

GARCH modelis AstraZeneca (AZN) akcijos grąžai

AstraZeneca yra viena iš pirmaujančių farmacijos kompanijų pasaulyje, kurios bendra vertė yra \$204 mlrd., o 2023m. įmonė pasiekė \$45,8 mlrd. pelno. Šio projekto tikslas – rasti tinkamiausią GARCH modelį AstraZeneca akcijos grąžai, pasitelkiant R programinę įrangą. Projektas gimė iš smalsumo, savarankiškai pritaikyti universitete įgautas žinias. Įmonės pasirinkimą nulėmė tai, kad mane domina farmacijos pramonė, kuri ne tik kuria žmoniją gelbstinčius vaistus, bet yra ir aršių diskusijų subjektas.

Pasitelkęs 10 metų duomenis, bandysiu sudaryti geriausią GARCH modelį. Šiame dokumente pateiksiu proceso eigą, komentarus apie savo mintis, sutiktus sunkumus. Tačiau, nepristatysiu itin techninių modelio aspektų, apsiribosiu tik įdomiausiais ar būtiniausiais punktais. R kodą čia pateiksiu nuotraukose, o GitHub platformoje kodą galite rasti .r formato dokumente.

Paketai ir duomenys

Projekte panaudoti paketai:

```
library(quantmod)
library(rugarch)
library(Hmisc)
library(forecast)
library(lmtest)
```

Quantmod paketas suteikia galimybę iš gauti duomenis apie įmonės akcijos kainą, grąžą ir prekybos apimtį tiesiai iš Yahoo Finance portalą. Gautai duomenų lentelėi priskyriau pavadinimą AZN. Iš lentelės ištraukęs akcijos grąžos stulpelį, jį panaudojau grąžos procentiniam pokyčiui apskaičiuoti. Procentinis pokytis yra tinkamesnis modeliavimui dėl stacionarumo. Kintamąjį transformavau į numeric formą ir pašalinau pirmąjį stebėjimą, nes jis buvo tuscias.

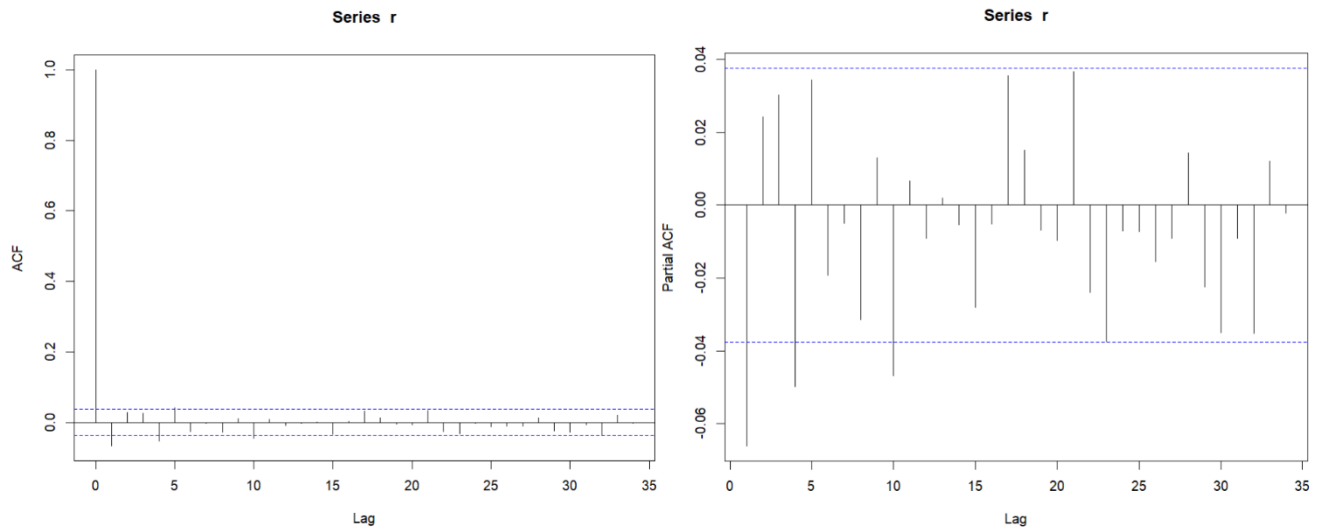
```
AZN<-getSymbols("AZN", from = '2014-01-01', to = '2024-10-23', auto.assign = FALSE)
r<-diff(log(AZN$AZN.Adjusted))
r<-as.numeric(r[-1])
```

Čia galite matyti ištrauką iš AZN lentelės.

| | AZN.Open | AZN.High | AZN.Low | AZN.Close | AZN.Volume | AZN.Adjusted |
|------------|----------|----------|---------|-----------|------------|--------------|
| 2014-01-02 | 29.285 | 29.300 | 29.175 | 29.285 | 2026000 | 20.25181 |
| 2014-01-03 | 29.405 | 29.550 | 29.385 | 29.430 | 1841400 | 20.35209 |
| 2014-01-06 | 29.450 | 29.585 | 29.380 | 29.445 | 1931200 | 20.36246 |
| 2014-01-07 | 29.220 | 29.320 | 29.145 | 29.255 | 2772200 | 20.23107 |
| 2014-01-08 | 29.210 | 29.420 | 29.165 | 29.260 | 4639400 | 20.23452 |
| 2014-01-09 | 29.520 | 29.710 | 29.345 | 29.685 | 3495200 | 20.52843 |
| 2014-01-10 | 29.895 | 30.295 | 29.890 | 30.240 | 5064400 | 20.91224 |
| 2014-01-13 | 29.910 | 30.095 | 29.780 | 29.810 | 3226000 | 20.61488 |
| 2014-01-14 | 30.615 | 31.175 | 30.575 | 31.085 | 8026600 | 21.49659 |
| 2014-01-15 | 31.290 | 31.520 | 31.245 | 31.475 | 4957400 | 21.76629 |
| 2014-01-16 | 31.570 | 31.810 | 31.530 | 31.795 | 4055600 | 21.98758 |
| 2014-01-17 | 31.860 | 31.885 | 31.690 | 31.865 | 4204600 | 22.03599 |

Duomenų analizė

Pasitelkiau tiesinę (ACF) ir dalinę (PACF) autokorelogramas, kad būtų galima susidaryti vaizdą apie tiriamą kintamąjį. Matoma, kad egzistuoja koreliacija tarp kintamojo ir jo pirmojo, ketvirtojo ir kitų vėlavimų.



Tuomet sudariau paprastą ARMA(1,1) modelį.

```
arma1<-Arima(r, order=c(1, 0, 1))
coeftest(arma1)
```

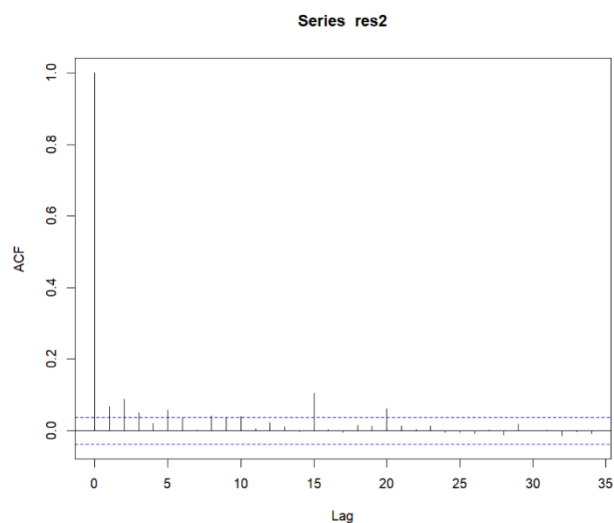
Nei slankiųjų vidurkių, nei autoregresinis vėlavimas nėra statistiškai reikšmingas. Žinoma, šiuo atveju vėlavimų derėtų įtraukti daugiau, bet mus domina ne ARMA, o GARCH modelis.

| | Estimate | Std. Error | z value | Pr(> z) |
|-----------|-------------|------------|---------|----------|
| ar1 | -0.21905296 | 0.16695584 | -1.3120 | 0.1895 |
| ma1 | 0.15250610 | 0.16837188 | 0.9058 | 0.3651 |
| intercept | 0.00049181 | 0.00028389 | 1.7324 | 0.0832 |

Todėl sekančiame žingsnyje siekiau sužinoti ar egzistuoja sąlyginis heteroskedastiškumas. Apskaičiavau ARMA modelio paklaidų kvadratus ir tas reikšmes priskyriau res2 kintamajam.

```
res2<-arma1$residual^2
acf(res2)
```

res2 vizualizavau naudojant tiesinę autokorelogramą. Matoma daug paklaidų autokoreliacijos, o tai yra sąlyginio heteroskedastiškumo signalas.



Svarbu būti objektyviam, todėl sąlyginio heteroskedastiškumo egzistavimui nusprendžiau panaudoti formalų Engle testą.

```
engle<-lm(res2~Lag(res2, 1)+ Lag(res2, 2)+ Lag(res2, 3))
summary(engle)$r.squared*length(res2)#testo reiksme
qchisq(0.95, 3)#kritine reiksme
```

Engle testo nulinė hipotezė teigia, jog modeliuojamame procese sąlyginis heteroskedastiškumas neegzistuoja. Testo nulinė hipotezę atmesti galima tuomet, kai jo vertė yra didesnė už Chi-Kvadratu kritinę reikšmę. Žemiau galite matyti, kad testo vertė yra didesnė, todėl galima teigti, kad sąlyginis heteroskedastiškumas egzistuoja.

```
> summary(engle)$r.squared*length(res2)#testo reiksme
[1] 35.11243
> qchisq(0.95, 3)#kritine reiksme
[1] 7.814728
```

GARCH modelių sudarymas

Patvirtinus sąlyginio heteroskedastiškumo egzistavimą, buvo galima pradėti sudarinėti GARCH modelius. Pradėjau nuo standartinio GARCH modelio.

Svarbu paminėti – šiame projekte svarbiausiu modelio tinkamumo kriterijumi laikiau informacinį Akaike(AIC) kriterijų. Akaike kriterijų naudoju kiekvieno modelio sudarymo metu, siekiant nustatyti kiek vėlavimų įtraukti į GARCH ir ARMA dalis. Projekto pabaigoje AIC naudoju palyginant visus sudarytus modelius. Geriausias modelis tas, kurio AIC reikšmė yra mažiausia.

```
azn.garch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "sGARCH",
                                           submodel="GARCH"),
                      mean.model=list(armaOrder=c(2,1)))
garch.azn<-ugarchfit(spec= azn.garch, data=r)
garch.azn
```

GARCH modelių rezultatų atsakai yra labai ilgi, todėl čia pakomentuosiu ir nuotraukose pateiksiu tik svarbiausius aspektus. Žemiau pateikiu sudaryto modelio statistikas.

| Optimal Parameters | | | | |
|--------------------|-----------|------------|---------|----------|
| | Estimate | Std. Error | t value | Pr(> t) |
| mu | 0.000438 | 0.000278 | 1.5764 | 0.114935 |
| ar1 | -0.886239 | 0.138160 | -6.4146 | 0.000000 |
| ar2 | -0.037272 | 0.022645 | -1.6459 | 0.099779 |
| ma1 | 0.867637 | 0.136478 | 6.3573 | 0.000000 |
| omega | 0.000022 | 0.000005 | 4.1826 | 0.000029 |
| alpha1 | 0.045768 | 0.009057 | 5.0533 | 0.000000 |
| beta1 | 0.863503 | 0.027617 | 31.2674 | 0.000000 |

Nyblom stability test

Joint Statistic: 0.8048

Individual Statistics:

```
mu      0.06261
ar1     0.03417
ar2     0.09496
ma1     0.03408
omega   0.07159
alpha1  0.09630
beta1   0.07354
```

Adjusted Pearson Goodness-of-Fit Test:

| | group | statistic | p-value(g-1) |
|---|-------|-----------|--------------|
| 1 | 20 | 198.6 | 6.505e-32 |
| 2 | 30 | 219.0 | 4.591e-31 |
| 3 | 40 | 228.5 | 1.237e-28 |
| 4 | 50 | 238.1 | 1.174e-26 |

Asymptotic Critical Values (10% 5% 1%)

```
Joint Statistic:      1.69 1.9 2.35
Individual Statistic: 0.35 0.47 0.75
```

Information Criteria

Akaike -5.5277
 Bayes -5.5125
 Shibata -5.5278
 Hannan-Quinn -5.5222

Weighted Ljung-Box Test on Standardized Residuals

statistic p-value
 Lag[1] 0.5032 0.4781
 Lag[2*(p+q)+(p+q)-1][8] 4.2702 0.6317
 Lag[4*(p+q)+(p+q)-1][14] 6.9685 0.5665
 d.o.f=3
 H0 : No serial correlation

Modelio parametų dalyje galima matyti, kad antrasis autoregresinis vėlavimas nėra statistiškai reikšmingas. Tačiau, jo įtraukimas pašalina serijinę koreliaciją, kuria tiria Ljung-Box testas. Nyblom stabilumo testas indikuoja, kad stebėjimuose nėra struktūrinio lūžio. Pearson Goodness-of-Fit testas rodo, jog modelio paklaidos neatitinka normaliojo skirstinio.

Toliau sudariau FGARCH modelį, kuriame nurodžiau, kad paklaidos atitiktų normalųjį skirstinį.

```
azn.fgarch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "fGARCH",
submodel="GARCH"),
, mean.model=list(armaOrder=c(2,1)),
distribution.model = "std")
fgarch.azn<-ugarchfit(spec= azn.fgarch, data=r)
fgarch.azn
```

FGARCH Modelio statistikos:

Optimal Parameters

| | Estimate | Std. Error | t value | Pr(> t) |
|--------|-----------|------------|-----------|----------|
| mu | 0.001259 | 0.000191 | 6.59728 | 0.000000 |
| ar1 | 0.876585 | 0.053355 | 16.42918 | 0.000000 |
| ar2 | -0.039405 | 0.021055 | -1.87149 | 0.061278 |
| ma1 | -0.880588 | 0.050073 | -17.58614 | 0.000000 |
| omega | 0.000004 | 0.000005 | 0.74547 | 0.455985 |
| alpha1 | 0.074570 | 0.028991 | 2.57222 | 0.010105 |
| beta1 | 0.922416 | 0.029288 | 31.49450 | 0.000000 |
| shape | 3.842276 | 0.368899 | 10.41551 | 0.000000 |

Information Criteria

Akaike -5.5145
 Bayes -5.4946
 Shibata -5.5145
 Hannan-Quinn -5.5072

Sign Bias Test

| | t-value | prob sig |
|--------------------|---------|-----------|
| Sign Bias | 0.9035 | 0.36636 |
| Negative Sign Bias | 0.3313 | 0.74048 |
| Positive Sign Bias | 1.8725 | 0.06126 * |
| Joint Effect | 3.6691 | 0.29949 |

Sign Bias testas fiksuoja neasimetrinį šokų(žinių) poveikį dispersijai. Omega koeficientas yra statistiškai nereikšmingas. Pasirinkau šiame projekte nereikšmingų koeficientų nefiksuoti(neišmesti), bet turiu pripažinti, kad egzistuoja trūkumas.

Sekantis mano sudarytas modelis buvo GARCH-M arba GARCH vidurkyje.

```
azn.garchm<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "fGARCH",
submodel="GARCH")
, mean.model=list(armaOrder=c(2,1), archm=TRUE, archpow = 1)
,distribution.model = "std")
garchm.azn<-ugarchfit(spec= azn.garchm, data=r)
garchm.azn
```

Optimal Parameters

| | Estimate | Std. Error | t value | Pr(> t) |
|--------|-----------|------------|-----------|----------|
| mu | 0.001492 | 0.000655 | 2.27706 | 0.022783 |
| ar1 | 0.878735 | 0.052824 | 16.63516 | 0.000000 |
| ar2 | -0.039337 | 0.021052 | -1.86857 | 0.061682 |
| ma1 | -0.882687 | 0.049441 | -17.85335 | 0.000000 |
| archm | -0.015936 | 0.042831 | -0.37206 | 0.709850 |
| omega | 0.000004 | 0.000005 | 0.77098 | 0.440718 |
| alpha1 | 0.074415 | 0.027948 | 2.66260 | 0.007754 |
| beta1 | 0.922457 | 0.028286 | 32.61206 | 0.000000 |
| shape | 3.847886 | 0.366257 | 10.50598 | 0.000000 |

Information Criteria

Akaike -5.5137
 Bayes -5.4913
 Shibata -5.5137
 Hannan-Quinn -5.5055

Sign Bias Test

| | t-value | prob sig |
|--------------------|---------|-----------|
| Sign Bias | 0.9327 | 0.35107 |
| Negative Sign Bias | 0.3396 | 0.73421 |
| Positive Sign Bias | 1.8592 | 0.06312 * |
| Joint Effect | 3.6066 | 0.30720 |

Archm koeficientas nėra reikšmingas. Taip pat nereikšmingas išliko omega. Šiame modelyje Sign Bias testo signalas nepasikeitė.

Tuomet sudariau IGARCH arba integruotas GARCH. Šis modelis neturėtų turėti pranašumo, nes modeliuojamas procesas, procentinis grąžos pokytis, jau yra stacionarus. Kitaip, nei praeituose modeliuose, šiame nereikėjo įtraukti du autoregresinius ir vieną slankiųjų vidurkių vėlavimą paklaidų autokoreliacijos panaikinimui.

```
azn.igarch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "iGARCH",
submodel="GARCH"),
, mean.model=list(armaOrder=c(0,0)),
distribution.model = "std")
igarch.azn<-ugarchfit(spec= azn.igarch, data=r)
igarch.azn
```

| Optimal Parameters | | | | | Information Criteria | |
|--------------------|----------|------------|---------|----------|----------------------|---------|
| | Estimate | Std. Error | t value | Pr(> t) | | |
| mu | 0.000638 | 0.000230 | 2.7668 | 0.005662 | Akaike | -5.7262 |
| omega | 0.000002 | 0.000001 | 3.1661 | 0.001545 | Bayes | -5.7175 |
| alpha1 | 0.025491 | 0.002641 | 9.6513 | 0.000000 | Shibata | -5.7262 |
| beta1 | 0.974509 | NA | NA | NA | Hannan-Quinn | -5.7230 |
| shape | 3.093705 | 0.124419 | 24.8652 | 0.000000 | | |

Sekantis modelis buvo TGARCH. Šis modelis, kaip ir du sekantys, talpina asimetrišką dispersiją.

```
azn.tgarch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "fGARCH",
submodel="TGARCH"),
, mean.model=list(armaOrder=c(2, 1)),distribution.model = "std")
tgarch.azn<-ugarchfit(spec= azn.tgarch, data=r)
tgarch.azn
```

| Optimal Parameters | | | | | Information Criteria | |
|--------------------|-----------|------------|---------|----------|----------------------|---------|
| | Estimate | Std. Error | t value | Pr(> t) | | |
| mu | 0.000539 | 0.000220 | 2.4495 | 0.014304 | Akaike | -5.7525 |
| ar1 | -0.693674 | 0.128661 | -5.3915 | 0.000000 | Bayes | -5.7330 |
| ar2 | -0.031069 | 0.018524 | -1.6772 | 0.093495 | Shibata | -5.7525 |
| ma1 | 0.680584 | 0.128246 | 5.3068 | 0.000000 | Hannan-Quinn | -5.7455 |
| omega | 0.000526 | 0.000333 | 1.5821 | 0.113619 | | |
| alpha1 | 0.054076 | 0.020760 | 2.6048 | 0.009192 | | |
| beta1 | 0.927036 | 0.037689 | 24.5972 | 0.000000 | | |
| eta11 | 0.607465 | 0.153653 | 3.9535 | 0.000077 | | |
| shape | 3.843386 | 0.282659 | 13.5973 | 0.000000 | | |

eta11 koeficientas yra susijęs su asimetriniais efektais. Faktas, kad jis yra statistiškai reikšmingas, nors mes tai jau ir žinojome, indikuoja asimetrinę dispersiją.

EGARCH modelis. ARMA dalyje parinkau tik pirmąjį slankiųjų vidurkių vėlavimą.

```
azn.egarch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "eGARCH"),
mean.model=list(armaOrder=c(0,1)),distribution.model = "std")
egarch.azn<-ugarchfit(spec= azn.egarch, data=r)
egarch.azn
```

| Optimal Parameters | | | | | Information Criteria | |
|--------------------|-----------|------------|-----------|----------|----------------------|---------|
| | Estimate | Std. Error | t value | Pr(> t) | | |
| mu | 0.000539 | 0.000199 | 2.7059 | 0.006811 | Akaike | -5.7515 |
| ma1 | -0.014078 | 0.013935 | -1.0103 | 0.312358 | Bayes | -5.7363 |
| omega | -0.228513 | 0.001921 | -118.9376 | 0.000000 | Shibata | -5.7515 |
| alpha1 | -0.055948 | 0.012448 | -4.4946 | 0.000007 | Hannan-Quinn | -5.7460 |
| beta1 | 0.972945 | 0.000223 | 4363.0251 | 0.000000 | | |
| gamma1 | 0.079164 | 0.006411 | 12.3478 | 0.000000 | | |
| shape | 3.826652 | 0.279212 | 13.7052 | 0.000000 | | |

APARCH modelis. Šiame modelyje vėl reikėjo sugražinti AR ir MA vėlavimus.

```
azn.aparch<-ugarchspec(variance.model=list(garchOrder=c(1,1), model= "apARCH")
, mean.model=list(armaOrder=c(2,1)),distribution.model = "std")
aparch.azn<-ugarchfit(spec= azn.aparch, data=r)
aparch.azn
```

Optimal Parameters

| | Estimate | Std. Error | t value | Pr(> t) |
|--------|-----------|------------|----------|----------|
| mu | 0.000512 | 0.000114 | 4.4888 | 0.000007 |
| ar1 | -0.699130 | 0.029778 | -23.4780 | 0.000000 |
| ar2 | -0.028795 | 0.003929 | -7.3294 | 0.000000 |
| ma1 | 0.686976 | 0.029249 | 23.4871 | 0.000000 |
| omega | 0.001449 | 0.000839 | 1.7269 | 0.084179 |
| alpha1 | 0.053167 | 0.006391 | 8.3188 | 0.000000 |
| beta1 | 0.934631 | 0.016385 | 57.0409 | 0.000000 |
| gamma1 | 0.665496 | 0.115836 | 5.7452 | 0.000000 |
| delta | 0.716163 | 0.085994 | 8.3280 | 0.000000 |
| shape | 3.868059 | 0.284554 | 13.5934 | 0.000000 |

Information Criteria

| | |
|--------------|---------|
| Akaike | -5.7528 |
| Bayes | -5.7311 |
| Shibata | -5.7528 |
| Hannan-Quinn | -5.7450 |

Modelių palyginimas

Taigi, įvertinus visus šiuos modelius, beliko juos palyginti. Kaip ir minėjau anksčiau, naudosis Akaike informacinį kriterijų.

```

infocriteria((garch.azn))[1]
infocriteria((garchm.azn))[1]
infocriteria((igarch.azn))[1]
infocriteria((tgarch.azn))[1]
infocriteria((egarch.azn))[1]
infocriteria((aparch.azn))[1]
> infocriteria((garch.azn))[1]
[1] -5.528067
> infocriteria((garchm.azn))[1]
[1] -5.744604
> infocriteria((igarch.azn))[1]
[1] -5.726165
> infocriteria((tgarch.azn))[1]
[1] -5.752522
> infocriteria((egarch.azn))[1]
[1] -5.751486
> infocriteria((aparch.azn))[1]
[1] -5.752814

```

Mažiausią AIC vertę turi APARCH modelis. Šiame projekte jis tampa čempionu. Įdomu tai, jog visi trys modeliai, modeliuojantys asimetrišką dispersiją, nurungė kitus modelius.

Mintys pabaigai

Šis projektas turėjo savo trūkumų. Sudarinėdamas modelius pajaučiau, kad kai kurie dalykai yra man dar sunkokai apčiuopiami. Reikėtų padirbėti su nereikšmingais koeficientais, patikrinti ar modeliai tampa tikslesni juos apribojant. Taip pat, norint patikrinti modelių paaiškinamąją galią geriau, būtų galima juos išbandyti apribojant dalį duomenų, kurie vėliau būtų panaudoti testavimui.