Implementation of GARCH models and VaR Estimation

Sayantan Paul

Supervisor
Dr. R. Krishnan

Indira Gandhi Institute of Development Research
Mumbai

May 14, 2013

Contents

1 Introduction

In this paper I have tried to make a guide which shows how to model implement GARCH models in practice. I start by showing the autocorrelation and partial autocorrelation properties of the returns series. I then demonstrate how to model for GARCH effects assuming various error distributions. I then demonstrate how to calculate the Value-at-Risk for the portfolio under consideration. SAS and R codes and Figures are given wherever relevant.

2 Data and Methodology

2.1 Data

I consider a portfolio of 10 units of 'TATA STEEL' shares. Data on the closing prices is taken for a period of four years from the 1^{st} of February, 2009 till the 31^{st} of Decenber, 2012. The data is taken from the PROWESS Database.

2.2 Methodology

2.2.1 Diagnostics

The time series plot of the portfolio returns series is shown in Figure 1. The returns seems to be mean reverting. To test for the presence of an unit root, an Augmented Dickey–Fuller (ADF) Test is carried out on the returns series. The package 'tseries' in R can be used to perform an ADF test. The following R commands are used to conduct the ADF test –

$> library(tseries)$
$> adf.test(ret)$

An ADF test on the returns series gives a $p - value = 0.01$. The null hypothesis of the ADF test is that the series is non-stationary. At a 5% level of significance, the null is rejected iff the $p - value$ is < 0.05. Thus the null hypothesis of non stationarity is rejected and the returns series can be considered to be stationary in the mean.

Figure 1 shows that the series is mean reverting but also shows a pattern of alternating quiet and volatile periods of substantial duration. Thus the returns series exhibits volatility clustering i.e. the conditional variance of the time series varies over time.

The next step is to plot the acf and pacf of the returns series. The easiest way to generate the acf and pacf in R is to use the 'forecast' package. The following R commands are used to obtain the acf and the pacf of the returns series –

$> library(forecast)$
$> Acf(ret)$
$> Pacf(ret)$

The autocorrelation function (acf) plot of the portfolio returns series is shown in Figure 2. There are significant autocorrelations at lags 8 and 9. This implies a possible presence of a MA terms in the series.

The partial autocorrelation function (pacf) of the returns series is shown in Figure 3. There are significant autocorrelations at lags 8 and 9. This implies a possible presence of an AR terms in the series.

The acf and pacf of the squared and absolute returns can be obtained from the following R commands –

$> Acf(ret^2)$
$> Pacf(ret^2)$
$> Acf(abs(ret))$
$> Pacf(abs(ret))$

The acf and pacf of squared and absolute exchange rate returns are shown in Figure 4, Figure 5, Figure 6 and Figure 7. The substantial correlations in the autocorrelation and partial autocorrelation plots of the squared and absolute returns series indicates that the returns series is not independent.

The next step is to determine the order of the ARMA model to be applied on the returns series. The 'R' package 'forecast' which has an inbuilt function 'auto.arima()' to automatically compute the best ARIMA model for a time series. The following 'R' commands

$> library(forecast)$
$> auto.arima(ret, trace = T)$

produce the output in Figure 8. The best model is ARMA (3,2) with zero mean.

To formally reject the normality of the normality of the returns series, the Jarque Bera test is conducted. The following 'R' commands are used to conduct the Jarque Bera Test –

$> library(tseries)$
$> jarque.bera.test(ret)$

A $p-value < (2.2e-16)$ implies that the null that the distribution is normal is rejected.

To test for ARCH effects using the Box-Ljung statistic, the McLeod- Li test can be used. In 'R', the package 'TSA' is used to conduct the McLeod- Li test. The 'R' commands are shown below –

$> library(TSA)$
$> McLeod.Li.test(y = ret)$

From Figure 9, the plotted $p - values$ of the McLeod-Li tests are all significant at the 5% significance level. This gives formal strong evidence for ARCH in this series.

2.2.2 Model Specification

The next step is to model the returns series accounting for the time changing variance. The GARCH family models are used for this purpose. A very widely used framework is the ARMA GARCH specification –

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{1}$$

$$e_t = \upsilon_t \sigma_t \tag{2}$$

$$\sigma_t{}^2 = \omega + \sum_{i=1}^{r} \alpha_i e_{t-i}{}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}{}^2 \tag{3}$$

where

$$\upsilon_t \sim N(0,1)$$

To account for the covariance stationarity and volatility clustering, certain restrictions are imposed on the parameters of the conditional variance equation. Since a general formula giving the restrictions on the parameters cannot be obtained, the restrictions are shown for the GARCH (1,1) case. The restriction to ensure covariance stationarity is $\alpha + \beta < 1$ and the restriction to ensure the presence of the fourth moment is $3\alpha_1{}^2 + 2\alpha_1\beta_1 + \beta_1{}^2 < 1$.

Since it is very difficult to apply the restrictions applicable to higher order GARCH models, therefore I shall use a GARCH (1,1) specification throughout. GARCH models are estimated by making an assumption about the distribution of the error term. The parameters are estimated by maximizing the log likelihood. In the simplest case, the errors are assumed to be normally distributed.

The model to be estimated is –

$$r_t = \phi_1 r_{t-1} + \phi_2 r_{t-2} + \phi_3 r_{t-3} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + e_t \tag{4}$$

$$e_t = \upsilon_t \sigma_t \tag{5}$$

$$\sigma_t{}^2 = \omega + \alpha e_{t-1}{}^2 + \beta \sigma_{t-1}{}^2 \tag{6}$$

where

$$\upsilon_t \sim N(0,1) \tag{7}$$
$$\omega > 0 \tag{8}$$
$$\alpha \geq 0 \tag{9}$$
$$\beta \geq 0 \tag{10}$$
$$\alpha + \beta < 1 \tag{11}$$
$$3{\alpha_1}^2 + 2\alpha_1\beta_1 + {\beta_1}^2 < 1 \tag{12}$$

2.2.3 GARCH modeling using R

There are quite a few packages in R which can model GARCH effects. One of the most popular is the 'fGarch' package. The following codes implement an ARMA(3,2) GARCH(1,1) model on the returns series.

$> library(fGarch)$
$> fit = garchFit(formula = \ arma(3,2) + garch(1,1), data = ret)$
$> summary(fit)$

The function 'garchFit()' fits the model. Any ARMA GARCH specification can be given in the 'formula=' option. The package assumes that the errors are normally distributed by default. The results are stored in the object 'fit'. The results of 'summary(fit)' are shown in Figure 10. However, a huge flaw in the estimation is that the estimation is undertaken without imposing the constraints on the parameters. Many times it is found out that condition (12) or even condition (11) is not satisfied. In such cases, the estimated model does not even ensure the covariance stationarity condition and is thus deeply flawed. In this particular example, the conditions are satisfied by the estimated parameters. However, it should be kept in mind that had the model been estimated with the constraints imposed, the results might have been different.

2.2.4 GARCH modeling in SAS

The same model can also be estimated in SAS. The 'AUTOREG' procedure in SAS can be used to model GARCH family models. The 'PROC AUTOREG' statement is also versatile, and allows for customizations in the conditional variance equation and allows restrictions to be imposed on the parameters. However, a major flaw in the statement is that it does not allow for an AR component in the mean equation, and only allows for autoregressive errors. Estimation using 'PROC AUTOREG' shall be demonstrated later on in this article.

Another way to model GARCH effects in SAS is by using the MODEL procedure. The MODEL procedure is very versatile. It allows complete freedom in specifying the mean and variance equations. Also, any number of restrictions can be imposed on the parameters. It seems that the only flaw in the 'MODEL' statement is that it is surprisingly difficult to output the values of the conditional variances for the successive iterations. This creates a problem while trying to calculate the VaR of a portfolio, since a forecast of the conditional variance is needed. As and when I overcome this problem, I will update this article. The following code implements an ARMA(3,2) GARCH(1,1) model on the returns series using normally distributed residuals.

```
/* Estimation of GARCH(1,1) with normally distributed residuals using PROC MODEL*/

proc model data = final ;
parms phi1 .1 phi2 .02 phi3 .03 theta1 .02 theta2 .1 arch0 .1 arch1 .2 garch1 .75;
restrict arch0 > 0;
restrict arch1 ≥ 0;
restrict garch1 ≥ 0;
restrict arch1 + garch1 < 1;
restrict 3*arch1*arch1 + 2*arch1*garch1 + garch1*garch1 < 1;

/* mean model */

r = phi1 * zlag1(r) + phi2 * zlag2(r) + phi3 * zlag3(r) + theta1 * zlag1(resid.r) + theta2 * zlag2(resid.r);

/* variance model */

h.r = arch0 + arch1*xlag(resid.r**2,mse.r) + garch1*xlag(h.r,mse.r);

/* fitting the model*/

fit r / method = marquardt fiml;
run;
quit;
```

The 'parms' option specifies the parameters. There is an option to put initial values in the 'parms' option (as is done in this particular code), otherwise the SAS system generates the initial values by itself. The 'restrict' option is used to impose restrictions on the parameter values. Any number of constraints can be imposed by the 'restrict' option (the 'bounds' option performs functions similar to the 'restrict' option, but does not handle non linear constraints, so one is better off using the 'restrict' statement). In this particular case, both the covariance stationarity and excess kurtosis conditions for the GARCH (1,1) model have been specified. One can specify both the mean equation and the variance equation, allowing for an ARMA GARCH model structure. This is an improvement over the PROC AUTOREG statement, which produces AR GARCH model estimations. The 'zlagi' operator is used to produce the i^{th} lag of the variable, with missing values replaced by zero. The xlagn(x,y) operator returns the n^{th} lag of x if x is nonmissing, or y if x is missing. The method = marquardt specifies the Marquardt-Levenberg method for iterative minimization. FIML corresponds to Full Information Maximum Likelihood Estimation. The output of the estimation is shown in Figure 11. The estimated values of the variance equation are slightly different from that produced by R, while that of the mean equation are totally different. However, since the estimation was carried out by imposing the covariance stationarity and excess kurtosis conditions on the parameters, this estimation is better.

The same model can also be estimated using t-distributed residuals. The following code implements an ARMA(3,2) GARCH(1,1) model on the returns series using normally distributed residuals.

```
/* Estimate of GARCH(1,1) with t-distributed residuals using PROC MODEL */

proc model data = final;
parms df 7.5 phi1 .1 phi2 .02 phi3 .03 theta1 .02 theta2 .1 arch0 .1 arch1 .2
garch1 .75 ;
restrict arch0 > 0;
restrict arch1 ≥ 0;
restrict garch1 ≥ 0;
restrict arch1 + garch1 < 1;
restrict 3*arch1*arch1 + 2*arch1*garch1 + garch1*garch1 < 1;

/* mean model */

r = phi1 * zlag1(r) + phi2 * zlag2(r) + phi3 * zlag3(r) + theta1 * zlag1(resid.r)
+ theta2 * zlag2(resid.r);

/* variance model */

h.r = arch0 + arch1 * xlag(resid.r **2, mse.r) + garch1*xlag(h.r, mse.r);

/* specifying error distribution */

errormodel r ~ t(h.r,df);

/* fitting the model */

fit r / method=marquardt;
run;
quit;
```

The 'errormodel' statement is used to specify the t-distributed residuals. The degrees of freedom (df)for the distribution are also estimated as a parameter in the MODEL procedure. The output is given in Figure 12. The estimated values are different from the previous case.

One can also estimate the E-GARCH model using PROC MODEL. The EGARCH(1,1) model is given by

$$ln\sigma_t{}^2 = \alpha_0 + \alpha_1 + ln\sigma_{t-1}{}^2 + \beta_1 g(\upsilon_{t-1}) \tag{13}$$

$$g(\upsilon_t) = \theta\upsilon_t + \gamma[|\upsilon_t| - E|\upsilon_t|] \tag{14}$$

The following code models an ARMA(3,2) EGARCH(1,1) MODEL –

```
/* Estimation of EGARCH(1,1) Model with PROC MODEL */

proc model data = final;
parms phi1 .1 phi2 .02 phi3 .03 theta1 .02 theta2 .1 earch0 .1 earch1 .2 egarch1
.75 theta .65;

/* mean model */

r = phi1 * zlag1(r) + phi2 * zlag2(r) + phi3 * zlag3(r) + theta1 * zlag1(resid.r)
+ theta2 * zlag2(resid.r);

/* variance model */

if (_obs_ = 1 ) then h.r = exp( earch0 + egarch1 * log(mse.r));

else h.r = exp(earch0 + earch1*zlag(g) + egarch1*log(zlag(h.r)));

g = theta*(-nresid.r) + abs(-nresid.r) - sqrt(2/constant('pi'));

/* fitting the model */

fit r / fiml method = marquardt;
run;
quit;
```

The results are shown in Figure 14

3 Calculation of Value-at-Risk

The one-step ahead forecast of the mean and volatility equations (1) and (3) are respectively –

$$\hat{r_t}(1) = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t+1-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{15}$$

$$\hat{\sigma_t}^2(1) = \omega + \sum_{i=1}^{r} \alpha_i e_{t+1-i}{}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t+1-j}{}^2 \tag{16}$$

I assume that υ_t is Gaussian. Then the conditional distribution of r_{t+1}, given the information available at time 't' is $N[\hat{r_t}(1), \hat{\sigma_t}^2(1)]$. Thus the 5% quantile for VaR calculation is $\hat{r_t}(1) - 1.65\hat{\sigma_t}(1)$

In this case, to get the predicted values, I use the 'AUTOREG' procedure of SAS. But to use the 'AUTOREG' procedure, the returns series must be demeaned. This results in a slight loss in accuracy of the model. The estimated model, along with the forecasting equations is

$$r_t = \mu + e_t \tag{17}$$

$$e_t = \upsilon_t \sigma_t \tag{18}$$

$${\sigma_t}^2 = \omega + \alpha {e_{t-1}}^2 + \beta {\sigma_t}^2 \tag{19}$$

$$\hat{\sigma_t}^2(1) = \omega + \alpha {e_t}^2 + {\sigma_t}^2 \tag{20}$$

The following SAS code is used to get the forecast values —

/ Estimation of GARCH(1,1) with normally distributed residuals using PROC AUTOREG */*

proc autoreg data = final;

/ model specification */*

model r = / garch = (q=1,p=1);
output out = vol cev = vhat rm = ehat;
run;

The out = vol option in the output statement stores the output in a dataset called vol(user specified). The cev = vhat option writes the conditional variances for each 't' in the variable vhat and stores it in the vol dataset. The rm = ehat option writes the residual values for each 't' in the variable ehat and stores it in the vol dataset. The values of vhat and ehat for the 31^{st} of December, 2012 give ${\sigma_t}^2$ and e_t. The vol dataset also contains the value of 'r' for each day(including the 31^{st} of December, 2012). Thus the 5% quantile can be calculated.

The results are shown in Figure 13. Also, the value of ${\sigma_t}^2 = 0.0002707442$, $e_t = -0.000262653$ and $hatr_t(1) = -0.000116679$ for $t = 31^{st}$ December, 2012 are obtained from the 'vol' dataset . We can calculate $\hat{\sigma_t}^2(1) = 0.000256814$ by putting the values of the estimated parameters and required variables in the equations for the one step ahead forecasts. The the 5% quantile is

$$-0.000116679 - 1.6449 * \sqrt{0.000256814} = -0.026476$$

The Rupee VaR for a long position of 10 Tata Steel shares is 10*(stock price at 31^{st} December, 2012) * .026476, which is = Rs. 113.45. Thus the worst expected loss over 1 day under normal market conditions at 5% level of confidence is Rs 113.45

4 Conclusion

The are many resources available to implement GARCH models. MATLAB is another very good software to model GARCH effects on. The web pages of Kevin Sheppard (www.kevinsheppard.com) offers an excellent set of tools for modelling GARCH in MATLAB. The MATLAB Web Site of Eric Jondeau & Michael Rockinger of the University of Lausanne (www.hec.unil.ch) offers a set of excellent free customized GARCH codes. But the problem with most codes available is that they assume demeaned returns and proceed with the solution. This approach is incorrect because the estimation should be a joint estimation of ARMA and GARCH parameters. The advantage of creating customized codes for the GARCH models is that it allows the researcher to modify the volatility equation, the assumption on the error term distribution and parameter restrictions. This knowledge will come in handy when specific questions regarding the volatility need to be answered.

5 References

1. Class Notes, Dr. R. Krishnan
2. Cryer and Chan, 'Time Series Analysis with applications in R', Second Edition, Springer Texts in Statistics
3. Bollerslev, T, 'Generalized Autoregressive Conditional Heteroskedasticity',Journal of Econometrics 31 (1986) 307-327
4. SAS documentation – www.support.sas.com

6 Appendix – Figures

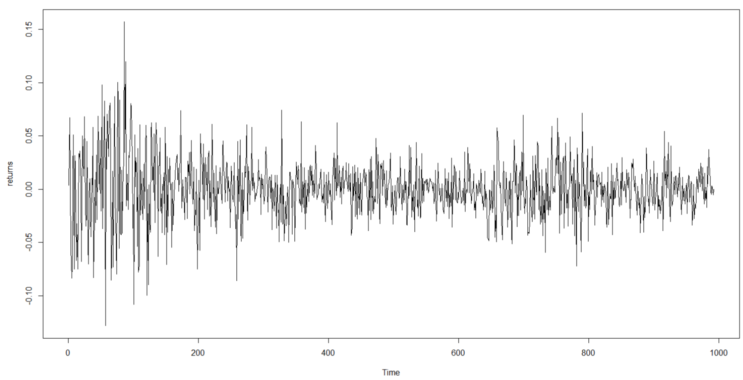

Figure 1: TS plot of portfolio returns

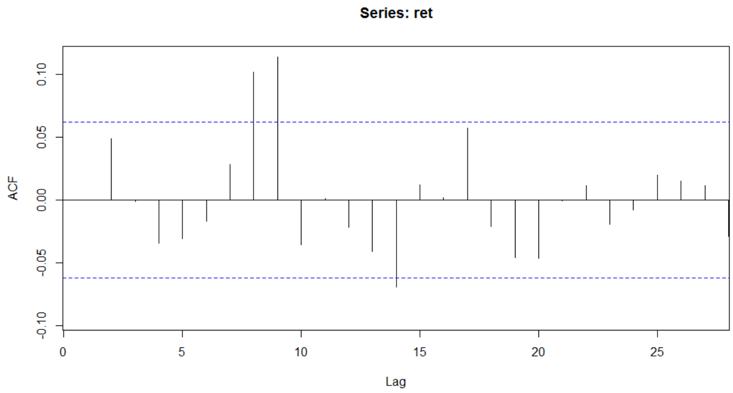

Figure 2: acf of daily portfolio returns

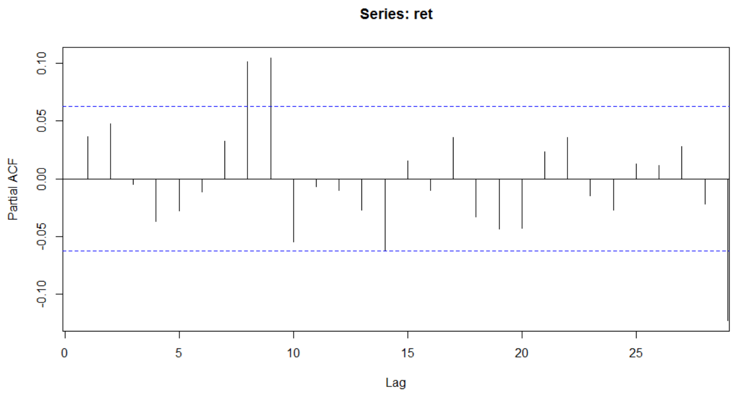

Figure 3: pacf of daily portfolio returns

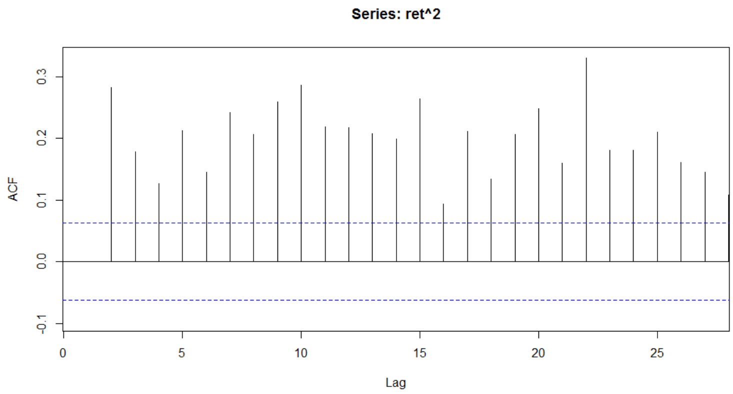

Figure 4: acf of squared returns

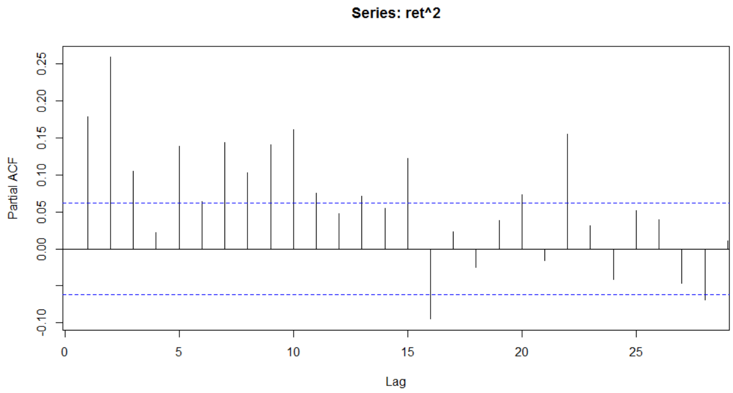

Figure 5: pacf of squared returns

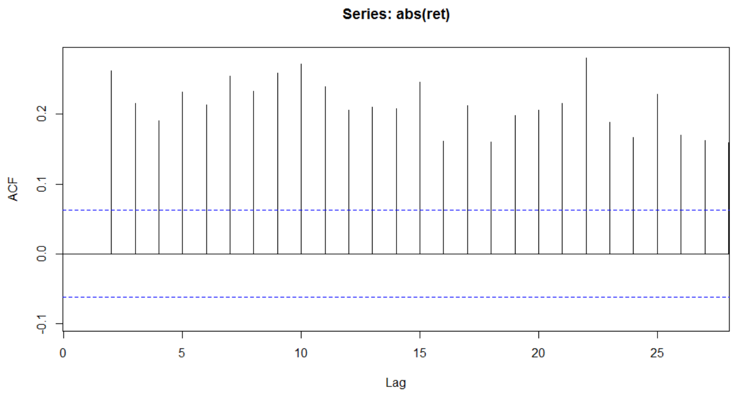

Figure 6: acf of absolute returns

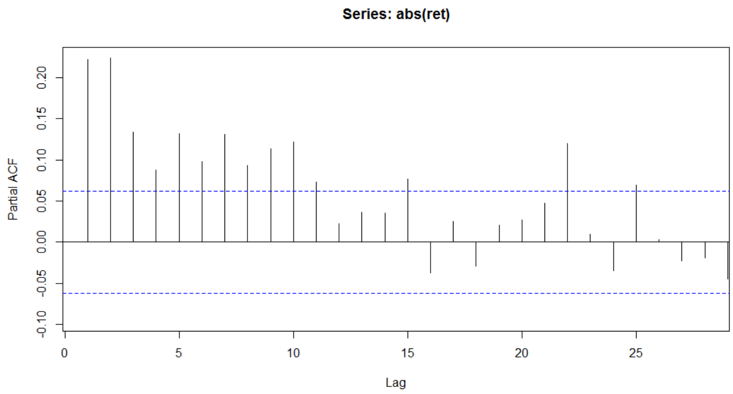

Figure 7: pacf of absolute returns

Figure 8: auto.arima results

```
> auto.arima(ret, trace = T)

 ARIMA(2,0,2) with non-zero mean : -4271.601
 ARIMA(0,0,0) with non-zero mean : -4250.804
 ARIMA(1,0,0) with non-zero mean : -4249.103
 ARIMA(0,0,1) with non-zero mean : -4249.975
 ARIMA(1,0,2) with non-zero mean : -4247.495
 ARIMA(3,0,2) with non-zero mean : -4282.283
 ARIMA(3,0,1) with non-zero mean : -4248.881
 ARIMA(3,0,3) with non-zero mean : -4266.39
 ARIMA(2,0,1) with non-zero mean : -4251.874
 ARIMA(4,0,3) with non-zero mean : -4277.176
 ARIMA(3,0,2) with zero mean     : -4283.784
 ARIMA(2,0,2) with zero mean     : -4273.231
 ARIMA(4,0,2) with zero mean     : -4279.595
 ARIMA(3,0,1) with zero mean     : -4250.584
 ARIMA(3,0,3) with zero mean     : -4268.131
 ARIMA(2,0,1) with zero mean     : -4253.57
 ARIMA(4,0,3) with zero mean     : -4278.731

 Best model: ARIMA(3,0,2) with zero mean

Series: ret
ARIMA(3,0,2) with zero mean

Coefficients:
         ar1      ar2      ar3      ma1     ma2
      1.2896  -0.8675  -0.0389  -1.2667  0.8861
s.e.  0.0600   0.0765   0.0353   0.0515  0.0529

sigma^2 estimated as 0.0007863:  log likelihood=2137.78
AIC=-4263.57   AICc=-4263.48   BIC=-4234.17
```

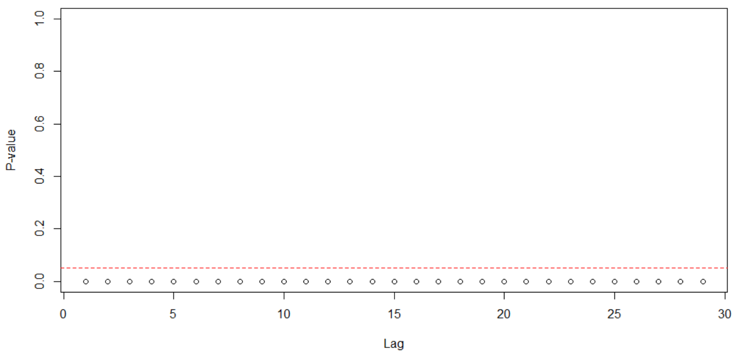

Figure 9: Mcleod Li test for portfolio returns

Figure 10: R garchFit summary

```
> summary(fit)

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(3, 2) + garch(1, 1), data = ret)

Mean and Variance Equation:
 data ~ arma(3, 2) + garch(1, 1)
<environment: 0x0000000010f04ae0>
 [data = ret]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1          ar2          ar3          ma1
 7.2417e-05   1.0000e+00  -5.6657e-01  -4.8406e-02  -9.9378e-01
        ma2        omega       alpha1        beta1
 6.1961e-01   6.0672e-06   6.0224e-02   9.2990e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      7.242e-05   4.361e-04    0.166   0.8681
ar1     1.000e+00   2.478e-01    4.035 5.46e-05 ***
ar2    -5.666e-01   1.418e-01   -3.995 6.47e-05 ***
ar3    -4.841e-02   3.399e-02   -1.424   0.1544
ma1    -9.938e-01   2.532e-01   -3.924 8.70e-05 ***
ma2     6.196e-01   1.325e-01    4.677 2.91e-06 ***
omega   6.067e-06   3.353e-06    1.809   0.0704 .
alpha1  6.022e-02   1.528e-02    3.942 8.07e-05 ***
beta1   9.299e-01   1.721e-02   54.026  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 2272.2    normalized:  2.290524

Description:
 Mon May 13 23:47:17 2013 by user: ShaUryA


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  7.80642   0.02017704
 Shapiro-Wilk Test  R    W      0.9970337 0.06269295
 Ljung-Box Test     R    Q(10)  7.391757  0.6880149
 Ljung-Box Test     R    Q(15)  11.59595  0.7093187
 Ljung-Box Test     R    Q(20)  14.38121  0.8106537
 Ljung-Box Test     R^2  Q(10)  4.165763  0.9395594
 Ljung-Box Test     R^2  Q(15)  7.875108  0.928692
 Ljung-Box Test     R^2  Q(20)  10.27467  0.9629135
 LM Arch Test       R    TR^2   6.229344  0.9040822

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-4.562904 -4.518451 -4.563066 -4.546002
```

Figure 11: SAS PROC MODEL results using normally distributed residuals

Observations Processed

Read	993
Solved	993
Used	992
Missing	1

The SAS System 21:17 Thursday, May 13, 2013 9

The MODEL Procedure

Nonlinear FIML Summary of Residual Errors

Equation	DF Model	DF Error	SSE	MSE	Root MSE	R-Square	Adj R-Sq
r	8	984	0.7860	0.000799	0.0283	0.0134	0.0064
resid.r		984	1001.2	1.0174	1.0087		

Nonlinear FIML Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
phi1	1.086418	0.3073	3.53	0.0004
phi2	-0.68036	0.2283	-2.98	0.0029
phi3	-0.04212	0.0366	-1.15	0.2504
theta1	1.07325	0.3073	3.49	0.0005
theta2	-0.71637	0.2012	-3.56	0.0004
arch0	6.272E-6	3.51E-6	1.79	0.0743
arch1	0.06106	0.0158	3.86	0.0001
garch1	0.92859	0.0181	51.44	<.0001

Number of Observations	
Used	992
Missing	1

Statistics for System	
Log Likelihood	2268

Figure 12: SAS PROC MODEL results using t-distributed residuals

Observations Processed

Read	993
Solved	993
Used	992
Missing	1

The SAS System 21:17 Thursday, May 13, 2013

The MODEL Procedure

Nonlinear Liklhood Summary of Residual Errors

Equation	DF Model	DF Error	SSE	MSE	Root MSE	R-Square	Adj R-Sq
r	9	983	0.7852	0.000799	0.0283	0.0145	0.0065
NRESID.r		983	1160.3	1.1804	1.0865		

Nonlinear Liklhood Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
df	14.20558	6.5700	2.16	0.0308
phi1	1.148481	0.2137	5.37	<.0001
phi2	-0.74158	0.1938	-3.83	0.0001
phi3	-0.03612	0.0368	-0.98	0.3261
theta1	1.136036	0.2127	5.34	<.0001
theta2	-0.77238	0.1695	-4.56	<.0001
arch0	5.356E-6	3.31E-6	1.62	0.1060
arch1	0.053473	0.0150	3.58	0.0004
garch1	0.92773	0.0197	47.06	<.0001

Number of Observations		Statistics for System	
Used	992	Log Likelihood	2271
Missing	1		

Figure 13: SAS PROC AUTOREG results

```
                    The SAS System           21:17 Thursday, May 13, 2013

                 The AUTOREG Procedure

                Dependent Variable    r


           Ordinary Least Squares Estimates

   SSE                 0.79669449    DFE                   991
   MSE                  0.0008039    Root MSE          0.02835
   SBC                 -4247.9173    AIC              -4252.817
   Regress R-Square        0.0000    Total R-Square     0.0000
   Durbin-Watson           1.9277


                                     Standard                Approx
Variable        DF     Estimate         Error    t Value    Pr > |t|

Intercept        1     0.000633      0.000900       0.70      0.4822


  Algorithm converged.


                   GARCH Estimates

    SSE                0.79692973    Observations            992
    MSE                 0.0008034    Uncond Var       0.00067594
    Log Likelihood     2264.48679    Total R-Square            .
    SBC                -4501.3747    AIC              -4520.9736
    Normality Test         7.4072    Pr > ChiSq           0.0246


                                     Standard                Approx
Variable        DF     Estimate         Error    t Value    Pr > |t|

Intercept        1     0.000146      0.000694       0.21      0.8333
ARCH0            1    6.6148E-6     3.2473E-6       2.04      0.0417
ARCH1            1       0.0661        0.0149       4.44      <.0001
GARCH1           1       0.9241        0.0159      58.00      <.0001
```

Figure 14: SAS PROC MODEL EGARCH results

```
                    Lambda              1E-7


                      Observations Processed

                          Read        993
                          Solved      993
                          Used        992
                          Missing       1
                          The SAS System           21:17 Thursday, May 13, 2013

                        The MODEL Procedure

                Nonlinear FIML Summary of Residual Errors

                DF      DF                                                        Adj
Equation     Model   Error       SSE        MSE   Root MSE   R-Square      R-Sq

r                9     983    0.7848   0.000798     0.0283     0.0150    0.0070
resid.r                983    1006.4     1.0238     1.0118


                 Nonlinear FIML Parameter Estimates

                                   Approx                   Approx
       Parameter       Estimate   Std Err    t Value     Pr > |t|

       phi1            1.201437    0.2168       5.54       <.0001
       phi2            -0.73422    0.2312      -3.18       0.0015
       phi3            -0.04039    0.0406      -0.99       0.3202
       theta1           1.18594    0.2153       5.51       <.0001
       theta2          -0.76143    0.2009      -3.79       0.0002
       earch0          -0.02558    0.0305      -0.84       0.4025
       earch1           0.10968    0.0274       4.00       <.0001
       egarch1         0.996633   0.00414     240.45       <.0001
       theta           -0.30802    0.1429      -2.15       0.0314


         Number of Observations      Statistics for System

         Used               992    Log Likelihood        2270
         Missing              1
```