

Моделирование волатильности финансовых активов с помощью GARCH-моделей

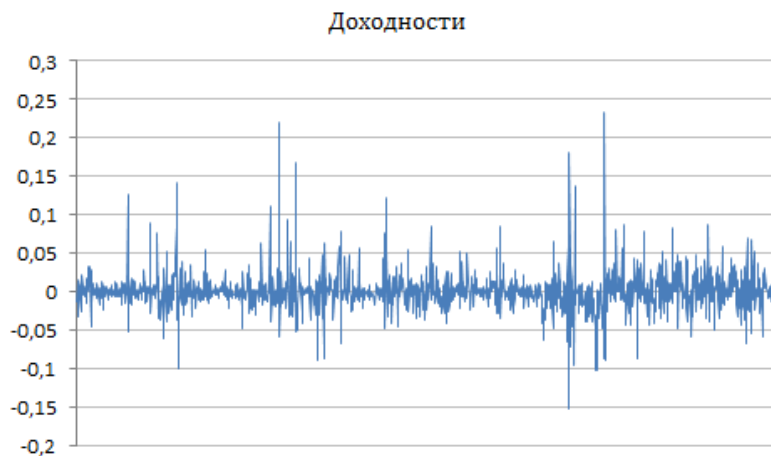
Финансовая эконометрика

Моделирование средней доходности

Пусть y_t — доходность актива, тогда уравнение для средней доходности по модели ARMA(m,n) записывается так:

$$y_t = \mu + \sum_{i=1}^m a_i y_{t-i} + \sum_{j=1}^n b_j \varepsilon_{t-j} + \varepsilon_t = \mu + a(L)y_t + b(L)\varepsilon_t$$

Для финансовых временных рядов характерен эффект кластеризации волатильности



Возникает задача моделирования дисперсии доходности

Тест на ARCH-эффекты

Тест множителей Лагранжа (LM-тест)

Пусть $e_t = y_t - \hat{y}_t$. Рассмотрим регрессию:

$$e_t^2 = \delta_0 + \delta_1 e_{t-1}^2 + \dots + \delta_q e_{t-q}^2 + \varepsilon_t$$

$H_0: \delta_1 = \dots = \delta_q = 0$ (нет ARCH-эффектов)

$H_{alt}: \exists j \in \{1; \dots; q\} : \delta_j \neq 0$

Пусть $ESS_0 = \sum_{t=1+q}^T (e_t^2 - \overline{e^2})$, $\overline{e^2} = \frac{1}{T} \sum_{t=1}^T e_t^2$ и

$ESS_1 = \sum_{t=1+q}^T \hat{\varepsilon}_t^2$, тогда

$$S = \frac{(ESS_0 - ESS_1)/q}{ESS_1/(T-2q-1)} \sim^{H_0} \chi^2(q)$$

LM-тест в R

исходные данные

```
library(datasets)
dax <- EuStockMarkets[, "DAX"]

T <- length(dax)-1
dax <- dax[2:(T+1)]/dax[1:T] - 1
```

LM-тест

```
library(FinTS)
ArchTest(dax, lags=12)
```

```
ARCH LM-test; Null hypothesis: no ARCH effects
data:  dax
Chi-squared = 85.4761, df = 12, p-value = 3.686e-13
```

Моделирование волатильности

- Уравнения для дисперсии по модели GARCH(p,q):

$$\varepsilon_t = z_t \sigma_t, \quad z_t \sim iid(0; 1)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 = \omega + \alpha(L) \varepsilon_t^2 + \beta(L) \sigma_t^2$$

Если $\forall i \beta_i = 0$, то $GARCH(p,q) \sim ARCH(p,q)$

- Степенное обобщение — модель APARCH(p,q):

$$\varepsilon_t = z_t \sigma_t, \quad z_t \sim iid(0; 1)$$

$$\sigma_t^\delta = \omega + \sum_{i=1}^p \alpha_i (|\varepsilon_{t-i}| - \gamma_i \varepsilon_{t-i})^\delta + \sum_{j=1}^q \beta_j \sigma_{t-j}^\delta,$$

где $\delta > 0$, $-1 < \gamma_i < 1$

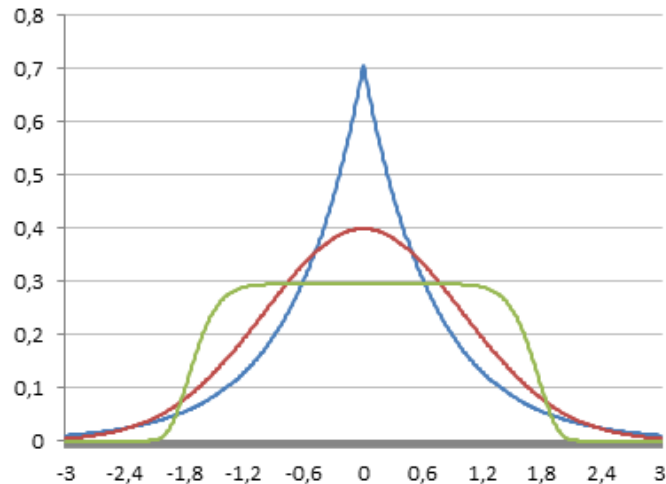
Если $\forall i \gamma_i = 0$ и $\delta = 2$, то $APARCH(p,q) \sim GARCH(p,q)$

Generalized Error Distribution (GED)

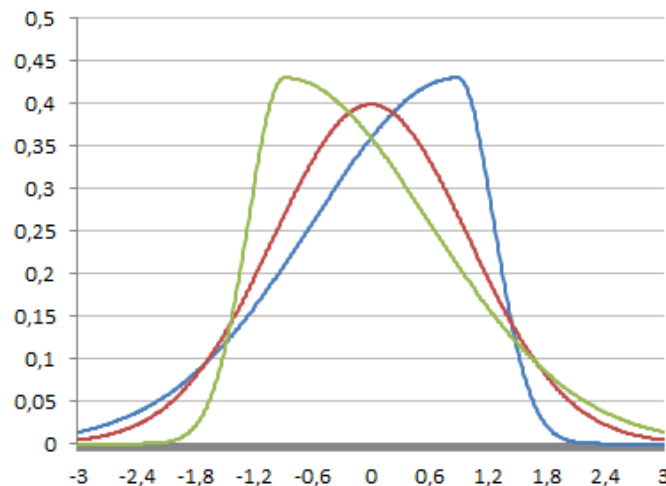
$$f_{GED}(x; \mu, \sigma, \nu) = \frac{\frac{1}{\nu} \exp\left(-\frac{1}{2} \left|\frac{x - \mu}{\sigma}\right|^\nu\right)}{2^{\frac{1}{\nu} + 1} \sigma \cdot \Gamma\left(\frac{1}{\nu} + 1\right)}$$

$\Gamma(.)$ – гамма-функция

Формы GED в зависимости от параметров



Параметр ν определяет
общую форму
распределения



Существует также
асимметричный вариант GED
с дополнительным
параметром $\xi > 0$

Общая схема расчёта модели APARCH

```
library(fGarch)
```

оценка параметров модели

```
dax.gfit <- garchFit(formula=~arma(m,n)+aparch(p,q), data=dax,  
cond.dist=[...], include.delta=T/F, leverage=T/F, trace=FALSE)
```

вместо [...] следует подставить распределение z_t :

“norm”, “snorm”, “std”, “sstd”, “ged”, “sged” или другие

используя комбинации степенного параметра и рычага

можно получать различные частные случаи модели

Расчёт частных случаев модели APARCH

GARCH(1,1)

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

```
garchFit(formula=~aparch(1,1), data=dax, delta=2,  
include.delta=FALSE, leverage=FALSE, trace=FALSE)
```

TS-GARCH(1,1)

$$\sigma_t = \omega + \alpha_1 |\varepsilon_{t-1}| + \beta_1 \sigma_{t-1}$$

```
garchFit(formula=~aparch(1,1), data=dax, delta=1,  
include.delta=FALSE, leverage=FALSE, trace=FALSE)
```

T-GARCH(1,1) (GJR-GARCH)

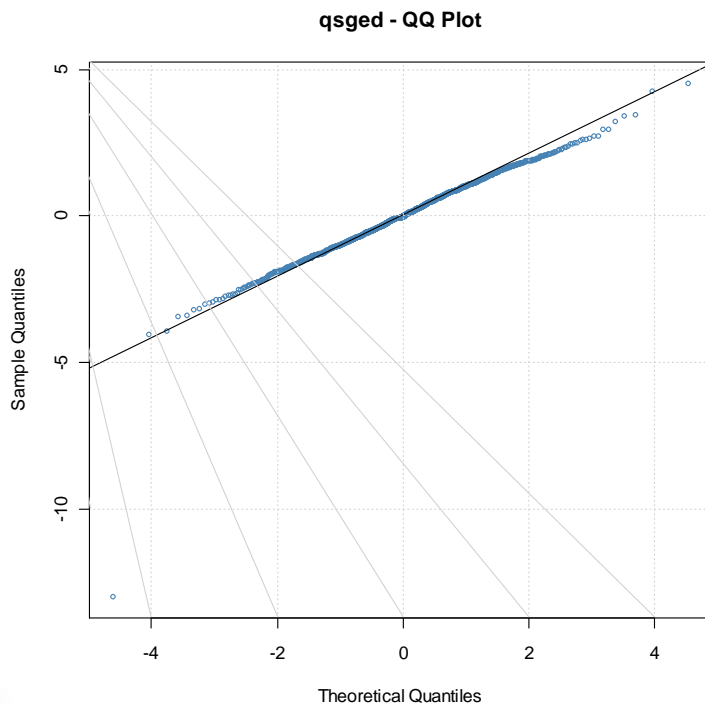
$$\sigma_t^2 = \omega + \alpha_1^* \varepsilon_{t-1}^2 + \gamma_1^* \cdot I(\varepsilon_{t-1} > 0) + \beta_1 \sigma_{t-1}^2$$

```
garchFit(formula=~aparch(1,1), data=dax, delta=2,  
include.delta=FALSE, leverage=TRUE, trace=FALSE)
```

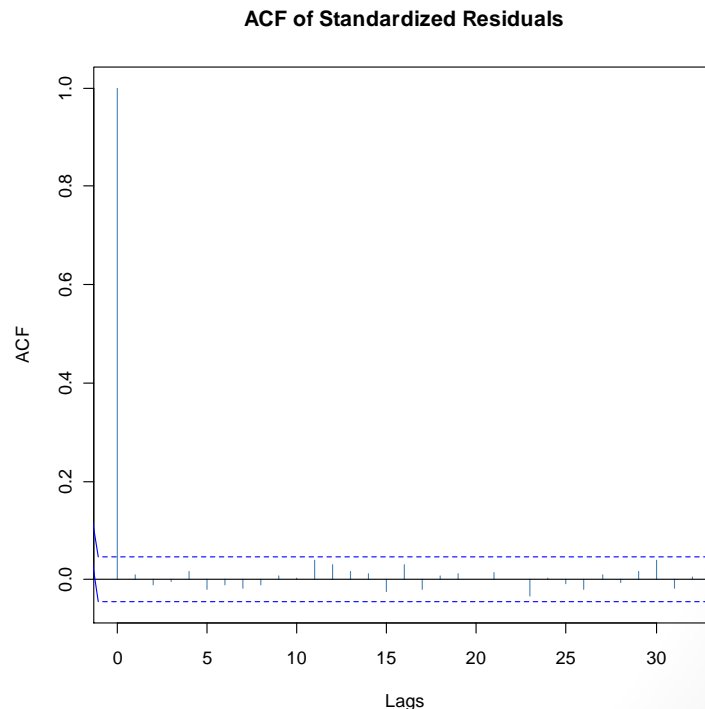
Графический анализ модели

```
dax.gfit <- garchFit(formula=~aparch(1,1),data=dax,delta=2,  
include.delta=FALSE,leverage=TRUE,cond.dist="sged",  
shape=1.25,include.shape=FALSE,trace=FALSE)
```

```
plot(dax.gfit,which=[...])
```



which=13



which=10

Стационарность

Пусть $\{\xi_t\}$ — стохастический процесс, $F_{\vec{\xi}}(\xi_{t_1}, \dots, \xi_{t_k})$ — совместная функция распределения $\{\xi_t\}$ в период (t_1, \dots, t_k)

Процесс $\{\xi_t\}$ называется стационарным, если

$$\forall k, \forall \tau F_{\vec{\xi}}(\xi_{t_1}, \dots, \xi_{t_k}) = F_{\vec{\xi}}(\xi_{t_1+\tau}, \dots, \xi_{t_k+\tau})$$

Стационарность в широком смысле:

$$\begin{cases} E(\xi_t) = E(\xi_{t+\tau}) \\ \text{corr}(\xi_{t_1}, \xi_{t_2}) = \text{corr}(\xi_{t_1+\tau}, \xi_{t_2+\tau}) \end{cases}, \forall \tau$$

Тесты на единичный корень в R

```
library(tseries)
```

ADF-тест

```
adf.test(dax)
```

```
Dickey-Fuller = -11.1348, Lag order = 12, p-value = 0.01  
alternative hypothesis: stationary
```

PP-тест

```
pp.test(dax)
```

```
Dickey-Fuller Z(alpha) = -1759.696, Truncation lag parameter = 8,  
p-value = 0.01 alternative hypothesis: stationary
```

KPSS-тест

```
kpss.test(dax, null="Level")
```

```
KPSS Level = 0.4634, Truncation lag parameter = 9,  
p-value = 0.04991
```

Прогноз по модели ARMA-GARCH

прогноз среднего и дисперсии на i шагов вперёд

```
dax.frc <- predict(dax.gfit,n.ahead=i)
```

```
dax.frc[,1] # вектор средних
```

```
dax.frc[,3]^2 # вектор дисперсий
```

расчёт границы потерь

```
alpha <- 0.05
```

```
VaR <- dax.frc[1,1]+dax.frc[1,3]*qged(alpha,mean=0,sd=1,  
nu=dax.gfit@fit$par["shape"])
```

Кривая VaR

Кривая VaR — набор последовательных во времени значений VaR

```
T1 <- 6*260; T2 <- T - T1 # обучающая и экзаменующая выборки
```

```
# на пространстве экзаменующей выборки построим набор
```

```
# последовательных значений VaR
```

```
VaR <- numeric()
```

```
h <- 0.5*260
```

```
for (i in (T1+1):(T1+T2)) {
```

```
  h.dax <- dax[(i-h):(i-1)]
```

```
  dax.gfit <- garchFit(formula=~aparch(1,1), data=h.dax,  
    delta=2, include.delta=FALSE, leverage=TRUE, cond.dist="sged",  
    shape=1.5, include.shape=FALSE, trace=FALSE)
```

```
  dax.frc <- predict(dax.gfit, n.ahead=1)
```

```
  VaR[i-T1] <- dax.frc[1,1] + dax.frc[1,3] * qsged(alpha, mean=0, sd=1,  
    nu=1.5, xi=dax.gfit@fit$par["skew"])
```

```
}
```

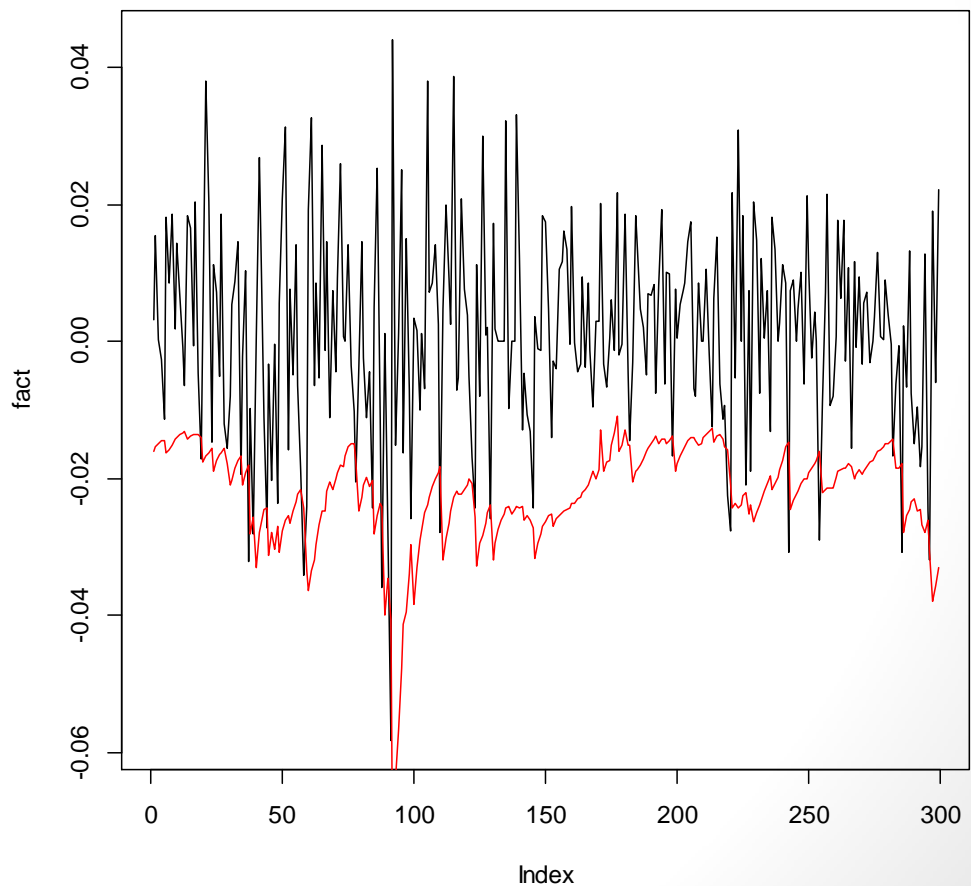
Кривая VaR

сравнение оценок риска с фактом

```
fact <- dax[(T1+1):(T1+T2)]
```

```
plot(fact, type="l")
```

```
lines(VaR, col="red")
```



Кривая VaR

Тест Купика

Идея состоит в сравнении модельной и эмпирической частот превышений фактическими убытками границы VaR

$$K = \sum I(x_t < VaR_t), \quad \alpha_0 = \frac{K}{T_2}$$

$$H_0: \alpha_0 = \alpha$$

Статистика:

$$S = -2 \ln((1 - \alpha)^{T_2 - K} \alpha^K) + 2 \ln((1 - \alpha_0)^{T_2 - K} \alpha_0^K) \sim \chi^2(1)$$

тест Купика в R:

```
K <- sum(fact<VaR); 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)
```

alpha	0.050
alpha0	0.067
p.value	0.202

Модель «Copula–GARCH»

Формализация модели

Уравнения для дисперсии по частным GARCH-моделям:

$$\varepsilon_{i,t} = z_{i,t}\sigma_{i,t}, \quad z_{i,t} \sim iid(0; 1)$$

$$\sigma_{i,t}^2 = \omega_i + \sum_{k=1}^p \alpha_{i,k} \varepsilon_{i,t-k}^2 + \sum_{k=1}^q \beta_{i,k} \sigma_{i,t-k}^2$$

$$i \in \{1; \dots; d\}$$

Этапы моделирования:

1. Оценка частных GARCH-моделей;
2. Расчёт условных стандартизированных остатков $z_{i,t}$
3. Моделирование многомерной величины z_t

Модель «copula-GARCH» в R

одномерные GARCH-модели

```
library(fGarch)
dax.gfit <- garchFit(data=dax, formula=~garch(1,1),
  shape=1.25, include.shape=F, cond.dist="ged", trace=F)
smi.gfit <- garchFit(data=smi, formula=~garch(1,1),
  shape=1.3, include.shape=F, cond.dist="sged", trace=F)
```

стандартизированные остатки

```
z <- matrix(nrow=T, ncol=2)
z[,1] <- dax.gfit$residuals / dax.gfit@sigma.t
z[,2] <- smi.gfit$residuals / smi.gfit@sigma.t
```

частные распределения остатков

```
mean <- c(0,0); sd <- c(1,1); nu <- c(1.25,1.3)
xi <- c(1, smi.gfit@fit$par["skew"])

cdf <- matrix(nrow=T, ncol=2)
for (i in 1:2) cdf[,i] <- psged(z[,i], mean=mean[i],
  sd=sd[i], nu=nu[i], xi=xi[i])
```

Модель «copula–GARCH» в R

подгонка копул

```
norm.fit <- fitCopula(cdf, copula=norm.cop)
stud.fit <- fitCopula(cdf, copula=stud.cop)
gumb.fit <- fitCopula(cdf, copula=gumb.cop)
clay.fit <- fitCopula(cdf, copula=clay.cop)
```

метод Монте-Карло

```
cdf.sim <- rcopula(n=N, copula=stud.fit@copula)

z.sim <- matrix(nrow=N, ncol=2)
for (i in 1:2) z.sim[,i] <- qsged(cdf.sim[,i],
  mean=mean[i], sd=sd[i], nu=nu[i], xi=xi[i])

frc1 <- predict(dax.gfit, n.ahead=1)
frc2 <- predict(smi.gfit, n.ahead=1)

mu <- c(frc1[,1], frc2[,1])
sigma <- c(frc1[,3], frc2[,3])
```

Оценка финансового риска

модельные доходности портфеля

```
prt.sim <- w[1]*(mu[1]+sigma[1]*z.sim[,1]) +  
           w[2]*(mu[2]+sigma[2]*z.sim[,2])
```

измерители риска

```
prt.sim <- sort(prt.sim)  
VaR <- prt.sim[alpha*N]  
ES <- mean(prt.sim[1:(alpha*N-1)])
```

VaR	-0.017
ES	-0.026

Домашнее задание

- рассчитать показатели VaR и ES для портфеля финансовых активов
- построить кривую VaR
- провести тест Купика и рассчитать значения функций потерь
- написать комментарии

Исходные данные – котировки с сайтов finam.ru, finance.yahoo.com и др.