

Оценка GARCH моделей в R

Подключаем пакет для оценки гарчей в R и прочие удобства:

```
library("rugarch")
library("forecast")
library("lubridate")
library("xts")
```

В пакете rugarch есть встроенный набор данных по индексу SP500. Достаём этот набор данных из недр пакета:

```
data("sp500ret")
```

Посмотрим на начало ряда:

```
head(sp500ret)
```

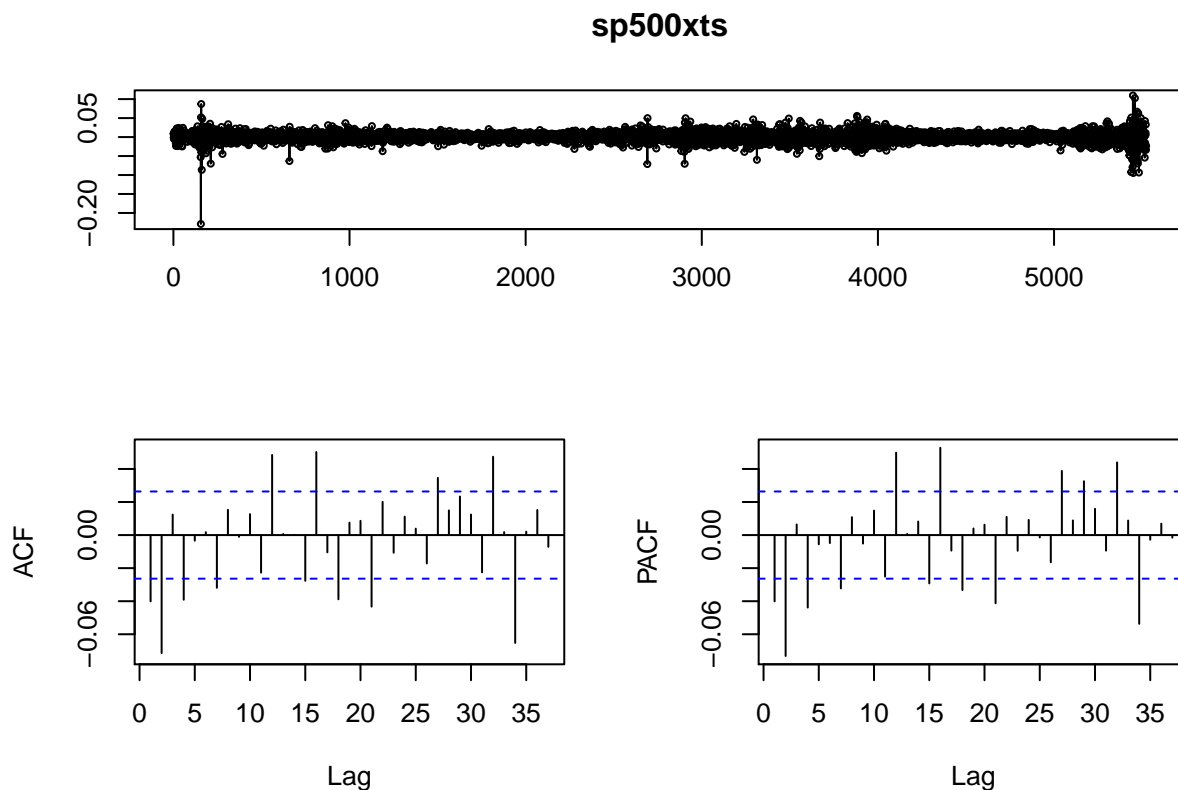
```
##           SP500RET
## 1987-03-10 0.008840447
## 1987-03-11 -0.001892734
## 1987-03-12 0.003129678
## 1987-03-13 -0.004577455
## 1987-03-16 -0.005742768
## 1987-03-17 0.014603325
```

Исходный ряд — это data.frame. Преобразуем его в xts временной ряд. Это немного другой формат хранения данных. Более удобный для временных рядов.

```
y <- sp500ret$SP500RET
t <- ymd(rownames(sp500ret))
sp500xts <- xts(y, order.by=t)
```

Построим график доходностей и ACF/PACF:

```
tsdisplay(sp500xts)
```



Пакет `rugarch` оценивает целый зоопарк GARCH моделей. Мы выберем для примера самую простую, ARMA(1,1)-GARCH(1,1) модель.

То есть уравнение для доходности имеет вид:

$$(y_t - \mu) = \phi_1(y_{t-1} - \mu) + \theta_1\varepsilon_{t-1} + \varepsilon_t$$

А уравнение для волатильности, $\varepsilon_t = \sigma_t \cdot \nu_t$, имеет вид:

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

Сначала указываем желаемую спецификацию, а затем оцениваем.

```
model <- ugarchspec(
  variance.model = list(garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(1, 1)))
model_est <- ugarchfit(spec = model,
  data=sp500xts)
```

Смотрим на результаты оценивания:

```
model_est
```

```
##
## *-----*
## *      GARCH Model Fit      *
## *-----*
##
```

```

## Conditional Variance Dynamics
## -----
## GARCH Model : sGARCH(1,1)
## Mean Model : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## -----
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.000522  0.000087  5.9878 0.00000
## ar1      0.871332  0.071687 12.1547 0.00000
## ma1     -0.898444  0.064089 -14.0188 0.00000
## omega    0.000001  0.000001  1.3891 0.16481
## alpha1   0.087715  0.013720  6.3930 0.00000
## beta1    0.904954  0.013767 65.7328 0.00000
##
## Robust Standard Errors:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.000522  0.000130  4.014118 0.000060
## ar1      0.871332  0.087565  9.950649 0.000000
## ma1     -0.898444  0.079625 -11.283381 0.000000
## omega    0.000001  0.000014  0.093146 0.925788
## alpha1   0.087715  0.186997  0.469074 0.639016
## beta1    0.904954  0.192511  4.700801 0.000003
##
## LogLikelihood : 17902.41
##
## Information Criteria
## -----
##
## Akaike      -6.4807
## Bayes       -6.4735
## Shibata     -6.4807
## Hannan-Quinn -6.4782
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##              statistic p-value
## Lag[1]              5.523 1.876e-02
## Lag[2*(p+q)+(p+q)-1][5] 6.409 1.422e-05
## Lag[4*(p+q)+(p+q)-1][9] 7.162 1.132e-01
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##              statistic p-value
## Lag[1]              1.103 0.2936
## Lag[2*(p+q)+(p+q)-1][5] 1.495 0.7410
## Lag[4*(p+q)+(p+q)-1][9] 1.955 0.9104
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----

```

```

##          Statistic Shape Scale P-Value
## ARCH Lag[3]  0.01958 0.500 2.000 0.8887
## ARCH Lag[5]  0.17499 1.440 1.667 0.9713
## ARCH Lag[7]  0.53740 2.315 1.543 0.9749
##
## Nyblom stability test
## -----
## Joint Statistic: 174.7772
## Individual Statistics:
## mu      0.2104
## ar1     0.1471
## ma1     0.1047
## omega   21.3752
## alpha1  0.1345
## beta1   0.1128
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.49 1.68 2.12
## Individual Statistic:  0.35 0.47 0.75
##
## Sign Bias Test
## -----
##          t-value      prob sig
## Sign Bias      0.4302 6.670e-01
## Negative Sign Bias 2.9467 3.226e-03 ***
## Positive Sign Bias 2.3928 1.675e-02 **
## Joint Effect     28.9717 2.270e-06 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##   group statistic p-value(g-1)
## 1   20    179.0   4.775e-28
## 2   30    187.8   3.641e-25
## 3   40    218.4   8.218e-27
## 4   50    228.1   6.432e-25
##
##
## Elapsed time : 0.8314869

```

Можно на каждый из результатов глянуть отдельно. Например, информационные критерии:

```
infocriteria(model_est)
```

```

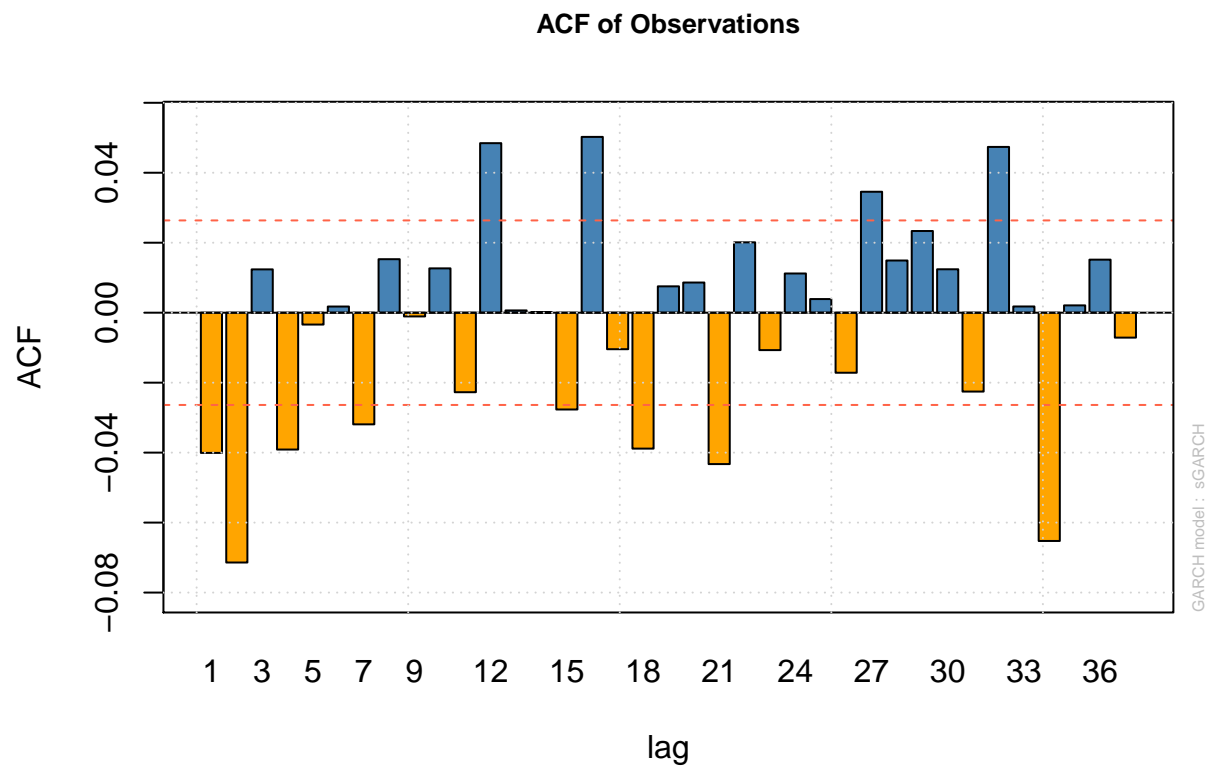
##
## Akaike      -6.480683
## Bayes      -6.473495
## Shibata    -6.480686
## Hannan-Quinn -6.478177

```

Можно построить кучу заготовленных графиков. Мы построим два, и настоятельно советуем попробовать `plot(model_est)`:

График автокорреляционной функции самого ряда:

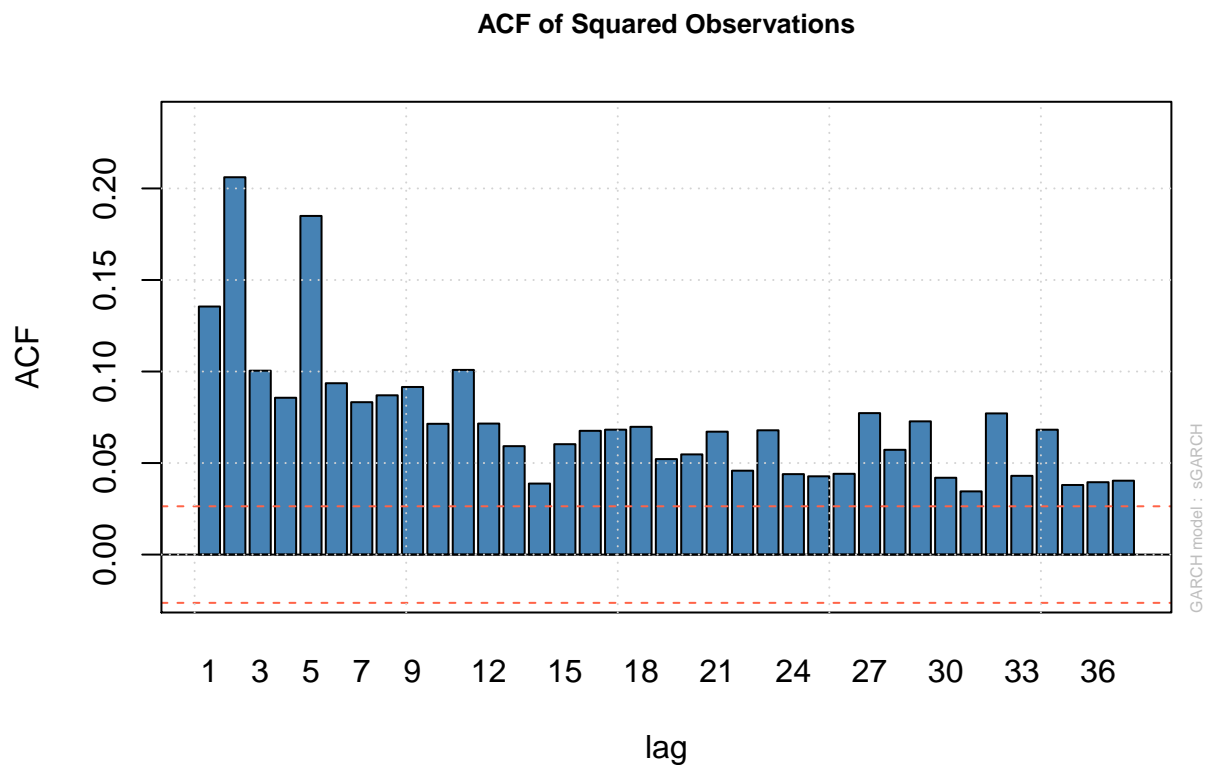
```
plot(model_est, which=4)
```



Есть и положительные, и отрицательные корреляции, отчасти значимые.

График автокорреляционной функции для квадратов доходностей:

```
plot(model_est, which=5)
```



Все

автокорреляции положительны, первые автокорреляции квадратов значимы.

Прогнозируем на 10 шагов вперёд:

```
prognos <- ugarchforecast(model_est)
prognos
```

```
##
## *-----*
## *      GARCH Model Forecast      *
## *-----*
## Model: sGARCH
## Horizon: 10
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=2009-01-30]:
##      Series  Sigma
## T+1  0.0016642 0.02480
## T+2  0.0015172 0.02473
## T+3  0.0013892 0.02467
## T+4  0.0012775 0.02460
## T+5  0.0011803 0.02454
## T+6  0.0010956 0.02448
## T+7  0.0010217 0.02442
## T+8  0.0009574 0.02435
## T+9  0.0009013 0.02429
## T+10 0.0008525 0.02423
```