# Variance models

This computer exercise concerns some common dynamic models for varying variance and an introduction to non-parametric modelling. You should solve all the regular problems to pass the computer exercise, and you need to solve the extra problems if you want to get 1 bonus credits.

It is preferred if you use Matlab, but you are allowed to use the programming language or package of your choice. If you choose not to use Matlab, please note that you are required to document your code extra carefully.

## 1 Preparations for the exercise

Read chapter 5.5.2 in [1], the handouts and this instruction. Then you should prepare the computer exercise by writing down the Matlab functions needed for the exercise.

Before the computer exercise some of the questions below will be posed. All of the posed question must be answered correctly in order to pass the computer exercise.

## 2 Catalogue of questions

You should be able to answer the following questions before the computer exercise.

1. Discuss the difference (qualitatively) between ARCH, GARCH and EGARCH processes.

2. Assume $X_t$ is generated by a GARCH(1,1) process. Show how $X_t^2$ is generated by an ARMA(1,1) process according to:

$$X_t^2 - (\alpha + \beta) X_{t-1}^2 = \omega + \nu_t - \beta \nu_{t-1} \tag{1}$$

where $\nu_t$ is a sequence of white noise.

## 3 Computer Exercises

In this exercise you will use two different tools for dynamic modeling of variance; parametric time-series models to describe time varying variance. These can easily be combined with conditional mean models (e.g. AR or STAR models).

## 3.1 Variance models

Financial data often show heteroscedastic properties and hence we can not apply traditional linear or non-linear models. Heteroscedasity admittedly leads to consistent estimates, but not effective estimates. To add insult to injury, the confidence intervals are often incorrect.

Since we work with dynamic models the focus of this exercise will be dynamic heteroscedastic models. You should estimate the parameters using Maximum Likelihood.

**Hint:** When writing down likelihood functions it is often an advantage for the understanding (but not computationally) to use `for` loops where you first calculate the residual $\varepsilon_t$, then calculating $\sigma_{t+1}$ then $\varepsilon_{t+1}$ and so on. If this is to slow, feel free to optimize the code.

Note that you do *not* have to assume normally distributed innovations to be able to estimate a process using ML. Feel free to try some other suitable distribution, e.g. student-$t$.

### 3.1.1 GARCH-processes

Estimate a GARCH(1,1) process with conditionally gaussian innovations from a given simulated data set with parameters $\omega = 0.1$, $\alpha = 0.25$, $\beta = 0.6$ and $\mu = 8$. Note that the first three parameters must be positive to guarantee a positive variance. Load the data set by writing:

```
>> load garchdata.dat
```

**Assignment:** Estimate the parameters and construct the 95 % confidence interval using ML.

**Assignment:** From equation (1) we know that the GARCH process can be formulated as an ARMA process. Transform the process and estimate an ARMA process by using Matlab's System Identification Toolbox routine `armax`. Compare and discuss the results.

### 3.1.2 EGARCH-processes

Sometimes we want to model the fact external shocks affect the variance asymmetrically. A possible model is then exponential GARCH, or EGARCH. It also has the additional advantage of not putting any restrictions on the parameter values. You will again estimate the parameters from a given simulated data set. The model is given by:

$$\log \sigma_t^2 = \omega + \beta \log \sigma_{t-1}^2 + \alpha\left(|\varepsilon_{t-1}| - c\varepsilon_{t-1}\right)$$

where the constants in the data set are $\omega = 0.1$, $\alpha = 0.25$, $\beta = 0.6$, $c = 0$ and $\mu = 8$. Load the data by writing:

```
>> load egarchdata.dat
```

**Assignment:** Estimate the parameters and construct a 95 % confidence interval using ML. You don't have to estimate the $c$ parameter.

## 3.2 Extra: Multivariate GARCH models and market data

Fit a bivariate CCC-MVGARCH model to Swedish stocks.

**Assignment:** Start by simulating data from a CCC-MVGARCH, using the parameters you estimated for the univariate GARCH model for the stocks, and $\rho = 0.45$ as correlation. Fit the parameters by

- Estimating the individual standardized returns.
- Estimating the standardized correlation.

and then optimizing the joint (log-)likelihood. Feel free to validate your estimates with the free `MFE Toolbox`[1] and/or with the parameters you used.

What happens if you increase the sample size of the data - does the estimates converge to the true parameters?

**Assignment:** Next, download the data in `SwedishStockData.mat`. You will find eight different stocks from 20051011 to 20101008.

```
>> load SwedishStockData.mat
```

Choose two stocks (e.g. Lundin and MTG) and fit a CCC-MVGARCH to the data.

**Hint:** You can download more recent stock data from sources like Yahoo Finance[2] .

[1] https://www.kevinsheppard.com/code/matlab/mfe-toolbox/
[2] http://finance.yahoo.com

# 4 Feedback

Comments and ideas relating to the computer exercise are always welcome. Send them to Magnus wiktorsson, `magnus.wiktorsson@matstat.lu.se`

# 5 MATLAB–routines

| | |
|---|---|
| `fminsearch` | Simplex based optimisation routine. |
| `fminunc` | Numerical minimization of a multidimensional function. The routine is based on quasi-Newton (BFGS) and it can return the minising parameter value, the minimal function value and the Hessian. The cryptic name comes from *function minimization unconditional*. |
| `MLmax` | Customized Quasi-Newton based optimization algorithm for maximum likelihood estimation. Maximises the likelihood-function by using the score function's quadratic variations to estimate Fisher's Information matrix. Needs the log-likelihood returned as a vector. `>> [xout,logL,CovM]=MLmax(@lnL,x0,indata)` |

# References


[1] Lindström, E., Madsen, H., Nielsen, J. N., (2015) *Statistics for Finance*, Chapman and Hall/CRC, ISBN 9781482228991.

[2] Silvennoinen, A. and Teräsvirta, T. (2009) *Multivariate GARCH models*, In *Handbook of Financial Time Series*, pp 201–229