Overview

Further packages for time series analysis

- **dse** – Multivariate time series modeling with state-space and vector ARMA (VARMA) models.
- **FinTS** – R companion to Tsay (2005).
- **forecast** – Univariate time series forecasting, including exponential smoothing, state space, and ARIMA models.
- **fracdiff** – ML estimation of ARFIMA models and semiparametric estimation of the fractional differencing parameter.
- **longmemo** – Convenience functions for long-memory models.
- **mFilter** – Time series filters, including Baxter-King, Butterworth, and Hodrick-Prescott.
- **Rmetrics** – Some 20 packages for financial engineering and computational finance, including GARCH modeling in fGarch.
- **tsDyn** – Nonlinear time series models: STAR, ESTAR, LSTAR.
- **vars** – (Structural) vector autoregressive (VAR) models.

GARCH Modelling via tseries
\textbf{GARCH models}

\textit{tseries} function \texttt{garch()} fits GARCH($p$, $q$) with Gaussian innovations. Default is GARCH(1, 1):

\begin{align*}
  y_t &= \sigma_t \nu_t, \quad \nu_t \sim \mathcal{N}(0, 1) \text{i.i.d.}, \\
  \sigma_t^2 &= \omega + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \omega > 0, \alpha > 0, \beta \geq 0.
\end{align*}

\textbf{Example:} DEM/GBP FX returns for 1984-01-03 through 1991-12-31

```R
R> library("tseries")
R> mp <- garch(MarkPound, grad = "numerical", trace = FALSE)
R> summary(mp)
```

\textbf{Call:}

\begin{verbatim}
garch(x = MarkPound, grad = "numerical", trace = FALSE)
\end{verbatim}

\textbf{Model:}

GARCH(1,1)

\textbf{Residuals:}

\begin{verbatim}
      Min 1Q Median 3Q Max
-6.79739 -0.53703 -0.00264 0.55233 5.24867
\end{verbatim}

\textbf{Diagnostic Tests:}

\begin{verbatim}
Jarque Bera Test
data: Residuals
X-squared = 1100, df = 2, p-value <2e-16

Box-Ljung test
data: Squared.Residuals
X-squared = 2.5, df = 1, p-value = 0.1
\end{verbatim}

\textbf{Remarks:}

- \textit{Warning:} OPG standard errors assuming Gaussian innovations.
- More flexible GARCH modeling via \texttt{garchFit()} in \texttt{fGarch}.
Rmetrics

- Initiated and mainly developed by D. Würtz (ETH, Dept. of Theoretical Physics).
- Environment for financial engineering and computational finance.
- Currently comprises some 20 packages: fArma, fAsianOptions, fAssets, fBasics, fBonds, fCalendar, fCopulae, fEcofin, fExoticOptions, fExtremes, fGarch, fImport, fMultivar, fNonlinear, fOptions, fPortfolio, fRegression, fSeries, fTrading, fUnitRoots, fUtilities.
- Unified framework, initially designed for teaching purposes.
- Unified naming conventions via standardized wrappers. For example, arima() from stats appears as armaFit().
- We consider GARCH modelling via garchFit() from fGarch.

GARCH modeling via garchFit()

Example: DEM/GBP FX returns for 1984-01-03 through 1991-12-31
R> library("fGarch")
R> mp_gf <- garchFit(~garch(1,1), data = MarkPound, trace = FALSE)
R> summary(mp_gf)

Title: GARCH Modelling

Call:
garchFit(formula = ~garch(1, 1), data = MarkPound, trace = FALSE)

Mean and Variance Equation:
data ~ garch(1, 1)

Conditional Distribution:
norm

Coefficient(s):

|      | estimate | Std. Error | t value | Pr(>|t|) |
|------|----------|------------|---------|----------|
| mu   | -0.0061903 | 0.008462 | -0.732  | 0.464447 |
| omega| 0.010761 | 0.002838 | 3.793   | 0.000149 |
| alpha| 0.153134 | 0.026422 | 5.796   | 6.8e-09  |
| beta | 0.805974 | 0.033381 | 24.144  | < 2e-16  |

Log Likelihood:
-1107 normalized: -0.5606

Description:
Thu Mar 16 09:50:07 2017 by user: zeileis

Standardised Residuals Tests: Statistic p-Value
Jarque-Bera Test R Chi^2 0.0000

---

GARCH modeling via garchFit()

Shapiro-Wilk Test R W 0.9623 0
Ljung-Box Test R Q(10) 10.12 0.4299
Ljung-Box Test R Q(15) 17.04 0.3163
Ljung-Box Test R Q(20) 19.3 0.5026
Ljung-Box Test R^2 Q(10) 9.063 0.5262
Ljung-Box Test R^2 Q(15) 16.08 0.3769
Ljung-Box Test R^2 Q(20) 17.51 0.6198
LM Arch Test R TR^2 9.771 0.636

Information Criterion Statistics:
AIC BIC SIC HQIC
1.125 1.137 1.125 1.129

Remarks:
- Benchmark data set for GARCH(1, 1), see McCullough and Renfro (J. Economic and Social Measurement 1998).
- garchFit() hits the benchmark.
- Note that constant included by default (not possible with tseries).
- Standard errors are from the Hessinan.

---
More on garchFit()
garchFit() provides
- ARMA models with GARCH-type innovations
- Various innovation distributions: Gaussian, t, GED, including skewed generalizations.
- Several algorithms for maximizing log-likelihood, default is nlminb.
- Two methods for initializing recursions.

ARMA models with APARCH components

Mean equation is ARMA
\[ y_t = \mu + \sum_{i=1}^{m} \phi_i y_{t-i} + \sum_{j=1}^{n} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

Variance equation for APARCH(\(p, q\)) is
\[ \varepsilon_t = \sigma_t \nu_t, \quad \nu_t \sim D(0, 1) \text{ i.i.d.}, \]
\[ \sigma_t^\delta = \omega + \sum_{i=1}^{p} \alpha_i (|\varepsilon_{t-i}| - \gamma_i \varepsilon_{t-i})^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta. \]

where \( \delta > 0 \) and the leverage parameters \(-1 < \gamma_i < 1\).

APARCH comprises various GARCH-type models, including ARCH, GARCH, Taylor/Schwert-GARCH, GJR-GARCH, TARCH, NARCH, log-ARCH, …

More complex example: Ding, Granger, Engle (J. Emp. Fin. 1993)
MA(1)-APARCH(1,1) model for S&P 500 returns (17055 observations)

\[
\text{R> sp_ap <- garchFit(~ arma(0,1) + aparch(1,1), + data = ts(100 * sp500dge), trace = FALSE)}
\]

Excerpt from summary(sp_ap):

Std. Errors:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| mu       | 0.020595   | 0.006342| 3.247    | 0.00116  |
| ma1      | 0.144709   | 0.008346| 17.338   | <2e-16   |
| omega    | 0.009991   | 0.001066| 9.373    | <2e-16   |
| alpha1   | 0.083792   | 0.004343| 19.293   | <2e-16   |
| gamma1   | 0.374182   | 0.028027| 13.351   | <2e-16   |

Results broadly agree with original paper (p. 99, eq. (19)), where algorithm was BHHH. (Note: percentage returns!)
### ARMA models with APARCH components

#### Further ARCH-type models:

- **Taylor-Schwert ARCH** (compare Ding, Granger, Engle, eq. (16))

  ```R
  R> sp_tsarch <- garchFit(~ arma(0,1) + garch(1,1), delta = 1,
  + data = ts(100 * sp500dge), trace = FALSE)
  ```

- **Threshold ARCH (TARCH)**

  ```R
  R> sp_tarch <- garchFit(~ arma(0,1) + garch(1,1), delta = 1,
  + leverage = TRUE, data = ts(100 * sp500dge), trace = FALSE)
  ```

- **GJR-GARCH**

  ```R
  R> sp_garch_tarch <- garchFit(~ arma(0,1) + garch(1,1), delta = 2,
  + leverage = TRUE, data = ts(100 * sp500dge), trace = FALSE)
  ```

#### Specifying innovation distributions:

- **cond.dist** – specification of conditional distributions allowing for "dnorm", "dged", "dstd", "dsnorm", "dsged", "dsstd". Three of these ("dsnorm", "dsged", "dsstd") are skewed. – Thus

- **GARCH(1,1) with Student-t (shape parameter estimated)**

  ```R
  R> sp_garch_std <- garchFit(~ garch(1,1), cond.dist = "dstd",
  + data = ts(100 * sp500dge), trace = FALSE)
  ```

- **GARCH(1,1) with Student-t_3 (shape parameter fixed at 3)**

  ```R
  R> sp_garch_std3 <- garchFit(~ garch(1,1), cond.dist = "dstd",
  + shape = 3, include.shape = FALSE, data = ts(100 * sp500dge), trace = FALSE)
  ```

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

  ```R
  R> sp_garch_ged <- garchFit(~ garch(1,1), cond.dist = "dged",
  + shape = 1, include.shape = FALSE, data = ts(100 * sp500dge), trace = FALSE)
  ```

### Further remarks:

- More details regarding fitting process, defaults, etc. upon setting `trace = TRUE`
- `plot()` method offers 12 types of plots: time series, conditional std. dev., ACF of obs. and squared obs., residuals, ACF of residuals and squared residuals, etc.

Example: (ARMA-APARCH cont’d)

Series with superimposed conditional std. dev. is

```R
R> plot(sp_ap, which = 3)
```
Additional tools for financial engineering

- Portfolio management: \texttt{fPortfolio}, \texttt{portfolio} offer portfolio selection and optimization.
- Risk management:
  - Classical Value-at-Risk: \texttt{VaR}.
  - Extreme Value Theory models: \texttt{evd}, \texttt{evdbayes}, \texttt{evir}, \texttt{extRemes}, \texttt{ismec}, \texttt{POT}.
  - Multivariate modeling: \texttt{fCopulae}, \texttt{copula}, \texttt{fgac}
- High-frequency data: \texttt{realized}.

More complete overview in CRAN Task View Empirical Finance at

http://CRAN.R-project.org/view=Finance