Nonlinear Time Series Analysis

Available R packages for GARCH modeling and a real data analysis

指導老師：徐南蓉
姓 姓 名：張明中
Outline

• Available R packages or codes for making inference for GARCH models
  ▫ FinTS
  ▫ Tseries
  ▫ fGarch

• GARCH modeling for a real data set
Part I

• Available R packages or codes for making inference for GARCH models

• GARCH modeling for a real data set
# Package--FinTS

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA</td>
<td>Arima with Ljung-Box</td>
</tr>
<tr>
<td>Acf</td>
<td>Autocorrelation Function</td>
</tr>
<tr>
<td>ArchTest</td>
<td>ARCH LM Test</td>
</tr>
<tr>
<td>AutocorTest</td>
<td>Box-Ljung autocorrelation test</td>
</tr>
<tr>
<td>FinTS.stats</td>
<td>Financial Time Series summary statistics</td>
</tr>
<tr>
<td>TsayFiles</td>
<td>List of the names of files downloaded from the &quot;Analysis of Financial Data&quot; web site.</td>
</tr>
<tr>
<td>Unitroot</td>
<td>unit root tests</td>
</tr>
<tr>
<td>apca</td>
<td>Asymptotic PCA</td>
</tr>
<tr>
<td>as.yearmon2</td>
<td>Conditionally convert x to yearmon if the conversion is unique, retaining x as names.</td>
</tr>
<tr>
<td>ch0ldata</td>
<td>financial time series for Tsay (2005, chapter 1[text])</td>
</tr>
<tr>
<td>compoundInterest</td>
<td>compute compound interest</td>
</tr>
<tr>
<td>findConjugates</td>
<td>Find complex conjugate pairs</td>
</tr>
<tr>
<td>package.dir</td>
<td>Directory of a package</td>
</tr>
<tr>
<td>plot.loadings</td>
<td>Plot loadings</td>
</tr>
<tr>
<td>plotArmaTrueacf</td>
<td>plot the theoretical ACF corresponding to an ARMA model</td>
</tr>
<tr>
<td>read.yearmon</td>
<td>Reading Monthly zoo Series</td>
</tr>
<tr>
<td>runscript</td>
<td>Run a package script</td>
</tr>
<tr>
<td>url2data</td>
<td>Create local copies of files read from urls.</td>
</tr>
</tbody>
</table>

1. ArchTest：檢查有無ARCH effect
2. AutocorTest：檢查有無autocorrelation
3. FinTS.stats：列出一些常用的統計量
Package--tseries

1. bds.test: 檢查 independence
   A test for independence based on the correlation dimension

2. garch: 配適GARCH模型
Package--fGarch

Time Series Simulation
contains functions to simulate artificial GARCH and APARCH time series processes.
Functions:
- `garchSpec`: Specifies an univariate GARCH time series model.
- `garchSim`: Simulates a GARCH/APARCH process.

Parameter Estimation
contains functions to fit the parameters of GARCH and APARCH time series processes.
Functions:
- `garchFit`: Fits the parameters of a GARCH process.
- `residuals`: Extracts residuals from a fitted 'fGARCH' object.
- `fitted`: Extracts fitted values from a fitted 'fGARCH' object.
- `volatility`: Extracts conditional volatility from a fitted 'fGARCH' object.
- `coef`: Extracts coefficients from a fitted 'fGARCH' object.
- `formula`: Extracts formula expression from a fitted 'fGARCH' object.

Forecasting
contains functions to forecast mean and variance of GARCH and APARCH processes.
Functions
- `predict`: Forecasts from an object of class 'fGARCH'.

1. `garchSpec`: 指定GARCH分佈
2. `garchSim`: simulation(可搭配garchSpec)
3. `garchFit`: 配適GARCH模型
# Packages for GARCH Modeling

<table>
<thead>
<tr>
<th>Package</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FinTS.stats</td>
<td>Financial Time Series summary statistics</td>
</tr>
<tr>
<td></td>
<td>ArchTest</td>
<td>ARCH LM Test</td>
</tr>
<tr>
<td>tseries</td>
<td></td>
<td>Package for time series analysis and computational finance</td>
</tr>
<tr>
<td></td>
<td>garch</td>
<td>Fit GARCH Models to Time Series</td>
</tr>
<tr>
<td>fGarch</td>
<td></td>
<td>Environment for teaching &quot;Financial Engineering and Computational Finance&quot;</td>
</tr>
<tr>
<td></td>
<td>garchFit</td>
<td>Univivariate GARCH Time Series Fitting</td>
</tr>
<tr>
<td></td>
<td>garchSpec</td>
<td>Univivariate GARCH Time Series Specification</td>
</tr>
<tr>
<td></td>
<td>garchSim</td>
<td>Univivariate GARCH/APARCH Time Series Simulation</td>
</tr>
</tbody>
</table>
## Package--FinTS

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinTS.stats</td>
<td>Summary statistics as in Table 1.2, Tsay (2005), including the start date, number of observations, mean, standard deviation, skewness, excess kurtosis, min and max</td>
</tr>
<tr>
<td>ArchTest</td>
<td>Lagrange Multiplier (LM) test for autoregressive conditional heteroscedasticity (ARCH) &lt;br&gt;ArchTest (x, lags=12, demean = FALSE)</td>
</tr>
</tbody>
</table>

- **x**           | numeric vector           |
- **lags**        | positive integer number of lags                           |
- **demean**      | logical: If TRUE, remove the mean before computing the test statistic. |
Package--tseries

Function | Description
---|---
garch | Fit a Generalized Autoregressive Conditional Heteroscedastic GARCH($p$, $q$) time series model to the data by computing the maximum-likelihood estimates of the conditionally normal model.
garch(x, order = c(1, 1), series = NULL, ...)

Value

A list of class "garch" with the following elements:
- order: the order of the fitted model.
- coef: estimated GARCH coefficients for the fitted model.
- n.likeli: the negative log-likelihood function evaluated at the coefficient estimates (apart from some constant).
- n.used: the number of observations of x.
- residuals: the series of residuals.
- fitted.values: the bivariate series of conditional standard deviation predictions for x.
- series: the name of the series x.
- frequency: the frequency of the series x.
- call: the call of the garch function.
- vcov: outer product of gradient estimate of the asymptotic-theory covariance matrix for the coefficient estimates.
Package--fGarch

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>garchFit</td>
<td>Estimates the parameters of an univariate ARMA-GARCH/APARCH process.</td>
</tr>
<tr>
<td></td>
<td>garchFit(formula = ~ garch(1, 1), data = dem2gbp, skew = 1, shape = 4,</td>
</tr>
<tr>
<td></td>
<td>cond.dist = c(&quot;norm&quot;, &quot;snorm&quot;, &quot;ged&quot;, &quot;sged&quot;, &quot;std&quot;, &quot;sstd&quot;, &quot;snig&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;ged&quot;, &quot;sged&quot;, &quot;std&quot;, &quot;sstd&quot;, &quot;snig&quot;, &quot;QMLE&quot;)</td>
</tr>
<tr>
<td></td>
<td>To specify for example an ARMA(2,1)-APARCH(1,1) use formula = ~arma(2,1)+</td>
</tr>
<tr>
<td></td>
<td>aparch(1,1).</td>
</tr>
</tbody>
</table>

Arguments

Note:
g=garch()
g@fit
g$fit

Value

garchFit

returns a S4 object of class "[GARCH]" with the following slots:

@call         the cell 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 "MaxLogLikelihood 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 (h = sigma^2 data).
### Package—fGarch (cont.)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>garchSpec</td>
<td>Specifies an univariate GARCH time series model. garchSpec(model = list(), cond.dist = c(&quot;norm&quot;, &quot;ged&quot;, &quot;std&quot;, &quot;snorm&quot;, &quot;sged&quot;, &quot;sstd&quot;), rseed = NULL)</td>
</tr>
<tr>
<td>garchSim</td>
<td>Simulates a univariate GARCH/APARCH time series model. garchSim(spec = garchSpec(), n = 100,)</td>
</tr>
</tbody>
</table>

# ARCH(2) - use default omega and specify alpha, set beta=0!
spec = garchSpec(model = list(alpha = c(0.2, 0.4), beta = 0))
garchSim(spec, n = 10)

# GARCH(1,2) - use default omega and specify alpha[1]/beta[2]
spec = garchSpec(model = list(alpha = 0.1, beta = c(0.4, 0.4)))
garchSim(spec, n = 10)
Simulation

> # GARCH(1,1) - specify omega/alpha/beta
>   spec = garchSpec(model = list(omega = 1e-6, alpha = 0.1, beta = 0.8))
>   garchSim(spec, n = 10)

GMT

garch
2011-04-11  0.0011061959
2011-04-12  0.0043104219
2011-04-13  0.0012216477
2011-04-14  0.0038244440
2011-04-15  0.0002189997
2011-04-16  0.0018549289
2011-04-17  0.0004236330
2011-04-18  -0.0012781357
2011-04-19  -0.0037990367
2011-04-20  -0.0045713311

> # GARCH(1,2) - use default omega and specify alpha[1]/beta[2]
>   spec = garchSpec(model = list(alpha = 0.1, beta = c(0.4, 0.4)))
>   garchSim(spec, n = 10)

GMT

garch
2011-04-11  0.0011670887
2011-04-12  -0.0051718710
2011-04-13  -0.0014409610
2011-04-14  0.0036908789
2011-04-15  0.0004331539
2011-04-16  0.0058839832
2011-04-17  -0.0051915505
2011-04-18  -0.0010442181
2011-04-19  0.0082175574
2011-04-20  0.0019800280

日期是從執行的那一天回推10天
Simulation (cont.)

> # snorm-ARCH(1) - use defaults with skew Normal
>   spec = garchSpec(model = list(beta = 0, skew = 0.8), cond.dist = "snorm")
>   garchSim(spec, n = 10)

GMT
garch
2011-04-11  4.425991e-04
2011-04-12  6.198554e-04
2011-04-13  1.830656e-03
2011-04-14  4.871050e-04
2011-04-15  6.700083e-04
2011-04-16  2.855344e-04
2011-04-17  -5.750475e-04
2011-04-18  7.915645e-04
2011-04-19  -5.870521e-04
2011-04-20  4.575665e-05
Test ARCH Effects

- GARCH(1,1)
  - Alpha=0.2
  - Beta=0.7

> library(e1071)
> ArchTest(y)

ARCH LM-test; Null hypothesis: no ARCH effects

data: y
Chi-squared = 82.737, df = 12, p-value = 1.237e-12
Modeling

- `garch(y, order=c(1,1))`
- $a_0 = 0$
- $a_1 = 0.202$
- $b_1 = 0.711$
- Goodness of fit test
  - Do not reject

```r
> summary(g11)
Call:
garch(x = y, order = c(1, 1))

Model:
GARCH(1,1)

Residuals:
       Min      1Q  Median       3Q      Max
-2.68705 -0.69943  0.06862  0.61720  2.80679

Coefficient(s):
      Estimate Std. Error   t value  Pr(>|t|)
 a0  8.997e-07  4.071e-07   2.210  0.027108 *
 a1  2.020e-01  5.809e-02   3.477  0.000508 ***
 b1  7.110e-01  7.829e-02   9.081  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Diagnostic Tests:
  Jarque Bera Test
data:  Residuals
X-squared = 1.2744, df = 2, p-value = 0.5288

  Box-Ljung test
data:  Squared.Residuals
X-squared = 2.9158, df = 1, p-value = 0.08772
Modeling (cont.)

- `garchFit(y ~ garch(1,1))`

**Conditional Distribution:**

| norm |

**Coefficient(s):**

mu     omega   alpha1   beta1  
-0.0061903  0.0107614  0.1531341  0.8059737

**Std. Errors:**

based on Hessian

**Error Analysis:**

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| mu       | -0.0061903 | 0.008462 | -0.732 0.464447 |
| omega    | 0.0107614 | 0.002838 | 3.793 0.000149 *** |
| alpha1   | 0.153134  | 0.026422 | 5.796 6.8e-09 *** |
| beta1    | 0.805974  | 0.033381 | 24.144 < 2e-16 *** |

---

**Standardised Residuals Tests:**

<table>
<thead>
<tr>
<th>Test</th>
<th>R</th>
<th>Statistic</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jarque-Bera Test</td>
<td>R</td>
<td>Chi^2</td>
<td>1059.851</td>
</tr>
<tr>
<td>Shapiro-Wilk Test</td>
<td>R</td>
<td>W</td>
<td>0.9622817</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R</td>
<td>Q(10)</td>
<td>10.12141</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R</td>
<td>Q(15)</td>
<td>17.04349</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R</td>
<td>Q(20)</td>
<td>19.29764</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R^2</td>
<td>Q(10)</td>
<td>9.062556</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R^2</td>
<td>Q(15)</td>
<td>16.07769</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R^2</td>
<td>Q(20)</td>
<td>17.50715</td>
</tr>
<tr>
<td>LM Arch Test</td>
<td>R</td>
<td>TR^2</td>
<td>9.771217</td>
</tr>
</tbody>
</table>

**Information Criterion Statistics:**

<table>
<thead>
<tr>
<th>AIC</th>
<th>BIC</th>
<th>SIC</th>
<th>HQIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.125236</td>
<td>1.136559</td>
<td>1.125228</td>
<td>1.129396</td>
</tr>
</tbody>
</table>
Diagnostic

- `plot(g11)`

Make a plot selection (or 0 to exit):

1:  Time Series
2:  Conditional SD
3:  Series with 2 Conditional SD Superimposed
4:  ACF of Observations
5:  ACF of Squared Observations
6:  Cross Correlation
7:  Residuals
8:  Conditional SDs
9:  Standardized Residuals
10: ACF of Standardized Residuals
11: ACF of Squared Standardized Residuals
12: Cross Correlation between r^2 and r
13: QQ-Plot of Standardized Residuals

結合 `par(mfrow=c())` 可畫出在一起的圖
Diagnostic (cont.)

- `garchFit(y ~ garch(1,1), cond.dist = c('std'))`

![Standardized Residuals](image1)
![ACF of Standardized Residuals](image2)
![ACF of Squared Standardized Residuals](image3)
![qstd - QQ Plot](image4)

(df = 4)
Part II

• Available R packages or codes for making inference for GARCH models

• GARCH modeling for a real data set
Data

- [http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts2/](http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts2/)
- Chapter 3: Conditional Heteroscedastic Models
- (4) Monthly returns of IBM stock: m-ibm2697.txt

```
> data # 樣本數 864
  stock
1   -0.010381
2   -0.024476
3   -0.115591
4    0.089783
5    0.036932
6    0.068493
7    0.000000
8    0.000000
9    0.065104
10   0.032258
```
```
> summary(data)

  stock
Min.   : -0.26190
1st Qu. : -0.02471
Median :  0.01185
Mean   :  0.01418
3rd Qu. :  0.05184
Max.   :  0.35117
```
Exploratory Data Analysis

> ArchTest(y)

ARCH LM-test: Null hypothesis: no ARCH effects

data: y
Chi-squared = 77.4382, df = 12, p-value = 1.269e-11
## Modeling

<table>
<thead>
<tr>
<th>Model</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garch(1,0)</td>
<td>1.224</td>
</tr>
<tr>
<td>Garch(2,0)</td>
<td>1.118</td>
</tr>
<tr>
<td>Garch(3,0)</td>
<td>1.168</td>
</tr>
<tr>
<td>Garch(1,1)</td>
<td>1.244</td>
</tr>
<tr>
<td>Garch(1,2)</td>
<td>1.123</td>
</tr>
<tr>
<td>Garch(2,1)</td>
<td>1.125</td>
</tr>
<tr>
<td>Garch(2,2)</td>
<td>1.125</td>
</tr>
</tbody>
</table>

> `AutocorTest(residuals(g))`

**Box-Ljung test**

data: residuals(g)
X-squared = 6.2093, df = 8, p-value = 0.6238
Modeling (cont.)

\[ X_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \sim t(4) \]

\[ \sigma_t^2 = 0.113 + 0.398 X_{t-1}^2 + 0.237 X_{t-2}^2 \]
Forecasting

h-step forecast

\[ E(\sigma^2_{t+h} \mid \mathcal{F}_t) = 0.113 + 0.398 E(X^2_{t+h-1} \mid \mathcal{F}_t) + 0.237 E(X^2_{t+h-2} \mid \mathcal{F}_t) \]

> forecast

<table>
<thead>
<tr>
<th>Forecast for volatility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-step</td>
</tr>
<tr>
<td>2-step</td>
</tr>
<tr>
<td>3-step</td>
</tr>
<tr>
<td>4-step</td>
</tr>
<tr>
<td>5-step</td>
</tr>
<tr>
<td>6-step</td>
</tr>
<tr>
<td>7-step</td>
</tr>
<tr>
<td>8-step</td>
</tr>
<tr>
<td>9-step</td>
</tr>
<tr>
<td>10-step</td>
</tr>
</tbody>
</table>
• Thanks for your listening