Georgia Southern University

Digital Commons@Georgia Southern

Electronic Theses and Dissertations

Graduate Studies, Jack N. Averitt College of

Spring 2016

GARCH(1,1) with Sifted Gamma-distributed Errors

Alan C. Budd

Follow this and additional works at: https://digitalcommons.georgiasouthern.edu/etd

Part of the Longitudinal Data Analysis and Time Series Commons

Recommended Citation

Budd, Alan C., "GARCH(1,1) with Sifted Gamma-distributed Errors" (2016). *Electronic Theses and Dissertations*. 1409.
https://digitalcommons.georgiasouthern.edu/etd/1409

This thesis (open access) is brought to you for free and open access by the Graduate Studies, Jack N. Averitt College of at Digital Commons@Georgia Southern. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.

GARCH(1,1) WITH SHIFTED GAMMA-DISTRIBUTED ERRORS

by

ALAN CANDLER BUDD

(Under the Direction of Tharanga Wickramarachchi)

ABSTRACT

Typical General Autoregressive Conditional Heteroskedastic (GARCH) processes involve normally-distributed errors, and they model strictly-positive error processes poorly. This thesis will present a method for estimating the parameters of a GARCH(1,1) process with shifted Gamma-distributed errors, conduct a simulation study to test the method, and apply the method to real time series data.

INDEX WORDS: Time Series, GARCH, Gamma distribution, strictly-positive errors

2009 *Mathematics Subject Classification*: Statistics

GARCH(1,1) WITH SHIFTED GAMMA-DISTRIBUTED ERRORS

by

ALAN CANDLER BUDD

B.S., Georgia Southern University, 2014

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment

of the Requirement for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

©2016

ALAN CANDLER BUDD

All Rights Reserved

GARCH(1,1) WITH SHIFTED GAMMA-DISTRIBUTED ERRORS

by

ALAN CANDLER BUDD

Major Professor:	Tharanga Wickramarachchi
Committee:	Patricia Humphrey
	Stephen Carden
	Arpita Chatterjee

Electronic Version Approved:

May 2016

DEDICATION

This thesis is dedicated to the girl who always supported me through my research and studies; who always challenged me to become a better person and mathematician; who saw things in me that I never saw in myself. Thank you, Kerigan LeMoine; my favorite Georgia Southern University alumna, and my best friend.

ACKNOWLEDGMENTS

I wish to acknowledge the hard work and patience of my major professor, Dr. Tharanga Wickramarachchi, without whom this thesis would surely not be possible. I would also like to acknowledge the three professors who kindly volunteered to be on my thesis committee: Drs. Patricia Humphrey, Arpita Chatterjee, and Stephen Carden. Finally, I'd like to acknowledge both the Department of Mathematical Sciences and the School of Nursing at Georgia Southern University, for providing me with both the knowledge and financial assistance needed to complete my Master of Science degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview of Financial Time Series Analysis

One of the most crucial assumptions in classical time series analysis, as well as other statistical analysis techniques, is constant variance. This leads to the most popular linear time series models, such as the autoregressive moving average (ARMA) model, posed by Peter Wittle [12]. However, strides have been made by many to relax the assumption of constant variance to account for real-world properties portrayed by financial time series. The two greatest strides made for nonlinear modeling of time series come from Robert F. Engle [3] and Tim Bollerslev [2]. Engle, who won a Nobel Peace Price in Economics for his work, developed the *autoregressive conditional heteroskedastic* (ARCH) time series model [3]. Bollerslev [2] went on to generalize this model, coining the *generalized autoregressive conditional heteroskedastic* (GARCH) model. Both of these models focus on creating a conditional variance equation, allowing for nonconstant variance.

This thesis focuses primarily on the GARCH(1,1) model. As Bollerselv [2] defines it, the random error terms are assumed to be distributed normally. However, for our research, the GARCH(1,1) model with error terms distributed non-normally, in particular shifted Gamma, is considered. For the remainder of this chapter, facts regarding log-returns of financial data are presented, as well as key definitions of stationary time series processes. Also in Chapter 1, the definition for the GARCH(1,1) process of Bollerslev [2], as well as properties of this model, are introduced. In Chapter 2, properties of the GARCH(1,1) model with shifted Gamma-distributed errors are presented and proven. In Chapter 3, a method for estimating the parameters in the GARCH(1,1) model of interest is proposed, and a simulation study is conducted to test the method. In Chapter 4, log-return data of exchange rates is analyzed using the GARCH(1,1) model with shifted Gamma-distributed errors. Chapter 5 concludes the thesis with remarks and potential future study.

1.2 Log-Returns

Suppose there is a discrete financial time series process, P_t, observed at equally spaced times t. Many authors, such as Mikosch and Stărică [11] have asserted the usefulness of using a log-transformation of the financial time series process, rather than just the process itself. This log-transformation, called the *log-returns* is mathematically defined as

$$x_t = \log P_t - \log P_{t-1}, \tag{1.1}$$

where $t = 0, 1, 2, 3, \cdots$. Figure 1.1 show two time series plots of the Dow Jones Industrial Average (DJIA) from January 2010 to December 2014. The plot on the left is of the daily closing price. The plot on the right is of the log-return daily closing price. The log-return daily closing price exposes two primary attributes: it is de-trended (from the differencing), and the volatility of closing prices is exposed (with obvious non-constant volatility, which was highest around late 2011).

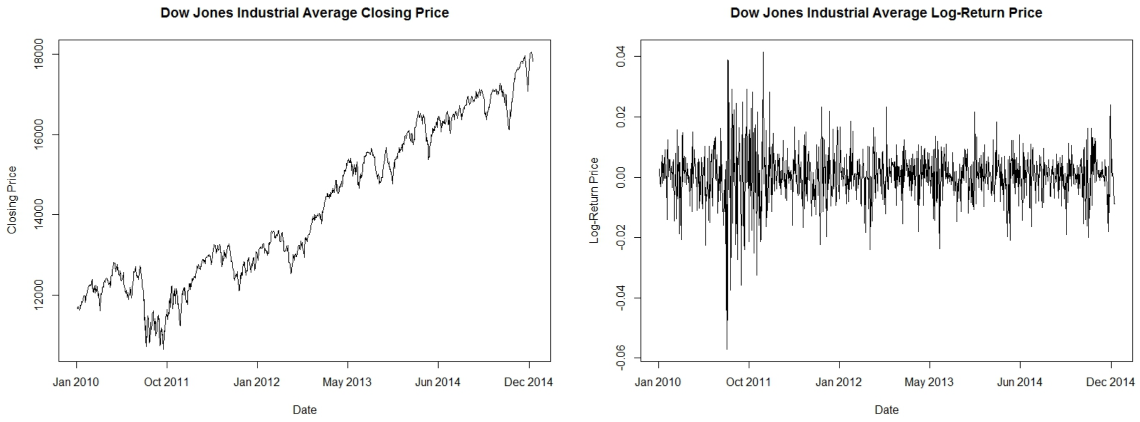


Figure 1.1: Time Series Plots of the Dow Jones Industrial Average (DJIA)

Mikosch and Stărică [11] state the following three properties of log-returns of financial time series data:

1. The distribution of log-returns is not normal, but rather with a high peak at zero and heavy-tailed. This comes from most days having little change, i.e. a ”flat” day.
2. There is dependence in the tails of the distribution. These tails represent major changes in returns, which do not happen often, but tend to trend together.
3. There is long-range dependence in the log-returns process.

To help visualize some of the properties, consider property 1. Figure 1.2 is the kernel density of the log-return daily closing prices of the DJIA. This kernel density has a very high peak, with heavy tails. This affirms the first property listed above.

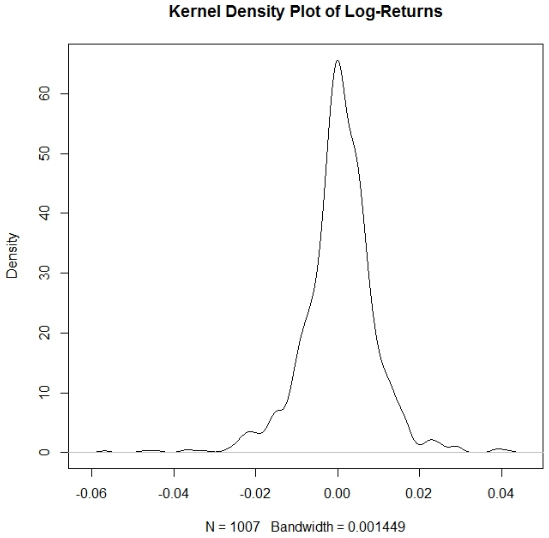


Figure 1.2: Kernel Density Plot of Log-Return Daily Closing Prices of DJIA

1.3 Classical GARCH(1,1) Process

Many models have been proposed to capture the listed effects. However, the most widely used is the classical GARCH(1,1) model of Tim Bollerslev [2]. This model is mathematically defined as

$$X_t = \sigma_t \epsilon_t;\ \epsilon_t \sim \text{iid Normal}(0,1), \tag{1.2}$$

$$\sigma_t^2 = \omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2, \tag{1.3}$$

where α, β, and γ are called the GARCH(1,1) parameters. It is well-known that the variance structure of X_t can be rewritten as an ARMA(1,1) process, as defined by Engle[3], i.e.

$$X_t - \phi X_{t-1} = Z_t + \theta Z_{t-1};\ Z \sim \text{WN}(0,1),$$

where WN denotes a strong white noise process (a process with mean 0 and no correlation at different times). The derivation of this fact is as follows: From Equations 1.2 and 1.3, the difference $\nu_t = X_t^2 - \sigma_t^2$ is

$$\begin{aligned}
X_t^2 - \sigma_t^2 &= X_t^2 - (\omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2) \\
&= X_t^2 - \omega - \alpha X_{t-1}^2 - \beta \sigma_{t-1}^2 \\
&= X_t^2 - \omega - \alpha X_{t-1}^2 - \beta \sigma_{t-1}^2 + (\beta X_{t-1}^2 - \beta X_{t-1}^2) \\
&= X_t^2 - \omega + \beta (X_{t-1}^2 - \sigma_{t-1}^2) - (\alpha + \beta) X_{t-1}^2 \\
\nu_t &= X_t^2 - \omega + \beta \nu_t - (\alpha + \beta) X_{t-1}^2 \\
X_t^2 &= \omega + (\alpha + \beta) X_{t-1}^2 - (1 - \beta) \nu_t \\
X_t^2 - (\alpha + \beta) X_{t-1}^2 &= \omega - (1 - \beta) \nu_t.
\end{aligned}$$

If it can be shown that ν_t follows a white noise process, this sufficiently proves that X_t^2 can be written as an ARMA(1,1) process.

$$\begin{aligned}
E[\nu_t] &= E[X_t^2 - \sigma_t^2] \\
&= E[\sigma_t^2(\epsilon_t^2 - 1)] \\
&= E[\sigma_t^2]E[\epsilon_t^2 - 1] \\
E[\epsilon_t^2 - 1] &= E[\epsilon_t^2] - 1 = 1 - 1 = 0 \\
&\Rightarrow E[\nu_t] = 0
\end{aligned}$$

$$\begin{aligned}
E[\nu_t\nu_{t-h}] &= E[(X_t^2 - \sigma_t^2)(X_{t-h}^2 - \sigma_{t-h}^2)] \\
&= E[\sigma_t^2(\epsilon_t^2 - 1)\sigma_{t-h}^2(\epsilon_{t-h}^2 - 1)] \\
&= E[\sigma_t^2\sigma_{t-h}^2]E[(\epsilon_t^2 - 1)(\epsilon_{t-h}^2 - 1)] \\
&= E[\sigma_t^2\sigma_{t-h}^2]E[(\epsilon_t^2 - 1)][E[(\epsilon_{t-h}^2 - 1)] \\
&\Rightarrow E[\nu_t\nu_{t-h}] = 0
\end{aligned}$$

1.4 Stationarity

One of the most prominent, important characteristics of time series processes is *stationarity*.

Definition 1.1 (Strictly Stationary). *Let $\{X_t\}$ be a time series process with joint cumulative distribution function $F_X(x_{t_1}, x_{t_2}, \cdots, x_{t_k})$. The process $\{X_t\}$ is called* ***strictly stationary*** *if*

$$F_X(x_{t_1}, x_{t_2}, \cdots, x_{t_k}) = F_X(x_{t_{h+1}}, x_{t_{h+2}}, \cdots, x_{t_{h+k}})$$

for all time lags h.

Definition 1.2 (Weakly Stationary). *Let $\{X_t\}$ be a time series process such that $E[X_t^2] < \infty$ (finite second moment). The process $\{X_t\}$ is called* ***weakly stationary*** *if*

1. *$E[X_t]$ is independent of t*

2. *$Cov[X_t, X_{t+h}]$ is independent of t for all time lags h*

Both of these definitions define a time series process in which statistical characteristics, such as mean and covariance, are constant over time. This consistency over all time possible time lags is important for statistical inference, such as estimating model parameters and forecasting. As implied by the names, proving strict stationarity is a stronger result than proving weak stationarity, since the equivalence of joint cumulative distributions over time directly implies weak stationarity. In practice, however, proving weak stationarity under certain conditions is sufficient for valid statistical inference, and can prove to be significantly more efficient than proving strict stationarity.

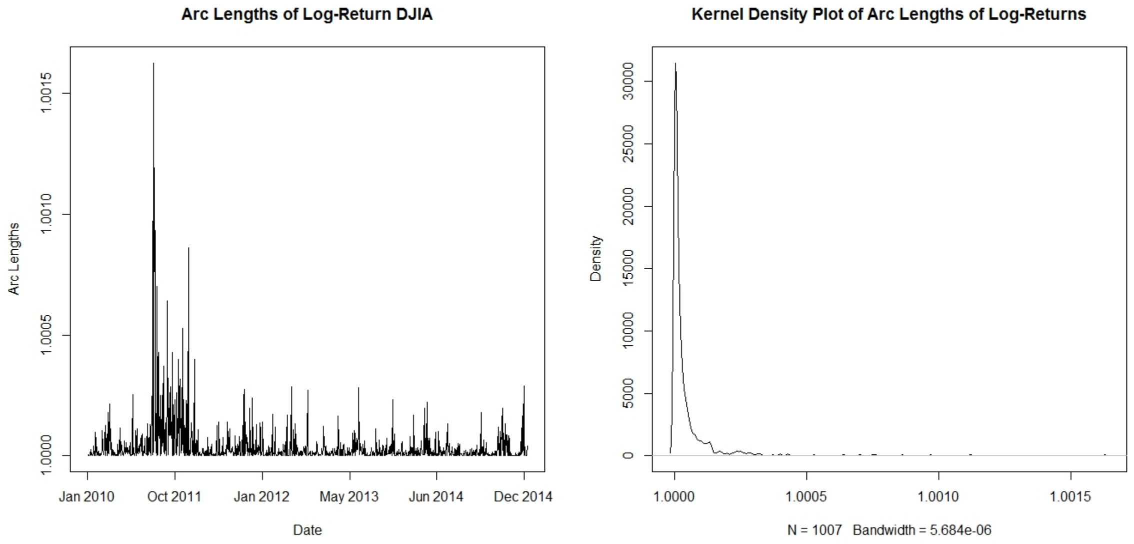
1.5 Motivation

The original intent for this thesis was to investigate a potential model for the *arc length* of time series data. For a time series at times $t_1, t_2, \cdots, t_n$, define the k-step ahead arc length as

$$A_n = \sum_{j=2}^{n} \sqrt{(t_j - t_{j-k})^2 + (X_{t_j} - X_{t_{j-k}})^2} \tag{1.2}$$

Typically, k is considered to be one. This is a natural, intuitive measure of volatility in time series analysis. Asymptotic behavior of these arc lengths is investigated by Wickramarachchi et. al [13]. If applied to log-return financial data, the arc length transformation can give insight into what time periods the data is less/more volatile (possibly seasonality). Additionally, since the arc lengths of daily stock price return vary across different time periods, if an adequate ARCH(q) or GARCH(p, q) model can be fit to arc length data, then inference can be used to help predict future volatility of stocks through forecasting. Also, if an adequate model can be fit to arc length data, then cluster analysis could potentially be used to help group stocks by volatility, which could then be used by investment firms to help investment decisions.

Trying to model arc length data, where $A_t = \sqrt{1 + (X_t - X_{t-1})^2}$, (not the sum of all arc lengths, as given in Equation 1.2; rather, individual arc length measures) with a GARCH(p,q) process, however, becomes problematic. The issue derives from the assumption in Equation 1.2, where the error terms are distributed as strong white noise. It is obvious that the distribution of arc lengths is strictly positive; see Figure 1.3. This eliminates the possibility of negative error terms, meaning that the error assumption required for GARCH(p,q) processes is violated. Therefore, a new type of time series model must be considered: one that both takes into account varying volatility across time periods, and error terms that are strictly positive.



Arc Lengths of Log-Return DJIA

Kernel Density Plot of Arc Lengths of Log-Returns

Figure 1.3: Plots of Arc Lengths of Log-Returns

CHAPTER 2

GARCH(1,1) WITH GAMMA-DISTRIBUTED ERRORS

2.1 The Model

As stated in the introduction, a typical GARCH(1,1) process has the mathematical form

$$X_t = \sigma_t \epsilon_t, \tag{2.1}$$

where $\epsilon_t \sim N(0,1)$, and the recursive equation for σ_t is

$$\sigma_t^2 = \omega + \alpha X_{t-1} + \beta \sigma_{t-1}^2. \tag{2.2}$$

However, the basis of this thesis is that ϵ_t is not normally distributed. GARCH processes with innovations that follow a student t-distribution have been discussed by Bollerslev [1]. Even though some discussion can be found about modeling option prices and exchange rates using skewed distributions, no extensive study has been carried out in the literature. In this study we assume that innovations of a GARCH(1,1) process follow a shifted gamma distribution as follows:

$$X_t = \sigma_t \gamma_t, \tag{2.3}$$

$$\sigma_t^2 = \omega + \alpha X_{t-1} + \beta \sigma_{t-1}^2. \tag{2.4}$$

where γ_t follows a *shifted Gamma distribution*, defined by the equation

$$\gamma_t = \epsilon_t - \frac{a}{b};\ \epsilon_t \sim \Gamma(a,b). \tag{2.5}$$

Note that ϵ has probability density function

$$f_{\epsilon_t}(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-xb}. \tag{2.6}$$

In this section we will solve Equations 2.3 and 2.4 in order to obtain the necessary and sufficient conditions that guarantee the existence of a unique strictly (weakly) stationary solution.

First, it is beneficial to express Equation 2.2 explicitly, rather than recursively. For a given time t, and with a maximum time lag n, that derivation is

$$\begin{aligned}
\sigma_t^2 &= \omega + \alpha X_{t-1}^2 + \beta \sigma_{t-1}^2 \\
&= \omega + \alpha(\sigma_{t-1}^2 \gamma_{t-1}^2) + \beta \sigma_{t-1}^2 \\
&= \omega + (\alpha \gamma_{t-1}^2 + \beta)\sigma_{t-1}^2 \\
&= \omega + (\alpha \gamma_{t-1}^2 + \beta)(\omega + \alpha X_{t-2}^2 + \beta \sigma_{t-2}^2) \\
&= \omega + (\alpha \gamma_{t-1}^2 + \beta)(\omega + \alpha(\sigma_{t-2}^2 \gamma_{t-2}^2) + \beta \sigma_{t-2}^2) \\
&= \omega + (\alpha \gamma_{t-1}^2 + \beta)(\omega + (\alpha \gamma_{t-2}^2 + \beta)\sigma_{t-2}^2) \\
&\vdots \\
&= \omega \sum_{i=1}^{t} \prod_{j=1}^{i-1} M_t(j) + \sigma_{t-n} \prod_{j=1}^{n} M_t(j), \qquad (2.6)
\end{aligned}$$

where $M_t(j) = \alpha \gamma_{t-j}^2 + \beta$, and the empty product has been set to one.

Of particular interest for this process is determining under what circumstances the process has stationary solutions, both strict and weak. The necessary and sufficient condition for strict stationarity must ensure that the summation term in Equation 2.6 converges and the product term disappears at the limit. Thus the necessary and sufficient condition for the stationarity will follow from the existence of the random variable

$$S^2 = \omega \sum_{i=1}^{\infty} \prod_{j=1}^{i-1} \alpha \gamma_{-j}^2 + \beta \qquad (2.7)$$

Note that, when the random variable S^2 exists we get $S^2 = \sigma_0^2$.

Before those theorems are introduced and proven, extra notation is necessary. Particularly, define $\log^+(x)$ as

$$\log^+(x) = \log(\max\{x, 1\}) \qquad (2.7)$$

Theorem 2.1 (Strictly Stationary Solution)**.** *Suppose that $\{X_t | t \in \mathbb{Z}\}$ is a GARCH(1,1) process specified by Equations 2.3 and 2.4 that satisfies*

$$E\left[log^+(\alpha \gamma_0^2 + \beta)\right] < \infty$$

Then $\{X_t|t \in \mathbb{Z}\}$ *admits a unique, non-anticipative, strictly stationary solution if and only if*

$$E\left[log(\alpha\gamma_0^2 + \beta)\right] < 0 \tag{2.8}$$

Proof. Alluding to Equation 2.6,

$$\begin{aligned}\sigma_t^2 &= \omega \sum_{i=1}^{n} \prod_{j=1}^{i-1} M_t(j) + \sigma_{t-n} \prod_{j=1}^{n} M_t(j) \\ &= S_n^2 + R_n^2,\end{aligned}$$

where $S_n^2 = \omega \sum\limits_{i=1}^{n} \prod\limits_{j=1}^{i-1} (\alpha\gamma_{t-j}^2 + \beta)$ and $R_n^2 = \sigma_{t-n}^2 \prod\limits_{j=1}^{n} (\alpha\gamma_{t-j}^2 + \beta)$.
First we will show that R_n^2 converges to zero almost surely.

$$\begin{aligned}R_n^2 &= \sigma_{t-n}^2 \prod_{j=1}^{n} (\alpha\gamma_{t-j}^2 + \beta) \\ &= \sigma_{t-n}^2 \text{exp}\left\{ n \left[\frac{1}{n} \sum_{j=1}^{n} \text{ln}(\alpha\gamma_{t-j}^2 + \beta) \right] \right\}.\end{aligned}$$

By the Strong Law of Large Numbers, for a fixed t, as $n \to \infty$,

$$\frac{1}{n} \sum_{j=1}^{n} \text{ln}(\alpha\gamma_{t-j}^2 + \beta) \xrightarrow{a.s.} E\left[\text{ln}(\alpha\gamma_0^2 + \beta)\right].$$

Therefore,

$$R_n^2 \approx \sigma_{t-n}^2 \text{exp}\left\{ n \left[E\left[\text{ln}(\alpha\gamma_0^2 + \beta) \right] \right] \right\}.$$

Since σ_{t-n}^2 is bounded, $R_n^2 \xrightarrow{a.s.} 0$ if and only if $E\left[\text{ln}(\alpha\gamma_0^2 + \beta)\right] < 0$.

Now we see that as $n \longrightarrow \infty$,

$$\sigma_0^2 = S^2.$$

Therefore we need to show that the random variable S^2 converges if and only if E $\left[log|\alpha\gamma_0^2 + \beta|\right] < 0$.

$$
\begin{aligned}
S^2 &= \omega \sum_{i=1}^{\infty} \prod_{j=1}^{i-1} (\alpha \gamma_{-j}^2 + \beta) \\
&= \omega \sum_{i=1}^{\infty} exp \left\{ (i-1) \sum_{j=1}^{i-1} \frac{1}{i-1} ln(\alpha \gamma_{-j}^2 + \beta) \right\}.
\end{aligned}
$$

By the Strong Law of Large Numbers, for a fixed t, as $i \to \infty$,

$$\sum_{j=1}^{i-1} \frac{1}{i-1} ln(\alpha \gamma_{t-j}^2 + \beta) \xrightarrow{a.s.} E\left[ln(\alpha \gamma_0^2 + \beta) \right].$$

Therefore,

$$S^2 \approx \omega \sum_{i=1}^{n} exp \left\{ (i-1) \sum_{j=1}^{i-1} E\left[ln(\alpha \gamma_0^2 + \beta) \right] \right\}.$$

Therefore, S^2 converges if and only if E $\left[ln(\alpha \gamma_0^2 + \beta) \right] < 0$, which concludes the proof. □

Theorem 2.2 (Weakly Stationary Solution)**.** *Suppose that* $\{X_t | t \in \mathbb{Z}\}$ *is a GARCH(1,1) process specified by equations 2.3 and 2.4. Then* $\{X_t | t \in \mathbb{Z}\}$ *admits a unique weakly stationary solution if and only if*

$$\frac{a}{b^2} \alpha + \beta < 1 \tag{2.10}$$

Proof. To prove weak stationarity, it only needs to be shown that $E[X_t]$ and $E[X_t^2]$ are independent of t. Since $E[\gamma_t] = 0$, $E[X_t] = E[\sigma_t]E[\gamma_t] = 0$. Next, $E[X_t^2] = E[\sigma_t^2]E[\gamma_t^2] = E[\sigma_t^2]\frac{a}{b^2}$. Therefore, if it can be shown that $E[\sigma_t^2]$ is independent of t, then the proof is complete.

$$E\left[\sigma_t^2\right] = \omega + \omega E\left[M_t(1) M_t(2) \cdots M_t(i) \right]$$

$$\begin{aligned}
E\left[M_t(1)\cdots M_t(k)\right] &= E\left[\alpha\gamma_{t-k}^2\left(M_t(1)\cdots M_t(k-1)\right)+\beta\left(M_t(1)\cdots M_t(k-1)\right)\right] \\
&= \alpha E\left[\gamma_{t-k}^2\right]E\left[\left(M_t(1)\cdots M_t(k-1)\right)\right]+\beta E\left[\left(M_t(1)\cdots M_t(k-1)\right)\right] \\
&= \alpha\left(\frac{a}{b^2}\right)E\left[\left(M_t(1)\cdots M_t(k-1)\right)\right]+\beta E\left[\left(M_t(1)\cdots M_t(k-1)\right)\right] \\
&= \left(\alpha\frac{a}{b^2}+\beta\right)E\left[\left(M_t(1)\cdots M_t(k-1)\right)\right] \\
&\vdots \\
&= \left(\alpha\frac{a}{b^2}+\beta\right)^k
\end{aligned}$$

Therefore,

$$\begin{aligned}
E\left[\sigma_t^2\right] &= \omega+\omega\sum_{i=1}^{\infty}\left(\alpha\frac{a}{b^2}+\beta\right)^i \\
&= \omega\sum_{i=0}^{\infty}\left(\alpha\frac{a}{b^2}+\beta\right)^i \\
&= \frac{\omega}{1-\left(\alpha\frac{a}{b^2}+\beta\right)}
\end{aligned}$$

It is apparent that $E\left[\sigma_t^2\right]$ is independent of t. The condition under equation 2.10 comes from the convergence criteria for an infinite geometric series, requiring the radius be less than 1. Thus this proves the condition for a weak stationary solution. □

2.2 Maximum Likelihood Estimation

The probability distribution function (pdf) and the corresponding log-likelihood function is derived for the GARCH(1,1) model with shifted-Gamma distributed errors.

$$\begin{aligned}
P(X_t \leq x_t) &= P(\sigma_t \gamma_t \leq x_t) \\
&= P\left(\sigma_t \left(\epsilon_t - \frac{a}{b}\right) \leq x_t\right) \\
&= P\left(\epsilon_t \leq \frac{x_t}{\sigma_t} + \frac{a}{b}\right) \\
f_{X_t}(x_t) &= \frac{1}{\sigma_t} f_{\epsilon_t}\left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right) \\
&= \frac{1}{\sigma_t} \frac{b^a}{\Gamma(a)} \left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right)^{a-1} exp\left\{-b\left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right)\right\} \\
\text{ln}(f) &= -\text{ln}(\sigma_t) + a\text{ln}(b) - \text{ln}(\Gamma(a)) + (a-1)\text{ln}\left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right) - b\frac{x_t}{\sigma_t} - a \\
&= -\text{ln}(\sigma_t) + a(\text{ln}(b) - 1) - \text{ln}(\Gamma(a)) + (a-1)\text{ln}\left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right) - b\frac{x_t}{\sigma_t}
\end{aligned}$$

Therefore, the conditional log-likelihood function of the observations $x_1, x_2 \ldots x_n$ of a SG-GARCH(1,1) process when x_0 and σ_0^2 are given can be written as,

$$L = \sum_{t=1}^{n} \text{ln}(\sigma_t) - na(\text{ln}(b) - 1) - n\text{ln}(\Gamma(a)) + (a-1)\sum_{t=1}^{n} \text{ln}\left(\frac{x_t}{\sigma_t} + \frac{a}{b}\right) - b\sum_{t=1}^{n} \frac{x_t}{\sigma_t} \quad (2.11)$$

Since σ_t^2 also depends on parameters (ω, α, β), one must solve for σ_t^2 before maximizing the likelihood function. When σ_0^2 and x_0 are given, the solution for σ_t^2 can be written in the following form.

$$\sigma_t^2 = \omega \sum_{i=0}^{t-1} \beta^i + \beta^t \sigma_0^2 + \alpha \sum_{i=0}^{t-1} \beta^i x_{t-1-i}^2$$

Hence, the maximum likelihood estimates of the parameters can be estimated after substituting σ_t^2 into the likelihood function and maximizing it with respect to the GARCH(1,1) parameters and the two Gamma distribution parameters.

In order to better understand the behavior of the log-likelihood function, a visualization is useful. However, this can be challenging, since there is no useful visualization for six-dimensional space. In order visualize the log-likelihood function, the Gamma distribution parameters a and b, as well as the GARCH(1,1) parameter ω, are fixed. Therefore, the only values that vary will be α and β, and different values of these two parameters will be plotted against the log-likelihood function. Two plots will be presented to show the behavior of the log-likelihood function:

1. A two-dimensional plot with colors corresponding to the intensity of the log-likelihood.
2. A three-dimensional plot to help visualize increasing/decreasing behavior.

In order to construct the plots, one set of data are simulated (reference Section 3.1) with parameter values $\vec{\theta} = (a, b, \omega, \alpha, \beta) = (2, 3, 1E^{-6}, 0.1, 0.8)$. Using simulated data, the log-likelihood function is computed for fixed values of a, b, and ω corresponding to the actual parameter values, and varying values of α and β. Figure 2.1 is the two-dimensional plot, and Figure 2.2 is the three-dimensional plot. Note that for both plots, the values for β range only from 0.6 to 1, because for all other values, the log-likelihood is zero. A value of zero for the log-likelihood function represents the unfeasible region for the parameters α and β, based on Equation 2.10.

Both figures reveal a decreasing pattern for the log-likelihood for increasing values of β in the feasible region. This would suggest a peak at around $\beta = 0.8$, which is expected. However, it is difficult to observe in Figure 2.2 by the naked eye what the behavior of the log-likelihood is for increasing values of α. In order to better visualize the behavior, β is fixed at the actual parameter value, 0.8, and the log-likelihood function is plotted against the different values of α (Figure 2.3).

As it can be seen in Figure 2.3, there is a peak at around $\alpha = 0.1$, which is also expected. The result of the two peaks for corresponding values of α and β shows sufficient evidence that there should be a global maximum for the log-likelihood function in general.

2.3 Numerical Estimation of MLE

This process for finding maximum likelihood estimators works well for processes with nicely-defined gradient vectors, and with few parameters. However, in the case of the

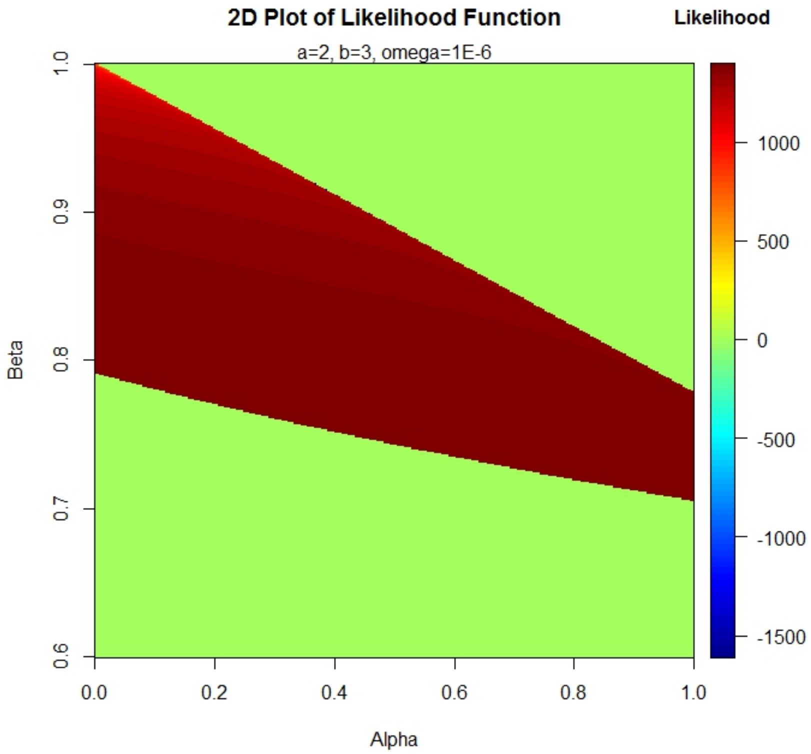


Figure 2.1: Two-Dimensional Plot of Log-Likelihood Values

GARCH(1,1) model with shifted-Gamma distributed errors, there are five parameters of interest, leading to a 5x5 system of equations, and even though the gradient vector is analytic, deriving the gradient vector is messy, and then solving the resulting system of equations would be tedious.

Considering the impracticality of analytically deriving the maximum likelihood estimates of five parameters, numerical methods to find maximum likelihood estimates is a viable, practical alternative. If a numerical method can provide consistently accurate maximum likelihood estimates for the five parameters of a GARCH(1,1) model with shifted Gamma-distributed errors for a given set of data, this can be used in practice for forecasting future data points. For this thesis, two different numerical methods are considered, each with distinct advantages and disadvantages.

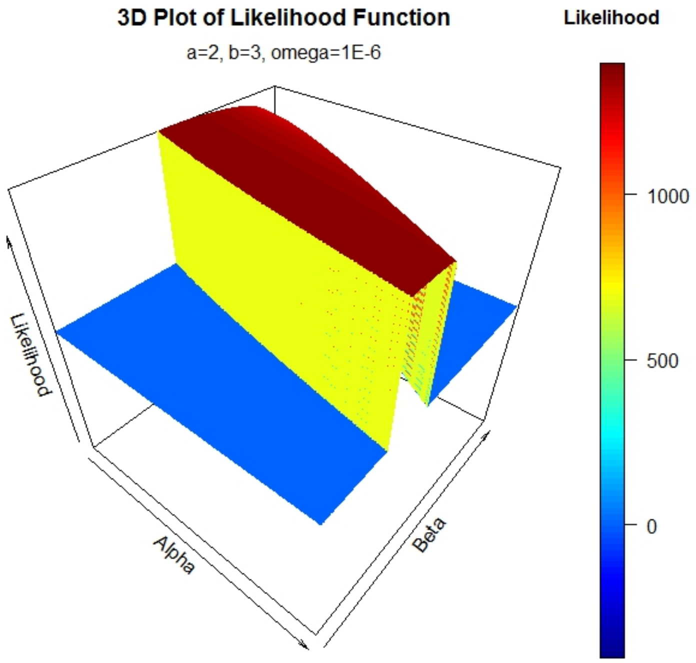


Figure 2.2: Three-Dimensional Plot of Log-Likelihood Values

1. **Grid Search**

   A grid search algorithm take a range of values for every parameter of an objective function (the negative log-likelihood function, $-l(\vec{\theta})$), and evaluates the objective function for every distinct combination of parameters values. Once all of these values are computed, the minimum is computed. This can be used as a tool to obtain an initial set of estimates for model parameters. In R, the function `gridSearch` from the *"NMOF"* package [5] can be used to implement a grid search algorithm. The only required inputs are the objective function and box constraints for each of the parameters.

   Advantages

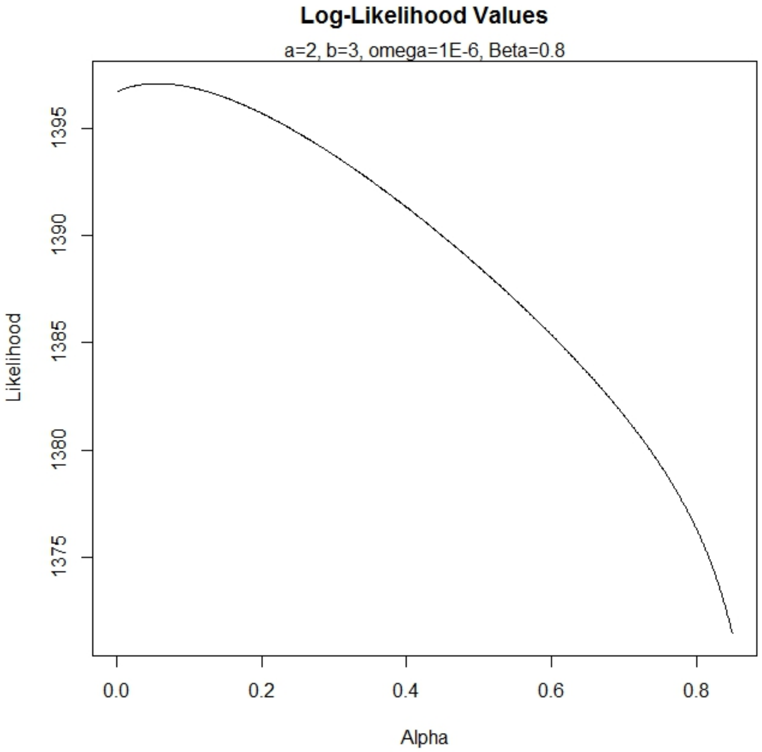


Figure 2.3: Two-Dimensional Plot of Log-Likelihood Values with Fixed Beta

- A grid search algorithm does not require an initial guess, guaranteeing convergence to a result.
- A grid search algorithm is more likely to find a global maximum, rather than a local maximum.
- The results of a grid search algorithm are consistent across different random samples with similar parameter values.

Disadvantages

- A grid search algorithm requires exceedingly high computation time, especially with an objective function with many parameters.
- The results of grid search algorithm can not exceed the box constraints set by

the user. If the real parameters exceeds the box constraint set, the grid search will achieve a perfect result with probability 0.

2. **Nonlinear Constrained Optimization**

   When an objective function is nonlinear, a numerical implementation of a nonlinear optimization is appropriate. Furthermore, many optimization problems involve constraints, either linear or nonlinear, on the parameters in the objective function. Therefore, nonlinear constrained optimization can be implemented for these types of problems. The negative log-likelihood function is obviously nonlinear, and Equation 2.10 can be seen as a nonlinear constraint on the objective function.

   There are numerous algorithms for nonlinear constrained optimization, each falling under one of two categories: those requiring an explicit form of the gradient vector, and those that do not. Given the complexity of the negative log-likelihood function, the latter is considered. Of those algorithms that do not require an explicit form for the gradient vector, there are global and local maximum algorithms. Examples of global maximum algorithms are the DIRECT algorithm by Jones et.al. [7] and the controlled random search with a local mutation modification posed by Kaleo and Ali [8]. These algorithms, and numerous more, can be implemented in R using the `nloptr` function in the *"nloptr"* package [6].

   Advantages

   - The nonlinear constrained optimization algorithms are generally quick in computation time.
   - If a nonlinear constrained optimization algorithm can converge to an accurate result consistently, then precision can be increased with little cost to computation time.

- Most of the algorithms require the parameters to be bounded, but high/low finite bounds can be set if the user is unsure of actual parameter values, with little cost to computation time.

Disadvantages

- The results for the nonlinear constrained optimization algorithms seem to be inconsistent across different algorithms, especially for higher number of parameters.
- An initial guess close to the actual parameter values is required to guarantee convergence, which is troublesome for objective functions with many parameters.

Given the inconsistent results of the nonlinear constrained optimization, a grid search algorithm to find the minimum of the negative log-likelihood function is chosen. Further comments can be found in the conclusion chapter.

CHAPTER 3

NUMERICAL STUDY

3.1 Simulation Study

As mentioned in Section 2.3, a grid search algorithm will be used to compute numerical maximum likelihood estimates for the GARCH(1,1) model with shifted Gamma-distributed errors. Note that this is not the optimal method for parameter estimation since the likelihood is maximized over a set of pre-selected values for respective parameters in a gird. Future study to find an alternative algorithm for more precise, faster estimation is discussed in Chapter 4.

It is of interest to determine the accuracy of these estimates. To this end, two separate calculations are useful: bias and mean squared error (MSE). Bias is mathematically defined as

$$\text{Bias}\left(\vec{\theta}, \hat{\vec{\theta}}\right) = E\left[\hat{\vec{\theta}} - \vec{\theta}\right] = E\left[\hat{\vec{\theta}}\right] - \vec{\theta}, \tag{3.1}$$

where $\hat{\vec{\theta}}$ is the estimator of the parameters of interest. Bias is a measure of how far off (and in which direction) the expected value of the estimator is from the actual parameter value(s). Mean squared error is mathematically defined as

$$\text{MSE} = E\left[\left(\vec{\theta} - \hat{\vec{\theta}}\right)^2\right] = Var\left(\hat{\vec{\theta}}\right) + \left[Bias\left(\vec{\theta}, \hat{\vec{\theta}}\right)\right]^2. \tag{3.2}$$

Mean Squared Error is a measure of deviation of an estimator from the real parameter value(s) under a squared-loss function. Both of these measures paint a picture of how well an estimator performs. As described in Chapter 2, however, an explicit equation for the maximum likelihood estimator would be extremely tedious, if not impossible, to analytically derive. The best alternative, therefore, is to estimate the bias and MSE numerically. This can be done through a simulation study.

The general idea of a simulation study is that data are generated under a process with set, fixed parameters. For this instance, a GARCH(1,1) process with shifted-Gamma errors is simulated into data. Once data are generated, a numerical estimation process is applied to the data to generate an estimate for the parameter(s) of interest. For this instance, a grid search algorithm is implemented to find a maximum likelihood estimate. This process is repeated a fixed number of times. After a collection of estimates is computed, some kind of accuracy/consistency measure is taken. For this instance, the accuracy measures are bias and mean squared error.

A psuedocode for the simulation study involving maximum likelihood estimation of the GARCH(1,1) model is provided in Algorithm 1. Let $\Gamma(a, b)$ denote a random sample of a Gamma distribution with shape parameter a and scale parameter b.

```
input  : $N$, $n$, $a$, $b$, $\omega$, $\alpha$, $\beta$
output: Bias, MSE
for i=1 to N do
    Simulate Data
    $\sigma_0 = \sqrt{\frac{\omega}{1-(\alpha\frac{a}{b^2}+\beta)}}$;
    $\gamma_0 = \Gamma(a,b) - \frac{a}{b}$;
    $x_0 = \sigma_0\gamma_0$;
    for j=1 to n do
        $\sigma_j = \sqrt{\omega + \alpha x_{j-1}^2 + \beta s_{j-1}^2}$;
        $\gamma_j = \Gamma(a,b) - \frac{a}{b}$;
        $x_j = \sigma_j\gamma_j$;
    end
    $x = \{x_0, \ldots, x_n\}$;
    Parameter Estimation
    $\hat{\vec{\theta}} = \vec{\hat{\theta}}$ such that $max\{L(\vec{\theta})\}$ is achieved;
end
Bias $\left(\vec{\theta}, \hat{\vec{\theta}}\right) = E\left[\hat{\vec{\theta}} - \vec{\theta}\right]$;
MSE $= E\left[\left(\vec{\theta} - \hat{\vec{\theta}}\right)^2\right]$;
```

Algorithm 1: Psuedocode for Simulation Study

Algorithm 1 is implemented in the statistical program R, with many computations involving the *"base"* package [9]. For the parameter estimation step, a grid search algorithm is used. In particular, the R function `gridSearch` under the *"NMOF"* package [5] is used. The R code for the simulation study, as well as the graphics generated in Chapters 1 and 2, can be found in Appendix A.

3.2 Simulation Study Results

For this thesis, three different simulation studies are conducted for varying values of α and β. The three set of parameter values are

1. $\vec{\theta_1} = (a, b, \omega, \alpha_1, \beta_1) = (2, 3, 1E^{-6}, 0.3, 0.6)$
2. $\vec{\theta_2} = (a, b, \omega, \alpha_2, \beta_2) = (2, 3, 1E^{-6}, 0.2, 0.7)$
3. $\vec{\theta_3} = (a, b, \omega, \alpha_3, \beta_3) = (2, 3, 1E^{-6}, 0.1, 0.8)$

For each simulation study, 100 estimates for $\vec{\theta}$ are numerically computed using a grid search algorithm, each estimate coming from a data set of 250 simulated observations of a GARCH(1,1) process with shifted Gamma-distributed errors. Once these estimates are computed numerically, both bias and mean squared error are estimated. Table 3.1 contains average estimated parameter values, and Table 3.2 contains estimated biases, and Table 3.3 contains estimated mean squared errors.

Table 3.1: Table of Average Estimated Parameter Values

$\vec{\theta}$	$\bar{\hat{a}}$	$\bar{\hat{b}}$	$\bar{\hat{\omega}}$	$\bar{\hat{\alpha}}$	$\bar{\hat{\beta}}$
$(2, 3, 1E^{-6}, 0.3, 0.6)$	2.650	3.215	$7.700E^{-7}$	0.461	0.489
$(2, 3, 1E^{-6}, 0.2, 0.7)$	2.375	3.030	$7.300E^{-7}$	0.425	0.542
$(2, 3, 1E^{-6}, 0.1, 0.8)$	2.060	2.895	$7.450E^{-7}$	0.406	0.579

Table 3.2: Table of Estimated Biases

$\vec{\theta}$	Bias$_{\hat{a}}$	Bias$_{\hat{b}}$	Bias$_{\hat{\omega}}$	Bias$_{\hat{\alpha}}$	Bias$_{\hat{\beta}}$
$(2, 3, 1E^{-6}, 0.3, 0.6)$	0.650	0.215	$-2.30E^{-7}$	0.161	-0.111
$(2, 3, 1E^{-6}, 0.2, 0.7)$	0.375	0.030	$-2.70E^{-7}$	0.225	-0.158
$(2, 3, 1E^{-6}, 0.1, 0.8)$	0.060	-0.105	$-2.55E^{-7}$	0.306	-0.221

Table 3.3: Table of Estimated Mean Squared Errors

$\vec{\theta}$	MSE$_{\hat{a}}$	MSE$_{\hat{b}}$	MSE$_{\hat{\omega}}$	MSE$_{\hat{\alpha}}$	MSE$_{\hat{\beta}}$
$(2, 3, 1E^{-6}, 0.3, 0.6)$	4.00	6.16	2.69	1.79	1.89
$(2, 3, 1E^{-6}, 0.2, 0.7)$	3.12	5.62	2.69	1.87	1.80
$(2, 3, 1E^{-6}, 0.1, 0.8)$	2.26	5.73	2.69	1.91	1.82

One note from the numerical study is that as values of α decrease and β increase, the average parameter estimates of those two also decrease and increase, respectively. This means that the average estimates are tending to the correct direction. Also, since the bias (defined in Equation 3.1) of α is always positive, α is generally overestimated. Similarly, since the bias of β is negative, β is generally underestimated.

3.3 Application to Exchange Rate Data

For a real-data application, data regarding the exchange rate of the U.S. Dollar to the Euro from March 1, 2015 until February 29, 2016, is used. The data was extracted from the International Monetary Fund website [4]. Figure 3.1 shows the time series and density plots for the exchange rate data. As mentioned previously, it is advantageous to work with

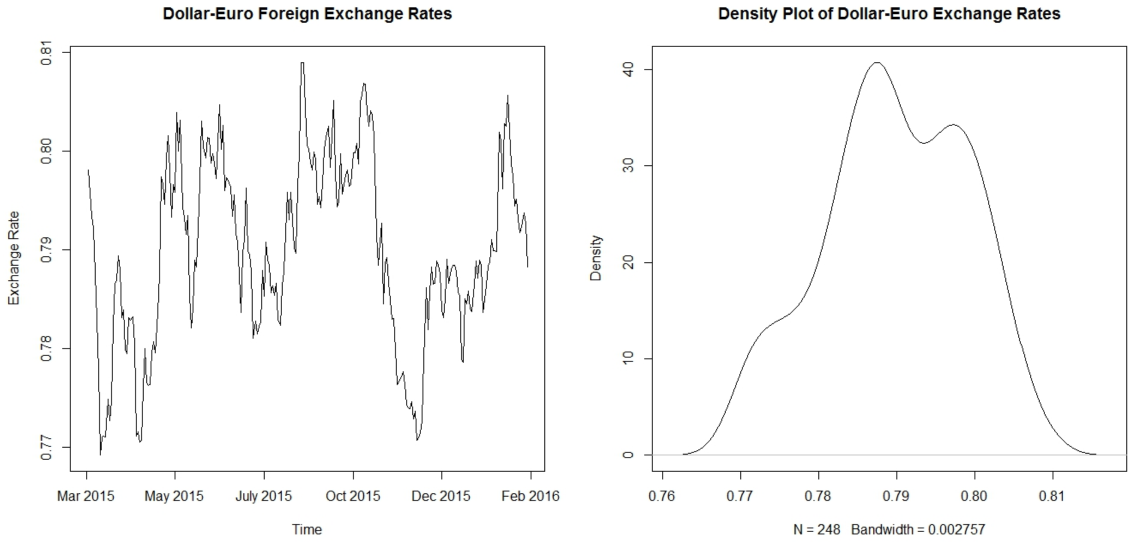


Figure 3.1: Plots for Exchange Rate Data

the log-returns of financial data, rather than the raw data. Figure 3.2 shows the time series and density plots of the log-returns of the exchange rate data.

Note that in Figure 3.2, there is a slight skew to the right in the density plot of the log-returns, and literature suggests employing a skewed conditional distribution in analyzing exchange rates [10], meaning that a shifted Gamma distribution can be appropriate for a GARCH(1,1) model. Given this information, the grid search algorithm for estimating the parameters for an SG-GARCH(1,1) model is applied to the *mean-shifted* log-returns of the exchange rate data. The estimated parameters of the SG-GARCH(1,1) model is given as

$$\hat{\vec{\theta}}_{SG-GARCH} = (\hat{a}, \hat{b}, \hat{\omega}_1, \hat{\alpha}_1, \hat{\beta}_1) = (74.5, 13.0, 1E^{-7}, 0.042, 0.980) \tag{3.3}$$

Although the estimates for the Gamma parameters are quite large, the estimated variance, $\frac{\hat{a}}{\hat{b}^2}$, is less than 1. A large estimate for the shape parameter a is expected, since this corresponds to a shape closer to symmetric, as seen in Figure 3.2.

In order to evaluate the accuracy of the parameter estimates, prediction can be used to determine how close the predicted values are to the actual shifted log-returns. However,

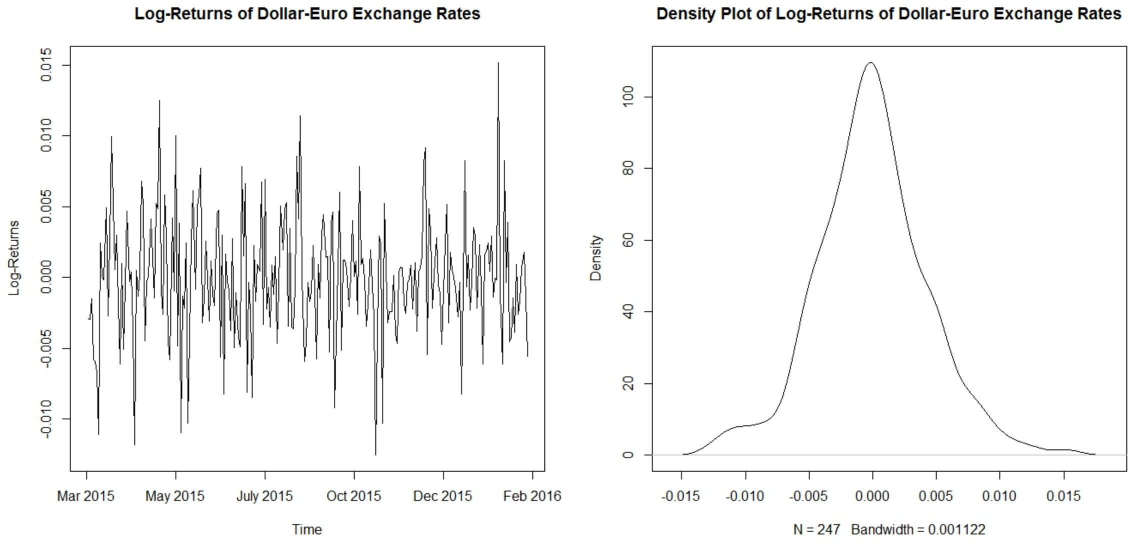


Figure 3.2: Plots for Exchange Rate Data

prediction must be conducted on the squared values of the log-returns. This leads to the concept of *volatility predicting*. Predicting the square values of the log-returns is necessary because trying to predict the log-returns, X_t, assuming they follow a SG-GARCH process, given previous data Ψ, yields

$$E[X_t|\Psi] = E[\sigma_t\gamma_t|\Psi] = E[\sigma_t|\Psi]E[\gamma_t|\Psi] = E[\sigma_t|\Psi]E[\gamma_t] = E[\sigma_t|\Psi](0) = 0,$$

meaning that every predicted value will be 0. Therefore, prediction on X_t^2 is conducted, yielding

$$E[X_t^2|\Psi] = E[\sigma_t^2\gamma_t^2|\Psi] = E[\sigma_t^2|\Psi]E[\gamma_t^2|\Psi] = E[\sigma_t^2|\Psi]E[\gamma_t^2] = \sigma_t^2\frac{a}{b^2}. \tag{3.4}$$

Therefore, the prediction of X_t^2 is

$$\hat{X}_t^2 = \hat{\sigma}_t^2\frac{\hat{a}}{\hat{b}^2}. \tag{3.5}$$

It follows from Equation 3.5 that if prediction is to be conducted on X_t^2, σ_t^2 must also be predicted. This can be easily done using Equation 2.2, i.e.

$$\widehat{\sigma_t^2} = \hat{\omega} + \hat{\alpha}\hat{X}_{t-1}^2 + \hat{\beta}\hat{\sigma}_{t-1}^2, \tag{3.6}$$

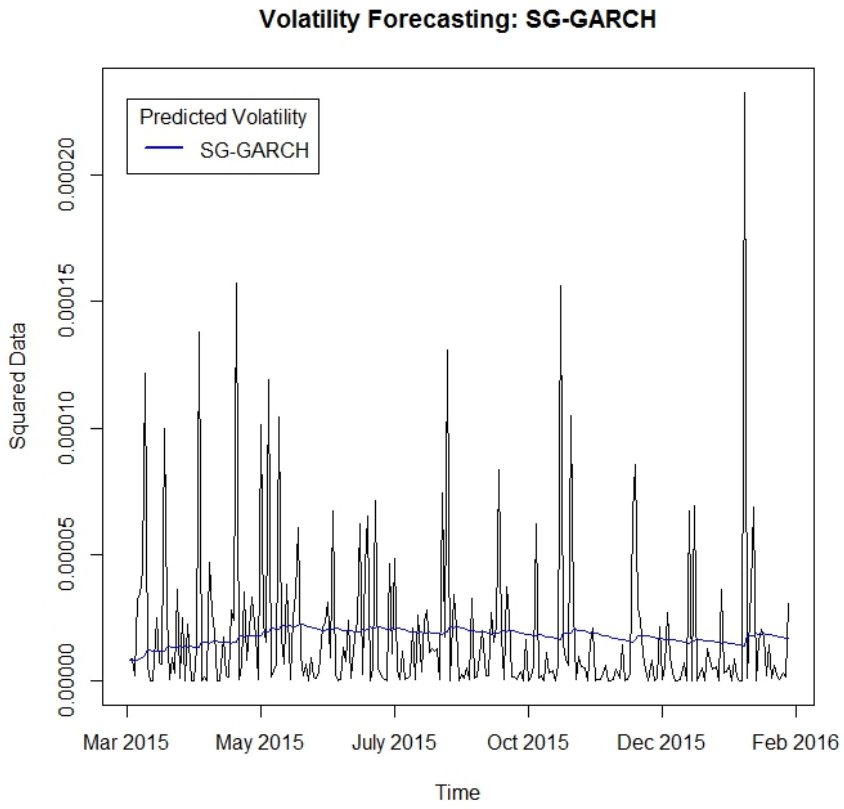


Figure 3.3: Volatility Forecasting using SG-GARCH Model

where σ_0^2 can be predicted as the mean of the squared mean-shifted log-returns, i.e.

$$\hat{\sigma}_0^2 = \bar{X}_t^2. \tag{3.7}$$

Given the initial estimates $\hat{X}_0^2$ and σ_0^2, predicted values are computed by alternating between Equations 3.5 and 3.6, where each of the predicted parameters come from Equation 3.3.

Figure 3.3 shows a comparison of the squared mean-shifted log-returns and the forecasted volatilities using a SG-GARCH(1,1) model, as well as two classical GARCH(1,1) models, one with normally-distributed errors, the other with t-distributed errors. This was done in order to compare the forecasted volatilities of the SG-GARCH model to see if the

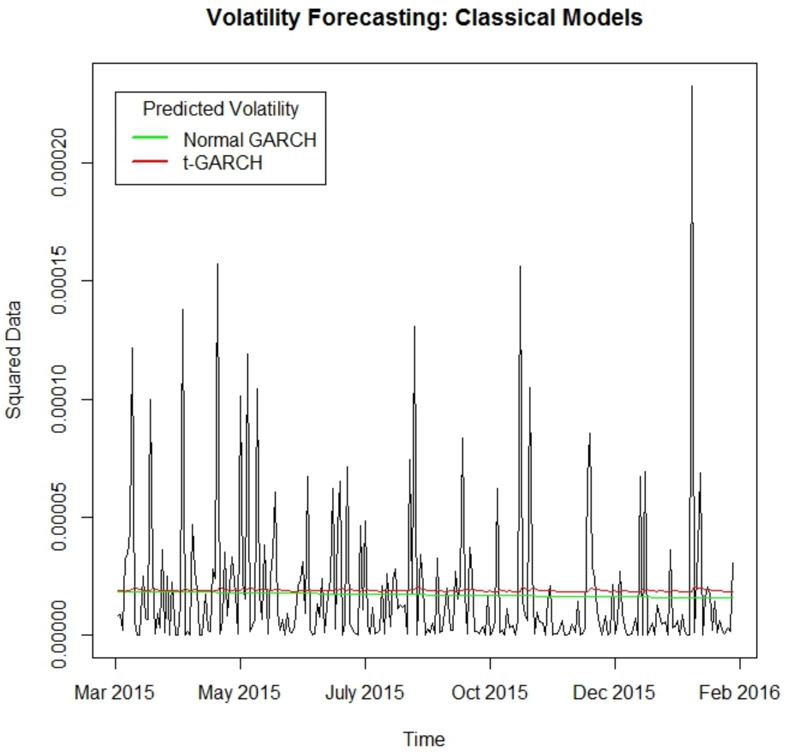


Figure 3.4: Volatility Forecasting using Classical GARCH Models

forecasted volatilities are better or worse than those of the more classical GARCH models. These classical model parameters were calculated using the R package *fGarch* [14]. The parameter estimates for each of those models are

$$\hat{\vec{\theta}}_{\text{GARCH-Normal}} = (\hat{\omega}_2, \hat{\alpha}_2, \hat{\beta}_2) = (1.878E^{-11}, 1.000E^{-8}, 0.999)$$
$$\hat{\vec{\theta}}_{\text{GARCH-t}} = (\hat{\omega}_3, \hat{\alpha}_3, \hat{\beta}_3) = (6.307E^{-6}, 0.011, 0.658)$$

For the classical GARCH model with normally distributed errors, the predicted volatilities do not follow the pattern closely. For the GARCH model with t-distributed errors, the forecasted volatilities appear to follow a similar pattern to the squared log-returns, but not as closely as the SG-GARCH model. Although the predicted volatilities from the SG-GARCH model are far from perfect, they still appear to respond to the volatility pattern of the log-returns better than the other two models.

CHAPTER 4

CONCLUSION

4.1 Overview of Results

In this thesis, a prominent problem in financial time series analysis was presented in the form of modeling a strictly-positive conditionally distributed time series with varying volatility. To address this problem, a new class of GARCH(1,1) with shifted Gamma distributed errors (SG-GARCH) was introduced. Conditions for strict and weak stationary solutions to the SG-GARCH(1,1) process were presented and proven.

The likelihood function for the GARCH(1,1) process was also derived. In order to find maximum likelihood estimates for the process with a complicated likelihood function, numerical estimates were considered. Numerically estimating maximum likelihood estimates were generated using a grid search algorithm. Nonlinear constrained optimization was desired, but given the observed inconsistencies of that algorithm, the former was used. The author notes that using a grid search algorithm for obtaining numerical estimates is limited in practice. For the SG-GARCH model to be useful in practice, nonlinear constrained optimization must be used.

A simulation study was conducted to test the results of the grid search algorithm for finding numerical estimates for maximum likelihood estimates. The simulation study found that as the GARCH parameters α and β were changed, the maximum likelihood estimates also changed the correct direction, although the bias of the estimates was obvious. Finally, the grid search algorithm to compute maximum likelihood estimates for the SG-GARCH(1,1) model was applied to the log-returns of exchange rate data, and the parameter estimates were used to estimate volatility. It was shown that, although far from perfect, the SG-GARCH estimates lead to better volatility forecasting than classical GARCH models.

4.2 Future Research

Future research in the uses and implementation of SG-GARCH parameter estimation is planned. First and foremost, the problems regarding the inconsistencies of nonlinear constrained optimization need to be resolved for that algorithm to be applied to finding maximum likelihood estimates for the parameters of interest. This will allow for more precise, accurate, and timely estimates. Secondly, once nonlinear constrained optimization is implemented well, using the model for volatility forecasting must be improved. Once that is improved, the SG-GARCH model can be used for volatility clustering, forecasting, and many other useful applications, particularly in financial time series analysis.

Another possible route for future research is standardizing the error by subtracting the mean and dividing by the standard deviation of the errors. This will force the mean and variance of the errors to be zero and one, respectively. Doing this would allow for estimation of the mean and standard deviation, rather than the shape and scale parameters. This is advantageous for interpretation, rather than trying to interpret shape and scale parameters.

REFERENCES

[1] T. Bollerslev. A conditionally heteroskedastic time sereis model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3):542–547, 1987.

[2] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307 – 327, 1986.

[3] Robert Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[4] International Monetary Fund. Exchange rate archives by month, 2016. `https://www.imf.org/external/np/fin/data/param_rms_mth.aspx`.

[5] Manfred Gilli, Dietmar Maringer, and Enrico Schumann. *Numerical Methods and Optimization in Finance*. Academic Press, Waltham, MA, USA, 2011.

[6] Steven G. Johnson. The nlopt nonlinear-optimization package.

[7] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *J. Optim. Theory Appl.*, 79(1):157–181, October 1993.

[8] P. Kaelo and M. M. Ali. Some variants of the controlled random search algorithm for global optimization. *Journal of Optimization Theory and Applications*, 130(2):253–264, 2006.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. `http://www.R-project.org/`.

[10] Howell Tong Tak Kuen Siu and Hailiang Yang. On pricing derivatives under garch models: A dynamic gerber-shiu's approach. *North American Actuarial Journal*, 8:17–31, 2004.

[11] Catalin Starica Thomas Mikosch. Limit theory for the sample autocorrelations and extremes of a garch (1, 1) process. *The Annals of Statistics*, 28(5):1427–1451, 2000.

[12] Peter Whittle and Thomas J. Sargent. *Prediction and Regulation by Linear Least-Square Methods*. University of Minnesota Press, ned - new edition edition, 1983.

[13] Tharanga D. Wickramarachchi, Colin Gallagher, and Robert Lund. Arc length asymptotics for multivariate time series. *Applied Stochastic Models in Business and Industry*, 31(2):264–281, 2015.

[14] Diethelm Wuertz and Yohan Chalabi. *Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*. RMetrics, 2013. `http://www.Rmetrics.org`.

Appendix A

R CODE

Appendix A contains R code for the simulation study, as well as code for the graphics generated in Chapters 1 and 2.

```
#####Simulation Study######


###Create Function to Simulate Data###
##N - Number of Repetitions Desired##
##n - Size of Simulated Data Sets##
##a,b - Gamma Parameters##
##omega,alpha,beta - GARCH(1,1) Parameters##

simulation=function(N,n,a,b,omega,alpha,beta){
set.seed(5) #Set Random Seed for Replicability#
require(NMOF)

#Initialization#

theta=matrix(0,N,5)
lev=list(seq(.5,4,.5),seq(.5,4,0.5),
seq(0.0000005,.0000015,.0000005),seq(0.01,0.99,0.01),seq(0.01,0.99,0.01))

#Create Function for Data Simulation#

sim=function(n,a,b,omega,alpha,beta){
s=rep(0,n)
e=rep(0,n)
```

```
x=rep(0,n)
s[1]=sqrt((omega/(1-(a*alpha/b^2+beta))))
e[1]=rgamma(1,a,b)-(a/b)
x[1]=s[1]*e[1]
for (i in 1:n){
s[i+1]=sqrt(omega+beta*s[i]^2+alpha*x[i]^2)
e[i+1]=rgamma(1,a,b)-(a/b)
x[i+1]=s[i+1]*e[i+1]}
x=x[1:n]
s=s[1:n]
return(x)}


#Create Negative Log-Likelihood Function#
#Requires Levels Input#

L=function(w){
n=length(x)
s1=0
s2=0
s3=0
s0=w[3L]/(1-(w[1L]*w[4L]/w[2L]^2+w[5L]))

if((w[1L]*w[4L]/w[2L]^2+w[5L])>=1){ #Stationarity Constraint#
like=0}

else{
for (i in 1:n){
```

```
s1[i]=w[5L]^(i-1)

s2[i]=w[5L]^(i-1)*x[n-i+1]^2

s3[i]=w[5L]^(i)}

s1=cumsum(s1)

s2=cumsum(s2)


s=sqrt(w[3L]*s1+s3*s0+w[4L]*s2)


if(min(x/s+w[1L]/w[2L])<=0){

like=0}

else{

q=x/s+w[1L]/w[2L]


like=sum(log(s))-n*w[1L]*(log(w[2L])-1)+n*log(gamma(w[1L]))-

     (w[1L]-1)*sum(log(q))+w[2L]*sum(x/s)

}}

return(like)}


for (i in 1:N){

x=sim(n,a,b,omega,alpha,beta)

sol=suppressMessages(gridSearch(fun=L,levels=lev))

theta1[i,]=sol$minlevels

}

return(theta1)

}


#Set Values of Theta#
```

```
theta1=c(2,3,1e-6,.3,.6)

theta2=c(2,3,1e-6,.2,.7)

theta3=c(2,3,1e-6,.1,.8)


#Create 100 Estimates for Each Value of Theta#


g1=simulation(100,250,2,3,1e-6,.3,.6)

g2=simulation(100,250,2,3,1e-6,.2,.7)

g3=simulation(100,250,2,3,1e-6,.1,.8)


#Calculate Average Estimation#


theta1_hat_bar=colMeans(g1)

theta2_hat_bar=colMeans(g2)

theta3_hat_bar=colMeans(g3)


#Calculate Bias#


bias1=colMeans(g1)-theta1

bias2=colMeans(g2)-theta2

bias3=colMeans(g3)-theta3


#Calculate MSE#


MSE1=colMeans((g1-theta1)^2)

MSE2=colMeans((g2-theta1)^2)

MSE3=colMeans((g3-theta1)^2)
```

```
#####Graphics Code for 2D and 3D Plots#####

##Simulate Data##

x=sim(n,a,b,omega,alpha,beta)

#Create Positive Log-Likelihood Function#

L2=function(w){
n=length(x)
s1=0
s2=0
s3=0
s0=w[3L]/(1-(w[1L]*w[4L]/w[2L]^2+w[5L]))

if((w[1L]*w[4L]/w[2L]^2+w[5L])>=1){
like=0}

else{
for (i in 1:n){
s1[i]=w[5L]^(i-1)
s2[i]=w[5L]^(i-1)*x[n-i+1]^2
s3[i]=w[5L]^(i)}
s1=cumsum(s1)
s2=cumsum(s2)

s=sqrt(w[3L]*s1+s3*s0+w[4L]*s2)
if(min(x/s+w[1L]/w[2L])<=0){
like=0}
else{
q=x/s+w[1L]/w[2L]

like=-sum(log(s))+n*w[1L]*(log(w[2L])-1)-
     n*log(gamma(w[1L]))+(w[1L]-1)*sum(log(q))-w[2L]*sum(x/s)
}}
return(like)}

#Calculate Likelihood Function Values#
#Full range for alpha, beta from .6 to .999#

alpha1=seq(.001,.999,.001)
beta1=seq(.6,.999,.001)
z1=matrix(0,999,400)
for (i in 1:999){
for (j in 1:400){
w=c(2,3,1e-6,alpha1[i],beta1[j])
z1[i,j]=L2(w)}}

alpha2=seq(.001,.85,.001)
z2=rep(0,length(alpha2))
for (i in 1:length(alpha2)){
z2[i]=L2(c(2,3,1e-6,alpha2[i],.8))}

beta2=seq(.78,.975,.001)
```

```
z3=rep(0,length(beta2))
for (i in 1:length(beta2)){
z3[i]=L2(c(2,3,1e-6,.1,beta2[i]))}

#Create 2D Plot with Colors Corresponding to Likelihood Function Values#

image2D(z1,y=seq(.6,1,length.out = ncol(z1)),xlab="Alpha",ylab="Beta",
clab="Likelihood",main="2D Plot of Likelihood Function")
mtext("a=2, b=3, omega=1E-6")

#Create 3D Plot#

persp3D(z=z1,clab="Likelihood",xlab="Alpha",ylab="Beta",zlab="Likelihood",
main="3D Plot of Likelihood Function")
mtext("a=2, b=3, omega=1E-6")

#Create 2D Plots for Alpha and Beta#

plot(alpha2,z2,type="l",main="Log-Likelihood Values",xlab="Alpha",
ylab="Likelihood")
mtext("a=2, b=3, omega=1E-6, Beta=0.8")

plot(beta2,z3,type="l",main="Log-Likelihood Values",xlab="Beta",
ylab="Likelihood")
mtext("a=2, b=3, omega=1E-6, Alpha=0.1")


#####Application: Exchange Rate Data#####
```

```
#Load Data, Create Plots for Exchange Rates#
setwd("D:/Research") #Sets Working Directory for Loading Data#
rate=scan("rate.txt")
rate1=ts(rate)
plot(rate1,main="Dollar-Euro Foreign Exchange Rates",ylab="Exchange Rate",
xaxt="n")
axis(1,at=c(0,50,100,150,200,250),labels=c("Mar 2015","May 2015","July 2015",
"Oct 2015","Dec 2015","Feb 2016"))
d1=density(rate)
plot(d1,main="Density Plot of Dollar-Euro Exchange Rates")
```

```
#Create Log-Returns of Exchange Rates#
n=length(rate)
lreturn=log(rate[-1]/rate[-n])
d2=density(lreturn)
plot(d2,main="Density Plot of Log-Returns of Dollar-Euro Exchange Rates")
lreturnts=ts(lreturn)
plot(lreturnts,xaxt="n",main="Log-Returns of Dollar-Euro Exchange Rates",
ylab="Log-Returns")
axis(1,at=c(0,50,100,150,200,250),labels=c("Mar 2015","May 2015","July 2015",
"Oct 2015","Dec 2015","Feb 2016"))
```

```
#Attempt to Fit Classical GARCH(1,1) Models for MS Log-Returns#
fit1=garchFit(formula=˜garch(1,1),data=x,include.mean=FALSE,trace=FALSE)
fit2=garchFit(formula=˜garch(1,1),data=x,include.mean=FALSE,trace=FALSE,
cond.dist="std")
```

```
v1=volatility(fit1,"h")
v2=volatility(fit2,"h")

#Estimate Parameters for SG-GARCH(1,1) Model#
x=lreturn-mean(lreturn)
lev=list(seq(67,75,.5),seq(10,17,0.5),seq(0.1e-6, 1e-6, 0.1e-6),
seq(0.03,0.045,0.001),seq(0.99,0.9,-0.01)))
sol=gridSearch(fun=L,levels=lev)
t_hat=sol$minlevels

#Compute Forecasted Volatilities#


omega=t_hat[3]
alpha=t_hat[4]
beta=t_hat[5]
s0=mean(x^2)
sig=0
sig[1]=omega+alpha*s0+beta*s0
for (i in 2:length(x)){
sig[i]=omega+alpha*x[i-1]^2+beta*sig[i-1]}
sig=sig*(a/b^2)

#Create Comparison Plots#
plot(ts(x^2),main="Volatility Forecasting: SG-GARCH", ylab="Squared Data",xaxt=
axis(1,at=c(0,50,100,150,200,250),labels=c("Mar 2015","May 2015","July 2015",
"Oct 2015","Dec 2015","Feb 2016"))
lines(ts(sig),type="l",col="blue")
legend(0,.00023,c("SG-GARCH"),
lty=c(1),lwd=c(2.5),col=c("blue"),
title="Predicted Volatility")

plot(ts(x^2),main="Volatility Forecasting: Classical Models",
ylab="Squared Data",xaxt="n")
axis(1,at=c(0,50,100,150,200,250),labels=c("Mar 2015","May 2015","July 2015",
"Oct 2015","Dec 2015","Feb 2016"))
lines(ts(v1),type="l",col="green")
lines(ts(v2),type="l",col="red")
legend(0,.00023,c("Normal GARCH", "t-GARCH"),
lty=c(1,1),lwd=c(2.5,2.5),col=c("green","red"),
title="Predicted Volatility")
```