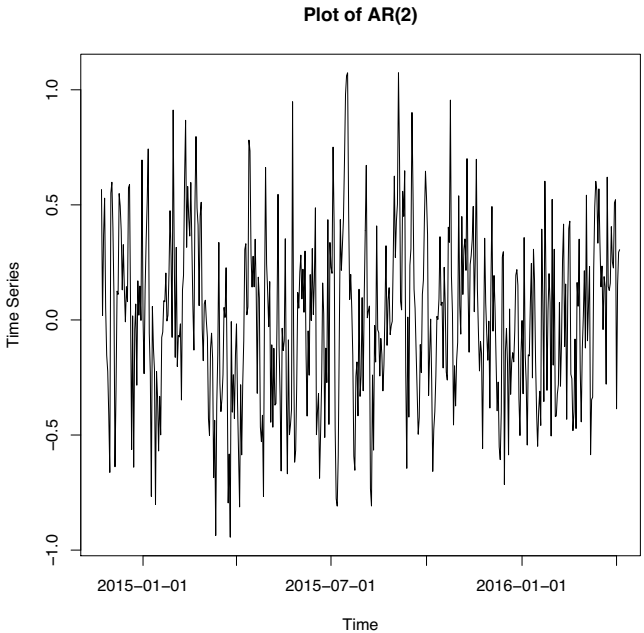
```
garchSim(spec = garchSpec(model = list(ar = c(0.5,0.1), omega = 0.2, alpha =
+ 0.2, beta = 0.65), cond.dist = "norm"), n = 100, n.start = 100,
+ extended = FALSE)
```

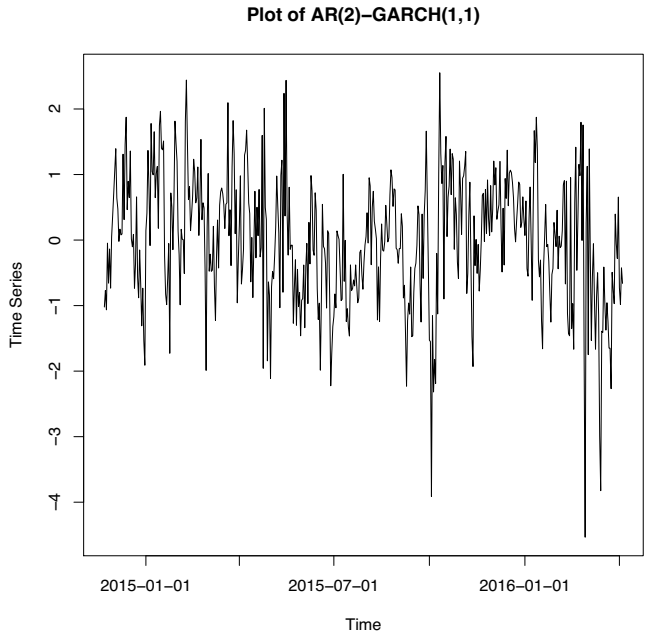
Simulations of Processes

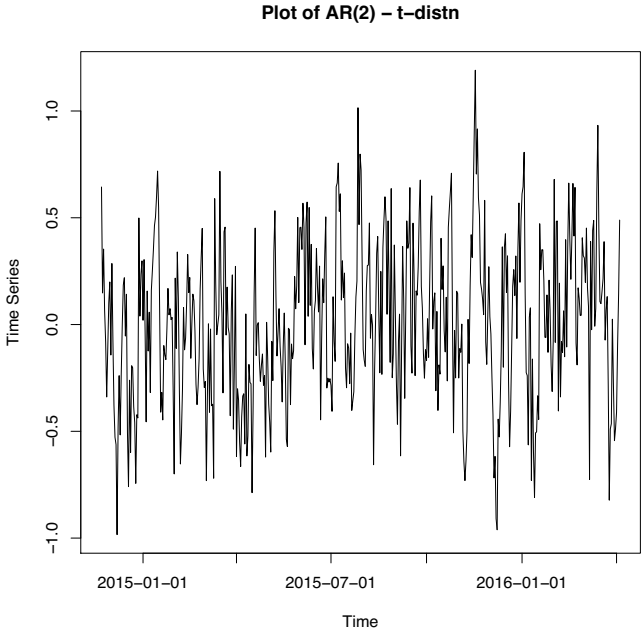
Remark. Parameter values

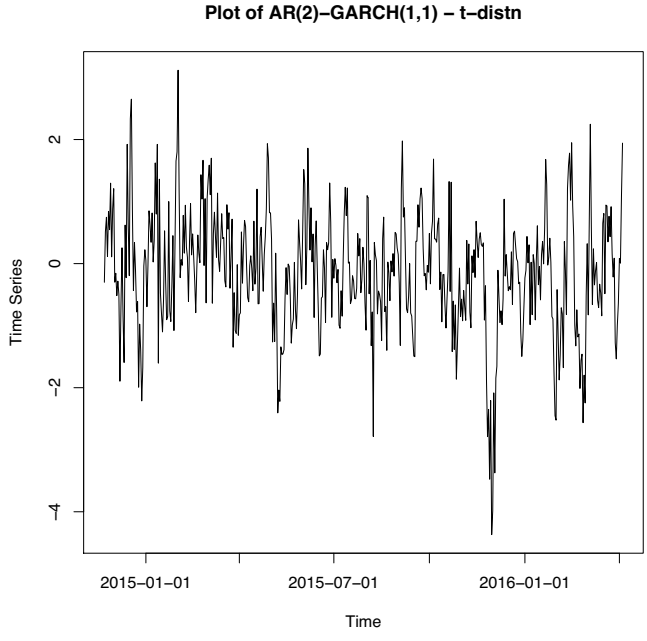
- AR parameters were .5 and .1
- GARCH parameters were omega = .1, alpha = .2 and beta = .65
- t-distn had df parameter (shape) = 4

t分布明显尾部更大






Plot of AR(2) – t–distn
Time Series
1.0
0.5
0.0
−0.5
−1.0
2015−01−01
2015−07−01
2016−01−01
Time


Plot of AR(2)–GARCH(1,1) – t–distn
Time Series
2
0
−2
−4
2015−01−01
2015−07−01
2016−01−01
Time

R-Commands for Simulations

```
> garchSim_AR = garchSim(garchSpec(model = list(ar = c(0.5,0.1),
+ omega = 0.1, alpha = 0, beta = 0, cond.dist='norm')),
+ n = 500, n.start = 100)
>
> plot(garchSim_AR, ylab='Time Series', main = 'Plot of AR(2)')
>
> garchSim_ARGARCH = garchSim(garchSpec(model = list(ar = c(0.5,0.1),
+ omega = 0.1, alpha = 0.2, beta = 0.65, cond.dist = "norm")), n = 500,
+ n.start = 100)
>
> plot(garchSim_ARGARCH, ylab='Time Series', main = 'Plot of AR(2)-GARCH(1,1)')
>
> garchSim_AR_tdist = garchSim(garchSpec(model = list(ar = c(0.5,0.1),
+ omega = 0.1, alpha = 0, beta = 0, shape = 4, cond.dist = "std")),
+ n = 500, n.start = 100)
>
> plot(garchSim_AR_tdist, ylab='Time Series', main = 'Plot of AR(2) - t-distn')
>
```

```
> garchSim_ARGARCH_tdist = garchSim(garchSpec(model = list(ar = c(0.5,0.1),
+ omega = 0.1, alpha = 0.2, beta = 0.65, shape = 4, cond.dist = "std")),
+ n = 500, n.start = 100)
>
> windows()
> plot(garchSim_ARGARCH_tdist,ylab='Time Series',
+ main = 'Plot of AR(2)-GARCH(1,1) - t-distn')
```

ARMA-GARCH Estimation in R

```
garchFit(formula = ~ garch(1, 1), data = dem2gbp,
    init.rec = c("mci", "uev"), delta = 2, skew = 1, shape = 4,
    cond.dist = c("norm", "snorm", "ged", "sged", "std", "sstd",
        "snig", "QMLE"),
    include.mean = TRUE, include.delta = NULL, include.skew = NULL,
        include.shape = NULL, leverage = NULL, trace = TRUE,
        algorithm = c("nlminb", "lbfgsb", "nlminb+nm", "lbfgsb+nm"),
    hessian = c("ropt", "rcd"), control = list(),
    title = NULL, description = NULL, ...)
```

Arguments

algorithm -- a string parameter that determines the algorithm used for maximum likelihood estimation.

cond.dist -- a character string naming the desired conditional distribution. Valid values are "dnorm", "dged", "dstd", "dsnorm", "dsged", "dsstd" and "QMLE". The default value is the normal distribution and snorm, sstd, and sged are skewed versions of normal, t, and ged.

control -- control parameters, the same as used for the functions from nlminb, and 'bfgs' and 'Nelder-Mead' from optim.

data -- an optional timeSeries or data frame object containing the variables in the model. If not found in data, the variables are taken from environment (formula), typically the environment from which armaFit is called. If data is an univariate series, then the series is converted into a numeric vector and the name of the response in the formula will be neglected.

delta -- a numeric value, the exponent delta of the variance recursion. By default, this value will be fixed, otherwise the exponent will be estimated together with the other model parameters if include.delta=FALSE.

description -- a character string which allows for a brief description.

formula -- formula object describing the mean and variance equation of the ARMA-GARCH/APARCH model. A pure GARCH(1,1) model is selected when e.g. formula=~garch(1,1). To specify for example an ARMA(2,1)-APARCH(1,1) use formula = ~arma(2,1)+apaarch(1,1).

gamma -- APARCH leverage parameter entering into the formula for calculating the expectation value.

hessian -- a string denoting how the Hessian matrix should be evaluated, either hessian ="rcd", or "ropt", the default, "rcd" is a central difference approximation implemented in R and "ropt" uses internal R function optimhess.

include.delta -- a logical flag which determines if the parameter for the recursion equation delta will be estimated or not. If include.delta=FALSE then the shape parameter will be kept fixed during the process of parameter optimization.

include.mean -- this flag determines if the parameter for the mean will be estimated or not. If include.mean=TRUE this will be the case, otherwise the parameter will be kept fixed durcing the process of parameter optimization.

include.shape -- a logical flag which determines if the parameter for the shape of the conditional distribution will be estimated or not. If include.shape=FALSE then the shape parameter will be kept fixed during the process of parameter optimization.

include.skew --
a logical flag which determines if the parameter for the skewness of the conditional distribution will be estimated or not. If include.skew=FALSE then the skewness parameter will be kept fixed during the process of parameter optimization.

init.rec -- a character string indicating the method how to initialize the mean and variance recursion relation.

leverage -- a logical flag for APARCH models. Should the model be leveraged? By default leverage=TRUE.

shape -- a numeric value, the shape parameter of the conditional distribution.

skew -- a numeric value, the skewness parameter of the conditional distribution

title -- a character string which allows for a project title.

trace -- a logical flag. Should the optimization process of fitting the model parameters be printed? By default trace=TRUE.

garchFit returns a S4 object of class "fGARCH" with the following slots:
@call the call of the garch function.

@formula a list with two formula entries, one for the mean and the other one for the variance equation.

@method a string denoting the optimization method, by default the returned string is "Max Log-Likelihood Estimation".

@data a list with one entry named x, containing the data of the time series to be estimated,the same as given by the input argument series.

@fit a list with the results from the parameter estimation. The entries of the list depend on the selected algorithm, see below.

@residuals a numeric vector with the (raw, unstandardized) residual values.

@fitted a numeric vector with the fitted values.

@h.t a numeric vector with the conditional variances

@sigma.t a numeric vector with the conditional standard deviation.

The entries of the @fit slot show the results from the optimization.

Example Summary Statistics for ARMA-GARCH

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1.024422  0.5991694
 Shapiro-Wilk Test  R    W      0.9974575 0.6464629
 Ljung-Box Test     R    Q(10)  4.303372  0.9326253
 Ljung-Box Test     R    Q(15)  8.760224  0.8897171
 Ljung-Box Test     R    Q(20)  14.66198  0.7954114
 Ljung-Box Test     R^2  Q(10)  8.194847  0.6098111
 Ljung-Box Test     R^2  Q(15)  9.436022  0.8536389
 Ljung-Box Test     R^2  Q(20)  11.32885  0.9372444
 LM Arch Test       R    TR^2   8.825492  0.717759

Information Criterion Statistics:
     AIC      BIC      SIC      HQIC
2.377287 2.427863 2.377004 2.397133
```

Test for ARMA $\hat{\epsilon}_n$

Test for Garch $\hat{\delta}_n = \frac{\hat{\epsilon}_n}{\hat{\sigma}_n}$

Used for final selection

Summary Statistics - garchFit

Remark. The **Shapiro-Wilk** test tests the null hypothesis that a sample was from a normally distributed population. The test statistic is

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

where

- $x_{(i)}$ (with parentheses enclosing the subscript index i) is the ith order statistic, i.e., the ith-smallest number in the sample;
- $\overline{x} = (x_1 + \cdots + x_n)/n$ is the sample mean;
- the constants a_i are derived from the mean/covariance of the order statistics assuming normal distribution holds.
- p-value derived from the distribution of W und the null hypothesis

Model Selection

Remark. There are 4 selection criteria for models

- AIC - Akaike information criteria
- BIC - Bayesian information criteria
- SIC - Schwarz information criteria
- HQIC - Hannan-Quinn information criteria

Simulation Example: ARMA-GARCH Estimation

```
> garchFit_ARGARCH <- garchFit(~arma(2,0)+garch(1,1),
+ data=garchSim_ARGARCH, cond.dist = c("norm"), include.mean = FALSE,
+ algorithm = c("nlminb"), hessian = c("ropt"))

> summary(garchFit_ARGARCH)

Mean and Variance Equation:
 ~arma(2, 0) + ~garch(1, 1)

Conditional Distribution:
 norm

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1      0.43124     0.04798    8.989  < 2e-16 ***
ar2      0.11457     0.04710    2.433  0.01499 *
omega    0.09577     0.03778    2.535  0.01124 *
alpha1   0.17390     0.05440    3.196  0.00139 **
beta1    0.68558     0.08373    8.188 2.22e-16 ***
---
```

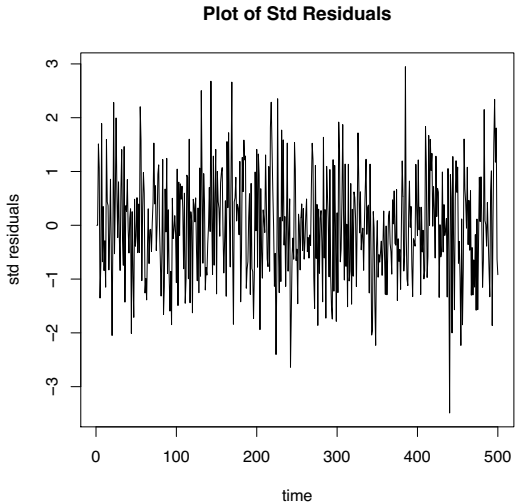
ARMA

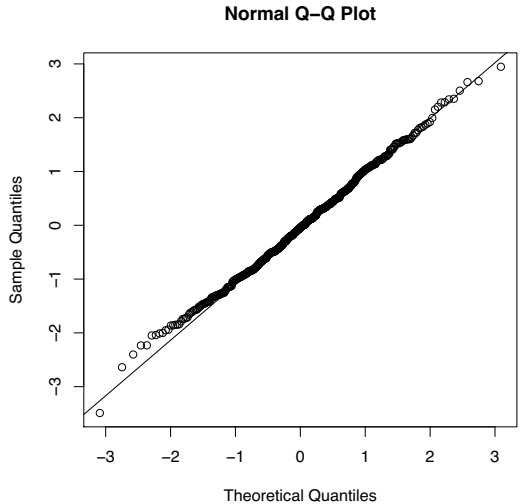
Garch

```
Log Likelihood:
 -588.3219    normalized:  -1.176644

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1.024422  0.5991694
 Shapiro-Wilk Test  R    W      0.9974575 0.6464629
 Ljung-Box Test     R    Q(10)  4.303372  0.9326253
 Ljung-Box Test     R    Q(15)  8.760224  0.8897171
 Ljung-Box Test     R    Q(20)  14.66198  0.7954114
 Ljung-Box Test     R^2  Q(10)  8.194847  0.6098111
 Ljung-Box Test     R^2  Q(15)  9.436022  0.8536389
 Ljung-Box Test     R^2  Q(20)  11.32885  0.9372444
 LM Arch Test       R    TR^2   8.825492  0.717759

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
2.377287 2.427863 2.377004 2.397133
```

Simulation: AR-GARCH Estimation Diagnostics

Remark. Below is time series plot and QQ plot of residuals





拟合的很好

```
# Plot and QQ plot of residuals
> windows()
> qqnorm(garchFit_ARGARCH@residuals/garchFit_ARGARCH@sigma.t)
> qqline(garchFit_ARGARCH@residuals/garchFit_ARGARCH@sigma.t)
> windows()
> plot(garchFit_ARGARCH@residuals/garchFit_ARGARCH@sigma.t,xlab='time',
+ ylab='std residuals',type='l',main='Plot of Std Residuals')
```

Simulation: Redo of AR-GARCH Estimation

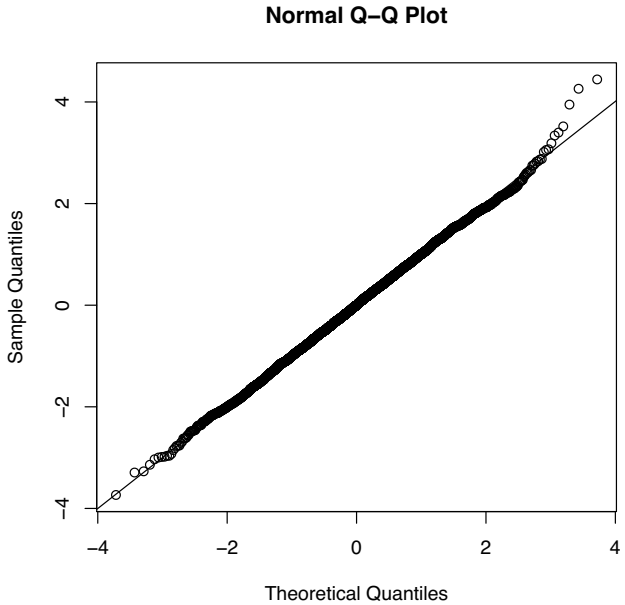
Remark. Redid the AR/GARCH Estimation with $n = 5000$. Results are below:

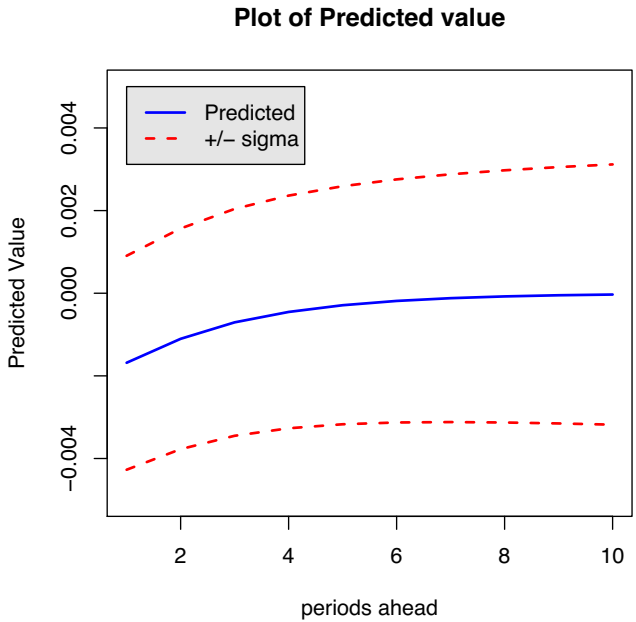
```
Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
ar1      0.50778     0.01529   33.202  < 2e-16 ***
ar2      0.10686     0.01482    7.209 5.63e-13 ***
omega    0.11372     0.01358    8.377  < 2e-16 ***
alpha1   0.21674     0.01915   11.319  < 2e-16 ***
beta1    0.60264     0.03167   19.030  < 2e-16 ***
```

```
Log Likelihood:
 -5665.848    normalized:  -1.13317

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  0.1278759 0.9380632
 Shapiro-Wilk Test  R    W      NA        NA
 Ljung-Box Test     R    Q(10)  13.31915  0.2063712
 Ljung-Box Test     R    Q(15)  16.52912  0.3477832
 Ljung-Box Test     R    Q(20)  17.236    0.6376
 Ljung-Box Test     R^2  Q(10)  10.34137  0.4110716
 Ljung-Box Test     R^2  Q(15)  11.29036  0.7317528
 Ljung-Box Test     R^2  Q(20)  13.41155  0.8590363
 LM Arch Test       R    TR^2   11.08868  0.5213345

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
2.268339 2.274856 2.268337 2.270623
```

Simulation: AR-GARCH Est with $n = 5000$

Remark. Below is QQ plot of residuals and predict ahead values.





AR drive mean, Garch drive std band

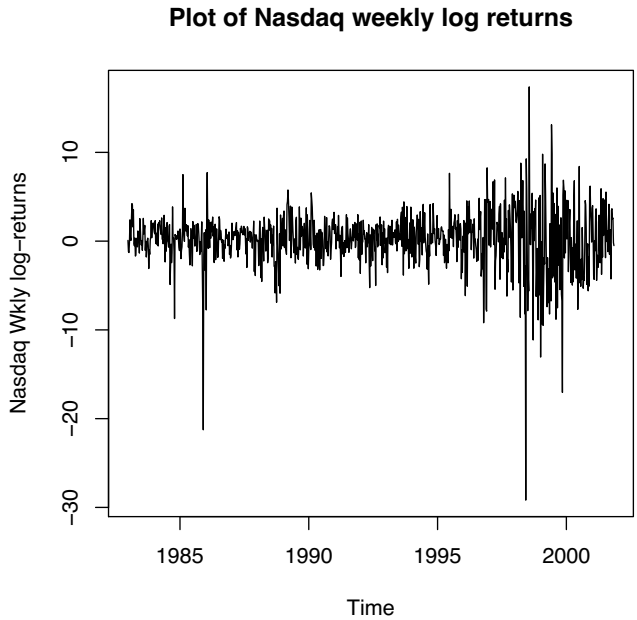
```
# Plot and QQ plot of residuals
> windows()
> qqnorm(garchFit_ARGARCH5000@residuals/garchFit_ARGARCH5000@sigma.t)
> qqline(garchFit_ARGARCH5000@residuals/garchFit_ARGARCH5000@sigma.t)
> windows()
> plot(garchFit_ARGARCH5000@residuals/garchFit_ARGARCH5000@sigma.t,
+ xlab='time', ylab='std residuals',type='l',main='Plot of Std Residuals')
```

```
> predict_ARGARCH5000 <- predict(garchFit_ARGARCH5000,n.ahead=10)
> windows()
> plot(predict_ARGARCH5000$meanForecast,type='l',lty=1,lwd=2,
+ ylim=c(-.005,.005),xlab='periods ahead',ylab='Predicted Value',col='blue',
+ main='Plot of Predicted value')
> lines(predict_ARGARCH5000$meanForecast+predict_ARGARCH5000$standardDeviation,
+ type='l',lty='dashed',lwd=2,col='red')
> lines(predict_ARGARCH5000$meanForecast-predict_ARGARCH5000$standardDeviation,
+ lty='dashed',lwd=2,col='red')
> legend(1,.005, c("Predicted","+/- sigma"), lty=c(1,2),lwd=c(2,2),
+ col=c("blue","red"), bg="gray90")
```

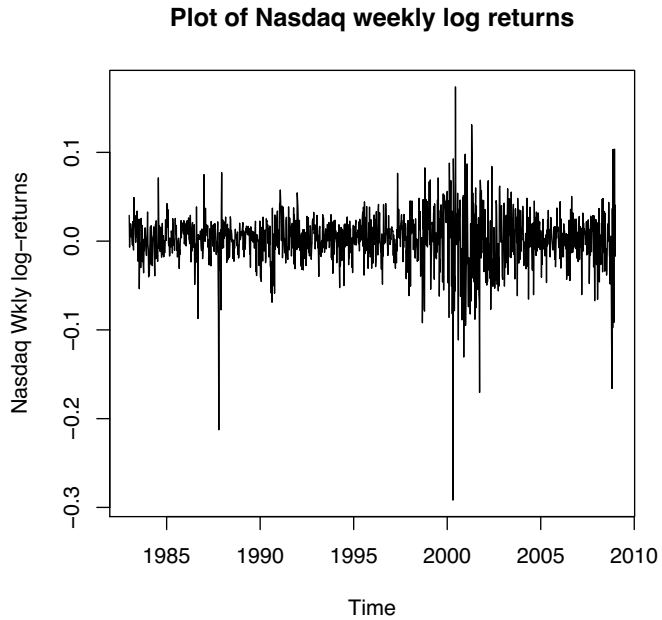
Nasdaq-Weekly Log Return: ARMA-GARCH

Background: Nasdaq weekly log returns (scaled by 1000) from 1983 to 2003.

1983-2003
Log Wkly Returns



1983-2008
Log Wkly Returns



Comments

Nasdaq-Weekly Log Return: ARMA-GARCH

Remark. Estimation with

GARCH(1,1) $\rightarrow$ AR(1) - GARCH(1,1) $\rightarrow$ AR(2) - GARCH(1,1)

R-Code/Results

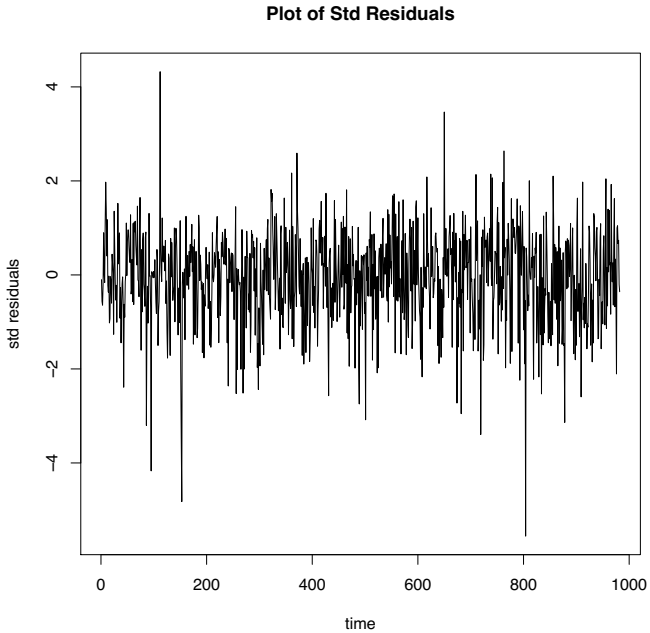
```
> X = read.table("Data//Nasdaq_wklogreturn_83-03.asc",,header=TRUE)
> X.ts <- ts(data=X[,2],start=c(1982,52),frequency=52)

# GARCH(1,1) Model
> Nasdaq_GARCH <- garchFit(~garch(1,1), data=X.ts, cond.dist =
+ c("norm"), include.mean = TRUE, trace = TRUE,
+ algorithm = c("nlminb"), hessian = c("ropt"))

> summary(Nasdaq_GARCH)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.40475     0.06586    6.146 7.96e-10 ***
omega    0.27410     0.09034    3.034  0.00241 **
alpha1   0.24959     0.04267    5.850 4.92e-09 ***
beta1    0.74700     0.03898   19.163  < 2e-16 ***
---
```

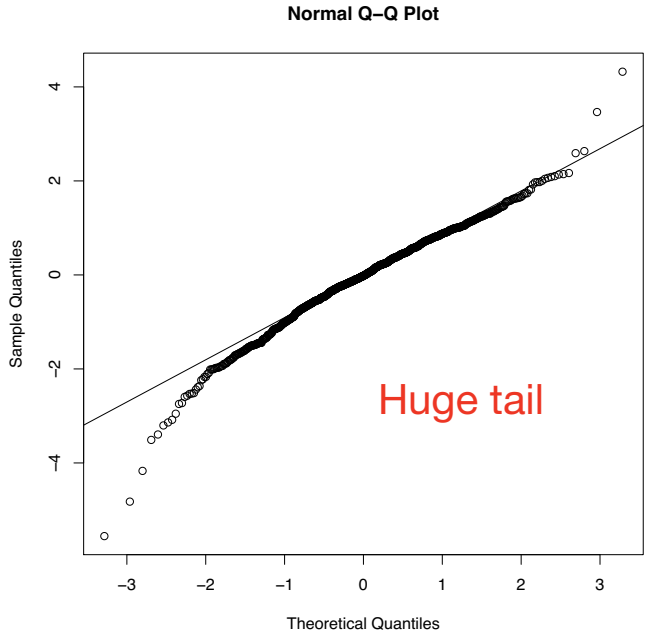
```
Standardized Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  205.9219  0
 Shapiro-Wilk Test  R    W      0.9771283 2.625878e-11
 Ljung-Box Test     R    Q(10)  21.13357  0.02018122
 Ljung-Box Test     R    Q(15)  25.44191  0.04431441
 Ljung-Box Test     R    Q(20)  28.97423  0.08826797
 Ljung-Box Test     R^2  Q(10)  6.775933  0.7464154
 Ljung-Box Test     R^2  Q(15)  8.096496  0.9198558
 Ljung-Box Test     R^2  Q(20)  16.83387  0.6637333
 LM Arch Test       R    TR^2   7.56719   0.8179732

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.704844 4.724761 4.704811 4.712421
```

p-value小说明AR模型更重要

Residual Diagnostic Plots - NASDAQ GARCH(1,1)

Remark. Below is time series and QQ plot of residuals.






```
> windows()
> qqnorm(Nasdaq_GARCH@residuals/Nasdaq_GARCH@sigma.t)
> qqline(Nasdaq_GARCH@residuals/Nasdaq_GARCH@sigma.t)
> windows()
> plot(Nasdaq_GARCH@residuals/Nasdaq_GARCH@sigma.t,type='l')
```

```
> Nasdaq_AR1GARCH <- garchFit(~arma(1,0)+garch(1,1), data=X.ts,
+ cond.dist = c("norm"), include.mean = TRUE, trace = TRUE,
+ algorithm = c("nlminb"), hessian = c("ropt"))

> summary(Nasdaq_AR1GARCH)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.35122     0.06714    5.231 1.68e-07 ***
ar1      0.09995     0.03641    2.745  0.00604 **
omega    0.25199     0.08529    2.954  0.00313 **
alpha1   0.23521     0.04082    5.762 8.31e-09 ***
beta1    0.76111     0.03773   20.172  < 2e-16 ***
---

Log Likelihood:
 2302.342    normalized:  2.344544
```

AR(1) is significant

```
Standadized Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  209.0677  0
 Shapiro-Wilk Test  R    W      0.9765914 1.783504e-11
 Ljung-Box Test     R    Q(10)  12.36935  0.2610973
 Ljung-Box Test     R    Q(15)  17.4209   0.2943317
 Ljung-Box Test     R    Q(20)  20.53337  0.42504
 Ljung-Box Test     R^2  Q(10)  6.66775   0.7563948
 Ljung-Box Test     R^2  Q(15)  8.161231  0.917156
 Ljung-Box Test     R^2  Q(20)  16.52449  0.6835994
 LM Arch Test       R    TR^2   7.772685  0.8026325

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.699271 4.724167 4.699220 4.708742
```

```
> Nasdaq_AR2GARCH <- garchFit(~arma(2,0)+garch(1,1), data=X.ts,
+ cond.dist = c("norm"), include.mean = TRUE, trace = TRUE,
+ algorithm = c("nlminb"), hessian = c("ropt"))

> summary(Nasdaq_AR2GARCH)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.32197     0.06766    4.759 1.95e-06 ***
ar1      0.09233     0.03583    2.577  0.00997 **
ar2      0.07642     0.03444    2.219  0.02647 *
omega    0.23396     0.08131    2.877  0.00401 **
alpha1   0.22470     0.03949    5.690 1.27e-08 ***
beta1    0.77164     0.03666   21.049  < 2e-16 ***
---

Log Likelihood:
 2299.661    normalized:  2.341814
```

Standadized Residuals Tests:

			Statistic	p-Value
Jarque-Bera Test	R	Chi^2	201.2304	0
Shapiro-Wilk Test	R	W	0.9776948	3.974645e-11
Ljung-Box Test	R	Q(10)	7.600067	0.667837
Ljung-Box Test	R	Q(15)	12.29988	0.6562029
Ljung-Box Test	R	Q(20)	14.94547	0.7795194
Ljung-Box Test	R^2	Q(10)	6.749689	0.7488456
Ljung-Box Test	R^2	Q(15)	8.05808	0.921433
Ljung-Box Test	R^2	Q(20)	16.30158	0.6977471
LM Arch Test	R	TR^2	7.70068	0.8080621

Information Criterion Statistics:

AIC	BIC	SIC	HQIC
4.695848	4.725723	4.695774	4.707212

```
> Nasdaq_AR3GARCH <- garchFit(~arma(3,0)+garch(1,1), data=X.ts,
+ cond.dist = c("norm"), include.mean = TRUE, trace = TRUE,
+ algorithm = c("nlminb"), hessian = c("ropt"))

> summary(Nasdaq_AR3GARCH)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.32114     0.06788    4.731 2.24e-06 ***
ar1      0.09036     0.03601    2.509  0.01209 *
ar2      0.07390     0.03482    2.122  0.03382 *
ar3      0.01615     0.03374    0.479  0.63211
omega    0.23850     0.08342    2.859  0.00425 **
alpha1   0.22845     0.04104    5.566 2.60e-08 ***
beta1    0.76828     0.03801   20.211  < 2e-16 ***
---

Log Likelihood:
 2299.311    normalized:  2.341457
```

Standadized Residuals Tests:

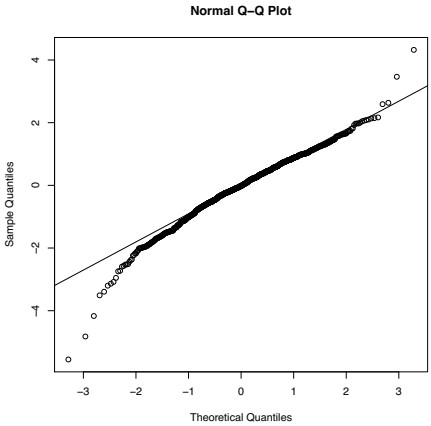
			Statistic	p-Value
Jarque-Bera Test	R	Chi^2	203.3060	0
Shapiro-Wilk Test	R	W	0.9776307	3.791254e-11
Ljung-Box Test	R	Q(10)	7.019782	0.7235761
Ljung-Box Test	R	Q(15)	11.78529	0.6952026
Ljung-Box Test	R	Q(20)	14.34609	0.8125236
Ljung-Box Test	R^2	Q(10)	6.64179	0.758774
Ljung-Box Test	R^2	Q(15)	7.952644	0.9256672
Ljung-Box Test	R^2	Q(20)	16.19199	0.7046441
LM Arch Test	R	TR^2	7.575006	0.8173985

Information Criterion Statistics:

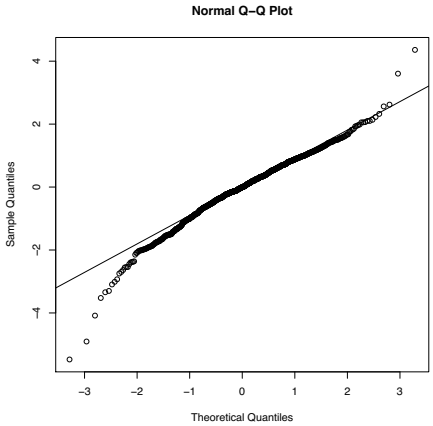
AIC	BIC	SIC	HQIC
4.697171	4.732026	4.697070	4.710430

QQ Plots

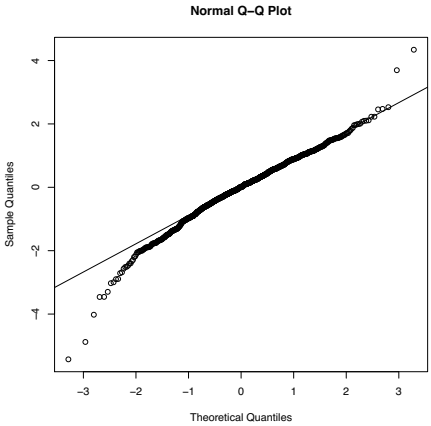
GARCH(1,1)



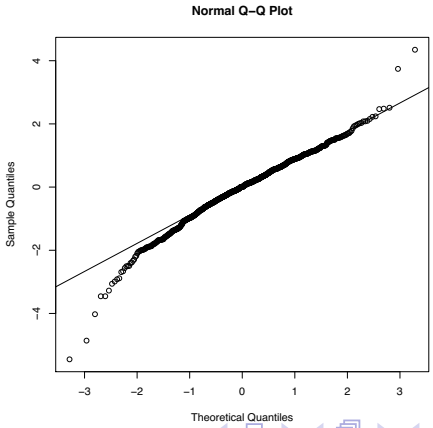
AR(1)-GARCH(1,1)



AR(2)-GARCH(1,1)



AR(3)-GARCH(1,1)



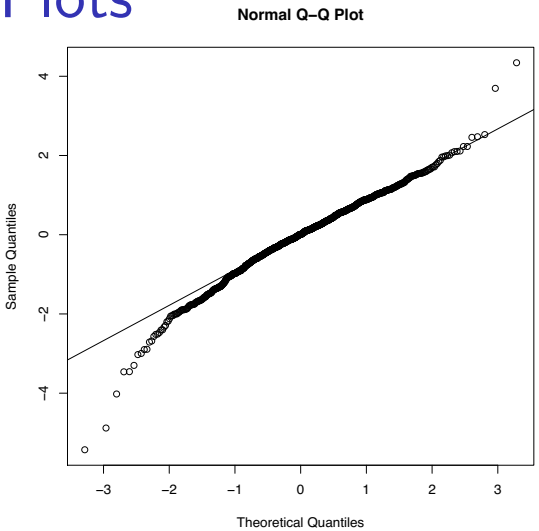
Comparison of Models via Information Criteria

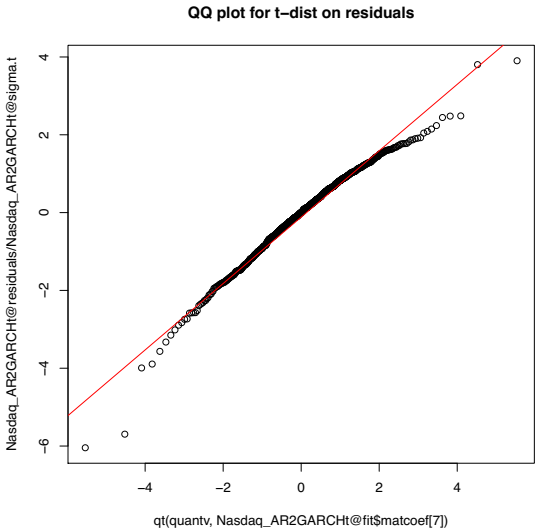
- All criteria should be multiplied by number of time series samples, i.e., $n = 982$

```
Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.704844 4.724761 4.704811 4.712421

AR(1)-GARCH(1,1) - Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.699271 4.724167 4.699220 4.708742

AR(2)-GARCH(1,1) - Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.695848 4.725723 4.695774 4.707212

AR(3)-GARCH(1,1) - Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.697171 4.732026 4.697070 4.710430

AR(2)-GARCH(1,1)-t-distn Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.647936 4.682790 4.647835 4.661194
```

Remark: For t-distribution, get shape estimate of 6.69 - interpretation

QQ Plots



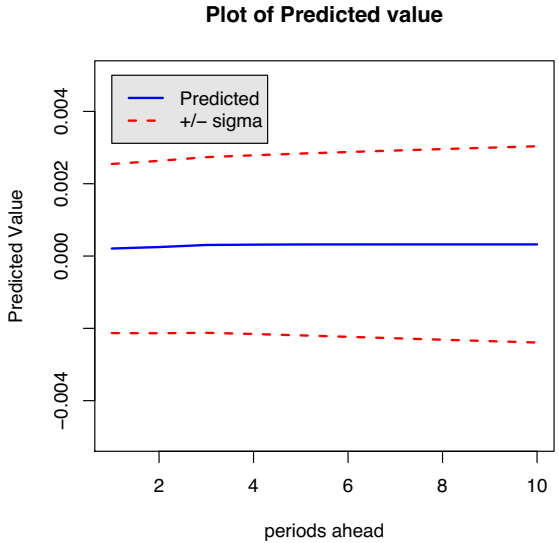


AR(2)-GARCH(1,1)-normal AR(2)-GARCH(1,1)-t-distn

```
> # QQ plot for t-distn - estimates are in Nasdaq_AR2GARCHt@fit$matcoef
> # Shape estimate is Nasdaq_AR2GARCHt@fit$$matcoef[7]
> N = length(Nasdaq_AR2GARCHt@residuals)
> quantv = (1/N)*seq(.5,N-.5,1)
> qqplot(qt(quantv, Nasdaq_AR2GARCHt@fit$matcoef[7]),
 + Nasdaq_AR2GARCHt@residuals/Nasdaq_AR2GARCHt@sigma.t,
 + main='QQ plot for t-dist on residuals')
> qqline(Nasdaq_AR2GARCHt@residuals/Nasdaq_AR2GARCHt@sigma.t,
+ distribution = function(p) qt(p,Nasdaq_AR2GARCHt@fit$matcoef[7]),
+ prob = c(0.1, 0.9), col = 2)
```

Prediction for Nasdaq wkly log returns

Remark. Below is plot of predict ahead values AR(2)-GARCH(1,1) .



```
> windows()
> predict_Nasdaq_AR2GARCH <- predict(Nasdaq_AR2GARCH,n.ahead=10)
> plot(.001*predict_Nasdaq_AR2GARCH$meanForecast,type='l',lty=1,lwd=2,
+ ylim=c(-.005,.005),xlab='periods ahead',ylab='Predicted Value',col='blue',
+ main='Plot of Predicted value')
> lines(.001*(predict_Nasdaq_AR2GARCH$meanForecast+
+ predict_Nasdaq_AR2GARCH$standardDeviation),type='l',lty='dashed',
+ lwd=2,col='red')
> lines(.001*(predict_Nasdaq_AR2GARCH$meanForecast-
+ predict_Nasdaq_AR2GARCH$standardDeviation),lty='dashed',lwd=2,col='red')
> legend(1,.005, c("Predicted","+/- sigma"), lty=c(1,2),lwd=c(2,2),
+ col=c("blue","red"), bg="gray90")
```

GARCH - Nonstationarity

For Garch(1,1) X_n^2 is ARMA(1,1)

alpha=alpha_1+beta_1,theta=-beta_1

```
> summary(Nasdaq_GARCH)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.40475     0.06586    6.146 7.96e-10 ***
omega    0.27410     0.09034    3.034  0.00241 **
alpha1   0.24959     0.04267    5.850 4.92e-09 ***
beta1    0.74700     0.03898   19.163  < 2e-16 ***
---
```

Remark. Note that $\alpha_1 + \beta_1 = .997$ – this is typical. What does this imply about the ACF of the log returns?

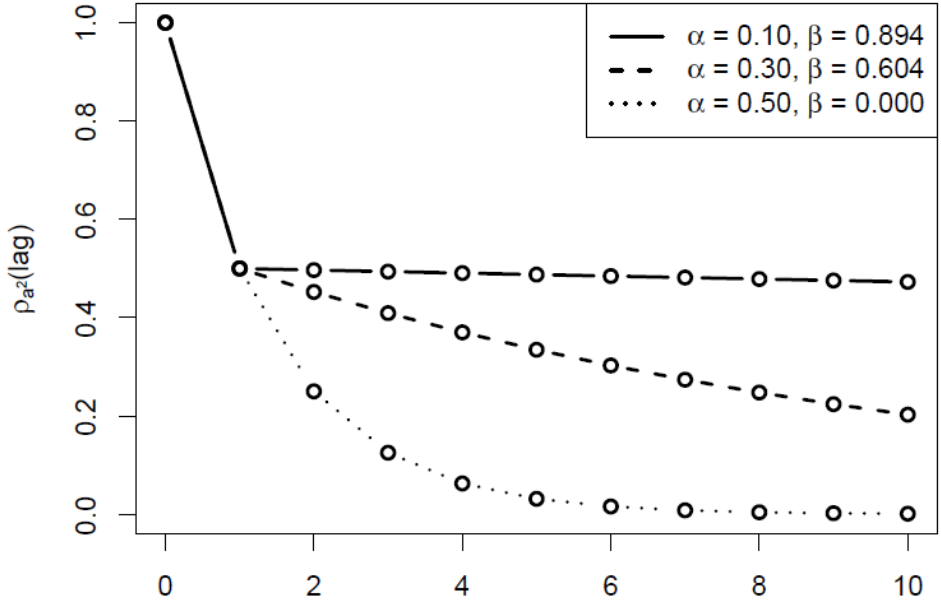
Answer . Recall GARCH(1,1) model us the errors, implies ARMA(1,1) on epsilon get coefficients $\alpha_1+\beta_1$ for AR part, $-\beta_1$ for MA part.

The first auto correlation decrease quickly, then slowly

$$\rho_{X_n^2}(h) \simeq (\alpha_1+\beta_1)^{|h|-1} \frac{\alpha_1[1-(\alpha_1+\beta_1)\beta_1]}{1+\beta_1^2-2(\alpha_1+\beta_1)\beta_1}$$

ACF Plot for X_t^2 when X_t ~GARCH(1,1)

$\alpha_1+\beta_1$决定了之后的下降速度
但是初始下降速度都一样



beta越大,越容易受到之前影响,下降越慢

Implications

APARCH: Overview

Motivation

拟合不对称分布

- Modeling of of asymmetry of the noise/residuals distribution
- Flexibility in evolutionary equations for σ_t
  - Standard model is to focus on σ_t^2
  - Alternative model is to focus on σ_t^δ with $\delta > 0$ is free parameter

APARCH Model

$$\sigma_t^\delta = \alpha_o + \sum_{i=1}^{p} \alpha_i \left(|X_{t-i}| - \gamma_i X_{t-i} \right)^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta$$

- power parameter $\delta > 0$
- leverage parameters $-1 < \gamma_j < 1$
- GARCH(1,1) corresponds to $\gamma=0, \delta=2$

APARCH Model: Leverage Effects

- For APARCH Model

$$\sigma_t^\delta = \alpha_o + \sum_{i=1}^{p} \alpha_i \left(|X_{t-i}| - \gamma_i X_{t-i}\right)^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta$$

- Function $q_\gamma(x) = (|x| - \gamma x)$ governs effects of X_{t-i} on σ_t

bigger contribution for volatility from positive return

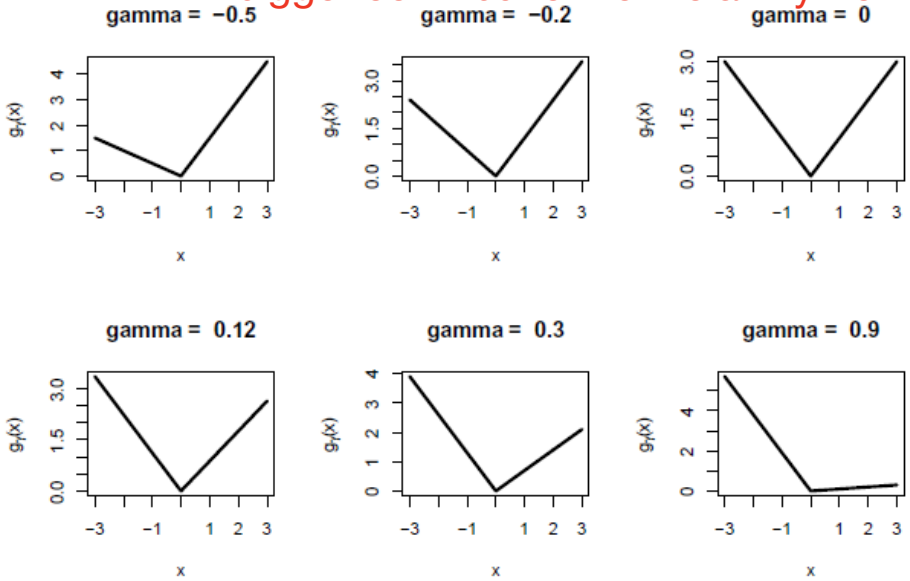


Fig. 18.7. *Plots of $g_\gamma(x)$ for various values of γ.*

R-Functions for APARCH

```
garchFit(formula = ~ arma(2,0) + aparch(1, 1), data = x,
    include.delta = T, cond.dist = c("std"),
    include.mean = T, include.shape = T)
```

- Output is similar to what was shown below

APARCH Example: BMW Returns

Background: bmw are the daily log returns on BMW share price from Tuesday 2nd January 1973 until Tuesday 23rd July 1996.

- Apply AR(2)-APARCH(1,1) model with t-distribution

```
> library(evir)
> library(fGarch)
> data(bmw)
> bmw_aparch <- garchFit(~arma(2,0)+aparch(1,1), data=bmw, cond.dist=c("std"),
+ include.mean=T,include.leverage = T,trace=F)
> summary(bmw_aparch)
Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      5.165e-05   1.370e-04    0.377  0.70628
ar1     6.566e-02   1.251e-02    5.246 1.55e-07 ***
ar2    -3.306e-02   1.207e-02   -2.740  0.00614 **
omega   5.581e-05   1.262e-05    4.422 9.77e-06 ***
alpha1  9.933e-02   1.274e-02    7.798 6.22e-15 ***
gamma1  1.183e-01   4.456e-02    2.655  0.00793 **
beta1   8.995e-01   1.352e-02   66.529  < 2e-16 ***
delta   1.452e+00   1.437e-01   10.102  < 2e-16 *** delta也需要估计,这里小于2
shape   4.032e+00   2.311e-01   17.444  < 2e-16 *** Pretty heavy tail
---
```

Interpretation:

APARCH Example: BMW Returns

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  10310.16  0
 Shapiro-Wilk Test  R    W      NA        NA
 Ljung-Box Test     R    Q(10)  28.39461  0.001560335
 Ljung-Box Test     R    Q(15)  33.4418   0.00407583
 Ljung-Box Test     R    Q(20)  42.55526  0.002339603
 Ljung-Box Test     R^2  Q(10)  8.287379  0.6007884
 Ljung-Box Test     R^2  Q(15)  9.901813  0.8258702
 Ljung-Box Test     R^2  Q(20)  13.14664  0.8710042
 LM Arch Test       R    TR^2   10.14465  0.6032719

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.909858 -5.900013 -5.909863 -5.906443
```

Interpretation

APARCH Example: BMW Returns and Prediction

Example. What is the appropriate 95% confidence interval on returns based on the AR(2)-APARCH(1,1) bmw data results? What is the VaR?

Answer.

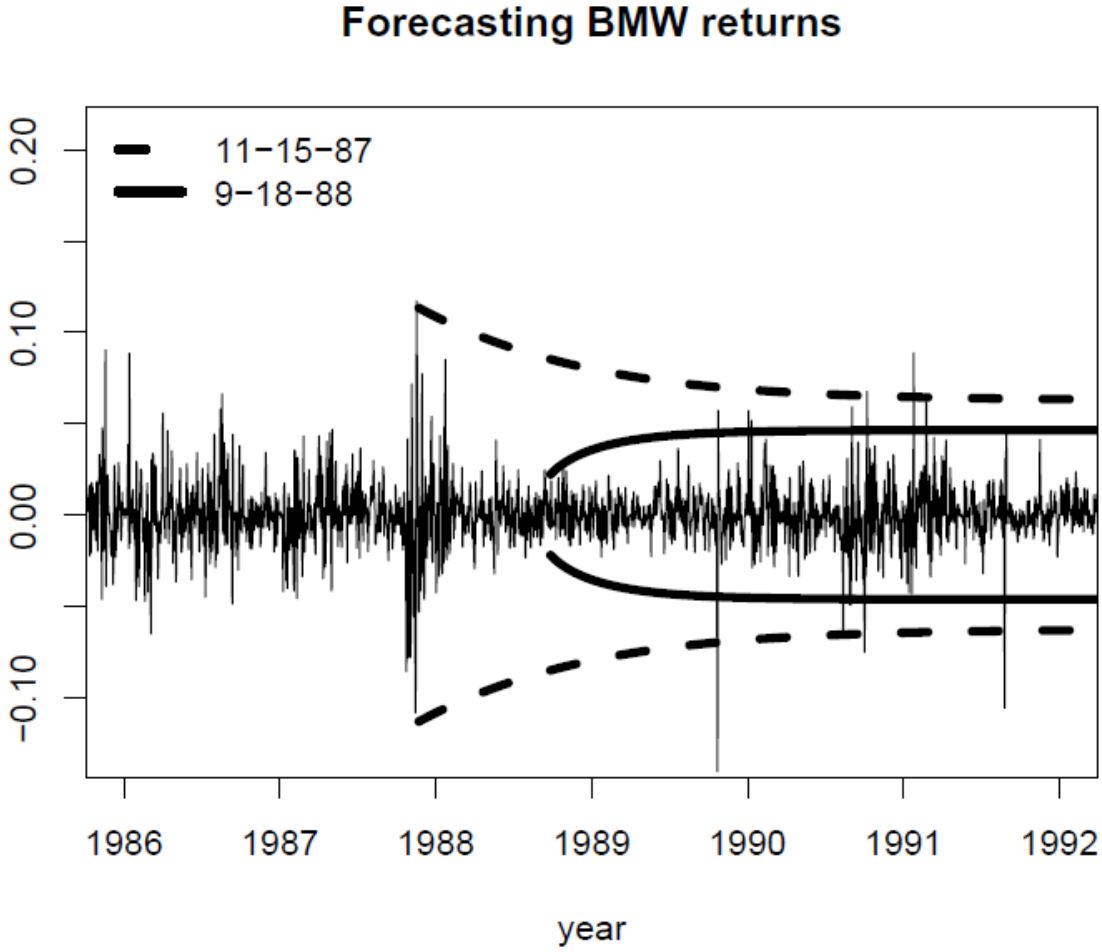
1. 拟合模型,估计参数
2. 计算ϵ_n对应t分布的0.025和0.975的分位数
3. 使用APARCH模型来估计 $\hat{\sigma}_{n+1}$

$$\hat{\sigma}_{n+1} = \left[\hat{\alpha}_0 + \hat{\alpha}_1 (|\hat{\epsilon}_n| - \hat{\gamma}\hat{\epsilon}_n)^{\hat{\delta}} + \hat{\beta}_1 \hat{\sigma}_n^{\hat{\delta}}\right)^{1/\hat{\delta}}$$

4. 使用估计值和分位数来获得置信区间

Example Predictions

Depends a lot on where you start



APARCH on NASDAQ Data

```
> Nasdaq_APARCHt <- garchFit(~arma(2,0)+aparch(1,1), data=X.ts,
 + cond.dist = c("std"), include.mean = T , include.leverage = T,
 + trace = F, algorithm = c("nlminb"), hessian = c("ropt"))
> summary(Nasdaq_APARCHt)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.33842          NA       NA       NA
ar1      0.07684          NA       NA       NA
ar2      0.05378     0.01239    4.339 1.43e-05 ***
omega    0.04321     0.01475    2.930  0.00338 **
alpha1   0.11706     0.02385    4.908 9.21e-07 ***
gamma1   0.29295     0.04230    6.926 4.32e-12 ***
beta1    0.88439     0.02452   36.062  < 2e-16 ***
delta    0.59082     0.24005    2.461  0.01385 *
shape    6.87541     0.56159   12.243  < 2e-16 ***
---
```

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  853.1819  0
 Shapiro-Wilk Test  R    W      0.9598977 9.089578e-16
 Ljung-Box Test     R    Q(10)  10.88828  0.3662882
 Ljung-Box Test     R    Q(15)  16.38039  0.3572308
 Ljung-Box Test     R    Q(20)  19.62729  0.481451
 Ljung-Box Test     R^2  Q(10)  18.15958  0.05233067
 Ljung-Box Test     R^2  Q(15)  19.53103  0.1906702
 Ljung-Box Test     R^2  Q(20)  23.88117  0.2476216
 LM Arch Test       R    TR^2   18.63074  0.09783802

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
4.690831 4.735644 4.690665 4.707878
```

AR(2)Garch(1,1) with t distribution

- Based on this which model to choose for Nasdaq??
- What additional tests ought to be done?? QQ plot or T-test
- How does one use these models for computing VaR out 1 week, or 4 weeks??

Example Problem. Suppose that $\epsilon_1, \epsilon_2, \ldots$ is a Gaussian white noise process with mean 0 and variance 1, and a_t and Y_t are stationary processes such that

$$a_t = \sigma_t \epsilon_t \quad \text{and} \quad \sigma_t^2 = 2 + 0.3 * \sigma_{t-1}^2 + 0.3 * a_{t-1}^2,$$

and

$$Y_t = 1 + 0.6 Y_{t-1} + a_t$$

(a) What type of process is a_t? Garch(1,1)

(b) What type of process is Y_t? AR(1) Garch(1)

(c) Is a_t Gaussian? If not, does it have heavy or lighter tails than a Gaussian distribution? Rigorously justify your answer.

(d) What is the variance and ACF of a_t?

(e) What is the ACF of a_t^2?

(f) What is the mean, variance, and ACF of Y_t?

E=1/(1-0.6)

Answer. .

(c): 是高斯和其他分布的混合,根据峰度可以发现much heavier tail than Gaussian

(d): $Var[\alpha_t^2] = E[\sigma_t^2] = \frac{\alpha_0}{1-\alpha_1-\beta_1} = \frac{2}{1-0.3-0.3} = 5$

$\gamma_{\alpha_t}(h) = 0$

(e): $\gamma_{\alpha_t^2}(h) = \gamma_{\sigma_t^2}(h) = (\alpha_1+\beta_1)^{|h|}\frac{\alpha_1[1-(\alpha_1+\beta_1)\beta_1]}{1+\beta_1^2-2(\alpha_1+\beta_1)\beta_1}$

(f): $E[Y_t] = \frac{1}{1-0.6} = 2.5$

$\sigma_Y^2 = \frac{\sigma_\alpha^2}{1-0.6^2} = \frac{5}{1-0.6^2}$

α_t是白噪声,因此可以直接使用AR公式

$\rho_Y(t) = |0.6|^{|t|}$

Overview of Volatility Models

Remarks.

- Primary focus of volatility models is .... ϵ_n^2
- GARCH model model the "apparent" changes in volatility via a conditional history depending on innovations process δ_n
- Stochastic volatility model is a widely used alternative to GARCH

本质上是通过delta来确保epsilon是白噪声,sigma来保存历史信息

Stochastic Volatility Models

AR-SV Model:

$$X_n = \alpha_o + \sum_{i=1}^{p} \alpha_i X_{n-i} + \epsilon_n$$

where $\{\epsilon_n\}$ is mean-zero white noise process which is modeled via

$$\epsilon = \sqrt{h_n}\delta_n$$

where δ_n is iid $\mathcal{N}(0,1)$ (or more generally a t-distribution), and $\log(h_n)$ satisfies an ARMA(p,q) model

$$\log(h_n) = \beta_o + \sum_{j=1}^{p} \phi_j \log(h_{n-j}) + \sum_{j=1}^{q} \theta_j \nu_{n-j} + \nu_n$$

和Garch类似,只是改变了历史信息的作用,且没有使用波动率而是nu

with $\{\nu_n\}$ being iid mean 0 process independent of $\{\delta_n\}$.

AR-GARCH Model:

$$\epsilon_n = \sigma_n \delta_n$$

where δ_n is iid $\mathcal{N}(0,1)$ (or more generally a t-distribution), and

$$\sigma_n^2 = \alpha_{g,o} + \sum_{j=1}^{p} \beta_{g,j}\sigma_{n-j}^2 + \sum_{j=1}^{q} \alpha_{g,j}\epsilon_{n-j}^2$$

Stochastic Volatility Models: Comments

- Positive for SV: More flexibility in modeling volatility
  - GARCH has one process δ_n and SV has 2 independent processes δ_n and ν_n
  - Volatility is independent of the mean process 只有nu没有epsilon
- Negative for SV: More difficulty/complications for estimation in SV
  - There is not a closed form for the log-likelihood (or even conditional log-likelihood)
  - Utilize Bayesian prior models on the parameters, and then utilizing Monte Carlo Markov Chain sampling techniques to sample from the posterior likelihood
- For more info/examples see Ruppert/Matteson