Project 2

The file *SP500.mat* contains monthly returns of the stock market index Standard & Poor's 500 [1] from 1926 to 1991 in the form of the time series

$$X = (X_t, t = 1, \ldots, 792).$$

Your task is to find a reasonable model for this data by following the steps below. You may use whatever MATLAB functions you can find, but as usual you need to make a detailed description of your implementation along with a discussion of the result. For this project you should make sure you are familiar with the contents of Chapter 5 (especially Sections 5.3–5.5) and Sections 7.1–7.2 of Chapter 7 in [BD]. The documentation for the Econometric Toolbox [2] in MATLAB is also vital to read. You are encouraged to use the functions of this toolbox, and we list a few useful ones below. Make sure you read the documentation for each of them.

1. (3 points) Mean correct X and plot the (sample) autocorrelation and partial autocorrelation functions for h up to 20. Also do a Ljung–Box test with $h = 20$. Discuss the results. Do you think the returns can/should be modeled as an $\mathrm{ARMA}(p, q)$ process?

2. (8 points) For all values of p and q in the range $1, 2, \ldots, 10$, fit an $\mathrm{ARMA}(p, q)$ process (driven by Gaussian noise) to the data and compute the BIC (Bayesian information criteria). Which model (i.e., choices of p and q) minimizes the BIC? Remember that you assume that the mean of the data set is zero so you should not estimate a constant term.

3. (4 points) For the model that minimized the BIC (called the optimal model from now on), perform a diagnostic check, i.e., compute the residuals and analyze these: What should the distribution and covariance structure of the residuals be if the model is correct? What does the autocorrelation plot of the residuals look like? What does the autocorrelation plot of the squared residuals look like? What about a qq-plot of the residuals? Are there problems?

---

[1] Source: https://datamarket.com/
[2] See https://se.mathworks.com/help/econ/

4. (8 points) Hopefully you see from the previous task that the residuals of the optimal model can potentially be modeled as a GARCH process (why?). For all values of p' and q' in the range $1, 2, \ldots, 10$, fit a $\mathrm{GARCH}(p', q')$ process (driven by Gaussian noise) to the *residuals* of the optimal model from Task 2 and 3. Which choices of p' and q' minimize the BIC? Do a diagnostic check by computing the residuals of this fit (i.e., the residuals of the residual modeling) and analyze these in the same way as in the previous task.

5. (2 points) Fit a $\mathrm{GARCH}(p', q')$ process (driven by t-distributed noise this time) to the residuals of the optimal model from Task 2 and 3 using the optimal values for p' and q' from the previous task. How do the diagnostics change?

6. (5 points) The final task of this project is to directly fit an $\mathrm{ARMA}(p, q)$ model driven by a t-distributed $\mathrm{GARCH}(p', q')$ noise to X (mean corrected as before). Use the values of p and q that you found in Task 2 and the values of p' and q' that you found in Task 4. Do a diagnostic check as before. Are there any reasons for concern?

If you follow the tasks above, you have successfully derived a potential model for the returns X. However, in order for us to know if this model is reliable, it should be estimated on a training data set and tested on a test data set. You are very welcome to suggest alternative models for the returns, and if you can find one that you can demonstrate performs better than the one you end up with by following the steps above, you will get high points! You should then still describe the alternative model and why you think it is better than the one we end up with by following the steps above.

Some useful MATLAB functions in no particular order:

- *arima* - Used to specify an ARMA model. Note that you should set the differencing degree D to 0.

- *garch* - Used to specify a GARCH model.

- *estimate* - Used for the model fitting of both ARMA and GARCH.

- *aicbic* - Calculate the AIC and BIC of a given fit.

- *infer* - Computes the residuals for a given fit. Note that they are not standardized by default.

- *makedist* - Used to specify a probability distribution object. For a t-distribution with x degrees of freedom, write `tdist =makedist(' tLocationScale ',' nu',x);`.

- *qqplot* - Creates a qqplot. Note that you can specify the target distribution yourself.

- *min* - Find the minimum value of a vector.

- *ind2sub* - Translates scalar indices for matrices to ordinary indices.

- *autocorr* - Computes the sample ACF.

- *parcorr* - Computes the sample PACF.

- *try* and *catch* - Useful for catching errors thrown by estimate in loops.

Deadline: May 7, 2018.

Requirement: You must do this project in MATLAB. For this project there are 30 points available. To qualify for bonus points you need to score at least 10 points. After that, every 2.5 points you score on this project will translate into 0.5 bonus points on the first exam.

Formalities: You are strongly encouraged to work in pairs. Send your project report as one pdf document to both

annika.lang@chalmers.se and *annika.lang.chalmers@analys.urkund.se*.

Also send your code as one or several m-files to *andreas.petersson@chalmers.se*. Your report should include all plots, explanations, and answers to the questions as well as your implemented code in an appendix. You should try to write your report in LaTeX.