1 Models

In this section we will describe the models that have been implemented throughout the process and highlight the changed and simplifications. To make a framework for model development we created a metaclass, so called BaseModel, where we created all the common functions that all these models share. We manage to implement a strategy to create bounds with existing parameter's transformation. Not only give us a unique opportunity to modeling with various parameter transformations, but in optimizations brought as a more stable results. The optimizer that we use the Python's scipy.optimize.minimize function with the setting of the L-BFGS-B method.

1.1 MIDAS

The first model was the MIDAS that was first introduced [**?**], where they compare distributed lag models with MIDAS regression. The MIDAS model can be described with the expressions used in [**?**]. Suppose y_t is the low-frequency dependent variable that can be observed once within a time-step t (say, monthly), then $x_t^{(m)}$ the high-frequency explanatory variable can be observed m times during one time-step (say, daily or m = 22). We want to describe the relationship between y_t and $x_t^{(m)}$, in the sense of using lagged observations of $x_t^{(m)}$. The model is the following:

$$y_t = \beta_0 + \beta_1 B(L^{\frac{1}{m}}, \theta) x_t^{(m)} + \epsilon_t^{(m)} \tag{1}$$

where $B(L^{\frac{1}{m}}, \theta) = \sum_{k=0}^{K} B(k, \theta) L^{\frac{k}{m}}$, where $L^{\frac{k}{m}}$ is a lag operator such that $L^{\frac{1}{m}} x_t^{(m)} = x_{t-\frac{1}{m}}^{(m)}$. The lag coefficients in $B(k, \theta)$ of the corresponding lag operator $L^{\frac{k}{m}}$ are parameterized as a function of a small-dimensional vector of parameters Θ. β_1 is a scale parameter for the lag coefficients

1.1.1 Specification of Weighting Function

In the MIDAS literature there is one weighting function that used the most, namely "Beta" Lag. [**???**]. For completeness, I mention the others, these are the Exponential Weighting and the Exponential Almon Lag. Beta Lag involves two parameters, $\Theta = (\theta_1, \theta_2)$, and the parametrization:

$$B(k, \theta_1, \theta_2) = \frac{f(\frac{k}{K}), \theta_1, \theta_2)}{\sum_{k=1}^{K} f(\frac{k}{K}), \theta_1, \theta_2)} \tag{2}$$

where

$$f(x, a, b) = \frac{x^{a-1}(1-x)^{b-1}\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \tag{3}$$

$$\Gamma(a) = \int_0^{\infty} e^{-x} x^{a-1} dx \tag{4}$$

The following figure will deonstrate how flexiable it is correspond to different parameters:

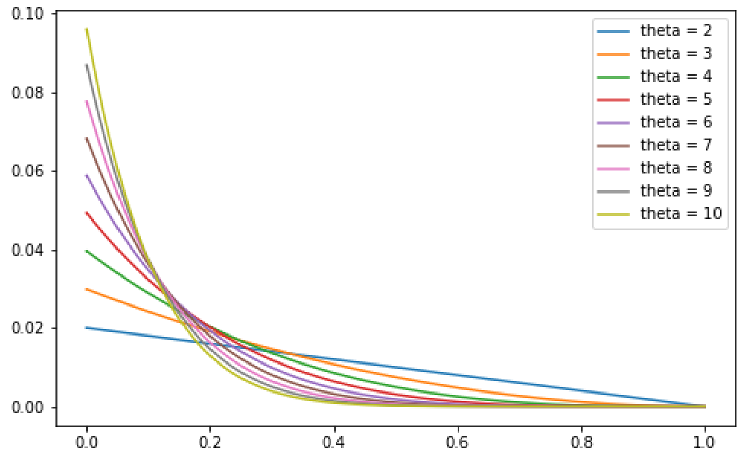

Figure 1: Plot of Beta Lag weighting function in equation 1 with K = 100, $\theta_1 = 1$ and $\theta_2 = 2, ..., 10$

We can see that if we choose to fix $\theta_1 = 1$ and in the case of $\theta_2 > 1$ cause a monoton decliyin weighting structure. This weight function specification provide us positive coefficients, which is crutual when we want to modeling volatility.

1.1.2 Parameter Transformation

In this section we will describe an approach to make parameter estmation more consistant and stabil, it is so called parameter transformation. The main idea behind this strategy is that estimators can treat bounds, but in practice it is much more convenient to transform our parameters. With this approach we can create bounds without explicitly programming to the estimator function. First we describe the transform and the back-transform function, then show how they incoperate to the function that will be estimated. Let denote θ with the parameter that we want to work with:

$$\tilde{\theta} = \begin{cases} \log(\theta) & \text{,if 'pos'} \\ \log(\theta) - \log(1-\theta) & \text{,if '01'} \\ \theta & \text{otherwise.} \end{cases}$$

$$\theta = \begin{cases} \exp(\tilde{\theta}) & \text{,if 'pos'} \\ \frac{1}{1+\exp(-\tilde{\theta})} & \text{,if '01'} \\ \tilde{\theta} & \text{otherwise.} \end{cases}$$

In the log likelihood function instead of calculting with the actual θ, we will make the estimation with θ. Than we transform back as the estimation finished. One issue raise from this estimation strategy, is that the standard error won't be correct. So there is another function called gradient. θ^* marked as the estimated parameters that were previously transformed.

$$gradient = \begin{cases} \exp(\theta^*) & \text{,if 'pos'} \\ \frac{\exp(\theta^*)}{(1+\exp(\theta^*))^2)} & \text{,if '01'} \\ 1 & \text{otherwise.} \end{cases}$$

As the L-BFGS-B method relies ont he approximation to the Hessian matrix of the loss function, se as we take advantage of information matrix equality we can calculate the standard errors easily.

1.1.3 Parameter Estimation

In the parameter estimation we will use the Python's function from scipy.optimize library, called minimize. I applied L-BFGS-B method, this method allow us to define bounds for parameters, and the biggest advantage is that approximate the inverse Hessian matrix. The estimation is happening throughout the sum of squared estimat of error:

$$SSE = \epsilon^T \epsilon = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 B(L^{\frac{1}{m}}, \theta) x_t^{(m)})^2 \tag{5}$$

$$\arg \min_{\beta_0, \beta_1, \theta_2} SSE$$

1.1.4 Simulations

The Monte-Carlo simulation was from [**?**]. Suppose we have X_t is an AR(1) process, that:

$$X_{i,t} = \phi X_{i-1,t} + \epsilon_t$$

where t = 1, ..., T show the low-frequency time-steps, i = 1, ...,I_t is the high-frequency. Set the I_t equals to 22, $\phi = 0.9$ and $\epsilon_t \sim \mathcal{N}(0,1)$ standard normal variable. The MIDAS equation will be:

$$y_t = \beta_0 + \beta_1 \sum_{k=0}^{K} \xi_k(1.0, \theta) X_{i-k,t} + z_t$$

with the parameters $\beta_0 = 0.1, \beta_1 = 0.3, \theta = 4.0$ and $z_t \sim \mathcal{N}(0, 0.5)$. We made simulations with T = 100, 200, 500. A A simulare approach was described in [**?**] with the difference of simulating quarterly/monthly data and they found out, as they increase the sample ssize the more accurate their parameter estimations will be, furthermore the more parsimonious will be the model's computational cost.

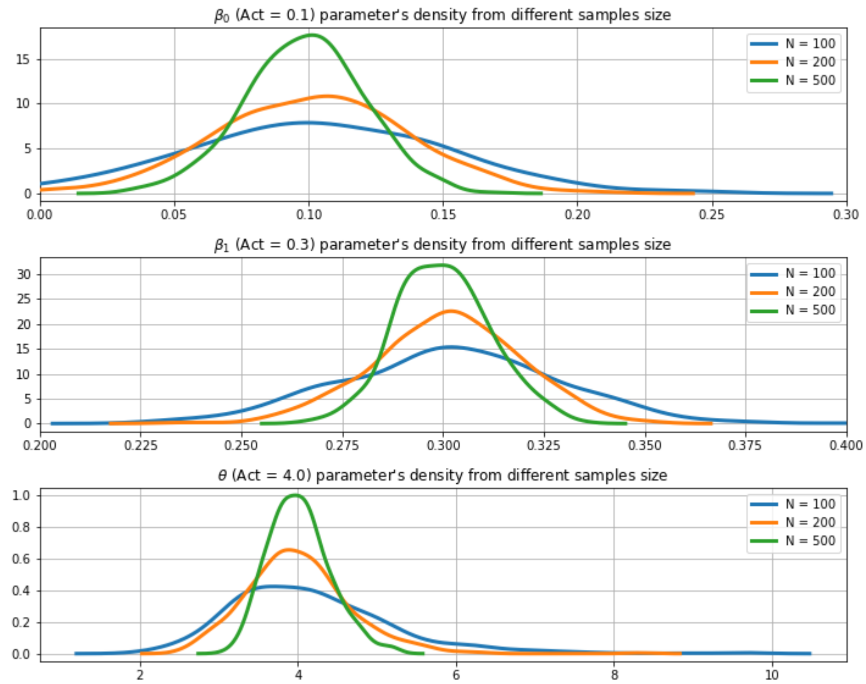

Figure 2: Plot of estimated parameter distributions with sample sizes of 100, 200, 500

1.2 Generalized Autoregressive Conditional Heteroscedasticity

In this section we would like to specify the vanilla GARCH(1, 1) model. First we assign the r_t to the daily log return ($r_t = \log P_t - \log P_{t-1}$, where P_t is the stock price at time t) for t = 1, ..., T. Assume, that the conditional mean of the returns are contants:

$$r_t = \mu + \epsilon_t \tag{6}$$

where ϵ_t denote a real-valued discrete-time stochastic process and $\mathcal{F}_t$ the information set of all information through time t. [**?**]

$$\epsilon_t \mid \mathcal{F}_{t-1} \sim \mathcal{N}(0, \sigma_t^2) \tag{7}$$

Than the GARCH(1, 1) process

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{8}$$

where $\alpha_0 > 0, \alpha_1 \geq 0, \beta_1 \geq 0$ and $1 > \alpha_1 + \beta_1$ enough wide-sense stacionarity.

1.2.1 Parameter Estimation

The estimation happens through maximum likelihood estimation. Let $\theta \in \Theta$, where $\theta = (\mu, \alpha_0, \alpha_1, \beta_1)$ and Θ is a compact subspace of an Euclidean space such that ϵ_t process finite second moments. The loglikelihood function for a sample of N observation is:

$$l_t(\Theta) = -\frac{1}{2} log 2\pi - \frac{1}{2} \log \sigma_t^2 - \frac{1}{2} \frac{\epsilon_t^2}{\sigma_t^2} \tag{9}$$

$$L_N(\Theta) = \frac{1}{N}\sum_{t=1}^{N} l_t(\Theta)$$

$$\arg\min_{\Theta} -L_N(\Theta)$$

1.2.2 Simulations

We preform Monte-Carlo simulations for GARCH(1, 1) model with different sample sizes. The generation happens through

$$\epsilon_t \sim \mathcal{N}(0, \sigma_t^2) \tag{10}$$

For t = 1, ..., T. The equation means, that we generate an ϵ_t every step with the current state of σ_t^2. We expect that as we increase the sample size, the more accurate estimation we get. The following figure will show the results of the simulation, where we apply the kernel density estimation process in order to get a smoother plot:

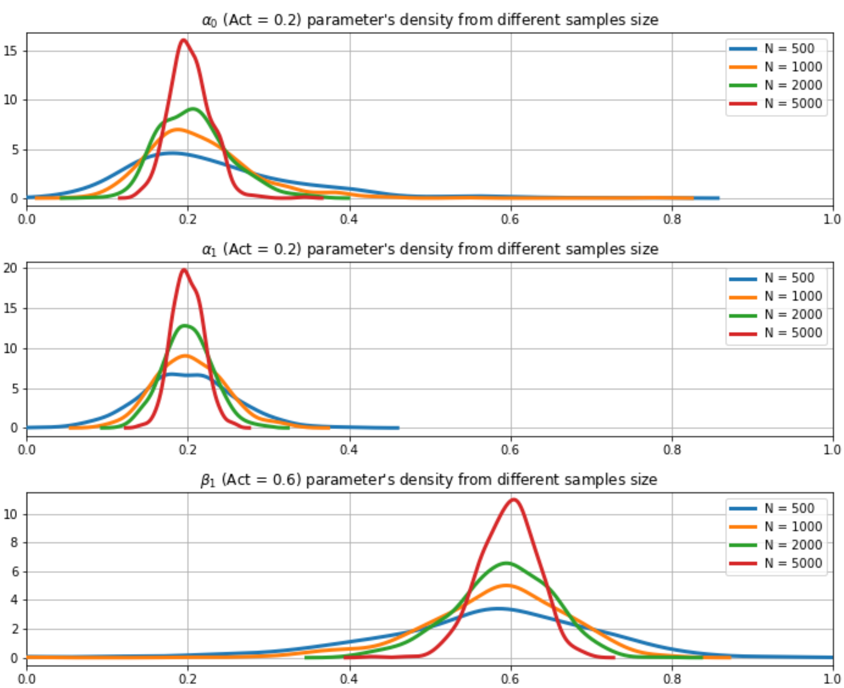

Figure 3: Plot of estimated parameter distributions with sample sizes of 500, 1000, 2000

As we expected, the parameter estimation get better and better, when we increase the sample size.

1.3 GARCH-MIDAS

This GARCH-MIDAS framework give us the opportunity to incorporate macroeconomic variables sampled at different frequency. In the recent years many study examined the effectiveness of this approach. In this model class we can utilies both models strengths. Let r_t the daily log return for t = 1, ..., T period (say monthly) and i = 1, ..., I_t (say, daily or $I_t = 22$) to the days within that period. Assume the daily log return on i in t follows:

$$r_{i,t} = \mu + \epsilon_{i,t} \tag{11}$$

$$\epsilon_{i,t} = \sqrt{g_{i,t}\tau_t} Z_{i,t} \tag{12}$$

where $\epsilon_{i,t} \mid \mathcal{F}_{i-1,t} \sim \mathcal{N}(0, g_{i,t}\tau_t)$ with $\mathcal{F}_{i-1,t}$ is the information set up to day i-1 of period t. We assuem the short-term volatility component $g_{i,t}$ is a GARCH(1, 1) process, as **?** described:

$$g_{i,t} = (1 - \alpha - \beta) + \alpha \frac{\epsilon^2_{i-1,t}}{\tau_t} + \beta g_{i-1,t} \tag{13}$$

and define the long-term volatility component τ_t in the spirit of MIDAS regression:

$$\tau_t = \beta_0 + \sum_{j=1}^{M} \beta_j \sum_{k=0}^{K} \phi_k(1.0, \theta_j) X^{(j)}_{t-k} \tag{14}$$

where $X^{(j)}_{t-k}$ referes to the j-th macroeconomic variable. τ_t is constant over the periods. As macroeconomic variables can be positive or negative values, we change the specification of tau_t as [**?** and **?**] previously did it in the following way:

$$\log \tau_t = \beta_0 + \sum_{j=1}^{N} \beta_j \sum_{k=0}^{K} \phi_k(1.0, \theta_j) X^{(j)}_{t-k} \tag{15}$$

1.3.1 Parameter Estimation

GARCH-MIDAS estimation made by maximum likelihood estimation, where the parameters for the two volatility component estimated by one-step. For our implementation and backtest raise some identification issue. The Loglikelihood function is:

$$\log L_N(\Theta) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{I_t} \sum_{i=1}^{I_t} \frac{1}{2} \log 2\pi + \frac{1}{2} \log g_{i,t}\tau_t + \frac{1}{2} \frac{\epsilon^2_{i,t}}{g_{i,t}\tau_t} \tag{16}$$

$$\arg\min_{\Theta} \log L_N(\Theta)$$

1.3.2 Simulations

IDE JÖNNEK MÉG EREDMÉNYEK.

1.4 Panel GARCH

In this section we specify th panel version of the GARCH(1, 1) model. We have daily returns $r_{i,t}$ for i = 1, ..., N and t = 1, ..., T, where i assigned the i-th stock's return and t is the t-th time step. We assume that the parameters which drive the dynamics of the return volatilites are common to all stocks. However the unconditional means of the volatilities are asset specific:

$$r_{i,t} = \sigma_{i,t}\epsilon_{i,t} \tag{17}$$

$$\sigma_{i,t}^2 = \mu_i(1-\alpha-\beta) + \alpha\epsilon_{i,t-1}^2 + \beta\sigma_{i,t-1}^2 \tag{18}$$

This means we have N + 2 numbers of parameters. This can be challenging to estimate as the number of assets increases. To tackle this issue we use the following estimation procedure.

1.4.1 Parameter estimation

First we calculate μ_i by moment matching. As μ_i is the unconditional variance of the returns we can estimate the μ_i parameters by averaging the squared returns.

$$\hat{\mu_t} = \frac{1}{N}\sum_{i=1}^{N} r_{i,t}^2 \tag{19}$$

In the second step given the unconditional variance estimates we can estimate the remaining two parameters by maximum likelihood:

$$\log L(\alpha, \beta \mid \mu_i) = \sum_{i=1}^{N}\sum_{t=1}^{T} -\frac{1}{2}\log 2\pi - \frac{1}{2}\sigma_{i,t}^2 - -\frac{1}{2}\frac{r_{i,t}}{\sigma_{i,t}^2} \tag{20}$$

where $\sigma_{i,t}^2$ is the function of the α, β.

1.4.2 Simulations

We applied the same simulation technic, that we described in the GARCH section. The aim of the simulation was still the same, but now we examined the impact of the increment in the size of N. The results can be seen in the following figures:

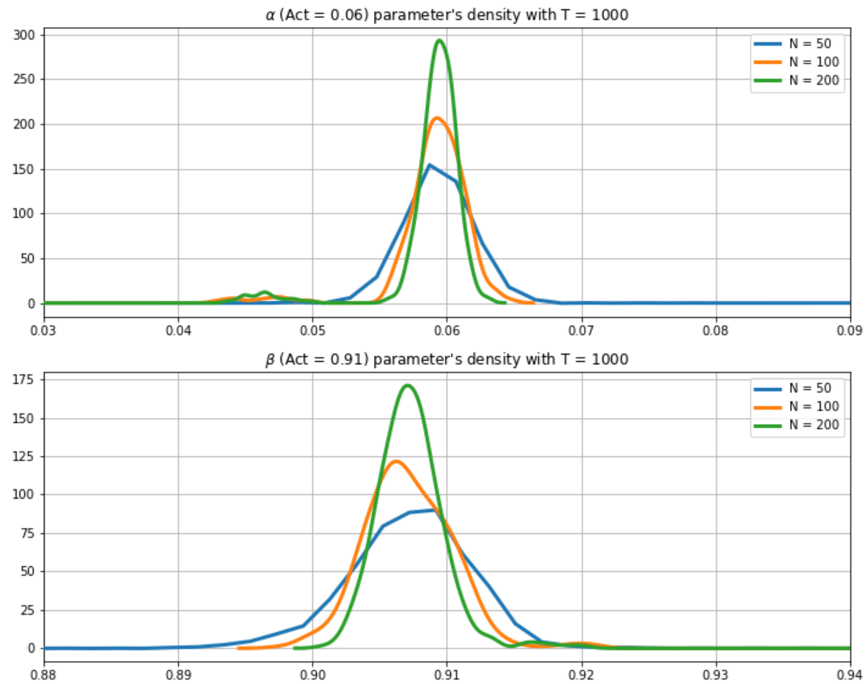

Figure 4: Plot of estimated parameter distributions with sample sizes of 1000 and N = 500, 1000, 2000

1.5 Panel GARCH with cross sectional adjustment

In this section we want to specify what we changed in contrast to the Panel GARCH section. We have:

$$r_{i,t} = \sigma_{i,t} c_t \epsilon_{i,t} \tag{21}$$

$$c_t = (1-\phi) + \phi \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_{it-1}}{\sigma_{it-1}c_{t-1}} - \frac{1}{N}\sum_{i=1}^{N}\frac{r_{it-1}}{\sigma_{it-1}c_{t-1}}\right)^2} \tag{22}$$

$$\sigma_{i,t}^2 = \mu_i(1-\alpha-\beta) + \alpha\epsilon_{i,t-1}^2 + \beta\sigma_{i,t-1}^2 \tag{23}$$

1.5.1 Parameter estimation

Estimation is done the same as the panel GARCH case. First we do the unconditional means by matching the second moment. We do the MLE on α, β and ϕ.

1.5.2 Simulations

The simulations remain the same, so the random generation happens through ϵ_t. We examined the cases of increase the sample sizes and the number of stocks. The results are the following:

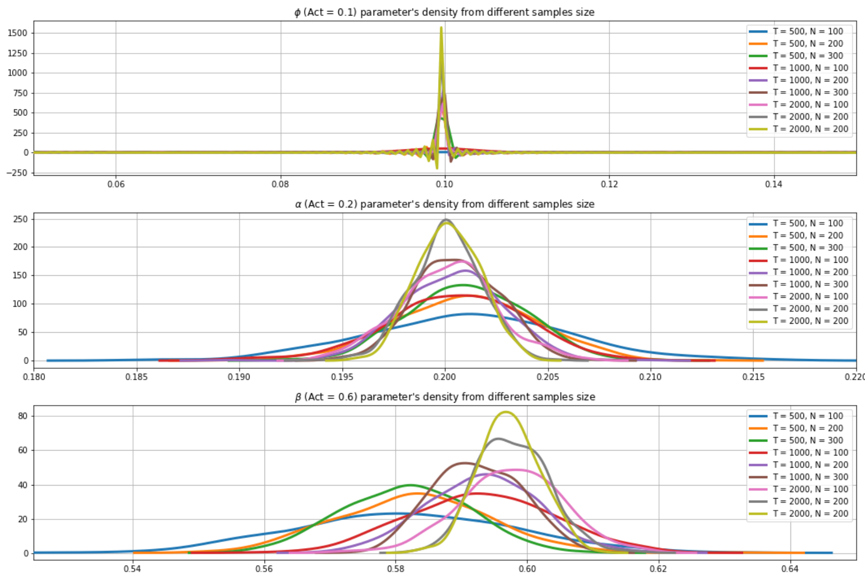

Figure 5: Plot of estimated parameter distributions with sample sizes of 1000 and N = 500, 1000, 2000

We can see that for the first to parameters (ϕ and α) the distributions of the parameter estimation's median is equal to the given parameters. In the case of β as we increase the sample size and number of stocks, we get closer and closer to the theoretical parameter.

1.6 Panel MIDAS

Let $r_{i,t}^{(j)}$ is the j-th stock's log return for j = 1, ..., M, where M is the number of stocks. We sign month index by t = 1, ..., T, the length of sequence is T and days in the t-th month sign with i = 1, ..., I_t. We assume that these stocks share the same underlying long-term volatility component, that will be marked as τ_t. τ_t is the t-th month's level of volatility whick can be explained with macroeconomic variables, so the equation:

$$\tau_t = \beta_0 + \sum_{m=1}^{N} \beta_m \sum_{k=1}^{K_m} \phi_k(1.0, \theta_m) X_{t-k}^{(m)} \tag{24}$$

where N is the number of macroeconomic variables, K is the number of lag. For simplicity we choose $K \equiv K_m$. 1 + N * 2 parameters will be estimated.

1.6.1 Parameter Estimation

The parameter estimation happens through maximum likelihood estimation as we did previously, where the loglikelihood function will connent the log returns and the long-term volatility component:

$$l_t^{(j)}(\Theta) = \frac{1}{2} \log 2\pi + \frac{1}{2}\tau_t + \frac{1}{2} \frac{(r_{i,t}^{(j)})^2}{\tau_t} \tag{25}$$

We sum them up:

$$\log l(\Theta) = \sum_{j=1}^{M} \sum_{t=1}^{T} -\frac{1}{2} \log 2\pi + \frac{1}{2}\tau_t + \frac{1}{2} \frac{(r_{i,t}^{(j)})^2}{\tau_t} \tag{26}$$

then minimize the value:

$$\arg\min_{\Theta} \log l(\Theta) \tag{27}$$