Applied econometrics 6, Copula-GARCH model

Евгений Орлов

21.11.2014

### Данные

В качестве входных данных взяты цены закрытия по обыкновенным акциям Газпрома и Норильского Никеля в период с 01.01.2010 по 07.11.2014 включительно.

Источником данных выступила ИТС QUIK.

gazp.df <- read.csv("ГАЗПРОМ ао [Price].txt", header=TRUE)  
gmkn.df <- read.csv("ГМКНорНик [Price].txt", header=TRUE)  
tail(gazp.df)

## X.TICKER. X.PER. X.DATE. X.TIME. X.OPEN. X.HIGH. X.LOW. X.CLOSE.  
## 2191 GAZP [TQBR] Daily 20141030 0 138.09 139.90 137.62 139.50  
## 2192 GAZP [TQBR] Daily 20141031 0 140.50 141.50 139.49 141.50  
## 2193 GAZP [TQBR] Daily 20141103 0 141.00 144.00 140.26 143.39  
## 2194 GAZP [TQBR] Daily 20141105 0 143.01 143.82 141.11 143.20  
## 2195 GAZP [TQBR] Daily 20141106 0 143.35 146.25 143.05 144.10  
## 2196 GAZP [TQBR] Daily 20141107 0 143.50 147.00 142.73 144.46  
## X.VOL.  
## 2191 5463327  
## 2192 7197166  
## 2193 1966065  
## 2194 4707525  
## 2195 5571670  
## 2196 8263385

# Цены закрытия начиная с 01.01.2010 по 07.11.2014 включительно  
gazp.close <- gazp.df[gazp.df$X.DATE. >= 20100101, "X.CLOSE."]  
gmkn.close <- gmkn.df[gmkn.df$X.DATE. >= 20100101, "X.CLOSE."]

Была произведена проверка соответствия дат цен активов:

# Даты  
gazp.dates <- gazp.df[gazp.df$X.DATE. >= 20100101, "X.DATE."]  
gazp.dates <- as.Date(as.character(gazp.dates), format="%Y%m%d")  
gmkn.dates <- gmkn.df[gmkn.df$X.DATE. >= 20100101, "X.DATE."]  
gmkn.dates <- as.Date(as.character(gmkn.dates), format="%Y%m%d")  
date.comp <- gazp.dates == gmkn.dates  
if (sum(date.comp) / length(date.comp) == TRUE) {  
 print("Даты совпадают!")  
} else {  
 print("Даты не совпадают!")  
}

## [1] "Даты совпадают!"

На основе цен закрытия получен вектор доходностей каждого актива:

# Размер выборки  
samp.len <- length(gazp.close)  
print(paste0("Размер выборки: ", samp.len))

## [1] "Размер выборки: 1214"

# Вектор доходностей  
gazp.ret <- gazp.close[2:samp.len] / gazp.close[1:(samp.len-1)] - 1  
gmkn.ret <- gmkn.close[2:samp.len] / gmkn.close[1:(samp.len-1)] - 1

### Оценка частных GARCH-моделей

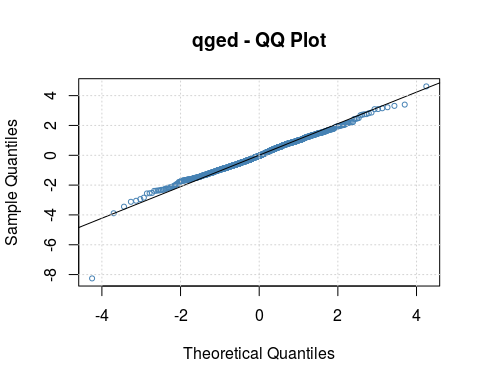
Использовалась простая модель GARCH(1, 1) c целевым условным распределением GED.

Графики "квантиль-квантиль" и значения логарифмических функций правдоподобия представлены ниже.

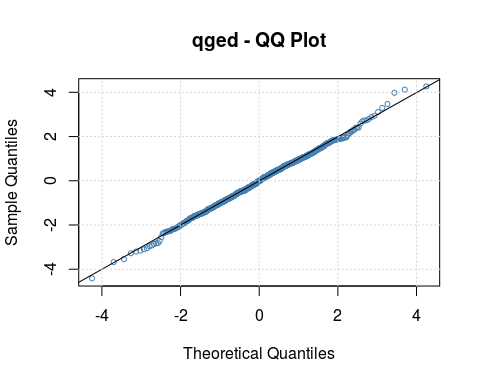
library(fGarch, quietly=TRUE)

## Loading required package: MASS  
##   
## Attaching package: 'fBasics'  
##   
## The following object is masked from 'package:base':  
##   
## norm

gazp.gfit <- garchFit(formula=~garch(1, 1), data=gazp.ret, cond.dist="ged",   
 shape=1.3, include.shape=FALSE, trace=FALSE)  
gmkn.gfit <- garchFit(formula=~garch(1, 1), data=gmkn.ret, cond.dist="ged",   
 shape=1.3, include.shape=FALSE, trace=FALSE)  
plot(gazp.gfit, which=13)



plot(gmkn.gfit, which=13)



print(paste(gazp.gfit@fit$params$llh, gmkn.gfit@fit$params$llh, sep=' '))

## [1] "1649.49770242649 1637.91639014874"

### Выбор копулы для моделирования двумерного распределения остатков

Формируем выборку наблюдений (значений частных функций распределения стандартизированных остатков) для подгона копулы.

# стандартизированные остатки  
z <- matrix(nrow=length(gazp.ret), ncol=2)  
z[, 1] <- gazp.gfit@residuals / gazp.gfit@sigma.t  
z[, 2] <- gmkn.gfit@residuals / gmkn.gfit@sigma.t  
z1.fit <- gedFit(z[, 1])  
z1.fit

## $par  
## mean sd nu   
## -0.01418302 0.99182578 1.38803970   
##   
## $objective  
## [1] 1691.029  
##   
## $convergence  
## [1] 0  
##   
## $iterations  
## [1] 12  
##   
## $evaluations  
## function gradient   
## 21 45   
##   
## $message  
## [1] "relative convergence (4)"

z2.fit <- gedFit(z[, 2])  
z2.fit

## $par  
## mean sd nu   
## -0.01217782 0.99288372 1.39225533   
##   
## $objective  
## [1] 1692.696  
##   
## $convergence  
## [1] 0  
##   
## $iterations  
## [1] 26  
##   
## $evaluations  
## function gradient   
## 44 120   
##   
## $message  
## [1] "relative convergence (4)"

# частные распределения остатков  
mean.z <- c(z1.fit$par[1], z2.fit$par[1])  
sd.z <- c(z1.fit$par[2], z2.fit$par[2])  
nu.z <- c(z1.fit$par[3], z1.fit$par[3])  
#xi.z <- c(1, gmkn.gfit@fit$par["skew"])  
cdf.z <- matrix(nrow=length(gazp.ret), ncol=2)   
for (i in 1:2) {  
 cdf.z[, i] <- pged(z[, i], mean=mean.z[i], sd=sd.z[i],   
 nu=nu.z[i])  
}

Среди возможных вариантов копулы по принципу максимального значения функции правдоподобия был выбрана копула Стьюдента с 10 степенями свободы.

library(copula)  
# объявление копулы  
norm.cop <- normalCopula(dim=2)  
stud5.cop <- tCopula(dim=2, param=0.5, df=5, df.fixed=TRUE,dispstr="un")   
stud10.cop <- tCopula(dim=2, param=0.5, df=10, df.fixed=TRUE,dispstr="un")   
stud20.cop <- tCopula(dim=2, param=0.5, df=20, df.fixed=TRUE,dispstr="un")   
gumb.cop <- gumbelCopula(dim=2)   
clay.cop <- claytonCopula(dim=2)   
# подгонка копулы  
norm.fit <- fitCopula(cdf.z, copula=norm.cop)   
stud5.fit <- fitCopula(cdf.z, copula=stud5.cop)   
stud10.fit <- fitCopula(cdf.z, copula=stud10.cop)   
stud20.fit <- fitCopula(cdf.z, copula=stud20.cop)   
gumb.fit <- fitCopula(cdf.z, copula=gumb.cop)   
clay.fit <- fitCopula(cdf.z, copula=clay.cop)   
# выбор оптимальной копулы  
norm.fit@loglik;stud5.fit@loglik;stud10.fit@loglik

## [1] 211.4267

## [1] 218.426

## [1] 221.2284

stud20.fit@loglik;gumb.fit@loglik;clay.fit@loglik

## [1] 218.7848

## [1] 186.2579

## [1] 184.9854

### Оценка финансового риска

C помощью метода Монте-Карло были получена выборка доходностей портфеля (50% Газпром, 50% Норильский никель). С помощью копулы, выбранной на предыдущем шаге, получаем векторы стандартизированных остатков. Параметры получаем из GARCH-моделей.

# моделирование доходностей портфеля методом Монте-Карло  
N <- 10^4  
cdf.sim <- rCopula(n=N, copula=stud10.fit@copula)  
z.sim <- matrix(nrow=N, ncol=2)  
for (i in 1:2) {  
 z.sim[, i] <- qged(cdf.sim[, i], mean=mean.z[i], sd=sd.z[i],   
 nu=nu.z[i])  
}  
gazp.frc <- predict(gazp.gfit, n.ahead=1)  
gmkn.frc <- predict(gmkn.gfit, n.ahead=1)  
mu <- c(gazp.frc[, 1], gmkn.frc[, 1])  
sigma <- c(gazp.frc[, 3], gmkn.frc[, 3])  
w <- c(0.5, 0.5) # веса активов в портфеле  
port.sim <- w[1]\*(mu[1]+sigma[1]\*z.sim[, 1]) + w[2]\*(mu[2]+sigma[2]\*z.sim[, 2])

Был рассчитан 90%-й VaR и ES по всей выборке наблюдений.

# измерители риска  
alpha <- 0.1  
port.sim <- sort(port.sim)  
VaR <- port.sim[alpha\*N]  
ES <- mean(port.sim[1:(alpha\*N-1)])

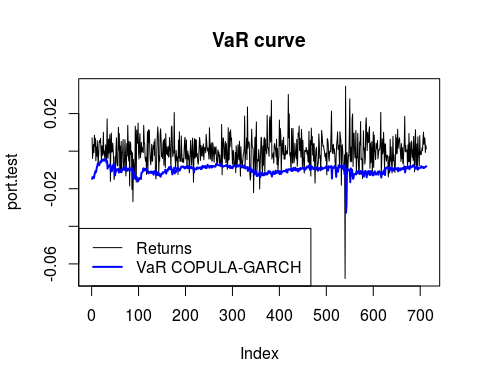
### Кривая VaR на основе COPULA-GARCH модели

При расчете кривой VaR размер обучающей выборки пришлось уменьшить до 100, иначе получалось низкое p-value в тесте Купика.

# Параметры для расчета VaR  
T1 <- 500  
T2 <- length(gazp.ret) - T1  
alpha <- 0.1  
N <- 10^3  
w <- c(0.5, 0.5) # Веса активов в портфеле  
VaR.copula\_garch <- numeric()  
  
h <- 100 # длина обучающей выборки  
for (i in (T1+1):(T1+T2)) {  
 # обучающая выборка  
 gazp.train <- gazp.ret[(i-h):(i-1)]  
 gmkn.train <- gmkn.ret[(i-h):(i-1)]   
 # одномерные GARCH-модели  
 gazp.gfit <- garchFit(formula=~garch(1, 1), data=gazp.train, cond.dist="ged",   
 shape=1.3, include.shape=FALSE, trace=FALSE)  
 gmkn.gfit <- garchFit(formula=~garch(1, 1), data=gmkn.train, cond.dist="ged",  
 shape=1.3, include.shape=FALSE, trace=FALSE)  
 # частные распределения стандартизированных остатков  
 z <- matrix(nrow=T1, ncol=2)  
 z[, 1] <- gazp.gfit@residuals / gazp.gfit@sigma.t  
 z[, 2] <- gmkn.gfit@residuals / gmkn.gfit@sigma.t  
 z1.fit <- gedFit(z[, 1])  
 z2.fit <- gedFit(z[, 2])  
 mean.z <- c(z1.fit$par[1], z2.fit$par[1])  
 sd.z <- c(z1.fit$par[2], z2.fit$par[2])  
 nu.z <- c(z1.fit$par[3], z1.fit$par[3])  
 cdf.z <- matrix(nrow=T1, ncol=2)  
 for(t in 1:2) {  
 cdf.z[, t] <- pged(z[, t], mean=mean.z[t], sd=sd.z[t],   
 nu=nu.z[t])  
 }  
 # подгонка копулы  
 stud10.cop <- tCopula(dim=2, param=0.5, df=10, df.fixed=TRUE, dispstr="un")   
 stud10.fit <- fitCopula(cdf.z, copula=stud10.cop)  
 # моделирование доходностей портфеля  
 cdf.sim <- rCopula(n=N, copula=stud10.fit@copula)  
 z.sim <- matrix(nrow=N, ncol=2)  
 for (t in 1:2) {  
 z.sim[, t] <- qged(cdf.sim[, t], mean=mean.z[t], sd=sd.z[t],   
 nu=nu.z[t])  
 }  
 gazp.frc <- predict(gazp.gfit, n.ahead=1)  
 gmkn.frc <- predict(gmkn.gfit, n.ahead=1)  
 mu <- c(gazp.frc[, 1], gmkn.frc[, 1])  
 sigma <- c(gazp.frc[, 3], gmkn.frc[, 3])  
 port.sim <- w[1]\*(mu[1]+sigma[1]\*z.sim[, 1])   
 + w[2]\*(mu[2]+sigma[2]\*z.sim[, 2])  
 port.sim <- sort(port.sim)  
 # вычисление VaR  
 VaR.copula\_garch[i-T1] <- port.sim[alpha\*N]  
}

# График кривой VaR  
port.test <- w[1]\*gazp.ret[(T1+1):(T1+T2)]   
 + w[2]\*gmkn.ret[(T1+1):(T1+T2)]

plot(port.test, type="l", main="VaR curve")  
lines(VaR.copula\_garch, col="blue", lwd=2)  
legend('bottomleft', c("Returns", "VaR COPULA-GARCH"), col=c("black", "blue"), lty=c(1, 1), lwd=c(1, 2))



Для полученной кривой VaR был проведен тест Купика и рассчитаны значения функции потерь Лопеса и Бланко-Ила.

# Верификация VaR  
# Частота пробоев  
kupiec.test <- function(ret, VaR, alpha) {  
 # Тест Купика:  
 # H0: модельная и эмпирическая частоты пробоя VaR совпадают  
 K <- sum(ret < VaR)  
 T2 <- length(ret)  
 alpha0 <- K / T2  
 S <- -2\*log((1-alpha)^(T2-K) \* alpha^K) + 2\*log((1-alpha0)^(T2-K) \* alpha0^K)  
 p.value <- 1-pchisq(S, df=1)  
 return(c(alpha0, p.value))  
}  
  
# Глубина пробоев  
lopez.lf <- function(ret, VaR) {  
 # Функция потерь Лопеса  
 K <- sum(ret < VaR)  
 value <- sum((ret-VaR)^2\*(ret < VaR)) / K  
}  
  
blanco.lf <- function(ret, VaR) {  
 # Функция потерь Бланко-Ила  
 K <- sum(ret < VaR)  
 value <- sum((ret-VaR)/VaR \* (ret < VaR)) / K  
}  
  
# Тест VaR-кривой на частоту пробоев  
print(paste0("Kupiec test, alpha = ", alpha))

## [1] "Kupiec test, alpha = 0.1"

kup <- kupiec.test(port.test, VaR.copula\_garch, alpha)  
print(paste0("COPULA-GARCH: alpha0 = ", round(kup[1], 6),   
 ", p-value = ", round(kup[2], 6)))

## [1] "COPULA-GARCH: alpha0 = 0.096774, p-value = 0.772956"

# Оценка глубины пробоев  
print(paste0("Lopez loss function: ",   
 round(lopez.lf(port.test, VaR.copula\_garch), 6)))

## [1] "Lopez loss function: 7.4e-05"

print(paste0("Blanco loss function: ",   
 round(blanco.lf(port.test, VaR.copula\_garch), 6)))

## [1] "Blanco loss function: 0.481494"