Excercise_2_R.R

lukas

2024-10-24

```r
######Computational part#####

#####libaries####
library(gmm) #for gmm estimation of SV model parameters
```

```
## Lade nötiges Paket: sandwich
```

```r
library(quantmod) # for S&P data
```

```
## Lade nötiges Paket: xts

## Lade nötiges Paket: zoo

##
## Attache Paket: 'zoo'

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     as.Date, as.Date.numeric

## Lade nötiges Paket: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

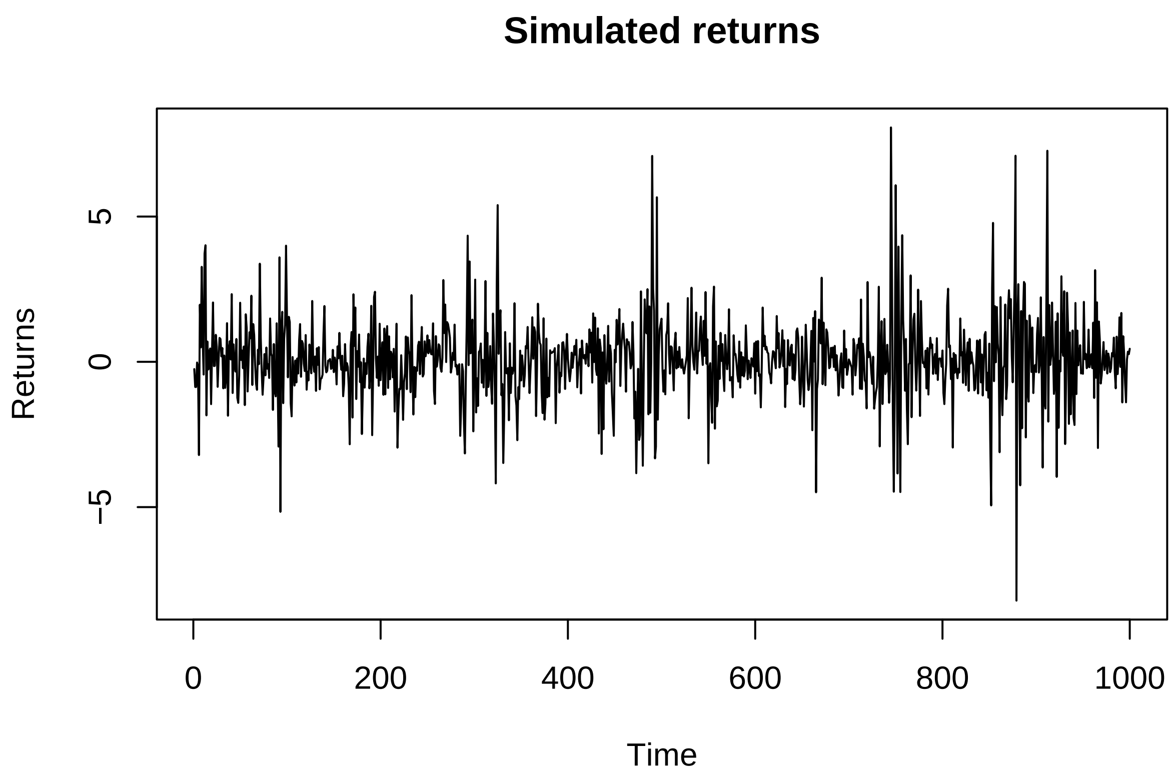
```r
source("help_functions.R")


#####Simulation of SV model####

set.seed(123)

y = simulate_SV(1000,0, 0.9, 0.25)

#Plot the simulated data
plot(y,type = "l",main = "Simulated returns", xlab = "Time", ylab = "Returns")
```




```r
# Define moment conditions

#Initial parameter guess
theta_start = c(0, 0.9, 0.25)

gmm_model = gmm(moment_conditions_SV, y, t0 = theta_start)
```

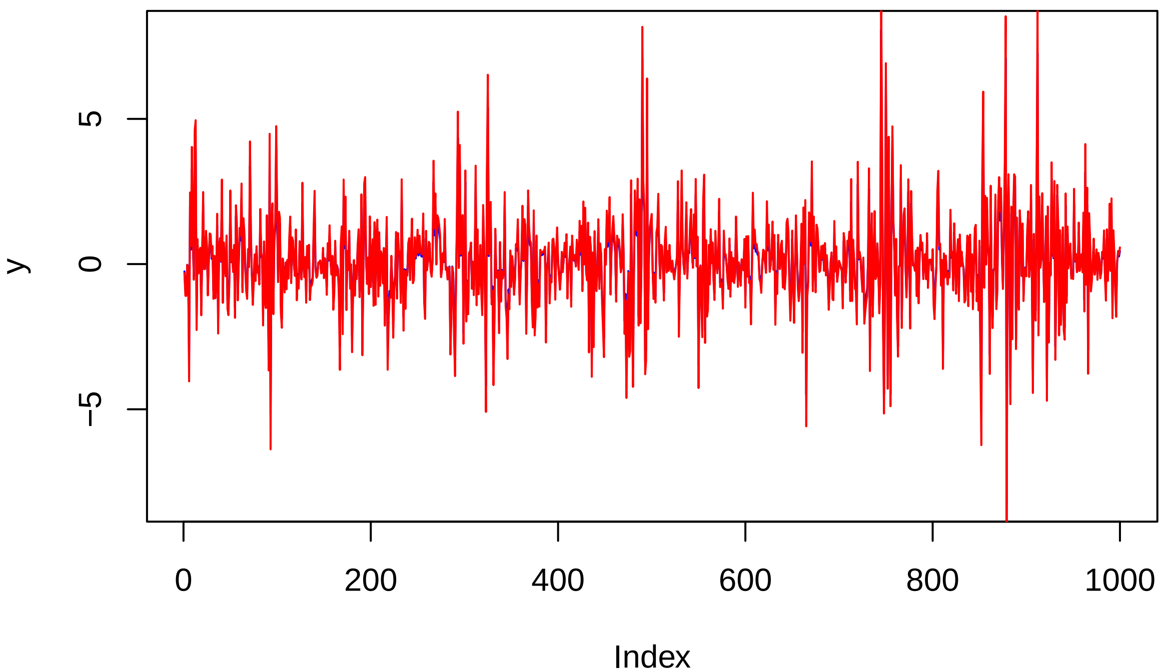
```
## Warning in FinRes.baseGmm.res(z, Model_info): The covariance matrix of the
## coefficients is singular
```

```r
summary(gmm_model)
```

```
##
## Call:
## gmm(g = moment_conditions_SV, x = y, t0 = theta_start)
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral(with bw =  0.72926 )
##
## Coefficients:
##           Estimate  Std. Error  t value   Pr(>|t|)
## Theta[1]  0.053333        Inf    0.000000  1.000000
## Theta[2]  0.885523        Inf    0.000000  1.000000
## Theta[3]  0.295717        Inf    0.000000  1.000000
##
## J-Test: degrees of freedom is -2
```

```
##                  J-test              P-value
## Test E(g)=0:    2.1537801573174e-08  *******
##
## Initial values of the coefficients
##   Theta[1]    Theta[2]    Theta[3]
## 0.05333313 0.88552341 0.29571668
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  74
## Gradian eval. =  NA
```

```r
#Plot actual against fitted values
fitted_returns = simulate_SV(1000,coef(gmm_model)[1],coef(gmm_model)[2],coef(gmm_model)[3])
plot(y, type = "l", col = "blue")
lines(fitted_returns,type = "l", col = "red")
```




```r
####Real Data Application####

#Download S&P 500 data
getSymbols("^GSPC", from = "2005-01-01", to = "2018-01-01")
```

```
## [1] "GSPC"
```

```r
sp500 = Cl(GSPC) #closing returns

#Compute log returns
log_returns = diff(log(sp500))[-1] * 100
#Replace zero returns with their empirical mean (to ensure that we can take exp)
log_returns[log_returns == 0] = mean(log_returns)
```

```r
#Estimate SV model using GMM
gmm_model_real = gmm(moment_conditions_SV, log_returns, t0 = c(0,0.9,0.25),optfct = "optim")
```

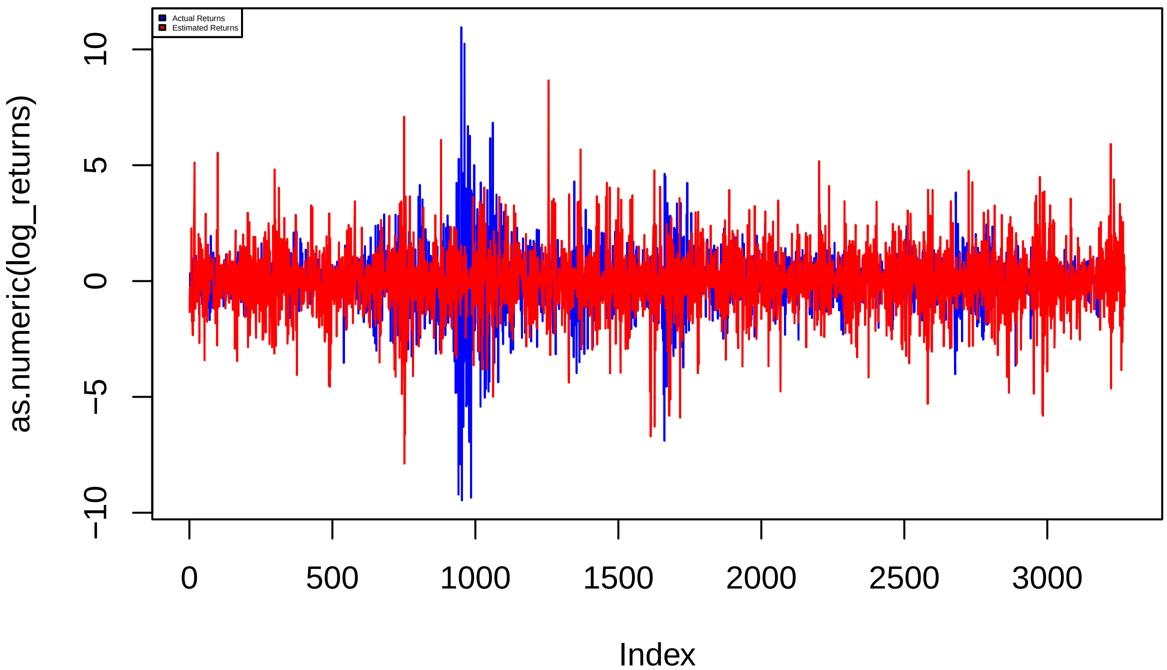
```
## Warning in FinRes.baseGmm.res(z, Model_info): The covariance matrix of the
## coefficients is singular
```

```r
summary(gmm_model_real)
```

```
##
## Call:
## gmm(g = moment_conditions_SV, x = log_returns, t0 = c(0, 0.9,
##     0.25), optfct = "optim")
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral(with bw =  1.44348 )
##
## Coefficients:
##           Estimate  Std. Error  t value   Pr(>|t|)
## Theta[1]  0.022874       Inf    0.000000  1.000000
## Theta[2]  0.857001       Inf    0.000000  1.000000
## Theta[3]  0.305335       Inf    0.000000  1.000000
##
## J-Test: degrees of freedom is -2
##                 J-test                P-value
## Test E(g)=0:    1.31021586474215e-10  *******
##
## Initial values of the coefficients
##   Theta[1]   Theta[2]   Theta[3]
## 0.02287367 0.85700094 0.30533542
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  96
## Gradian eval. =  NA
```

```r
#Plot real vs fitted values

fitted_returns = simulate_SV(length(log_returns),coef(gmm_model_real)[1],coef(gmm_model_real)[2],coef(g
plot(as.numeric(log_returns), type = "l", col = "blue")
lines(fitted_returns,type = "l", col = "red")
legend("topleft",legend = c("Actual Returns", "Estimated Returns"), fill = c("blue","red"),cex = 0.25)
```



Actual Returns
Estimated Returns
as.numeric(log_returns)
10
5
0
−5
−10
0
500
1000
1500
2000
2500
3000
Index