Introduction to Some R Packages

We introduce two additional R packages that are useful for analyzing financial time series.

1 The rugarch package

This is a general R package for univariate financial time series analysis. A model of the package consists of the mean equation and volatility equation. The mean equation can assume ARFIMA models with ARCH-M characteristics and exogenous variables. The volatility equation can assume many volatility models, including exponential GARCH and threshold GARCH and exogenous variables. We describe the format of the mean and variance models used in the package so that readers can understand the commands and the output.
The univariate time series model of the **rugarch** model is

$$\Phi(B)(1-B)^d(r_t - \mu_t) = \Theta(B)a_t, \quad a_t = \sigma_t\epsilon_t \tag{1}$$

where B is the back-shift (or lag) operator, ϵ_t is an iid sequence with mean zero and variance 1, σ_t is the volatility, d is a real number satisfying $0 < d < 1$, which denotes the long-memory parameter equivalent to the Hurst Exponent $H - 0.5$, and $\Phi(B)$ and $\Theta(B)$ are the usual AR and MA polynomials. The package allows ϵ_t to follow standard Gaussian, standardized Student t, generalized error distribution, standardized generalized Hyperbolic, and various skew distributions. The function μ_t here is defined as

$$\mu_t = \mu + \sum_{i=1}^{m-v} \delta_i x_{it} + \sum_{i=m-v+1}^{m} \delta_i x_{it}\sigma_t + \xi\sigma_t^k,$$

where μ is a constant, x_{it} (for $i = 1, \ldots, m$) are exogenous variables and σ_t and $k = 1$ or 2.
From the formulation, the equation for μ_t is rather general. It allows the first $m - v$ exogenous variables to affect the conditional mean directly and the last v to affect the conditional mean in conjunction with volatility, the parameter ξ is the ARCH-in-mean effect with $k = 1$ for volatility in mean and $k = 2$ for variance in mean.
Recall that in the lecture notes, a univariate financial time series is written as

$$r_t = \mu_t + a_t, \quad a_t = \sigma_t\epsilon_t, \tag{2}$$

where $\mu_t = E(r_t|F_{t-1})$ is the conditional mean of the time series r_t given the information set F_{t-1} and $\sigma_t^2 = \text{Var}(r_t|F_{t-1})$ is the conditional variance,and ϵ_t is the iid innovations with mean zero and variance 1. This definition of μ_t is identical to that of the package in Equation (1) if $\Phi(B) = \Theta(B) = 1$. On the other hand, when $\Phi(B)$ contains AR lags or Θ contains MA lags, then the two μ_t become different. The difference lies in the fact that μ_t of Equation (1) is conditional mean given exogenous variables x_{it}, not given F_{t-1}. For the model in Equation (1), we have

$$E(r_t|F_{t-1}) = \sum_{i=1}^{p} \Phi_i(r_{t-i} - \mu_{t-i}) + \sum_{i=1}^{q} \Theta_i a_{t-i},$$

where p and q denote the orders of $\Phi(B)$ and $\Theta(B)$, respectively.
The **rugarch** package allows for many choices of volatility models. We only introduce those that are commonly used and discussed in the lecture.

1.1 The standard GARCH model

The standard GARCH(q, p) model, denoted by **sGARCH**, is

$$\sigma_t^2 = \omega + \sum_{i=1}^{m} \zeta_i v_{it} + \sum_{i=1}^{q} \alpha_i a_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2, \tag{3}$$

where ω is the constant term and v_{it} denotes exogenous variables. Even though the package uses (q, p), instead of (p, q) used in the lecture note, the meaning is the same. Specifically, the first element denotes the ARCH order and the second the GARCH order.

1.2 The exponential GARCH model

The exponential GARCH model of Nelson (1991), denoted by **eGARCH** in the **rugarch** package, is written as

$$\ln(\sigma_t^2) = \omega + \sum_{i=1}^{m} \zeta_i v_{it} + \sum_{i=1}^{q} [\alpha_i \epsilon_{t-i} + \gamma_i(|\epsilon_{t-i}| - E(|\epsilon_{t-i}|))] + \sum_{i=1}^{p} \beta_i \ln(\sigma_{t-i}^2), \tag{4}$$

where v_{it} denotes exogenous variables. This formulation is slightly different from those used in the lecture notes. In particular, the constant term ω differs because the mean $E(|\epsilon_{t-i}|)$ is kept in Equation (4). Another difference lies in α_i and γ_i. To see the difference, the effect of a nonnegative ϵ_{t-i} on $\ln(\sigma_t^2)$ is $\alpha_i + \gamma_i$ whereas that of a negative ϵ_{t-i} is $\alpha_i - \gamma_i$. Therefore, the leverage-effect parameter is α_i.

1.3 The GJR-GARCH model

The GJR-GARCH or Threshold GARCH model, denoted by **gjrGARCH** in the **rugarch** package, is written as

$$\sigma_t^2 = \omega + \sum_{i=1}^{m} \zeta_i v_{it} + \sum_{i=1}^{q} (\alpha_i \epsilon_{t-i}^2 + \gamma_i I_{t-i} \epsilon_{t-i}^2) + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2, \tag{5}$$

where $I_{t-i} = 1$ if and only if $\epsilon_{t-i} < 0$, denoting the indicator variable for negative ϵ_{t-i}. From the formulation, it is seen that the leverage-effect parameter is γ_i.

1.4 Model specification of the rugarch package

To specify a univariate GARCH model in the **rugarc** package, one uses the command **ugarchspec**. See below:

```
> args(ugarchspec)
function (variance.model = list(model = "sGARCH", garchOrder = c(1,1),
submodel = NULL, external.regressors = NULL, variance.targeting = FALSE),
mean.model = list(armaOrder = c(1, 1), include.mean = TRUE,
archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL,
archex = FALSE), distribution.model = "norm", start.pars = list(),
fixed.pars = list(), ...)
```

Thus, the command basically consists of **variance.model** and **mean.model**.

variance.model
The variance.model is a list that gives the type of volatility model to fit, the order of the volatility model, and the existence of any exogenous variables. The subcommands **submodel** and **variance.targeting** are not used in the lectures. Interested readers can consult the vignette associated with the **rugarch** package. The models commonly used in the lectures are:

1. Standard GARCH model: model = 'sGARCH'
2. GARCH-M model with variance in mean: model = 'sGARCH', archm = TRUE, archpow = 2.
3. GARCH-M model with standard error in mean: model = 'sGARCH', archm = TRUE, archpow = 1.
4. Exponential GARCH model: model = 'eGARCH'.
5. GJR-GARCH model: model = 'gjrGARCH'

mean.model
The default mean model is ARMA(1,1). This is not typical for financial return series. The most commonly used mean specification is
mean.model = list(armaOrder=c(0,0), include.mean=TRUE)

1.5 Demonstrations

In this subsection, I use the data set, **sp500.txt**, of the lecture to demonstrate fitting various volatility models via the **rugarch** package and compare the results with those from other packages.

```
> sp5=scan(file='sp500.txt')
> require(rugarch)
#### Specify a standard GARCH(1,1) model with mean-equation being a constant.
> spec1=ugarchspec(variance.model=list(model="sGARCH"),
        mean.model=list(armaOrder=c(0,0)))
```

```
### Estimmation command
> fit1=ugarchfit(data=sp5,spec=spec1)
> show(fit1)
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*
Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.00745    0.001538   4.8450 0.000001
omega    0.00008    0.000028   2.8287 0.004674
alpha1   0.12226    0.022102   5.5315 0.000000
beta1    0.85435    0.021811  39.1707 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.00745    0.001717   4.3382 0.000014
omega    0.00008    0.000034   2.3921 0.016754
alpha1   0.12226    0.028162   4.3412 0.000014
beta1    0.85435    0.026420  32.3373 0.000000

LogLikelihood : 1269.455

Information Criteria
------------------------------------
Akaike       -3.1956
Bayes        -3.1720
Shibata      -3.1956
Hannan-Quinn -3.1865

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.6270  0.4285
Lag[2*(p+q)+(p+q)-1][2]    0.6322  0.6347
Lag[4*(p+q)+(p+q)-1][5]    2.7319  0.4582
d.o.f=0
H0 : No serial correlation
```

```
Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      0.622  0.4303
Lag[2*(p+q)+(p+q)-1][5]     1.867  0.6502
Lag[4*(p+q)+(p+q)-1][9]     3.671  0.6450
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]    0.6535 0.500 2.000  0.4189
ARCH Lag[5]    1.3390 1.440 1.667  0.6356
ARCH Lag[7]    1.9550 2.315 1.543  0.7269

Nyblom stability test
------------------------------------
Joint Statistic:  1.0288
Individual Statistics:
mu     0.08925
omega  0.20221
alpha1 0.27132
beta1  0.16481

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.07 1.24 1.6
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value      prob sig
Sign Bias           3.1028 0.0019855 ***
Negative Sign Bias  0.9191 0.3583022
Positive Sign Bias  0.7085 0.4788445
Joint Effect       17.6096 0.0005294 ***

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     30.73      0.04324
2    30     40.27      0.07958
3    40     48.40      0.14373
4    50     61.41      0.10989

Elapsed time : 0.249603
```

```
##### Obtain fitted volatility series
> v1=sigma(fit1)
> ts.plot(as.numeric(v1))  ### Not shown.
### You can also obtain various plots
> plot(fit1) ## There are 12 choices.

######### ARCH-M model with volatility in mean
> spec2=ugarchspec(variance.model=list(model="sGARCH"),
                 mean.model=list(armaOrder=c(0,0),archm=TRUE))
> fit2=ugarchfit(data=sp5,spec=spec2)
> fit2
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002204    0.005572  0.39559 0.692409
archm   0.121406    0.123909  0.97980 0.327186
omega   0.000081    0.000029  2.81438 0.004887
alpha1  0.122520    0.022145  5.53273 0.000000
beta1   0.853627    0.022092 38.64041 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002204    0.005720  0.38538 0.699959
archm   0.121406    0.128470  0.94501 0.344654
omega   0.000081    0.000034  2.39047 0.016827
alpha1  0.122520    0.027598  4.43950 0.000009
beta1   0.853627    0.026368 32.37299 0.000000

LogLikelihood : 1269.934

Information Criteria
------------------------------------

Akaike       -3.1943
Bayes        -3.1648
```

```
Shibata      -3.1944
Hannan-Quinn -3.1829

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.7179  0.3968
Lag[2*(p+q)+(p+q)-1][2]    0.7382  0.5915
Lag[4*(p+q)+(p+q)-1][5]    2.9675  0.4130
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.6481  0.4208
Lag[2*(p+q)+(p+q)-1][5]    1.7515  0.6782
Lag[4*(p+q)+(p+q)-1][9]    3.3983  0.6918
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]    0.4789 0.500 2.000  0.4889
ARCH Lag[5]    1.0769 1.440 1.667  0.7103
ARCH Lag[7]    1.6873 2.315 1.543  0.7831

Nyblom stability test
------------------------------------
Joint Statistic:  1.4567
Individual Statistics:
mu     0.06797
archm  0.02854
omega  0.21456
alpha1 0.27174
beta1  0.16953

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            3.662 2.665e-04 ***
```

```
Negative Sign Bias   1.170 2.423e-01
Positive Sign Bias   0.504 6.144e-01
Joint Effect        22.122 6.152e-05 ***

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     27.60      0.09152
2    30     38.23      0.11736
3    40     49.82      0.11487
4    50     56.36      0.21897

##### Comparison with my R script "garchM.R"
> source(‘‘garchM.R’’)
> m2=garchM(sp5,type=2)
Maximized log-likehood:  1269.072

Coefficient(s):
         Estimate  Std. Error  t value   Pr(>|t|)
mu    2.86266e-03 5.80467e-03  0.49317  0.6218953
gamma 6.23819e-02 1.28096e-01  0.48699  0.6262635
omega 8.13893e-05 2.91723e-05  2.78995  0.0052716 **
alpha 1.21976e-01 2.21002e-02  5.51920 3.4055e-08 ***
beta  8.54361e-01 2.21412e-02 38.58699 < 2.22e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> names(m2)
[1] "residuals" "sigma.t"
>
#### Even though the ARCH-M estimates differ, but they are both insignificant!!!

#### Exponential GARCH models
> spec3=ugarchspec(variance.model=list(model="eGARCH"),
                 mean.model=list(armaOrder=c(0,0)))
> fit3=ugarchfit(data=sp5,spec=spec3)
> fit3
*---------------------------------*
*          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      0.006865    0.001552   4.4225 0.000010
omega  -0.153514    0.056374  -2.7231 0.006466
alpha1 -0.058325    0.022367  -2.6076 0.009117
beta1   0.973449    0.009257 105.1570 0.000000
gamma1  0.226980    0.033927   6.6902 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006865    0.001699   4.0395 0.000054
omega  -0.153514    0.073875  -2.0780 0.037708
alpha1 -0.058325    0.030263  -1.9273 0.053947
beta1   0.973449    0.011611  83.8412 0.000000
gamma1  0.226980    0.044154   5.1407 0.000000

LogLikelihood : 1271.916

Information Criteria
------------------------------------

Akaike       -3.1993
Bayes        -3.1698
Shibata      -3.1994
Hannan-Quinn -3.1879

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.5376  0.4634
Lag[2*(p+q)+(p+q)-1][2]    0.5377  0.6762
Lag[4*(p+q)+(p+q)-1][5]    2.3987  0.5277
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.5113  0.4746
Lag[2*(p+q)+(p+q)-1][5]    1.0762  0.8422
Lag[4*(p+q)+(p+q)-1][9]    2.7560  0.7985
d.o.f=2
```

```
Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]    0.2573 0.500 2.000  0.6119
ARCH Lag[5]    0.5748 1.440 1.667  0.8613
ARCH Lag[7]    1.0361 2.315 1.543  0.9076

Nyblom stability test
------------------------------------
Joint Statistic:  1.0587
Individual Statistics:
mu     0.1857
omega  0.3173
alpha1 0.0813
beta1  0.3000
gamma1 0.2371

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           2.9215 0.003583 ***
Negative Sign Bias  1.1913 0.233895
Positive Sign Bias  0.3372 0.736062
Joint Effect       12.9038 0.004849 ***

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     29.21      0.06272
2    30     37.17      0.14201
3    40     45.47      0.22045
4    50     65.32      0.05932

### Comparison with my R script "Egarch.R"
> source("Egarch.R")
> m3=Egarch(sp5)
Maximized log-likehood:  1271.914

Coefficient(s):
          Estimate  Std. Error  t value   Pr(>|t|)
mu      0.00686778  0.00154598  4.44234 8.8985e-06 ***
```

```
alpha0 -0.33460109  0.06682552 -5.00709 5.5260e-07 ***
alpha1  0.22698844  0.03399750  6.67662 2.4451e-11 ***
gamma  -0.25695360  0.10521202 -2.44225   0.014596 *
beta    0.97345332  0.00982935 99.03538 < 2.22e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> names(m3)
[1] "residuals"  "volatility"
>
## omega of rugarch differs from alpha0 of my Egarch program,
          because in rugarch formulation, E|epsilon_t| is not removed.
##  gamma1 of rugarch = alpha1 of my Egarch program
##  alpha1 of rugarch = alpha1 * gamma of my Egarch program.

######## GJR-GARCH or TGARCH models
> spec4=ugarchspec(variance.model=list(model="gjrGARCH"),
           mean.model=list(armaOrder=c(0,0)))
> fit4=ugarchfit(data=sp5,spec=spec4)
> fit4
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : gjrGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006680    0.001579   4.2298 0.000023
omega   0.000094    0.000032   2.9344 0.003342
alpha1  0.074050    0.026119   2.8351 0.004581
beta1   0.853944    0.022660  37.6859 0.000000
gamma1  0.079963    0.038227   2.0918 0.036459

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006680    0.001724   3.8752 0.000107
omega   0.000094    0.000041   2.2824 0.022463
alpha1  0.074050    0.037191   1.9911 0.046470
beta1   0.853944    0.029094  29.3511 0.000000
gamma1  0.079963    0.061172   1.3072 0.191150
```

```
LogLikelihood : 1271.88

Information Criteria
------------------------------------

Akaike       -3.1992
Bayes        -3.1697
Shibata      -3.1993
Hannan-Quinn -3.1878

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.5513  0.4578
Lag[2*(p+q)+(p+q)-1][2]    0.5552  0.6683
Lag[4*(p+q)+(p+q)-1][5]    2.6470  0.4753
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                      1.016  0.3135
Lag[2*(p+q)+(p+q)-1][5]     1.725  0.6845
Lag[4*(p+q)+(p+q)-1][9]     3.344  0.7011
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]    0.1069 0.500 2.000  0.7436
ARCH Lag[5]    0.6333 1.440 1.667  0.8437
ARCH Lag[7]    1.2558 2.315 1.543  0.8687

Nyblom stability test
------------------------------------
Joint Statistic:  1.1296
Individual Statistics:
mu     0.1668
omega  0.2779
alpha1 0.3147
beta1  0.2251
gamma1 0.3323
```

```
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            3.019 0.002618 ***
Negative Sign Bias   1.277 0.201847
Positive Sign Bias   0.493 0.622169
Joint Effect        14.456 0.002346 ***

Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     29.57      0.05759
2    30     35.58      0.18625
3    40     38.51      0.49227
4    50     51.43      0.37863

### Comparison with my "Tgarch11.R" script.
> source("Tgarch11.R")
> m4=Tgarch11(sp5)
Log likelihood at MLEs:
[1] 1271.972
Conditional distribution used:
norm

Coefficient(s):
         Estimate  Std. Error  t value   Pr(>|t|)
mu    6.67454e-03 1.57867e-03  4.22794 2.3584e-05 ***
omega 9.43819e-05 3.20434e-05  2.94544  0.0032249 **
alpha 7.31372e-02 2.59373e-02  2.81977  0.0048058 **
gam1  8.12536e-02 3.82393e-02  2.12487  0.0335974 *
beta  8.53859e-01 2.26580e-02 37.68460 < 2.22e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> names(m4)
[1] "residuals"  "volatility" "par"
### The results are essential the same.
```

Prediction: The command for forecast in `rugarch` is `ugarchforecast`.

```
p1 <- ugarchforecast(modelname,n.ahead=5)
p1
```

```
fitted(p1) ## mean point forecast
sigma(p1) ## volatility forecast
```

2 Stochastic Volatility Model

In this section we introduce the **stochvol** package for estimating univariate stochastic volatility model. The basic model used by the package is

$$\begin{aligned} r_t|h_t &\sim N(0, \exp h_t) & (6)\\ h_t|h_{t-1}, \mu, \phi, \sigma_\eta &\sim N[\mu + \phi(h_{t-1} - \mu), \sigma_\eta^2] & (7)\\ h_0|\mu, \phi, \sigma_\eta &\sim N[\mu, \sigma_\eta^2/(1 - \phi^2)]. & (8)\end{aligned}$$

Equation (8) specifies the unconditional distribution of the starting value of the h_t series. Equation (7) states that the h_t process follows an AR(1) model with mean μ, AR coefficient ϕ and normal innovations with mean zero and variance σ_η^2. The return r_t of Equation (6) has zero mean and variance $\exp h_t$. Consequently, $\text{Var}(r_t|F_{t-1}) = \exp h_t$ or equivalently, h_t is the logarithm of conditional variance. The volatility of r_t is then

$$v_t = \sqrt{\exp(h_t)} = \exp(h_t/2).$$

To estimate the stochastic volatility model in Equations (6)-(8), the package uses an efficient Markov chain Monte Carlo (MCMC) method. The command is **svsample**.

```
> args(svsample)

function (y, draws = 10000, burnin = 1000, priormu = c(0, 100),
    priorphi = c(5, 1.5), priorsigma = 1, thinpara = 1, thinlatent = 1,
    thintime = 1, quiet = FALSE, startpara, startlatent, expert,
    ...)
```

Therefore, the default setting is as follows: The number of MCMC iterations is 10,000. The initial burn-in is 1000. The prior for m is $N(0, 100)$, that for ϕ is Beta(5,1.5), and the prior for σ_η is 1 so that $\sigma_\eta^2 \sim$ inverted Gamma(1/2,1/2). In general, the prior for ϕ is Beta(a_0, b_0) so that

$$\begin{aligned} E(\phi) &= \frac{2a_0}{a_0 + b_0} - 1\\ V(\phi) &= \frac{4a_0b_0}{(a_0 + b_0)^2(a_0 + b_0 + 1)}, \end{aligned}$$

and the prior for σ_η is C (a positive number) so that $C \times \chi_1^2$ is Gamma$(1/2, 1/2C)$.

2.1 Illustration

Again, I use the data set **sp500.txt** to demonstrate the analysis. Figure **??** shows the time plots of the fitted volatility. The solid line is the volatility of my R script **svfit.R** whereas the red line is that of the **stochvol** package. The two volatility series are close.

```
> require(stochvol)
### Remove sample mean, because the program requires zero mean.
> rtn=sp5-mean(sp5)
> sv1=svsample(rtn)

Calling GIS_C MCMC sampler with 11000 iter. Series length is 792.
Timing (elapsed): 6.41 seconds.
1716 iterations per second.

Converting results to coda objects... Done!
Summarizing posterior draws... Done!

> names(sv1)
[1] "para"     "latent"   "latent0"  "y"        "runtime"  "priors"   "thinning"
[8] "summary"
### latent contains the draws of the stochastic volatility

### posterior means and standard errors of the three parameters.
> dim(sv1$para)
[1] 10000     3
> apply(sv1$para,2,mean)
        mu        phi      sigma
-6.1934532  0.9627546  0.2296903
> sqrt(apply(sv1$para,2,var))
        mu        phi      sigma
0.27048224 0.01431130 0.03708176
### Obtain fitted volatility
> dim(sv1$latent)
[1] 10000   792
> ht=apply(sv1$latent,2,median)
> v1=exp(ht/2) ## volatility
> ts.plot(v1)
```