Econometrics for QRM
Exercise CAPM-t

Elvan Toygarlar (etr430) & Pim Burgers (pbs218)

September 15, 2017

Outline of Exercise

For this exercise we look again at the CAPM data which we imported from ....
This time we use only one stock: IBM.
We use the following formula for CAPM:

$$r_{i,e}^t = \beta_0 + \beta_1 r_{m,t}^e + \varepsilon_t$$

Normal distribution

We know that the following holds:

$$\varepsilon_t \sim N(0, \sigma^2)$$

This means that we also know:

$$E(r_{i,e}^t) = \beta_0 + \beta_1 r_{m,t}^e$$

and

$$\text{Var}(r_{i,e}^t) = \text{Var}(\varepsilon_t) = \sigma^2$$

This leads to:

$$r_{i,e}^t \sim N(\beta_0 + \beta_1 r_{m,t}^e, \sigma^2)$$

Now we can compute the density function:

$$f(r_{i,e}^t|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{(r_{i,e}^t - \beta_0 - \beta_1 r_{m,t}^e)^2}{2\sigma^2}\right)}$$

With:

$$\theta = \begin{pmatrix} \sigma \\ \beta_0 \\ \beta_1 \end{pmatrix}$$

We compute now the log likelihood:

$$l_n = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(r_{i,e,j}^t - \beta_0 - \beta_1 r_{m,e,j}^t)^2$$

We take the first derivative from the loglikelihood with respect to θ and set it equal to zero.

$$\frac{\partial l_n}{\partial \theta} = 0$$

This leads to:

$$\hat{\theta} = \begin{pmatrix} \hat{\sigma} \\ \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}$$

When computing this in Python the results for the normal distribution are:

$$\hat{\theta_N} = \begin{pmatrix} 1.3055 \\ -0.0043 \\ 0.8482 \end{pmatrix}$$

After this we compute the Hessian of the Normal distribution (empirical form), which should be a $3x3$ matrix.

$$H = \begin{pmatrix} \frac{\partial^2 l_n}{\partial^2 \sigma^2} & \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_0} & \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_1} \\ \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_0} & \frac{\partial^2 l_n}{\partial^2 \beta_0} & \frac{\partial^2 l_n}{\partial \beta_0 \partial \beta_1} \\ \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_1} & \frac{\partial^2 l_n}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 l_n}{\partial^2 \beta_1} \end{pmatrix}$$

We compute this in python and changed it into:

$$\Sigma(\theta) = -H^{-1}(\theta)$$

and get:

$$\Sigma(\theta) = \begin{pmatrix} 3.64e^{-06} & 1.46e^{-05} & 2.05e^{-06} \\ 1.46e^{-05} & 1.14e^{-04} & 7.97e^{-06} \\ 2.05e^{-06} & 7.97e^{-06} & 3.81e^{-05} \end{pmatrix}$$

After this we calculate the robust sandwich form:

$$\Sigma(\theta)' = H^{-1} J H^{-1}$$

$$\Sigma(\theta_n)' = \begin{pmatrix} 0.00012 & 0.00052 & -0.00065 \\ 0.00052 & 0.00294 & -0.00250 \\ -0.00065 & -0.00250 & 0.00559 \end{pmatrix}$$

After the student t distribution we explain why there is a difference between the robust sandwich form and the empirical form.

T distribution

We do the same for the t distribution and get the following results:

$$\hat{\theta}_t = \begin{pmatrix} 0.4188 = \sigma^2 \\ 0.00130 = \beta_0 \\ 0.8465 = \beta_1 \\ 2.240 = \nu \end{pmatrix}$$

The Hessian is different for the t distribution than for the normal distribution:

$$H = \begin{pmatrix} \frac{\partial^2 l_n}{\partial^2 \sigma^2} & \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_0} & \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_1} & \frac{\partial^2 l_n}{\partial \sigma^2 \partial \nu} \\ \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_0} & \frac{\partial^2 l_n}{\partial^2 \beta_0} & \frac{\partial^2 l_n}{\partial \beta_0 \partial \beta_1} & \frac{\partial^2 l_n}{\partial \beta_0 \partial \nu} \\ \frac{\partial^2 l_n}{\partial \sigma^2 \partial \beta_1} & \frac{\partial^2 l_n}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 l_n}{\partial^2 \beta_1} & \frac{\partial^2 l_n}{\partial \beta_1 \partial \nu} \\ \frac{\partial^2 l_n}{\partial \sigma^2 \partial \nu} & \frac{\partial^2 l_n}{\partial \beta_0 \partial \nu} & \frac{\partial^2 l_n}{\partial \beta_1 \partial \nu} & \frac{\partial^2 l_n}{\partial^2 \nu} \end{pmatrix}$$

This leads to:

$$\Sigma(\theta_t) = \begin{pmatrix} 0.00027 & 0.00073 & 0.00018 & -0.00142 \\ 0.00074 & 0.00137 & 0.00023 & 0.00298 \\ 0.00018 & 0.00023 & 0.00028 & 0.00127 \\ -0.00142 & 0.00298 & 0.00127 & -0.03812 \end{pmatrix}$$

After this we calculate the robust sandwich form:

$$\Sigma(\theta_t)' = H^{-1} J H^{-1}$$

The robust form for the t distribution is:

$$\Sigma(\theta_t)' = \begin{pmatrix} 0.00379 & 0.00567 & 0.00102 & 0.02443 \\ 0.00567 & 0.01071 & 0.00243 & 0.02605 \\ 0.00102 & 0.00243 & 0.00168 & 0.00280 \\ 0.02443 & 0.02604 & 0.00281 & 0.21196 \end{pmatrix}$$

One can see that there is a difference between the robust sandwich form and the empirical form. This is because the robust sandwich form is more accurate than the empirical form. Though some values are equal in approximation.

To check whether this was correct we used instead of the market index and the IBM stock a uniform distribution for X and Y. We use $n = 4400$ because it is approximately the same length as the stock and market index vectors.

$$X, Y \sim UNIF(0, 1, 4400)$$

When using the uniform distribution we saw that the robust sandwich form and the empirical inverse hessian where almost equal to each other. This means that the model is correctly specified in Python and the actual hypothesis that the sandwich form is more accurate than the empirical form is not rejected.