The Report Committee for Chang Sun
Certifies that this is the approved version of the following report:

Modeling Stock Volatility with Stochastic ARCH, GARCH and Stochastic Volatility model

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Stephen Walker

Michael Daniels

Modeling Stock Volatility with Stochastic ARCH, GARCH and Stochastic Volatility model

by

Chang Sun, B.A.; M.Ed.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Statistics

The University of Texas at Austin

December 2016

Abstract

Modeling Stock Volatility with Stochastic ARCH, GARCH and Stochastic Volatility model

Chang Sun, M.S.Stat.

The University of Texas at Austin, 2016

Supervisor: Stephen Walker

Modeling volatility within the log stock return is key to the stock price prediction. Despite numerous researches that modeled the volatility with conditional heavy-tailed error distributions, the unconditional distribution remains unknown. In this report, we use and follow the method introduced by Pitt and Walker (2005) by assigning a Student-t distribution for the marginal density of log return and constructing three models respectively, with similar structures to Autoregressive Conditional Heteroskedasticity (ARCH), Generalized ARCH (GARCH) and Stochastic Volatility model in a Bayesian way. We demonstrate the capability of the three models for stock price prediction with S&P 500 index and show that all our models outperform the standard GARCH model (Bollerslev, 1986).

Table of Contents

Introduction

Stock trading is one of the most common investment activities. Investors and Researchers continuously develop various stock analysis methods to help them predict future equity prices. One of the common ways to model the equity future price is to analyze the current financial information and news (Milosevic, 2016). A number of analysis methods are based on financial balance sheets and various ratios which are often used to describe the financial health of the company (Milosevic, 2016). Experienced analysts could apply some mathematical models to evaluate company's intrinsic value, such as the well-known Graham number or Graham's criteria (Graham, 1949). However, the increased and continuously changing volatility in the current market make it hard to find a company that satisfy Graham's principles on today's stock exchanges (Milosevic, 2016). Uncertainty has become central to much of the modern financial theory (Bollerslev, Chou and Kroner, 1992). For example, most asset pricing theories involve measuring the risk premium. In option pricing, the uncertainty associated with the future price of the underlying asset is the most important determinant in the pricing function (Bollerslev, Chou and Kroner, 1992).

It is well recognized that the uncertainty of the speculative prices, as changing through time, is measured by its variances and covariance (Bollerslev, Chou and Kroner, 1992). For this report, we require our time series to be strictly stationary so we start from modeling the stock return instead of the prices directly. Under such condition, one of the most popular models for characterizing the changing variances is the Autoregressive Conditional Heteroskedasticity (ARCH) model of Engle (1982), which allows the conditional variance to change over time as a function of the past errors. Later on, a Generalized Autoregressive Conditional Heteroskedastic (GARCH) model was introduced by Bollerslev (1986) to allow for both longer memory and more flexible lag structure. Another commonly used model, first introduced by Taylor (1982), is called discrete time stochastic volatility model in which the error term has the variance that follows a stochastic process. Since then, many extensions to these models were published. One of the interesting and also widely recognized problem was that the unconditional or marginal return distributions tend to have flat tails rather than the normal distribution. Although the unconditional error in GARCH(p, q) model with conditional normal errors given by Bollerslev (1986) show somewhat flatter tails, it couldn't fully account for the leptokurtosis (Bollerslev, Chou and Kroner, 1992). Bollerslev (1987) suggested to use the standardized t-distribution as the conditional error distribution in the ARCH/GARCH model. Other parametric densities for the conditional distribution in the estimation of ARCH include normal-Poisson mixture distribution in Jorion (1989), the power exponential distribution in Baillie and Bollerslev (1989) and the generalized exponential distribution in Nelson (1990) (Bollerslev, Chou and Kroner, 1992).

Despite such appealing constructions, the unconditional density of the stationary process remains unknown. In this report, we take an approach introduced by Pitt and Walker (2005) which modeled the volatility of the strictly stationary time series with specified marginal density and linear expectations. Specifically, Pitt and Walker (2005) specified the stationary marginal density of the stock return as a Student t distribution with degree of freedom being estimated through Monte Carlo Markov Chain (MCMC) sampling. Three types of models, ARCH(1), GARCH(1,1) and Stochastic Volatility model with Student-t marginal density are introduced. By using the auxiliary variables, Pitt and Walker (2005) showed that the likelihood estimation becomes very efficient and it is easy to obtain linear expectation of the volatility, which make the autocorrelations of the series and the point forecasting several steps ahead also possible. In this report, we will show in detail how such three types of models are constructed and used for volatility prediction. Pitt and Walker (2005) examined their GARCH(1,1) on daily continuously compounded percentage returns of US dollars against five currencies and compared the volatility estimation with the standard GARCH(1,1) (Bollerslev, 1986). Each of their series consisted of only 1,000 data points. In our report, we will examine all three models introduced on S&P 500 index daily closing price from January 1st 2005 to November 11th 2016, a larger dataset with around 3,000 data points. Given our goal is to predict stock prices, the performance of those three models will be evaluated based on accuracy of stock price prediction, compared with that from the same benchmark model, standard GARCH(1,1) by Bollerslev (1986).

This article is organized as follows: In section 2, we give the rationale to model the stock return, even though our primary interest is the price. We also illustrate the reason for modeling the marginal density as a Student t distribution. In section 3, we briefly review the traditional ARCH, GARCH and discrete time stochastic volatility model. Then show how our models are different from those by incorporating the Student t marginal density with the auxiliary variable and how to achieve the parameter estimation and forecast in a Bayesian setting. In section 4, we show the results for parameter estimation from MCMC sampling based on real stock return data, then perform the model checking and stock price prediction. We also compare our result with that from the traditional GARCH model. In section 5, we give a brief summary to the methods introduced by Pitt and Walker (2005) and discuss the model preference regarding to the computation efficiency.

Stock return and its marginal density

There are basically two reasons to model the stock return rather than the prices. Financially speaking, the stock market may be considered close to perfectly competitive, so that the size of the investment does not affect price changes (Campbell, Lo, and MacKinley 1997, chap. 1). Thus the return is a complete and scale-free summary of the investment opportunity (Campbell, Lo, and MacKinley 1997, chap. 1). Statistically speaking, returns have more attractive properties than prices such as stationarity and ergodicity (Campbell, Lo, and MacKinley 1997, chap. 1). By stationarity, we mean the series has finite variation, constant mean and the covariance does not depend on time, but only on the time difference.

DEFINITION OF THE STOCK RETURN

Given the nice scale-free property, the stock return can be constructed only through the stock prices. We follow the definitions given by Campbell, Lo, and MacKinley (1997). Denote P_t as the stock price at time t, then the *simple net return*, R_t between $t-1$ and t is simply:

$$R_t = \frac{P_t}{P_{t-1}} - 1 \tag{1}$$

This definition is quite straightforward and serves as a guideline to calculate the annualized multiyear returns. However, it is often difficult to manipulate the geometric average involved in such calculation, so another notion, *continuously compounded return* or *log return* r_t of an asset is introduced in Finance, which is often used in mathematical modeling as rate of return and is defined as:

$$r_t \equiv \log(1+R_t) = log\frac{P_t}{P_{t-1}} = logP_t - logP_{t-1} \tag{2}$$

In this report, we use the log return rather than the simple net return to construct the dataset. We denote $y_t = logP_t - logP_{t-1}$ instead of using r_t. For prediction purpose, P_t could be written as:

$$P_t = P_{t-1}e_t^{y_t} \tag{3}$$

MARGINAL DENSITY

The early-stage belief that the stock return could be adequately characterized by the normal distribution was rejected by many studies later which have shown the empirical distributions of such returns have more kurtosis (i.e., “flatter tails”) than that predicted by the normal distribution (Blattberg and Gonedes, 1974). Various comparisons of the Student t distribution and stable distributions were made by Blattberg and Gonedes (1974) who showed that the Student model outperformed the symmetric-stable models. Of course there are plenty of choices on the marginal density besides the Student t distribution, such as a discrete mixture of normal distributions (Kon, 1984) and normal inverse Gaussian distributions (Barndorff-Nielsen, 1997), but using the Student t has some nice properties. First, it is fairly simple to use, with only one parameter, that is, the degree of freedom to be estimated. Second, the Student model allows the use of well-defined density functions, thus, the likelihood function of the Student model can be expressed in closed form (Blattberg and Goedes, 1974), then either Bayesian inference or maximum-likelihood estimates may be obtained straightforwardly.

Algorithms

MODELS OF ARCH (1) TYPE

The ARCH model proposed by Engle (1982) has all discrete time error term $\{\varepsilon_t\}$ of the form

$$\varepsilon_t = z_t \sigma_t, \tag{4}$$

where z_t is $\mathrm{i,i,d.}$ normal random variable, with $E(z_t) = 0$, $\mathrm{Var}(z_t) = 1$, with σ_t a time-varying, positive and measureable function of the time $t-1$ information set (Bollerslev, Chou and Kroner, 1992). By definition, ε_t is serially uncorrelated with mean zero (Bollerslev, Chou and Kroner, 1992). Usually ε_t corresponds to the innovation in the mean for some stochastic process, but in our report, we make ε_t itself observable, that is, $\varepsilon_t = y_t$.

Thus, the ARCH (1) model has the form:

$$y_t = z_t \sigma_t \tag{5}$$

$$w_t = \sigma_t^2 = a + b y_{t-1}^2, \quad a > 0, b \geq 0, \tag{6}$$

Then we have:

$$y_t = \sqrt{a + b y_{t-1}^2} z_t \tag{7}$$

with the parameter constraints ensuring that the variance remains positive (Pitt and Walker, 2005). The w_t, in our case represents the volatility, which given y_{t-1}^2 , is deterministic shown in (6). The estimation of those parameters can be achieved through maximum likelihood (ML) or Generalized Method of Moments (GMM) (Bollerslev, Chou and Kroner, 1992). In this report, we follow the method introduced by Pitt and Walker (2005) to incorporate the marginal Student t distribution constructed through latent variable in the ARCH (1) type model and use MCMC to derive Bayesian inference for the parameters.

First, we specify a joint density

$$\begin{aligned} f_{Y,W}(y, w) &= f_{Y|W}(y|w) * f_W(w) \\ &= N(0, w) * Ig(\frac{\nu}{2}, \frac{\nu\beta^2}{2}) \end{aligned} \tag{8}$$

By definition, the conditional error distribution is normal $\mathrm{y_t} \sim N(0, w_t)$. The reason to use inverse-gamma distribution for $\mathrm{w_t}$, as a latent variable in this case, is to guarantee the marginal density as a Student-t distribution. As can be seen from below, when we integrate out $\mathrm{w_t}$, $\mathrm{y_t}$ follows a scaled Student-t distribution $\mathrm{t_\nu}(0, \beta^2)$. The parameters ν and β^2 will be estimated through MCMC updates.

$$\mathrm{f_Y}(y) = \int f_{Y|W}(y|w) * f_W(w) dw$$
$$= \frac{1}{\sqrt{\nu\beta^2\pi}} * \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{\mathrm{y}^2}{\nu\beta^2}\right)^{-\frac{\nu+1}{2}} \tag{9}$$

Remember our goal is to construct a Markov process $\{\mathrm{y_t}\}$. Following the fashion specified by Pitt and Walker (2005), we first generate $\mathrm{w_1}$ from $\mathrm{f_W}(w)$ and then $\mathrm{y_1} \sim f_{Y|W}(y|w_1)$. We then generate $\mathrm{w_t} \sim f_{W|Y}(w|y_{t-1})$ and $\mathrm{y_t} \sim f_{Y|W}(y|w_t)$ for $\mathrm{t} = 2, 3, 4$ ..., which gives the dependency structure for ARCH(1) model shown in Figure 1 (Pitt and Walker, 2005).

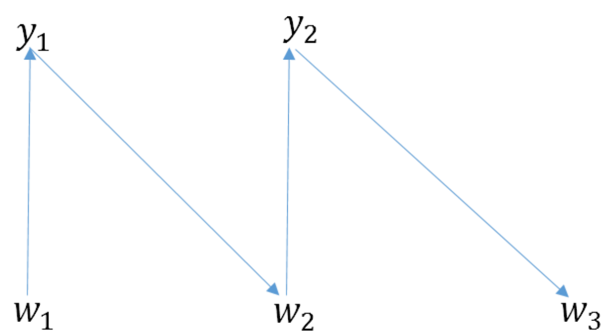

Figure 1: Dependency structure of ARCH(1).

The conditional density $\mathrm{f_{W|Y}}(w|y)$ is easy to obtain as:

$$\mathrm{f_{W|Y}}(w|y) \propto f(w, y)$$
$$\propto \mathrm{w}^{-\frac{\nu+1}{2}-1} e^{-\left(\frac{1}{2}y^2 + \frac{\nu\beta^2}{2}\right)w^{-1}} \tag{10}$$

which is an inverse-gamma distribution $\mathrm{Ig}(\frac{\nu+1}{2}, \frac{1}{2}y^2 + \frac{\nu\beta^2}{2})$. Compared with (6), obviously, the w_t in Pitt and Walker (2005)'s model given y_{t-1} is a random variable. Thus this ARCH(1) model, is actually a stochastic ARCH type model. Based on this conditional density, we are able to write the predictive density $\mathrm{p}(\mathrm{y_t}|y_{t-1})$ explicitly:

$$p(y_t|y_{t-1}) = \int f_{Y|W}(y_t|w_t)f(w_t|y_{t-1})dw_t$$

$$\propto \sqrt{\frac{y_{t-1}^2 + \nu\beta^2}{1+\nu}}\left[\frac{\Gamma\left(\frac{\nu+2}{2}\right)}{\Gamma\left(\frac{\nu+1}{2}\right)} * \frac{1}{\sqrt{\nu+1}}\left(1+\frac{S^2}{\nu+1}\right)^{-\frac{\nu+2}{2}}\right] \quad (11)$$

where $S^2 = (\nu+1)\frac{y_t^2}{y_{t-1}^2+\nu\beta^2}$. Such predictive density implies that

$$y_t = \sqrt{\frac{y_{t-1}^2 + \nu\beta^2}{1+\nu}}S_{\nu+1} \quad (12)$$

where $S_{\nu+1} \sim t_{\nu+1}$, a Student-t variable with $\nu + 1$ degree of freedom. Compared with (7), it is obviously that this model is different from the ARCH(1) of Engle (1982), but the likelihood function is still able to obtain directly through the predictive density above, which make it fairly easy to obtain the posterior of parameters.

For the choice of priors for parameters, we give an equal probability for ν from 3 to 20. The degree of freedom over 20 would make the Student-t distribution more like a Gaussian distribution while below 3 would lead the variance to infinity. The prior for β^2 is given by Gamma(1,1). We leave the derivation of the posterior of ν and β^2 in the Appendix.

MODELS OF GARCH(1, 1) TYPE

The extension of the ARCH process to the GARCH process is much similar to the extension of standard AR model to the general ARMA process (Bollerslev, 1986). Compared with ARCH(q) model, GARCH(p, q) process allows more flexible lag structure and longer memory. The GARCH(p, q) proposed by Bollersleve (1986) has the following structure:

$$\varepsilon_t \sim N(0, \sigma_t^2) \quad (13)$$

$$\sigma_t^2 = a + \sum_{i=1}^{q} b_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} c_i \sigma_{t-i}^2 \quad (14)$$

where $p \geq 0, q > 0; a > 0, b_i \geq 0, i = 1, 2, \ldots, q; c_i \geq 0, i = 1, 2, \ldots, p$.

To keep the notation consistent with our previous sections, the standard GARCH(1,1) from Bollerslev (1986) can be written as:

$$y_t \sim N(0, w_t) \tag{15}$$

$$w_t = a + b y_{t-1}^2 + c w_{t-1} \tag{16}$$

where w_t is still deterministic. Again, the model introduced by Pitt and Walker (2005) differs from the structure above because the marginal density of y_t and w_t are kept fixed and known (Pitt and Walker, 2005). To enable a longer dependency, a new auxiliary variable z_t is introduced into the model, but it does not affect the marginal distribution of y. The joint density has the form:

$$\begin{aligned} f_{Y,W,Z}(y, w, z) &= f_{Y|W}(y|w) * f_W(w) * f_{Z|W}(z|w) \\ &= N(0, w) * Ig\left(\frac{\nu}{2}, \frac{\nu\beta^2}{2}\right) * Ga(\alpha, w^{-1}) \end{aligned} \tag{17}$$

where w_t is still an inverse-gamma process and we define $z_t|w_t$ follows a Gamma distribution. Our goal is to construct a Markov process $\{y_t, w_t\}$ with parameters $\alpha, \beta^2$and ν being estimated through MCMC.

Similar to the update schema for ARCH(1) model, we first generate w_1 from $f_W(w)$ and then $y_1 \sim N(0, w_1), z_1 \sim Ga(\alpha, w_1^{-1})$. Then we generate w_2 from $f_{W|Y,Z}(w_t|y_{t-1}, z_{t-1}), y_t \sim N(0, w_t)$ and $z_t \sim Ga(\alpha, w_t^{-1})$ for $t = 2, 3, \ldots$. Such dependency structure for GARCH(1, 1) model can be shown in Figure 2 (Pitt and Walker, 2005)

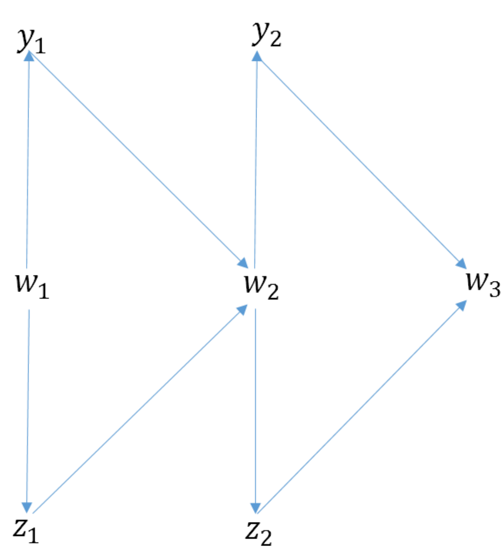

Figure 2: Dependency structure of GARCH(1,1).

The conditional distribution of $f_{W|Y,Z}(w|y,z)$ can be obtained through:

$$
\begin{aligned}
f_{W|Y,Z}(w|y,z) &\propto f_{W,Y,Z}(w,y,z) \\
&\propto w^{-\left(\frac{1+\nu}{2}+\alpha\right)-1} e^{-\left(\frac{y^2+\nu\beta^2}{2}+z\right)w^{-1}}
\end{aligned} \tag{18}
$$

which is an inverge-gamma distribution. Given $w_t|y_{t-1},z_{t-1} \sim Ig(\frac{1+\nu}{2}+\alpha, \frac{y_{t-1}^2+\nu\beta^2}{2}+z_{t-1})$ as a random variable, we know Pitt and Walker (2005)'s model is not the same as Standard GARCH(1,1) proposed by Bollerslev (1986), but more like a stochastic GARCH(1,1). Unfortunately, we are unable to integrate out z_t and obtain the predictive density $p(y_t|y_{t-1},w_{t-1})$ explicitly. To make prediction for the next data point, we have to use the Markov chain specified above. For example, to predict y_{t+1}, we first sample w_{t+1} from $f_{W|Y,Z}(w_{t+1}|y_t,z_t)$, then we sample y_{t+1} from $N(0,w_{t+1})$. The likelihood function can be expressed as:

$$
L_\theta = \prod_{t=2}^{n} f_{Y|W}(y_t|w_t) f_{Z|W}(z_t|w_t) f_{W|Y,Z}(w_t|y_{t-1},z_{t-1}) \tag{19}
$$

where the posterior of parameters and w_t and z_t can be obtained. Here, we only illustrate the posterior of w_t to show the longer-range dependence allowed from z_t. We denote the set of parameters (α, β^2, ν) as θ:

$$
\begin{aligned}
P_\theta(w_t|y_t,y_{t-1},z_t,z_{t-1}) &\propto f_{Y|W}(y_t|w_t) f_{Z|W}(z_t|w_t) f_{W|Y,Z}(w_t|y_{t-1},z_{t-1}) \\
&\propto w_t^{-\left(\frac{2+\nu}{2}+\alpha\right)-1} e^{-\left(\frac{y_t^2+y_{t-1}^2+\nu\beta^2}{2}+z_t+z_{t-1}\right)w_t^{-1}}
\end{aligned} \tag{20}
$$

which is an inverge-gamma distribution $Ig(\frac{2+\nu}{2}+\alpha, \frac{y_t^2+y_{t-1}^2+\nu\beta^2}{2}+z_t+z_{t-1})$. Compared with the ARCH(1) model dependency structure, w_t in GARCH(1, 1) depends not simply on y_t, y_{t-1}, but also on z_t and z_{t-1}. Since z_t and z_{t-1} are auxiliary variables only related with w, they only contain the information set of w. Thus, w_{t-1} can feed back in predicting w_t.

The posteriors for the parameters β^2 and ν are derived from the log-likelihood function, with the same prior as those in ARCH(1). The new parameter α is updated in the similar way as β^2 and ν. We also give it Gamma(1, 1) as the prior. In fact, the choice of priors for our three models turns out to be not important at all, since the prior contribution to the posterior estimation is extremely small compared with the likelihood given the large dataset we will use. Their posterior derivations are given in Appendix.

STOCHASTIC VOLATILITY MODEL

Stochastic volatility (SV) models are motivated economically by the mixture-of-distribution hypothesis (MDH) proposed by Clark (1973) that the asset returns follow a mixture of normal distributions with a mixing process depending on unobserved information arrival process (Hautsch and Ou, 2008). The basic idea behind this type of model is that the return volatility follows its own stochastic process updated by some unobserved innovations, with no influence from the asset return itself. Such structure, therefore, is different from GARCH and ARCH type models.

The standard SV model by Taylor (1982) models the log return y_t as:

$$\mathrm{y_t} = \mathrm{e}^{\frac{\mathrm{h_t}}{2}} u_t \tag{21}$$

$$\mathrm{h_t} = \mu + \phi(h_{t-1} - \mu) + \eta_t \tag{22}$$

where $\mathrm{u_t} \sim N(0, 1), \eta_t \sim N(0, \sigma_\eta^2), |\phi| < 1$, and $\mathrm{h_t}$ is the log volatility assumed to follow a AR(1) process. The unconditional distribution of h_t is:

$$h_t \sim N(\mu, \sigma_h^2) \tag{23}$$

$$\sigma_h^2 = \frac{\sigma_\eta^2}{1 - \phi^2} \tag{24}$$

Such SV model, like ARCH and GARCH type models, is able to model the typical volatility for most financial and time series, but the model implied kurtosis is often far too small because of the inflexible normal-log normal mixture structure (Hautsch and Ou, 2008). Chib, Nardari and Shephard (2002) introduced an extension of Taylor (1982)'s model, which assumes, $\mu_t$follows a standardized t distribution with degree of freedom ν: $\mu_t \sim t_\nu$. Thus, the conditional density $y_t | h_t$is a Student-t distribution, but the marginal density of y_t, of course, is not.

The SV type model introduced by Pitt and Walker (2005), different from models above, is very similar to their GARCH(1, 1) model except that y does not form the evolution of the volatility w. The volatility w is updated only through the auxiliary variable z, rather than the return y:

$$f_{W|Z}(w|z) \propto f_{Z|W}(z|w) f_W(w)$$
$$\propto Ga(\alpha, w^{-1}) Ig\left(\frac{\nu}{2}, \frac{\nu\beta^2}{2}\right)$$
$$\propto w^{-\left(\frac{\nu}{2}+\alpha\right)-1} e^{-\left(z+\frac{\nu\beta^2}{2}\right)w^{-1}} \quad (25)$$

Therefore, $w_t|z_{t-1}$ follows an inverse-gamma distribution $Ig(\frac{\nu}{2} + \alpha, z_{t-1} + \frac{\nu\beta^2}{2})$. Our goal is still to construct a Markov process $\{y_t, w_t\}$ with parameters α, β^2 and ν being estimated through MCMC. Similar to the update schema for GARCH(1, 1) model, we first generate w_1 from $f_W(w)$ and then $y_1 \sim N(0, w_1), z_1 \sim Ga(\alpha, w_1^{-1})$. Now we generate w_2 from $f_{W|Y,Z}(w_t|z_{t-1})$, update $y_t \sim N(0, w_t)$ and $z_t \sim Ga(\alpha, w_t^{-1})$ for $t = 2, 3, \ldots$. Such dependency structure for our SV model can be shown in Figure 3 (Pitt and Walker, 2005):

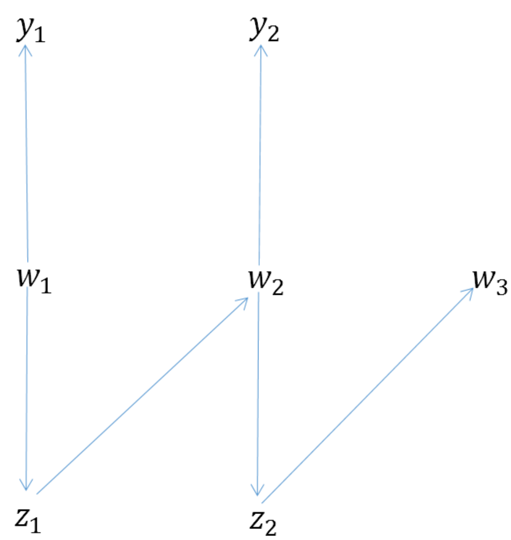

Figure 3: Dependency structure of SV model.

Similar to GARCH(1, 1), we are unable to obtain the predictive density $f(y_t|y_{t-1}, w_{t-1})$ explicitly. So to make prediction for the next data point, we use the above SV model Markov chain. The likelihood function can be expressed as:

$$L_\theta = \prod_{t=2}^{n} f_{Y|W}(y_t|w_t) f_{Z|W}(z_t|w_t) f_{W|Y,Z}(w_t|z_{t-1}) \quad (26)$$

The exact formula for the likelihood function can be found in Appendix. From the likelihood function, we could derive the posterior of w_t as:

$$\begin{aligned} \mathrm{P}_\theta(w_t|\mathrm{y}_t, z_t, z_{t-1}) &\propto f_{Y|W}(y_t|w_t) f_{Z|W}(z_t|w_t) f_{W|Y,Z}(w_t|z_{t-1}) \\ &\propto w_t^{-\left(\frac{1+\nu}{2}+2\alpha\right)-1} e^{-\left(\frac{y_t^2+\nu\beta^2}{2}+z_t+z_{t-1}\right)w_t^{-1}} \end{aligned} \tag{27}$$

which is an inverse-gamma distribution $Ig(\frac{1+\nu}{2}+2\alpha, \frac{y_t^2+\nu\beta^2}{2}+z_t+z_{t-1})$.

The MCMC updates for parameters and z_t are very similar to those for GARCH(1, 1) and details are given in Appendix.

Application

DATA

We have found that many papers include the S&P 500 index to apply their volatility models. See e.g. Bollerslev (1987), Chib, Nardari and Shephard (2002). To make the result comparable, we also use the S&P 500 index, specifically, its daily closing prices from Yahoo Finance as our data. To access the data, we use an R package called "TTR", which allow us to fetch the stock data from Yahoo Finance website. The "TTR" package is often used to conduct technical trading analysis in R and it is enhanced from commonly used R package "quantmod". To examine the prediction performance of our models, we split the data into the training set and test set. The training set contains the prices from January 1st, 2005 to December 31st, 2009, intentionally covering the 2008 financial crisis. We can expect the clusters of very high volatilities in the training set. The test set contains the prices from January 1st, 2010 to November 11th, 2016. Figure 4 shows the histograms of the log return from training and test dataset, which look pretty similar to Student-t distribution.

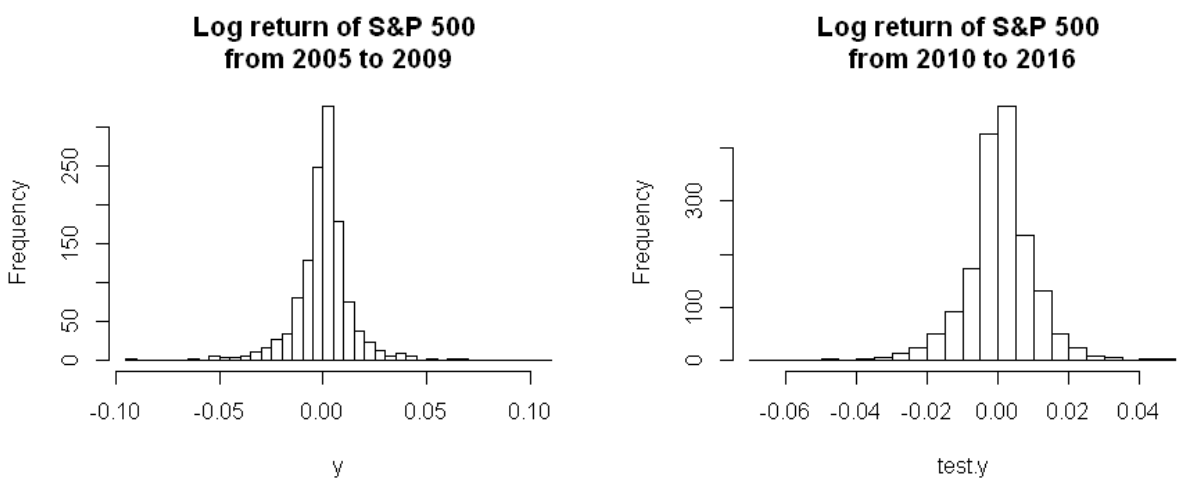

Figure 4: Histograms of log returns from the training and test set. Histogram on the left plots the 1,258 log returns from S&P 500 index from January 1st, 2005 to December 31st, 2009. Histogram on the right plots 1,728 log returns from S&P 500 index from January 1st, 2010 to November 11th, 2016

PARAMETER ESTIMATION

For ARCH(1) model, the parameter update structure is fairly simple. Since the predictive density $p(y_t|y_{t-1})$ is explicitly available, the likelihood function does not have w. Therefore, the posterior of ν and β^2 are directly updated from the data and we do not necessarily update w in MCMC. We ran the MCMC sampling for 10,000 iterations. For each iteration, we sampled ν from its posterior first. Next sampled β^2 from its posterior with the updated ν. Then we used the latest β^2 to update ν again. After 10,000 iterations, we obtained the posterior mean of β^2 as 6.78e-5 with standard deviation 4.58e-6, the mean of ν samples as 3.004 with standard deviation 0.019997.

For GARCH(1,1) model, the update of parameter relies on the updates of w and z since the predictive density $p(y_t|y_{t-1}, w_{t-1})$ is not explicitly available and thus the likelihood function have z. Therefore, to update parameters α, β^2 and ν, we also need to initialize w and z which is done by sampling from w. We ran the MCMC for 10,000 iterations. For each iteration, we sampled ν from its posterior first. Next sampled β^2 from its posterior with the updated ν. Then we use the latest β^2 and ν to update α. After parameters were updated, we sampled z from its posterior and finally update w given all latest variables. After 10,000 iterations, we obtained the posterior means (and standard deviations) of β^2, α and ν as 6.69e-5 (4.73e-6), 0.045 (0.0013) and 3 (0).

For SV model, the estimation procedure for parameters, w and z are exactly the same as GARCH(1,1). After 10,000 iterations, the posterior means (and standard deviations) of β^2, α and ν were 6.47e-5 (3.66e-6), 0.036 (0.00098) and 3 (0).

Figure 5 shows the MCMC sampled parameters after it is converged. Compared with ARCH(1) parameter updates which mix and converge less than 100 iterations, it usually takes much longer time for the parameters in GARCH(1,1) and SV model to converge, depending on the initial values. For example, β^2 in GARCH(1,1) given an initial value of 0.0005 took about 3000 iterations to converge and in SV model with the same initial value took about 1500 iterations. A larger initial value will further slow down the convergence. There are two main reasons for such slow convergence. First, we need to update three more variables α, w and z in GARCH(1,1) and SV model. Second, most of the variables do not have an easy form of posterior to sample from, thus Metropolis-Hasting algorithm is often used. The variance of proposal in our case is the key adjustment that often decide the variability of sampled parameters, also, the convergence speed.

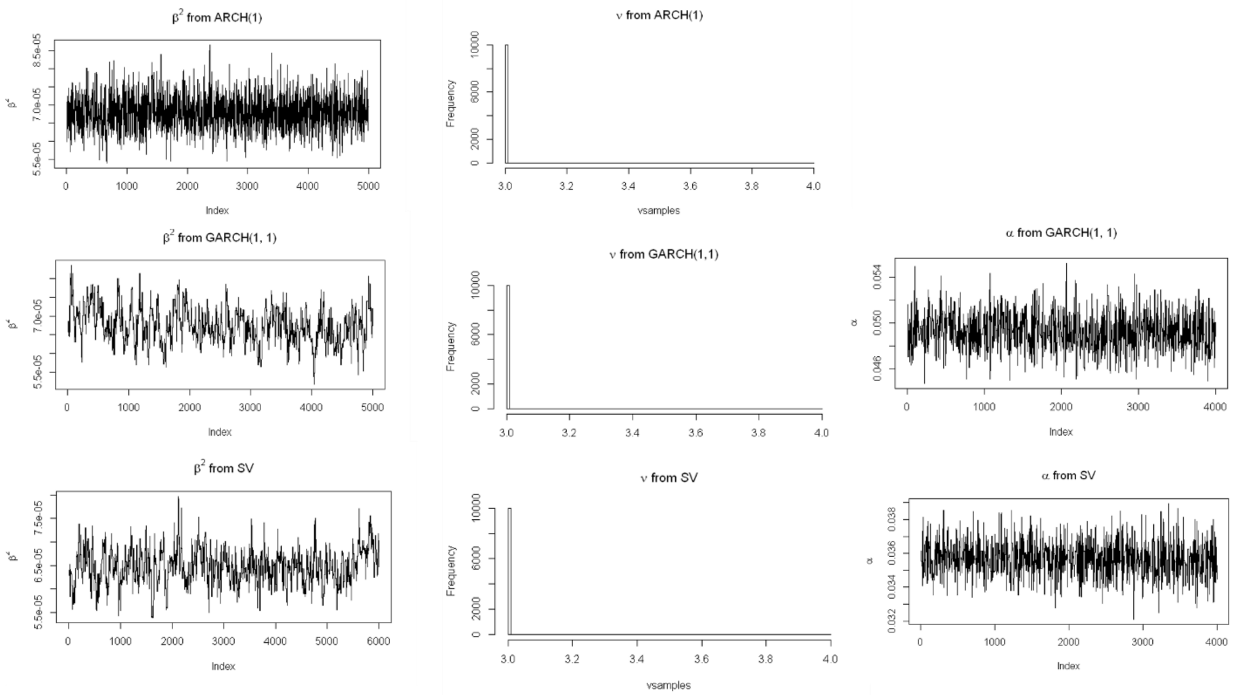

Figure 5: MCMC Sampled parameters after convergence. Plots on the left column are the sampled β^2 from its posterior in ARCH(1), GARCH(1,1) and SV model respectively after 5,000 or 6,000 iterations. Histograms in the middle are all samples of ν from its posterior in ARCH(1), GARCH(1,1) and SV model respectively. Plots on the right are the sampled α from its posterior in GARCH(1,1) and SV models respectively after 6,000 iterations.

MODEL CHECKING

We use a simple and intuitive way to check whether our models fit the data very well without doing any formal tests. The idea is that if our model fully explains the behaviors of the stock log return, the future log returns predicted by our models will behave similarly as the original returns from our training set, that is, have the same marginal distribution. To eliminate the autocorrelation, we looked at the log returns predicted 20 steps ahead for each of our model. Figure 6 shows the predicted log returns 20 steps ahead from our three models vs the log returns from the training data. The distributions of predicted values looked very similar to the log returns from our training set, except that each of our models produced longer tails than the original log returns, but overall, our models fit the data very well.

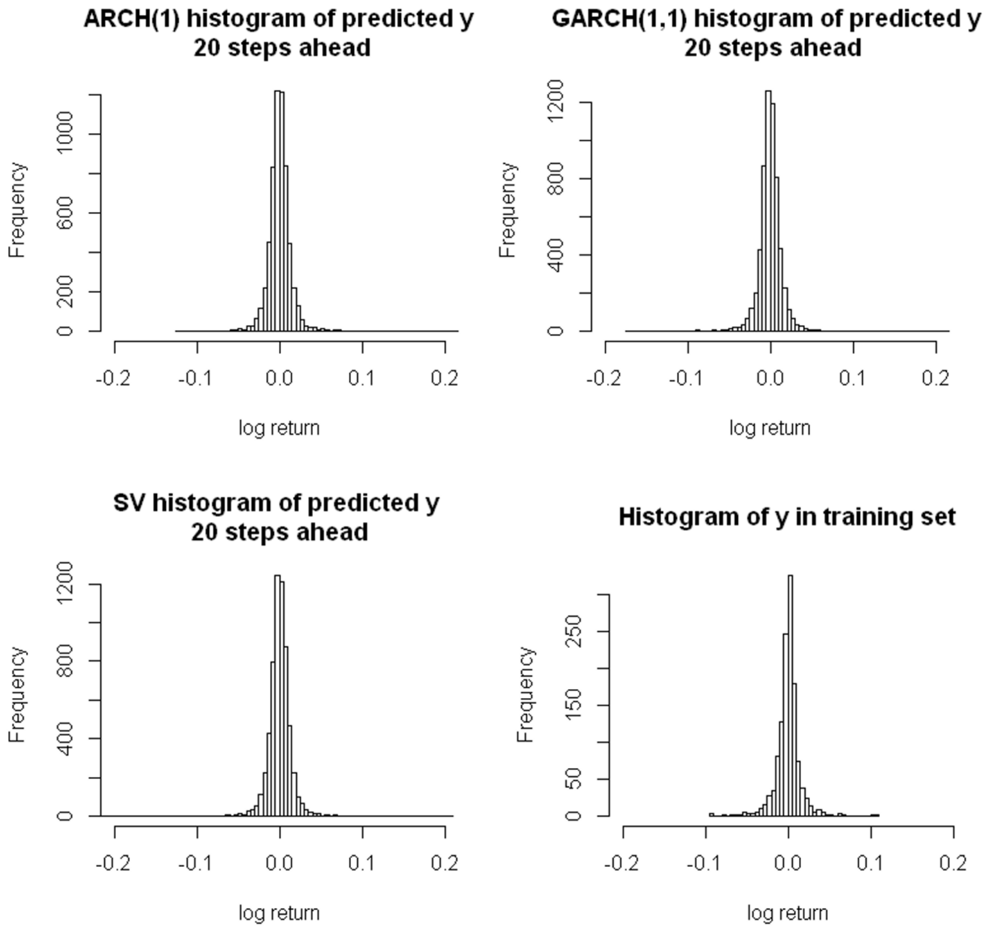

Figure 6: the predicted log returns 20 steps ahead from ARCH(1), GARCH(1,1) and SV model vs the log returns from the training data. The histograms from three models look extremely similar to the original training data, except that they have a much longer tails.

We also examined the volatility modeled in the training set by plotting the confidence intervals of the predicted log returns. As can be seen from Figure 7, both ARCH(1) and GARCH(1,1) gave large 95% credible intervals for data points that had high volatility, which almost fully covered those log returns while SV model only captured part of them and did not trace the data as well as the other two. From this sense, the ARCH(1) and GARCH(1,1) fit the data better than the SV model.

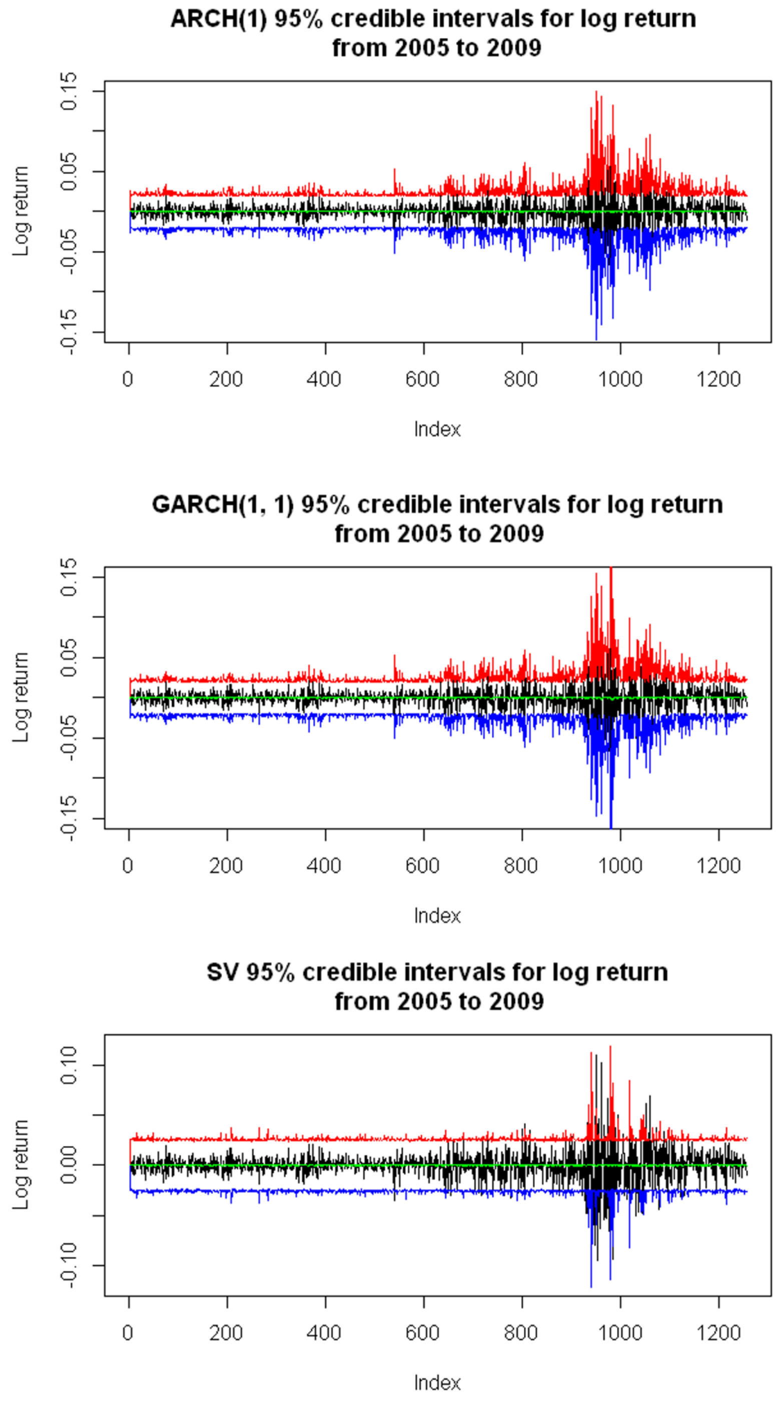

Figure 7: 95% credible intervals of the log returns from ARCH(1), GARCH(1,1) and SV model. The red line represents the values at 97.5% quantile of the predictions. The blue line represents the 2.5% quantile of the predictions. As can be seen, most of the true log returns are within the credible intervals generated by ARCH(1) and GARCH(1,1). The SV model credible interval does not cover the true data as well as the other two models.

STOCK PRICE PREDICTION

We only predicted one step ahead for each data point in the test dataset given the parameters estimates from the training set. Theoretically, the parameters should be re-estimated after we see each of the test log returns, which requires to rerun the 10,000 iterations for each test data point. To save time and computation cost, we fixed the parameter estimates obtained from the training set. The prediction procedure for ARCH(1) was as follows: given y_t, we sampled 1,000 predicted values from $p(y_{t+1}|y_t)$. We then took the average of the exponential values of those 1,000 predictions as the estimation of the expected price ratio $E(\frac{P_{t+1}}{P_t})$. Then the expected price p_{t+1} could be given by (3). After predicting y_{t+1}, we used the true y_{t+1} to predict y_{t+2} in the same fashion. The prediction procedures for GARCH(1,1) and SV models were similar to ARCH(1) except that the predictive density is not explicitly available, so we had to use the Markov chains shown in Figure (2) and (3) to update w and z at the same time, in order to derive y_{t+1} and the expected price ratio.

Figure 8 shows the predictions from three models vs the original test S&P 500 index. The green line represents the expected prices predicted from our models. The blue and red lines are 95% credible intervals for the predicted prices. The ARCH(1) and GARCH(1,1) predictions had nearly no difference while SV model predictions had larger credible intervals, but overall, all of our three models predicted the stock prices fairly well. Notice that for each model, the credible intervals at the tails of the time series were wider than that at the beginning, which indicated the necessity of re-estimating the parameters. We recorded the mean squared error (MSE) obtained from each model and compared them with that from the standard GARCH(1, 1) model which also used one-step-ahead predictions with fixed parameters from the training set. The result is shown in Table 1. It is obvious that our GARCH(1,1) had the lowest MSE, indicating the most accurate prediction. All of our three models outperformed the standard GARCH(1,1) with the evidence of smaller MSEs. One may argue that such difference was not substantial. However, considering the S&P 500 index is a measure of average performance of the stock market, we could expect larger differences in many individual stocks, especially those with higher volatility.

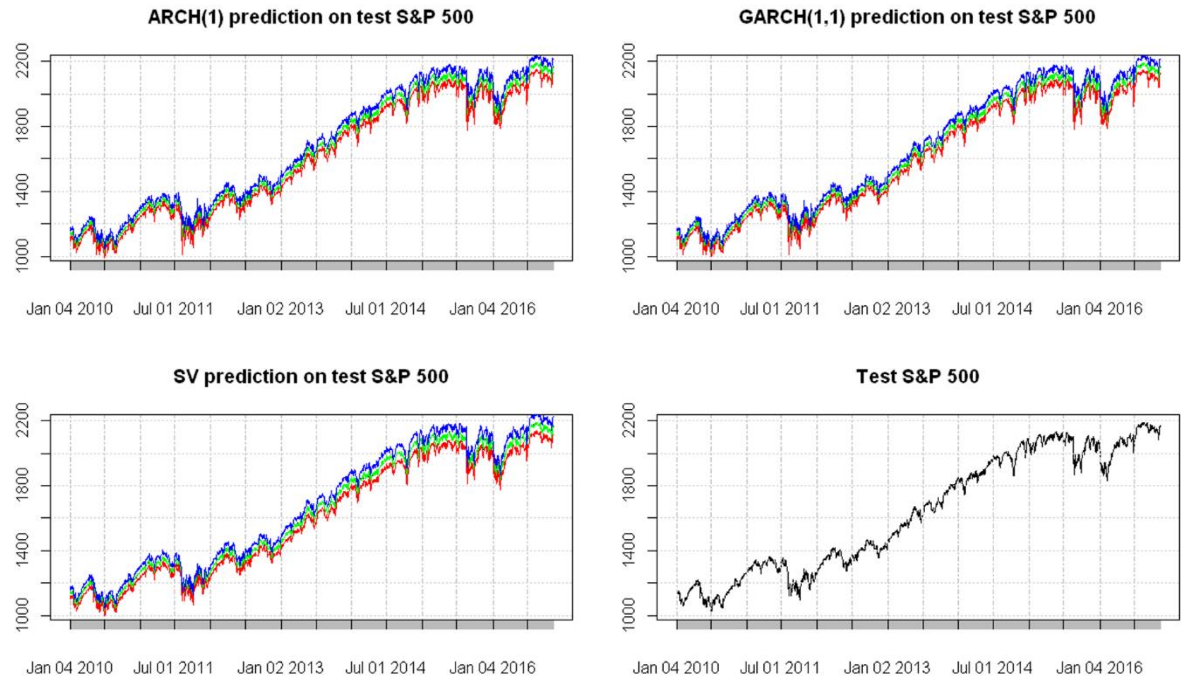

Figure 8: Predictions of test S&P 500 prices from ARCH(1), GARCH(1,1) and SV model vs the original test data. The green line is the expected stock price. The blue line represents the prediction at 97.5% quantile while the red line represents the prediction at 2.5% quantile. The ARCH(1) and GARCH(1,1) have very similar predictions with smaller credible intervals than that from the SV model. All three models show a wider credible interval at the tail.

Models	ARCH(1)	GARCH(1,1)	SV	Standard GARCH(1,1)
MSE	0.423	0.407	0.576	0.779

Table 1: Mean squared error obtained from our ARCH(1), GARCH(1,1), SV model and the standard GARCH(1,1). As is shown, all of our three models produced smaller MSE than that from the standard GARCH(1,1).

Discussion

In this report, we have shown that the methods proposed by Pitt and Walker (2005) could be used to model the stock volatility and the predictions from such methods are more accurate than that from the standard GARCH model. There are primarily two reasons for the model to be successful. First, a fixed marginal Student-t distribution for the log return seems to be a useful assumption which can be easily observed from the histograms, and proves to be a correct choice to account for the high volatility in the stock market. In fact, many financial time series behave similar as the log return when we take their difference at the log scale, in which case the marginal Student-t distribution assumption is also applicable. Figure 9 shows the differences of log foreign exchange rate between US dollars and Japan Yen, and the differences of log 10-year treasury constant maturity rate. All of the series have high peaks and heavy tails. Second, the purpose of introducing auxiliary variables w (in ARCH) and z as latent variables, is used not only to construct the marginal density of the log return and allow for longer-range dependence, but also to remain the structure of ARCH or GARCH. In our ARCH(1), different from the standard ARCH model of Engle (1982) given by (7) though, the structure is still an ARCH type as can be seen from (12), a restricted version of the heavy-tailed model by Bollerslev (1987) (Pitt and Walker, 2005). Our GARCH(1,1), if we take the expectation of w_t, will have the same structure of the standard GARCH given by (16):

$$\begin{aligned} E(w_t|y_{t-1}, w_{t-1}) &= E[E(w_t|y_{t-1}, z_{t-1})] \\ &= \frac{2}{\nu + 2\alpha - 1}\left[\frac{y_{t-1}^2 + \nu\beta^2}{2} + E(z_{t-1}|w_{t-1})\right] \\ &= \frac{y_{t-1}^2 + \nu\beta^2 + 2\alpha w_{t-1}}{\nu + 2\alpha - 1} \end{aligned} \tag{28}$$

If correspond to (16), we have $a = \frac{\nu\beta^2}{\nu+2\alpha-1}, b = \frac{1}{\nu+2\alpha-1}$ and $c = \frac{2\alpha}{\nu+2\alpha-1}$, a perfect standard GARCH structure. That's probably the reason why we need the latent variable z and prefer a Gamma distribution for it.

An interesting finding that Pitt and Walker (2005) does not cover is that their ARCH(1) has almost equally good performance as GARCH(1,1) but with substantially lower computation cost compared with GARCH(1,1) and SV model. The posterior mean for GARCH(1,1) and SV model often takes long time to converge and improvement on it primary relies on manually adjusting the proposal variance in Metropolis-Hasting algorithms for β^2, α and z. To the contrary, the sampling structure of ARCH(1,1) is quite simple and the prediction for y_{t+1} is only based on the y_t and parameters, since both the predictive density and the likelihood function does not involve the latent variable w. Therefore, for fast and decent prediction, ARCH(1) is highly preferred.

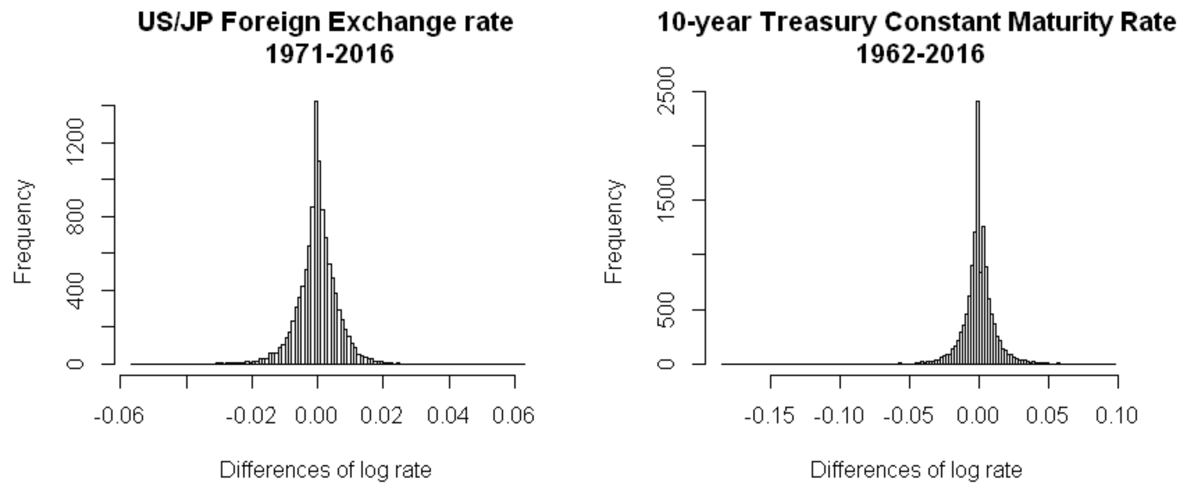

Figure 9: Histogram on the left: differences of log foreign exchange rate between US dollars and Japan Yen from 1971-2016. Histogram on the right: differences of log 10-year treasury constant maturity rate from 1962-2016. Both series resemble the Student-t distribution.

Appendix I: Posterior Derivation

1 ARCH(1)

To the contrary of traditional ARCH(1) model, Pitt and Walker (2005) uses the latent variable w_t such that the stock log return y_t comes from the marginal distribution: scaled t distribution.
The joint distribution of y and w:

$$\begin{aligned} f_{Y,W}(y,w) &= N(0,w) * Ig(\frac{\nu}{2}, \frac{\nu\beta^2}{2}) \\ &= \frac{1}{\sqrt{2\pi w}} \exp(-\frac{1}{2w}y^2)(\frac{\nu\beta^2}{2})^{\frac{\nu}{2}} \frac{1}{\Gamma(\frac{nu}{2})} w^{\frac{\nu}{2}-1} \exp(-\frac{\nu\beta^2}{2w}) \\ &= \frac{1}{\sqrt{2\pi}}(\frac{\nu\beta^2}{2})^{\frac{\nu}{2}} \frac{1}{\Gamma(\frac{\nu}{2})} w^{\frac{\nu+1}{2}+1} \exp(\frac{1}{2}(y^2+\nu\beta^2)w^{-1}) \end{aligned} \tag{1}$$

From (1), we have the marginal distribution of y as:

$$\begin{aligned} f_Y(y) &= \int f_{Y|W}(y|w) * f_W(w)dw \\ &= \frac{1}{\sqrt{2\pi}}(\frac{\nu\beta^2}{2})^{\frac{\nu}{2}} \frac{1}{\Gamma(\frac{\nu}{2})} \int w^{-(\frac{\nu}{2}+\frac{\nu}{2}+1)} \exp(-(\frac{1}{2}y^2+\frac{1}{2}\nu\beta^2))w^{-1}dw \\ &= \frac{1}{\sqrt{(2\pi)}} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} (\frac{y^2}{\nu\beta^2}+1)^{-\frac{\nu}{2}} (\frac{1}{2})^{-1/2} (y^2+\nu\beta^2)^{-\frac{1}{2}} \\ &= \frac{1}{\sqrt{(\pi)}} \frac{1}{\sqrt{\nu\beta^2}} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} [1+\frac{y^2}{\nu\beta^2}]^{-\frac{\nu+1}{2}} \end{aligned} \tag{2}$$

which is a scaled Studen-t distribution $t_\nu(0,\beta^2)$.
From (1), we have the conditional distribution $f_{W|Y}(w|y)$ as:

$$\begin{aligned} f_{W|Y}(w|y) &\propto f_{Y|W}(y|w) f_W(w) \\ &\propto \frac{1}{\sqrt{2\pi}}(\frac{\nu\beta^2}{2})^{\frac{\nu}{2}} \frac{1}{\Gamma(\frac{\nu}{2})} w^{\frac{\nu+1}{2}+1} \exp(\frac{1}{2}(y^2+\nu\beta^2)w^{-1}) \\ &\propto w^{-\frac{1}{2}} \exp(-\frac{1}{2w}y^2) w^{-\frac{\nu}{2}-1} \exp(-\frac{\nu\beta^2}{2w}) \\ &\propto w^{-\frac{\nu+1}{2}-1} \exp(-(\frac{1}{2}y^2+\frac{\nu\beta^2}{2})w^{-1}) \end{aligned} \tag{3}$$

which is an inverse-gamma distribution $Ig(\frac{\nu+1}{2}, \frac{1}{2}y^2 + \frac{\nu\beta^2}{2})$.
Given (3), the conditional distribution of $p(y_t|y_{t-1})$ is:

$$
\begin{aligned}
p(y_t|y_{t-1}) &= \int f_{Y|W}(y_t|w_t) f_{W|Y}(w_t|y_{t-1}) dw_t \\
&= \int w^{-\frac{\nu+1}{2}-1} \exp(-(\frac{1}{2}y_t^2 + \frac{1}{2}y_{t-1}^2 + \frac{\nu\beta^2}{2}))dw \\
&= (\frac{1}{2})^{\frac{\nu+1}{2}} (y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+1}{2}} \frac{1}{\Gamma(\frac{\nu+1}{2})} \Gamma(\frac{\nu+2}{2})(\frac{1}{2})^{-\frac{\nu+2}{2}} (y_t^2 + y_{t-1}^2 + \nu\beta^2)^{-\frac{\nu+2}{2}} \\
&= \sqrt{2}(y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+1}{2}} (y_t^2 + y_{t-1}^2 + \nu\beta^2)^{-\frac{\nu+2}{2}} \frac{1}{\Gamma(\frac{\nu+1}{2})} * \Gamma(\frac{\nu+2}{2}) \\
&\propto \frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{(\nu+1)}{2}}) * \frac{(y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+1}{2}}}{(y_t^2 + y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+2}{2}}} \\
&\propto \frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{(\nu+1)}{2}}) * (y_{t-1}^2 + \nu\beta^2)^{-\frac{1}{2}} * \frac{(y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+2}{2}}}{(y_t^2 + y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+2}{2}}} \\
&\propto \frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{(\nu+1)}{2}}) * (y_{t-1}^2 + \nu\beta^2)^{-\frac{1}{2}} * (\frac{1}{1 + \frac{y_t^2}{y_{t-1}^2 + \nu\beta^2}})^{\frac{-(v+2)}{2}} \qquad (4)
\end{aligned}
$$

Set $\frac{S^2}{\nu+1} = \frac{y_t^2}{y_{t-1}^2+\nu\beta^2}$ in (4), so we have:

$$
\begin{aligned}
p(y_t|y_{t-1}) &\propto \frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{(\nu+1)}{2}}) (y_{t-1}^2 + \nu\beta^2)^{-\frac{1}{2}} (1 + \frac{S^2}{\nu+1})^{\frac{-(\nu+2)}{2}} S^2 \frac{y_{t-1}^2 + \nu\beta^2}{\nu+1} \\
&\propto \sqrt{\frac{y_{t-1}^2 + \nu\beta^2}{1+\nu}} [\frac{\Gamma(\frac{\nu+2}{2})}{\Gamma(\frac{(\nu+1)}{2}}) \frac{1}{\sqrt{v+1}} (1 + \frac{S^2}{\nu+1})^{\frac{-(\nu+2)}{2}}] \qquad (5)
\end{aligned}
$$

Therefore, $y_t = \sqrt{\frac{y_{t-1}^2+\nu\beta^2}{1+\nu}} S_{\nu+1}$ where $S_{\nu+1} \sim t_{\nu+1}$, a Student-t random variable with $\nu+1$ degree of freedom.
Given (5), the likelihood function can be derived directly from $p(y_t|y_{t-1})$. We leave out $p(y_1)$ since its contribution to the likelihood can be neglected. We denote the set of our parameters ν and β^2 as θ:

$$
\begin{aligned}
L_\theta(y_1, y_2, ..., y_t) &= p(y_t|y_{t-1})p(y_{t-1}|y_{t-2})...p(y_2|y_1)p(y_1) \\
&\propto [\prod_{t=2}^{n} (y_{t-1}^2 + \nu\beta^2)^{\frac{\nu+1}{2}} (y_t^2 + y_{t-1}^2 + \nu\beta^2)^{-\frac{\nu+2}{2}} \frac{1}{\Gamma(\frac{\nu+1}{2})} \Gamma(\frac{\nu+2}{2})] \frac{1}{\sqrt{\nu\beta^2}} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \qquad (6)
\end{aligned}
$$

To achieve computation efficiency, we take the log of our likelihood function of (6):

$$
\begin{aligned}
\ln L_\theta(y_1, y_2, ..., y_t) &\propto \sum_{t=2}^{n} \frac{\nu+1}{2} \ln(y_{t-1}^2 + \nu\beta^2) - \sum_{t=2}^{n} \ln(y_t^2 + y_{t-1}^2 + \nu\beta^2) + \sum_{t=2}^{n} \ln\Gamma(\frac{\nu+2}{2}) \\
&- \sum_{t=2}^{n} \ln(\frac{\nu+1}{2}) \qquad (7)
\end{aligned}
$$

Given (7), we are able to derive the posterior of ν and β^2 in log scale. We give a discrete unifrom prior for ν, that is, ν has equal probability for each integer value from 3 to 20. Thus the prior of ν is a constant and can be neglected from its posterior:

$$\ln \Pi_n(\nu|...) \propto \sum_{t=2}^{n} \frac{\nu+1}{2} \ln(y_{t-1}^2 + \nu\beta^2) - \sum_{t=2}^{n} \frac{\nu+2}{2} \ln(y_t^2 + y_{t-1}^2 + \nu\beta^2) + (n-1)[\ln\Gamma(\frac{\nu+2}{2}) - \ln\Gamma(\frac{\nu+1}{2})] \tag{8}$$

We assign a Gamma(1,1) prior for β^2. The posterior of β^2 has the form:

$$\ln \Pi_n(\beta^2|...) \propto \sum_{t=2}^{n} \frac{\nu+1}{2} \ln(y_{t-1}^2 + \nu\beta^2) - \sum_{t=2}^{n} \frac{\nu+2}{2} \ln(y_t^2 + y_{t-1}^2 + \nu\beta^2) - \beta^2 \tag{9}$$

Since it is hard to sample from the posteriors of ν and β^2 directly, we use Metropolis-Hasting algorithm to facilitate the sampling.

2 GARCH(1,1)

Compared with our ARCH(1) model, the GARCH(1,1) introduces an auxiliary variable z and its predictive density is not explicitly available by integrating out z. Therefore, we not only need to update the parameters ν, β^2 and α in MCMC, but need to update w and z at the same time.

The joint density is:

$$f_{Y,W,Z}(y,w,z) = \frac{1}{sqrt(2\pi)\sqrt{w}} e^{-\frac{1}{2w}y^2} (\frac{\nu\beta^2}{2})^{\frac{\nu}{2}} \frac{1}{\Gamma(\frac{\nu}{2})} w^{-\frac{\nu}{2}-1} e^{-\frac{\nu\beta^2}{2w}} \frac{w^{-\alpha}}{\Gamma(\alpha)} z^{\alpha-1} e^{-w^{-1}z} \tag{10}$$

From (10), we are able to derive the conditional density $f_{W|Z,Y}(w|z,y)$ as:

$$\begin{aligned} f_{W|Z,Y}(w|z,y) &\propto w^{-\frac{1}{2}} e^{-\frac{1}{2w}y^2} w^{-\frac{1}{2w}-1} e^{-\frac{1}{2w}\nu\beta^2} w^{-\alpha} e^{-w^{-1}z} \\ &\propto w^{-\frac{1}{2}-\frac{\nu}{2}-\alpha-1} e^{-\frac{y^2}{2}w^{-1} - \frac{1}{2}\nu\beta^2 w^{-1} - zw^{-1}} \\ &\propto w^{-(\frac{1+v}{2}+\alpha)-1} e^{-(\frac{y^2+\nu\beta^2}{2}+z)w^{-1}} \end{aligned} \tag{11}$$

which is an inverse-gamma distribution $Ig(\frac{1+\nu}{2} + \alpha, \frac{y_{t-1}^2+\nu\beta^2}{2} + z_{t-1})$.

Denote the set of our parameters ν, β^2 and α as θ. Based on (11), the likelihood can be written as:

$$\begin{aligned} L_\theta(Y,W,Z) &= \prod_{t=2}^{n} f(y_t|w_t) f(z_t|w_t) f(w_t|y_{t-1}, z_{t-1}) \\ &= \prod_{t=2}^{n} N(0,w_t) Ga(\alpha, w^{-1}) Ig(\frac{1+\nu}{2} + \alpha, \frac{y_{t-1}^2 + \nu\beta^2}{2} + z_{t-1}) \\ &\propto [\frac{1}{\Gamma(\alpha)} \frac{1}{\Gamma(\frac{1+\nu}{2})}]^{n-1} \prod_{t=2}^{n} w_t^{-(\frac{2+\nu}{2}+2\alpha)-1} e^{-(\frac{y_t^2+y_{t-1}^2+\nu\beta^2}{2}+z_t+z_{t-1})w_t^{-1}} z_t^{\alpha-1} (\frac{y_{t-1}^2 + \nu\beta^2}{2} + z_{t-1})^{\frac{1+\nu}{2}+\alpha} \end{aligned} \tag{12}$$

We assign a Gamma(1,1) prior to α. The priors of β^2 and ν are the same with those in ARCH(1). Given (12), the posterior of α on the log scale can be derived as:

$$\ln \Pi_n(\alpha|...) \propto -(n-1)[\ln\Gamma(\alpha) + \ln\Gamma(\frac{1+\nu}{2})] - 2\alpha\sum_{t=2}^{n}\ln w_t + (\alpha-1)\sum_{t=2}^{n}\ln z_t + \alpha\sum_{t=2}^{n}\ln(\frac{y_{t-1}^2+\nu\beta^2}{2} + z_{t-1}) - \alpha \tag{13}$$

The posterior of β^2 on the log scale is:

$$\ln \Pi_n(\beta^2|...) \propto -\frac{\nu\beta^2}{2}(\sum_{t=2}^{n} w_t^{-1}) + (\frac{1+\nu}{2}+\alpha)\sum_{t=2}^{n}\ln(\frac{y_{t-1}^2+\nu\beta^2}{2}+z_{t-1}) - \beta^2 \tag{14}$$

The posterior of ν is:

$$\ln \Pi_n(\nu|...) \propto -(n-1)\ln\Gamma(\frac{1+\nu}{2}+\alpha) - \frac{2+\nu}{2}\sum_{t=2}^{n}\ln w_t - \frac{\nu\beta^2}{2}\sum_{t=2}^{n} w_t^{-1} + (\frac{1+\nu}{2}+\alpha)\sum_{t=2}^{n}\ln(\frac{y_{t-1}^2+\nu\beta^2}{2}+z_{t-1}) \tag{15}$$

Given (12), we are also able to derive the posterior of w_t:

$$\Pi_n(w_t|\theta, y_t, y_{t-1}, z_t, z_{t-1}) \propto w_t^{-(\frac{2+\nu}{2}+2\alpha)-1} e^{-(\frac{y_t^2+y_{t-1}^2+\nu\beta^2}{2}+z_t+z_{t-1})w_t^{-1}} \tag{16}$$

which is a inverse-gamma distribution $Ig(\frac{2+\nu}{2}+2\alpha, \frac{y_t^2+y_{t-1}^2+\nu\beta^2}{2}+z_t+z_{t-1})$.
The posterior of z_t has the form:

$$\begin{aligned}\Pi_n(z_t|\theta, w_t, w_{t+1}) &\propto f_{W|Z}(w_{t+1}|z_t) f_{Z|W}(z_t|w_t)\\ &\propto e^{-(w_{t+1}^{-1}+w_t^{-1})z_t} z_t^{\alpha-1}(\frac{y_t^2+\nu\beta^2}{2}+z_t)^{\frac{1+\nu}{2}+\alpha}\end{aligned} \tag{17}$$

We use Metropolis-Hasting algorithm to sample ν, β^2, α and z from their posteriors.

3 Stochastic Volatility model

The SV model shares a lot of similarities with GARCH(1,1) in the MCMC sampling scheme. The only difference is that the updates of w_t only rely on z_t, as is shown in Figure (3). The joint density is the same with that in GARCH(1,1). So from (10), we can obtain the conditional density $f_{W|Z}(w|z)$:

$$\begin{aligned}f_{W|Z}(w|z) &\propto z^{\alpha-1} e^{-w_t^{-1}z} w^{-\frac{\nu}{2}-1} e^{-\frac{\nu\beta^2}{2}w^{-1}}\\ &\propto w^{-(\frac{\nu}{2}+\alpha)-1} e^{-(z+\frac{\nu\beta^2}{2})w^{-1}}\end{aligned} \tag{18}$$

which is a inverse-gamma distribution $Ig(\frac{\nu}{2}+\alpha, z+\frac{\nu\beta^2}{2})$.
Given (18), the likelihood function can be obtained as:

$$
\begin{aligned}
L_\theta(Y,W,Z) &= \prod_{t=2}^{n} f(y_t|w_t)f(z_t|w_t)f(w_t|z_{t-1}) \\
&= \prod_{t=2}^{n} N(0,w_t)Ga(\alpha,w^{-1})Ig(\frac{\nu}{2}+\alpha,\frac{\nu\beta^2}{2}+z_{t-1}) \\
&\propto [\frac{1}{\Gamma(\alpha)}\frac{1}{\Gamma(\frac{\nu}{2})}]^{n-1}\prod_{t=2}^{n} w_t^{-(\frac{1+\nu}{2}+2\alpha)-1} e^{-(\frac{y_t^2+\nu\beta^2}{2}+z_t+z_{t-1})w_t^{-1}} z_t^{\alpha-1}(\frac{\nu\beta^2}{2}+z_{t-1})^{\frac{\nu}{2}+\alpha}
\end{aligned}
\tag{19}
$$

From (19), we have our log posterior of α as:

$$
\begin{aligned}
\ln\Pi_n(\alpha|...) \propto& -(n-1)[\ln\Gamma(\alpha)+\ln\Gamma(\frac{\nu}{2})]-2\alpha\sum_{t=2}^{n}\ln w_t+(\alpha-1)\sum_{t=2}^{n}\ln z_t+ \\
&\alpha\sum_{t=2}^{n}\ln(\frac{\nu\beta^2}{2}+z_{t-1})-\alpha
\end{aligned}
\tag{20}
$$

our log posterior of β^2 as:

$$
\ln\Pi_n(\beta^2|...) \propto -\frac{\nu\beta^2}{2}(\sum_{t=2}^{n} w_t^{-1})+(\frac{\nu}{2}+\alpha)\sum_{t=2}^{n}\ln(\frac{\nu\beta^2}{2}+z_{t-1})-\beta^2 \tag{21}
$$

and our log posterior of ν as:

$$
\begin{aligned}
\ln\Pi_n(\nu|...) \propto& -(n-1)\ln\Gamma(\frac{\nu}{2}+\alpha)-\frac{1+\nu}{2}\sum_{t=2}^{n}\ln w_t-\frac{\nu\beta^2}{2}\sum_{t=2}^{n} w_t^{-1}+ \\
&(\frac{\nu}{2}+\alpha)\sum_{t=2}^{n}\ln(\frac{\nu\beta^2}{2}+z_{t-1})
\end{aligned}
\tag{22}
$$

Compared with the posterior of w_t in GARCH(1,1), the posterior of w_t in SV model does not contain y_{t-1} :

$$
\Pi_n(w_t|\theta,y_t,z_t,z_{t-1}) \propto w_t^{-(\frac{1+\nu}{2}+2\alpha)-1} e^{-(\frac{y_t^2+\nu\beta^2}{2}+z_t+z_{t-1})w_t^{-1}} \tag{23}
$$

which is still a inverse-gamma distribution $Ig(\frac{1+\nu}{2}+2\alpha, \frac{y_t^2+\nu\beta^2}{2}+z_t+z_{t-1})$. The posterior of z_t in SV model is the same with that in GARCH(1,1), so we leave out its equation here.

Appendix II: R codes

1 Data

```
# See documentation for TTR package
library(TTR)
require(xts)

# Load training S&P 500 returns from Yahoo
Sys.setenv(tz = "UTC")
sp500 = getYahooData('^GSPC', start = 20050101, end = 20091231, freq = 'daily')
logp = log(sp500[,4])
y = diff(logp)
y = as.matrix(data.frame(y[-1], row.names = c()))
orig.y = y
n.y = length(y)

# test dataset
test.sp500 = getYahooData('^GSPC', start = 20100101, end = 20161111, freq = 'daily')
logptest = log(test.sp500[,4])
test.y = diff(logptest)
test.y = as.matrix(data.frame(test.y[-1], row.names = c()))
n.y2 = length(test.y)

# full dataset
full_y = c(y, test.y)

# rearrange plots
par(mfrow = c(1,2))
hist(y, breaks = 50, main = 'Log return of S&P 500 \nfrom 2005 to 2009')
hist(test.y, breaks = 35, main = 'Log return of S&P 500 \nfrom 2010 to 2016')
```

2 ARCH(1)

```
# functions
# posterior of v, degree of freedom
post.v = function(yvec, sqbeta, v_vec){
```

```
num = c()
n = length(yvec)
for (v in v_vec){
ix = which(v_vec == v)

a = sum((v+1)/2 * log(yvec[-n]^2 + v*sqbeta))
b = sum((v+2)/2 * log(yvec[2:n]^2 + yvec[1:n-1]^2 + v*sqbeta))
c = (n-1)*(log(gamma((v+2)/2)) - log(gamma((v+1)/2)))
loglikelihood = a-b+c

if (v == 3){
alpha = -loglikelihood
 }
num = c(num, exp(alpha + loglikelihood))
}

denominator = sum(num)
print(num)
probs = num/denominator
new_v = sample(v_vec, 1, prob = probs)

return(new_v)
}

# posterior of beta^2
post.sqbeta = function(yvec, sqbeta, v){
n = length(yvec)
a = sum((v+1)/2 * log(yvec[-n]^2 + v*sqbeta))
b = sum((v+2)/2 * log(yvec[2:n]^2 + yvec[1:n-1]^2 + v*sqbeta))
c = - sqbeta
return(a-b+c)
}

# Metropolis-Hasting for beta^2
MH.sqbeta = function(sqbeta, post.sqbeta, v, yvec){
sqbeta2 = rlnorm(1, meanlog = log(sqbeta), sdlog = 0.1)
ratio = post.sqbeta(yvec, sqbeta2, v)+log(dlnorm(sqbeta, log(sqbeta2), 0.1)) -
        (post.sqbeta(yvec, sqbeta, v) + log(dlnorm(sqbeta2, log(sqbeta), 0.1)))
ratio = exp(ratio)
u = runif(1, 0, 1)
if (u < min(1, ratio)){
sqbeta = sqbeta2
 }
return(sqbeta)
}
```

```
# predictive density
pred.y = function(yvec, sqbeta, v, step){
n = length(yvec)
lasty = yvec[n]
for (k in 1:step){
s = rt(1, v+1)
newy = sqrt((lasty^2 + v*sqbeta)/(1+v))*s
lasty = newy
}
return(newy)
}


#---------------------------------------------------------------
# initial values
# prior for v: descrete values, 3:20
v_vec = c(3:20)
init.sqbeta = 0.001
iter = 10000

# begin sampling
vsamples = rep(0, iter)
sqbetasamples = rep(0, iter)
predict_y = c()
predict_ysq = c()
sqbeta = init.sqbeta

# predictive distribution for each y
ypred.samples = matrix(0, nrow = iter, ncol = n.y)

for (i in 1:iter){
vsamples[i] = post.v(y, sqbeta, v_vec)
print(i)
sqbetasamples[i] = MH.sqbeta(sqbeta, post.sqbeta, vsamples[i], y)
sqbeta = sqbetasamples[i]

# model checking
for (j in 2:n.y){
   ypred.samples[i, j] = pred.y(y[1:(j-1)], sqbeta, vsamples[i], step = 1)
 }

if (i > iter*0.4){
 # predict new y, 20 steps ahead to eliminate dependence
 # check if the future value can mimic the past
```

```
   ynew = pred.y(y, sqbeta, vsamples[i], step = 20)
   predict_y = c(predict_y, ynew)
 }
}

# Posterior of beta^2
plot(sqbetasamples[5000:iter], type = 'l', ylab = bquote(beta^2),
main = expression(paste(beta^2, ' from ARCH(1)')))
paste('The posterior mean of beta^2 is', mean(sqbetasamples[5000:iter]))

# Posterior of degree of freedom: v
plot(vsamples[2000:iter], type='l')
hist(vsamples, breaks = 100, main = expression(paste(nu, ' from ARCH(1,1)')))

# Model checking: predictive distribution of y vs original
par(mfrow = c(1,2))
hist(predict_y, breaks = 100, xlab = 'log return', xlim = c(-0.2, 0.2),
main = 'ARCH(1) histogram of predicted y\n 20 steps ahead')
hist(orig.y, breaks = 50, xlab = 'log return',  xlim = c(-0.2, 0.2),
main = 'Histogram of y in training set')


# 95% credible intervals for log return prediction
mean_pred = apply(ypred.samples[2000:iter,], 2, mean)
bounds = apply(ypred.samples, 2, function(z) quantile(z, c(0.025, 0.975)))
plot(orig.y, type = 'l', main = 'ARCH(1) 95% credible intervals for log return
     \nfrom 2005 to 2009', ylab = 'Log return', ylim = c(-0.15, 0.15))
lines(bounds[1,], col = 'blue')
lines(bounds[2,], col = 'red')
lines(mean_pred, col = 'green')


# Stock price prediction on test data
meansqbeta = mean(sqbetasamples[7000:iter])
# the majority of v
meanv = 3
test.ypred = rep(0, n.y2)
test.bounds = matrix(0, nrow = 2, ncol = n.y2)

for (k in 1:n.y2){
print(k)
testsamples = rep(0, 1000)
testsamples = sapply(testsamples, function(z) pred.y(full_y[1:(n.y+k-1)],
                     meansqbeta, meanv, step = 1))
```

```
# expected value
test.ypred[k] = mean(exp(testsamples))

# 95% credible intervals
testbound = quantile(testsamples, c(0.025, 0.975))
test.bounds[1,k] = exp(testbound[1])
test.bounds[2,k] = exp(testbound[2])
}

test.p = test.sp500[-(n.y2+1),4]*test.ypred
test.plower = test.sp500[-(n.y2+1),4]*test.bounds[1,]
test.pupper = test.sp500[-(n.y2+1),4]*test.bounds[2,]

# generate prediction plots
plot(test.sp500[,4], type = 'l', main = 'ARCH(1) prediction on test S&P 500')
lines(test.p, cex = 0.3, col = 'green')
lines(test.plower, col = 'red')
lines(test.pupper, col = 'blue')

# MSE of ARCH(1) predictions
mse.arch = mean((test.sp500[,4]-test.p)^2)
```

3 GARCH(1,1)

```
# functions
# posterior of alpha
post.alpha = function(yvec, w, z, alpha, v, sqbeta){
# returnn: log of posterior of alpha
# prior for alpha: gamma(1, 1)
w = w[-1]
zt = z[-1]
n = length(yvec)
zt1 = z[-n]
yvec = yvec[-n]

a = -(n-1)*(log(gamma(alpha)) + log(gamma((1+v)/2 + alpha)))
b = -2*alpha*sum(log(w)) + (alpha - 1)*sum(log(zt))
c = alpha*sum(log((yvec^2 + v*sqbeta^2)/2 + zt1)) - alpha
return(a + b + c)
}

# Metropolis-Hasting for alpha
MH.alpha = function(yvec, w, z, alpha, v, sqbeta){
alpha2 = rlnorm(1, meanlog = log(alpha), 0.1)
ratio = post.alpha(yvec, w, z, alpha2, v, sqbeta) + log(dlnorm(alpha,
```

```
        log(alpha2), 0.1)) - (post.alpha(yvec, w, z, alpha, v, sqbeta) +
        log(dlnorm(alpha2, log(alpha), 0.1)))
ratio = exp(ratio)
u = runif(1, 0, 1)
if (u < min(1, ratio)){
alpha = alpha2
}
# print(alpha)
return(alpha)
}


# posterior of beta^2
post.sqbeta = function(yvec, w, z, alpha, v, sqbeta){
# returnn: log of posterior of sqbeta
# prior for sqbeta: gamma(1, 1)
w = w[-1]
n = length(yvec)
zt1 = z[-n]
yvec = yvec[-n]

a = -v*sqbeta/2 * sum(1/w)
b = ((1+v)/2 + alpha)*sum(log((yvec^2 + v*sqbeta)/2 + zt1)) - sqbeta
return(a + b)
}

# Metropolis-Hasting for beta^2
MH.sqbeta = function(yvec, w, z, alpha, v, sqbeta){
sqbeta2 = rlnorm(1, meanlog = log(sqbeta), 0.1)
ratio = post.sqbeta(yvec, w, z, alpha, v, sqbeta2) + log(dlnorm(sqbeta,
        log(sqbeta2), 0.1)) - (post.sqbeta(yvec, w, z, alpha, v, sqbeta) +
        log(dlnorm(sqbeta2, log(sqbeta), 0.1)))
ratio = exp(ratio)
u = runif(1, 0, 1)
if (u < min(1, ratio)){
sqbeta = sqbeta2
}
return(sqbeta)
}


# posterior of v
post.v = function(yvec, w, z, alpha, sqbeta, v_vec){
# returnn: log of posterior of sqbeta
# prior for sqbeta: gamma(1, 1)
```

```
w = w[-1]
n = length(yvec)
zt1 = z[-n]
yvec = yvec[-n]
num = c()

for (v in v_vec){
a = -(n-1)*log(gamma((1+v)/2 + alpha))
b = -v/2*sum(log(w)) - v*sqbeta/2*sum(1/w)
c = ((1+v)/2 + alpha)*sum(log((yvec^2 + v*sqbeta)/2 + zt1))
loglikelihood = a + b + c

if (v == 3){
benchmark = -loglikelihood
}

num = c(num, exp(benchmark + loglikelihood))
}
print(num)
denominator = sum(num)
# print(num)
probs = num/denominator
new_v = sample(v_vec, 1, prob = probs)
return(new_v)
}

# posterior of w
post.w = function(yvec, z, alpha, sqbeta, v){
# return: new w sampled from posterior of wt
n = length(yvec)
z_t = z[-1]
z_t1 = z[-n]
y_t = yvec[-1]
y_t1 = yvec[-n]

# when t > 1
a = (2+v)/2 + 2*alpha
b = (y_t^2 + y_t1^2 + v*sqbeta)/2 + z_t + z_t1
w = sapply(b, function(x) rinvgamma(1, a, x))

# when t = 1
w1 = rinvgamma(1, a, (yvec[1]^2 + v*sqbeta)/2 + z[1])
w = c(w1, w)

return(w)
```

```
}


# posterior of z
post.z = function(w, z, alpha, sqbeta, v){
# return: posterior of z
n = length(w)
w_t1 = w[-1]
w_t = w[-n]
z_t = z[-n]

# when t < n
a = exp(-(1/w_t1 + 1/w_t)*z_t)*z_t^(alpha - 1)
b = (v*sqbeta/2 + z_t)^(v/2 + alpha)
zn_1 = a*b

# when t = n
c = exp(-(2/w[n])*z[n])*z[n]^(alpha - 1)
d = (v*sqbeta/2 + z[n])^(v/2 + alpha)
zn = c*d

post_z = c(zn_1, zn)
return(post_z)
}

# Metropolis-Hasting for z
MH.z = function(w, z, alpha, sqbeta, v){
# return: new z sampled from the posterior
n = length(w)
z2 = sapply(z, function(x) rlnorm(1, meanlog = log(x), 0.1))
ratios = post.z(w, z2, alpha, sqbeta, v) * dlnorm(z, log(z2), 0.1)/
(post.z(w, z, alpha, sqbeta, v) * dlnorm(z2, log(z), 0.1))

#  ratios = exp(ratios)

u = runif(n, 0, 1)
for (i in 1:n){
if (u[i] < min(1, ratios[i])){
z[i] = z2[i]
}
}
return(z)
}

# predictive density
```

```
pred.y = function(yvec, zvec, alpha, sqbeta, v, step){
# return: predicted y
# yvec is different from above, y with user-defined length
n = length(yvec)
lasty = yvec[n]
lastz = zvec[n]
for (k in 1:step){
new_w = rinvgamma(1, (1+v)/2 + alpha, (lasty^2 + v*sqbeta)/2 + lastz)
newy = rnorm(1, 0, sqrt(new_w))
newz = rgamma(1, alpha, rate = 1/new_w)
lasty = newy
lastz = newz
}
structure(list(pred_y = newy, pred_z = newz))
}


#-------------------------
# initial values
# prior for v: 3:20
v_vec = c(3:20)
init.sqbeta = 0.0005
alpha = 0.05
iter = 10000
w = rinvgamma(n.y, 3/2, 3*init.sqbeta/2)
z = sapply(w, function(x) rgamma(1, alpha, 1/x))


# begin sampling
vsamples = rep(0, iter)
sqbetasamples = rep(0, iter)
alphasamples = rep(0, iter)
predict_y = c()
ypred.samples = matrix(0, nrow = iter, ncol = n.y)
sqbeta = init.sqbeta
last_w = c()
last_z = c()

for (i in 1:iter){
print(i)
# update parameters
vsamples[i] = post.v(y, w, z, alpha, sqbeta, v_vec)
sqbetasamples[i] = MH.sqbeta(y, w, z, alpha, vsamples[i], sqbeta)
alphasamples[i] = MH.alpha(y, w, z, alpha, vsamples[i], sqbetasamples[i])
```

```
# update z
z = MH.z(w, z, alphasamples[i], sqbetasamples[i], vsamples[i])
last_z = c(last_z, z[n.y])

# update w
w = post.w(y, z, alphasamples[i], sqbetasamples[i], vsamples[i])
last_w = c(last_w, w[n.y])

# model checking
  for (j in 2:n.y){
    ypred.samples[i, j] = pred.y(y[1:(j-1)], z[1:(j-1)], alphasamples[i],
                                 sqbetasamples[i], vsamples[i], step = 1)$pred_y
  }

 if (i > iter*0.4){
   # predict new y, 20 steps ahead to eliminate dependence
   # check if the future values can mimic the past
   ynew = pred.y(y, z, alphasamples[i], sqbetasamples[i], vsamples[i],
              step = 20)$pred_y
   predict_y = c(predict_y, ynew)
 }

# loop
sqbeta = sqbetasamples[i]
alpha = alphasamples[i]
}


# Posterior of beta^2
plot(sqbetasamples[6000:iter], type = 'l', ylab = bquote(beta^2),
main = expression(paste(beta^2, ' from GARCH(1, 1)')))
paste('The posterior mean of beta^2 is', mean(sqbetasamples[6000:iter]))

# Posterior of degree of freedom: v
hist(vsamples, main = 'v from GARCH(1,1)')

# Posterior of alpha
plot(alphasamples[6000:iter], type='l', ylab = bquote(alpha),
main = expression(paste(alpha, ' from GARCH(1, 1)')))
paste('The posterior mean of alpha is', mean(alphasamples[6000:iter]))

# Model checking: predictive distribution of y vs original
par(mfrow = c(1,2))
hist(predict_y, breaks = 100, xlab = 'log return', xlim = c(-0.2, 0.2),
main = 'GARCH(1,1) histogram of predicted y \n 20 steps ahead')
```

```
hist(orig.y, breaks = 50, xlab = 'log return', xlim = c(-0.2, 0.2),
main = 'Histogram of y in training set')

# 95% credible intervals for log return prediction
mean_pred = apply(ypred.samples[5000:iter,], 2, mean)
bounds = apply(ypred.samples, 2, function(z) quantile(z, c(0.025, 0.975)))
plot(orig.y, type = 'l',
main = 'GARCH(1, 1) 95% credible intervals for log return\n from 2005 to 2009',
ylab = 'Log return', ylim = c(-0.15, 0.15))
lines(bounds[1,], col = 'blue')
lines(bounds[2,], col = 'red')
lines(mean_pred, col = 'green')

# prediction on test data
meansqbeta = mean(sqbetasamples[5000:iter])
meanalpha = mean(alphasamples[6000:iter])
# the majority of v
meanv = 3
test.ypred = rep(0, n.y2)
trained_z = last_z[9001:iter]
test.bounds = matrix(0, nrow = 2, ncol = n.y2)

for (k in 1:n.y2){
print(k)
testsamples = rep(0, 1000)
for (t in 1:1000){
pred = pred.y(full_y[1:(n.y+k-1)], c(z[1:(n.y+k-2)], trained_z[t]),
              alpha = meanalpha, meansqbeta, meanv, step = 1)
testsamples[t] = pred$pred_y

if (t != iter){
trained_z[t+1] = pred$pred_z
 }
}

# 95% credible intervals
testbound = quantile(testsamples, c(0.025, 0.975))
test.bounds[1,k] = exp(testbound[1])
test.bounds[2,k] = exp(testbound[2])

# expected value
test.ypred[k] = mean(exp(testsamples))
}
test.p = test.sp500[-(n.y2+1),4]*test.ypred
test.plower = test.sp500[-(n.y2+1),4]*test.bounds[1,]
```

```
test.pupper = test.sp500[-(n.y2+1),4]*test.bounds[2,]

# generate plots of predictions
plot(test.sp500[,4], type = 'l', main = 'GARCH(1,1) prediction on test S&P 500')
lines(test.p, col = 'green')
lines(test.plower, col = 'red')
lines(test.pupper, col = 'blue')

# MSE of GARCH(1,1) predictions
mse.garch = mean((test.sp500[,4]-test.p)^2)
```

4 Stochastic Volatility model

```
# functions
# posterior of alpha
post.alpha = function(w, z, alpha, v, sqbeta){
# returnn: log of posterior of alpha
# prior for alpha: gamma(1, 1)
wt = w[-1]
zt = z[-1]
n = length(w)
zt1 = z[-n]

a = -(n-1)*(log(gamma(alpha)) + log(gamma(v/2 + alpha)))
b = -2*alpha*sum(log(wt)) + (alpha - 1)*sum(log(zt))
c = alpha*sum(log(v*sqbeta^2/2 + zt1)) - alpha
return(a + b + c)
}

# Metropolis-Hasting for alpha
MH.alpha = function(w, z, alpha, v, sqbeta){
alpha2 = rlnorm(1, meanlog = log(alpha), 0.1)
ratio = post.alpha(w, z, alpha2, v, sqbeta) + log(dlnorm(alpha, log(alpha2), 0.1)) -
(post.alpha(w, z, alpha, v, sqbeta) + log(dlnorm(alpha2, log(alpha), 0.1)))
ratio = exp(ratio)
u = runif(1, 0, 1)
if (u < min(1, ratio)){
alpha = alpha2
}
# print(alpha)
return(alpha)
}

# posterior of beta^2
post.sqbeta = function(w, z, alpha, v, sqbeta){
```

```
# returnn: log of posterior of sqbeta
# prior for sqbeta: gamma(1, 1)
wt = w[-1]
n = length(w)
zt1 = z[-n]

a = -v*sqbeta/2 * sum(1/wt)
b = (v/2 + alpha)*sum(log(v*sqbeta/2 + zt1)) - sqbeta
return(a + b)
}

# Metropolis-Hasting for beta^2
MH.sqbeta = function(w, z, alpha, v, sqbeta){
sqbeta2 = rlnorm(1, meanlog = log(sqbeta), 0.1)
ratio = post.sqbeta(w, z, alpha, v, sqbeta2) +
        log(dlnorm(sqbeta, log(sqbeta2), 0.1))-
        (post.sqbeta(w, z, alpha, v, sqbeta) +
        log(dlnorm(sqbeta2, log(sqbeta), 0.1)))
ratio = exp(ratio)
u = runif(1, 0, 1)
if (u < min(1, ratio)){
sqbeta = sqbeta2
}
return(sqbeta)
}

# posterior of v
post.v = function(w, z, alpha, sqbeta, v_vec){
# returnn: log of posterior of sqbeta
# prior for sqbeta: gamma(1, 1)
n = length(w)
wt = w[-1]
zt1 = z[-n]
num = c()

for (v in v_vec){
a = -(n-1)*log(gamma(v/2 + alpha))
b = -v/2*sum(log(wt)) - v*sqbeta/2*sum(1/wt)
c = (v/2 + alpha)*sum(log(v*sqbeta/2 + zt1))
loglikelihood = a + b + c

if (v == 3){
benchmark = -loglikelihood
}
```

```
num = c(num, exp(benchmark + loglikelihood))
}
print(num)
denominator = sum(num)
# print(num)
probs = num/denominator
new_v = sample(v_vec, 1, prob = probs)
return(new_v)
}

# posterior of w
post.w = function(yvec, z, alpha, sqbeta, v){
# return: new w sampled from posterior of wt
n = length(yvec)
z_t = z[-1]
z_t1 = z[-n]
y_t = yvec[-1]
y_t1 = yvec[-n]

# when t > 1
a = (1+v)/2 + 2*alpha
b = (y_t^2 + v*sqbeta)/2 + z_t + z_t1
w = sapply(b, function(x) rinvgamma(1, a, x))

# when t = 1
w1 = rinvgamma(1, a, (yvec[1]^2 + v*sqbeta)/2 + z[1])
w = c(w1, w)

return(w)
}


# posterior of z
post.z = function(w, z, alpha, sqbeta, v){
# return: posterior of z
n = length(w)
w_t1 = w[-1]
w_t = w[-n]
z_t = z[-n]

# when t < n
a = exp(-(1/w_t1 + 1/w_t)*z_t)*z_t^(alpha - 1)
b = (v*sqbeta/2 + z_t)^(v/2 + alpha)
zn_1 = a*b
```

```
# when t = n
c = exp(-(1/w[n])*z[n])*z[n]^(alpha - 1)
d = (v*sqbeta/2 + z[n])^(v/2 + alpha)
zn = c*d

post_z = c(zn_1, zn)
return(post_z)
}

# Metropolis-Hasting for z
MH.z = function(w, z, alpha, sqbeta, v){
# return: new z sampled from the posterior
n = length(w)
z2 = sapply(z, function(x) rlnorm(1, meanlog = log(x), 0.1))
ratios = post.z(w, z2, alpha, sqbeta, v) * dlnorm(z, log(z2), 0.1)/
(post.z(w, z, alpha, sqbeta, v) * dlnorm(z2, log(z), 0.1))

u = runif(n, 0, 1)
for (i in 1:n){
if (u[i] < min(1, ratios[i])){
z[i] = z2[i]
}
}
return(z)
}

# predictive density
pred.y = function(zvec, alpha, sqbeta, v, step){
# return: predicted y and z
# zvec is different from above, z with user-defined length
n = length(zvec)
lastz = zvec[n]
for (k in 1:step){
new_w = rinvgamma(1, v/2 + alpha, v*sqbeta/2 + lastz)
newy = rnorm(1, 0, sqrt(new_w))
newz = rgamma(1, alpha, rate = 1/new_w)
lastz = newz
}
structure(list(pred_y = newy, pred_z = newz))
}


#------------------------------
v_vec = c(3:20)
init.sqbeta = 0.0001
```

```
alpha = 0.05
iter = 10000
w = rinvgamma(n.y, 3/2, 3*init.sqbeta/2)
z = sapply(w, function(x) rgamma(1, alpha, 1/x))


# begin sampling
vsamples = rep(0, iter)
sqbetasamples = rep(0, iter)
alphasamples = rep(0, iter)
predict_y = c()
ypred.samples = matrix(0, nrow = iter, ncol = n.y)
sqbeta = init.sqbeta
last_w = c()
last_z = c()

for (i in 1:iter){
print(i)
# update parameters
vsamples[i] = post.v(w, z, alpha, sqbeta, v_vec)
sqbetasamples[i] = MH.sqbeta(w, z, alpha, vsamples[i], sqbeta)
alphasamples[i] = MH.alpha(w, z, alpha, vsamples[i], sqbetasamples[i])

# update z
z = MH.z(w, z, alphasamples[i], sqbetasamples[i], vsamples[i])
last_z = c(last_z, z[n.y])

# update w
w = post.w(y, z, alphasamples[i], sqbetasamples[i], vsamples[i])
last_w = c(last_w, w[n.y])

# model checking
for (j in 2:n.y){
   ypred.samples[i, j] = pred.y(z[1:(j-1)], alphasamples[i],
                                sqbetasamples[i], vsamples[i], step = 1)$pred_y
}

if (i > iter*0.4){
   # predict new y, 20 steps ahead to eliminate dependence
   # check if the future can mimic the past
   ynew = pred.y(z, alphasamples[i], sqbetasamples[i], vsamples[i],
          step = 20)$pred_y
   predict_y = c(predict_y, ynew)
}
```

```
# loop
sqbeta = sqbetasamples[i]
alpha = alphasamples[i]
}


# Posterior of beta^2
plot(sqbetasamples[5000:iter], type = 'l', ylab = bquote(beta^2),
main = expression(paste(beta^2, ' from SV')))
paste('The posterior mean of beta^2 is', mean(sqbetasamples[5000:iter]))

# Posterior of degree of freedom: v
hist(vsamples, breaks = 100, main = 'Degree of freedom v')

# Posterior of alpha
plot(alphasamples[6000:iter], type='l', ylab = bquote(alpha),
main = expression(paste(alpha, ' from SV')))
paste('The posterior mean of alpha is', mean(alphasamples[6000:iter]))

# Model checking: predictive distribution of y vs original
par(mfrow = c(1,2))
hist(predict_y, breaks = 100, xlab = 'log return', xlim = c(-0.2, 0.2),
main = 'SV histogram of predicted y \n20 steps ahead')
hist(orig.y, breaks = 50, xlab = 'log return', xlim = c(-0.2, 0.2),
main = 'Histogram of y in training set')

# 95% credible intervals for training log return prediction
mean_pred = apply(ypred.samples[5000:iter,], 2, mean)
bounds = apply(ypred.samples, 2, function(z) quantile(z, c(0.025, 0.975)))
plot(orig.y, type = 'l', main = 'SV 95% credible intervals for log return\n from
     2005 to 2009', ylab = 'Log return', ylim = c(-0.12, 0.12))
lines(bounds[1,], col = 'blue')
lines(bounds[2,], col = 'red')
lines(mean_pred, col = 'green')


# Stock price prediction on test data
meansqbeta = mean(sqbetasamples[5000:iter])
meanalpha = mean(alphasamples[6000:iter])
# the majority of v
meanv = 3
test.ypred = rep(0, n.y2)
trained_z = last_z[9001:iter]
test.bounds = matrix(0, nrow = 2, ncol = n.y2)
```

```
for (k in 1:n.y2){
print(k)
testsamples = rep(0, 1000)
for (t in 1:1000){
pred = pred.y(c(z[1:(n.y+k-2)], trained_z[t]), alpha = meanalpha,
meansqbeta, meanv, step = 1)
testsamples[t] = pred$pred_y

if (t != iter){
trained_z[t+1] = pred$pred_z
 }
}

# 95% credible intervals of the prediction
testbound = quantile(testsamples, c(0.025, 0.975))
test.bounds[1,k] = exp(testbound[1])
test.bounds[2,k] = exp(testbound[2])

# expected value
test.ypred[k] = mean(exp(testsamples))
}
test.p = test.sp500[-(n.y2+1),4]*test.ypred
test.plower = test.sp500[-(n.y2+1),4]*test.bounds[1,]
test.pupper = test.sp500[-(n.y2+1),4]*test.bounds[2,]

# generate plots of predictions
plot(test.sp500[,4], type = 'l', main = 'SV prediction on test S&P 500')
lines(test.p, col = 'green')
lines(test.plower, col = 'red')
lines(test.pupper, col = 'blue')

# MSE of predictions from SV model
mse.sv = mean((test.sp500[,4]-test.p)^2)
```

5 Standard GARCH(1,1)

```
# compare with standard GARCH(1,1)
library(fGarch)
garchpred = rep(0, n.y2)
garchbounds = matrix(0, nrow = 2, ncol = n.y2)
m = garchFit(formula = ~garch(1,1), data = y)
coeff = coef(m)
lasth = m@h.t[n.y]
lastu = m@residuals[n.y]
```

```
# the model is:
# y_t = mu + u_t
# u_t = sqrt(h_t)v_t
# h_t = omega + alpha*u_{t-1}^2 + beta*h_{t-1}

# One-step ahead prediction with fixed parameters from training data
for (k in 1:n.y2){
h_t = coeff['omega'] + coeff['alpha1']*lastu^2 + coeff['beta1']*lasth
u_t = sqrt(h_t)*rnorm(1000)

# expected value
garchpred[k] = mean(exp(coeff['mu'] + u_t))

# 95% credible intervals of the prediction
bounds = quantile(exp(coeff['mu'] + u_t), c(0.025, 0.975))
garchbounds[1,k] = bounds[1]
garchbounds[2,k] = bounds[2]

lasth = h_t
lastu = full_y[n.y+k] - coeff['mu']
}

test.pgarch = test.sp500[-(n.y2+1),4]*garchpred
test.pgarch.lower = test.sp500[-(n.y2+1),4]*garchbounds[1,]
test.pgarch.upper = test.sp500[-(n.y2+1),4]*garchbounds[2,]

# generate plots of predictions
plot(test.sp500[,4], type = 'l', main = 'GARCH prediction on test stock prices')
lines(test.pgarch, cex = 0.1, col = 'green')
lines(test.pgarch.lower, cex = 0.1, col = 'blue')
lines(test.pgarch.upper, cex = 0.1, col = 'red')

# MSE of predictions from standard GARCH(1,1)
mse.garch2 = mean((test.sp500[,4]-test.pgarch)^2)
```

References

Baillie, R. T., & Bollerslev, T. (1989). The Message in Daily Exchange Rates: A Conditional-Variance Tale. *Journal of Business & Economic Statistics, 7(3),* 297-305.

Barndorff-Nielsen, O. E. (1997). Normal Inverse Gaussian Distributions and Stochastic Volatility Modelling. *Scand J Stat Scandinavian Journal of Statistics, 24(1),* 1-13.

Blattberg, R. C., & Gonedes, N. J. (1974). A Comparison of the Stable and Student Distributions as Statistical Models for Stock Prices. *The Journal of Business J BUS, 47(2),* 244.

Bollerslev, T., Chou, R. Y., & Kroner, K. F. (1992). ARCH modeling in finance. *Journal of Econometrics, 52*(1-2), 5-59.

Bollerslev, T. (1986, February). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31(3),* 307-327.

Bollerslev, T. (1987, August). A Conditionally Heteroskedastic Time Series Model for Speculative Prices and Rates of Return. *The Review of Economics and Statistics, 69(3),* 542-547.

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). The econometrics of financial markets. Princeton, *NJ: Princeton University Press*.

Chib, S., Nardari, F., & Shephard, N. (2002). Markov chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics, 108(2),* 281-316.

Clark, P. K. (1973). A Subordinated Stochastic Process Model with Finite Variance for Speculative Prices. *Econometrica, 41(1),* 135.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. Quantitative *Finance, 1(2),* 223-236.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*. Econometrica, 50(4),* 987.

Giovannini, A., & Jorion, P. (1989). The Time Variation of Risk and Return in the Foreign Exchange and Stock Markets. The Journal of Finance, 44(2), 307-325.

Graham, B. (2006). *The intelligent investor: The definitive book on value investing*. New York: Harper.

Hautsch, N., & Ou, Y. (2008). Discrete-Time Stochastic Volatility Models and MCMC-Based Statistical Inference. SSRN Electronic Journal. doi:10.2139/ssrn.1292494

Kon, S. J. (1984). Models of Stock Returns-A Comparison. The Journal of Finance, 39(1), 147-165.

Milosevic, N. (2016). Equity Forecast: Predicting Long Term Stock Price Movement using Machine Learning. *Journal of Economics Library, Vol2, No2 (2016).*

Pitt, M. K., & Walker, S. G. (2005, June). Constructing Stationary Time Series Models Using Auxiliary Variables With Applications. *Journal of the American Statistical Association, 100(470),* 554-564.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica, 59(2),* 347.

Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes,a study of daily sugar prices 1961-79, in O. D. Anderson (ed.), *Time Series Analysis:Theory and Practice 1,* North-Holland, Amsterdam, pp. 203–226.