Моделирование волатильности временных рядов с помощью GARCH-моделей

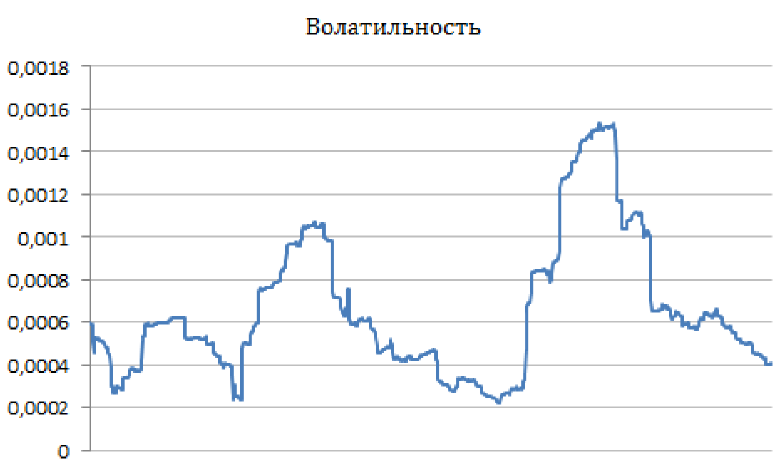
ЦМФ

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

Выбор модели для среднего

```
library(tseries)
```

рассматриваемые варианты порядка arma-модели (значений m и n)

```
order = expand.grid(0:5, 0:5)
```

переменные для хранения моделей и критериев их качества

```
nmodels = nrow(order)
models = vector("list", nmodels)
aics = numeric(nmodels)
```

оценка параметров моделей

```
for (i in 1:nrow(order)) {
  models[[i]] = arma(x, order=c(order[i,1], order[i,2]))
  aics[i] = summary(models[[i]])$aic
}
```

на случай возникновения ошибок в моделях

```
aics[unlist(lapply(models, is.null))] = Inf
```

просмотр наилучшей модели

```
summary(models[[which.min(aics)]])
```

Тест на ARCH-эффекты

Тест множителей Лагранжа (LM-тест)

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

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

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

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

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

$\varepsilon_t = z_t\sigma_t, \qquad z_t \sim idd(0;1)$

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

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

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

$\varepsilon_t = z_t\sigma_t, \qquad z_t \sim idd(0;1)$

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

где $\delta > 0, \quad -1 < \gamma_i < 1$

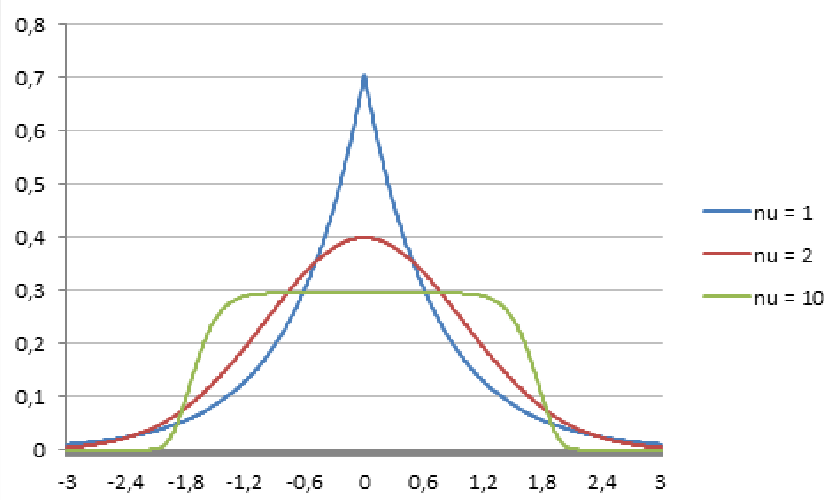
Если $\forall i\ \gamma_i = 0$ и $\delta = 2$, то APARCH(p,q) ~ GARCH(p,q)

Generalized Error Distribution (GED)

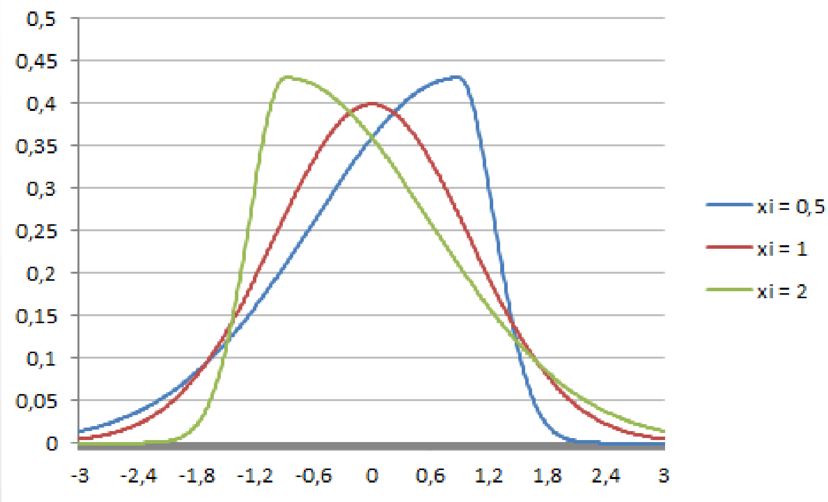
$$f_{GED}\left(x;\mu,\sigma,\nu\right)=\frac{\frac{1}{\nu}\exp\left(-\frac{1}{2}\left|\frac{x-\mu}{\sigma}\right|^{\nu}\right)}{2^{\frac{1}{\nu}+1}\sigma\cdot\Gamma\left(\frac{1}{\nu}+1\right)}$$

Г(.) – 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)

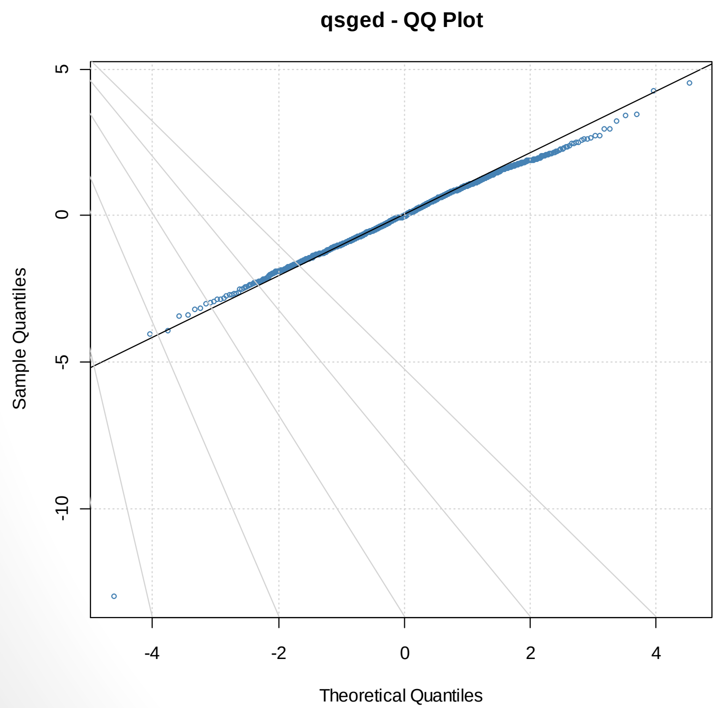
$\sigma_t^2 = \omega + \alpha_1^* \varepsilon_{t-1}^2 + \gamma_1^* \cdot I(\varepsilon_{t-1} > 0) + \beta_1 \sigma_{t-1}^2$

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

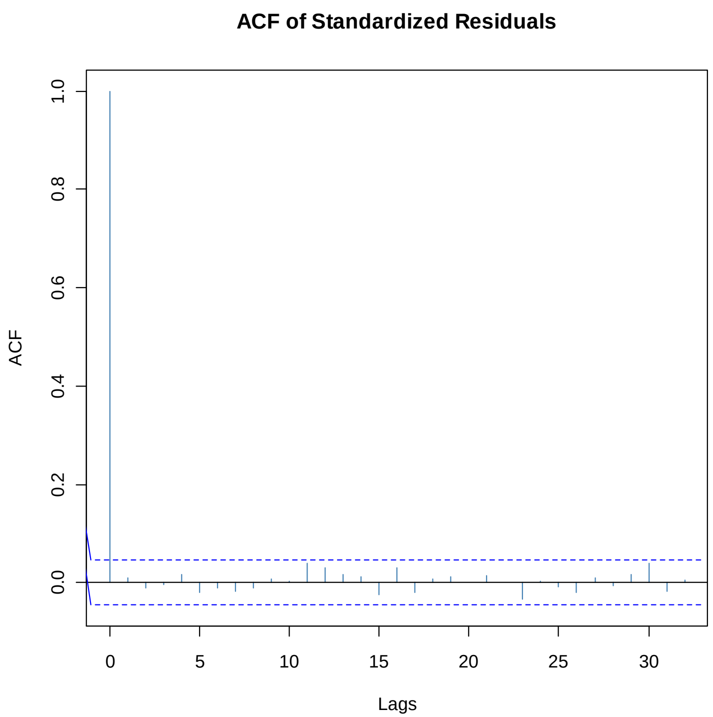
Графический анализ модели

```
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:

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

Тесты на стационарность (на отсутствие единичных корней)

- Расширенный тест Дики-Фуллера (ADF)

$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t$

$H_0: \gamma = 0$ (единичный корень)

$H_{alt}: \gamma < 0$

Тест Филлипса-Перрона (PP)

$y_t = \delta y_{t-1} + \varepsilon_t$

$H_0: |\delta| = 1$ (единичный корень)

$H_{alt}: |\delta| < 1$

Тест Квятковского-Филлипса-Шмидта-Шина (KPSS)

$y_t = \alpha t + r_t + \varepsilon_t,\ r_t \sim RW, \varepsilon_t \sim I(0)$

$r_t = r_{t-1} + u_t,\ u_t \sim iid(0; \sigma_u^2)$

$H_0: \sigma_u^2 = 0$ (стационарность)

$H_{alt}: \sigma_u^2 > 0$

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

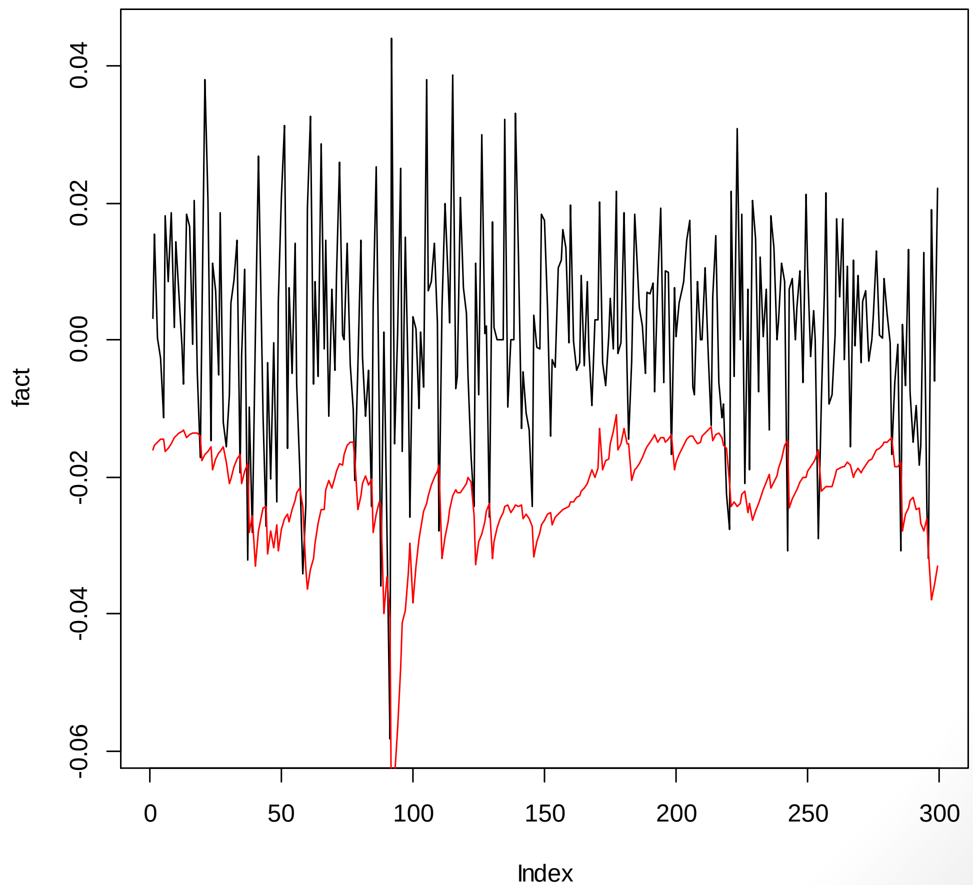
последовательных значений VaR

```
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")
```



Домашнее задание

В файле «ts_garch.csv» находятся 1000 значений временного ряда. Вашей задачей является построение кривой VaR для последних 500 значений

Ответы принимаются на

https://kaggle.com/join/cmf_garch