GAL Buckle 95

François Pelletier

8 mars 2014

Table des matières

0.1 Chargement des paquets 1
0.2 Constantes et données 1
0.3 Test de normalité 2
0.4 Données mises à l'échelle 2
0.5 Première estimation par QEE 2
0.6 Résultats de la première estimation par QEE 3
0.7 Seconde estimation par QEE 4
0.8 Résultats de la seconde estimation par QEE 4
0.9 Estimation par GMM 5

0.1 Chargement des paquets

```
> setwd("~/git/GAL-Buckle95/")
> library(actuar)
> library(MASS)
> library(xtable)
> library(multicore)
> library(moments)
> library(TTR)
> library(FourierStuff)
> library(GeneralizedAsymmetricLaplace)
> library(GMMStuff)
> library(OptionPricingStuff)
> library(QuadraticEstimatingEquations)
```

0.2 Constantes et données

```
> #Nombre de décimales affichées
> options(digits=6)
> #Marge pour intervalles de confiance
> alpha.confint <- 0.05
> #Marge pour test d'hypothèses
> alpha.test <- 0.05
```

```
> #Chargement des données
> RETURNS <- head(read.csv("abbeyn.csv",sep="\t",header=TRUE)[,1],-1)
> #Taille de l'échantillon
> n <- length(RETURNS)
```

0.3 Test de normalité

```
> EppsPulley.test(RETURNS)

Epps-Pulley Normality test

 T: 0.626033
 T*: 0.635568
p-value: 0.007178

$Tstat
[1] 0.626033

$Tmod
[1] 0.635568

$Zscore
[1] 2.44824

$Pvalue
[1] 0.00717788

$Reject
[1] TRUE
```

0.4 Données mises à l'échelle

```
> scaledRETURNS <- as.vector(scale(RETURNS))
```

0.5 Première estimation par QEE

```
> ## Point de départ
> pt.depart <- startparamGAL(scaledRETURNS)
> ## Fonctions pour les moments
> meanQEE <- function(param) mGAL(param,1)
> varianceQEE <- function(param) cmGAL(param,2)
> sdQEE <- function(param) sqrt(cmGAL(param,2))
> skewnessQEE <- function(param) cmGAL(param,3)
> kurtosisQEE <- function(param) cmGAL(param,4)
```

```
> ## Fonctions pour les dérivées
> dmeanQEE <- function(param) dmGAL(param,1)
> dsdQEE <- function(param) dmGAL(param,2)
> ## Estimation gaussienne
> optim1 <- optim(pt.depart,obj.gauss,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,dmeanQEE,dsd
> pt.optim1 <- optim1$par
> ## Estimation de crowder
> optim2 <- optim(pt.depart,obj.Crowder,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,skewnessQE
> pt.optim2 <- optim2$par
> ## Estimation de crowder modifiée
> optim3 <- optim(pt.depart,obj.Crowder.Mod,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,dmeanQ
> pt.optim3 <- optim3$par
```

0.6 Résultats de la première estimation par QEE

```
>         cov.optim1 <- covariance.QEE(M.gauss(pt.optim1,scaledRETURNS,meanQEE,varianceQEE,d
+                         V.gauss(pt.optim1,scaledRETURNS,meanQEE,varianceQEE,skewnessQEE,ku
>         cov.optim2 <- covariance.QEE(M.Crowder(pt.optim2,scaledRETURNS,varianceQEE,skewnes
+                         V.Crowder(pt.optim2,scaledRETURNS,varianceQEE,skewnessQEE,kurtosis
>         cov.optim3 <- covariance.QEE(M.Crowder.Mod(pt.optim3,scaledRETURNS,varianceQEE,ske
+                         V.Crowder.Mod(pt.optim3,scaledRETURNS,varianceQEE,dmeanQEE,dsdQEE)
>         confidence.interval.QEE(pt.optim1,cov.optim1,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.780018 -0.726048 -0.672077
[2,]  0.436002  0.596316  0.756630
[3,]  0.262650  0.359186  0.455722
[4,]  1.994757  2.021370  2.047982

>         confidence.interval.QEE(pt.optim2,cov.optim2,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.694457 -0.627404 -0.560351
[2,]  0.413764  0.640292  0.866820
[3,]  0.232650  0.334028  0.435405
[4,]  1.839966  1.878296  1.916626

>         confidence.interval.QEE(pt.optim3,cov.optim3,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.765288 -0.711439 -0.657589
[2,]  0.455485  0.606642  0.757798
[3,]  0.264669  0.362932  0.461195
[4,]  1.932691  1.960299  1.987906
```

0.7 Seconde estimation par QEE

```
> ## Estimation gaussienne
> optim4 <- optim(pt.optim1,obj.gauss,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,dmeanQEE,dsd
+                 ginv(V.gauss(pt.optim1,scaledRETURNS,meanQEE,varianceQEE,skewnessQEE,kurto
> pt.optim4 <- optim4$par
> ## Estimation de crowder
> optim5 <- optim(pt.optim2,obj.Crowder,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,skewnessQE
+                 ginv(V.Crowder(pt.optim2,scaledRETURNS,varianceQEE,skewnessQEE,kurtosisQEE
> pt.optim5 <- optim5$par
> ## Estimation de crowder modifiée
> optim6 <- optim(pt.optim3,obj.Crowder.Mod,gr=NULL,scaledRETURNS,meanQEE,varianceQEE,dmeanQ
+                 ginv(V.Crowder.Mod(pt.optim3,scaledRETURNS,varianceQEE,dmeanQEE,dsdQEE)))
> pt.optim6 <- optim6$par
```

0.8 Résultats de la seconde estimation par QEE

```
>         cov.optim4 <- covariance.QEE(M.gauss(pt.optim4,scaledRETURNS,meanQEE,varianceQEE,d
+                         V.gauss(pt.optim4,scaledRETURNS,meanQEE,varianceQEE,skewnessQEE,ku
>         cov.optim5 <- covariance.QEE(M.Crowder(pt.optim5,scaledRETURNS,varianceQEE,skewnes
+                         V.Crowder(pt.optim5,scaledRETURNS,varianceQEE,skewnessQEE,kurtosis
>         cov.optim6 <- covariance.QEE(M.Crowder.Mod(pt.optim6,scaledRETURNS,varianceQEE,ske
+                         V.Crowder.Mod(pt.optim6,scaledRETURNS,varianceQEE,dmeanQEE,dsdQEE)
>         confidence.interval.QEE(pt.optim4,cov.optim4,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.779792 -0.725853 -0.671914
[2,]  0.436017  0.596319  0.756622
[3,]  0.262456  0.358969  0.455482
[4,]  1.995452  2.022048  2.048644

>         confidence.interval.QEE(pt.optim5,cov.optim5,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.692712 -0.625874 -0.559036
[2,]  0.414139  0.640445  0.866750
[3,]  0.231568  0.332845  0.434122
[4,]  1.842116  1.880376  1.918636

>         confidence.interval.QEE(pt.optim6,cov.optim6,n)

         LOWER  ESTIMATE     UPPER
[1,] -0.766288 -0.712450 -0.658612
[2,]  0.455051  0.606193  0.757334
[3,]  0.264972  0.363196  0.461419
[4,]  1.934050  1.961614  1.989178
```

0.9 Estimation par GMM

```
>         ## GMM régulier
>         optim7 <- optim.GMM(pt.depart,conditions.vector=meanvariance.gmm.vector,data=scale
+                         meanf=meanQEE,variancef=varianceQEE)
>         ## GMM itératif
>         optim8 <- iterative.GMM(pt.depart,conditions.vector=meanvariance.gmm.vector,data=s
+                         meanf=meanQEE,variancef=varianceQEE)
```