SMF PROBLEMS 10. 7.6.2013.

Kalman filter and CAPM with time-varying betas.
Before starting the exercise, we recall some generalities about the CAPM model with time-varying beta. This models how well a stock performs against the market, and can be described by the following set of equations:

$$\beta(n+1) = \beta(n) + \epsilon(n+1),$$
$$r(n) - r_f(n) = \beta_n(r_M(n) - r_f(n)) + \eta(n),$$

where the $r(n)$ denote the return of a stock, the $r_f(n)$ the risk free return, and $r_M(n)$ the market return. with independent $\epsilon \sim \mathcal{N}(0, Q)$ and $\eta \sim \mathcal{N}(0, R)$; Q and R being actually scalars, since we are in dimension one. The purpose of this exercise is to estimate the β one step ahead of time.

Functions in R
Functions in R are defined through the keyword *function* and take a certain number of parameters as an input. However, if the **return** command is not written in the function, the return value will be the last expression evaluated in the function body. For example:

```
 norm<-function(x){
list(norm=sqrt(x%*%x),const=1337)
}
```

will return an item with two fields: the Euclidian norm of the vector x and the number 1337. Then, calling the function can be done as follows:

```
 normofx<-norm(seq(3,4))
```

returns an item containing two fields: the norm of the input and the number 1337.($seq(a, b)$ creates a vector of integers ranging from a to b). If one wants to manipulate the norm of the input, it can be done by the syntax **normofx$norm**. Similarly, if one wants to use the constant value, the corresponding object will be **normofx$const**.

(i) *Kalman updating function:* The purpose of this subquestion is to create a function that updates the Kalman estimates of the signal at time $n+1$, knowing everything we need to know at time n. In other words, and using the notation of the course, create a function **kalmanf.update** that computes $x_{n+1|n}$ and $\Sigma_{n+1|n}$ through the recursive formulae given above. Normally, this function takes the following parameters as input: y, the value of the observation at time n, the matrices Φ and G from above, the previous estimates $x_{n|n-1}$ and $\Sigma_{n|n-1}$, Q and R the covariance matrices of ϵ and η. (Assume that there is no correlation between ϵ and η.)

(ii) *Vector of Kalman estimates*: Create a function that returns the list of the Kalman filter estimates for x and Σ at all times of the considered period. This function should do two things:

- initialise the estimates of $x(0)$ and $\Sigma(0)$. The estimates can be extracted from a regression with no intercept of the observations against the Φ matrix, which is the identity in our case (for simplicity, you can hardcode $\Phi \equiv 1$ at this point already).

- make use of the function **kalman.update** to generate the vector of estimates of x and Σ across time.

Hence, the function takes as input the vector of observations post-initialisation y, the vector of values post-initialisation H, the initial estimate $H(0)$, a regression-type object $fit0$ (provided that you do the aforementioned regression outside of the kalmanf.estimate function), and Q.
(iii) *Application to the CAPM with time-varying betas.* Assume that the risk-free return is zero. Use the previous Kalman function to estimate the time-varying betas of Intel, American Express and JP Morgan, with Q set to 0.01. The calibration period can be taken to be, say, the first 20 observations. The data for this stock, as well as the Market returns (assimilated to the Dow Jones index), can be downloaded from my webpage.

NHB/PMBF