REMIGIJUS LAPINSKAS

EKONOMETRIJA SU KOMPUTERIU. II

Laikinės sekos

http://uosis.mif.vu.lt/~rlapinskas

Turinys

0. Įvadas

Vienas pagrindinių ekonominių duomenų analizės tikslų yra tiriamų kintamųjų prognozė. Tam vartojamus metodus galima grubiai suskirstyti į dvi grupes – regresinius ir laikinių sekų[1] (priminsime – laikine seka vadiname stebėjimų, atliekamų reguliariais laiko momentais, seką). Pirmuoju atveju sudaromas ekonominis (pavidalo $Y = f(X_1, X_2, ..., X_k)$) ir po to statistinis (pavidalo $Y = f(X_1, X_2, ..., X_k) + \varepsilon$) modeliai, surišantys atsako ir prognozinius kintamuosius (žr., pvz., http://uosis.mif.vu.lt/~rlapinskas, Ekonometrija I). Antrasis modelis vartojamas trumpalaikėms prognozėms, jis suriša mus dominančio kintamojo reikšmes su jo paties ankstesnėmis reikšmėmis. Jei tokių kintamųjų tik vienas – laikinė seka vadinama vienmate (angl. univariate time-series). Vienmatės sekos skirstomos į tam tikras klases, iš kurių šiame kurse nagrinėsime autoregresinius (AR), slenkamojo vidurkio (MA), autoregresinius integruotus slenkamojo vidurkio (ARIMA) ir kai kuriuos kitus procesus. Antra vertus, jei stebime ne vieno kintamojo evoliuciją, o kelių – laikinės sekos vadinamos daugiamatėmis (angl. multivariate time-series) – šiame kurse aptarsime vektorinį autoregresinį procesą (VAR) ir kelias jo modifikacijas.

Komputerinė laikinių sekų analizė šiame kurse bus paprastai atliekama su R 2.5.1 (arba kartais su **EViews**‘o) programa. R yra nemokamas, didžiules galimybes teikiantis produktas, kurį galima atsisiųsti iš, pvz., http://cran.at.r-project.org/. Pagrindinis jo „trūkumas“ – R yra programavimo kalba, tad jos teks mokytis. **EViews**‘as yra ekonometrinei analizei specializuota programa, su ja patogu dirbti, tačiau ją teks pirkti (galima įsigyti gana pigią studentišką versiją). MIF tinkle yra instaliuota **EViews**‘o 4.1 versija. Taip pat galima vartoti nemokamą ekonometrijai skirtą produktą **Gretl**.

Trumpai paaiškinsime, kaip instaliuojamas R. Iš http://cran.at.r-project.org/|Precompiled Binary Distributions|Windows (95 and later)|base atsisiųskite failą rw2051.exe. Su jo pagalba instaliuosite R kartu su pagrindiniais R paketais, kurių sąrašą reikėtų papildyti dar keliais, ekonometrijai skirtais. Tam reikia įjungti R, meniu eiluteje spustelėti Packages|Install package(s)..., pasirodžiusiame sąraše pasirinkti, pvz., Austria ir komandiniame lange surinkti `install. packages("ctv")`. Po to surinkti

```
library(ctv)
install.views("Econometrics")
install.views("Finance")
```

Internete ir R paketuose galima rasti daug laikinių sekų, kurias galima naudoti analizei ar įvairioms statistinėms procedūroms iliustruoti. Dalį jų rasite šiame konspekte. Kai kurias kitas galite rasti taip:

```
library(fEcofin); ?TimeSeriesData
library(help=Ecdat)
library(help=fma)
library(help=CDNmoney)
library(help=pwt)
library(help=Mcomp)
```

http://research.stlouisfed.org/fred2/
http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html
http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/

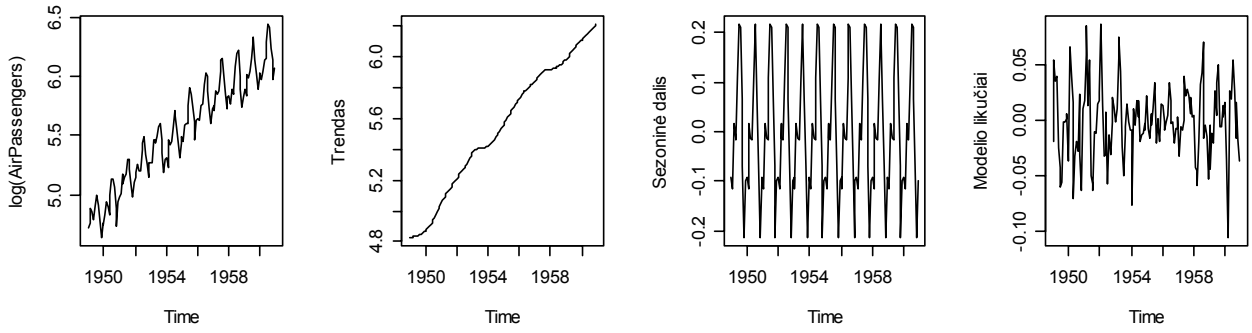
[1] Sinonimai – dinamikos seka, laiko eilutė, dinamikos eilutė, chronologinė seka (angl. time series). EViews‘as laikinę seką vadina tiesiog seka (angl. series (of observations)). Iš esmės, laikinė seka yra (diskrečiojo laiko) atsitiktinis procesas.

http://faculty.chicagogsb.edu/ruey.tsay/teaching/fts2/

ir t.t.

1. Laikinės sekos ir jų trys komponentės

1.1 pav. išbrėžti tipiškos laikinės sekos y_t [1] ir jos trijų komponenčių grafikai. Laikinės sekos paprastai turi kelias komponentes: gerai akies išskiriamą reguliariąją dalį, vadinamąjį trendą m_t, sezoninę dalį[2] s_t (žiemą keleivių, ko gero, mažiau) ir nelabai didelį atsitiktinį vadinamųjų paklaidų priedą e_t; bendruoju atveju rašysime $y_t = m_t + s_t + e_t$. Tokios struktūros[3] laikinių sekų teorijos tikslas – iš y_t išskirti šias tris dedamasias ir jas panaudoti laikinių sekų analizei, palyginimui arba prognozei.



1.1 pav. Laikinės sekos `log(AirPassengers)` ir jos trijų komponenčių (trendo, sezoninės ir paklaidų) grafikai

Jei paklaidos sudaro stacionarų procesą[4], laikoma, kad modelis sudarytas gerai (t. y., komponentės išskirtos teisingai). Todėl pirmiausiai aptarsime stacionariuosius procesus.

1.1. Stacionariosios laikinės sekos

Laikinės sekos (priminsime – laikine seka vadiname atsitiktinių stebėjimų, atliekamų reguliariais laiko momentais, seką) žymėsime simboliu Y_t, $t \in T$ (diskreti indeksų aibė T paprastai sutampa su sveikų skaičių aibe $\boldsymbol{Z}$ arba su jos poaibiu $\{t_0, t_0+1, t_0+2, \ldots\}$). Pirmiausiai aptarsime svarbias stacionariąsias atsitiktines laikines sekas (norint, kad išvados apie atsitiktinio proceso tikimybinę struktūrą pagal jo vieną trajektoriją būtų pagrįstos, būtinas proceso stacionarumas). Vaizdžiai kalbant, laikinė seka yra stacionari, jei ją valdantys tikimybiniai dėsniai nesikeičia laikui bėgant. Kalbant tiksliau, atsitiktinių dydžių rinkinys arba *atsitiktinis procesas* Y_t, $t \in T$, yra stacionarus (plačiąja prasme), jei

[1] Tai klasikinis Box'o ir Jenkins'o `airline` duomenų rinkinys (jame pateikti 1949-1960 metų mėnesiniai duomenys apie tarptautinėmis linijomis lėktuvais skraidintų keleivių skaičių; šiuos duomenis galima rasti pakete `datasets`, duomenų rinkinyje `AirPassengers`). Atkreipkite dėmesį – 1.1 pav. išbrėžtas šių dydžių logaritmų sekos grafikas. Šie duomenys taip pat nagrinėjami 1.12 užduotyje.

[2] Trendas ir sezoninė komponentė suprantamos kaip neatsitiktinės kreivės.

[3] Yra dar viena labai svarbi laikinių sekų klasė – sekos su stochastiniu trendu (žr. 1-11). Jų analizė kokybiškai skiriasi, apie jas plačiai kalbėsime 3 skyriuje.

[4] Šis skaidinys dažnai vartojamas laikinės sekos (t.y., kiekvienos jos komponentės) prognozei. Jei paklaidos sudaro paprasčiausią stacionarų procesą, būtent, yra nekoreliuotų atsitiktinių dydžių seka (t.y., baltasis triukšmas; žr. žemiau), tuomet šios komponentės geriausia prognozė yra tiesiog jos vidurkis. Jei (stacionarios) paklaidos nariai yra koreliuoti (t.y., praeitis turi įtakos ateičiai), teks išsiaiškinti šios koreliacinės struktūros pavidalą ir ją panaudoti prognozei. Ši procedūra aprašyta 2 skyriuje.

1. $m_t = EY_t \equiv a,\ t \in T$ (t.y., proceso vidurkis m_t pastovus)

2. $cov(Y_t, Y_{t+s}) = (E(Y_t - a)(Y_{t-s} - a) =)\ cov(Y_{t+h}, Y_{t+s+h}) = \gamma(s)$ koks bebūtų $h = ..., -2, -1, 0, 1, 2, ...$ (t.y., proceso (auto)kovariacinė funkcija[5] γ priklauso tik nuo atstumo tarp laiko momentų, bet ne nuo pačių momentų). Beje, iš šios sąlygos išplaukia, kad proceso dispersija $DY_t = cov(Y_t, Y_{t+0}) = \gamma(0)$ yra taip pat pastovi.

Atkreipsime dėmesį į tai, kad procesą arba laikinę seką visuomet galima traktuoti dvejopai: 1) kaip *vieną* konkrečių stebėjimų (skaičių) seką (ji vadinama (atsitiktinio) proceso Y_t realizacija arba trajektorija ir žymima mažąja raide y_t: $y_t = Y_t(\omega)$) arba 2) kaip atsitiktinį procesą, t.y., jo *realizacijų* (jų pasirodymo šansus valdo proceso tikimybinis skirstinys) *šeimą* $Y_t = \{y_t\}$. Štai keli stacionariųjų procesų pavyzdžiai.

Baltasis triukšmas. Procesas $Y_t,\ t \in T,$ vadinamas baltuoju triukšmu (angl. white noise), jei

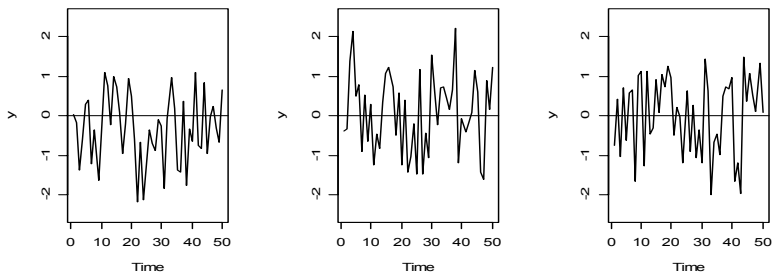
$$\text{i1) } EY_t \equiv 0 \text{ ir i2) } cov(Y_t, Y_s) = \begin{cases} \sigma^2, & jei\ t = s, \\ 0, & jei\ t \neq s. \end{cases}$$

1.1 UŽDUOTIS. Įrodykite, kad baltasis triukšmas yra stacionarusis procesas. Užrašykite jo autokovariacinę funkciją γ ir autokoreliacinę funkciją ρ. ◀◀

Pateiksime trumpesnį baltojo triukšmo apibrėžimą – tai nekoreliuotų nulinio vidurkio ir pastovios dispersijos atsitiktinių dydžių seka. Pažymėsime, kad pateiktame apibrėžime niekur neminimas Y_t skirstinys, tačiau dažnai tariama, kad jis normalusis. Paprastai baltasis triukšmas suprantamas kaip „visiško chaoso“ modelis.

Išbrėšime tris (normaliojo) baltojo triukšmo trajektorijas $y_t^{(1)}, y_t^{(2)}, y_t^{(3)}$ (neįmanoma išbrėžti visų trajektorijų (jų be galo daug), todėl apsiribosime keliomis).

```
set.seed(10)
opar=par(mfrow=c(1,3))
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)
par(opar)
```

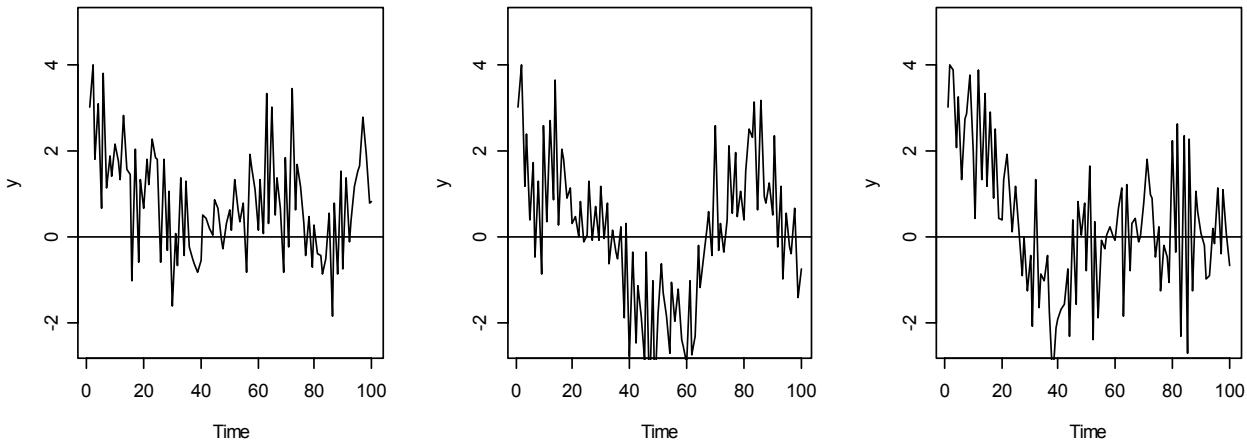


1.2 pav. Trys (normaliojo) baltojo triukšmo trajektorijos $y_t^{(1)}, y_t^{(2)}, y_t^{(3)}$

[5] Kovariacinė funkcija nusako ryšio tarp gretimų proceso reikšmių stiprumą (pobūdį).

Baltojo triukšmo procesą ateityje žymėsime simboliu W_t (pagal *W*hite noise), o jo trajektorijas (realizacijas) - w_t.

Autoregresinis procesas AR(*p*). Procesas $Y_t = \alpha_1 Y_{t-1} + W_t$ vadinamas pirmosios eilės autoregresiniu procesu ir žymimas simboliu AR(1), o jį apibendrinantis procesas $Y_t = \alpha_1 Y_{t-1} + ... + \alpha_p Y_{t-p} + W_t$ yra žymimas simboliu AR(*p*) ir vadinamas *p*-tosios eilės autoregresiniu procesu. Šį kartą Y_t priklauso ir nuo ankstesnių proceso reikšmių, tačiau sekančiame skyriuje įrodysime, kad, nežiūrint to, šis procesas (jei koeficientai α_i tenkina tam tikras sąlygas) vis dėlto yra stacionarus. Pavyzdžiui, galima įrodyti, kad, egzistuoja stacionarus proceso $Y_t = 0,0897 Y_{t-1} + 0.6858 Y_{t-2} + W_t$, $t = ..., -1, 0, 1, ...$ variantas. Antra vertus, procesas $Y_t = 0,0897 Y_{t-1} + 0.6858 Y_{t-2} + W_t$, $t = 1, 2, ...$, su pradinėmis sąlygomis $Y_1 = 3,\ Y_2 = 4$ nėra stacionarus, tačiau jis gana greitai (nuo maždaug $t = 10$ (plg. 1.3 pav.)) tampa



1.3 pav. Trys aukščiau aptarto AR(2) proceso su pradinėmis sąlygomis $Y_1 = 3,\ Y_2 = 4$ trajektorijos $y_t^{(1)}, y_t^{(2)}, y_t^{(3)}$; po $t = 10$ procesas tampa „praktiškai stacionarus“ (t.y., proceso vidurkis ir dispersija (beveik) nepriklauso nuo *t*, kai *t*>10; antra vertus, trajektorijos gali elgtis gana nereguliariai)

„praktiškai stacionarus“[6]. Beje, matome, kad net tuomet, kai $t > 10$, trajektorijos nėra panašios į „labai gero“ proceso trajektorijas, todėl, norint pagal vieną trajektoriją spręsti apie pačio proceso tikimybines savybes (pvz., stacionarus ar ne?), mums reikės formalių kriterijų.

Vienas tokių kriterijų remiasi laikinių sekų (empirine) autokoreliacine funkcija (angl. acf) *r*. Ji apibrėžiama taip: jei stacionarų procesą y_t stebime laiko momentais 1, 2, ..., *n*, $c(s) = \frac{1}{n} \sum_{i=\max(1,-s)}^{\min(n-s,n)} (y_{i+s} - \bar{y})(y_i - \bar{y})$, tai $r(s) = c(s)/c(0)$ (čia funkcija *c* yra ne kas kita, bet empirinis proceso autokovariacinės funkcijos γ analogas). Aišku, kad tuomet, kai stebime baltąjį triukšmą, o stebėjimų daug, pagal didžiųjų skaičių dėsnį

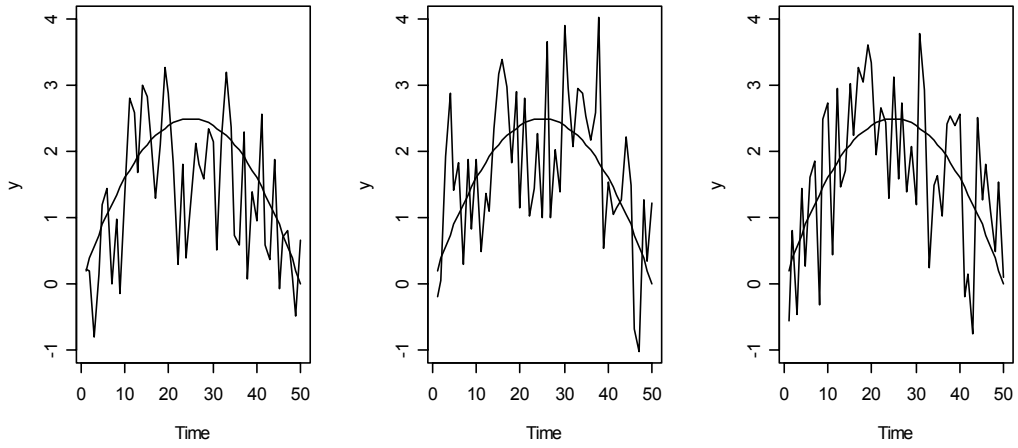
[6] Galima įrodyti, kad šio atsitiktinio proceso vidurkis ir dispersija gana greitai artėja į savo ribas (nuo $t \approx 10$ proceso vidurkis ir dispersija praktiškai pastovūs).

$$r(s) \approx \rho(s) = \begin{cases} 1, & s = 0, \\ 0, & s \neq 0. \end{cases}$$

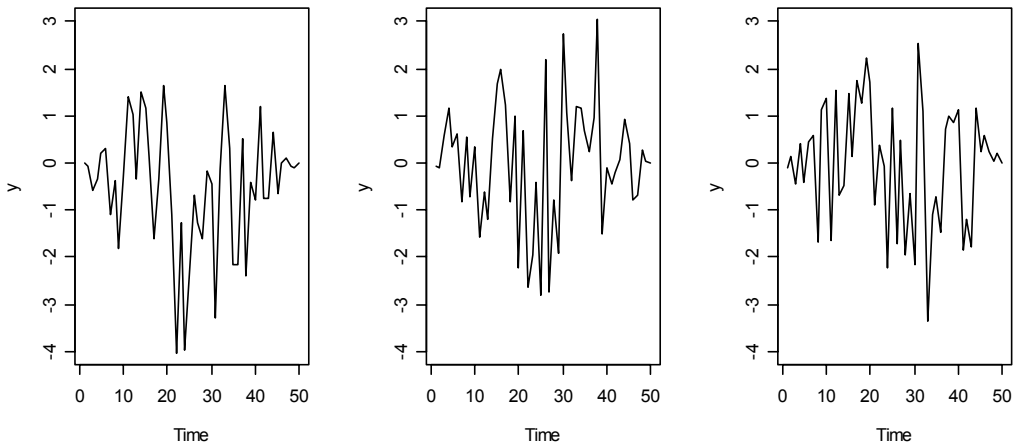
Taigi, jei funkcija *r* smarkai skiriasi nuo tik ką apibrėžtos (žr. 1.8 pav. - jame „didelį“ funkcijos *r* reikšmių nuokrypį nuo 0 žymi mėlyna trūki linija), stebimoji laikinė seka matyt nėra baltasis triukšmas.

Iki šiol kalbėjome apie stacionariuosius procesus. Dauguma ekonominių eilučių (BVP, kainų indeksas ir pan.) yra nestacionarios. Tokių laikinių sekų pavyzdžių nėra sunku sugalvoti ir patiems – pvz., jei proceso vidurkis ar dispersija priklauso nuo *t*, procesas bus nestacionarus.

1.2 UŽDUOTIS. Parašykite programą, skirtą brėžti grafikus, panašius į pateiktus 1.4 pav. ir 1.5 pav.



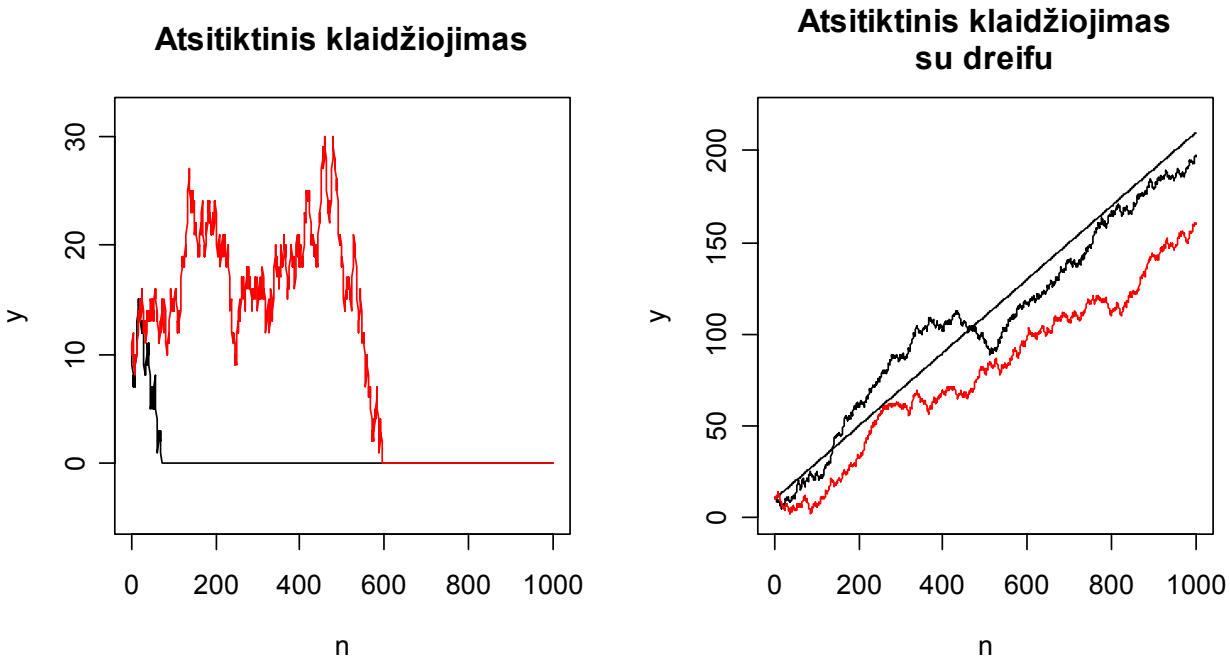
1.4 pav. Laikinės sekos su paraboliškai kintančiu vidurkiu trys trajektorijos (vėliau šį kintantį vidurkį (proceso „centro“ kitimo tendenciją) pavadinsime trendu



1.5 pav. Trys laikinės sekos (su paraboliškai kintančiu standartu) trajektorijos

1.3 UŽDUOTIS. Lošėjo pradinis kapitalas lygus $y_0 = 10$ lt, kiekvieno lošimo metu jis su tikimybe 0,5 laimi arba pralaimi 1 lt. Nesunku įsitikinti, kad jo kapitalas laiko momentu *t* yra užrašomas

formule $Y_t = y_0 + \sum_{i=1}^{t} \varepsilon_t$; čia ε_t yra jo laimėjimas momentu t (procesas Y_t vadinamas atsitiktiniu klaidžiojimu). Šį procesą galima apibendrinti, tarus, kad kiekvieno lošimo metu žaidėjas iš kasos gauna a=20 ct premiją; užrašykite ir šio proceso lygtį (tai vadinamasis *atsitiktinio klaidžiojimo su dreifu a* (angl. drift) procesas). Įrodykite, kad abu procesai yra nestacionarūs. Išbrėžkite dvi kiekvieno proceso realizacijas. ◀◀



1.6 pav. Du atsitiktinio klaidžiojimo iki bankroto grafikai (kairėje) ir du atsitiktinio klaidžiojimo su dreifu grafikai (dešinėje; „juoda“ tiesė yra proceso vidurkio y=10+0.2n grafikas)

Jau minėjome, kad, apskritai kalbant, reikėtų skirtingai žymėti atsitiktinį procesą ir jo realizacijas; vis dėlto, ateityje vadovausimės visuotinai priimta praktika ir dažniausiai abu šiuos objektus žymėsime mažosiomis raidėmis. Tiesa, esant reikalui, grįšime prie mūsų susitarimo.

1.2. Laikinės sekos ir R

R kalba laikinė seka yra užrašoma kaip vektorius su keliais papildomais požymiais (angl. attributes). Pasižymėkite žemiau pateiktą vektorių `ship1` ir su Copy+Paste perkelkite jį į R:

```
shipm1 = c(42523, 46029, 47485, 46692, 46479, 48513, 42316, 45717, 48208, 47761,
47807, 47772, 46020, 49516, 50905, 50226, 50678, 53124, 47252, 47522, 52612,
53800, 52019, 49705, 48864, 53281, 54668, 53740, 53346, 56421, 49603, 52326,
56724, 57257, 54335, 52095, 49714, 53919, 54750, 53190, 53791, 56790, 49703,
51976, 55427, 53458, 50711, 50874, 49931, 55236, 57168, 56257, 56568, 60148,
51856, 54585, 58468, 58182, 57365, 55241, 54963, 59775, 62049, 61767, 61772,
64867, 56032, 61044, 66672, 66557, 65831, 62869, 63112, 69557, 72101, 71172,
71644, 75431, 66602, 70112, 74499, 76404, 75505, 70639, 71248, 78072, 81391,
80823, 82391, 86527, 77487, 83347, 88949, 89892, 85144, 75406)
```

Vektorius `shipm1` yra Value of shipments, in millions of dollars, monthly from January, 1967 to December, 1974. This represents manufacturers' receipts, billings, or the value of products shipped,

less discounts, and allowances, and excluding freight charges and excise taxes. Shipments by foreign subsidiaries are excluded, but shipments to a foreign subsidiary by a domestic firm are included. Šaltinis: U. S. Bureau of the Census, Manufacturer's Shipments, Inventories and Orders.

Nurodžius matavimų pradžią ir dažnį[7], šį vektorių galima paversti mėnesine laikine seka.

```
shipm = ts(shipm1, start=1967, freq=12)
```

```
> shipm
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1967 42523 46029 47485 46692 46479 48513 42316 45717 48208 47761 47807 47772
1968 46020 49516 50905 50226 50678 53124 47252 47522 52612 53800 52019 49705
.....................................................................................
1973 63112 69557 72101 71172 71644 75431 66602 70112 74499 76404 75505 70639
1974 71248 78072 81391 80823 82391 86527 77487 83347 88949 89892 85144 75406
```

`shipm` dabar turi laikinės sekos struktūrą:

```
> class(shipm)
[1] "ts"
```

`shipm` išraiška R kodu yra tokia:

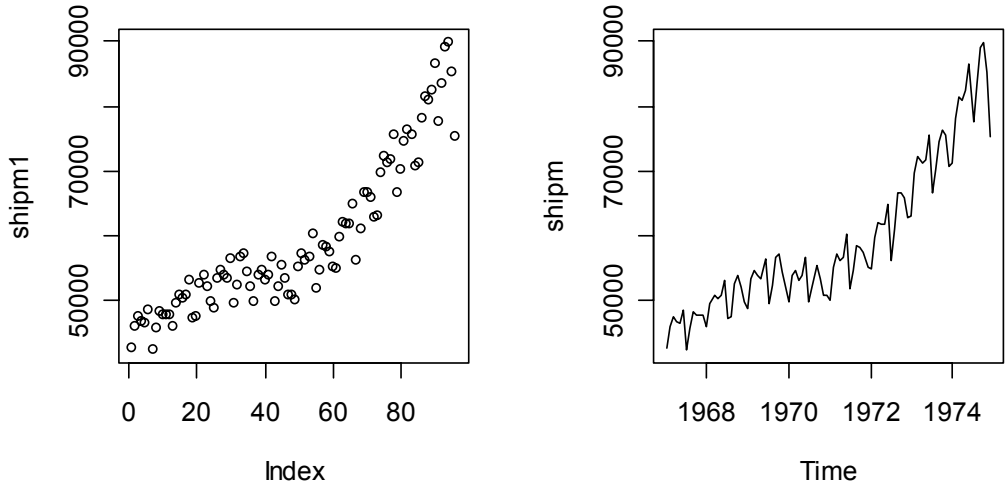
```
> dput(shipm) # Komanda dput pateikia R objekto shipm struktūrą
structure(c(42523, 46029, 47485, 46692, 46479, 48513, 42316, 45717, 48208, 47761, 47807,
47772, 46020, 49516, 50905, 50226, 50678, 53124, 47252, 47522, 52612, 53800, 52019,
49705, 48864, 53281, 54668, 53740, 53346, 56421, 49603, 52326, 56724, 57257, 54335,
52095, 49714, 53919, 54750, 53190, 53791, 56790, 49703, 51976, 55427, 53458, 50711,
50874, 49931, 55236, 57168, 56257, 56568, 60148, 51856, 54585, 58468, 58182, 57365,
55241, 54963, 59775, 62049, 61767, 61772, 64867, 56032, 61044, 66672, 66557, 65831,
62869, 63112, 69557, 72101, 71172, 71644, 75431, 66602, 70112, 74499, 76404, 75505,
70639, 71248, 78072, 81391, 80823, 82391, 86527, 77487, 83347, 88949, 89892, 85144,
75406), .Tsp = c(1967, 1974.91666666667, 12), class = "ts")
```

```
> tsp(shipm)
[1] 1967.000 1974.917   12.000
> start(shipm)
[1] 1967    1
> end(shipm)
[1] 1974   12
> frequency(shipm)
[1] 12
> attributes(shipm)
$tsp
[1] 1967.000 1974.917   12.000
$class
[1] "ts"
```

Su funkcija `plot` galime išbrėžti `shipm1` ir `shipm` grafikus – atkreipkite dėmesį į tai, kad grafiko išvaizda priklauso nuo objekto klasės.

```
opar=par(mfrow=c(1,2))
plot(shipm1) # Funkcijos plot elgesys priklauso nuo objekto klasės
plot(shipm)
par(opar)
```

[7] Jei `freq=12`, R duomenis interpretuos kaip mėnesinius, o jei `freq=4` - kaip ketvirtinius.



1.7 pav. Vektoriaus `shipm1` ir laikinės sekos `shipm` grafikai

Eviews

Norint sukurti laikinę seką `ship` EViews'e, R objektą `shipm` eksportuosime į tekstinį failą ship.txt.

R: `write(shipm, file = "ship.txt")` – autoriaus komputeryje failas ship.txt bus C:\Program Files\R\ts direktorijoje.

EViews: sukursime projektą: File|New|Workfile...|Monthly|Range 1967.01 1974.12|OK, po to File|Import|Read Text-... ir nuvairuosime į C:\Program Files\R\ts\ship.txt; ASCII Text Import lentelėje Name for series... įrašykite ship, Data order pakeiskite į in Rows, skyrelyje Rectangular File Layout – ištrinkite paukščiuką ir paspauskite OK – projekte atsiras naujas objektas `ship`. Projektą išsaugokite vardu `shipment`.

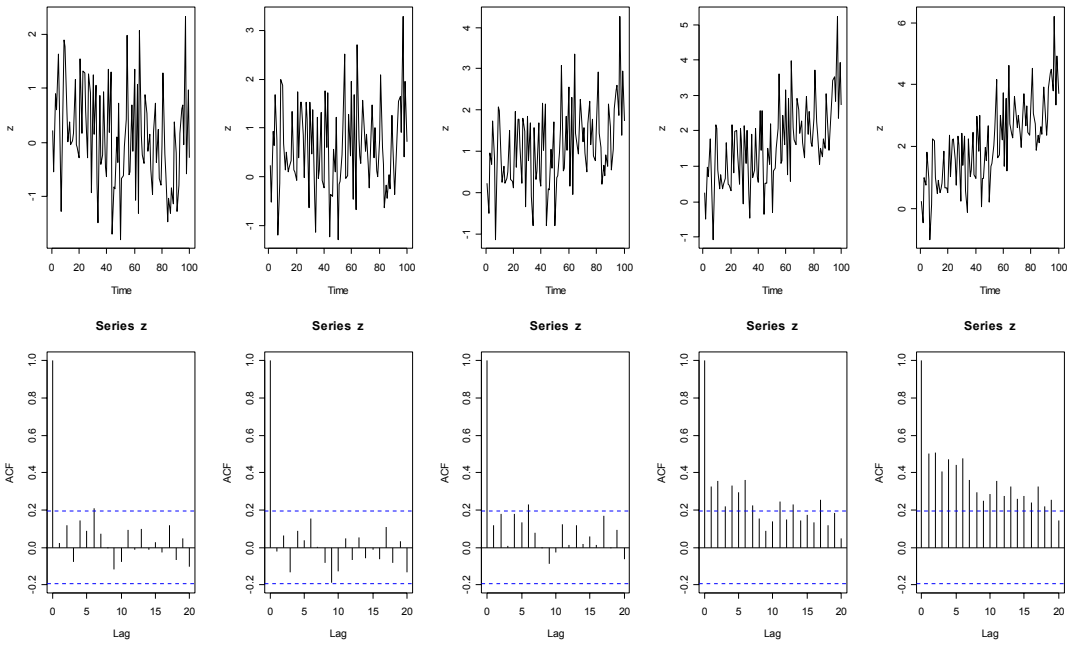
1.3. Trys laikinių sekų komponentės

Lengva matyti, kad laikinė seka (tiksliau kalbant – laikinės sekos realizacija) `shipm` yra neblogai aprašoma vadinamuoju *adityviuoju* modeliu $y_t = m_t + s_t + e_t$. Be jo dar nagrinėjami ir *multiplikatyvieji* modeliai, kuriuos galime apibrėžti dviem būdais: **1**) $y_t = m_t \cdot s_t \cdot e_t$, $s_t > 0$, $e_t > 0$, (arba, naudojant [L] žymenis, $S_t = m_t \cdot s_t \cdot e^{Z_t}$), ir **2**) $y_t = m_t \cdot s_t + e_t$ (pažymėsime, kad pirmąjį iš jų galima paversti adityviuoju jį išlogaritmavus, t.y. perėjus prie modelio $\log y_t = \log m_t + \log s_t + \log e_t$). Multiplikatyvusis modelis vartojamas tuomet, kai y_t sezoninių svyravimų amplitudė didėja kartu su y'ko reikšmėmis. Antra vertus, visa regresinių modelių teorija skirta adityviems modeliams, todėl su jais dirbti lengviau.

1.4 UŽDUOTIS. Sumodeliuokite ir išbrėžkite po du **1**) ir **2**) variantų grafikus. Kaip pasikeičia grafikai, perėjus prie logaritmų?

Nustatyti, ar laikinė seka turi (nelygų konstantai) trendą galima pagal laikinės sekos ir jos acf grafikus. Paprasčiausią pavyzdį galima gauti su tokia programa:

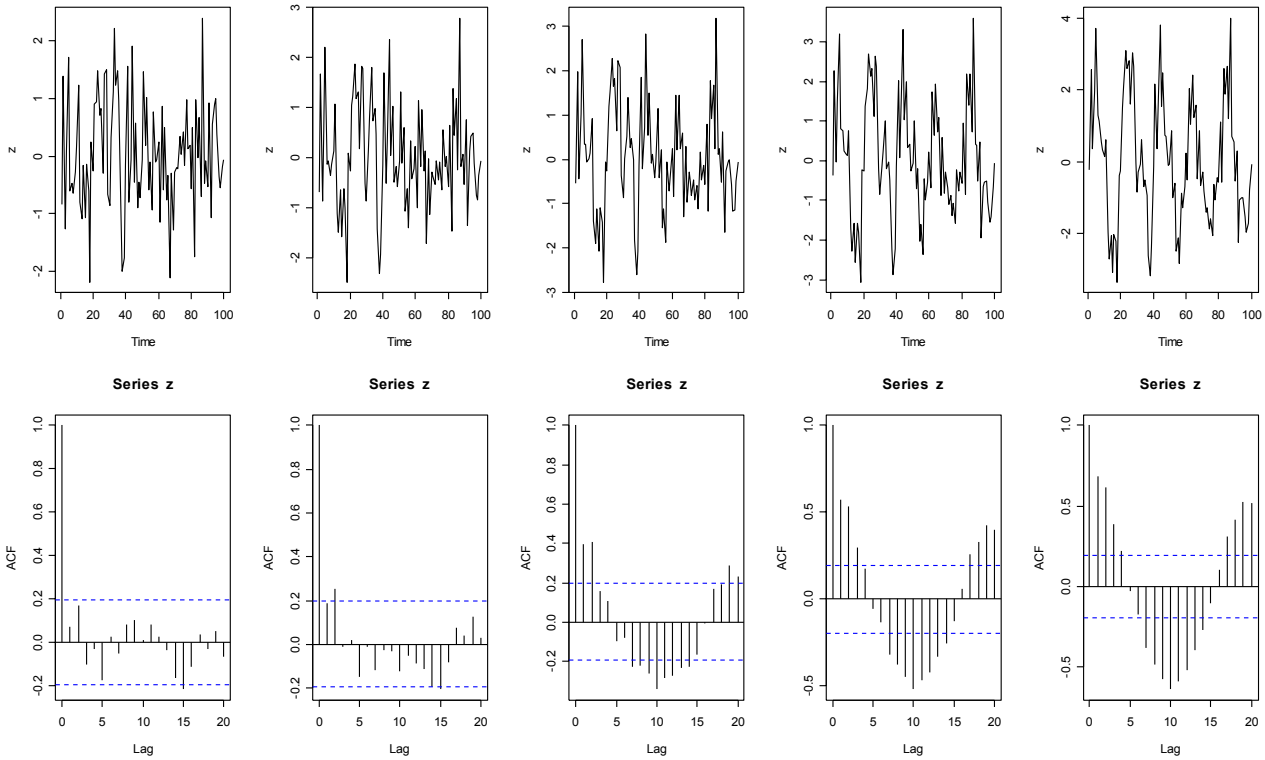
```
set.seed(4)
x=1:100
y=rnorm(100);opar=par(mfcol=c(2,5))
for (i in 1:5)
{
z=ts(0.01*(i-1)*x+y) #  Generuojame laikines sekas su vis ryškesniu trendu
                     #  (tiksliau kalbant, su vis didesniu tiesės krypties
                     #  koeficientu)
plot(z)
acf(z)
}
par(opar)
```



1.8 pav. Čia išbrėžtos laikinės sekos su tiesiškai augančiu vidurkiu ir pastovia dispersija; ryškėjantį trendą (atkreipkite dėmesį į z ašies reikšmes) rodo tiek pačios sekos tiek ir jos acf grafikai (lėtai mažėjanti acf (kartu su įžiūrimu sekos trendu) yra nestacionarumo požymis)

Sezoninę laikinės sekos dalį irgi galima įžiūrėti pagal pačios laikinės sekos ir/arba jos acf grafikus. Štai paprasčiausias variantas.

```
set.seed(5)
x=1:100
y=rnorm(100)
opar=par(mfcol=c(2,5))
for (i in 1:5)
{
z=ts(0.5*(i-1)*sin(2*pi*x/20)+y) # Periodinės dalies amplitudė vis didėja
plot(z)
acf(z)
}
par(opar)
```



1.9 pav. Sezoninė dalis sekos grafike vis ryškėja, tačiau sezoniškumui nustatyti patogesnė yra acf

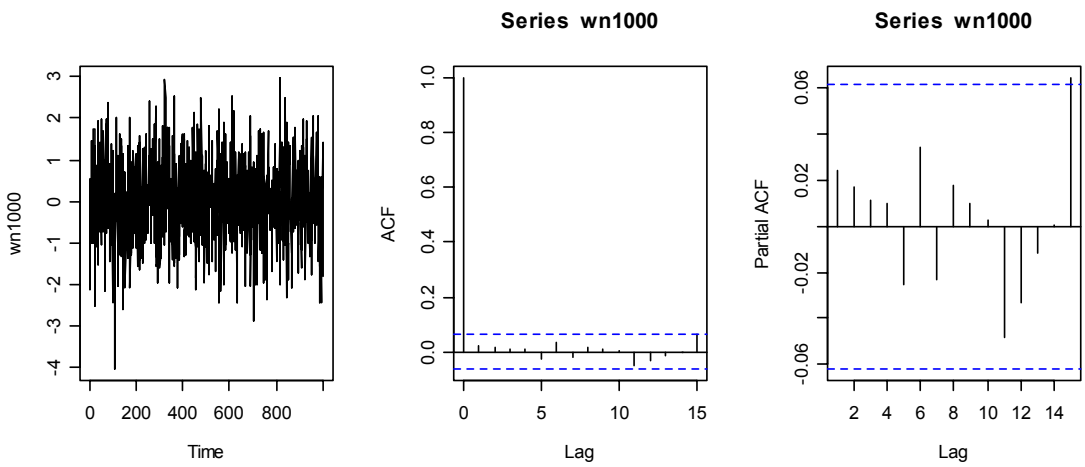
Taisyklė

Norint patikrinti, ar seka sudaro baltąjį triukšmą, reikia išbrėžti 1) jos grafiką (jame neturi būti trendo ar sezoniškumo pėdsakų), 2) sekos autokoreliacinės funkcijos `acf` grafiką (visos autokoreliacinės funkcijos reikšmės, išskyrus nulinę, turi būti melsvos juostos viduje) ir 3) dalinės autokoreliacinės funkcijos `pacf` grafiką (visos autokoreliacinės funkcijos reikšmės, pradedant pirmąja, turi būti melsvos juostos viduje) (`pacf` apibrėžimas pateiktas 2-7 psl., o pirmas grafikas – 1.10 pav.)

1.1 pavyzdys. Panagrinėkime baltąjį triukšmą.

```
opar=par(mfrow=c(1,3))
wn1000=ts(rnorm(1000))  # Generuojame ilgio 1000 baltojo triukšmo seką
plot(wn1000)            # Brėžiame jos grafiką
acf(wn1000,15)          # Brėžiame jos autokoreliacinės funkcijos grafiką
                        # (maksimalus lagų skaičius 15)
pacf(wn1000,15)         # Brėžiame jos dalinės autokoreliacinės funkcijos grafiką
par(opar)
```

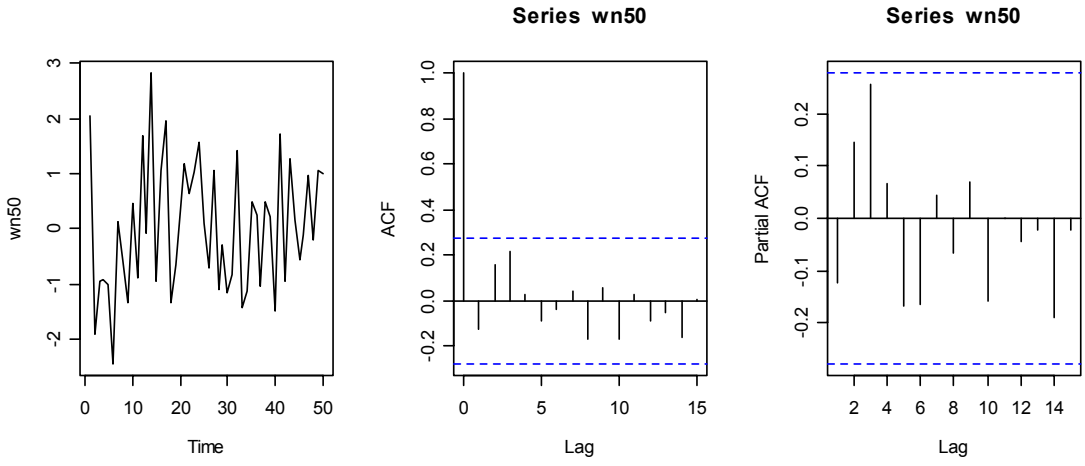
Matome, kad netgi šiuo idealiu atveju (sekos `wn1000` nariai „tikrai“ nekoreliuoti ir jų „daug“) kai kurios ACF ir PACF reikšmės „iššoka“ (žr. 1.10 pav.) iš mėlynų rėžių (teisybė, paprastai domimasi tik šių funkcijų elgesiu arti koordinačių pradžios).



1.10 pav. Baltojo triukšmo `wn1000` grafikas (kairėje), jo autokoreliacinės (viduryje) ir dalinės autokoreliacinės (dešinėje) funkcijų grafikai

Ekonometrijoje nagrinėjamos eilutės dažnai yra trumpos, panagrinėkime ir šį atvejį.

```
opar=par(mfrow=c(1,3))
wn50=ts(rnorm(50))      # Generuojame ilgio 50 baltojo triukšmo seką
plot(wn50)              # Brėžiame jos grafiką
acf(wn50,15)            # Brėžiame jos autokoreliacinės funkcijos grafiką
                        # (maksimalus lagų skaičius 15)
pacf(wn50,15)           # Brėžiame jos dalinės autokoreliacinės funkcijos grafiką
par(opar)
```



1.11 pav. Toks pat paveikslas, bet dabar seka turi tik 50 narių

Matome, kad šį kartą koreliacinių funkcijų elgesys labiau atitinka teoriją (beje, palyginkite mėlynų linijų aukščius[8] abiem atvejais). Pažymėsime, kad mėlynos linijos yra skaičiuojamos su prielaida, jog tiriamos laikinės sekos elementai yra nepriklausomi (taip būna tik tuomet, kai stebime normalųjį baltąjį triukšmą), todėl apskritai į šias linijas ir daromas išvadas reikia žiūrėti atsargiai.

[8] Linijos brėžiamos aukštyje $\pm 1,96/\sqrt{T}$ - tai 95% `acf` pasiklauties intervalas.

Iki šiol kalbėjome apie nestacionarias laikines sekas pavidalo $y_t = m_t + s_t + e_t$. Vienaip ar kitaip iš jų išskyrus ir pašalinus trendą m_t ir sezoninę dalį s_t (apie pašalinimo procedūras kalbėsime vėliau), likusi dalis e_t bus stacionari. Tokios y_t vadinamos TS eilutėmis (angl. Trend Stationary). Yra dar viena nestacionarių eilučių rūšis, vadinamosios DS (angl. Difference Stationary) eilutės – tai eilutės y_t, kurios pačios nėra stacionarios, bet kurių skirtumai $\Delta y_t = y_t - y_{t-1}$ yra stacionarūs. Štai du tokių eilučių pavyzdžiai:

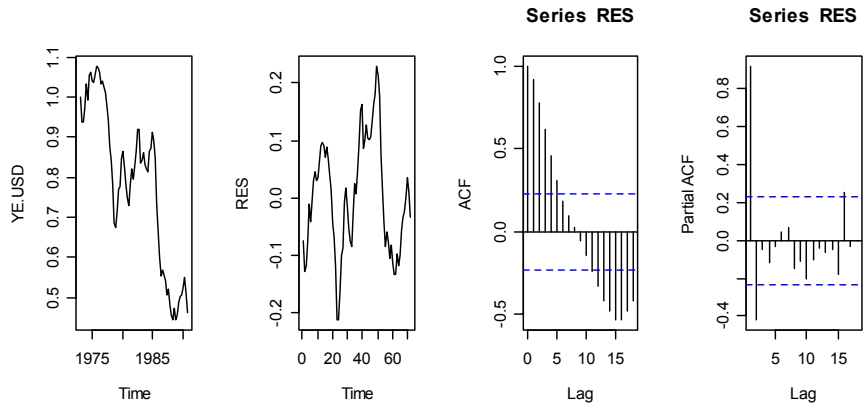
- $y_t = y_{t-1} + w_t$, $t = 1, 2, \ldots$ (kas ekvivalentu $y_t = y_0 + \sum_{i=1}^{t} w_t$ arba $\Delta y_t = w_t$, $t = 1, 2, \ldots$; tai vadinamasis atsitiktinio klaidžiojimo modelis)
- $y_t = y_{t-1} + a_0 + w_t$, $t = 1, 2, \ldots$ (kas ekvivalentu $y_t = y_0 + a_0 t + \sum_{i=1}^{t} w_t$ (šis procesas turi dvi nestacionarias komponentes – determinuotąją $a_0 t$ ir stochastinę $\sum_{i=1}^{t-1} w_t$) arba $\Delta y_t = a_0 + w_t$, $t = 1, 2, \ldots$; tai vadinamasis atsitiktinio klaidžiojimo su dreifu a_0 modelis).

Jei y_t yra TS seka, jos paklaidas rasime perėję prie skirtumų $y_t - m_t - s_t$; jei y_t yra DS seka, jos paklaidas rasime perėję prie (pirmųjų) skirtumų $y_t - y_{t-1}$

Apie DS sekų[9] analizę kalbėsime 3 skyriuje, dabar pateiksime tik vieną pavyzdį.

1.2 pavyzdys. Žemiau pateikta laikinė seka, kuri aprašo Japonijos jenos kursą (JAV dolerio atžvilgiu) nuo 1973 m. 1-ojo ketvirčio iki 1990 m. 4-ojo ketvirčio.

```
YE.USD=structure(c(1, 0.939, 0.939, 0.974, 1.031, 0.992, 1.055, 1.063, 1.04, 1.036, 1.056, 1.076,
1.072, 1.061, 1.032, 1.041, 1.012, 0.976, 0.943, 0.876, 0.842, 0.783, 0.684, 0.675, 0.714, 0.771,
0.776, 0.846, 0.863, 0.825, 0.78, 0.747, 0.729, 0.78, 0.822, 0.796, 0.828, 0.866, 0.918, 0.92,
0.836, 0.842, 0.86, 0.83, 0.819, 0.814, 0.863, 0.872, 0.913, 0.889, 0.846, 0.734, 0.666, 0.603,
0.552, 0.568, 0.543, 0.506, 0.521, 0.481, 0.454, 0.445, 0.474, 0.444, 0.455, 0.489, 0.504, 0.507,
0.524, 0.55, 0.515, 0.464), .Tsp = c(1973, 1990.75, 4), class = "ts")
```

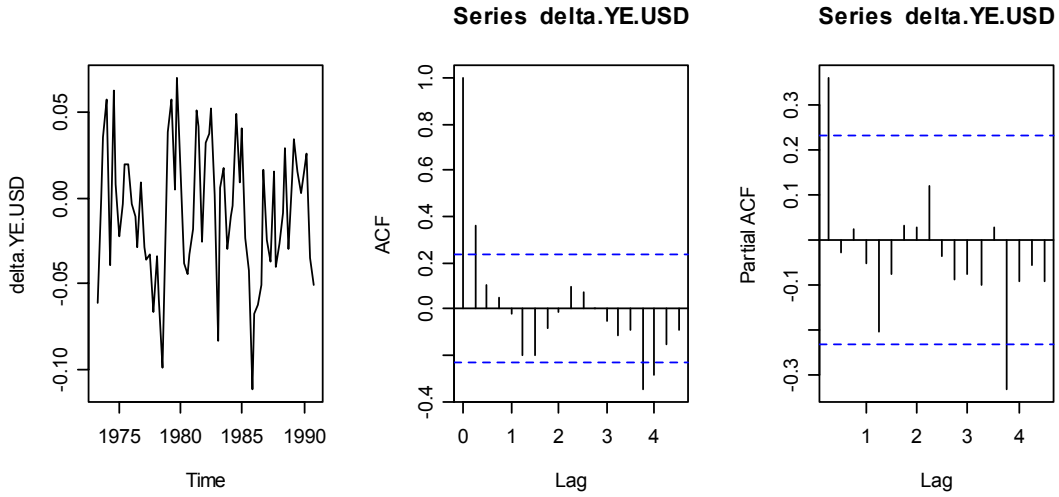


1.12 pav. Iš kairės į dešinę - jenos kursas; tiesinio modelio likučių (t.y. `lm(YU.USD~I(1:72))$res`) grafikas; likučių acf ir pacf grafikai

[9] Jos dar vadinamos eilutėmis su vienetine šaknimi (nes koeficientas prie y_{t-1} lygus 1). Čia $a_0 t$ yra *deterministinis*, o $a_0 t + \sum_{i=1}^{t-1} w_i$ - *stochastinis trendai*.

1.12 pav. matome, kad jenos kursas (kairėje) nėra stacionarus. Antra vertus, tiesinio trendo išskyrimas taip pat nepašalina nestacionarumo (likučiai `RES=ts(lm(YU.USD~I(1:72))$res)` (antras grafikas iš kairės)) tikrai nesudaro stacionarios eilutės (tai rodo ne tik pačių likučių grafikas (didelės nereguliarios ekskursijos aukštyn ir žemyn), bet ir jų acf grafikas, kuris neužgeso netgi po 16 ketvirčių!). Kadangi pacf pirmoji reikšmė lygi maždaug 0.9 (≈ 1), jenos kurso dinamikos seka ko gero turi vienetinę šaknį ir stochastinį trendą (plačiau apie tai 3 sk.), kurį galima pašalinti diferencijuojant. Iš tikrųjų, skirtumų eilutė (žr 1.13 pav., kairėje) yra, ko gero, stacionari, nors ir ne baltasis triukšmas (acf antrasis stulpelis reikšmingai skiriasi nuo 0).

```
opar=par(mfrow=c(1,3))
delta.YE.USD=diff(YE.USD)
plot(delta.YE.USD)
acf(delta.YE.USD)
pacf(delta.YE.USD)
par(opar)
```



1.13 pav. Jenos kurso skirtumų eilutė yra stacionari

1.4. Trendo išskyrimas

1.4.1. Mažiausiųjų kvadratų metodas

Būtų gerai, jei trendą galėtume užrašyti užrašyti funkciniu pavidalu – tuomet jį galėtume pratęsti į ateitį[10] ir naudoti prognozei. Tam galėtų tikti tiesė, parabolė, eksponentė ir pan. (deja, trendas retai aprašomas „reguliaria“ parametrine kreive).

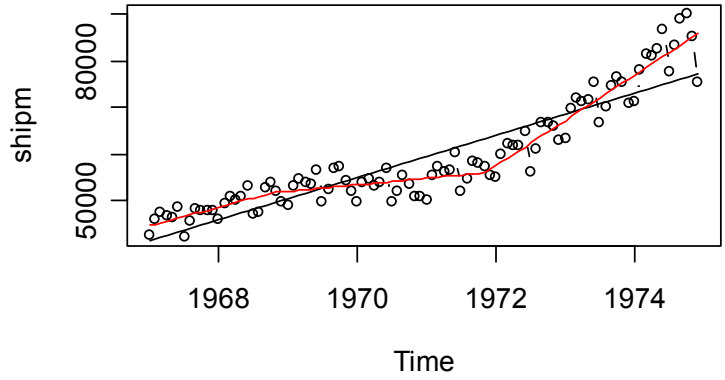
1.3 pavyzdys.

```
TIME=as.numeric(time(shipm))   # time(shipm) yra laikinė seka, o TIME - skaitinis
                               # vektorius
plot(shipm,type="b")           # „b"=both - išbrėš linijas ir rutuliukus
shipm.lm=lm(shipm~TIME)        # Išskiriame tiesinį trendą
lines(TIME,predict(shipm.lm)) # Trendas yra juoda tiesė grafike
```

Aišku, kad tiesė čia (žr. 1.13 pav.) mažai tinka. Labiau tiktų „laužyta“ tiesė iš trijų gabaliukų.

```
library(segmented)
```

[10] Regresinį modelį pratęsti už prognozinių kintamųjų reikšmių aibės, apskritai kalbant, rizikinga.



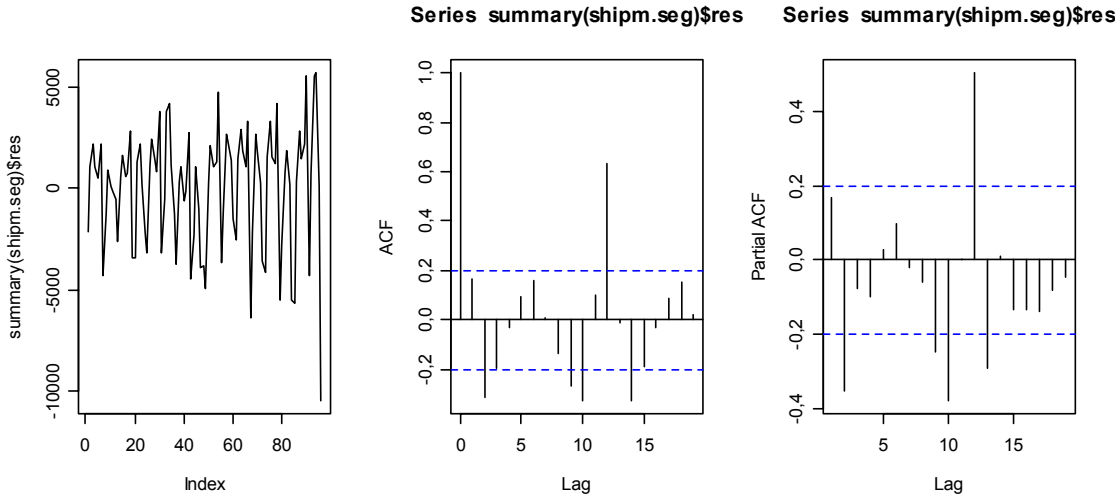
1.14 pav. `shipm` grafikas, tiesinės ir laužtinės regresijų kreivės

```
shipm.seg=segmented(shipm.lm,seg.Z=~TIME,psi=c(1969,1972))
# seg.Z nurodo kintamąjį, pagal kurį skaidoma į intervalus
# psi yra skaidymo taškų nulinių iteracijų reikšmės
lines(TIME, predict(shipm.seg),col=2) # Laužtinė regresija (raudona kreivė)
summary(shipm.seg)

Call:
segmented.lm(obj = shipm.lm, seg.Z = ~TIME, psi = c(1969, 1972))

Estimated Break-Point(s):
           Est. St.Err
psi1.TIME 1969 0.6248
psi2.TIME 1972 0.1696

t value for the gap-variable(s) V:  1.341232e-13 3.782368e-13

Meaningful coefficients of the linear terms:
                Estimate   Std. Error   t value    Pr(>|t|)
(Intercept) -7817734.202 2405269.4107 -3.250253 0.001622602
TIME            3997.138    1222.2673  3.270265 0.001523482
U1.TIME        -2595.227    1344.4572 -1.930315          NA
U2.TIME         8310.771     791.9946 10.493470          NA

Residual standard error: 3031 on 90 degrees of freedom
Multiple R-Squared: 0.9357,  Adjusted R-squared: 0.9322

Convergence attained in 8 iterations with relative change 0
```

Reikalui esant, dešinįjį laužtės gabalą galima pratęsti ir panaudoti trendo prognozei.

```
> names(summary(shipm.seg))
 [1] "call"          "terms"         "residuals"     "coefficients"  "aliased"
 [6] "sigma"         "df"            "r.squared"     "adj.r.squared" "fstatistic"
[11] "cov.unscaled"  "psi"           "Ttable"        "gap"           "it"
[16] "epsilon"       "short"

plot(summary(shipm.seg)$res,type="l")
acf(summary(shipm.seg)$res)
pacf(summary(shipm.seg)$res)
```

Laužtinės regresijos modelio likučių grafikas pakankamai stabilus (trendo nematyti), todėl jį galima panaudoti tolimesnei analizei. Beje, sezoninė komponentė čia dar likusi (`acf` ir `pacf` reikšmės taške 12 yra tikrai ne nulinės (išlenda iš „mėlynos“ juostos)), ją dar reikės pašalinti.

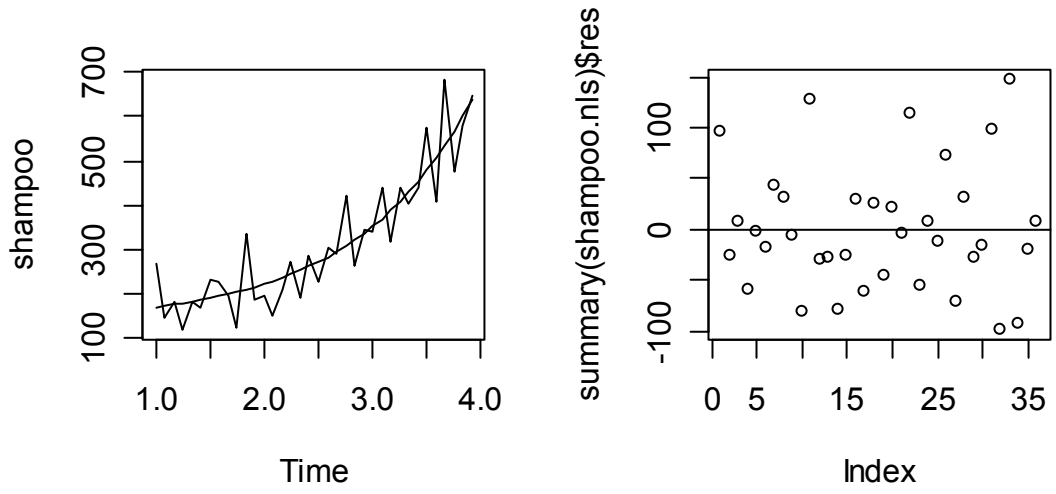


1.15 pav. `ship` laužtinės regresijos likučių ir jų acf grafikai

1.5 UŽDUOTIS. Kadangi `shipm` likučiai auga kartu su `TIME` (t.y., laiko) reikšmėmis, todėl geresnis turėtų būti multiplikatyvusis modelis. Pakartokite analizę su `log(ship)`. Išbrėžkite `shipm` grafiką, `shipm` laužtinės regresijos kreivę (kaip 1.14 pav.) ir tokią pat kreivę, apskaičiuotą analizuojant su `log(shipm)` (čia reikės maždaug tokios eilutės: `lines(TIME, exp(predict(log.shipm.seg),col=3))`). ◀

1.4 pavyzdys. Jei ekonomika kasmet didėja tuo pačiu procentu, tai ji auga eksponentiškai. Šis samprotavimas paaiškina, kodėl taip dažnai sutinkamas eksponentinis modelis. Papildoma tokiu atveju atsirandanti komplikacija yra ta, kad trendas dabar išskiriamas netiesinės regresijos metodais.

```
library(forecast)
data(shampoo) # ?shampoo - tai šampūno pardavimų apimtys; šampūno pardavimų
              # apimtys matyt neturėtų priklausyti nuo sezono
TIME=as.numeric(time(shampoo))
par(mfrow=c(1,2))
plot(shampoo) # Pardavimai auga maždaug eksponentiškai (žr. 1.15 pav.)
shampoo.nls=nls(shampoo~a+b*exp(c*TIME),start=list(a=100,b=10,c=1))
# Nulinės iteracijos "start" sąraše parinktos bandymų keliu
lines(TIME,predict(shampoo.nls))
plot(summary(shampoo.nls)$res) # Likučių dispersija beveik pastovi, tačiau jie,
                               # ko gero, nėra normalūs (jų skirstinys nėra
                               # simetriškas - patikrinkite su hist)
abline(0,0)

> summary(shampoo.nls)

Parameters:                    # Parametrų įverčiai
  Estimate Std. Error t value Pr(>|t|)
a 135.8101    45.0485   3.015 0.004917 **
b  13.4541    13.2165   1.018 0.316092
c   0.9248     0.2420   3.821 0.000558 ***
```



1.16 pav. `shampoo` duomenys ir eksponentinės regresijos kreivė (kairėje); eksponentinės regresijos likučių grafikas (dešinėje)

1.6 UŽDUOTIS. Išbrėžkite akcijų kainos grafiką

```
library(stats)
data(EuStockMarkets) # Daily Closing Prices of Major European Stock Indices,
                     # 1991-1998
plot(EuStockMarkets[,"DAX"])
```

ir brėžinį papildykite paraboliniu ir eksponentiniu trendais.

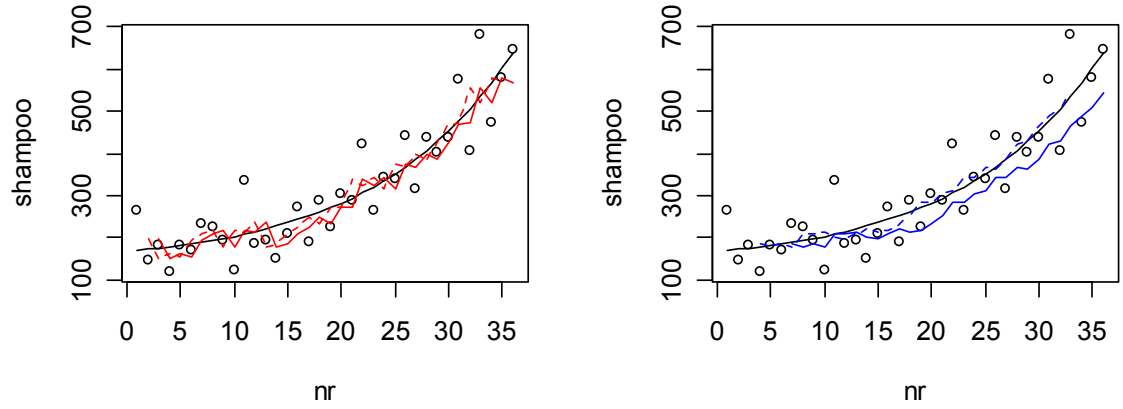
1.4.2. Slenkamojo vidurkio metodas

`shampoo` duomenis tik ką pateikėme tokiu pavidalu: `shampoo = 135,8 + 31.5* exp(0.9*TIME) + u`. Čia `u` yra regresinio modelio paklaida, o regresinė kreivė, nusakoma pirmaisias dviem dėmenimis, šį kartą (t.y., kai prognozinis kintamasis `x` yra laikas) vadinama trendu. Dažnai stebimos laikinės sekos trendą užrašyti žinomos funcijos pavidalu nepavyksta – tuomet kitimo tendenciją bandoma pateikti kitaip. Vadinamasis slenkamojo vidurkio metodas siūlo pardavimų lygį $h(t)$ pakeisti gretimų reikšmių vidurkiu, pavyzdžiui, $(h(t-2)+h(t-1)+h(t))/3$ (raudona linija) arba $(h(t-1)+h(t)+h(t+1))/3$ (raudona trūki linija; abiem atvejais visų stebinių svoriai vienodi ir lygūs 1/3). 1.17 pav. kairysis grafikas išbrėžtas naudojant tokias komandas:

```
nr=1:length(shampoo)
plot(nr,shampoo)
lines(nr,predict(shampoo.nls))
lines(nr,filter(shampoo,c(1,1,1)/3,sides = 1),col=2)  #vidurkinta pagal praeitį
lines(nr,filter(shampoo,c(1,1,1)/3,sides = 2),col=2,lty=2) #dvipusis glodinimas
```

o dešinysis grafikas – tokias:

```
plot(nr,shampoo)
lines(nr,predict(shampoo.nls))
lines(nr,filter(shampoo, rep(1,7)/7, sides = 1),col=4) # Visi svoriai = 1/7
lines(nr,filter(shampoo, rep(1,7)/7),col=4,lty=2)      # Visi svoriai = 1/7
```

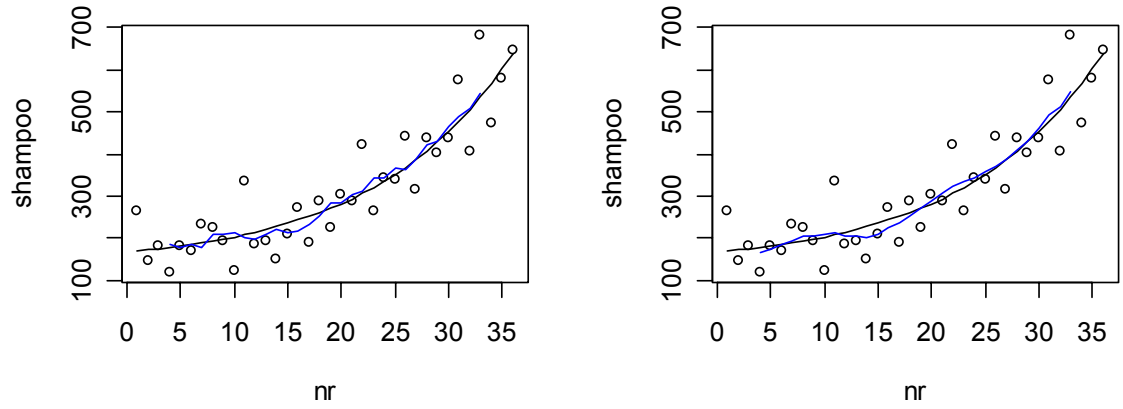


1.17 pav. `shampoo` duomenys glodinti pagal 3 taškus (kairėje) ir 7 taškus (dešinėje) (vidurkinimo pagal 7 taškus kreivė yra glodesnė); matome, kad eksponentė (juodoji linija) labai sėkmingai aprašo `shampoo` kitimą

1.4.3. Svertinio slenkamojo vidurkio metodas

1. Aišku, kad šios dienos (ketvirčio, mėnesio ir t.t.) reikšmei didžiausią įtaką daro artimiausios proceso reikšmės - ankstesni duomenys yra ne tokie svarbūs, todėl jų svoriai gali būti mažesni.

```
opar=par(mfrow=c(1,2))
plot(nr,shampoo)
lines(nr,predict(shampoo.nls))
lines(nr,filter(shampoo, rep(1,7)/sum(rep(1,7))),
col=4) # Stebinių svoriai vienodi ir lygūs 1/7 (dvipusis glodinimas)
plot(nr,shampoo)
lines(nr,predict(shampoo.nls))
lines(nr,filter(shampoo, c(1,2,3,4,3,2,1)/sum(c(1,2,3,4,3,2,1))),
col=4) # Stebinių svoriai proporcingi skaičiams 1,2,3,4,3,2,1 (dvipusis glod.)
par(opar)
```



1.18 pav. Glodinimas su lygiais svoriais (kairėje) ir gęstančiais svoriais (dešinėje); skirtumas šį kartą nėra didelis

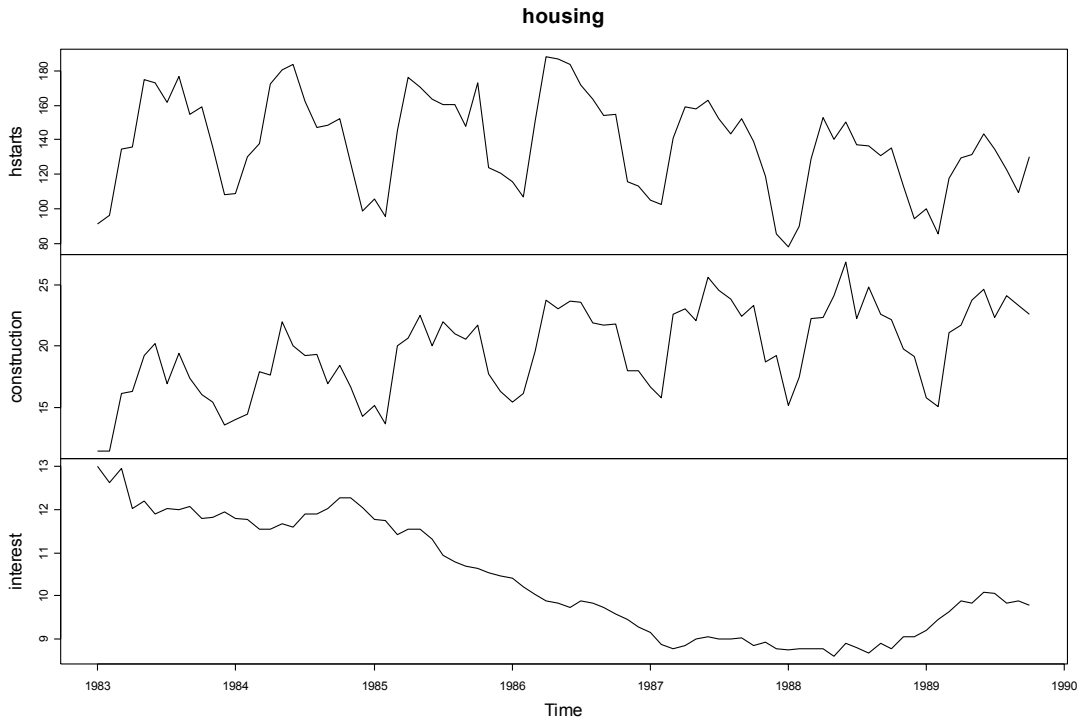
2. Tais atvejais, kai glodinama pagal lyginį taškų skaičių (pvz., pagal taškus 1, 2, 3 ir 4), sumą atitiks data, kuri nesutaps su turimomis (mūsų atveju gautume 2,5). Problemą galima pataisyti, suvidurkinus suglodintas reikšmes taškuose 2,5 ir 3,5 :

$$h''(3)=\frac{h'(2.5)+h'(3.5)}{2}=\left(\frac{h(1)+h(2)+h(3)+h(4)}{4}+\frac{h(2)+h(3)+h(4)+h(5)}{4}\right)/2=$$
$$=\frac{h(1)+2h(2)+2h(3)+2h(4)+h(5)}{8}.$$

Kitais žodžiais, dabar glodiname su nelygiais svoriais. Pažymėsime, kad šiuo atveju ne tik apskaičiuojame trendą, bet dalinai pašaliname ir ketvirtinį sezoniškumą (jei jis egzistuoja).

Panagrinėkime pavyzdį.

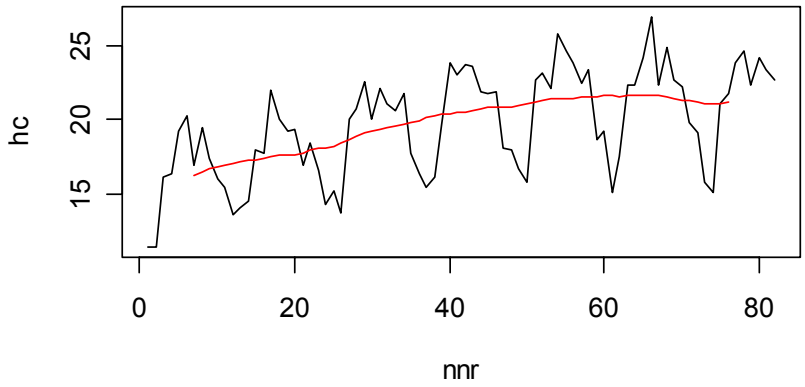
```
library(fma)
data(housing)
?housing # Namų statybą aprašanti trimatė laikinė seka
> tsp(housing)
[1] 1983.00 1989.75   12.00 # Duomenų periodiškumas lygus 12
> housing
         hstarts construction interest
Jan 1983    91.3       11.358    13.00
Feb 1983    96.3       11.355    12.62
Mar 1983   134.6       16.100    12.97
Apr 1983   135.8       16.315    12.02
.....................................

plot(housing) # Nagrinėsime 2-ąją komponentę (tai statybos kontraktų suma)
hc=housing[,"construction"]
```



1.19 pav. Trys trimatės laikinės sekos `housing` komponentės

Natūralu vidurkinti pagal 12 (lyginis skaičius!) mėnesių:

```
nnr=1:length(hc)
plot(nnr,hc,type="l")
lines(nnr,filter(hc, c(1,rep(2,11),1)/24),col=2) # Svoriai nelygūs!
```

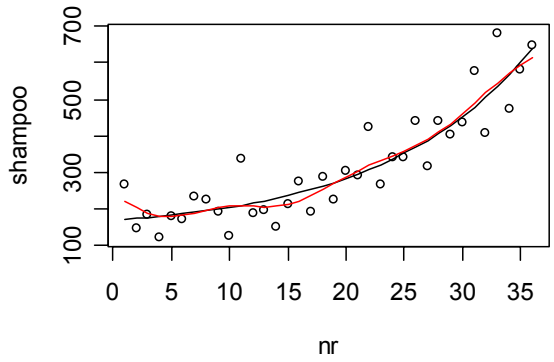


1.20 pav. Suglodinta (raudona) kreivė nepasiekia galinių taškų (trende likęs tik nežymus sezoniškumas)

Trendas panašus į parabolę, tačiau netgi įvertinus šią kreivę mažiausių kvadratų metodu, pratęsti ją į dešinę rizikinga!

3. Svertinių vidurkių teorija gerai išvystyta (R paketas `locfit` turi Spencer'io 15 ir 21 taško glodinimo procedūras `spence.15` ir `spence.21`[11], o EViews'as turi dar ir Henderson'o procedūrą). Abi šios procedūros yra dalinai išsprendusios kraštinių taškų problemą (kraštuose dvipusis glodinimas pakeičiamas vienpusiu).

```
library(locfit)
plot(nr,shampoo); lines(nr,predict(shampoo.nls))
lines(nr,spence.21(shampoo),col=2)   # Raudona kreivė
```



1.21 pav. Spencer'io glodinimas su 21 tašku (dešinėje norėtųsi kitokio raudonos kreivės elgesio!)

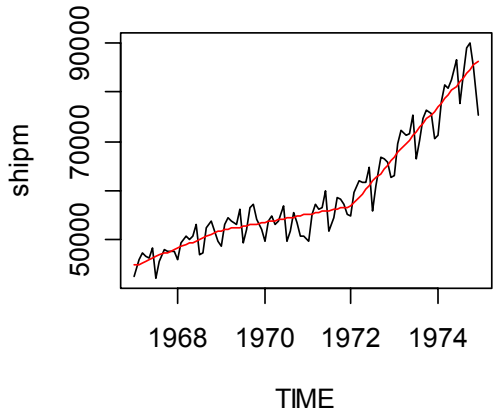
[11] Išskyrus kraštinius taškus, `spence.21` tiksliai "pagauna" polinomus iki trečios eilės.

1.4.4. Splaininė regresija

Skyrelį pradėsime keliais žodžiais apie splainus. Jei *xy* koordinačių sistemoje turime *n* taškų $\{(x_i, y_i),\ i = 1,...,n\}$, tai funkcija, padaryta iš *p*-tosios eilės (dažniausiai *p*=3) polinomų (kiekviename *x*'sų intervale vis kitų), glodžiai sujungianti minėtus taškus, vadinama splainu. Kita vertus, polinomus galima vartoti ir regresijos uždaviniuose. Deja, polinominė regresija yra „nelokali" – pakeitus vieną tašką, polinomas gali smarkiai pasikeisti ir toli nuo jo. Kaip išeitį, visą *x*-sų sritį galima suskaidyti į kelis intervalus (jų galiniai taškais vadinami mazgais) ir kiekviename iš jų ieškoti savo regresinio polinomo. Deja, tokie polinomai mazguose paprastai „nesulimpa". *Kubiniai regresiniai splainai* yra minėti polinomai su papildoma „sulipimo" mazguose sąlyga. *Natūralieji kubiniai regresiniai splainai* turi dar du mazgus minimalaus ir maksimalaus *x*-sų taškuose (šiuo atveju papildomai reikalaujama, kad splainas būtų tiesinis už šios srities – tai garantuoja „tinkamą" splaino elgesį srities pakraščiuose). Jei mazgai yra iš anksto fiksuoti, tai splaininė regresija yra įprastinis tiesinis parametrinis modelis. Pvz., tiesinė[12] splaininė regresija su dviem mazgais c_1 ir c_2 yra užrašoma taip: $y = A + B_1 x + B_2 (x - c_1)_+ + B_3 (x - c_2)_+$; čia $z_+ = \begin{cases} z, & jei\ z > 0 \\ 0, & jei\ z \le 0 \end{cases}$. Jei grįžtume prie `shipm` pavyzdžio (žr. 1.13 pav.) ir pasirinktume $c_1 = 1969$ ir $c_2 = 1972$, tai programos

```
plius=function(x) ifelse(x>0,x,0)
TIME=as.numeric(time(shipm))
shipm.s.lin=lm(shipm~TIME+plius(TIME-1969)+plius(TIME-1972))
plot(TIME,shipm,type="l")
lines(TIME,predict(shipm.s.lin),col=2)
```

rezultatas praktiškai sutaptų su 1.13 pav. (žr. 1.21 pav.) (laužtinės regresijos procedūra dar šiek tiek „stumdo" mazgus).



1.22 pav. Tiesinė splaininė regresija su fiksuotais mazgais

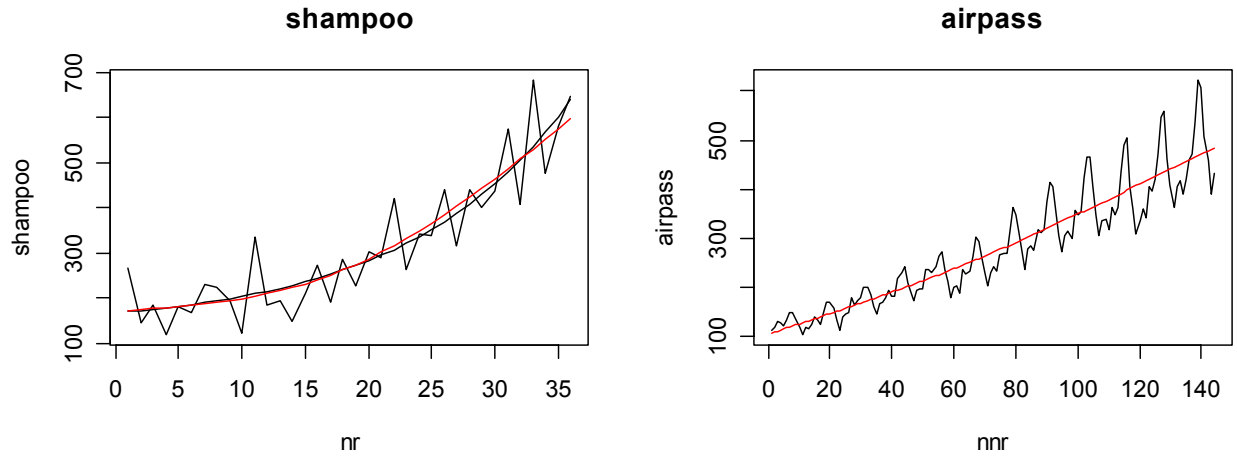
Natūraliuosius kubinius splainus su (iš anksto pasirinktais) *K* mazgais galima apibrėžti taip:

$$y = A_1 + B_1 x + B_2 (x - c_1)_+^3 + B_3 (x - c_2)_+^3 + ... + \\ + B_{K+1} (x - c_K)_+^3$$

Yra daug splaininės regresijos variantų. Čia mes taikysime `gam` funkciją iš `mgcv` paketo.

```
opar=par(mfrow=c(1,2))
library(mgcv);library(fma);data(shampoo)
nr=1:length(shampoo);TIME=as.numeric(time(shampoo))
shampoo.nls=nls(shampoo~a+b*exp(c*TIME),start=list(a=100,b=10,c=1))
shampoo.gam <- gam(shampoo~s(nr))  # "s" nuo "spline"
plot(nr,shampoo,type="l",main="shampoo")
lines(nr,predict(shampoo.nls))     # Eksponentinė regresija
lines(nr,shampoo.gam$fitted,col=2) # Splainų regresija; abi kreivės beveik sutampa
data(airpass)                      # International airline passengers (1949-1956)
nnr=1:length(airpass)
plot(nnr,airpass,type="l",main="airpass")
airpass.gam <- gam(airpass~s(nnr))
lines(nnr,airpass.gam$fitted,col=2);par(opar)
```

[12] T.y., ne kubinė.

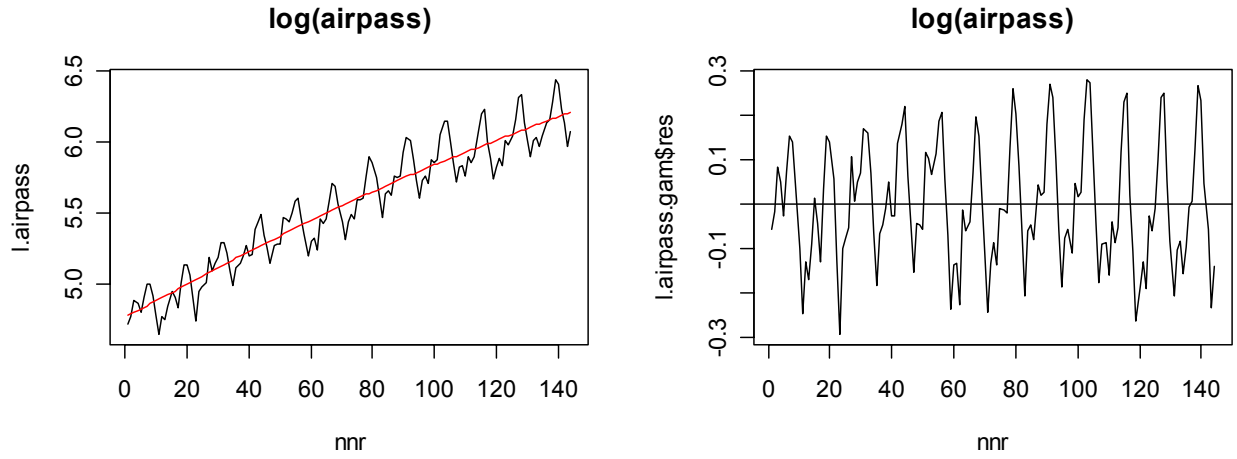


1.23 pav. `shampoo` ir `airpass` duomenys (splainai išbrėžti raudona spalva)

Aiškiai matyti, kad kaip beglodintume, `airpass` duomenų atveju modelio paklaidos laikui augant vis didėja, t.y., ši laikinė seka ko gero turi multiplikatyvųjį pavidalą: $airpass_t = m_t \cdot s_t \cdot u_t,\ u_t > 0$. Norint taikyti tradicinį adityvųjį modelį, laikinę seką `airpass` reikia pirmiau išlogaritmuoti.

```
opar=par(mfrow=c(1,2))
l.airpass=log(airpass)
plot(nnr,l.airpass,type="l",main="log(airpass)")
l.airpass.gam <- gam(l.airpass~s(nnr))
lines(nnr,l.airpass.gam$fitted,col=2)
plot(nnr,l.airpass.gam$res,main="log(airpass)",type="l")
abline(0,0)
par(opar)
```

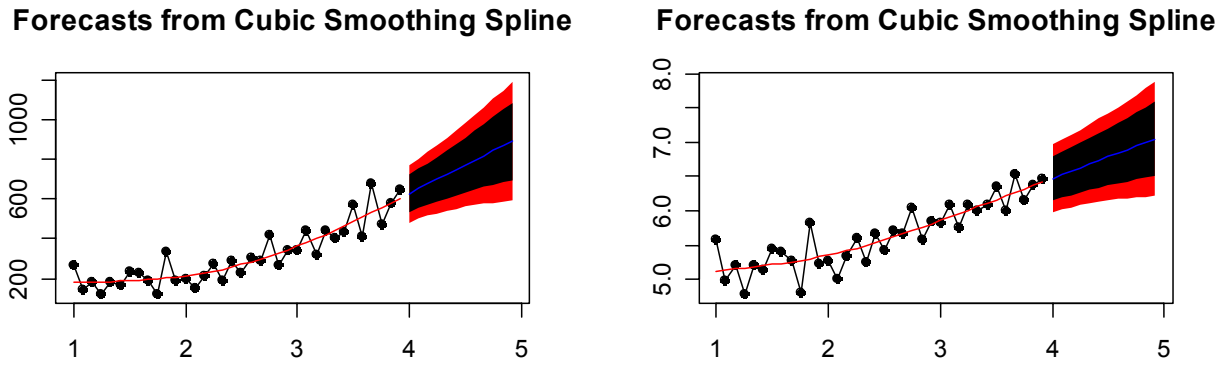
Matome, kad splaininės regresijos likučiai dabar yra homoskedatiški, t.y., pastovios dispersijos (teisybė, jų sezoniškumą dar reikės pašalinti).



1.24 pav. `log(airpass)` trendas (raudona kreivė kairėje) nėra tiesinis; modelio likučiai turi akivaizdų sezoniškumą

Jau matėme, kad duomenis labai patogu aprašyti splainais. Čia trumpai aptarsime `splinef` funkciją iš `forecast` paketo, kuri istorinius duomenis aproksimuoja kubiniais splainais, o juos prognozuoja tiesine funkcija.

```
library(forecast)
library(fma)
data(shampoo)
par(mfrow=c(1,2))
fcast <- splinef(shampoo,h=12)
plot(fcast)
fcast.l <- splinef(log(shampoo),h=12)
plot(fcast.l)
```



1.25 pav. `shampoo` auga maždaug eksponentiškai (grafikas kairėje), todėl jų logaritmai (dešinėje) maždaug tiesiškai.

1.4.5. Diferencijavimas

Kai kurias nestacionarias laikines sekas galima paversti stacionariomis, jas diferencijuojant (kitaip sakant, jų skirtumų procesas $\Delta y_t = y_t - y_{t-1}$ trendo nebeturi). Priminsime, kad tokiu atveju sakome, kad eilutė turi stochastinį trendą[13] arba vienetinę šaknį, o eilutės tyrimas reikalauja specialių metodų (žr. 3 sk.). Čia tik pademonstruosime tik tai, kad diferencijavimas „reguliarizuoja“ laikinę seką.

Nagrinėkime daugiamatę laikinę seką `EuStockMarkets` (tai Daily Closing Prices of Major European Stock Indices, 1991-1998).

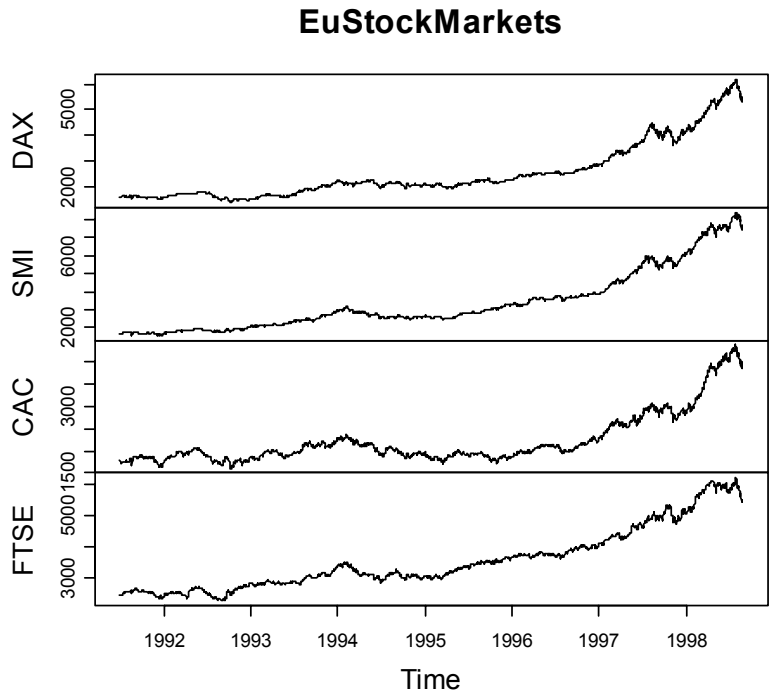
```
 > attributes(EuStockMarkets)
$dim
[1] 1860    4
$dimnames
$dimnames[[1]]
NULL
$dimnames[[2]]
[1] "DAX"  "SMI"  "CAC"  "FTSE"
$tsp
[1] 1991.496 1998.646  260.000
```

[13] Įprastinio trendo (t.y., neatsitiktinės funkcijos m_t tokios, kad $y_t - m_t$ yra stacionarus procesas plius, gal būt, sezoninė dalis) tokia laikinė seka neturi.

```
$class
[1] "mts" "ts"
```

Visos keturios sekos yra akivaizdžiai nestacionarios:

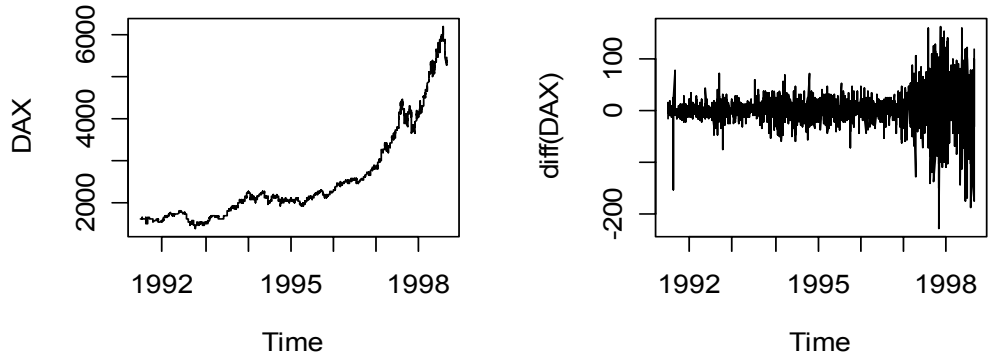
```
plot(EuStockMarkets)
```



1.26 pav. Keturių pagrindinių Europos vertybinių popierių biržų indeksai

Antra vertus, skirtumų sekos elgiasi žymiai reguliariau:

```
DAX=EuStockMarkets[,"DAX"] # Apsiribokime Vokietijos DAX (Ibis) indeksu
opar=par(mfrow=c(1,2))
plot(DAX)                  # Indekso grafikas
plot(diff(DAX))            # Indeksų skirtumų grafikas
par(opar)
```



1.27 pav. Skirtumų grafikas (dešinėje) trendo, atrodo, nebeturi

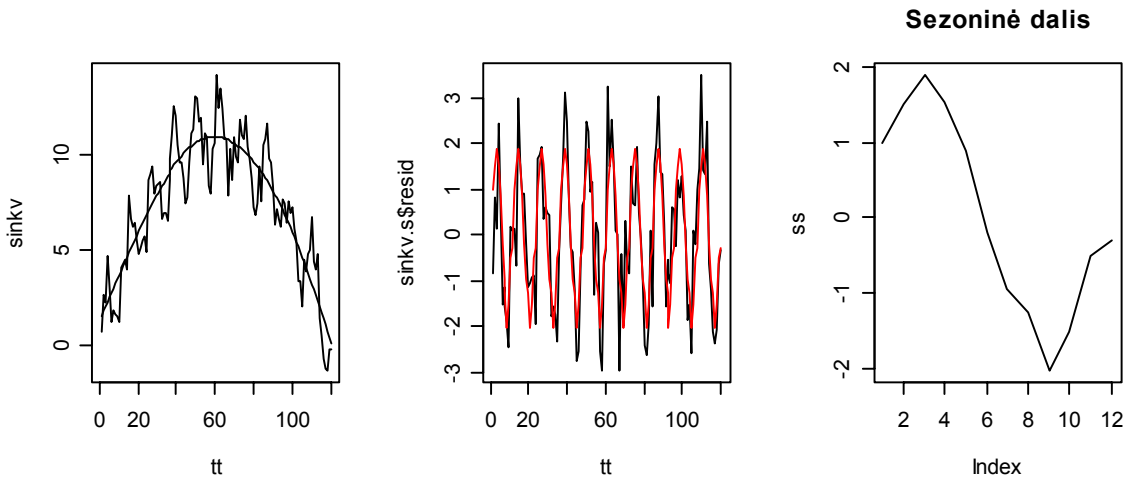
1.7 UŽDUOTIS. Parašykite trumpą referatą apie DAX ir Lietuvos vertybinių popierių biržą.

1.5. Sezoninės dalies išskyrimas

Tarkime, kad stebima laikinė seka yra aprašoma adityviuoju modeliu: $y_t = m_t + s_t + e_t$. Įvertinus jos trendą $\hat{m}_t$ kokiu nors būdu (mažiausių kvadratų, slenkamuoju vidurkiu, splainu ir t.t.), skirtumas $z_t = y_t - \hat{m}_t$ vis dar turės sezoninę komponentę s_t. Tarkime, kad laikinės sekos dažnis lygus 4 (t.y., nagrinėjame ketvirtinius duomenis). 1-ojo ketvirčio efektą galime įvertinti, suvidurkinę visus pirmuosius ketvirčius: $\hat{s}_1 = (z_1 + z_5 + ...) / \textit{metų skaičiaus}$, 2-ojo – visus antruosius ir t.t.

Panagrinėkime dirbtinių duomenų pavyzdį – tai „pagadinta“ sinusoidė (10 metų po 12 mėnesių) su kvadratiniu trendu, kurį išskirsime splainų metodu.

```
opar=par(mfrow=c(1,3))
tt = 1:120
set.seed(1)
e = rnorm(120)                                  # Modelio paklaidos
sinkv=2*sin(2*pi*tt/12)+e+0.003*tt*(120-tt)     # Žalia spalva pažymėtas trendas
plot(tt,sinkv,type="l")
sinkv.s = gam(sinkv~s(tt))                      # Išskiriame splainininį trendą
lines(tt,sinkv.s$fitted)
plot(tt,sinkv.s$resid,type="l")
lines(tt,rep((ss <- apply(matrix(sinkv.s$resid,ncol=12,byrow=TRUE),2,mean)),10),
col=2) # Brėžiame sezoninę dalį ss(=ŝ_t) (raudona kreivė)
plot(ss,type="l",main="Sezoninė dalis")
par(opar)
```

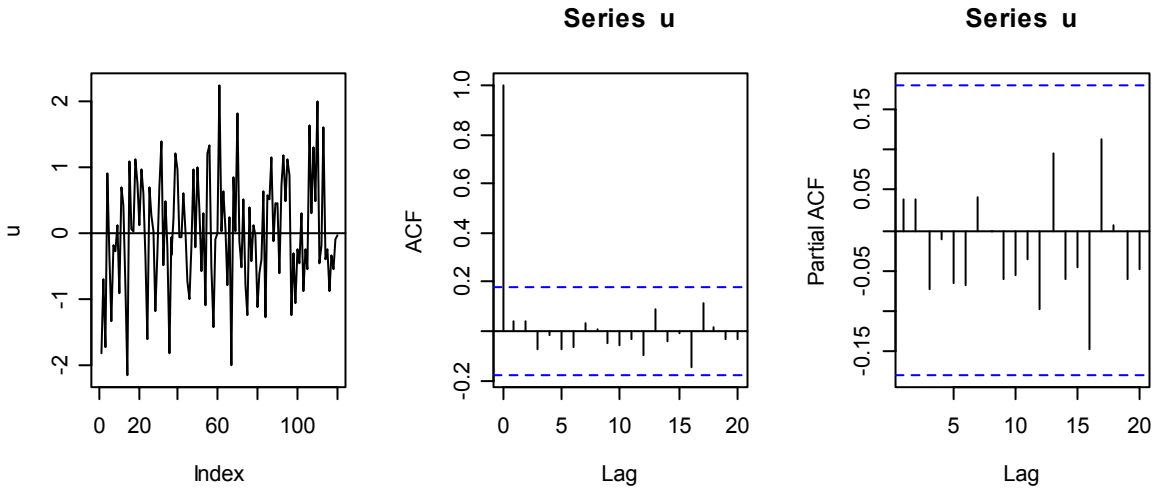


1.28 pav. Dirbtiniai duomenys su kvadratiniu trendu (trendas įvertintas splainais, jis išbrėžtas kairiajame brėžinyje); iš likučių išskirta sezoninė dalis (raudona kreivė viduryje); vieno sezoninės dalies periodo grafikas (dešinėje)

Nesunku įsitikinti, kad modelio likučiai $\hat{e}_t = y_t - (\hat{m}_t + \hat{s}_t)$ sudaro baltąjį triukšmą:

```
opar=par(mfrow=c(1,3))
u=sinkv.s$resid-rep(apply(matrix(sinkv.s$resid,ncol=12,byrow=TRUE),2,mean),10)
plot(u,type="l")
```

```
abline(0,0)
acf(u)
pacf(u)
par(opar)
```

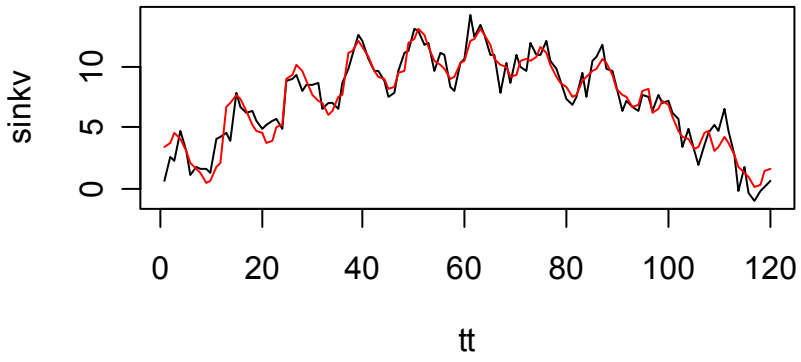


1.29 pav. Sprendžiant pagal paklaidų $\hat{u}_t$ grafiką (kairėje), paklaidos sudaro stacionarią seką; ši stacionari seka yra, matyt, baltasis triukšmas (grafikai viduryje ir dešinėje)

Iš esmės tą patį rezultatą gautume, jei sezoninę dalį išskirtume regresiniais metodais. Įveskime du žymimuosius kintamuosius, kuriuos R programoje galima pakeisti faktoriais: `met` (=1,2,…,10) ir `men` (=1,2,…,12) (metų kintamasis vaizduos trendą, o mėnesio – sezoninę dalį).

```
tt = 1:120
set.seed(1)
e = rnorm(100) # modelio paklaidos
sinkv=2*sin(2*pi*tt/12)+e+0.003*tt*(120-tt)
met= factor(rep(1:10,rep(12,10))) # "Metų" faktorius
men=factor(rep(1:12,10))          # "Mėnesių" faktorius
(sinkv.lm=lm(sinkv~met+men))
```

```
Coefficients:          # Priminsime: žymimųjų kintamųjų turi būti vienu mažiau
(Intercept)         met2         met3         met4         met5         met6
     3.4622       3.1825       5.5969       7.5645       8.4110       8.5478
       met7         met8         met9        met10         men2         men3
     7.0170       6.1102       2.6913      -0.3627       0.3035       1.1127
       men4         men5         men6         men7         men8         men9
     0.6342      -0.3103      -1.3436      -1.8237      -2.1061      -2.9406
      men10        men11        men12
    -2.7289      -1.6078      -1.3825
# Žali koeficientai nusako trendą, o melsvi - sezoninę dalį
plot(tt,sinkv,type="l")             # Pradiniai duomenys
lines(tt,predict(sinkv.lm),col=2) # Regresinio modelio prognozė
```



1.30 pav. `sinkv` ir jo (tiesine regresija aprašytas) modelis; jei vietoje laiptuotos metų dedamosios vartotume kokią glodžią kreivę, grafikas būtų gražesnis

1.8 UŽDUOTIS. Užrašykite lygybes, pagal kurias modelis skaičiuoja pirmų 15 mėnesių reikšmes. ◀◀

Šiame regresiniame modelyje nesunku išskirti metinį „trendą“ ir sezoninę dalį (žr. 1.30 pav.).

```
par(mfrow=c(1,3))
plot(tt,sinkv,type="l")
lines(tt,predict(lm(sinkv~met)),col=2)    # „Metų“ įtakos dedamoji
plot(tt,predict(lm(sinkv~men)),type="l") # „Neteisinga“ sezoninė dalis
```

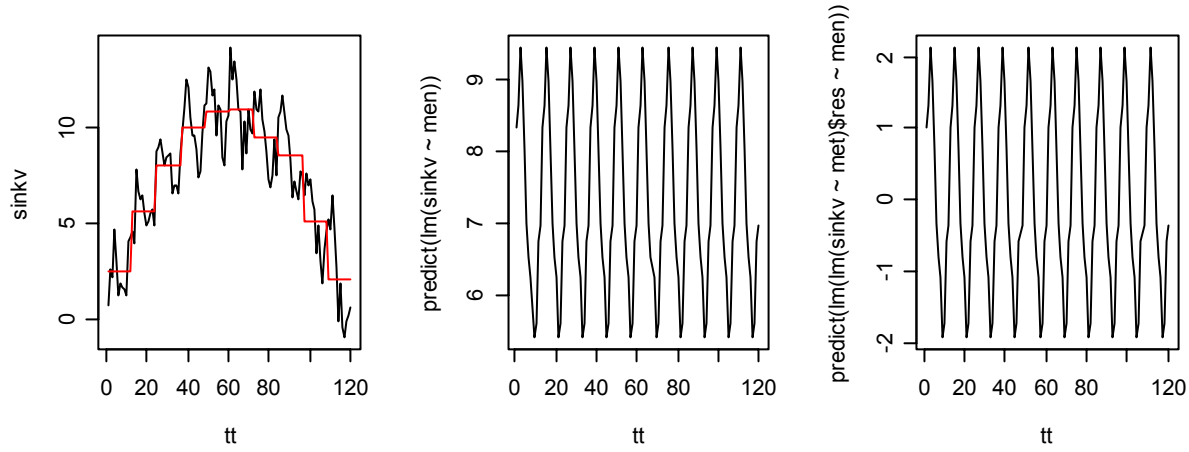
Ko gero tikslingiau ieškoti ne visų pirmųjų „mėnesių“ vidurkio (čia netiesiogiai įeina trendas), bet adityviąją sezoninę dalį apibrėžti kaip skirtumo tarp `sinkv` ir jo metinio trendo (t.y., `lm(sinkv~met)$res`) sezoninę dalį.

```
plot(tt,predict(lm(lm(sinkv~met)$res~men)),type="l") # "Teisinga" sezoninė dalis
```

Kadangi metinis trendas išskirtas gana grubiai, netgi ši „teisinga“ sezoninė dalis dar nėra panaši į sinusoidę (žr 1.30 pav., dešinėje). Jei metinį trendą išskirtume splainais (tai darėme anksčiau) arba slenkamojo vidurkio pagalba (plg. `plot(decompose(ts(sinkv,freq=12)))` arba kokia nors glodinimo procedūra (plg. `plot(stl(ts(sinkv,freq=12),"per"))`), atsakymas būtų „gražesnis“.

1.9 UŽDUOTIS. Pakomentuokite funkcijas `decompose` ir `stl`.

1.10 UŽDUOTIS. Išskirkite sezoninę dalį iš `MASS` paketo `nottem` duomenų (tam galite panaudoti regresiją kintamojo `men` atzvilgiu). Naudodami `acf` ir `pacf` funkcijas, ištirkite likučių procesą.



1.31 pav. Kiekvieno mėnesio vidurkis (raudona linija kairėje) ir sezoninės proceso `sinkv` dalys: „neteisinga“ (viduryje) ir „teisinga“ (dešinėje; atkreipkite dėmesį į y ašies mastelius); kadangi metinis „trendas“ išskirtas gana grubiai, sezoninė dalis nėra sinusoidė

Dabar tarkime, kad stebimoji laikinė seka yra aprašoma multiplikatyviuoju modeliu: $y_t = m_t \cdot s_t \cdot e_t$, $e_t > 0$. Šį kartą $z_t = y_t / \hat{m}_t$, o visa kitą paliksime kaip anksčiau. Tiksliau sakant, kiek modifikuosime savo procedūrą (variantų čia daug) ir dabar pateiksime laikinę seką su sezonine pataisa (plg. [PR, Part 4]). Tarkime, kad mūsų sekos periodas lygus 12.

1. Apskaičiuokime 12 mėnesių slenkamąjį vidurkį (jis turėtų „beveik“ nepriklausyti nuo sezoninės ir paklaidos komponenčių ir turėtų būti grubus trendo įvertis): $\hat{m}_t = (x_{t-5} + x_{t-4} + ... + x_{t+6})/12$
2. Padalinę y_t iš $\hat{m}_t$, gauname pradinį sezoninio nario s_t įvertį: $\hat{s}_t = \widehat{s_t \cdot e_t} = y_t / \hat{m}_t$
3. Dar kartą vidurkindami, pabandysime „visai“ pašalinti atsitiktinę paklaidą:

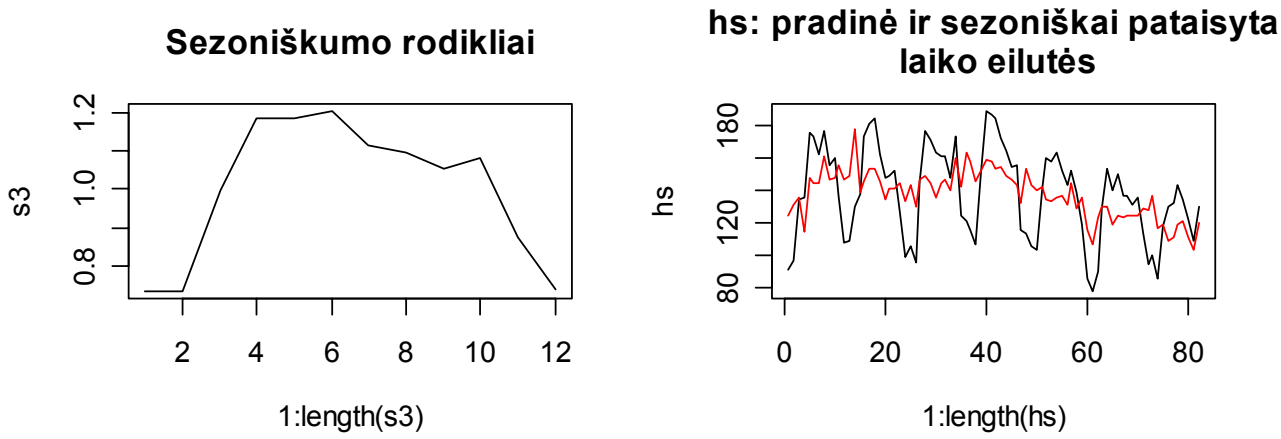
$$\hat{\hat{s}}_1 = (\hat{s}_1 + \hat{s}_{13} + ...) / \textit{metų skaičiaus}$$
$$\hat{\hat{s}}_2 = (\hat{s}_2 + \hat{s}_{14} + ...) / \textit{metų skaičiaus} \text{ ir t.t.}$$

Skaičiai $\hat{\hat{s}}_1, \hat{\hat{s}}_2, ..., \hat{\hat{s}}_{12}$ vadinami sezoniškumo rodikliais (tiksliau sakant, juos, gal būt, dar reikės normuoti (taip, kad jų suma būtų lygi 12)). Laikinė seka $y_1^p = y_1 / \hat{\hat{s}}_1, ..., y_{12}^p = y_{12} / \hat{\hat{s}}_{12}, y_{13}^p = y_{13} / \hat{\hat{s}}_1, ...$ vadinama laikine seka be sezoninės dalies (seka su pašalintu sezono poveikiu).

Panagrinėkime namų statybos pavyzdį.

```
opar=par(mfrow=c(1,2))
library(forecast); library(fma); data(housing)
hs <- housing[,"hstarts"] # Sutrumpiname žymėjimą
tr <- filter(hs, rep(1,12)/12)
s1 <- hs/tr
s2 <- apply(matrix(s1,ncol=12,byrow=TRUE),2,mean,na.rm=TRUE)
s3 <- 12*s2/sum(s2)       # Normuojame
plot(1:length(s3),s3,type="l",main="Sezoniškumo rodikliai")
hs.p <- hs/c(rep(s3,6),s3[1:10])
plot(1:length(hs),hs,type="l", main="hs: pradinė ir sezoniškai pataisyta\n laiko
eilutės")
```

```
# pradinių duomenu grafikas (aiškiai matyti sezono poveikis)
lines(1:length(hs),hs.p,col=2) # hs su pašalintu sezono poveikiu
par(opar)
```



1.32 pav. Sezoniškumo rodikliai (kairėje) ir `hs` grafikai (dešinėje; pradinių duomenų (juodas) ir su pašalintu sezono poveikiu (raudonas))

1.6. Integruotieji metodai

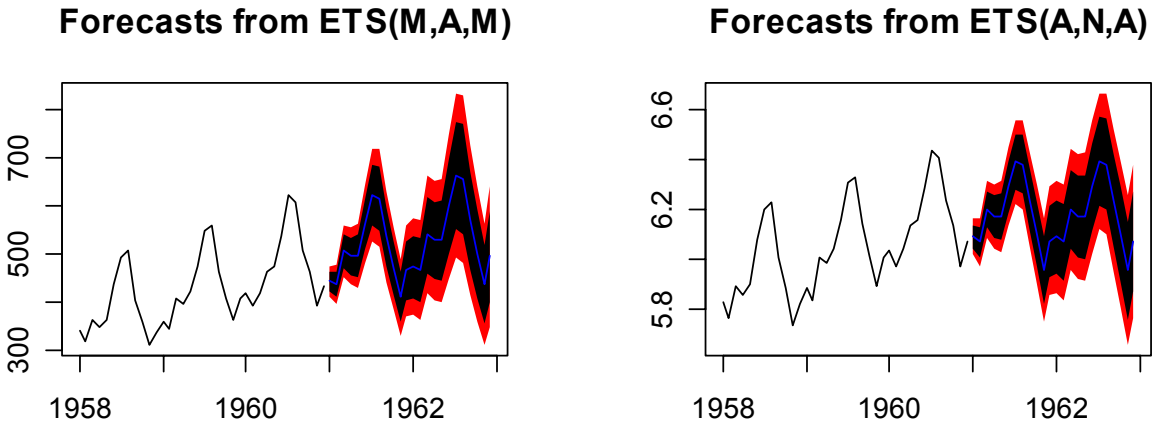
Aptarsime kelis metodus, kurie trendą ir sezoninę dalį išskiria vienu kartu.

1.6.1. Eksponentinis glodinimas ir prognozavimas

Slenkamojo vidurkio metodai fiksuoja esamą padėtį, tačiau nebando laikinės sekos reikšmių pratęsti į ateitį. Šį prognozavimo uždavinį sprendžia įvairūs eksponentinio glodinimo variantai, jie smulkiai aprašyti [MWH] knygoje (žr., taip pat, [D, 362 psl.]). Čia pažymėsime tik tiek, kad paketo `forecast` funkcijos `ses` (single exponential smoothing), `holt` (Holt'o ir Winters'o eksponentinis glodinimas procesamas su trendu) ir `hw` (Holt'o ir Winters'o glodinimas procesamas su trendu ir sezonine dedamąja) reikalingų glodinimo parametrų parinkimą atlieka automatiškai. Tiksliau kalbant, visos šios trys funkcijos yra tik patogūs funkcijos `ets` pavidalai (rekomenduočiau visuomet vartoti pastarąją, nes ji automatiškai parenka ne tik parametrus, bet ir tinkamą modelį, įskaitant ir adityviojo ar multiplikatyviojo varianto parinkimą).

Panagrinėkime pakete `fma` esantį duomenų rinkinį `airpass`, kuriame pateikti klasikiniai Box'o ir Jenkins'o „airline data" – tai mėnesiniai 1949-1956 m. duomenys apie tarptautinėmis oro linijomis skraidintų keleivių skaičių) (žr. 1.1 pav.; `airpass` duomenys aprašomi multiplikatyviuoju modeliu (įsitikinkite), o minėtame paveiksle yra išbrėžtas keleivių skaičiaus logaritmų grafikas – logaritmai aprašomi adityviuoju modeliu, ar ne?).

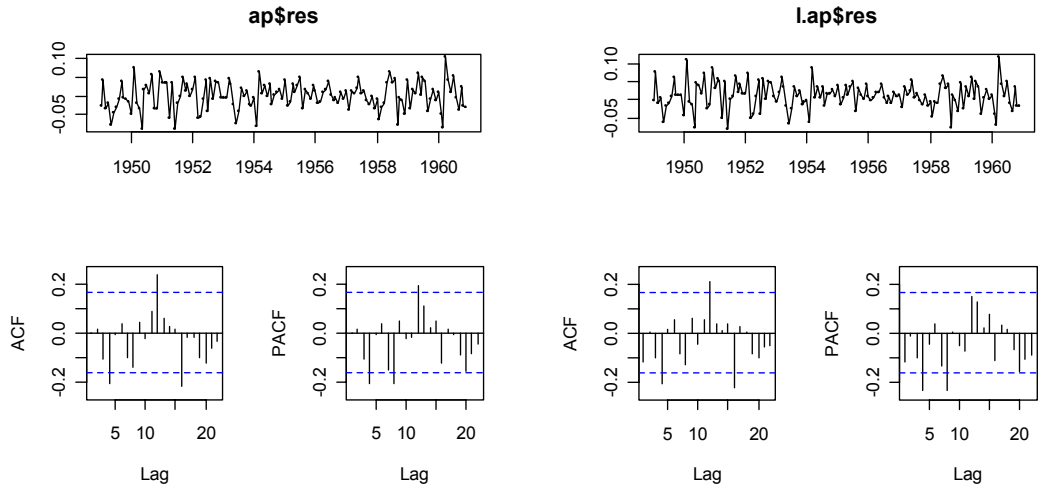
```
library(fma)
data(airpass)
par(mfrow=c(1,2))
ap=ets(airpass)                 # pradinių duomenų modelis
plot(forecast(ap),include=36)  # į grafiką įtrauksime tik paskutinių 36 mėnesių
                                # duomenis
l.ap=ets(log(airpass))          # logaritmuotų duomenų modelis
plot(forecast(l.ap),include=36)
```



1.33 pav. Pradinių duomenų (kairėje) ir logaritmuotų duomenų (dešinėje) paskutinių 36 mėnesių reikšmės ir 24 mėnesių prognozė. Grafikų viršuje esančios raidės reiškia, kad `ets` funkcija automatiškai nustatė (žr. kairėje), kad paklaidos yra multiplikatyvios (M), trendas – adityvus (A), o sezoninė dedamoji – multiplikatyvi (M); logaritmuotų duomenų atveju (žr. dešinėje) paklaidos yra adityvios, trendo nėra (N), o sezoninė dedamoji – adityvi. Atkreipkite dėmesį į y ašies reikšmes.

Priminsime, kad laikinės sekos skaidymas laikomas baigtu, jei modelio paklaidos sudaro stacionarią seką. Patikrinsime šį faktą mūsų atveju.

```
tsdisplay(ap$res)
tsdisplay(l.ap$res)
```



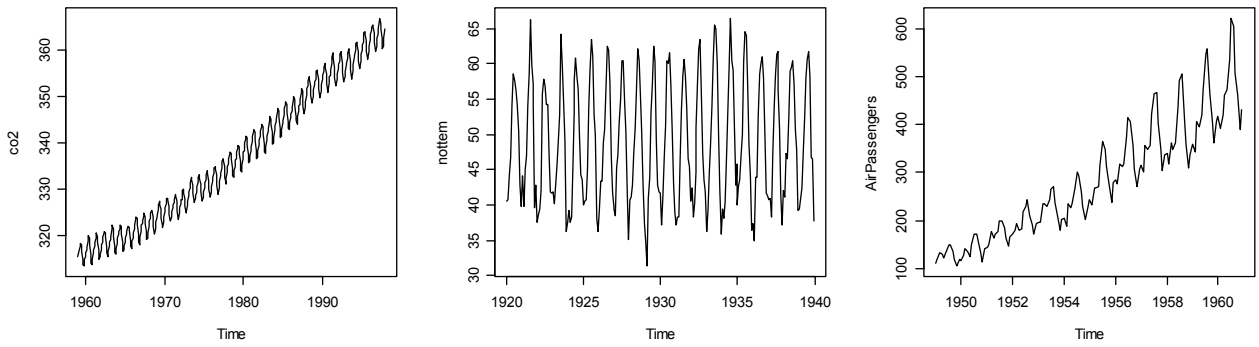
1.34 pav. Abiejų modelių liekanos panašios į stacionarias. Antra vertus, jos lyg ir turi sunkiai paaiškinamą dydžio 4 periodiškumą (žr. ACF ir PACF grafikus).

1.6.2. Trendo ir sezoninės dalies išskyrimas vienu metu

R turi keletą procedūrų, kurios laikinių sekų trendo ir sezoninės dalies išskyrimą atlieka vienu metu – tai funkcijos `decompose` (gali būti taikoma tiek adityviesiems, tiek ir multiplikatyviesiems mo-

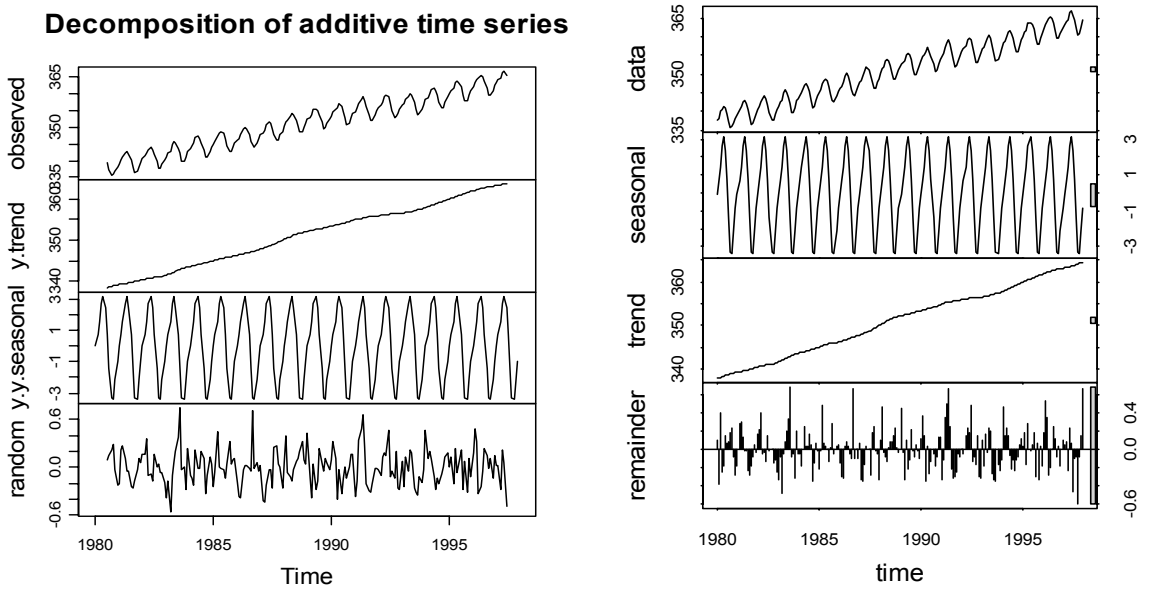
deliams) ir modernesnė `stl` (= decomposition into seasonal and trend components with loess procedure; taikoma tik adityviesiems modeliams) iš `stats` paketo. Funkcija `decompose` trendui išskirti taiko slenkamojo vidurkio, o `stl` – lokaliosios regresijos procedūras.

Adityviuoju modeliu aprašomoms laikinėms sekoms skaidyti geriausiai tinka `stl` funkcija



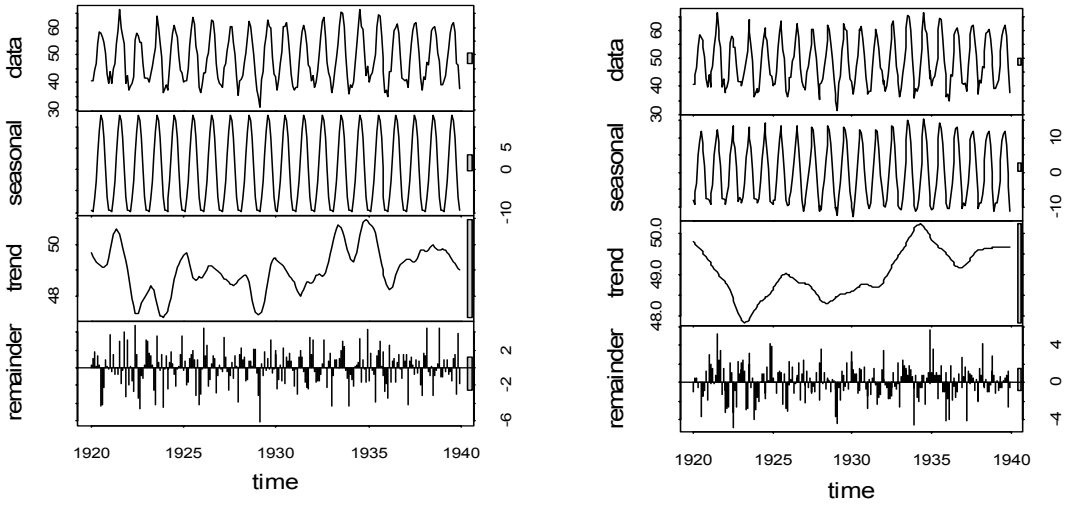
1.35 pav. Dvi laikinės sekos su adityviu sezoniniu dėmeniu (`co2` kairėje ir `nottem` viduryje) ir viena seka su multiplikatyviu sezoniniu daugikliu (`AirPassengers` dešinėje; šios sekos logaritmai aprašomi adityviuoju modeliu)

```
data(co2) # ?co2
CO2=window(co2,start=1980) # dėl vaizdumo, sutrumpinsime seką
m = decompose(CO2)
m$figure # sezoninė dalis (12 reikšmių)
[1] -0.02742839  0.71772786  1.50548828  2.71777995  3.19486328  2.43119141
[7]  0.72861328 -1.41307943 -3.34026693 -3.42826172 -2.14620443 -0.94042318
plot(m)                  # skaidymas su decompose funkcija
plot(stl(CO2, "per")) # skaidymas su stl funkcija (sezoninė dalis visur vienoda)
```



1.36 pav. Laikinės sekos `co2` adityvusis skaidinys su `decompose` funkcija (kairėje) ir `stl` funkcija (dešinėje)

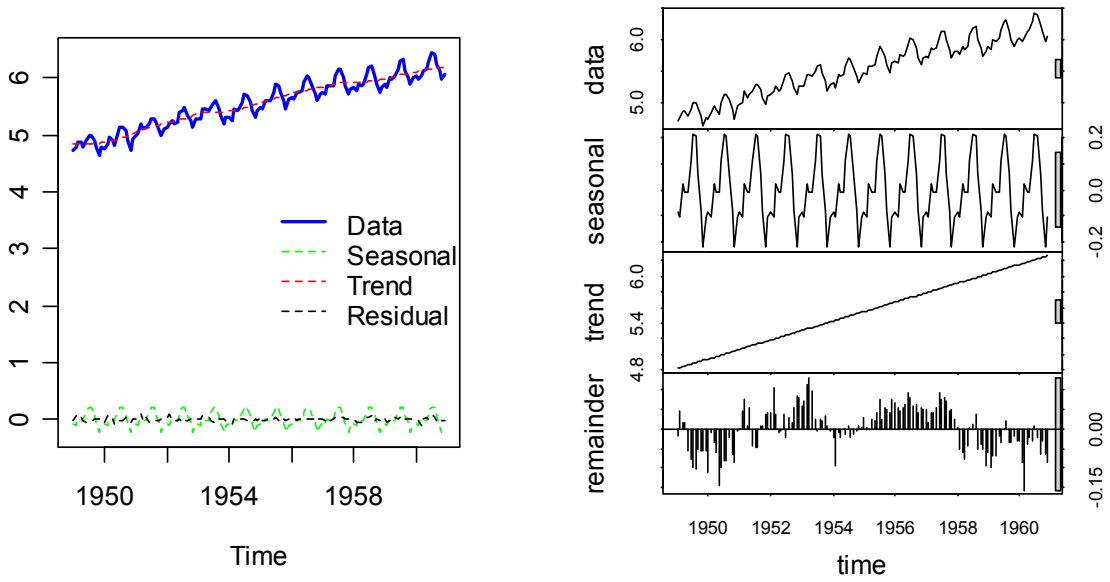
```
plot(stl(nottem, "per"))
plot(stl(nottem, s.window = 4, t.window = 50, t.jump = 1))
```



1.37 pav. `nottem` skaidinys su periodine sezonine komponente (kairėje) ir su kintama (dešinėje)

Dabar panagrinėsime du adityviuoju modeliu aprašomos laikinės sekos `log(AirPassengers)` skaidymo variantus (pačią seką galima būtų skaidyti, pvz., taip : `AP.m=decompose(AirPassengers,type="mult"); plot(AP.m)` arba su kokia kita multiplikatyviojo modelio skaidymo funkcija, žr. `pastecs` paketo aprašymą žemiau).

```
data(AirPassengers)
m <- stl(log(AirPassengers), "per") # per=periodic
ts.plot(log(AirPassengers),
        m$time.series[,1], m$time.series[,2], m$time.series[,3],
        gpars=list(col=c("blue", "green", "red", "black"),
        lwd=c(2,1,1,1), lty=c(1,2,2,2)))
legend(x=1955, y=4, bty="n",
       lwd=c(2,1,1,1), lty=c(1,2,2,2), col=c("blue", "green", "red", "black"),
       legend=c("Data", "Seasonal", "Trend", "Residual"))

plot(stl(log(AirPassengers), s.window="per", t.window=1000))
```



1.38 pav. Du adityviojo modelio `log(AirPassenger)` skaidinio variantai

1.11 UŽDUOTIS. Išnagrinėkite pakete `MASS` esančius duomenis `deaths`, keliais būdais išskirkite trendą ir sezoninę dalį.

1.12 UŽDUOTIS. Išnagrinėkite pakete `forecast` esančius duomenis `fancy`, keliais būdais išskirkite trendą ir sezoninę dalį.

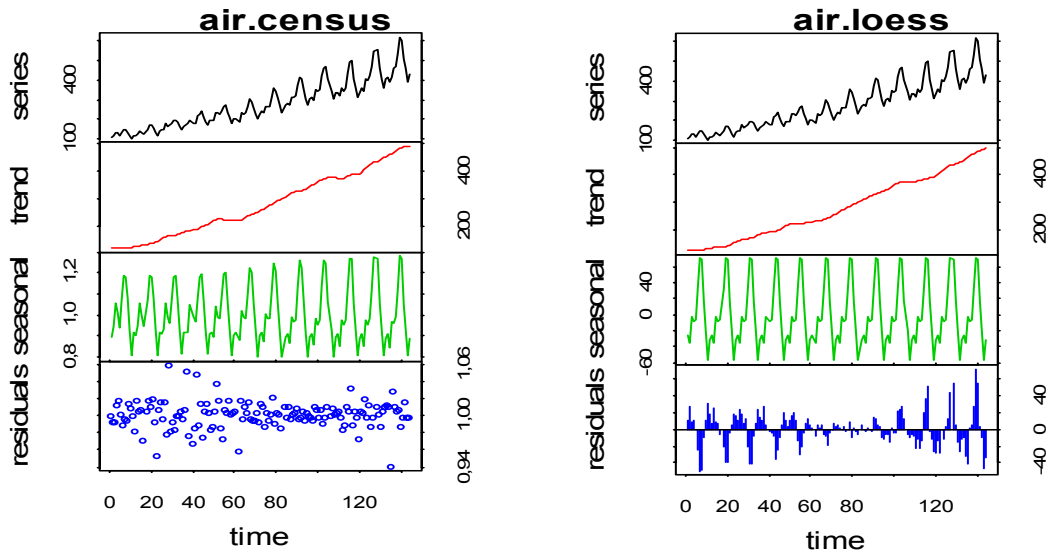
Paminėsime dar kelias laikinių sekų skaidymo funkcijas iš `pastecs` paketo. Tai `decdiff` (time series decomposition using differences), `decevf` (time series decomposition using eigenvector filtering (EVF)), `decloess` (time series decomposition by the LOESS method), `decmedian` (time series decomposition using a running median), `deccensus` (time series decomposition by the CENSUS II method; it decomposes a regular time series into a trend, a seasonal component and residuals, according to a multiplicative model) ir `decreg` (time series decomposition using a regression model). Bet pirmiausiai pateiksime skaidymo metodų apžvalgą (žr. funkcijos `tsd` aprašymą).

Use a decomposition method to split the series into two or more components. Decomposition methods are either series filtering/smoothing (difference, average, median, evf), deseasoning (loess) or model-based decomposition (reg, i.e., regression).

To eliminate trend from a series, use "diff" or use "loess" with `trend=TRUE`. If you know the shape of the trend (linear, exponential, periodic, etc.), you can also use it with the "reg" (regression) method. To eliminate or extract seasonal components, you can use "loess" if the seasonal component is additive, or "census" if it is multiplicative. You can also use "average" with argument `order="periodic"` and with either an additive or a multiplicative model, although the later method is often less powerful than "loess" or "census". If you want to extract a seasonal cycle with a given shape (for instance, a sinusoid), use the "reg" method with a fitted sinusoidal equation. If you

want to identify levels in the series, use the "median" method. To smooth the series, you can use preferably the "evf" (eigenvector filtering), or the "average" methods, but you can also use "median". To extract most important components from the series (no matter if they are cycles -seasonal or not-, or long-term trends), you should use the "evf" method. For more information on each of these methods, see online help of the corresponding `decXXXX()` functions.

```
library(pastecs)
# Skaidymas CENSUS II būdu (standartinė opcija - multiplikatyvusis modelis)
air.census=deccensus(AirPassengers,trend=T)
plot(air.census,col=1:4)
title(main="air.census")

# Skaidymas loess metodu   (standartinė opcija - adityvusis modelis)
air.loess=decloess(AirPassengers,s.window="periodic",trend=T)
plot(air.loess,col=1:4)
title(main="air.loess")
```



1.39 pav. Laikinės sekos `AirPassengers` skaidymas į tris komponentes dviem būdais (atkreipkite dėmesį į *y* ašies mastelius). (Teisingai pasirinkto) multiplikatyviojo modelio paklaidos homoskedastiškos, o adityviojo – ne.

Atkreipiame dėmesį: pradiniai duomenys multiplikatyvūs, todėl teisingas tik `air.census` modelis (`air.loess` modelis pateiktas iliustravimo tikslu). Be to, multiplikatyviojo modelio sezoninė komponentė svyruoja apie 1, o adityviojo – apie 0 (nes $\log 1 = 0$).

1.13 UŽDUOTIS. Tokią pačią analizę atlikite su `log(AirPassengers)` duomenimis. Išbrėžkite 1.1 paveikslą.

****

Šiame skyriuje nagrinėjome įvairius laikinių sekų skaidymo metodus. Jau minėjome, kad pagrindiniai laikinių sekų analizės tikslai yra du: jų aprašymas ir prognozė. Pirmuoju atveju svarbu sudaryti sekos modelį, nustatyti jos didėjimo ir mažėjimo intervalus, rasti minimumo ir maksimumo taškus bei reikšmes ir pan. Antruoju atveju prognozuojame jos būsimas reikšmes ir nurodome jų pasikliau-

ties intervalus. Ne visi skaidymo metodai šioms dviems procedūroms vienodai tinka. Pvz., slenkamojo vidurkio, splainų metodai (funkcija `gam`) ar lokaliosios regresijos metodai (funkcija `loess`) prognozei netinka, nes jie gali aproksimuoti tik istorines reikšmes. Antra vertus, regresiniai, eksponentinio glodinimo ir kai kurie splaininiai metodai leidžia sekos reikšmes „pratęsti“ į ateitį.

Sekų aprašymui tinka visi šiame skyriuje paminėti metodai. Sekų prognozei tinka visos MK procedūros (`lm` ir `nls` funkcijos), eksponentinis glodinimas ir kai kurios splaininės procedūros. ARIMA procesams prognozuoti taikysime `arima` funkciją (žr. 3 skyrių).

2. AR, MA ir ARMA procesai

Laikinę seką išskaidžius į tris dedamasias $y_t=m_t+s_t+e_t$, liekanos e_t turėtų sudaryti stacionarų procesą. Jei šis procesas yra baltasis triukšmas, jo artima ar tolima prognozė yra triviali – tai tiesiog jo vidurkis (t.y., 0). Kitais atvejais prognozė priklauso nuo stacionaraus proceso struktūros. Šiame skyriuje aptarsime kai kurių stacionariųjų procesų klases.

2.1. Baltasis triukšmas ir tiesinės laikinės sekos

Žinoma Wold'o skaidinio teorema teigia, kad „praktiškai kiekvienas" stacionarus (plačiąja prasme) procesas y_t, $t \in \mathbb{Z}$, gali būti užrašytas baltojo triukšmo reikšmių w_t, $t \in \mathbb{Z}$, tiesine kombinacija:

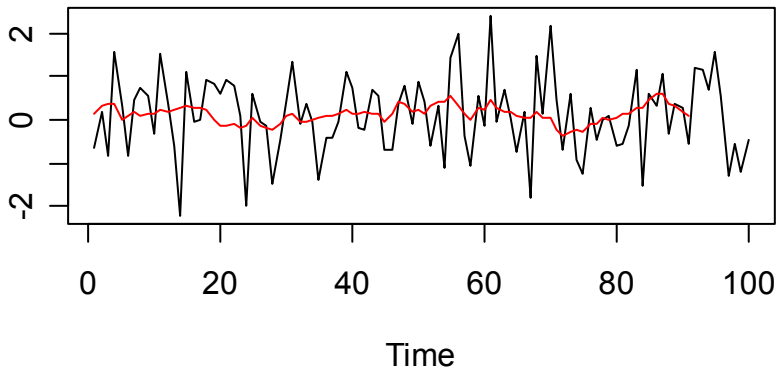
$$y_t = \mu + \sum_{j=0}^{\infty} k_j w_{t-j}, \ \sum k_j^2 < \infty ;$$

skaičius μ (tai y_t vidurkis) paprastai laikomas lygiu nuliui.

2.1 pavyzdys. $k_0 = 1$, o visi kiti k_j lygūs 0. Tuomet $y_t = w_t$, t.y., y_t yra tiesiog baltasis triukšmas.

2.2 pavyzdys. $k_0 = \ldots = k_9 = 0.1$, o visi kiti k_j lygūs 0. Dabar stacionarusis procesas užrašomas pavidalu $y_t = (w_t + \ldots + w_{t-9})/10$, t.y., kaip dešimties baltojo triukšmo narių slenkamasis vidurkis.

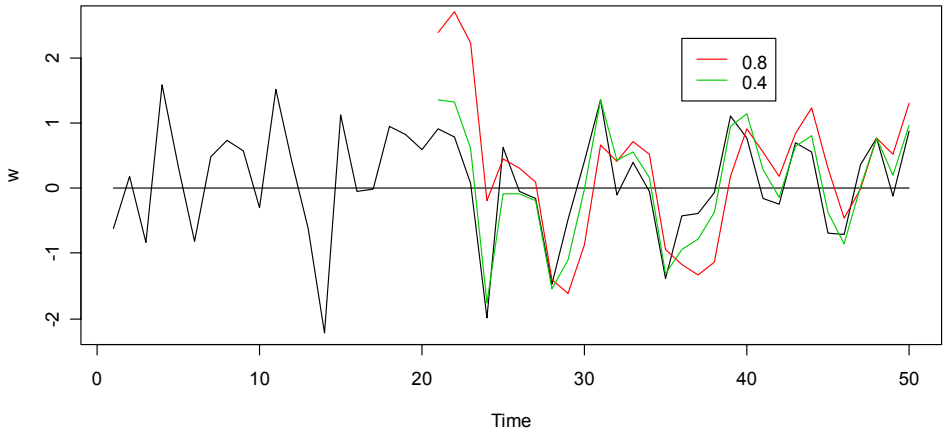
```
set.seed(1)
w=ts(rnorm(100))
x1=ts(w)
x2=ts(w[10:100])
for(i in 9:1) x2=x2+ts(w[i:(90+i)])    # x2/10 bus slenkamasis vidurkis
ts.plot(x1,x2/10,gpars=list(col=1:2))  # gpars=grafical parameters
                                       # arba: x2=filter(w,rep(0.1,10),sides=1)
                                       # ts.plot(x1,x2,gpars=list(col=1:2))
```



2.1 pav. Baltojo triukšmo (juodas) ir suglodinto baltojo triukšmo (raudonas) (vienos realizacijos) grafikai (abu šie procesai stacionarūs, ar ne?); beje, jei vidurkintume ne pagal 10, o pagal 50 narių, raudona kreivė būtų dar labiau panašesnė į konstantą (kodėl?)

2.3 pavyzdys. Koeficientų seka k_j gali būti ir begalinė, pvz., $k_j = a^j, |a|<1$ (modeliuodami imsime dvi baigtines, bet „pakankamai ilgas“ sekas: $k_j = 0.8^j$ ir $k_j = 0.4^j$, $j = 0,1,...,20$). Galima įrodyti, kad šiuo atveju teisinga lygybė $y_t = a y_{t-1} + w_t$ (nuoroda: įsitikinkite, kad procesas $y_t = w_t + a w_{t-1} + a^2 w_{t-2} + ...$ tenkina minėtą lygybę), taigi, praeitame skyriuje minėtas AR(1) procesas, kai $|a|<1$, yra stacionarus.

```
set.seed(1)
w=ts(rnorm(50))
plot(w,ylim=c(-2.2,2.6)) # Baltojo triukšmo grafikas
lines(rep(0,50))         # brėžiame x-sų ašį
k=0.8
lines(filter(w,k^(0:20),sides=1),col=2) #Geltonai pažymėjome koeficientus
k=0.4
lines(filter(w,k^(0:20),sides=1),col=3)
legend(36,2.3,c("0.8","0.4"),lty=1,col=2:3)
```



2.2 pav. Baltasis triukšmas w_t (juodas grafikas); raudono grafiko reikšmė y_t lygi baltojo triukšmo reikšmei w_t plius $0.8\, y_{t-1}$ (žalio grafiko atveju y_t labiau priklauso nuo dabarties“ w_t negu nuo praeities y_{t-1})

▶ Stacionarių procesų vidurkis μ ir dispersija $\sigma^2 (= \sigma_y^2) = \sigma_w^2 \sum_{i=0}^{\infty} k_i^2$ yra pastovūs, todėl visa informacija apie proceso elgesį yra sukaupta (auto)kovariacinėje funkcijoje (ACF) $\gamma(s)$ arba (auto)koreliacinėje funkcijoje $\rho(s)$:

$$\gamma(s) = \mathrm{cov}(y_t, y_{t+s}) = \sigma_w^2 \left(\sum_{i=0}^{\infty} k_i k_{i+s} \right),$$
$$\rho(s) = \mathrm{corr}(y_t, y_{t+s}) = (kam?),\ s = 1,2,...$$

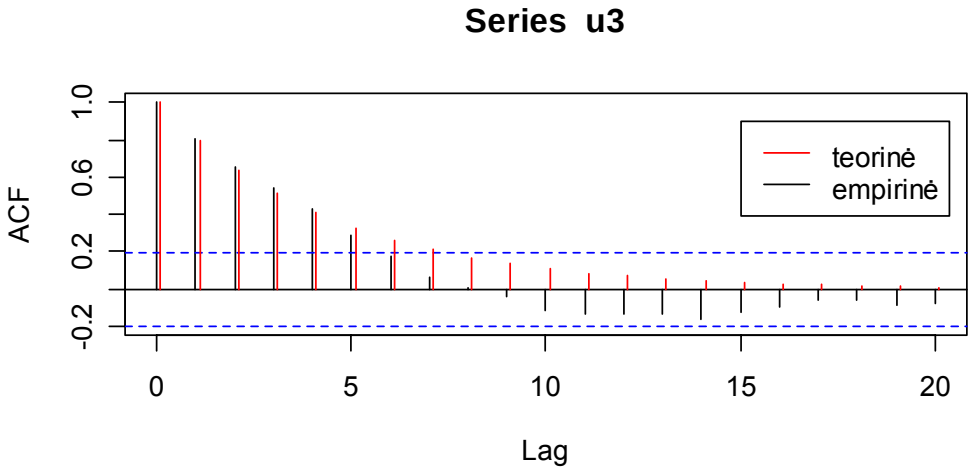
Reikia skirti teorinę kovariacinę funkciją $\gamma(s)$ nuo jos įverčio (nuo empirinės kovariacinės funcijos)

$$c(s) = \frac{1}{n} \sum_{i=\max(1,-s)}^{\min(n-s,n)} (y_{i+s} - \bar{y})(y_i - \bar{y}),$$

kuris tik asimptotiškai (t.y., kai laikinės sekos ilgis n yra „labai didelis“) yra lygus (dėl didžiųjų skaičių dėsnio) $\gamma(s)$. Toks pat teiginys galioja ir koreliacinei funkcijai $\rho(s)$ bei jos įverčiui $r(s) = c(s)/c(0)$.

2.4 pavyzdys. Nesunku įrodyti, kad 2.3 pavyzdyje pateikto proceso (teorinė) kovariacinė funkcija $\gamma(s)$ lygi a^s, tačiau empirinė kovariacinė funkcija $c(s)$ gali pastebimai skirtis nuo jos.

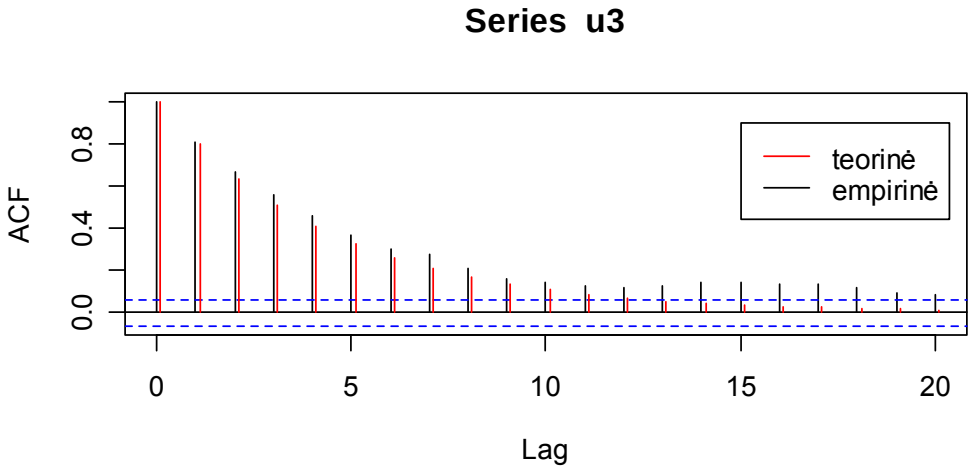
```
u3=filter(ts(rnorm(100)),0.8^(0:20),sides=1) # modeliuojame 2.3 pvz. procesą
acf(u3, na.action = na.pass)                 # brėžiame jo emp. autokovar. f-ją
lines((0:20)+0.1,0.8^(0:20),type="h",col=2)  # brėžiame teorinę autokovar f-ją
legend(15,0.9,c("teorinė","empirinė"),lty=1,col=2:1)
```



2.3 pav. Teorinė ACF pastebimai skiriasi nuo empirinės (n=100)

Antra vertus, jei laikinės sekos ilgis ne 100, o 1000 – empirinė ACF nuo teorinės ACF skiriasi mažiau (dėl didžiųjų skaičių dėsnio).

```
u3=filter(ts(rnorm(1000)),.8^(0:20),sides=1);acf(u3,na.action=na.pass,lag.max=20)
lines((0:20)+0.1,0.8^(0:20),type="h",col=2)
legend(15,0.9,c("teorinė","empirinė"),lty=1,col=2:1)
```



2.4 pav. Teorinė ACF mažai skiriasi nuo empirinės (n=1000)

2.2. ARMA procesai

Paprastai turime tik baigtinį laikinės sekos stebinių skaičių, todėl „atkurti“ (t.y., įvertinti) be galo daug koeficientų k_j nepavyks. Garsusis Box‘o ir Jenkins‘o ARMA modelis siūlo stebimą stacionarų procesą aproksimuoti procesu, aprašomu baigtiniu parametrų skaičiumi.

- Procesas vadinamas AR(p) procesu (*p*-osios eilės autoregresiniu procesu (angl. AutoRegressive process of order *p*), jei

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + w_t$$

- Procesas vadinamas MA(q) procesu (*q*-osios eilės slenkamojo vidurkio procesu (angl. Moving Average process of order *q*)), jei[1]

$$y_t = \sum_{i=0}^{q} b_j w_{t-j},\ b_0 = 1$$

- Procesas vadinamas ARMA(p,q) procesu[2], jei

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=0}^{q} b_j w_{t-j}\ .$$

Konkrečią laikinę seką aprašant ARMA modeliu, modelio parenkamas keliais žingsniais.

1. Modelio (AR, MA ar ARMA)) parinkimas. Jis remiasi pačios laikinės sekos, jos autokoreliacinės ir dalinės autokoreliacinės funkcijų (ACF ir PACF) grafikais.
2. Modelio parametrų apskaičiavimas. Paprastai tai atliekama su `arima` funkcija, nors, pvz., jei nagrinėtume AR procesą, jo parametrus galime apskaičiuoti ir su `ar` funkcija.
3. Modelio ar modelių kritika ir galutinio modelio pasirinkimas. Iš kelių modelių tinkamiausią renkamės pagal jo AIC reikšmę, nors galime remtis ir `tsdiag` funkcija.
4. Dažnai sudarytas modelis naudojamas prognozei - ją galima atlikti su `predict` funkcija (arba su `forecast` funkcija iš `forecast` paketo).

Dauguma ARMA procesų analizei skirtų funkcijų yra `stats` pakete. Kitas funkcijas ir laikinių sekų pavyzdžius galima rasti `MASS`, `tseries` ir `forecast` paketuose.

2.2.1. AR procesai

Labai dažnai dabartinė atsitiktinio proceso reikšmė nėra „visai atsitiktinė“, ji gali priklausyti ir nuo ankstesnių reikšmių. Laikinė seka $y_t,\ t \in T$ [3], tenkinanti lygtį

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + ... + a_p y_{t-p} + w_t,\ t \in T$$

---

[1] Apibrėžiant MA procesą nėra reikalo kartu žinoti proceso w_t dispersiją σ^2 ir β_0 . Mes renkamės β_0=1.

[2] Kai kurie autoriai ARMA procesą apibrėžia kiek kitaip.

[3] Priminsime - diskreti indeksų aibė T paprastai sutampa su sveikų skaičių aibe $\boldsymbol{Z}$ arba su jos poaibiu $\{t_0, t_0+1, t_0+2, ...\}$.

(čia w_t baltasis triukšmas) vadinama p-osios eilės autoregresiniu[4] procesu (žymėsime AR(p)).

Norėdami išsiaiškinti jo savybes, trumpai aptarsime skirtumines lygtis. Tarkime, kad funkcija $f = f(t)$ yra apibrėžta aibėje T. Jei simboliu Δ pažymėtume skirtuminį operatorių:

$$(\Delta f)(t) = ((\Delta^1 f)(t) =)\ f(t+1) - f(t),$$
$$(\Delta^{m+1} f)(t) = \Delta(\Delta^m f)(t),\ m = 1, 2, ...$$

tai p-osios eilės skirtumine lygtimi vadiname lygtį pavidalo $\tilde{G}(\Delta^p f(t), ..., \Delta f(t), f(t), t) = 0$; čia $\tilde{G}$ yra žinoma, o f – ieškomoji funkcijos. Kadangi $\Delta^2 f(t) = f(t+2) - 2f(t+1) + f(t)$ (panašiai galima užrašyti ir $\Delta^m f(t)$), tai skirtuminę lygtį visuomet galima užrašyti pavidalu $G(f(t+p), ..., f(t+1), f(t), t) = 0$. Specialus šios lygties atvejis

$$f(t+p) + \alpha_1 f(t+p-1) + ... + \alpha_p f(t) = 0,\ t \geq 1,\ \alpha_p \neq 0, \qquad (*)$$

vadinamas homogenine p-osios eilės lygtimi su pastoviais koeficientais. Aišku, kad jos sprendinys yra apibrėžiamas p pradinėmis reikšmėmis: jei $f(t_0) = \varphi_0, f(t_0+1) = \varphi_1, ..., f(t_0+(p-1)) = \varphi_{p-1}$, tai $f(t_0+p) = -\alpha_1\varphi_{p-1} - ... - \alpha_p\varphi_0$ ir t.t. Pvz., pirmosios eilės lygties

$$f(t+1) + \alpha_1 f(t) = 0$$

sprendinį[5]

$$f(t) = f(t_0)(-\alpha_1)^{t-t_0} = C_1(-\alpha_1)^t$$

visiškai apibrėžia viena konstanta C_1 (beje, lygtimi ir jos sprendiniu pateikiame du – rekurentinį ir bendrojo nario – geometrinės progresijos apibrėžimus).

Jei (*) lygties dešinė pusė nelygi 0, lygtis vadinama nehomogenine, o jos sprendinys yra lygus bendrojo homogeninės lygties sprendinio (priklausančio nuo p konstantų C_p) ir bet kokio (paprastai vadinamo *atskiruoju*) nehomogeninės lygties sprendinio sumai.

Grįžkime prie AR(p) proceso. Skirsime du atvejus.

1. $T = \boldsymbol{Z}$. Pradėsime procesu AR(1) $y_t - a_1 y_{t-1} = w_t$, $t \in T$. Jau žinome, kad bendrasis homogeninės lygties $y_t - a_1 y_{t-1} = 0$ sprendinys yra $y_t = C_1 A_1^t$; čia A_1 yra *charakteristinės lygties* $A - a_1 = 0$ šaknis (t.y., $A_1 = a_1$), o C_1 – bet koks realusis skaičius. Nesunku įsitikinti, kad funkcija $y_t = w_t + a_1 w_{t-1} + a_1^2 w_{t-2} + ...,\ t \in \boldsymbol{Z}$ tenkina (nehomogeninę) AR(1) lygtį, todėl bendrasis sprendinys atrodo taip:

$$y_t = C_1 A_1^t + w_t + a_1 w_{t-1} + a_1^2 w_{t-2} + ...,\ t \in \boldsymbol{Z}\ .$$

[4] Kadangi regresoriai šiuo atveju yra paties proceso ankstesnės reikšmės, tai šis regresinis modelis vadinamas *auto*regresiniu.
[5] Norint įsitikinti, kad tai sprendinys, užtenka įstatyti šį reiškinį į lygtį.

Lengva apskaičiuoti šio proceso vidurkį ir dispersiją: $Ey_t = C_1 A_1^t$, $Dy_t = \sum_{i=0}^{\infty} a_1^{2i}$. Akivaizdu, kad vidurkis yra pastovus tik tuomet, kai $C_1 = 0$, o dispersija yra baigtinė, kai $|a_1| < 1$. Taigi, jei $|a_1| < 1$, aibėje ***Z*** apibrėžtas AR(1) procesas visuomet turi stacionarų variantą.

Priminsime, kad bendruoju atveju AR(p) procesas y_t yra nusakomas lygtimi $y_t - (a_1 y_{t-1} + ... + a_p y_{t-p}) = w_t$. Bendrasis homogeninės lygties sprendinys yra pavidalo[6] $y_t = C_1 A_1^t + C_2 A_2^t + ... + C_p A_p^t$; čia A_j yra charakteristinės lygties $A^p - (a_1 A^{p-1} + ... + a_{p-1} A + a_p) = 0$ šaknys[7]. Galima įrodyti, kad atskirasis nehomogeninės lygties sprendinys yra pavidalo $y_t = B_0 w_t + B_1 w_{t-1} + B_2 w_{t-2} + ...$; čia koeficientai B_j priklauso nuo visų koeficientų a_j, o formulės gana sudėtingos.

Taigi, kaip ir AR(1) atveju, jei dispersijų eilutė $\sum_{i=0}^{\infty} B_i^2 < \infty$ konverguoja (pvz., jei $p = 2$, tai ši eilutė konverguos, kai $\{(a_1, a_2) : |a_2| < 1, a_1 + a_2 < 1, a_2 - a_1 < 1\}$), egzistuoja stacionarus AR(p) proceso variantas $y_t = B_0 w_t + B_1 w_{t-1} + B_2 w_{t-2} + ...$ (jį gauname, visus koeficientus C_j prilyginę 0).

2. $T = \{t_0, t_0 + 1, ...\}$ Pradėsime procesu AR(1) $y_t - a_1 y_{t-1} = w_t$, $t \in T$. Homogeninė lygtis turi tą patį sprendinį $y_t = y_{t_0} a_1^{t - t_0}$, o atskirasis nehomogeninės lygties sprendinys yra seka, nusakoma formule $y_{t_0+i} = a_1^{i-1} w_{t_0+1} + a_1^{i-2} w_{t_0+2} + ... + a_1 w_{t+(i-1)} + w_{t+i}$, $i \geq 1$. Bendrasis nehomogeninės lygties sprendinys yra šių dviejų reiškinių suma. Skirsime du atvejus.

i. Pradinė sąlyga y_{t_0} yra atsitiktinė. Jei jos skirstinys sutampa su $w_{t_0} + a_1 w_{t_0-1} + a_1^2 w_{t_0-2} + ...$ skirstiniu ir $|a_1| < 1$, AR(1) procesas $y_t = \sum_{i=0}^{\infty} a_1^i w_{t-i}$ yra stacionarus. Kitais atvejais šis procesas nėra stacionarus.

ii. Pradinė sąlyga y_{t_0} nėra atsitiktinė, tai žinomas skaičius. Šį kartą stacionaraus varianto neegzistuoja (apskaičiuokite Ey_t ir Dy_t, kai $y_{t_0} = 0$), tačiau jei $|a_1| < 1$, koks bebūtų y_{t_0}, proceso vidurkis greitai artėja į 0, o dispersija – į $\sigma_w^2 / (1 - a_1^2)$ (kodėl?), taigi procesas $y_{t_0+i} = y_{t_0} a_1^i + a_1^{i-1} w_{t_0+1} + a_1^{i-2} w_{t_0+2} + ... + a_1 w_{t+(i-1)} + w_{t+i}$, $i \geq 1$, gana greitai[8] tampa „praktiškai stacionariu“ (jei visos charakteristinės lygties $A^p - (a_1 A^{p-1} + ... + a_{p-1} A + a_p) = 0$ šaknys bus moduliu mažesnės už 1, tai panašiai bus ir AR(p) atveju).

AR(p) yra stacionarus, jei visos jo charakteristinės lygties $A^p - (a_1 A^{p-1} + ... + a_{p-1} A + a_p) = 0$, $A \in \mathbf{C}$, šaknys moduliu mažesnės už 1

[6] Jei lygtis turi kompleksinių šaknų porą $A_j^{\pm} = r(\cos\varphi_j \pm i \sin\varphi_j)$, tai dėmuo $C_j (A_j^{\pm})^t$ duoda osciliuojantį narį.

[7] Aukščiau užrašytas bendrasis sprendinys bus nurodyto pavidalo, jei (kaip dažniausiai ekonometrijoje ir būna) visos *p* charakteristinio polinomo šaknys yra skirtingos.

[8] Praktinė taisyklė AR(p) procesui yra tokia: kai $i \geq p + \texttt{ceiling}(6/\textit{log(minroots)})$; čia *minroots* yra moduliu mažiausia atvirkštinės charakteristinės lygties šaknis.

2.1 UŽDUOTIS. Paskutinioji sąlyga dažnai formuluojama taip: jei stebime AR(p) procesą ir visos jo atvirkštinės charakteristinės lygties $1-a_1x-...-a_px^p=0, x\in\mathbf{C},$ šaknys (tarp jų gali būti kompleksinių) moduliu didesnės už vieną, tai šis procesas yra stacionarus. Įrodykite, kad abi formuluotės ekvivalenčios. ◀◀

ii. atvejo rezultatas praverčia, prognozuojant AR(1) procesus. Tarkime, žinomi skaičiai $y_1,...,y_{t_0}$ yra šio proceso reikšmės. Tuomet „geriausia“ y_{t_0+i} prognozė yra $\hat{y}_{t_0+i}=E(y_{t_0+i}\,|\,y_1,...,y_{t_0})=y_{t_0}a_1^i$, $i\geq 0$. Prognozė artėja į 0 ir greitai darosi nebeįdomi, tačiau labai svarbus jos elgesys su keliomis pirmosiomis *i* reikšmėmis.

Panašiai prognozuoti galime ir tuomet, kai tiriamasis procesas yra AR(p) procesas $y_t-(a_1y_{t-1}+...+a_py_{t-p})=w_t, T=\{t_0,t_0+1,...\}$.

AR(p) proceso sprendiniai ir prognozė

- Sudarykite homogeninę lygtį ir raskite visus jos homogeninius sprendinius
- Raskite atskirąjį nehomogeninės lygties sprendinį
- Sudarykite bendrąjį sprendinį kaip atskirojo sprendinio ir tiesinės homogeninių sprendinių kombinacijos sumą
- Remdamiesi pradinėmis sąlygomis, eliminuokite laisvąsias konstantas.

2.2 UŽDUOTIS. AR(p) procesus $y_t=a_1y_{t-1}+a_2y_{t-2}+...+a_py_{t-p}+w_t,\ t\in T$ (jų vidurkis lygus 0) galima apibendrinti ir nagrinėti procesus pavidalo $y_t=a_0+a_1y_{t-1}+a_2y_{t-2}+...+a_py_{t-p}+w_t,\ t\in\mathbf{Z}$ (šio proceso vidurkis $\tilde{a}\neq 0$ (ir $\tilde{a}\neq a_0$!)). Aišku, kad centruoto proceso $\tilde{y}_t=y_t-\tilde{a}$ vidurkis jau lygus nuliui, todėl šį procesą galima užrašyti ankstesniu pavidalu. Apskaičiuokite $\tilde{a}$. ◀◀

2.3 UŽDUOTIS. AR(2) procesą $y_t=3+0.9y_{t-1}-0.2y_{t-2}+w_t$ užrašykite centruotu pavidalu. Tarkite, kad $y_{-1}=y_0=0$ ir sumodeliuokite 120 šio proceso reikšmių. Išbrėžkite abiejų procesų grafikus nuo $t=20$. Ar tai stacionarūs procesai? ◀◀

Taigi AR(1) procesas $y_t=a_1y_{t-1}+w_t$ yra stacionarus, jei $|a_1|<1$, o AR(p) – jei atvirkštinės charakteristinės lygties šaknys nepriklauso vienetiniam skrituliui. Stacionaraus AR(p) proceso autokovariacijos koeficientus $\gamma(k)=cov(y_t,y_{t+k})$ tiesiogiai apskaičiuoti nėra lengva, tačiau tikslinga pasiremti tuo faktu, kad jie yra Yule-Walker'io lygčių sistemos

$$\sum_{k=1}^{p}\gamma(k-i)a_k=\gamma(i),\ i=1,...,p,$$

sprendiniai. Antra vertus, kitą svarbią AR(p) proceso charakteristiką - vadinamuosius dalinės (partial) autokovariacijos[9] koeficientus $\delta(k)$ - rasti lengva: tai koeficientai prie y_{t-k}, t.y., tiesiog a_k (taigi koeficientai $\delta(p+1),\delta(p+2),...$ lygūs nuliui).

[9] Dalinės koreliacijos PACF (angl. Partial AutoCorrelation Function) koeficientai $\delta(k)$ apibrėžiami kaip koreliacija tarp y_t ir y_{t+k} po to kai buvo pašalinta jų tiesinė priklausomybė nuo tarpinių kintamųjų y_{t+1}, $y_{t+2},...,y_{t+k-1}$. Paprasčiausiai juos galima apskaičiuoti taip: sudarykime regresinius modelius, $y_t=\vartheta_{11}y_{t-1}+w_t$,

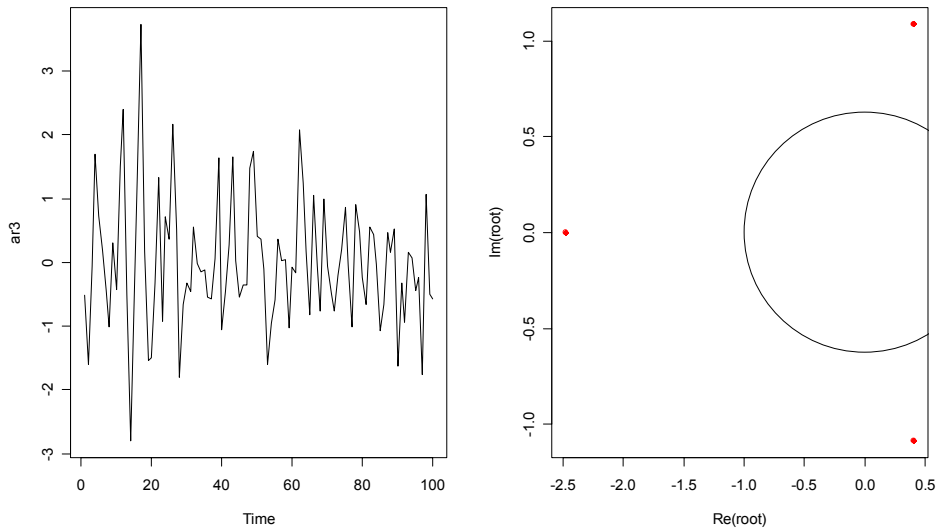
AR(p) proceso teorinė autokoreliacinė funkcija būna maždaug eksponentiniuose rėžiuose, o teorinė dalinė autokovariacija (t.y. ir autokoreliacija) nuo tam tikros vietos (nuo p+1) lygi nuliui. Empirinių analogų elgesys gali kiek skirtis nuo ką tik aprašyto, tačiau minėta informacija gali padėti nustatyti, ar stebima laikinė seka yra AR(p) procesas.

Pailiustruosime aukščiau išdėstytus teiginius. AR(p) procesą galima generuoti su funkcija `arima.sim`.

```
opar=par(mfrow=c(1,2));set.seed(1)
ar3 <- arima.sim(100, model=list(ar=c(0.2,-0.5,-0.3)))
plot(ar3)
```

Generavome AR(3) procesą $y_t = 0.2y_{t-1} - 0.5y_{t-2} - 0.3y_{t-3} + w_t$; jis stacionarus, nes visos trys atvirkštinės charakteristinės lygties šaknys nepriklauso vienetiniam skrituliui:

```
a <- c(0.2,-0.5,-0.3)
root <- polyroot(c(1,-a))   # Randame lygties 1-0.2x+0.5x^2+0.3x^3=0 šaknis
plot(root, col=2,pch=16)    # Pažymime jas plokštumoje
symbols(0,0,circles=1,add=T,inches=F)   # Brėžiame vienetinį apskritimą
par(opar)
```

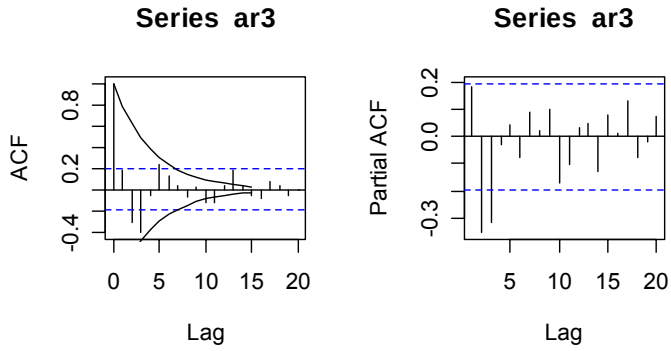


2.5 pav. Generuota AR(3) laikinė seka ir atvirkštinės charakteristinės lygties šaknys (visos jos nepriklauso vienetiniam skrituliui)

Laikinės sekos analizė paprastai pradedama nuo jos grafiko brėžimo (žr. 2.5 pav., kairėje – ši seka, greičiausiai, stacionari), o po to brėžiami autokovariacijų ir dalinių autokovariacijų grafikai:

```
opar=par(mfrow=c(1,2))
acf(ar3,20)
pacf(ar3,20)
par(opar)
```

$y_t = \vartheta_{21}y_{t-1} + \vartheta_{22}y_{t-2} + w_t$, $y_t = \vartheta_{31}y_{t-1} + \vartheta_{32}y_{t-2} + \vartheta_{33}y_{t-3} + w_t$ ir t.t. Tuomet $\delta(1) = \vartheta_{11}$, $\delta(2) = \vartheta_{22}$, $\delta(3) = \vartheta_{33}$ ir t.t.



2.5 pav. Proceso `ar3` empirinės autokoreliacinė (kairėje) – ACF reikšmės lėtai gęsta; dalinė autokoreliacinė funkcijos (dešinėje) - reikšmingi stulpeliai baigiasi taške Lag=3

2.6 pav. grafikuose mėlynos brūkšniuotos linijos rodo apytikslius 95% pasikliovimo intervalus, taigi, sprendžiant pagal PACF funkcijos grafiką, `ar3`, ko gero, yra AR(3) procesas. Jo parametrus tuomet galima nustatyti su `ar` funkcija:

```
> (ar3.YW=ar(ar3)) # Parametrai skaičiuojami Yule-Walker'io metodu

Coefficients:
      1         2         3
 0.1388   -0.2751   -0.3164   # Tikrosios reikšmės yra 0.2,-0.5,-0.3

Order selected 3  sigma^2 estimated as  0.7952
```

Funkcija `ar` dar kartą patvirtina, kad proceso `ar3` eilė (order selected) lygi 3. Ši funkcija proceso eilę parenka taip. Tarkime, kad "tikroji" proceso eilė yra 0, t.y.[10], $y_t = (0+)e_t$. Modelio paklaidos šį kartą sutampa su y_t, o paklaidų kvadratų suma $RSS_0 = \sum_{i=1}^{n} y_i^2$. Dabar tarkime, kad modelio eilė yra 1, t.y., $y_t = a_1 y_{t-1} + e_t$. Yule-Walker'io metodu suradę koeficiento įvertį $\hat{a}_1$, apskaičiojame $RSS_1 = \sum_{i=1}^{n} (y_t - \hat{a}_1 y_{t-1})^2$ ir t.t. Aišku, kad mažesnė RSS reikšmė reiškia geresnį modelį, tačiau didinant modelio eilę, RSS visuomet mažėja, todėl reiktų įvesti baudą už papildomus parametrus. Vienas iš būdų yra Akaike's informacinis kriterijus AIC, kuris yra apibrėžiamas lygybe $AIC = n\log(RSS/n) + 2p + const$. Proceso eilę parenkame pagal AIC minimumą (mūsų atveju išeina, kad eilė lygi 3).

```
> ar(ar3)$aic
        0         1         2         3         4         5         6         7
21.473268 19.980864  8.548799  0.000000  1.912120  3.740170  5.167902  6.387215
        8         9        10        11        12        13        14        15
 8.347221  9.316338  8.350393  9.326652 11.232732 13.022487 13.405825 14.748848
       16        17        18        19        20
16.731929 17.009647 18.434204 20.392554 21.833488
```

Kadangi AIC nustatomas tik konstantos tikslumu, `ar(ar3)$aic` minimalią AIC reikšmę prilygina nuliui.

[10] Funkcija `ar` iš tikrųjų tiria ne procesą y_t, bet $y_t - \bar{y}$.

Nežinomus AR(3) proceso koeficientus galima įverti ir didžiausio tikėtinumo metodu (tariama, kad paklaidų skirstinys yra Gauso):

```
> (ar3.mle=ar(ar3,method="mle"))  # Parametrai skaičiuojami DT metodu

Coefficients:
      1        2        3
 0.1325  -0.2758  -0.3246

Order selected 3  sigma^2 estimated as  0.755
```

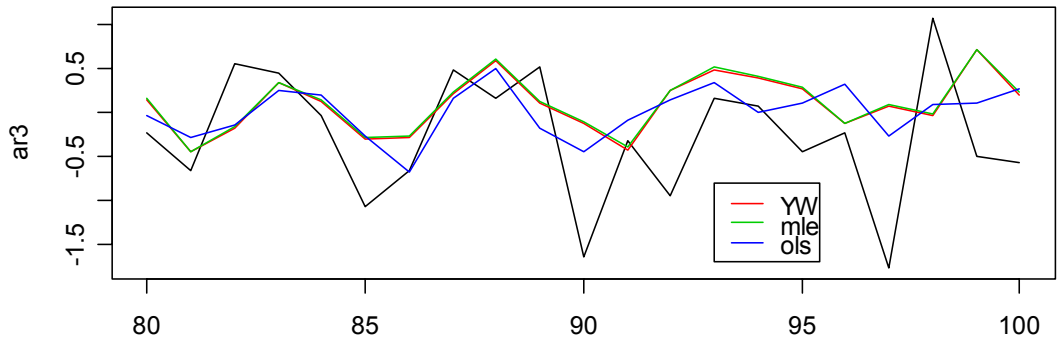
arba standartiniu mažiausių kvadratų metodu:

```
> (ar3.ols=ar(ar3,method="ols")) # Parametrai skaičiuojami Ordinary Least
                                 # Squares metodu
Coefficients:
      1        2        3        4        5        6        7        8
 0.1356  -0.1764  -0.2170  -0.0210   0.0494  -0.0786   0.1348  -0.2431
      9       10       11       12       13       14       15       16
 0.2016  -0.0749  -0.1786   0.0081   0.1125  -0.0857   0.1655  -0.0401
     17
 0.1815

Intercept: -0.0593 (0.08344) # Tai a0 (hipotezė H0:a0=0 neatmestina (kodėl?))
Order selected 17  sigma^2 estimated as  0.5055
```

(modelio eilė dabar net 17(!)). Štai ištrauka iš `?ar.ols`, kurioje teigiama, kad MK metodas AR proceso koeficientams vertinti nelabai tinka : Order selection is done by AIC if `aic` is true. This is problematic, as `ar.ols` does not perform true maximum likelihood estimation. The AIC is computed as if the variance estimate (computed from the variance matrix of the residuals) were the MLE, omitting the determinant term from the likelihood. Note that this is not the same as the Gaussian likelihood evaluated at the estimated parameter values.

Pademonstruosime, kad visi metodai yra panašaus tikslumo, t.y., ar remtumės YW metodu (ir formule $\hat{y}_t - \bar{y} = 0.0424(y_{t-1} - \bar{y}) - 0.4036(y_{t-2} - \bar{y}) - 0.2894(y_{t-3} - \bar{y})$) ar kuriuo kitu, $\hat{y}_t$ reikšmės bus panašios:

```
ind=80:100
plot(ind,ar3[ind],xlab="",ylab="ar3",type="l") # Pradinė laikinė seka (juoda)
lines(ind,(ar3-ar3.YW$resid)[ind],col=2)
lines(ind,(ar3-ar3.mle$resid)[ind],col=3) # Turėtų būti tiksliausias
lines(ind,(ar3-ar3.ols$resid)[ind],col=4)
legend(93,-0.8,c("YW","mle","ols"),lty=1,col=2:4)
```



2.6 pav. Visi trys metodai (raudona, žalia ir mėlyna laužtės) yra panašaus tikslumo.

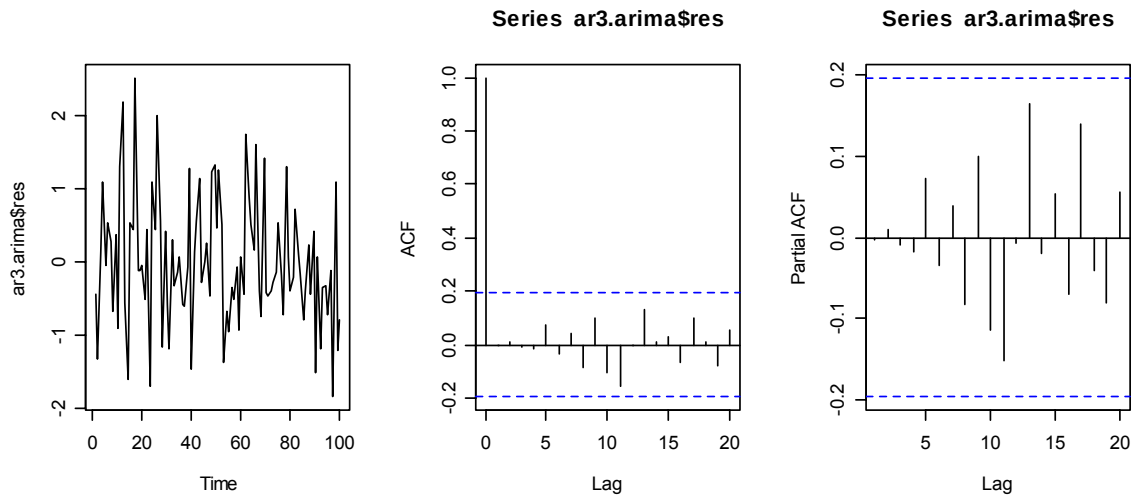
Paminėsime dar vieną funkciją AR proceso parametrams vertinti, būtent, `arima`:

```
> (ar3.arima=arima(ar3,order=c(3,0,0)))

Coefficients:
         ar1      ar2      ar3  intercept
      0.1325  -0.2758  -0.3246     0.0103
s.e.  0.0941   0.0915   0.0948     0.0598

sigma^2 estimated as 0.755:  log likelihood = -128.16,  aic = 266.32
```

Čia `order=c(3,0,0)` reiškia, kad ARIMA proceso (apie juos kalbėsime vėliau) AR dalies eilė turi būti 3, I dalies (apie ją taip pat kalbėsime vėliau) ir MA dalies eilės yra nuliai. Nesunku patikrinti (patikrinkite), kad šiuo modeliu nusakoma prognozė labai panaši į ankstesnes. Sunku pasakyti, kuris iš modelių yra „teisingiausias“ – jie tiesiog remiasi skirtingomis prielaidomis, kurių teisingumą patikrinti sunku.

AR modelių analizė paprastai baigiama likučių analize – modelis yra geras, jei likučiai $y_t - \hat{y}_t$ sudaro baltąjį triukšmą (yra „visai atsitiktiniai“).

```
opar=par(mfrow=c(1,3))
plot(ar3.arima$res)
acf(ar3.arima$res)
pacf(ar3.arima$res)
par(opar)
```



2.7 pav. Modelio `ar3.arima` likučių, ACF ir PACF grafikai

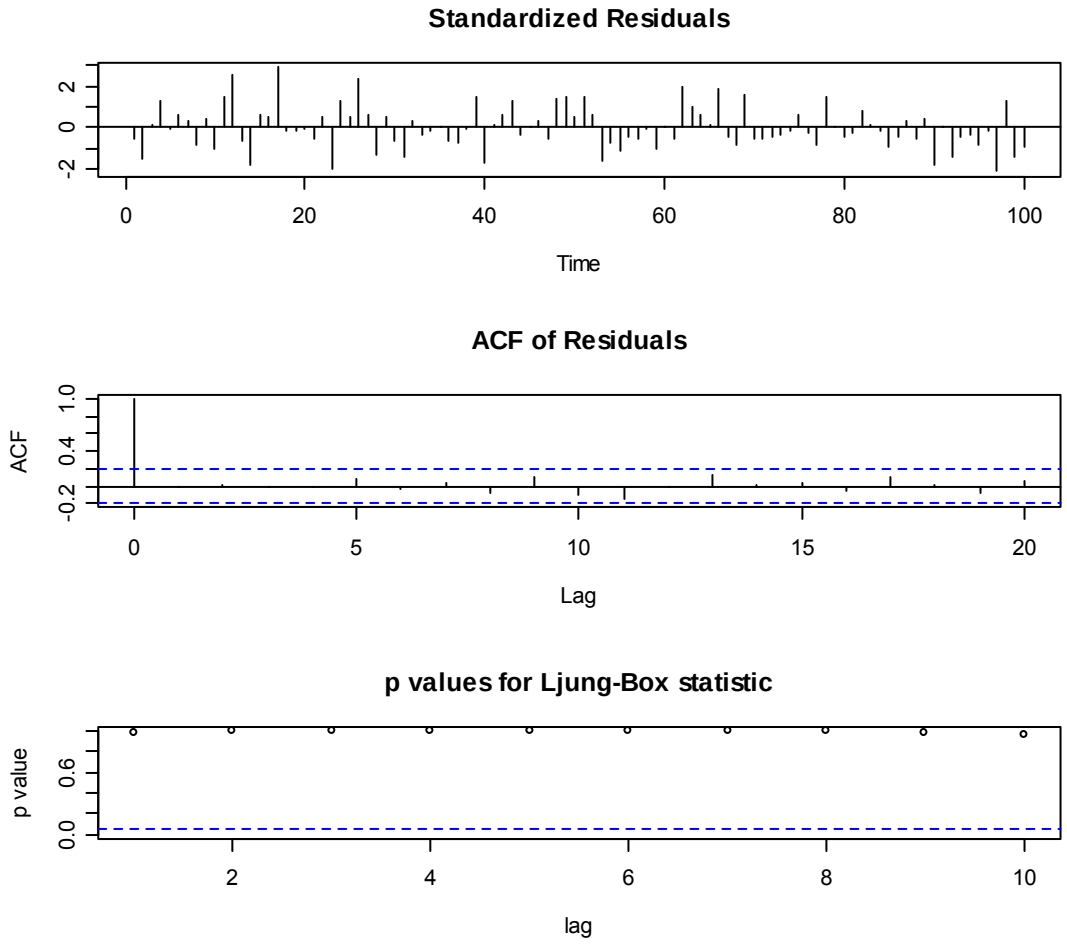
Matome, kad ACF grafikas neprieštarauja hipotezėms, kad kiekviena likučių autokoreliacija lygi nuliui[11], t.y., kad jie sudaro baltąjį triukšmą. Antra vertus, jei vienas ar keli ACF stulpeliai būtų iššokę iš mėlynos juostos, būtų galima pradėti diskusiją apie tai, kad „šie nežymūs iššokiai dar neįrodo, kad tai ne baltasis triukšmas“ ir t.t. Norint išvengti tokio neapibrėžtumo, paprastai taikomi jungtiniai (angl. portmanteau) testai, kuriuos neformaliai galima užrašyti taip: H_0: *likučių procesas sudaro baltąjį triukšmą* su alternatyva H_1: *yra ne taip* arba, tiksliau, H_0: $\rho_1 = \rho_2 = ... = \rho_m = 0$ su alternatyva H_1: *bent vienas* ρ_i, $i = 1,...,m$, *lygus 0*. Galima įrodyti, kad tikslingiausia imti (testas galingiausias, kai) $m \approx 10\log_{10} n$, o testo statistika pasirinkti sumą $Q(m) = n(n+2)\sum_{i=1}^{m} r_i^2/(n-i)$ (Ljung‘as ir Box‘as įrodė, kad tuomet, kai teisinga hipotezė H_0, $Q(m)$ turi χ_m^2 skirstinį).

```
> Box.test(ar3.arima$res,lag=10*log10(length(ar3.arima$res)),type="Ljung")

        Box-Ljung test

X-squared = 12.6973, df = 20, p-value = 0.89 # Nėra pagrindo atmesti H0
```

Paprasčiau šią diagnostikos procedūrą galima atlikti su `tsdiag` funkcija, kuri Ljung’o ir Box’o statistikos reikšmę apskaičiuoja kelioms (`Lag=`) *m* reikšmėms iš karto:

```
tsdiag(ar3.arima) # Argumentas turi būti Arima klasės objektas
```

[11] Visos (empirinės) `acf` funkcijos reikšmės yra „mėlynos“ juostos viduje.



2.8 pav. Modelio `ar3.arima` likučių diagnostikos procedūra (visos Ljung'o ir Box'o statistikos *p* reikšmės didesnės už 0,05)

Taigi galima tvirtinti, kad modelis `ar3.arima` yra visai priimtinas.

Kaip žinia, stacionaraus AR proceso atveju, y_t ateities prognozė $\hat{y}_t$ gana greitai artėja prie konstantos[12] $\mu\,(\equiv EX_t)$. Prognozę galima atlikti su `predict` funkcija.

```
opar=par(mfrow=c(1,2))
ar3.fore=predict(ar3.YW,n.ahead=20) # Prognozuojame 20 žingsnių į priekį
ts.plot(ar3,ar3.fore$pred,ar3.fore$pred+2*ar3.fore$se,ar3.fore$pred-
2*ar3.fore$se,lty=c(1,2,3,3))
abline(0,0,col=2)
```

Panašų paveikslą, bet su paprastesne sintakse, galime išbrėžti taip:

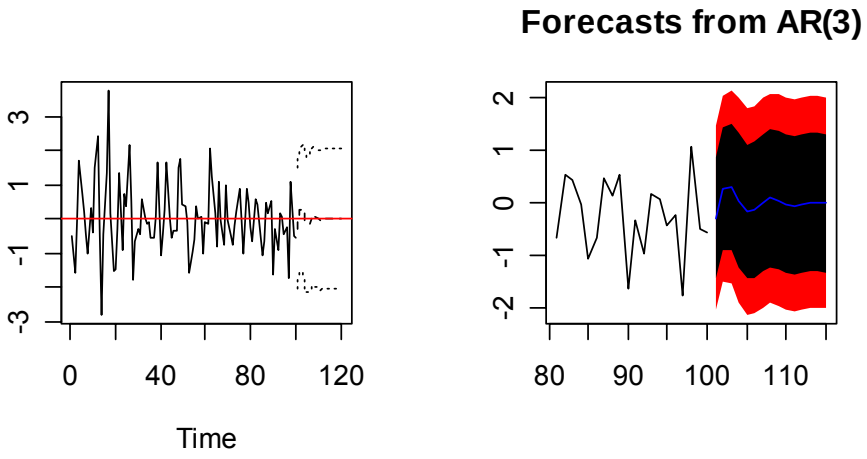
```
library(forecast)
plot(forecast(ar3.YW,15),include=20) # Grafiką brėšime tik su 20-čia istorinių
par(opar)                            # (turimų) reikšmių
```

Atkreipkite dėmesį į prognozuojamų reikšmių elgesį: jos artėja į 0 svyruodamos, kas reiškia, kad tarp charakteristinės lygties šaknų yra kompleksinių (beje, šis svyravimas yra labai svarbus

[12] Mūsų atveju, tai 0 (prisiminkite, iš kur atsirado `ar3`).

prognozuotojui!). Paskutiniu teiginiu nėra sunku įsitikinti: jei AR(3) proceso `ar3` koeficientus vertintume Yule-Walker'io metodu, procesas būtų aprašomas lygtimi $y_t = 0.1388 y_{t-1} - 0.2751 y_{t-2} - 0.3164 y_{t-2} + w_t$, jo charakteristinė lygtis būtų $A^3 - 0.1388 A^2 + 0.2751 A + 0.3164 = 0$, o tarp jos šaknų yra kompleksinių:

```
>  round(polyroot(c(0.3164,0.2751,-0.1388,1)),3)
[1]  0.327+0.711i -0.516+0.000i  0.327-0.711i
```



2.9 pav. Procesas `ar3.YW` ir jo prognozė 20 žingsnių į priekį kartu su $\pm 2*$se pasikliauties intervalu (kairėje – tolima prognozė artėja į konstantą ir darosi mažai įdomi); dešinėje toks pats grafikas išbrėžtas su `forecast` funkcija

2.4 UŽDUOTIS. Patikrinkite ar `ar(ar3.YW)$resid` yra baltas triukšmas (visos `pacf` reikšmės ir visos `acf` reikšmės išskyrus nulinę turi būti mėlynoje juostoje).

2.5 UŽDUOTIS. Žemiau pateikti JAV ketvirtiniai nedarbo lygio duomenys (su pašalintu sezoniškumu – kaip tai atliekama?) nuo 1948 m. 1-ojo ketvirčio iki 1978 m. 1-ojo ketvirčio imtinai.

```
Unemp=c(3.73, 3.67, 3.77, 3.83, 4.67, 5.87, 6.7, 6.97, 6.4, 5.57, 4.63, 4.23,
3.5, 3.1, 3.17, 3.37, 3.07, 2.97, 3.23, 2.83, 2.7, 2.57, 2.73, 3.7, 5.27, 5.8,
5.97, 5.33, 4.73, 4.4, 4.1, 4.23, 4.03, 4.2, 4.13, 4.13, 3.93, 4.1, 4.23, 4.93,
6.3, 7.37, 7.33, 6.37, 5.83, 5.1, 5.27, 5.6, 5.13, 5.23, 5.53, 6.27, 6.8, 7,
6.77, 6.2, 5.63, 5.53, 5.57, 5.53, 5.77, 5.73, 5.5, 5.57, 5.47, 5.2, 5, 5, 4.9,
4.67, 4.37, 4.1, 3.87, 3.8, 3.77, 3.7, 3.77, 3.83, 3.83, 3.93, 3.73, 3.57, 3.53,
3.43, 3.37, 3.43, 3.6, 3.6, 4.17, 4.8, 5.17, 5.87, 5.93, 5.97, 5.97, 5.97, 5.83,
5.77, 5.53, 5.27, 5.03, 4.93, 4.77, 4.67, 5.17, 5.13, 5.5, 6.57, 8.37, 8.9,
8.37, 8.4, 7.63, 7.43, 7.83, 7.93, 7.37, 7.07, 6.9, 6.63, 6.2)
```

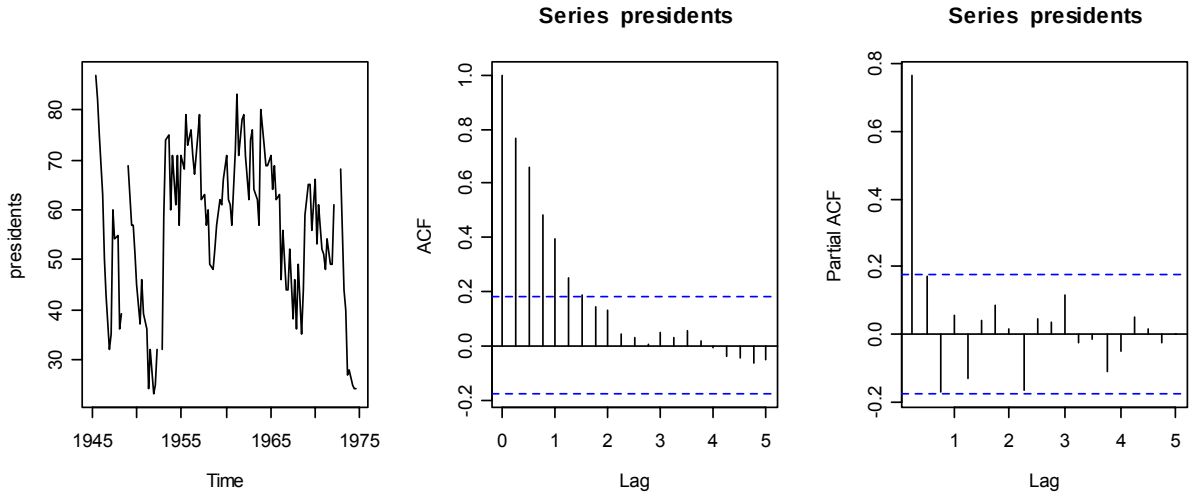
Suteikite šiems duomenims laikinės sekos struktūrą ir įrodykite, kad šiuos duomenis gerai aprašo AR(2) procesas.

2.6 UŽDUOTIS. Sugalvokite nestacionaraus AR proceso pavyzdį ir išbrėžkite jo grafiką.

2.5 pavyzdys. Duomenų rinkinyje `presidents` yra pateikti JAV prezidentų populiarumo ketvirtinių apklausų rezultatai.

```
data(presidents)
```

```
?presidents
> presidents # yra trūkstamų duomenų(!)
     Qtr1 Qtr2 Qtr3 Qtr4
1945   NA   87   82   75
1946   63   50   43   32
.........................
opar=par(mfrow=c(1,3))
plot(presidents)
acf(presidents, na.action = na.pass)
pacf(presidents, na.action = na.pass)
par(opar)
```



2.10 pav. Laikinė seka `presidents` (tai JAV prezidentų populiarumo duomenys), jos ACF ir PACF grafikai

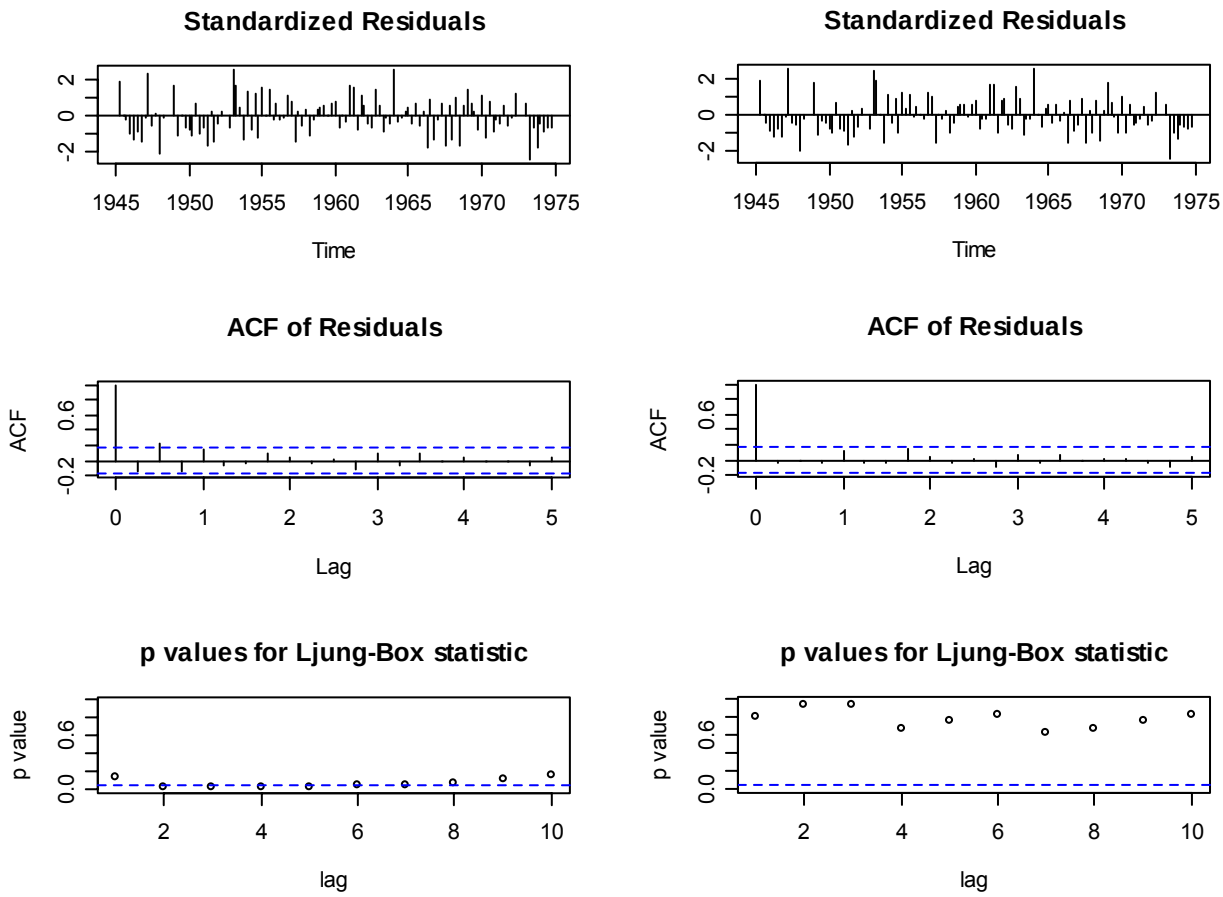
ACF grafikas gęsta eksponentiškai, o PACF grafikas tikrai turi reikšmingą komponentę, kai Lag=1/4 (kadangi `presidents` dažnis lygus 4 (kaip tą sužinoti?), tai Lag=1/4 iš tikrųjų atitinka vieną ketvirtį, o, pvz., Lag=3/4 atitinka 3 ketvirčius). Antra vertus, PACF reikšmės taškuose 2/4 ir 3/4 irgi įtartinos (įtartina ir reikšmė taške 2,25, bet paprastai tiek toli nežiūrima), todėl be AR(1) galime nagrinėti ir AR(3) procesus.

```
>(fit1 <- arima(presidents, c(1, 0, 0)))

Coefficients:
         ar1  intercept
      0.8242    56.1505
s.e.  0.0555     4.6434
sigma^2 estimated as 85.47:
log likelihood = -416.89,  aic = 839.78

> tsdiag(fit1)
```

Kadangi AR(1) modelis nėra patenkinamas (žr. 2.11 pav., kairėje), išbandysime AR(3) modelį.

```
>(fit3 <- arima(presidents, c(3, 0, 0)))  # AIC mažesnė

Coefficients:
         ar1     ar2      ar3  intercept
      0.7496  0.2523  -0.1890    56.2223
s.e.  0.0936  0.1140   0.0946     4.2845

sigma^2 estimated as 81.12:  log likelihood = -414.08,  aic = 838.16
```

```
> tsdiag(fit3)
```



2.11 pav. Laikinės sekos `presidents` modelių AR(1) (kairėje) ir AR(3) (dešinėje) diagnostika; Ljug-Box statistikos grafikas signalizuoja, kad AR(1)modelio likučiai nėra (o AR(3) – yra) baltasis triukšmas

Kadangi modelio AR(3) aic reikšmė mažesnė už AR(1), o likučiai elgiasi geriau, pasirinksime trečios eilės modelį. ◀

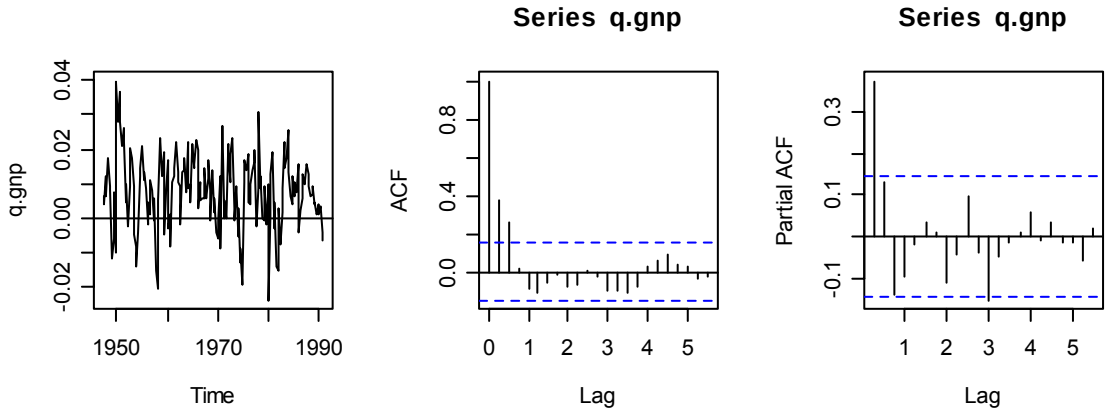
2.6 pavyzdys. Data\Tsay direktorijos faile q-gnp.dat yra pateiktos JAV ketvirtinės bendrojo nacionalinio produkto prieaugio normos (sezoniškai pataisytos, nuo 1947 m. 2-ojo ketvirčio iki 1991 m. 1-ojo ketvirčio).

```
> (q.gnp=ts(scan(file.choose()),start=1947.25,freq=4))
         Qtr1     Qtr2     Qtr3     Qtr4
1947            0.00632  0.00366  0.01202
1948  0.00627  0.01761  0.00918  0.00820
1949 -0.01170 -0.00587  0.00757 -0.00992
.....................................
1990  0.00420  0.00108  0.00358 -0.00399
1991 -0.00650
```

Išbrėšime pagrindinius grafikus.

```
opar=par(mfrow=c(1,3))
```

```
plot(q.gnp);abline(0,0)
acf(q.gnp);pacf(q.gnp);par(opar)
```



2.12 pav. Tai, kad `q.gnp` beveik visą laiką yra teigiamas (brėžinys kairėje), reiškia, jog JAV ekonomika beveik visą laiką augo

Sekančiame skyrelyje išdėstyta teorija teigia, kad procesas su tokiomis ACF ir PACF gali būti MA(2) procesas (PACF gęsta eksponentiškai, o ACF turi dvi reikšmingas komponentes). Antra vertus, Tsay [T, 34 psl.] siūlo šią laikinę seką interpretuoti kaip AR(3) procesą (nes ACF gęsta eksponentiškai, o PACF turi tikrai vieną, o gal ir tris reikšmingas komponentes). Pabandykime įgyvendintid pastarąją idėją.

```
> (gnp.mle=ar(q.gnp,method="mle"))

Coefficients:
      1         2         3
 0.3480    0.1793   -0.1423
Order selected 3  sigma^2 estimated as  9.427e-05

> (mm=mean(q.gnp))
[1] 0.00774125
```

Reikia turėti galvoje, kad `ar` funkcija centruoja dėmenis, t.y. gautasis modelis iš tikro yra toks:

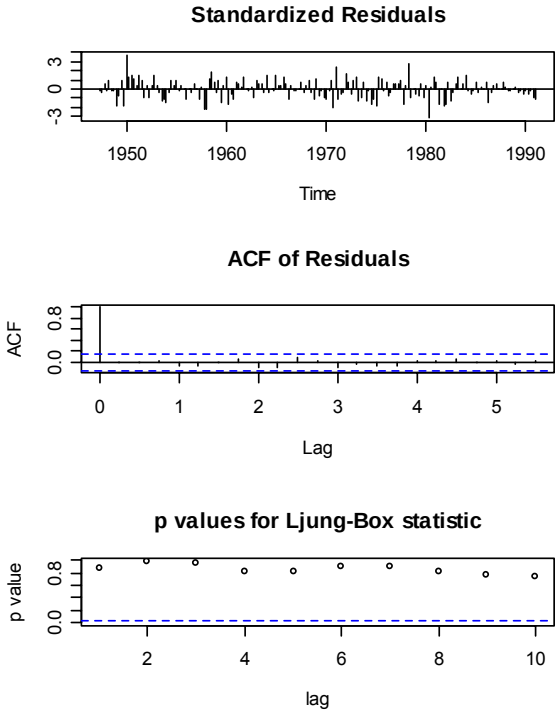
$$y_t - mm = 0.35(y_{t-1} - mm) + 0.18(y_{t-2} - mm) - 0.14(y_{t-3} - mm) + w_t$$

arba

$$y_t = 0.0047 + 0.35y_{t-1} + 0.18y_{t-2} - 0.14y_{t-3} + w_t,$$

taigi, tiksliai toks, koks pateiktas [T]. Diagnostikos procedūra sako, kad AR(3) yra geras modelis:

```
tsdiag(arima(q.gnp,order=c(3,0,0)))
# tsdiag funkcija reikalauja, kad jos argumentas būtų Arima klasės, todėl
# vietoje ar funkcijos naudojame funkciją arima (ar sutampa šie du modeliai?)
```



2.13 pav. `q.gnp` aprašančio AR(3) modelio diagnostika (visos *p* reikšmės didesnės už 0,05)

2.7 UŽDUOTIS. Prognozuokite `q.gnp` keliems metams į priekį. Suraskite internete tikrus duomenis ir palyginkite juos su prognoze.

2.8 UŽDUOTIS. Data\Tsay direktorijos m-vw.dat faile patalpintos CRSP value-weighted indekso mėnesinės paprastosios grąžos (nuo 1926 m. sausio iki 1997 m. gruodžio).

```
vw=ts(scan(file.choose()),start=1926,freq=12)
round(vw,3)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1926  0.000 -0.032 -0.065  0.037  0.015  0.056  0.028  0.028  0.002 -0.028  0.028  0.029
1927  0.002  0.045  0.007  0.007  0.057 -0.020  0.075  0.026  0.047 -0.039  0.069  0.023
.....................................................................
```

Interpretuokite šią laikinę seką kaip AR procesą ir nustatykite jo eilę. Pakartokite analizę su `arima` funkcija. Kai kurie autoregresijos koeficientai nėra reikšmingi, todėl dar kartą atlikite `arima` procedūrą, pvz., taip[13]: `arima(vw,order=c(9,0,0),fixed=c(NA,0,NA,0,NA,0,0,0,NA,NA))`. Kuris iš šių modelių geresnis AIC prasme? Pakartokite analizę su duomenimis nuo 1926 m. sausio iki 1996 m. gruodžio. Prognozuokite 1997-ųjų metų duomenis ir palyginkite su tikraisiais.

2.2.2. MA procesai

Nustatėme, kad AR(p) procesas „gerais atvejais“ yra stacionarus (tačiau tai ne baltasis triukšmas). Pateiksime dar vieną stacionaraus proceso pavyzdį : procesas

[13] `fixed` opcijoje reikia nurodyti tiek koeficientų, kiek jų yra modelyje (modelyje `arima(vw,order=c(9,0,0))` jų būtų, įskaitant laisvąjį narį, 10). NA reiškia, kad šis koeficientas bus vertinamas, o 0 – kad atitinkamas AR koeficientas bus 0.

$$y_t = \mu + b_0 w_t + b_1 w_{t-1} + ... + b_q w_{t-q},\ b_0 = 1,$$

yra vadinamas q-osios eilės slenkamojo vidurkio (angl. Moving Average) procesu (jis žymimas simboliu MA(q)). Žemiau pateiktame langelyje matome, kad MA procesas tam tikra prasme yra antisimetriškas AR procesui.

MA(q) procesas yra visuomet stacionarus, jo vidurkis lygus μ, pirmos q autokoreliacijos nelygios nuliui, o tolimesnės – nuliai:

$$\rho(k) = \begin{cases} \sum_{t=0}^{q-|k|} b_t b_{t+|k|} / \sum_{t=0}^{q} b_t^2, & |k| \le q \\ 0, & |k| > q \end{cases},$$

MA proceso dalinės autokoreliacijos gęsta eksponentiškai.

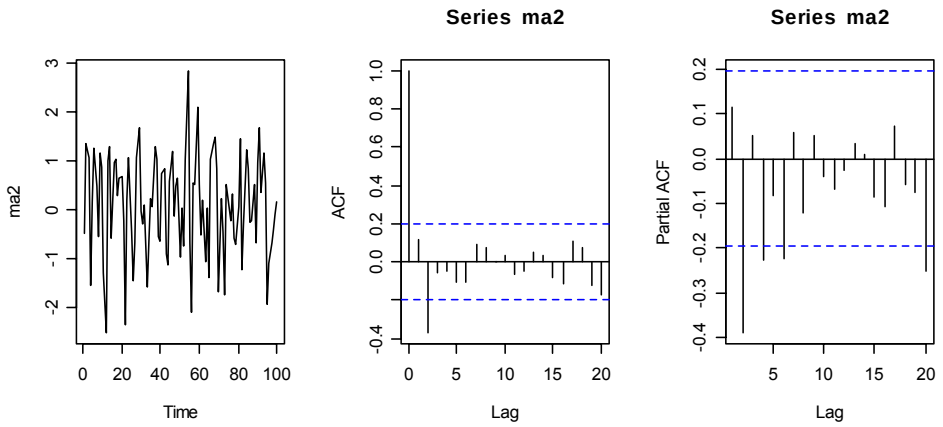
Modeliuosime vieną MA proceso pavyzdį:

```
set.seed(1)
ma2 <- arima.sim(100,model=list(ma=c(0.2,-0.5))) # yt=wt+0,2wt-1-0,5wt-2
```

Jei dabar „užmirštume“ `ma2` kilmę, tai iš gautų grafikų

```
opar=par(mfrow=c(1,3))
plot(ma2);acf(ma2);pacf(ma2)
par(opar)
```

matyti, kad stebimas procesas, ko gero, yra MA(2) procesas.



2.14 pav. Sumodeliuoto proceso `ma2`, jo ACF ir PACF grafikai; kadangi ACF turi tik du reikšmingus stulpelius, o PACF gęsta eksponentiškai, stebimasis procesas, ko gero, yra MA(2) procesas

MA proceso parametrams vertinti vartojame `arima` funkciją (pasirinksime opciją `order=c(0,0,2)`, kas reiškia, kad ieškosime MA(2) (t.y., ARIMA(0,0,2)) proceso koeficientų).

```
> (ma2.arima=arima(ma2,order=c(0,0,2),include.mean=FALSE))
```

```
Series: ma2
ARIMA(0,0,2) model

Coefficients:
         ma1      ma2
      0.1673  -0.5398
s.e.  0.0967   0.1026

sigma^2 estimated as 0.8066:  log likelihood = -131.56,  aic = 269.13
```

(priminsime, kad tikrosios koeficientų reikšmės yra 0,2 ir -0,5).

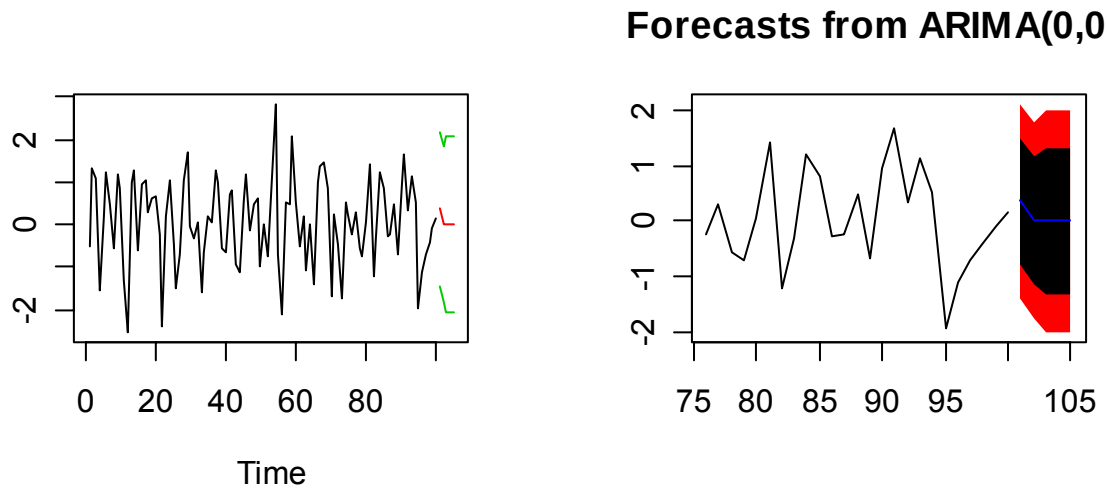
2.9 UŽDUOTIS. Kokią proceso `ma2` eilę rekomenduoja *AIC* kriterijus? (Parašykite ciklą, kuris atspausdintų `aic` reikšmes, kai argumento `order` MA parametro eilė yra 0, 1, 2, 3, 4, 5). Išsiaiškinkite paketo `forecast` funkciją `auto.arima` ir išbandykite `auto.arima(ma2, d=0,D=0,max.p=5,max.q=5,max.P=2,max.Q=2,max.order=5,alpha=0.05)`. ◀

MA procesų prognozuojamos reikšmės labai greitai artėja į nulį, todėl prognozuoti verta tik maždaug q žingsnių į priekį. Tai galima atlikti su

```
opar=par(mfrow=c(1,2))
ma2.fore=predict(ma2.arima,n.ahead=5)
ts.plot(ma2,ma2.fore$pred,ma2.fore$pred+2*ma2.fore$se,ma2.fore$pred-
2*ma2.fore$se,col=c(1,2,3,3))
```

arba su

```
plot(forecast(ma2.arima,5),include=25)
par(opar)
```



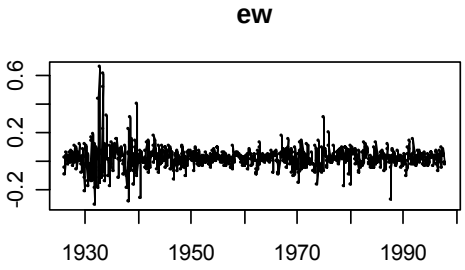
2.15 pav. `ma2` laikinės sekos prognozė

2.7 pavyzdys. Data\Tsay direktorijos m-ew.dat faile patalpintos CRSP equal-weighted[14] indekso mėnesinės paprastosios grąžos (nuo 1926 m. sausio iki 1997 m. gruodžio).

[14] 2.7 užduotyje nagrinėjome CRSP value-weighted indeksą.

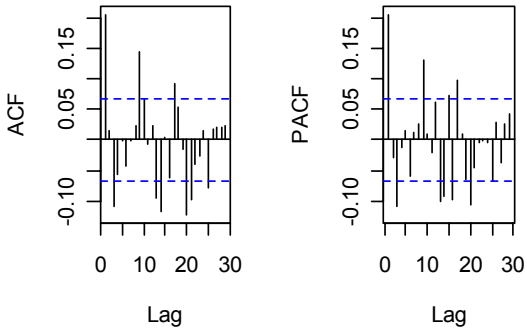
```
(ew=ts(scan(file.choose()),start=1926,freq=12)) # suraskite minėtą failą
> round(ew,3)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1926  0.023 -0.053 -0.097  0.031  0.004  0.051  0.013  0.028 -0.006 -0.031  0.024  0.026
1927  0.011  0.057 -0.019  0.009  0.062 -0.019  0.059  0.004  0.033 -0.038  0.097  0.030
.....................................
```

Išbrėšime pagrindinius `ew` grafikus. Vietoje to, kad kartotume standartines komandas, kreipsimės į `tsdisplay` komandą iš `forecast` paketo.




```
library(forecast)
tsdisplay(ew)
```

Pagrindiniai grafikai (žr. 2.16 pav.) gana komplikuoti. Kadangi PACF grafike reikšmingų stulpelių labai daug, patogu galvoti, kad ji gęsta eksponentiškai, o ACF turi 9 (gal būt 10) nenulines reikšmes, t.y. `ew` yra MA(9) procesas.



2.16 pav. `ew` laikinės sekos pagrindiniai grafikai

```
> arima(ew, order=c(0,0,9))

Coefficients:
          ma1     ma2      ma3      ma4     ma5      ma6     ma7      ma8     ma9  intercept
      0.2055  0.0487  -0.1323  -0.0382  0.0310  -0.0328  0.0320  -0.0352  0.1547     0.0132
s.e.  0.0338  0.0342   0.0355   0.0363  0.0344   0.0334  0.0401   0.0386  0.0344     0.0030
sigma^2 estimated as 0.005224:  log likelihood = 1043.82,  aic = -2065.65
```

Geltonai pažymėti koeficientai yra nereikšmingi (intervalas *koeficientas* ± *2s.e.* uždengia nulį), sudarykime modelį be jų:

```
> (ew.009=arima(ew,order=c(0,0,9),fixed=c(NA,0,NA,0, 0,0,0,0,NA,NA)))

Coefficients:
         ma1  ma2      ma3  ma4  ma5  ma6  ma7  ma8     ma9  intercept
      0.180    0  -0.1319    0    0    0    0    0  0.1373     0.0132
s.e.  0.031    0   0.0362    0    0    0    0    0  0.0327     0.0029

sigma^2 estimated as 0.005282:  log likelihood = 1039.1,  aic = -2068.21
```

Prisiminę, kad ACF ir PACF grafikai komplikuoti, pabandysime modelį parinkti su `auto.arima`.

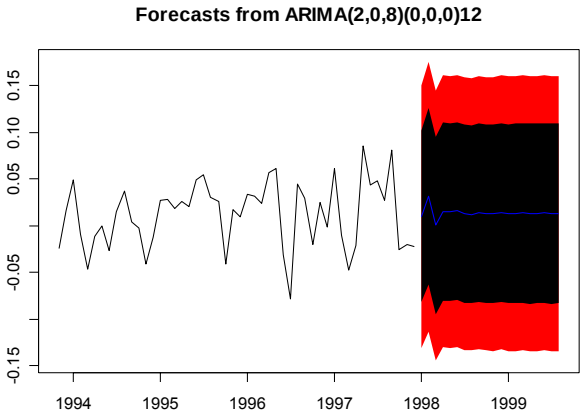
```
> auto.arima(ew, d=0, D=0, max.p=10, max.q=10, max.P=0, max.Q=0, max.order=10, alpha=0.05)
Coefficients:
         ar1      ar2     ma1     ma2     ma3      ma4      ma5      ma6      ma7      ma8  intercept
     -0.9811  -0.7962  1.2148  1.0858  0.1246  -0.1122  -0.1206  -0.0591  -0.0387  -0.1193     0.0132
s.e.  0.0707   0.0652  0.0759  0.0898  0.0673   0.0662   0.0616   0.0596   0.0558   0.0397     0.0026

sigma^2 estimated as 0.00516:  log likelihood = 1049.04,  aic = -2074.08
```

`ma6` ir `ma7` koeficientai tikrai nereikšmingi, pašalinkime juos:

```
> (ew.best=arima(ew,order=c(2,0,8),fixed=c(NA,NA,NA,NA,NA, NA,NA,0,0,NA,NA)))

Coefficients:
          ar1      ar2     ma1     ma2     ma3      ma4      ma5  ma6  ma7      ma8  intercept
      -0.9832  -0.7970  1.2150  1.0821  0.1193  -0.0961  -0.0791    0    0  -0.1024     0.0132
s.e.   0.0627   0.0644  0.0673  0.0892  0.0701   0.0554   0.0395    0    0   0.0230     0.0028

sigma^2 estimated as 0.005166:  log likelihood = 1048.53,  aic = -2077.05
```

Taigi `ew` aprašysime `ew.best` modeliu (nes jo AIC mažiausiais).

```
> plot(forecast(ew.best,h=20),include=50)
```



2.17 pav. `ew` grafikas ir jo prognozė 20-čiai mėnesių į priekį (prognozė greitai artėja į konstantą)

Nesunku įsitikinti (su `tsdiag` funkcija), kad ARMA(2,8) modelis `ew.best`, t.y., $y_t = 0.0132 - 0.9832 y_{t-1} - 0.7970 y_{t-2} + w_t + 1.2150 w_{t-1} + 1.0821 w_{t-2} + 0.1193 w_{t-3} - \ldots - 0.1024 w_{t-8}$, yra visai priimtinas.

2.10 UŽDUOTIS. Aprašykite `ew` su MA(10) modeliu. Ar šis modelis geresnis už tik ką sudarytąjį?

Apie AR ir MA

- Jei PACF grafiko stulpeliai nuo p+1-ojo artimi nuliui – stebimoji laikinė seka, ko gero, yra AR(p) procesas
- Jei ACF grafiko stulpeliai nuo q+1-ojo artimi nuliui – stebimoji laikinė seka, ko gero, yra MA(q) procesas
- MA laikinė seka visuomet stacionari, o AR tik tuomet, kai jos charakteristinės šaknys nepriklauso vienetiniam skrituliui
- Stacionarių sekų prognozė konverguoja į sekos vidurkį, o prognozės paklaidų dispersija – į sekos dispersiją

2.2.3. ARMA procesai

Jau žinome, kad stacionarūs procesai gali būti užrašyti kaip baigtinės arba begalinės sekos: $y_t = \mu + \sum_{j=0}^{\infty} k_j w_{t-j}$, $\sum k_j^2 < \infty$. Kadangi pagal baigtinę imtį neįmanoma įvertinti be galo daug

parametrų, paprastai begalinę koeficientų seką $1, k_2, k_3, ...$ bandoma pakeisti baigtine[15]. Paprasčiausia būtų tarti, kad $k_{q+1} = k_{q+2} = ... = 0$ (tuomet šis procesas vadinamas MA(q) procesu). Jei $k_j = a_1^j, |a_1| < 1$, procesas priklauso tik nuo vieno parametro a_1 ir gali būti užrašytas pavidalu $y_t = a_1 y_{t-1} + w_t$ - tai vadinamasis AR(1) procesas. Šį procesą patogu užrašyti su postūmio operatoriais $L: Ly_t = y_{t-1}$, $L^j y_t = y_{t-j}, j \in \mathbb{Z}$. Pvz., AR(1) procesas dabar gali būti užrašytas $(1 - a_1 L) y_t = w_t$ arba, prisiminus geometrinės progresijos sumos formulę[16],

$$y_t = \frac{1}{1 - a_1 L} w_t = (1 + a_1 L + (a_1 L)^2 + ...) w_t = w_t + a_1 w_{t-1} + a_1^2 w_{t-2} + ...$$

Taigi, jei procesas $\sum_{j=0}^{\infty} k_j L^j w_t$ gerai aproksimuojamas procesu $\sum_{j=0}^{q} b_j L^j w_t, b_0 = 1$ - jis vadinamas MA(q) procesu, jei procesu $\frac{1}{1 - a_1 L - a_2 L^2 - ... - a_p L^p} w_t$ - AR(p) procesu, o jei procesu $\frac{1 + b_1 L + ... + b^q L^q}{1 - a_1 L - a_2 L^2 - ... - a_p L^p} w_t$ - ARMA(p,q) procesu[17]. Priminsime, kad ARMA(p,q) procesą taip pat galima užrašyti pavidalu

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=0}^{q} b_j w_{t-j} \text{ arba } (1 - a_1 L - ... - a_p L^p) y_t = 1 \cdot w_t + (b_1 L + ... + b_q L^q) w_t .$$

Šiuos ARMA modelius visuomet galima papildyti laisvuoju nariu. Pvz., stacionarų AR(1) procesą galima užrašyti taip:

$$y_t = \frac{a_0 + w_t}{1 - a_1 L} = \frac{a_0}{1 - a_1} + w_t + a_1 w_{t-1} + a_1^2 w_{t-2} + ...$$

Kitaip sakant, jei AR(1) proceso laisvasis narys lygus a_0, tai šis procesas svyruoja apie konstantą (taigi proceso vidurkis lygus) $a_0/(1 - a_1)$. Bendresniu AR(p), p≥1, atveju galima samprotauti taip – jei $y_t = a_0 + a_1 y_{t-1} + ... + a_p y_{t-p}$, tai pusiausvyrinė reikšmė y^* (pereikite prie ribos, kai $t \to \infty$) tenkins lygtį $y^* = a_0 + a_1 y^* + ... + a_p y^*$, t.y., $y^* = a_0/(1 - (a_1 + ... + a_p))$. Būtent prie šio skaičiaus stacionariu atveju (ir eksponentiniu greičiu) artėja AR(p) prognozė (ją rasite, tarę, kad $w_{t+1} = w_{t+2} = ... = 0$). Antra vertus, jei procesas yra MA(q): $y_t = a_0 + \sum_{j=0}^{q} b_j L^j w_t, b_0 = 1$, tai ilgalaikei pusiausvyrai a_0 pasiekti užtenka $q+1$ žingsnio (paimkite $w_{t+1} = w_{t+2} = ... = w_{t+q+1} = 0$).

2.11 UŽDUOTIS. Generuokite[18]

[15] T.y., pradinį stacionarųjį procesą aproksimuoti irgi stacionariu, bet paprastesniu procesu.
[16] Čia gana drąsiai elgiamės su operatoriais, tačiau mūsų veiksmus galima pagrįsti.
[17] Polinomą dalinti iš polinomo nėra lengva, todėl ir parametrų vertinimas gana komplikuotas. Be to, čia dar reikalaujama, kad skaitiklis ir vardiklis neturėtų bendrų šaknų (jei turėtų, pvz., vieną bendrą šaknį, tai šį modelį vertėtų pakeisti modeliu ARMA(p-1,q-1)).
[18] Tai galima atlikti su `arima.sim` (ši funkcija generuoja procesą su nuliniu vidurkiu, todėl, reikalui esant, teks apskaičiuoti nurodyto proceso vidurkį ir jį pridėti prie generuotos sekos) arba tiesiog pagal apibrėžimą, užrašius reikalingą R ciklą.

- baltąjį triukšmą su 200 stebinių
- MA(2) procesą su 150 stebinių (pvz., $y_t = 0.3 + (1 - 1.3L + 0.4L^2)w_t$)
- AR(1) procesą su 100 stebinių (pvz., $(1 - 0.7L)y_t = 1.8 + w_t$)
- ARMA(1,1) procesą su 300 stebinių (pvz., $(1 - 0.9L)y_t = 0.7 + (1 - 0.5L)w_t$)

Išbrėžkite procesų, jų ACF ir PACF grafikus ir „nustatykite“ proceso modelį. Fiksuokite atsitiktinės procedūros pradžią (su `set.seed(4)`, nežymiai pakeiskite vieną kurį koeficientą ir pažiūrėkite, kokią įtaką procesų trajektorijoms turi ši kaita). ◀

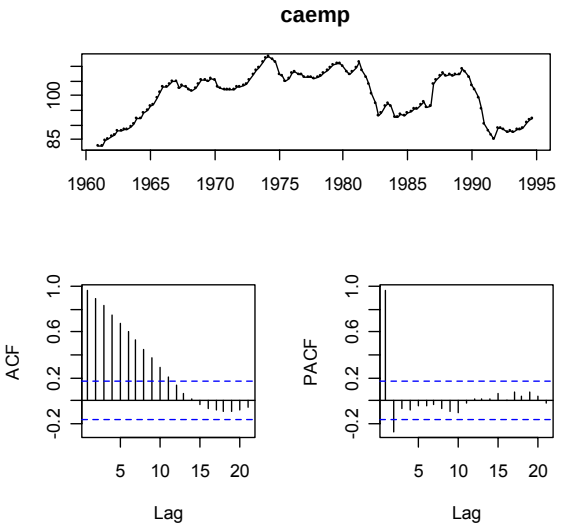
ARMA modelių identifikavimas nėra lengvas, kiek vėliau mes pasiūlysime automatizuotą (AIC minimumu pagrįstą) modelio parinkimo procedūrą. Bendra taisyklė yra tokia: jei stacionaraus proceso abi, ACF ir PACF, funkcijos turi „daug“ reikšmingų stulpelių, procesas ko gero yra ARMA tipo.

2.8 pavyzdys. Žemiau yra pateikti Kanados nedarbo indekso (su sezonine pataisa) ketvirtiniai duomenys nuo 1961 m. 1-ojo ketvirčio iki 1994 m. 4-ojo ketvirčio.

```
caemp = structure(c(83.09, 82.8, 84.634, 85.377, 86.198, 86.579, 88.05, 87.925, 88.465, 88.398, 89.449, 90.556,
92.272, 92.15, 93.956, 94.811, 96.583, 96.965, 98.995, 101.138, 102.882, 103.095, 104.006, 104.777, 104.702,
102.564, 103.558, 102.986, 102.098, 101.472, 102.551, 104.022, 105.094, 105.195, 104.594, 105.813, 105.15,
102.899, 102.355, 102.034, 102.014, 101.836, 102.019, 102.734, 103.134, 103.263, 103.866, 105.393, 107.081,
108.414, 109.297, 111.496, 112.68, 113.061, 112.377, 111.244, 107.305, 106.679, 104.678, 105.729, 107.837,
108.022, 107.282, 107.017, 106.045, 106.371, 106.05, 105.841, 106.045, 106.651, 107.394, 108.669, 109.629,
110.262, 110.921, 110.74, 110.049, 108.19, 107.058, 108.025, 109.713, 111.41, 108.765, 106.289, 103.918, 100.8,
97.4, 93.244, 94.123, 96.197, 97.275, 96.456, 92.674, 92.854, 93.43, 93.206, 93.956, 94.73, 95.567, 95.546,
97.095, 97.757, 96.161, 96.586, 103.875, 105.094, 106.804, 107.787, 106.596, 107.31, 106.897, 107.211, 107.135,
108.83, 107.926, 106.299, 103.366, 102.03, 99.3, 95.305, 90.501, 88.098, 86.515, 85.114, 89.034, 88.823, 88.267,
87.726, 88.103, 87.655, 88.4, 88.362, 89.031, 91.02, 91.673, 92.015), .Tsp = c(1961, 1994.75, 4), class = "ts")
```

Mes ne kartą brėžėme pagrindinius laikinės sekos grafikus. Būtų neblogai parašyti paprastą funkciją, kuri automatizuotų tą procedūrą. Pasirodo, kad tokia funkcija yra `forecast` pakete.

```
library(forecast)
tsdisplay(caemp)
```



2.18 pav. Kanados nedarbo indekso pagrindiniai grafikai

Nagrinėsime du variantus: `caemp` yra AR(2) procesas (nes yra tik du reikšmingi PACF stulpeliai) arba modelį dar papildysime MA nariais ir nagrinėsime (kol kas neaiškios eilės) ARMA procesą.

```
> (caemp200=arima(caemp,order=c(2,0,0)))

Coefficients:
         ar1      ar2  intercept
      1.4505  -0.4762    97.4979
s.e.  0.0749   0.0762     4.3940

sigma^2 estimated as 2.022:  log likelihood = -242.79,  aic = 493.58
```

Diagnostiniai grafikai (surinkite `tsdiag(caemp200)` arba `tsdisplay(caemp200$res)`) rodo, kad šis modelis visai priimtinas. Vis dėlto pabandykime modelį ARMA(3,1).

```
> (caemp301=arima(caemp,order=c(3,0,1)))

Coefficients:
         ar1     ar2      ar3     ma1  intercept
      0.6453  0.7218  -0.4127  0.7898    97.5622
s.e.  0.6235  0.8582   0.2623  0.6487     4.4389

sigma^2 estimated as 2.013:  log likelihood = -242.52,  aic = 497.05
```

Šio modelio AIC truputį didesnis, be to `ar2` ir `ma1` šaknys beveik lygios, todėl, ko gero, šis ARMA(3,1) modelis iš tikrųjų sutampa su ARMA(2,0)=AR(2) modeliu. Dėl viso pikto, pabandykime automatizuotą procedūrą, kuri perrinks visus ARMA modelius iš nurodyto sąrašo.

```
> library(forecast)
> auto.arima(caemp, d=0, D=0, max.p=4, max.q=4, max.P=0, max.Q=0, max.order=6)
Coefficients:
         ar1      ar2  intercept
      1.4505  -0.4762    97.4979
s.e.  0.0749   0.0762     4.3940

sigma^2 estimated as 2.022:  log likelihood = -242.79,  aic = 493.58
```

Funkcija `auto.arima` sudaro įvairius modelius ir apskaičiuoja jų AIC. Ši funkcija, apskritai kalbant, nagrinėja sezoninius ARIMA modelius, užrašomus formule[19] ARIMA(p,d,q)×$(P,D,Q)_s$, tačiau mes imsime $d = 0$ (t.y., proceso nediferencijuosime), o visus sezoninės dalies koeficientus prilyginsime nuliui.

Prognozės grafiką brėšime kartu su laikine seka (bet tik nuo 1985 m. 1-ojo ketvirčio[20] – taip bus vaizdžiau).

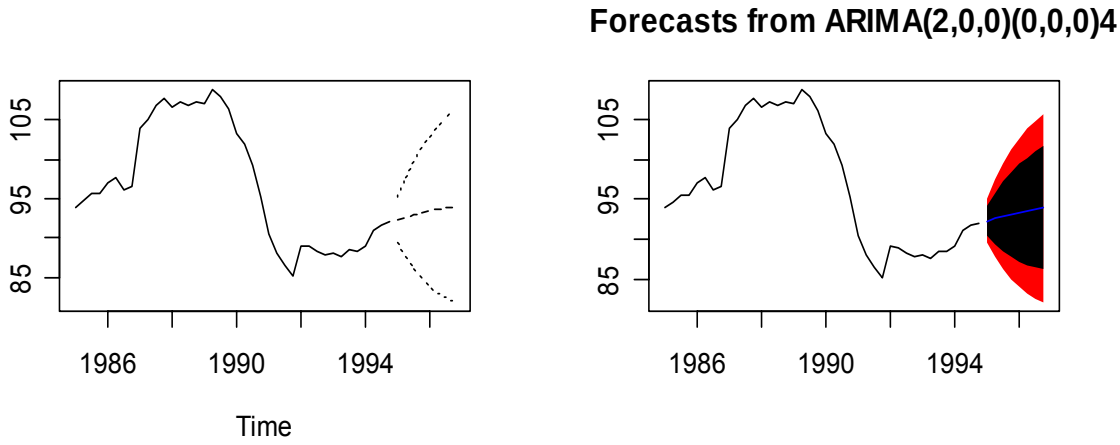
```
> (caemp200.fore=predict(caemp200,n.ahead=8))
$pred
         Qtr1     Qtr2     Qtr3     Qtr4
1995 92.31890 92.59684 92.85526 93.09775
1996 93.32641 93.54260 93.74729 93.94124
```

[19] Smulkiau šiuos modelius aptarsime sekančiame skyriuje.
[20] Vektoriaus poaibį sudarome su `[...]` funkcija, o laikinės sekos – su `window` funkcija (ši funkcija išsaugo laikinę sekos struktūrą).

```
$se
         Qtr1     Qtr2     Qtr3     Qtr4
1995 1.421946 2.505180 3.410708 4.156141
1996 4.770925 5.282165 5.711565 6.075754

opar=par(mfrow=c(1,2))
ts.plot(window(caemp,1985,c(1994,4)),caemp200.fore$pred,caemp200.fore$pred +
2*caemp200.fore$se, caemp200.fore$pred-2*caemp200.fore$se,lty=c(1,2,3,3))
```

arba tiesiog

```
plot(forecast(caemp200,8), include=40)
par(opar)
```



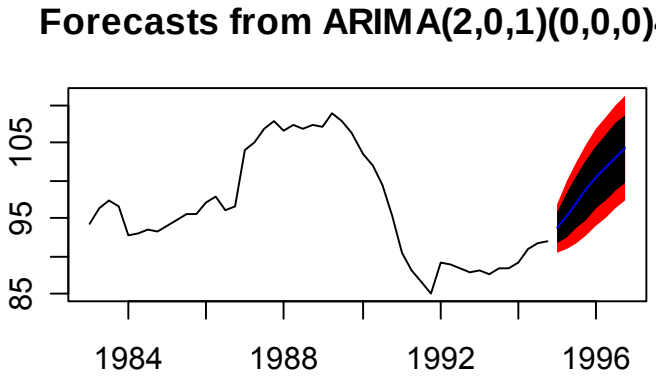
2.19 pav. caemp ir jo prognozės grafikas

Dabar pakartosime prognozės procedūrą, bet nagrinėsime tik duomenis nuo 1983 m. 1-ojo ketvirčio (šioje srityje duomenys yra homogeniškesni).

```
> caemp.n=window(caemp,1983,c(1994,4))
> caemp.n.fit=auto.arima(caemp.n, d=0, D=0, max.p=4, max.q=4, max.P=0, max.Q=0,
max.order=6)

Coefficients:
         ar1      ar2      ma1  intercept
      1.9417  -0.9672  -1.0000    97.4628
s.e.  0.0242   0.0249   0.0706     0.6893
sigma^2 estimated as 2.644:  log likelihood = -94.3,  aic = 198.6
```

Matome, kad dabar geriausias yra ARMA(2,1) modelis.

```
> plot(forecast(caemp.n.fit,h=8))
```



2.20 pav. `caemp` prognozė pagal 1983-1994 m. duomenis

Abi prognozės pastebimai skiriasi, tačiau pasakyti, kuri yra „teisingesnė“, neįmanoma (nebent turėtume informacijos, kad nuo 1983 m. „kažkas“ pasikeitė – tuomet būtų tikslinga remtis antruoju modeliu):

```
> predict(arima(caemp,order=c(2,0,0)),n.ahead=8)$pred
         Qtr1     Qtr2     Qtr3     Qtr4
1995 92.31890 92.59684 92.85526 93.09775
1996 93.32641 93.54260 93.74729 93.94124
> predict(arima(caemp.n,order=c(2,0,1)),n.ahead=8)$pred
          Qtr1      Qtr2      Qtr3      Qtr4
1995  93.68336  95.39312  97.09941  98.75895
1996 100.33103 101.77853 103.06872 104.17392
```

2.12 UŽDUOTIS. Žemiau pateikti JAV alkoholinių gėrimų mėnesinių pardavimų duomenys nuo 1968.01 iki 1993.12.

```
liquor=structure(c(480, 467, 514, 505, 534, 546, 539, 541, 551, 537, 584, 854, 522, 506,
558, 538, 605, 583, 607, 624, 570, 609, 675, 861, 605, 537, 575, 588, 656, 623, 661, 668,
603, 639, 669, 915, 643, 563, 616, 645, 703, 684, 731, 722, 678, 713, 725, 989, 687, 629,
687, 706, 754, 774, 825, 755, 751, 783, 804, 1139, 711, 693, 790, 754, 799, 824, 854,
810, 798, 807, 832, 1142, 740, 713, 791, 768, 846, 884, 886, 878, 813, 840, 884, 1245,
796, 750, 834, 838, 902, 895, 962, 990, 882, 936, 997, 1305, 866, 805, 905, 873, 1024,
985, 1049, 1034, 951, 1010, 1016, 1378, 915, 854, 922, 965, 1014, 1040, 1137, 1026, 992,
1052, 1056, 1469, 916, 934, 987, 1018, 1048, 1086, 1144, 1077, 1036, 1076, 1114, 1595,
949, 930, 1045, 1015, 1091, 1142, 1182, 1161, 1145, 1119, 1189, 1662, 1048, 1019, 1129,
1092, 1176, 1297, 1322, 1330, 1263, 1250, 1341, 1927, 1271, 1238, 1283, 1283, 1413, 1371,
1425, 1453, 1311, 1387, 1454, 1993, 1328, 1250, 1308, 1350, 1455, 1442, 1530, 1505, 1421,
1485, 1465, 2163, 1361, 1284, 1392, 1442, 1504, 1488, 1606, 1488, 1442, 1495, 1509, 2135,
1369, 1320, 1448, 1495, 1522, 1575, 1666, 1617, 1567, 1551, 1624, 2367, 1377, 1294, 1401,
1362, 1466, 1559, 1569, 1575, 1456, 1487, 1549, 2178, 1423, 1312, 1465, 1488, 1577, 1591,
1669, 1697, 1659, 1597, 1728, 2326, 1529, 1395, 1567, 1536, 1682, 1675, 1758, 1708, 1561,
1643, 1635, 2240, 1485, 1376, 1459, 1526, 1659, 1623, 1731, 1662, 1589, 1683, 1672, 2361,
1480, 1385, 1505, 1576, 1649, 1684, 1748, 1642, 1571, 1567, 1637, 2397, 1483, 1390, 1562,
1573, 1718, 1752, 1809, 1759, 1698, 1643, 1718, 2399, 1551, 1497, 1697, 1672, 1805, 1903,
1928, 1963, 1807, 1843, 1950, 2736, 1798, 1700, 1901, 1820, 1982, 1957, 2076, 2107, 1799,
1854, 1968, 2364, 1662, 1681, 1725, 1796, 1938, 1871, 2001, 1934, 1825, 1930, 1867, 2553,
1624, 1533, 1676, 1706, 1781, 1772, 1922, 1743, 1669, 1713, 1733, 2369, 1491, 1445, 1643,
1683, 1751, 1774, 1893, 1776, 1743, 1728, 1769, 2431), .Tsp = c(1, 28.9166666666667, 12),
class = "ts")
```

1. Išbrėžkite duomenų grafiką. Kadangi duomenų išsibarstymas kiekvienais metais didėja, naudosime multiplikatyvųjį modelį (pateikite jo apibrėžimą).

2. Multiplikatyviuoju atveju duomenis tikslinga logaritmuoti - tai modelį paverčia adityviu ir kartu stabilizuoja dispersiją (kodėl?).
3. Išlogaritmavę duomenis, išskirkite kvadratinį arba splaininį trendą.
4. Abiem atvejais liekanos turi akivaizdų sezoniškumą (labai daug gėrimų parduodama per Kalėdas). Pašalinkite jį (iš liekanų arba iš pačios logaritmų sekos) kokiu nors būdu (pvz., naudodami tiesinę regresiją su mėnesiniu faktoriumi).
5. Kaip atrodė liekanų ACF ir PACF prieš ir po šios procedūros?
6. Trendą galima išskirti ir iš nelogaritmuotos sekos, bet sezoninę dalį tuomet reikėtų išskirti su funkcija, turinčia multiplikatyviojo modelio opciją (pvz., `decompose`). Atlikite tai.
7. Splaininio modelio liekanos yra stacionarus procesas, bet ne baltasis triukšmas. Nustatykite šio proceso tipą.

Vėliau papildyti:

8. Kadangi likučiai yra koreliuoti, klasikinis mažiausių kvadratų metodas netinka – reikia atsižvelgti į serial correlation
9. Prognozė...

2.13 UŽDUOTIS. Nustatykite žemiau pateiktų ARMA procesų eilę, generuokite kelias ilgio 60 ir ilgio 300 trajektorijas (su `arima.sim`), ištirkite gautų procesų ACF ir PACF, parinkite parametrus bandymų būdu (pagal AIC minimumą) ir su `auto.arima` iš `forecast` paketo, prognozuokite 20 žingsnių į priekį su `predict` arba `forecast` (maždaug taip: `data(strikes); fit <- arima (strikes,c(0,1,1)); plot(forecast(fit))` arba `predict(auto.arima (strikes),n.ahead=20)`).

1. $y_t = (1-0.75L)w_t$
2. $y_t = (1+0.75L)w_t$
3. $(1-0.75L)y_t = w_t$
4. $(1+0.75L)y_t = w_t$
5. $(1-0.9L)y_t = (1+0.9L)w_t$
6. $(1-1.273L-0.81L^2)y_t = w_t$
7. $y_t = (1-1.273L-0.81L^2)w_t$
8. $(1-1.4745L+0.51L^2)y_t = (1-1.157L+0.81L^2)w_t$
9. $(1-0.95L^4)y_t = w_t$

2.14 UŽDUOTIS. Išnagrinėkite `bicoal` laikinę seką iš `forecast` paketo. Ar ją tinkamai aprašo modelis $y_t = c+\alpha_1 y_{t-1}+\alpha_2 y_{t-2}+\alpha_3 y_{t-3}+\alpha_4 y_{t-4}+w_t$? Kaip vadinamas šis ar kiti jūsų pasirinkti modeliai? Prognozuokite y_t reikšmes ketveriems metams į priekį.

3. ARIMA ir SARIMA modeliai. Vienetinės šaknys

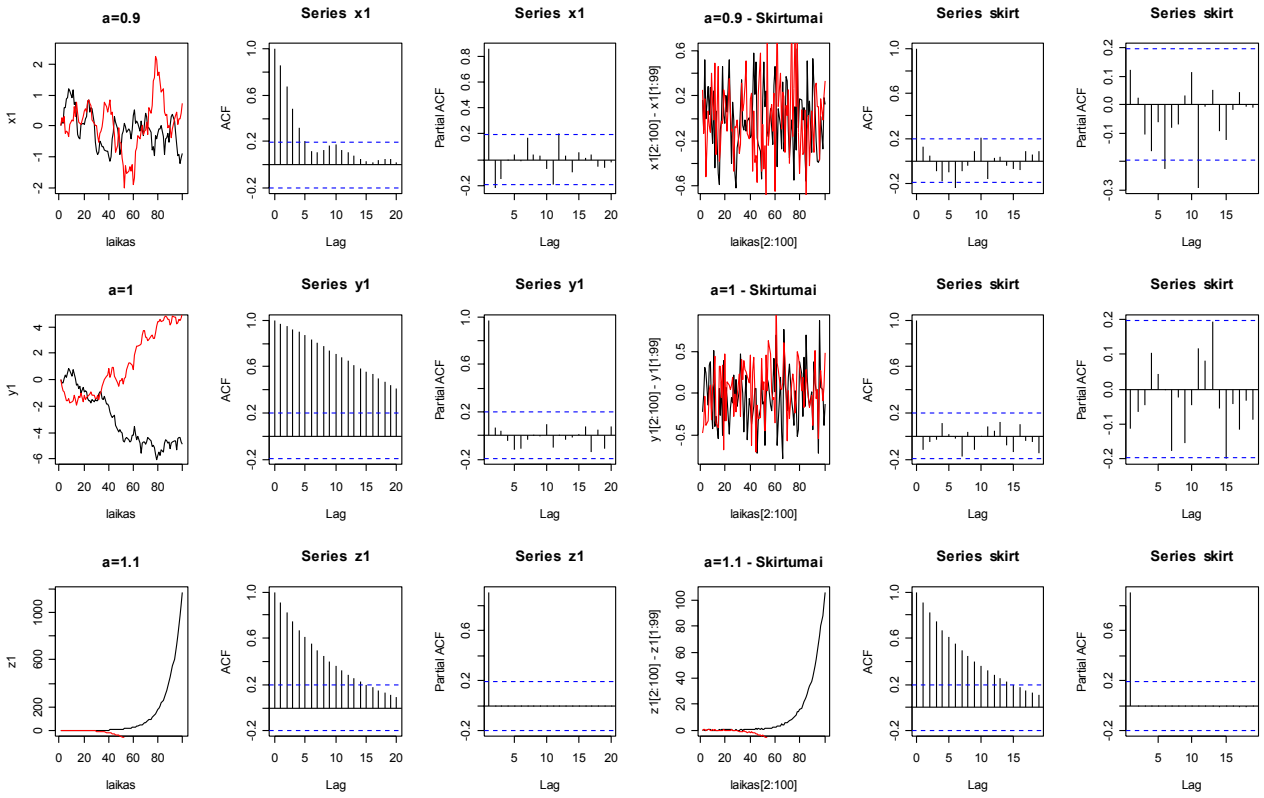
Nestacionarūs procesai ekonomikoje yra žymiai svarbesni negu stacionarūs. Šiame skyriuje aptarsime vieną jų klasę, vadinamuosius vienetinės šaknies procesus.

3.1. ARIMA modelis

Panagrinėkime tris AR(1) procesus $y_t - a_1 y_{t-1} = w_t$ su autoregresijos koeficientu a_1 atitinkamai lygiu 0,9; 1 ir 1,1):

- $y_t = 0,9 \cdot y_{t-1} + w_t = 0,9^t y_0 + 0,9^{t-1} w_1 + ... + 0,9 w_{t-1} + w_t$, $t = 0, 1, ...$ - stacionarus procesas (pradinė reikšmė y_0 ir praeities „inovacijos“ arba „šokai“ (dar kitaip – ekonominės sistemos impulsai arba postūmiai) w_i, $i = 1, 2, ..., t-1$ po truputį „užmirštami“, nes jie dėl nykstančių koeficientų $0,9^t$ artėja į nulį);
- $y_t = 1 \cdot y_{t-1} + w_t = y_0 + \sum_{i=1}^{t} w_i$ - atsitiktinis klaidžiojimas (praeities, t.y., w_i, $i < t$, įtaka nesilpnėja);
- $y_t = 1,1 \cdot y_{t-1} + w_t$ - „sprogimas“ (iš esmės y_t auga kaip geometrinė progresija $y_t = 1,1 \cdot y_{t-1}$).

Nors šių lygčių autoregresijos koeficientai beveik vienodi, trajektorijų charakteris pastebimai skiriasi.



3.1 pav. Pirma eilutė: a=0.9 (procesas stacionarus), antra eilutė: a=1 (tai atsitiktinis klaidžiojimas; šis procesas nėra stacionarus, bet jo skirtumai - stacionarūs), trečia eilutė: a=1.1 (nei pats procesas, nei jo skirtumai nėra stacionarūs)

Pirmoje grafiko eilutėje išbrėžtos dvi AR(1) proceso trajektorijos (a=0,9) ir vienos iš jų ACF ir PACF grafikai. Tai stacionarus procesas, jo skirtumų $\Delta y_t = y_t - y_{t-1}$ procesas irgi stacionarus, bet paprastai stacionaraus proceso skirtumai nėra įdomūs.

Antroje eilutėje išbrėžtos dvi AR(1) proceso trajektorijos (a=1) ir vienos iš jų ACF ir PACF grafikai. Pats procesas, ko gero, nėra stacionarus (tai matyti iš trajektorijų ir lėtai gęstančios ACF grafikų), tačiau jo skirtumai jau sudaro stacionarų procesą (beje, šį kartą – baltąjį triukšmą (pagrįskite)). Trečia eilutė yra mažai įdomi, nes pats procesas (ir netgi jo skirtumai) yra akivaizdžiai nestacionarūs ir koreliaciniai metodai čia netinka.

Nestacionarių procesų empirinės autokoreliacinė funkcijos
Stacionariu atveju empirinė ACF aprašo gretimų proceso reikšmių ryšio stiprumą. Nestacionariu atveju lėtai gęstanti ACF pateikia ne ACF reikšmes (šiuo atveju jos netgi nėra apibrėžiamos), bet tik informuoja apie tai, kad procesas nėra stacionarus.

Neįtikėtinai daug[1] ekonometrijoje nagrinėjamų nestacionarių procesų turi antroje eilutėje aptartą savybę – tokiais atvejais sakoma, kad procesas turi vienetinę šaknį (nes jo charakteristinės lygties $(A - a_1 =)A - 1 = 0$ šaknis A lygi 1). Yra daug testų (juos aptarsime vėliau), skirtų tikrinti hipotezei H_0 :*procesas turi vienetinę šaknį (ir yra nestacionarus)* su alternatyva H_1 :*procesas yra stacionarus*.

Pateiksime kelis apibrėžimus. Jei sekos y_t pirmųjų skirtumų seka $\Delta y_t = y_t - y_{t-1}$ yra stacionari, pati seka vadinama (vieną kartą) integruota seka ir žymima I(1) , o jei stacionari tik d-ųjų skirtumų[2] seka – d kartų (d-aja eile) integruota seka ir žymima I(d) (jei pati seka yra stacionari, ji žymima simboliu $I(0)$). Sakysime, kad ARMA(p+1,q) procesas y_t : $\Phi_{p+1}(L)y_t = \Theta_q(L)w_t$ turi[3] vienetinę (autoregresijos) šaknį, jei bent viena iš p+1 autoregresinių šaknų lygi 1, t.y., jei $\Phi_{p+1}(L) = \Phi'_p(L)(1-L)$. Tai ekvivalentu tam, kad skirtumų procesas Δy_t yra ARMA(p,q): $\Phi'_p(L)(1-L)y_t = \Phi'_p(L)\Delta y_t = \Theta_q(L)w_t$ (pats ARMA(p+1,q) procesas y_t dabar vadinamas ARIMA(p,1,q) procesu).

Jei proceso d-ųjų skirtumų procesas $\Delta^d y_t$ yra stacionarus ARMA(p,q) procesas, tai pats procesas y_t vadinamas ARIMA(p,d,q) procesu

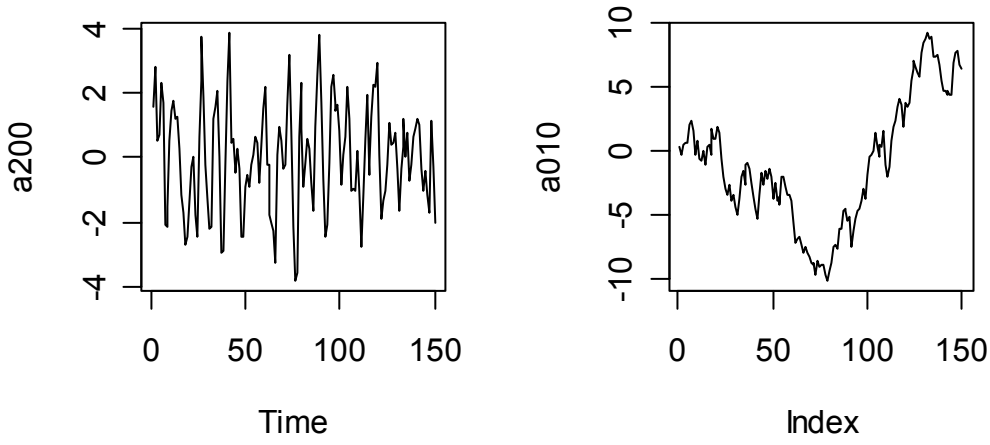
[1] Stacionarūs procesai „užmiršta praeitį“, jų reikšmės (kai išorės impulsai baigiasi) artėja prie proceso vidurkio. Antra vertus, procesai su vienetine šaknimi „prisimena viską“, jų reikšmės (kai išorės impulsai baigiasi) lieka kur buvusios. Ši savybė paaiškina, kodėl tiek daug ekonominių procesų turi stochastinius trendus. Pavyzdžiui, akivaizdu, kad šiuolaikinis technologijų lygis yra ankstesnių atradimų suma, todėl visi procesai, priklausantys nuo progreso, tikriausiai turės vienetinę šaknį. Struktūrinių pasikeitimų naftos rinkoje įtaka yra taip pat negrįžtama. Tie patys samprotavimai galioja nominaliajam turtui ir eksportui, todėl ir pajamoms ir išlaidoms (ir todėl nedarbui ir algoms). Dėl priežasčių bendrumo minėti procesai gali būti ne tik integruoti (t.y. turėti vienetinę šaknį), bet ir kointegruoti (žr. 7 skyrių).

[2] d-ųjų skirtumų operatorius apibrėžiamas taip: $\Delta^d = \Delta(\Delta^{d-1})$. Pvz., $\Delta^2 Y_t = \Delta(\Delta Y_t) = \Delta(Y_t - Y_{t-1}) = Y_t - Y_{t-1} - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$.

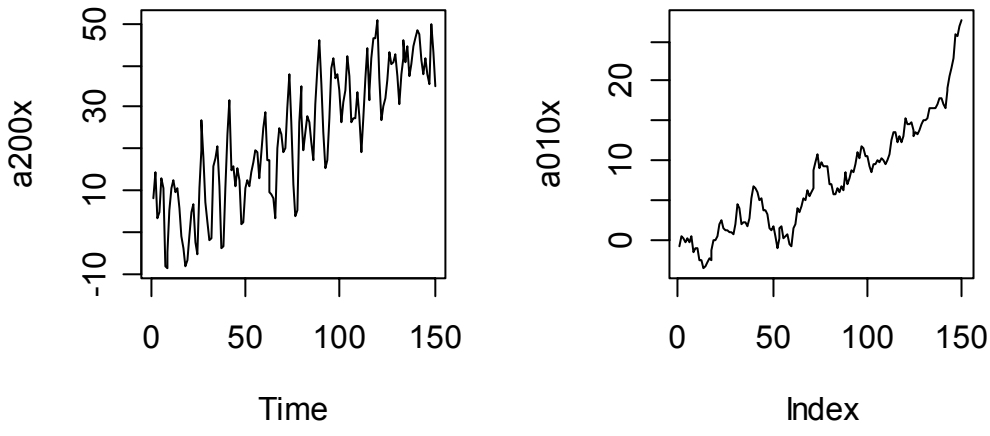
[3] Priminsime, kad raide L žymime postūmio operatorių: $Ly_t = y_{t-1}$ (plg. 2-23 psl.)

3.2. `arima` funkcija (1)

ARIMA procesų parametrams vertinami paprastai vartojama `arima` funkcija. Čia apie ją pakalbėsime plačiau ir aptarsime kelis skirtingus procesus; nežiūrint to, kad tai skirtingai modeliuojami procesai, jų trajektorijos gali būti visai panašios, taigi svarbu mokėti juos teisingai klasifikuoti.



3.2 pav. Stacionaraus AR(2) (taigi TS) proceso grafikas (kairėje) ir nestacionaraus atsitiktinio klaidžiojimo (taigi DS proceso) grafikas (dešinėje)

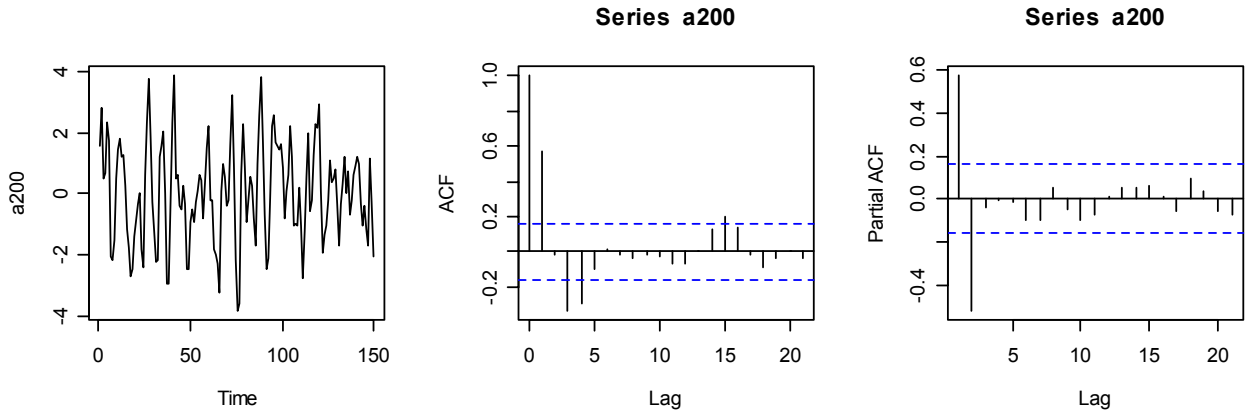


3.3 pav. TS procesas kairėje aprašomas kaip tiesinis trendas plius stacionarūs AR(2) nuokrypiai nuo jo; DS procesas dešinėje yra atsitiktinis klaidžiojimas su teigiamu dreifu

A. Skyrelį pradėsime paprasčiausiu TS procesu - stacionaria AR laikine seka.

```
set.seed(2)   # Pasirenkame „šokų" seką ir generuojame ARIMA(2,0,0) proceso rea-
              # lizaciją
a200=arima.sim(n=150,list(order = c(2,0,0), ar = c(0.88,-0.48)))
# Proceso realizacija turi 150 narių
```

Taigi generavome (stacionaraus) ARIMA(2,0,0) proceso $y_t = 0.88y_{t-1} - 0.48y_{t-2} + w_t$ 150 reikšmių (čia w_t yra baltas triukšmas).



3.4 pav. Laikinė seka `a200` yra, ko gero, AR(2) procesas (nes PACF turi tik du reikšmingus stulpelius, o ACF gęsta eksponentiškai)

Dabar nustatysime autoregresinio proceso `a200` parametrus.

```
> (fit200=arima(a200, order=c(2, 0, 0), xreg=rep(1,150), include.mean=FALSE))

Coefficients:
         ar1      ar2     xreg
      0.8841  -0.5316  0.0145
s.e.  0.0694   0.0697  0.1397  # Laisvasis narys nesiskiria reikšmingai nuo 0

sigma^2 estimated as 1.223:  log likelihood = -228.48,  aic = 464.95
```

Ši sudėtingos sintaksės (žr. žemiau, ten yra paprastesnis variantas) komanda sako, kad pirmiausiai sudarysime `a200` regresijos modelį konstantos atžvilgiu: $a200_t = c \cdot 1 + e_t = 0.0145 + e_t$, o paskui[4] apskaičiuosime du paklaidų e_t autokoreliacijos koeficientus. Čia e_t yra AR(2) struktūrą turinčios paklaidos – kadangi paklaidos nėra baltas triukšmas, įprastinė `lm` procedūra netinka. Mes vartojome `arima` funkciją, kuri pagrįsta apibendrintuoju[5] MK metodu. Modelio parametrus taip pat galima apskaičiuoti ir su `gls` (generalized least squares) funkcija iš `nlme` paketo:

[4] Iš tikrųjų, viskas vyksta tuo pačiu metu (taikome didžiausio tikėtinumo metodą).
[5] *Apibrėžimas*. Mažiausių kvadratų metodas, skirtas tam atvejui, kai paklaidos nėra baltasis triukšmas, vadinamas apibendrintuoju.

```
> library(nlme)
> gls(a200~1,correlation=corARMA(c(0.8,-0.5),p=2,q=0),method= "ML")
  Log-likelihood: -228.4767

Coefficients:
(Intercept)
 0.01453053
Correlation Structure: ARMA(2,0)
 Formula: ~1

Parameter estimate(s):
      Phi1       Phi2
 0.8840731 -0.5315646
Degrees of freedom: 150 total; 149 residual
Residual standard error: 1.598940
```

Taigi mūsų modelis yra $y_t = 0.0145 + e_t$; čia $e_t = 0.884e_{t-1} - 0.532e_{t-2} + w_t$. Atkreipsime dėmesį, kad šį procesą galima užrašyti ir kitaip: kadangi $e_t = y_t - 0.0145$, $e_{t-1} = y_{t-1} - 0.0145$ ir $e_{t-1} = y_{t-1} - 0.0145$, todėl padauginę šiuos narius iš atitinkamų koeficientų ir viską sudėje, gauname tokią lygtį (centruotam) procesui: $\tilde{y}_t = (y_t - 0.0145 =)\ 0.884\tilde{y}_{t-1} - 0.532\tilde{y}_{t-2} + w_t$ (kitaip sakant, AR(p) procesą su baltojo triukšmo paklaidomis galima užrašyti kaip konstantą su AR(p) paklaidomis). Kita vertus, jau minėjome, kad šį modelį galima sudaryti paprasčiau.

```
> (fit200=arima(a200, order=c(2, 0, 0)))

Coefficients:
         ar1      ar2  intercept
      0.8841  -0.5316     0.0145
s.e.  0.0694   0.0697     0.1397

sigma^2 estimated as 1.223:  log likelihood = -228.48,  aic = 464.95
```

Dar kartą matome, kad laisvasis narys (angl. intercept) yra nereikšmingas, todėl sudarykime modelį be jo.

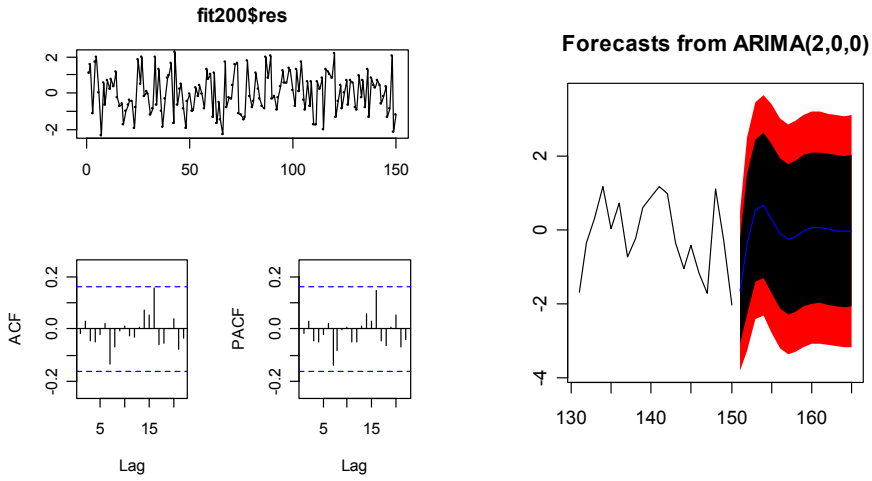
```
> (fit200=arima(a200, order=c(2, 0, 0),include.mean=F)) # a200 yra l. seka, o
                                                        # fit200 yra jos modelis
Coefficients:
         ar1      ar2
      0.8842  -0.5316
s.e.  0.0694   0.0697

sigma^2 estimated as 1.223:  log likelihood = -228.48,  aic = 462.96
```

Naujo modelio koeficientai praktiškai nepasikeitė, o jo AIC sumažėjo, todėl tarsime, kad šis modelis geriausiai aprašo mūsų duomenis: $y_t = 0.8842y_{t-1} - 0.5316y_{t-2} + w_t$. Beje, dar derėtų įsitikinti, kad šio modelio likučiai sudaro baltąjį triukšmą.

```
tsdisplay(fit200$res) # Žr. 3.5 pav.
```

Gautą modelį `fit200` galima panaudoti prognozei: tam surinkite `predict(fit200,15)` arba `forecast(fit200,15)`. Prognozės grafiką galima išbrėžti su

```
plot(forecast(fit200,15),include=20)# Paliksime tik 20 paskutinių sim200 narių;
                                    # prognozuojame 15 žingsnių į priekį
```



3.5 pav. Likučiai `fit200$res` yra prašomi baltuoju triukšmu (kairėje); maždaug 15 prognozės narių dar atspindi autokoreliacinę proceso struktūrą, o paskui prognozė tampa lygi 0 (dešinėje)

▶▶ Dauguma ekonomikos procesų yra aprašomi nestacionariais procesais. Deja, atskirti TS procesą su trendu nuo DS proceso dažnai nėra lengva (plg. 3.6, 3.7 ir 3.10 paveikslus). Norėdami geriau susigaudyti atsirandančiose problemose, aptarsime šių procesų modeliavimą ir vertinimą.

B. Ankstesniame pavyzdyje y_t buvo TS procesas su konstantai lygiu trendu ir koreliuotomis paklaidomis. Dabar nagrinėsime TS procesą su tiesiniu trendu ir koreliuotomis paklaidomis, pvz, $y_t = 0,3t + 5e_t$; čia $e_t = 0.8842e_{t-1} - 0.5316e_{t-2} + w_t$, $t = 1,...,150$. Šiuo atveju modelį parinksime keliais žingsniais.

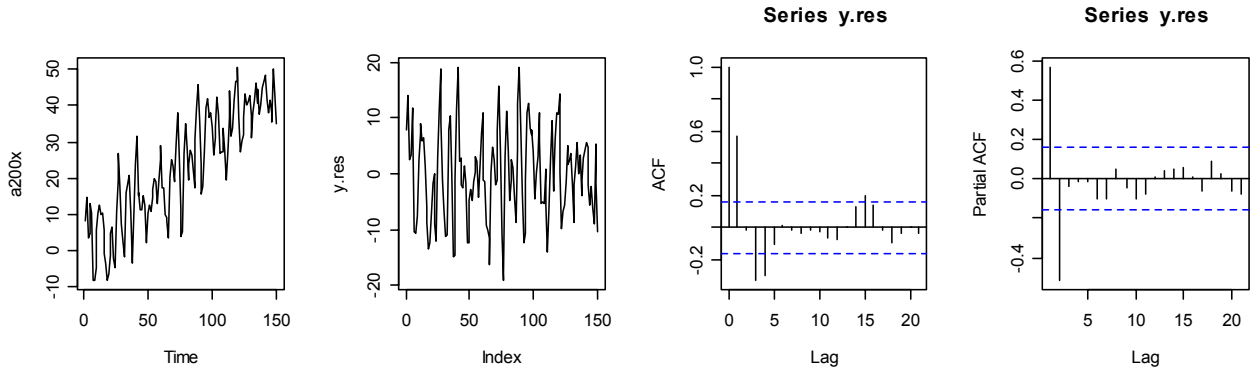
1. Nesunku įsitikinti (su `plot(a200x)`), kad `a200x=0.3*(1:150)+5*a200` turi tiesinį trendą (žr. 3.6 pav., kairėje). Klasikiniu mažiausių kvadratų metodu išskiriame trendą ir apskaičiuojame likučius: `y.res=lm(a200x~time(a200x))$res`. Išbrėžę pagrindinius `y.res` grafikus, matome, kad liekanos `y.res`, ko gero, turi AR(2) struktūrą.

3. Iš naujo, atsižvelgdami į nustatytą liekanų struktūrą ir dabar remdamiesi `arima` funkcija, skaičiuojame modelio parametrus:

```
fit200x=arima(a200x,c(2,0,0),xreg=time(a200x))
```

Paaiškinsime šią eilutę - pirmiausiai sudarome `a200x` regresiją tiesės `time(a200x)` atžvilgiu: $a200_t = \gamma_0 + \gamma_1 t + e_t$, o paskui (iš tikrųjų – tuo pačiu metu) apskaičiuojame du paklaidų e_t autokoreliacijos koeficientus. Čia e_t yra AR(2) struktūrą turinčios paklaidos – kadangi paklaidos nėra baltas triukšmas, įprastinė `lm` procedūra netinka.

4. Kadangi dabar modelio liekanos sudaro baltąjį triukšmą (surinkite `tsdisplay (fit200x$res)`), tarsime, kad modelis parinktas teisingai.



3.6 pav. TS laikinės sekos `a200x` ir jos MK modelio liekanų pagrindiniai grafikai

```
> fit200x

Call:
arima(x = a200x, order = c(2, 0, 0), xreg = time(a200x))

Coefficients:
         ar1      ar2  intercept  time(a200x)
      0.8840  -0.5323    -0.1742       0.3033
s.e.  0.0694   0.0697     1.4075       0.0162

sigma^2 estimated as 30.57:  log likelihood = -469.87,  aic = 949.74
```

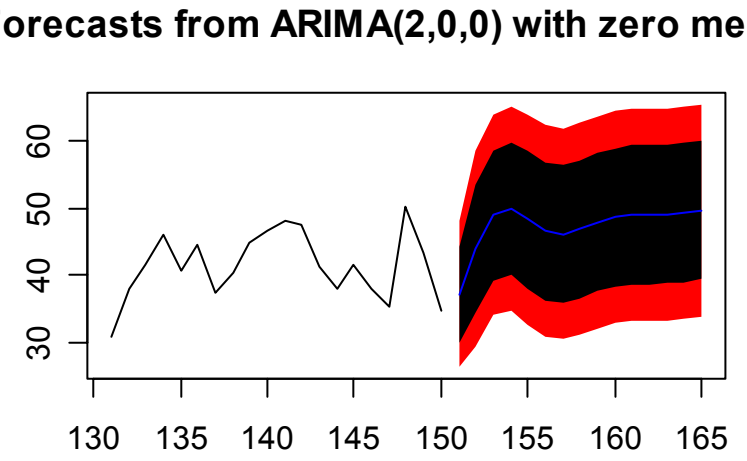
Kitais žodžiais, stebimas procesas aprašomas formule $y_t = -0.1742 + 0.3033*t + e_t$, o $e_t = 0.8840*e_{t-1} - 0.5323*e_{t-2} + w_t$. Atsižvelgdami dar į tai, kad regresijos laisvasis narys nereikšmingas, galutinį modelį sudarysime taip:

```
> (fit200x=arima(a200x,c(2,0,0),xreg=time(a200x),include.mean=F))

Call:
arima(x = a200x, order = c(2, 0, 0), xreg = time(a200x), include.mean = F)

Coefficients:
         ar1      ar2  time(a200x)
      0.8840  -0.5319       0.3015
s.e.  0.0694   0.0697       0.0080

sigma^2 estimated as 30.57:  log likelihood = -469.88,  aic = 947.76
```

Taigi galutinė formulė yra tokia: $y_t = 0.3015*t + e_t$, $e_t = 0.8840*e_{t-1} - 0.5319*e_{t-2} + w_t$ arba $y_t = a_0 + 0.8840y_{t-1} - 0.5319y_{t-2} + a_3 t + w_t$.

3.1 UŽDUOTIS. Apskaičiuokite konstantas a_0 ir a_3.

`a200x` reikšmes prognozuoti galime su komanda `predict(fit200x,newxreg=151:165)` arba `plot(forecast(fit200x,15,xreg=151:165),include=20)`.

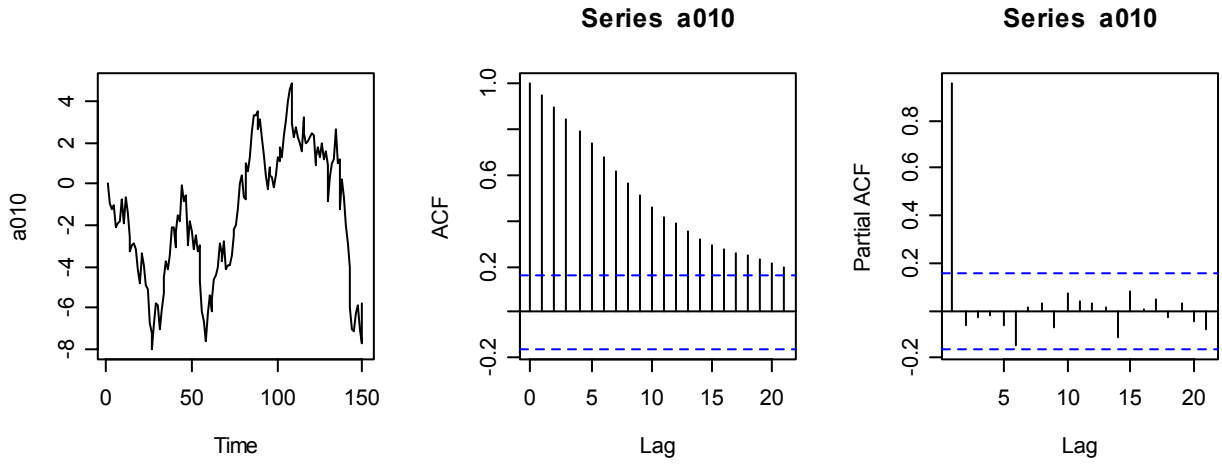


3.7 pav. Proceso `a200x` 15 žingsnių (tiesiškai augantis!) prognozės grafikas

C. y_t yra integruotasis procesas su stochastiniu trendu: $y_t = y_{t-1} + w_t$.

```
set.seed(3)
a010=ts(c(0,cumsum(rnorm(149)))) # arba A010=diffinv(rnorm(149))
# Sukūrėme atsitiktinio klaidžiojimo, t.y. ARIMA(0,1,0)[6], procesą, žr. 3.8 pav.
# Pagal konstrukciją, skirtumai Δa010 bus b. triukšmas su nuliniu vidurkiu
```

Šios sekos grafike (žr. 3.8 pav.) matyti, kad gautoji seka yra arba atsitiktinis klaidžiojimas (nes a_1,



3.8 pav. Laikinės sekos `a010` pagrindiniai grafikai: ilgos ekskursijos aukštyn ir žemyn, lėtai mažėjantis ACF grafikas ir artima vienetui pirmoji PACF reikšmė reiškia, kad `a010` yra, ko gero, atsitiktinis klaidžiojimas (arba, gal būt, AR(1) procesas su artimu 1 koeficientu a_1)

[6] Priminsime: toks užrašas reiškia, kad pirmųjų skirtumų procesas yra ARMA(0,0) procesas.

tikriausiai, lygus 1), arba stacionarus AR(1) procesas su artimu 1 koeficientu a_1. Pirmiausiai išnagrinėsime šią prielaidą.

```
> arima(a010, order=c(1, 0, 0)) # Įvertinsime AR 1-ąjį koef

Coefficients:
         ar1  intercept
      0.9585    -1.8801
s.e.  0.0217     1.5506

sigma^2 estimated as 0.8051:  log likelihood = -197.84,  aic = 401.68
```

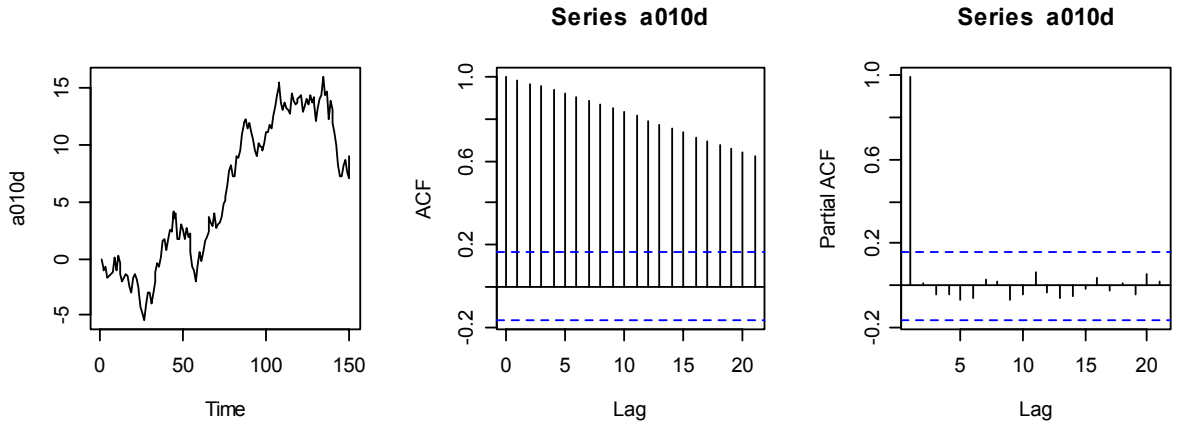
Taigi, jei ši laikinė seka yra stacionaraus AR(1) proceso realizacija, tai jo pirmasis koeficientas labai artimas 1. Antra vertus, gali būti, kad tai atsitiktinio klaidžiojimo (taigi AR(1) proceso su a_1=1) realizacija (yra žinoma, kad tuomet `arima` pateikia žemyn paslinktą a_1 įvertį). Šiuo atveju vienintelis vertinamas parametras yra atsitiktinį klaidžiojimą generuojančio baltojo triukšmo dispersija[7]:

```
> arima(a010, order=c(0, 1, 0))

sigma^2 estimated as 0.8223:  log likelihood = -196.85,  aic = 395.7
```

Taigi turime dvi konkuruojančias hipotezes ir kol kas jokio konkretaus atsakymo. Formalūs testai joms tikrinti bus pateikti sekančiame skyrelyje, o kol kas išnagrinėsime dar vieną procesą – atsitiktinį klaidžiojimą su (teigiamu) dreifu.

D. y_t yra atsitiktinis klaidžiojimas su dreifu, lygiu 0,1: $y_t = 0{,}1 + y_{t-1} + w_t$.

```
set.seed(3)
a010d=ts(c(0,cumsum(rnorm(149,mean=0.1))))
# Sukūrėme atsitiktinio klaidžiojimo su dreifu, procesą, žr. 3.9 pav.
# Pagal konstrukciją, skirtumai Δa010d bus b. triukšmas su vidurkiu lygiu 0,1
```



3.9 pav. Laikinės sekos `a010d` grafikas akivaizdžiai nestacionarus, todėl ACF ir PACF grafikai mažai prasmingi; sprendžiant pagal grafiką, pats procesas gali būti atsitiktinis klaidžiojimas, atsitiktinis klaidžiojimas su teigiamu dreifu arba TS procesas su tiesiniu trendu ir, gal būt, stacionariomis paklaidomis

Visų trijų variantų, išvardintų 3.9 pav. aprašyme, parametrų įverčius galima gauti taip:

[7] Priminsime, ji buvo lygi 1.

```
arima(a010d, order=c(0, 1, 0)) # Atsitiktinis klaidžiojimas
sigma^2 estimated as 0.8246:  log likelihood = -197.05,  aic = 396.1


arima(a010d, order=c(0, 1, 0), xreg=time(a010d)) # A. klaidžiojimas su dreifu
Coefficients:
      time(a010d)
           0.0612  # čia dreifo koeficiento (iš tikro jis lygus 0,1) įvertis
s.e.       0.0742

sigma^2 estimated as 0.8208:  log likelihood = -196.71,  aic = 397.42


arima(a010d,c(1,0,0),xreg=time(a010d)) # Tiesinis trendas su AR(1) paklaidomis
Coefficients:
         ar1  intercept  time(a010d)
      0.9607    -1.2608      0.0903 # trendo krypties koeficientas įvertintas
s.e.  0.0223     2.7951      0.0309 # tiksliau, paklaidos, ko gero, turi viene-
                                    # tinę šaknį
sigma^2 estimated as 0.8043:  log likelihood = -197.79,  aic = 403.57
```

Baigdami pažymėsime, kad **C** ir **D** atvejų analizė baigėsi dvejonėmis dėl proceso tipo. Šias abejones išsklaidyti turėtų 3.3 skyrelio medžiaga.

3.2 UŽDUOTIS. Atlikite 3.34 užduotį.

3.3. Dikio ir Fulerio vienetinės šaknies testas

Tarkime, kad tiriamus *duomenis* $y_0, y_1, ..., y_n$ *generuojantis* (stacionarusis) *procesas* (DGP) yra užrašomas formule $Y_t = a_1 Y_{t-1} + W_t, |a_1| < 1$. Jei koeficientas a_1 artimas 1, proceso elgesys gali būti panašus į elgesį proceso su vienetine šaknimi. Norint patikrinti hipotezę $H_0 : a_1 = 1$ *(procesas nestacionarus)* su alternatyva $H_1 : a_1 < 1$ *(procesas stacionarus)*, reikėtų apskaičiuoti šio koeficiento t statistiką $t = (\hat{a}_1 - 1)/se(\hat{a}_1)$ (čia $\hat{a}_1$ yra koeficiento a_1 MK įvertis, o $se(\hat{a}_1)$ - šio įverčio standartinė paklaida; šie du skaičiai skaičiuojami pagal žinomas formules, tačiau mes tuo neužsiimsime – tai paprastai atlieka komputeris). Žinome, kad t yra Student'o a.d. T_{n-2} realizacija – jei t mažesnis už $\alpha(\approx 0,05)$ eilės T_{n-2} kvantilį, H_0 atmesime ir teigsime, kad procesas stacionarus.

A.d. T_{n-2} kvantilius galima rasti Student'o skirstinio lentelėse, juos skaičiuoja ir R. Kita vertus, šiuos kvantilius galima apskaičiuoti „eksperimentiškai" (tai vadinamasis Monte Karlo metodas): (N=) keliasdešimt tūkstančių kartų pagal nurodytą formulę generuokite sekas $y_0, y_1, ..., y_n$ ir apskaičiuokite $t_1, t_2, ..., t_N$ - šios skaičių aibės empirinis α eilės kvantilis bus labai artimas tikrajam.

Dabar tarkime, kad DGP užrašomas formule $Y_t = a_1 Y_{t-1} + W_t$, $a_1 = 1$ (tai nestacionarus procesas su vienetine šaknimi). Dikis ir Fuleris (Dickey, Fuller) 1979 m. įrodė, kad t statistika šį kartą turi skirstinį, kuris pastebimai skiriasi nuo Student'o. Kvantilio skirstinys yra sudėtingas, hipotezėms tikrinti reikalingus kvantilius minėti autoriai sugebėjo rasti tik Monte Karlo būdu. Kiek žemiau pateiksime minėtas kritines reikšmes, tačiau pirmiau kiek apibendrinsime savo uždavinį.

Procesą $Y_t = a_1 Y_{t-1} + W_t$ galima perrašyti pavidalu $(Y_t - Y_{t-1} =)\Delta Y_t = \gamma Y_{t-1} + W_t$ (čia $\gamma = a_1 - 1$), o hipotezę $H_0 : a_1 = 1$ dabar formuluoti[8] kaip $H_0 : \gamma = 0$ (su alternatyva $H_1 : \gamma < 0$). Nagrinėjamą procesą galima dar apibendrinti, papildant jį determinuotomis komponentėmis – dreifu ir tiesiniu trendu:

- $\Delta Y_t = \gamma Y_{t-1} + W_t$ (*)
- $\Delta Y_t = a_0 + \gamma Y_{t-1} + W_t$ (procesas su dreifu a_0) (**)
- $\Delta Y_t = a_0 + \gamma Y_{t-1} + \beta t + W_t$ (procesas su dreifu a_0 ir tiesiniu trendu βt) (***)

Pasirodo, kad koeficiento γ įverčio *t* statistikos kvantiliai (taigi ir kritinės reikšmės) kiekvienu iš šių trijų atveju vis kitokie (kritines reikšmes žymėsime simboliais τ, τ_μ ir τ_τ, jų reikšmės pateiktos 3.1 lentelėje). Tiesą sakant, šitos lentelės mums nereikės – viską suskaičiuos R.

3.1 lentelė

Imties dydis	0,01	0,05
$\tau : a_0 = \beta = 0$ (*)		
25	-2,66	-1,95
50	-2,62	-1,95
100	-2,60	-1,95
250	-2,58	-1,95
$\tau_\mu : \beta = 0$ (**)		
25	-3,75	-3,00
50	-3,58	-2,93
100	-3,51	-2,89
250	-3,46	-2,88
τ_τ : (***)		
25	-4,38	-3,60
50	-4,15	-3,50
100	-4,04	-3,45
250	-3,99	-3,43

Viena pagrindinių Dikio ir Fulerio testo problemų yra tokia – įjungti determinuotuosius narius (konstantą ir trendą) į modelį ar ne? Čia aprašysime Perron'o nuosekliąją procedūrą [HS, 47 psl.] (žr. taip pat [E, 257 psl.] ir [S, 772 psl.]). Visuomet pradedame (1) modeliu (žr. žemiau esančią 3.2 lentelę), o po to, judėdami žemyn, bandome eliminuoti (gal būt, neįeinančius į DGP ir todėl trukdančius parametrus. Jei mums nepavyksta atmesti bendresnės nulinės hipotezės (ko gero, dėl mažos testo galios), judame žemyn, labiau ribojančios specifikacijos kryptimi. Testavimo procedūra baigiasi, kai tik pavyksta atmesti vienetinės šaknies hipotezę. Atkreipiame dėmesį, kad žingsniai (2a) ir (4a) yra atliekami tik tada, jei atmetame jungtinę hipotezę (2), ar, atitinkamai, (4). Beje, netgi šiuo atveju geriau remtis DF skirstiniu, taigi (2a) ir (4a) rezultatai turėtų būti vertinami atsargiai.

[8] Toks lygties pavidalas patogesnis, nes hipotezės $H_0 : \gamma = 0$ tikrinimo rezultatus automatiškai pateikia visos statistinės funkcijos (pvz., `lm`).

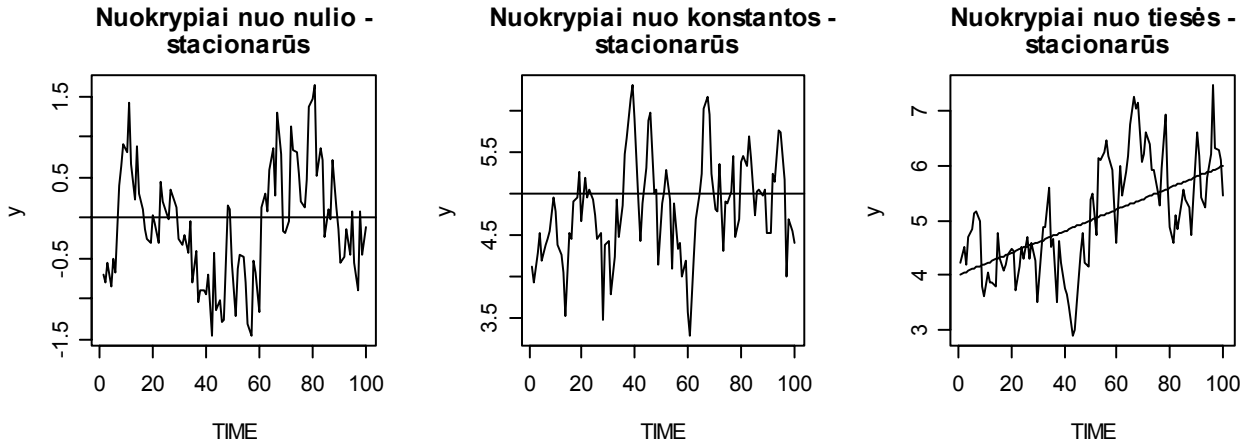
3.2 lentelė

Žingsnis	Modelis	Nulinė hipotezė	Testo statistika		Kritinės reikšmės
(1)	$\Delta y_t = \mu_c + \gamma_c t + (\rho_c - 1)y_{t-1} + u_t$	$\rho_c - 1 = 0$	τ_τ	tau3	žr. R
(2)	$\Delta y_t = \mu_c + \gamma_c t + (\rho_c - 1)y_{t-1} + u_t$	$\rho_c - 1 = \gamma_c = 0$	Φ_3	phi3	žr. R
(2a)	$\Delta y_t = \mu_c + \gamma_c t + (\rho_c - 1)y_{t-1} + u_t$	$\rho_c - 1 = 0$	t		Stand. normalusis
(3)	$\Delta y_t = \mu_b + (\rho_b - 1)y_{t-1} + u_t$	$\rho_b - 1 = 0$	τ_μ	tau2	žr. R
(4)	$\Delta y_t = \mu_b + (\rho_b - 1)y_{t-1} + u_t$	$\rho_b - 1 = \gamma_b = 0$	Φ_1	phi1	žr. R
(4a)	$\Delta y_t = \mu_b + (\rho_b - 1)y_{t-1} + u_t$	$\rho_b - 1 = 0$	t		Stand. normalusis
(5)	$\Delta y_t = (\rho_a - 1)y_{t-1} + u_t$	$\rho_a - 1 = 0$	τ	tau1	žr. R

Pažymėsime, kad mums nėra reikalo žinoti nestandartinius Fulerio ar Dikio ir Fulerio skirstinius – reikalingas kritines reikšmes pateiks R. Taip pat pastebėsime, kad DF testo statistikos neturi nusistovėjusių žymenų, todėl stulpelio „Testo statistika“ dešinėje yra įrašyti dar ir žymenys, vartojami pakete `urca`.

Dikio ir Fulerio testas yra skirtas nustatyti ar stebima laikinė AR(1) seka turi vienetinę šaknį (kitaip sakant, jis skirtas hipotezei $\gamma = 1$ ir įvairioms jos modifikacijoms tikrinti)

Šio testo taikymas gana subtilus, trys žemiau pateikti pavyzdžiai turėtų palengvinti testo rezultatų interpretavimą.



3.10 pav. Trys trendai (**1**. m_t=0, **2**. m_t=5 ir **3**. m_t=2+0.2t) plius stacionarus AR(1) procesas

1. 3.10 pav. kairėje yra išbrėžtas stacionarios laikinės sekos $y_t = 0.7y_{t-1} + w_t$ (kodėl ji stacionari?) su nuliniu vidurkiu realizacijos grafikas. Čia pradiniu laiko momentu t=1 procesas lygus 0; norėdami atsikratyti šios pradinės sąlygos įtakos (žr. programą žemiau), procesą „apšildome“ iki t = 100, o toliau procesą nagrinėjame tik nuo 101 iki 200.

```
par(mfrow=c(1,3)); seed=1; y1=numeric(200); gamma=0.7
y1[1]=0; TIME=1:100; set.seed(seed)
for(i in 2:200) y1[i]=gamma*y1[i-1]+rnorm(1,0,0.5)
plot(TIME,y1[101:200],type="l",ylab="y", main="Nuokrypiai nuo nulio -\n stacio-
narūs")
abline(0,0)
```

Nors mes žinome, kad ši laikinė seka stacionari, tačiau jos trajektorija labai panaši į atsitiktinį klaidžiojimą. Ar suderinami mūsų duomenys su vienetinės šaknies hipoteze? Dikio ir Fulerio testą visuomet geriausiai pradėti formuluoti alternatyva H_1 : *proceso nuokrypiai nuo nulio yra stacionarus AR(1) procesas* su nuline hipoteze H_0 : *stebime atsitiktinį klaidžiojimą be dreifo* $y_t = y_{t-1} + w_t$.

Su komanda `help.search("dickey-fuller")` nustatome, kad R turi tris Dikio ir Fulerio testavimui skirtas funkcijas – tai `adf.test` iš `tseries`, `ur.df` iš `urca` ir `ADF.test` iš `uroot` paketų. Mums šiuo metu tinkamiausia antroji funkcija.

```
> Y1=y1[101:200]      # Būtent šių duomenų grafikas išbrėžtas 3.10 pav. kairėje
> summary(ur.df(Y1, type = "none",lags=0)) # „none" reiškia, kad alternatyviojo
                                           # proceso vidurkis 0
Call:
lm(formula = z.diff ~ z.lag.1 - 1)  # Tai (*) formulė iš 3-10 psl.
                                    # Vieninteli koeficientą vertiname MK metodu
Coefficients:
        Estimate  Std.Error t-value Pr(>|t|)
z.lag.1 -0.27322    0.06866  -3.979 0.000133    # t-value=(Estimate-1)/Std.Error
                                    # Jei teisinga H0 - p reikšmė neteisinga
Residual standard error: 0.481 on 98 degrees of freedom
Multiple R-Squared: 0.1391,     Adjusted R-squared: 0.1303
F-statistic: 15.83 on 1 and 98 DF,  p-value: 0.0001326

Value of test-statistic is: -3.9792

Critical values for test statistics:
     1pct  5pct 10pct
tau1 -2.6 -1.95 -1.61
```

Apatinėje eilutėje įrašytos testo kritinės reikšmės (jos sutampa su 3.1 lentelės „žalios" eilutės reikšmėmis). Kadangi testo statistika yra mažesnė už 1% kritinę reikšmę -2.6, H_0 atmetame ir (teisingai) nutariame, kad `Y1` yra stacionarus AR(1) procesas su nuliniu vidurkiu.

Koks būtų testavimo rezultatas, jei tirtume klaidžiojimą be dreifo (t.y., laikinę seką, aprašomą alternatyvia hipoteze)?

```
set.seed(2)
W1=cumsum(c(0,rnorm(99,0,0.5))) # Generuojame ats. klaidžiojimą be dreifo
summary(ur.df(W1, type = "none",lags=0))
[...]
Value of test-statistic is: -1.3621

Critical values for test statistics:
     1pct  5pct 10pct
tau1 -2.6 -1.95 -1.61
```

Matome, kad šį kartą nuokrypio statistika lygi -1.3621, ji arčiau nulio negu 10% kritinė reikšmė, todėl nėra pagrindo atmesti nulinę hipotezę – Dikio ir Fulerio testas vėl elgiasi taip, kaip tikimasi.

2. Daugumos ekonominių sekų vidurkis nėra nulis, todėl gerai būtų turėti Dikio ir Fulerio testo variantą ir šiam atvejui.

```
mu=5; alpha=mu*(1-gamma)
set.seed(seed+5); y2=numeric(200)
```

```
for(i in 2:200) y2[i]= alpha + gamma*y2[i-1]+rnorm(1,0,0.5)
plot(TIME,y2[101:200],type="l",ylab="y",
main="Nuokrypiai nuo konstantos -\n stacionarūs")
abline(mu,0)
```

3.10 pav. viduryje yra išbrėžtas grafikas stacionaraus AR(1) proceso su $\gamma = 0.7$ ir nenuliniu vidurkiu $\mu = 5$: $(y_t - \mu) = \gamma(y_{t-1} - \mu) + w_t$ (žr. aukščiau pateiktą programą). Šį procesą galima užrašyti ir kitaip: $y_t = \alpha + \gamma y_{t-1} + w_t$; čia $\alpha = \mu(1-\gamma)$. Nors mes žinome, kad ši laikinė seka stacionari (su nenuliniu vidurkiu), tačiau jos trajektorija panaši į atsitiktinį klaidžiojimą (su nežymiu aukštyn nukreiptu dreifu). Natūralu paklausti, ar mūsų duomenys suderinami su vienetinės šaknies hipoteze? Dikio ir Fulerio testą geriausiai pradėti formuluoti alternatyva H_1: *proceso nuokrypiai nuo konstantos sudaro stacionarų AR(1) procesą* su nuline hipoteze H_0': *stebime klaidžiojimą be dreifo, t.y.,* $(\alpha, \gamma) = (0,1)$ (plg. 3.2 lentelę, ten nuokrypio statistika žymima `phi1`) arba su kita nuline hipoteze H_0'': *stebime klaidžiojimą su dreifu, t.y.,* $\gamma = 1$ (plg. 3.2 lentelę, ši nuokrypio statistika ten žymima `tau2` arba τ_μ).

```
> Y2=y2[101:200]
> summary(ur.df(Y2, type = "drift",lags=0))# „drift" reiškia, kad alternatyviojo
                                           # proceso modelyje yra narys α
Call:
lm(formula = z.diff ~ z.lag.1 + 1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  # Klasikinio MK modelio
(Intercept)  1.60455    0.36495   4.397 2.82e-05  # įverčiai ir p reikšmės
z.lag.1     -0.33160    0.07494  -4.425 2.53e-05

Value of test-statistic is: -4.4246 9.7905

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.51 -2.89 -2.58  # plg. 3.1 lentelės melsvą eilutę
phi1  6.70  4.71  3.86
```

Abi nuokrypio statistikos yra moduliu didesnės už atitinkamas `tau2` ir `phi1` 1% kritines reikšmes, todėl abi nulines hipotezes atmetame ir darome „teisingą" sprendimą H_1 - *stebima laikinė seka yra stacionarus AR(1) procesas su nenuliniu vidurkiu*.

3.3 UŽDUOTIS. Sugeneruokite du procesus, aprašomus H_0' ir H_0'' sąlygomis. Atlikite jiems tik ką aptartą Dukio ir Fulerio testo variantą.

3. 3.10 pav. dešinėje yra išbrėžtas (žr. žemiau pateiktą programą) grafikas proceso, kurio nuokrypiai nuo tiesinio trendo sudaro stacionarų AR(1) procesą su nuliniu vidurkiu ir $\gamma = 0.7$. Kitaip sakant, tai procesas, kurį galima užrašyti formule $y_t - a - bt = \gamma(y_{t-1} - a - b(t-1)) + w_t$ arba $y_t = \alpha + \beta t + \gamma y_{t-1} + w_t$ su $\alpha = a(1-\gamma) + b\gamma$ ir $\beta = b(1-\gamma)$.

Daugelio ekonominių sekų grafikai yra panašaus pavidalo, todėl svarbu nustatyti, ar tiriama seka turi vienetinę šaknį ir teigiamą dreifą ar tiesinį trendą ir stacionarius nuokrypius nuo jo. Dikio ir Fulerio testą pradėsime formuluoti nuo alternatyvios hipotezės – H_1: *proceso nuokrypiai nuo tiesinio trendo yra stacionarus AR(1) procesas* su nuline hipoteze H_0': *proceso skirtumai turi tiesinį*

trendą, t.y., $\gamma = 1$ (plg. 3.2 lentelę, ši nuokrypio statistika ten žymima `tau3` arba τ_τ) arba su kita nuline hipoteze H_0'': *stebime klaidžiojimą su dreifu, t.y.,* $(\beta, \gamma) = (0,1)$ (plg. 3.2 lentelę, ten nuokrypio statistika žymima `phi3`). Šį Dikio ir Fulerio testo variantą pritaikysime savo duomenims.

```
a=2; b=0.02; y3=numeric(200); alpha=a*(1-gamma)+b*gamma; beta=b*(1-gamma)
set.seed(seed+2)
for(i in 2:200) y3[i]= alpha + beta*i + gamma*y3[i-1]+rnorm(1,0,0.5)
plot(TIME,y3[101:200],type="l",ylab="y",
main="Nuokrypiai nuo tiesės -\n stacionarūs")
lines(1:100, a+b*(101:200))            # Išbrėžėme 3.10 paveikslo dešinį grafiką
Y3=y3[101:200]

> summary(ur.df(Y3, type = "trend",lags=0))# „trend" reiškia, kad alternatyviojo
                                          # proceso modelyje yra tiesinis
                                          # trendas
[...]
Value of test-statistic is: -3.9378 5.1919 7.7637

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -4.04 -3.45 -3.15
phi2  6.50  4.88  4.16
phi3  8.73  6.49  5.47
```

`tau3` eilutė sutampa su geltona 3.1 lentelės eilute; kadangi `-3.9378<-3.45`, H_0'' su 5% reikšmingumu atmetame ir tariame, kad teisinga prielaida H_1. Panašiai samprotaudami atmetame ir hipotezę H_0'.

3.4 UŽDUOTIS. Generuokite dvi laikines sekas pagal H_0' ir H_0'' modelius. Patikrinkite jas Dikio ir Fulerio testu.

****
****

Vienetinei šakniai nustatyti dažnai vartojama kita, `adf.test`, funkcija iš `tseries` paketo. Tariama, kad stebima seka aprašoma lygtimi $y_t = \alpha + \beta t + \gamma y_{t-1} + w_t$, o koeficientai, kaip visuomet, vertinami MK metodu. Kaip jau įpratome, pradedame alternatyviąja hipoteze: H_1: *proceso nuokrypiai nuo tiesinio trendo yra stacionarus AR(1) procesas* (taigi *stebime TS procesą*); nulinė hipotezė yra $H_0: \gamma = 1$ - taigi *stebime DS procesą su dreifu* (priminsime, kad tuomet, kai teisinga H_0, t statistika $(\hat{\gamma} - 1)/se(\hat{\gamma})$ turi ne Stjudento, bet (trečiąjį) Dikio ir Fulerio skirstinį).

Funkcija `adf.test` taikytina tik tuomet, kai alternatyva yra procesas su tiesiniu trendu. Kitais atvejais reikia vadovautis arba 3.2 lentelės nurodymais arba paprastesne procedūra, aprašyta 3 – 39 psl.

```
library(tseries)
adf.test(Y3,k=0)

        Augmented Dickey-Fuller Test
Dickey-Fuller = -3.9378, Lag order = 0, p-value = 0.01504
alternative hypothesis: stationary
```

Matome, kad nuokrypio (testo) statistika yra ne kas kita, kaip `ur.df` funkcijos `tau3` statistika; kadangi *p* reikšmė yra mažesnė už 0.05, H_0 atmetame.

Panašią analizę galime atlikti ir su kitais dviem procesais.

```
> summary(ur.df(Y2, type = "trend",lags=0))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.630120   0.366261   4.451 2.31e-05
z.lag.1     -0.353155   0.078532  -4.497 1.93e-05
tt           0.001571   0.001698   0.926    0.357

Value of test-statistic is: -4.497 6.803 10.2026

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -4.04 -3.45 -3.15
phi2  6.50  4.88  4.16
phi3  8.73  6.49  5.47

> adf.test(Y2,k=0)

        Augmented Dickey-Fuller Test

data:  Y2
Dickey-Fuller = -4.497, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary
```

Vadinasi, nulinę hipotezę atmetame (kitaip sakant, 1% reikšmingumo lygiu `Y2` yra TS procesas).

```
> summary(ur.df(Y1, type = "trend",lags=0))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0383405  0.0996875  -0.385 0.701379
z.lag.1     -0.2779652  0.0702980  -3.954 0.000147
tt           0.0004575  0.0017215   0.266 0.791011

Value of test-statistic is: -3.9541 5.2352 7.8452

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -4.04 -3.45 -3.15
phi2  6.50  4.88  4.16
phi3  8.73  6.49  5.47

> adf.test(Y1,k=0)

        Augmented Dickey-Fuller Test

data:  Y1
Dickey-Fuller = -3.9541, Lag order = 0, p-value = 0.01425
alternative hypothesis: stationary
```

Vadinasi, nulinę hipotezę atmetame (kitaip sakant, 5% reikšmingumo lygiu `Y1` yra TS procesas).

3.4. Dikio ir Fulerio praplėstasis (ADF) bei Filipso ir Perono (PP) vienetinės šaknies testai

Iki šiol tarėme, kad nagrinėjamas procesas yra pavidalo $y_t = a_1 y_{t-1} + w_t$. Testai turėtų daugiau lankstumo, jei tartume, kad proceso liekanos yra ne baltasis triukšmas W_t, o AR procesas. Kitaip sakant, tarsime, kad pats tiriamasis procesas yra AR(p), t.yi., pavidalo $y_t = a_1 y_{t-1} + a_2 y_{t-2} + ... + a_p y_{t-p} + w_t$, $p \geq 2$, ir tikrinsime hipotezę H_0: *procesas* Y_t *turi vienetinę šaknį*[9] su alternatyva H_1: *procesas yra stacionarus*. Šį procesą nesunku perrašyti pavidalu[10]

$$y_t = \alpha y_{t-1} + \sum_{i=2}^{p} \beta_i \Delta y_{t-i+1} + w_t \quad \text{arba} \quad \Delta y_t = (\alpha - 1) y_{t-1} + \sum_{i=2}^{p} \beta_i \Delta y_{t-i+1} + w_t \qquad (3.0)$$

(pvz., jei *p*=2, tai $y_t = a_1 y_{t-1} + (a_2 y_{t-1} - a_2 y_{t-1}) + a_2 y_{t-2} + w_t = (a_1 + a_2) y_{t-1} - a_2 (y_{t-1} - y_{t-2}) + w_t$) ir vėl tikrinti vienetinės šaknies hipotezę $H_0 : (\alpha - 1 =) \gamma = 0$ su stacionarumo alternatyva $H_1 : \gamma < 0$. Uždavinį galima apibendrinti, į modelį įtraukiant laisvąjį narį arba dar ir tiesinį trendą. Pasirodo, kad visais trimis atvejais šio *praplėstojo* (angl. augmented) Dikio ir Fulerio (ADF) testo kritinės reikšmės sutampa su ankstesnėmis. Funkcija `adf.test` automatiškai[11] parenka tam tikra prasme optimalią AR proceso eilę.

```
> adf.test(a010)

Dickey-Fuller = -2.3874, Lag order = 5, p-value = 0.415
alternative hypothesis: stationary
```

Nulinė hipotezė H_0 tvirtino, kad *stebime atsitiktinį klaidžiojimą su* (galbūt, nenuliniu) *dreifu*, o alternatyva - *proceso nuokrypiai nuo tiesinio trendo yra stacionarus AR(5) procesas*. Testas teigia, kad H_0 atmesti nėra pagrindo.

Jei DGP yra sudėtingesnis negu AR(1), ADF testas atsižvelgia į tai, į modelį įtraukdamas aukštesnės eilės proceso vėlinius. Filipsas ir Peronas (Phillips, Perron) 1988 m. pasiūlė kitą variantą – jie pakeitė *t* statistikos išraišką. Galima įrodyti, kad PP testas yra veiksmingas ne tik tuomet, kai paklaidos autokoreliuotos, bet ir tuomet, kai jos yra heteroskedatiškos. Kita vertus, PP testui reikia daugiau stebėjimų. Apskritai, nėra „idealaus" vienetinės šaknies testo, todėl geriau vartoti kelis. `urca` (unit root and cointegration analysis) pakete jų yra 6, tai

`ur.df`	Augmented-Dickey-Fuller Unit Root Test
`ur.ers`	Elliott, Rothenberg & Stock Unit Root Test
`ur.kpss`	Kwiatkowski et al. Unit Root Test
`ur.pp`	Phillips & Perron Unit Root Test
`ur.sp`	Schmidt & Phillips Unit Root Test
`ur.za`	Zivot & Andrews Unit Root Test

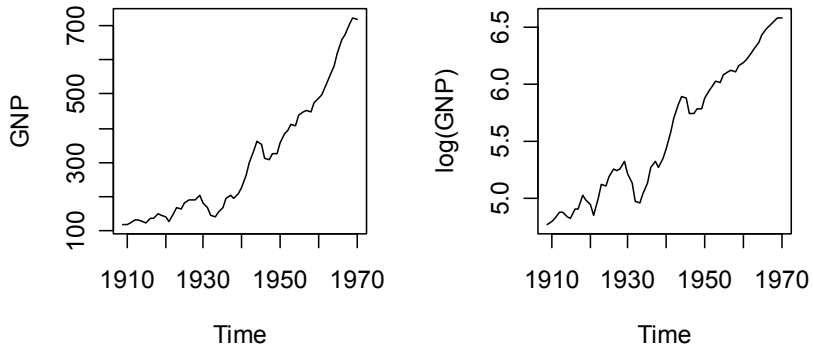
[9] T.y., jo skirtumai Δy_t sudaro TS procesą.

[10] Čia $\alpha = \sum_{j=1}^{p} a_j$, o $\beta_i = -\sum_{j=i}^{p} a_j, i = 2, ..., p$.

[11] Vėlinių skaičių parinkti galima ir „rankomis". Įprastinė, nors ir nelabai naudinga, rekomendacija yra tokia – jų turi būti tiek, kad liekanos sudarytų baltąjį triukšmą.

Jūs galite išbandyti šiuos testus su `a010` (geriau rašyti `summary(ur.df)` ir t.t.). Mes juos pritaikysime garsiems Nelson'o ir Plosser'io duomenims (tai `nporg` duomenų rinkinys iš `urca` paketo). Juose pateikta 14 JAV ekonomikos laikinių sekų nuo 1860 m. iki 1970 m. Iki 1982 m., kai tie duomenys ir jų analizė buvo paskelbti, dauguma ekonomistų buvo įsitikinę, kad visos laikinės sekos yra TS sekos (taigi pašalinus trendą, jos tampa stacionariomis). Nelson'as ir Plosser'is įrodė, kad taip nėra, kad dauguma ekonominių sekų yra DS sekos su vienetine šaknimi ir, gal būt, dreifu (taigi jų skirtumai yra stacionarūs, o jos pačios determinuotojo trendo neturi).

Žemiau ištirsime ne JAV bendrojo vidaus produkto, bet jo logaritmų seką. Jei tartume, kad ekonominis kintamasis kasmet didėja beveik vienodu procentu, tai šis dydis augs eksponentiškai, o jo logaritmas – tiesiškai, todėl augančios ekonominės sekos paprastai logaritmuojamos (plg. 3.7 skyrelį).



3.11 pav. JAV bendrojo vidaus produkto (panašus į eksponentę) grafikas (kairėje) ir jo logaritmo (panašus į tiesę) grafikas (dešinėje)

```
library(urca)
data(nporg)                                  # Nelson'o ir Plosser'io duomenys
gnp=ts(na.omit(nporg[,"gnp.r"]),start=1909) # Real GNP, Billions of 1958
                                             # Dollars,1909 -1970
l.gnp=log(gnp)
par(mfrow=c(1,2))
plot(gnp,ylab="GNP")                        # Žr. 3.11 pav.
plot(l.gnp,ylab="log(GNP)")                 # Žr. 3.11 pav.
summary(ur.df(l.gnp,type="drift")) # Dažniausiai apsiribojama šiuo DF testu
summary(ur.ers(l.gnp,model="trend"))
summary(ur.kpss(l.gnp,type="tau"))
summary(ur.pp(l.gnp, type="Z-tau", model="trend"))
summary(ur.sp(l.gnp, pol.deg=1))
summary(ur.za(l.gnp,model="both",lag=2))
```

****

```
> summary(ur.df(l.gnp,type="drift"))   # Įtrauksime dreifą ir formulėje (3.0)
                                       # vieną autoregresinį narį

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift
```

```
Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.034199   0.081624   0.419  0.67680
z.lag.1     -0.002673   0.014722  -0.182  0.85659
z.diff.lag   0.345106   0.126710   2.724  0.00856 **

Value of test-statistic is: -0.1815 2.4103

Critical values for test statistics:
        1%    5%   10%
tau2 -3.51 -2.89 -2.58      # H0 neatmetame
phi1  6.70  4.71  3.86

**********************************************

> summary(ur.ers(l.gnp,model="trend")) # Pašalinę iš laikinės sekos tiesinį
                                       # trendą, tirsime likučius
###############################################
# Elliot, Rothenberg & Stock Unit Root Test #
###############################################

Test of type DF-GLS
detrending of series with intercept and trend

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
yd.lag       -0.14734    0.07086  -2.079  0.04254 *
yd.diff.lag1  0.39706    0.13566   2.927  0.00507 **
yd.diff.lag2  0.06112    0.14288   0.428  0.67056
yd.diff.lag3 -0.07153    0.14241  -0.502  0.61757
yd.diff.lag4 -0.03216    0.13910  -0.231  0.81806

Value of test-statistic is: -2.0793

Critical values of DF-GLS are:
                    1%    5%   10%
critical values -3.58 -3.03 -2.74   # H0 neatmetame

**********************************************

> summary(ur.kpss(l.gnp,type="tau")) # Dėmesio!!! - Šį kartą H0 skiriasi nuo
# ankstesnių - H0: procesas minus tiesinis trendas (nes type="tau") yra
# stacionarus su alternatyva H1: procesas turi vienetinę šaknį

#######################
# KPSS Unit Root Test #
#######################

Test is of type: tau with 3 lags.

Value of test-statistic is: 0.1976

Critical value for a significance level of:
                10%   5%  2.5%    1%
critical values 0.119 0.146 0.176 0.216 # 2.5% reikšmingumo lygiu H0 atmetame
                                        # taigi procesas turi vienetinę šaknį
```

****

```
> summary(pp.gnp)

###################################
# Phillips-Perron Unit Root Test #
###################################

Test regression with intercept and trend

Call:
lm(formula = y ~ y.l1 + trend)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.01374    9.93593   1.410    0.164
y.l1         0.98538    0.03301  29.849   <2e-16 ***
trend        0.50203    0.32292   1.555    0.125
Value of test-statistic, type: Z-tau  is: -0.7734

           aux. Z statistics
Z-tau-mu              0.7316
Z-tau-beta            1.6657

Critical values for Z statistics:
                      1%        5%       10%
critical values -4.113484 -3.483605 -3.169576    # H0 neatmetame
```

****

```
> summary(ur.sp(l.gnp, pol.deg=1))

###################################
# Schmidt-Phillips Unit Root Test #
###################################

Value of test-statistic is: -2.7345
Critical value for a significance level of 0.01 0.05 0.1
is: -3.63 -3.06 -2.77                                  # H0 neatmetame
```

****

```
> summary(ur.za(l.gnp,model="both",lag=2))

###############################
# Zivot-Andrews Unit Root Test #
###############################

Test statistic: -5.0951
Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82 # H0 ~6% reikšm. neatmetame
```

3.5 UŽDUOTIS. Pasirinkite dar kokias nors tris laikines sekas iš `nporg` ir patikrinkite dviem testais ar jos turi vienetinę šaknį.

3.6 UŽDUOTIS. Iš Data\Stewart\jones.dat importuokite JAV bendrojo vidaus produkto (GDP) ir gyventojų skaičiaus (pop) 1880-1987 m. duomenis. Ištirkite ar `GDP` ir `GDP per capita` laikinės sekos turi vienetinę šaknį (su tinkama stacionarumo alternatyva). Pasirinkę „teisingą“ modelį, prognozuokite `GDP` dešimčiai metų į priekį.

3.7 UŽDUOTIS. `urca` pakete yra nemažai makroekonominių laikinių sekų. Duomenų sistema `Raotbl1` turi 5 stulpelius, kuriuos galima (nors ir nebūtina) paversti laikinėmis sekomis. Išsiaiškinkite, kokie duomenys yra 1-me, 3-me ir 5-me stulpeliuose bei ištirkite ar šios sekos turi vienetinę šaknį.

3.5. `arima` funkcija (2)

Hipotezes apie vienetines šaknis paprastai tikrinsime su `adf.test` funkcija iš `tseries` paketo

Grįžkime prie laikinės sekos `a010` tyrimo ir prisiminkime, kad ji turi vienetinę šaknį:

```
> library(tseries)
> adf.test(a010)
        Augmented Dickey-Fuller Test

Dickey-Fuller = -2.3874, Lag order = 5, p-value = 0.415
alternative hypothesis: stationary
```

tačiau dabar dar įsitikinsime, kad skirtumų seka vienetinės šaknies jau neturi (yra stacionari):

```
> adf.test(diff(a010)) # DF testas skirtumams

        Augmented Dickey-Fuller Test

data:  diff(a010)
Dickey-Fuller = -5.4927, Lag order = 5, p-value = 0.01  # Seka diff(a010) yra
alternative hypothesis: stationary                      # stacionari
```

Taigi `y=a010` yra I(1) procesas. Sudarysime jo modelį[12].

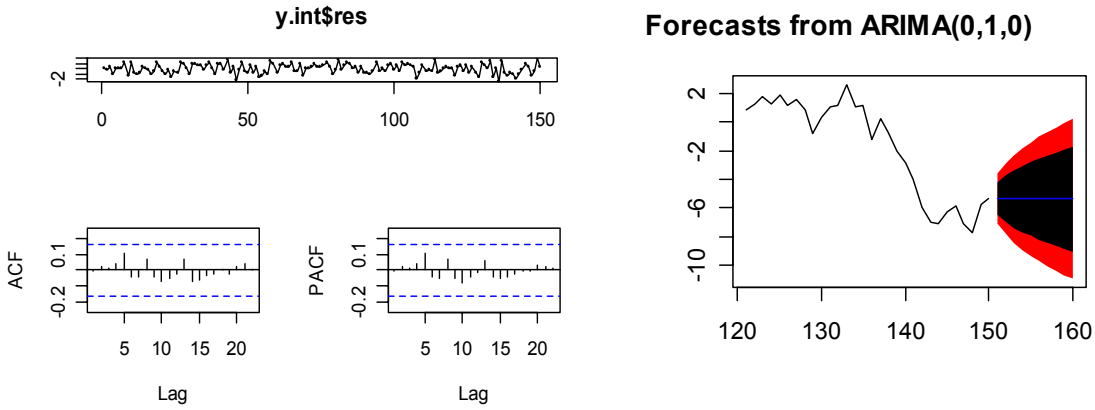
```
> y=a010                          # Žymėsime trumpiau
> (y.int=arima(y, c(0,1,0)))      # Šie du nuliai reiškia, kad nėra nei AR nei MA
                                  # narių (t.y., modelio paklaidos nekoreliuotos)
sigma^2 estimated as 0.8172:  log likelihood = -196.39,  aic = 394.77
```

Čia tarėme, kad $\Delta y_t = y_t - y_{t-1} = w_t$, funkcija `arima` šiuo atveju vertina tik w_t dispersiją (priminsime, su funkcija `rnorm(200)` mes generavome standartinius normaliuosius dydžius su vienetine dispersija – jos įvertis yra 0,8172). Pačio proceso struktūra yra $y_t = y_{t-1} + w_t$, o prognozės formulė yra $\hat{y}_{t+s} \equiv y_t,\ s \geq 1$.

Įsitikinsime, kad sudarytasis modelis geras, t.y., jo likučiai $y_t - \hat{y}_t = y_t - y_{t-1}$ sudaro baltąjį triukšmą:

```
library(forecast)
tsdisplay(y.int$res)
```

[12] Prisiminkime, kad nulinė hipotezė (kurios mes neatmetame) buvo *stebime atsitiktinį klaidžiojimą su* (galbūt, nenuliniu) *dreifu* (kitaip sakant, į modelį, gal būt, reiktų įtraukti ir dreifą). Nors iš `a010` grafiko (žr. 3.5 pav.) matyti, kad ši seka, tikriausiai, dreifo neturi, pastarąjį teiginį galima patikrinti ir formaliai. Tai galima atlikti dviem būdais: summary(`lm(diff(a010)~1)`) arba `arima(a010,c(0,1,0),xreg=1:150)`. Bet kuriuo atveju dreifas lygus -0.0296 su standartine paklaida 0.0743 - taigi dreifas nereikšmingas ir jį įtraukti į modelį nėra reikalo.



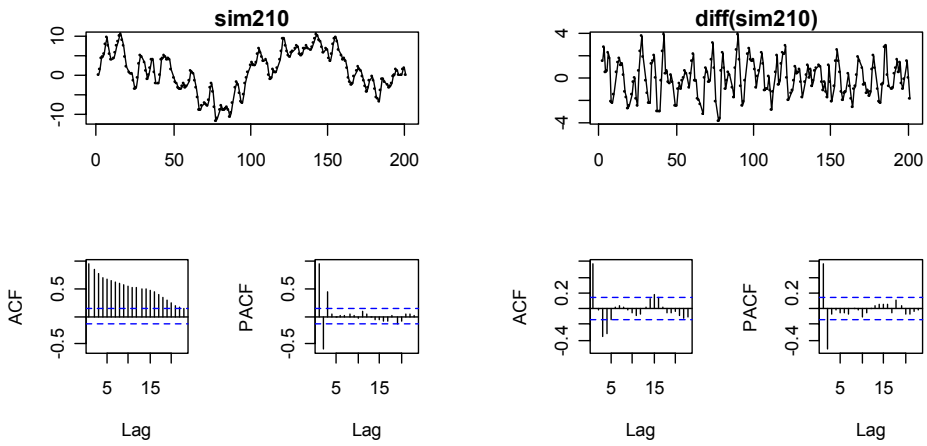
3.12 pav. Modelio `y.int` likučių grafikas (kairėje; likučiai sudaro baltąjį triukšmą) ir modeliu `y.int` nusakoma prognozė (dešinėje; prognozė tapatingai lygi paskutinei y_t reikšmei)

Prognozuoti tolimesnes y_t reikšmes galima su `predict` arba `forecast` funkcijomis.

```
plot(forecast(y.int),include=30) # 3.12 pav., brėžinys dešinėje
```

3.1 pavyzdys. Grįškime prie ARIMA procesų ir išnagrinėkime dar vieną pavyzdį. Modeliuokime ARIMA(2,1,0) procesą $\Delta y_t = 0.88\Delta y_{t-1} - 0.48\Delta y_{t-2} + w_t$ (šis skirtumų procesas yra stacionarus), arba, kitaip sakant, (nestacionarų) AR(3) procesą $y_t = 1.88y_{t-1} - 1.36y_{t-2} + 0.48y_{t-3} + w_t$ (kadangi pastarasis AR(3) procesas nėra stacionarus, `arima.sim` atsisakys jį generuoti; antra vertus, galima generuoti (stacionarius) šio proceso skirtumus, o paskui su `diffinv` apskaičiuoti jų „antiskirtumus“).

```
set.seed(2) # Pasirenkame „šokų“ seką ir generuojame ARIMA(2,1,0) procesą
sim210=arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))) # Skirtumai
tsdisplay(sim210)        # Pradiniai duomenys nėra stacionarūs
tsdisplay(diff(sim210)) # Skirtumų procesas yra stacionarus
```



3.13 pav. `sim210` (kairėje) yra nestacionarus (ir, ko gero, AR(3)) procesas, o skirtumų procesas `diff(sim210)` (dešinėje) – jau stacionarus ir, sprendžiant pagal ACF ir PACF, ARMA(2,0) procesas

Sprendžiant pagal `diff(sim210)` ACF ir PACF grafikus, skirtumų procesas yra AR(2) procesas, taigi pats `sim210` yra ARIMA(2,1,0) procesas. Rasime jo koeficientus.

```
> (fit210 <- arima(sim210, c(2, 1, 0)))
Coefficients:
         ar1      ar2
      0.8840  -0.5298    # Generavome su 0,88 ir -0,48
s.e.  0.0602   0.0604
sigma^2 estimated as 1.141:  log likelihood = -297.52,  aic = 601 # Dw_t įvertis
```

Galima rasti ir nediferencijuoto proceso koeficientus:

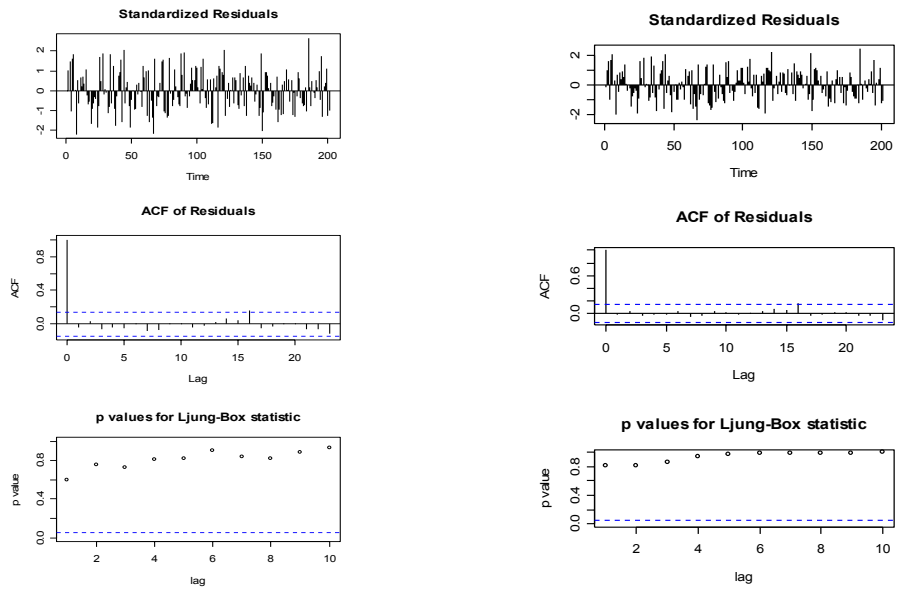
```
> (fit300 <- arima(sim210, c(3, 0, 0)))  # Ieškome nediferencijuoto proceso koeficientų

Coefficients:
         ar1      ar2     ar3  intercept
      1.8363  -1.3596  0.4844     0.9606 # Koefificientai artimi "tikrosioms" reikšmėms
s.e.  0.0617   0.1071  0.0618     1.7821 # 1,88, -1,36 ir 0,48; laisvasis narys nėra
                                         # reikšmingas
sigma^2 estimated as 1.104:  log likelihood = -297.24,  aic = 604.47
```

Modelį be laisvojo nario galima gauti taip (jo AIC geresnis, todėl rinksimės jį):

```
> (fit300F <- arima(sim210, c(3, 0, 0), include.mean=FALSE))

Coefficients:
         ar1      ar2     ar3
      1.8381  -1.3613  0.4858
s.e.  0.0616   0.1071  0.0618
sigma^2 estimated as 1.105:  log likelihood = -297.38,  aic = 602.75
```

Patikrinsime sudarytų modelių kokybę.

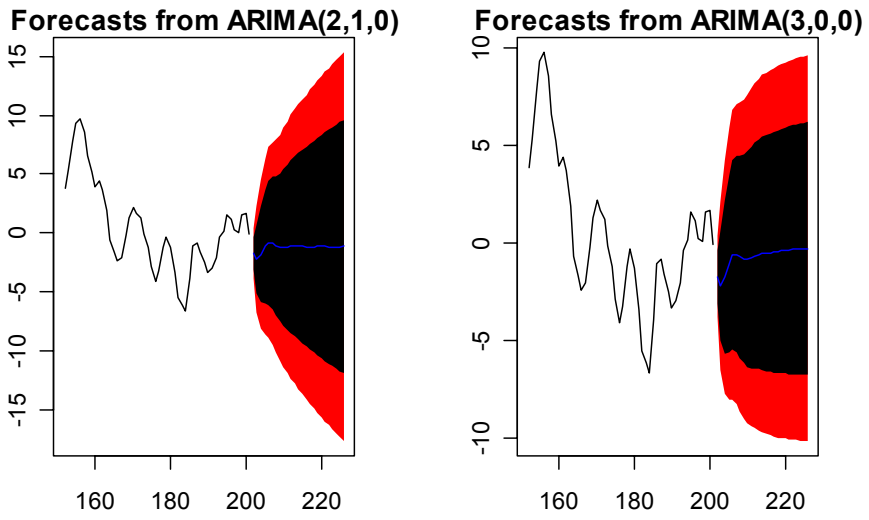
```
tsdiag(fit210)  # Ar likučiai sudaro baltąjį triukšmą? Atsakymas abiem atvejais
tsdiag(fit300F) # yra „taip" (žr. 3.14 pav.)
```



3.14 pav. Abu modeliai (`fit210` kairėje ir `fit300F` dešinėje) patenkinamai aprašo duomenis

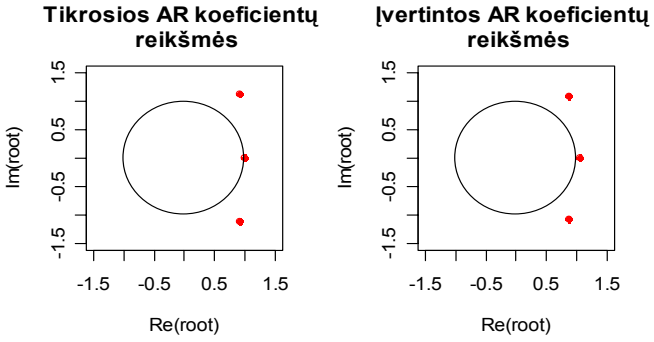
Išbrėšime du prognozės grafikus (žr. 3.15 pav. žemiau).

```
par(mfrow=c(1,2))
plot(forecast(fit210,25),include=50)
plot(forecast(fit300F,25),include=50)
```



3.15 pav. Dviejų modelių, `fit210` ir `fit300F`, prognozės grafikai

Matome, kad abi procedūros (`fit210`: pirma išdiferencijuoti, o paskui įvertinti parametrus ir `fit300F`: parametrus vertinti iš karto) pateikia išoriškai panašias prognozes. Antra vertus, antroji procedūra šį kartą pateikia įverčius iš stacionarių procesų srities[13] (t.y., su atvirkštinės charakteristinės lygties šaknimis moduliu didesnėmis už 1, žr. 3.16 pav.), todėl integruoto proceso `fit210` prognozės pasikliovimo sritis plečiasi kaip kvadratinė šaknis, o `fit300F` – konverguoja į konstantą.



3.16 pav. Duomenis generuojančio proceso atvirkštinės charakteristinės lygties šaknys (kairėje) yra `1.00+0.00i`, `0.92+1.11i` ir `0.92-1.11i` (t.y., vienos šaknies modulis lygus 1), o generuoto proceso (dešinėje) - `1.06+0.00i`, `0.87+1.09i` ir `0.87-1.09i` (t.y., visos jos moduliu didesnės už 1)

Pastaba. Laikinių sekų analizė didžia dalim priklauso nuo sėkmės (ir patyrimo). Žemiau pateiktos penkios tik ką nagrinėto ARIMA(2,1,0) proceso realizacijos – tik dviem atvejais (jie pažymėti žalia

[13] Jei įverčiai atitiktų nestacionarų procesą, R apie tai informuotų.

spalva) iš penkių gavome „teisingą“ atsakymą (t.y., `auto.arima`, remdamasi AIC minimumo principu, nustatė teisingą proceso tipą). Taigi AIC minimumo principas yra „protingas“, tačiau nebūtinai „teisingas“.

```
> set.seed(1) # Pasirenkama „šokų“ seka
> auto.arima(arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))))

Coefficients:
          ar1    ar2      ar3      ar4     ma1
      -0.1116  0.297  -0.4128  -0.0608  0.9866
s.e.   0.0744  0.072   0.0711   0.0747  0.0329

sigma^2 estimated as 0.8695:  log likelihood = -270.79,  aic = 553.58

> set.seed(2)  # Pasirenkama kitą „šokų“ seką
> auto.arima(arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))))

Coefficients:
         ar1      ar2
      0.8840  -0.5298
s.e.  0.0602   0.0604

sigma^2 estimated as 1.141:  log likelihood = -297.52,  aic = 601.05

> set.seed(3)
> auto.arima(arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))))

Coefficients:
         ar1      ar2
      0.8623  -0.4617
s.e.  0.0632   0.0631

sigma^2 estimated as 1.028:  log likelihood = -286.99,  aic = 579.97
> set.seed(4)
> auto.arima(arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))))

Coefficients:
         ar1      ar2     ar3     ar4      ar5  intercept    drift
      1.6972  -1.0894  0.2282  0.2541  -0.1525     9.8663  -0.0927
s.e.  0.0695   0.1376  0.1573  0.1385   0.0718     2.1044   0.0176

sigma^2 estimated as 0.9003:  log likelihood = -276.44,  aic = 568.89

> set.seed(5)
> auto.arima(arima.sim(n=200,list(order = c(2,1,0), ar = c(0.88,-0.48))))

Coefficients:
         ar1      ar2      ma1
      1.1532  -0.6305  -0.3308
s.e.  0.1017   0.0695   0.1265

sigma^2 estimated as 0.9211:  log likelihood = -276.11,  aic = 560.22
```

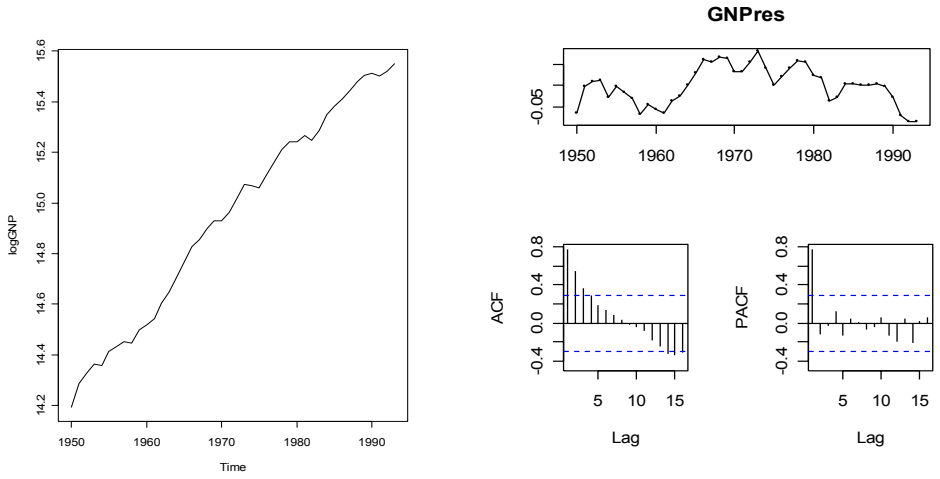
3.8 UŽDUOTIS. Sumodeliuokite ir ištirkite ARIMA(0,1,1) procesą $Y_t = Y_{t-1} + \varepsilon_t - 0.8\varepsilon_{t-1}$.

3.9 UŽDUOTIS. Ištirkite `forecast` paketo duomenis `ibmclose`. Ar jie stacionarūs? Gal jų skirtumai stacionarūs? Su `auto.arima` parinkite geriausią (kokia prasme?) modelį, aprašantį `ibmclose` (ne `diff(ibmclose)`!). Patikrinkite jį su `tsdiag`. Su `forecast` prognozuokite `ibmclose` 10 dienų į priekį ir išbrėžkite grafiką.

3.2 pavyzdys. Laikinė seka

```
lGNP = structure(c(14.1923, 14.2862, 14.328, 14.3646, 14.3577, 14.4125, 14.4322, 14.4513,
14.4464, 14.4995, 14.5197, 14.5445, 14.6042, 14.6470, 14.7053, 14.7661, 14.8290, 14.8546,
14.8987, 14.9293, 14.9293, 14.9622, 15.0173, 15.0738, 15.0680, 15.0592, 15.1122, 15.1624,
15.2127, 15.2413, 15.2413, 15.2679, 15.2486, 15.2868, 15.3513, 15.3836, 15.4119, 15.4435,
15.4807, 15.5055, 15.5138, 15.5017, 15.5221, 15.5517), .Tsp = c(1950, 1993, 1), class =
"ts")
```

yra JAV metinio bendrojo nacionalinio produkto logaritmų seka (nuo 1950 m. iki 1993 m., iš viso 44 skaičiai) (plg. 3.2 užduotį).



3.17 pav. `logGNP` grafikas (kairėje) ir tiesinio modelio likučių grafikas (dešinėje)

Kadangi duomenys turi akivaidų (beveik tiesinį) trendą, nagrinėsime du variantus: tai procesas **i**) su determinuotuoju trendu ir **ii**) su stochastiniu trendu (ir, gal būt, dreifu). **Pirmuoju** atveju panagrinėkime tiesinio modelio likučius.

```
TIME=time(lGNP)        # Laikinės sekos lGNP datų laikinė seka
GNPlm=lm(lGNP~TIME)    # Sudarome tiesinį modelį
GNPres=GNPlm$res       # Modelio likučiai
tsdisplay(GNPres)      # Žr. 3.17 paveikslą (dešinėje)
```

Sprendžiant pagal `GNPres` ACF ir PACF grafikus, likučiai turi didelį pirmąjį autoregresijos koeficientą (empirinis koeficientas lygus maždaug 0,8, o teorinis - gal net vienetui). Kadangi modelio likučiai nėra baltasis triukšmas (o gal likučiai turi dar ir vienetinę šaknį? - patikrinkite su Dikio ir Fulerio testu), visa mažiausių kvadratų procedūra yra klaidinga ir gautieji modelio koeficientai yra neteisingi. Vis dėlto, jei naudotumėmes gautais koeficientais

```
> summary(GNPlm)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.840e+01  1.030e+00  -46.99   <2e-16 ***
TIME         3.213e-02  5.224e-04   61.50   <2e-16 ***

Residual standard error: 0.04401 on 42 degrees of freedom
Multiple R-Squared: 0.989,      Adjusted R-squared: 0.9888
F-statistic:  3782 on 1 and 42 DF,  p-value: < 2.2e-16
```

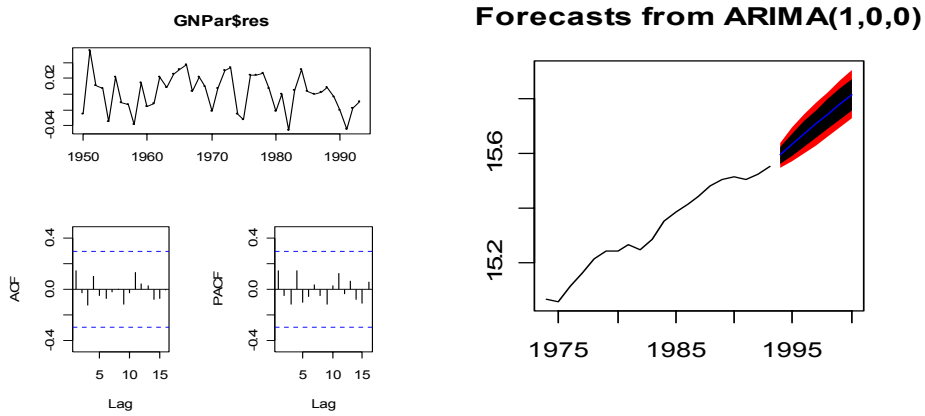
tai galėtume padaryti išvadą, kad `lGNP` kasmet vidutiniškai padidėja $3{,}213\cdot10^{-2}$, t.y., pats GNP padidėja 3,2652% (kodėl?).

Gautą tiesinį modelį galime pataisyti taip: `lGNP` interpretuokime kaip procesą su tiesiniu trendu ir autoregresiniais AR(1) likučiais:

```
> (GNPar100=arima(lGNP,c(1,0,0),xreg=TIME))
Coefficients:
         ar1  intercept     xreg
      0.8685   -47.8688   0.0319
s.e.     NaN     2.8677   0.0014  # Visi koeficientai reikšmingi

sigma^2 estimated as 0.000562:  log likelihood = 101.51,  aic = -195.02
```

Modelio diagnostiniai grafikai sako, kad jis visai tinkamas:

```
tsdisplay(GNPar100$res) # Modelis priimtinas, nes likučiai balt. triukšmas
plot(forecast(GNPar100,h=7,xreg=1994:2000),include=20)
```



3.18 pav. `GNPar100` modelio likučiai (kairėje) ir šio modelio nusakyta prognozė (dešinėje)

Kadangi šis modelis, t.y., modelis $lGNP_t = -47.8688 + 0.0319 * t + u_t$ (čia $u_t = 0.8685 u_{t-1} + w_t$, `t=1950,...,1993`) yra visai priimtinas, patikslintas vidutinis `lGNP` metinis didėjimas lygus $3{,}19 \cdot 10^{-2}$, kas beveik sutampa su ankstesniuoju rezultatu.

Jau matėme, kad tiesinio modelio likučiai turi didelį pirmąjį autoregresijos koeficientą. Ar gali būti taip, kad šis koeficientas lygus 1? Pasirodo, kad taip ir yra – Dikio ir Fulerio testo *p* reikšmė lygi 0,9483, todėl nėra jokio pagrindo atmesti hipotezę apie tiesinio modelio likučių (kadangi likučiai nėra baltasis triukšmas, visa šita tiesinės regresijos procedūra yra tik apytikslė!) vienetinę šaknį:

```
> adf.test(GNPres)

Dickey-Fuller = -0.8615, Lag order = 3, p-value = 0.9483
alternative hypothesis: stationary
```

Kadangi tiesinio modelio likučiai turi vienetinę šaknį, tikslinga panagrinėti **antrąjį** variantą: gal ir pats `lGNP` turi vienetinę šaknį (ir dar, ko gero, teigiamą dreifą)?

```
> adf.test(lGNP) # Automatiškai įtraukia konstantą ir trendą

Dickey-Fuller = -0.8615, Lag order = 3, p-value = 0.9483 # Turi vienetinę šaknį
alternative hypothesis: stationary
```

Kadangi `diff(lGNP)` yra baltasis triukšmas su nenuliniu vidurkiu[14], todėl jį galima aprašyti tokiu modeliu: $\Delta lGNP_t = a_0 + w_t$ arba $lGNP_t = a_0 + lGNP_{t-1} + w_t = ta_0 + \sum_{i=1}^{t} w_t$ [15]. Kitais žodžiais,

```
 > (GNPar010=arima(lGNP,c(0,1,0),xreg=time(lGNP)))

Coefficients:
         xreg  # tai dreifas[16](!!) - jis lygus mean(diff(lGNP))
       0.0316
s.e.   0.0037

sigma^2 estimated as 0.0005984:  log likelihood = 98.54,  aic = -193.09
```

Taigi atsakymas vėl panašus į ankstesnįjį - `lGNP` kasmet vidutiniškai padidėja $3{,}16*10^{-2}$.

Kuris iš modelių geresnis? Vienas „gerumo“ kriterijų yra vidutinė aproksimavimo paklaida, kurią galima apibrėžti keliais būdais, pvz.:

Root-Mean-Square Error:	$\mathrm{RMSE} = \sqrt{(1/n)\sum_{i=1}^{n} \hat{u}_i^2}$
Mean Absolute Error:	$\mathrm{MAE} = (1/n)\sum_{i=1}^{n} \lvert \hat{u}_i \rvert$
Mean Squared Error	$\mathrm{MSE} = (1/n)\sum_{i=1}^{n} \hat{u}_i^2$
Mean Absolute Percentage Error	$\mathrm{MAPE} = (1/n)\sum_{i=1}^{n} \lvert \hat{u}_i / y_i \rvert \times 100$
Mean Error	$\mathrm{ME} = (1/n)\sum_{i=1}^{n} \hat{u}_i$

Palyginsime abu modelius pagal dvi pirmąsias charakteristikas.

```
> sqrt(sum(GNPar100$res^2)/44)
[1] 0.02370719 # RMSE
```

ir

```
> sum(abs(GNPar100$res)/44)
[1] 0.01966845 #MAE
```

Kadangi atitinkami rodikliai stochastiniam modeliui panašūs:

```
> sqrt(sum(GNPar010$res^2)/44)
[1] 0.0252182259 # RMSE
> sum(abs(GNPar010$res)/44)
[1] 0.0199470026 #MAE
```

galime teigti, kad abu modeliai maždaug vienodi. Antra vertus, prognozių paklaidos nėra vienodos:

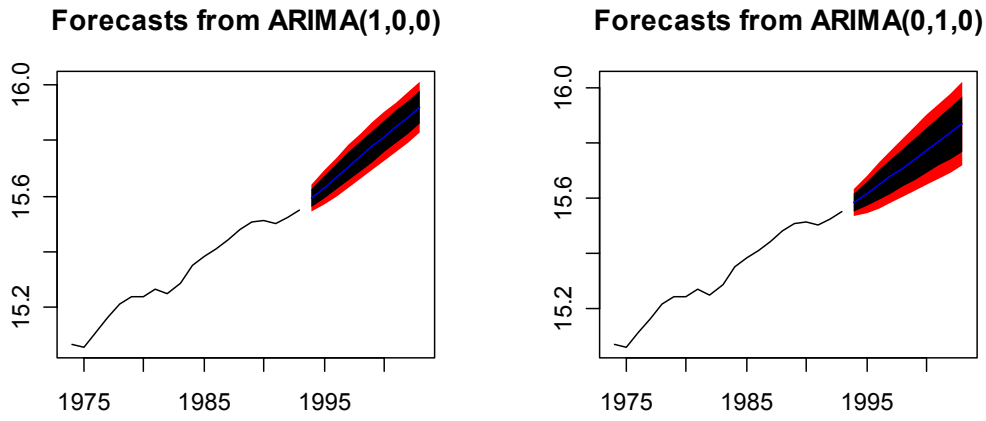
```
par(mfrow=c(1,2))
plot(forecast(GNPar100,10,xreg=1994:2003), include=20)
plot(forecast(GNPar010,10,xreg=1994:2003), include=20)
```

[14] Pabandykite `tsdisplay(diff(lGNP))`.

[15] Taigi regresinė dalis yra ta_0, o ARIMA dalis, t.y., $\sum_{i=1}^{t} w_t$, yra ARIMA(0,1,0) procesas.

[16] `best.arima(lGNP,d=1,max.p=1)` aiškiai rašo, kad 0.0316 yra `drift`.

Stacionariojo AR proceso (modelis `GNPar100`) prognozės pasikliauties intervalo plotis artėja prie konstantos.
Proceso su vienetine šaknimi (modelis `GNPar010`) prognozės pasikliauties intervalo plotis auga kaip kvadratinė šaknis.



3.19 pav. Determinuotasis trendas (modelis `GNPar100`, kairėje) ir stochastinis trendas (modelis `GNPar010`, dešinėje)

todėl, gal būt, geriau vartoti modelį `GNPar100`.

3.10 UŽDUOTIS. Atlikite panašią analizę su UK duomenimis iš Data\Internet_Misc\ gnp_1872_1993.txt failo.

3.11 UŽDUOTIS. Sudarykite pakete `forecast` esančių duomenų `dj` modelį, išbrėžkite jų 10 dienų prognozę.

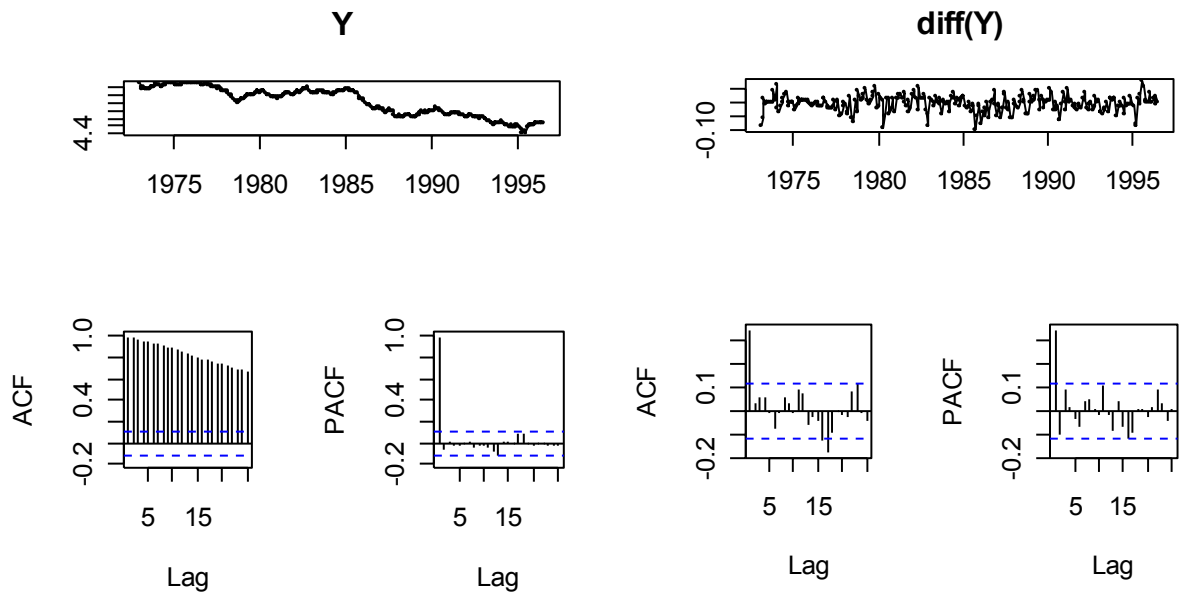
3.3 pavyzdys. Panagrinėsime jenos\JAV dolerio kurso laikinę seką (pateiktieji skaičiai žymi jenos ir dolerio kursą mėnesio pabaigoje, nuo 1973 m. 01 mėn. iki 1996 m. 07 mėn.). Surinkime komandą

```
YY=ts(read.table(file.choose(), header=TRUE), start=1973, freq=12)
```

ir nuvairuokime į Data\Diebold\EXCH.txt failą. Jenos kursas yra antrasis šios trimatės laikinės sekos stulpelis, mes nagrinėsime kurso logaritmus `Y` ir jų skirtumus `y=diff(Y)`.

```
> (Y=log(YY[,2]))
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
1973 5.709720 5.625683 5.567767 5.581583 5.578402 5.577820 5.578005 5.580574 5.581529 5.584747 5.628462 5.635414
1974 5.697035 5.673244 5.642104 5.626794 5.631080 5.645325 5.673098 5.711351 5.700617 5.701665 5.704033 5.705143
.........................
```

```
tsdisplay(Y)
tsdisplay(diff(Y))
```



3.20 pav. Jenos kurso logaritmų grafikas (kairėje) ir logaritmų skirtumų (grąžų) grafikas (dešinėje; grąžos sudaro AR(1) arba AR(2) procesą)

Laikinę seką `Y` ištirsime trimis būdais, kuriuos šį kartą palyginsime kiek kitaip: „sutrumpintą“ modelį sudarysime pagal 1973.01-1994.12 duomenis, o paskui palyginsime šio modelio prognozę iki 1996.07 su tikraisiais duomenimis (apskaičiuosime RMSE ir MAE).

1. 3.20 pav. matyti, kad `Y` yra I(1) procesas. Apibrėžę sutrumpintą laikinę seką

```
Ysh=window(Y,start=1973,end=c(1994,12)) # Ysh=Y short
TIME=time(Y)          # Tai datų laikinė seka
TIMEsh=time(Ysh)      # Tai sutrumpinto proceso datų seka
TIMErest=time(window(Y,start=c(1995,1),end=c(1996,7))) # Likęs laikas
```

nesunkiai įsitikinsime, kad `Ysh` yra procesas su vienetine šaknimi:

```
> adf.test(Ysh)

        Augmented Dickey-Fuller Test

data:  Ysh
Dickey-Fuller = -2.4098, Lag order = 6, p-value = 0.4033 # Vienetinė šaknis
alternative hypothesis: stationary
```

Procesas `Y` labai panašus į atsitiktinį klaidžiojimą su dreifu, todėl sudarysime tokį modelį:

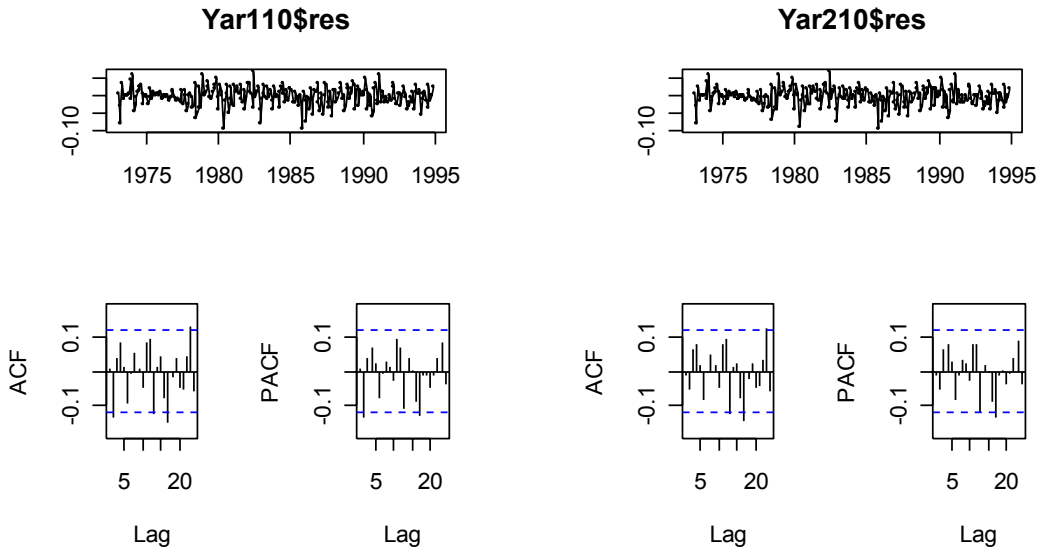
```
> (Yar010=arima(Ysh,c(0,1,0),xreg=time(Ysh)))

Coefficients:
         xreg
      0.0026
s.e.  0.0212

sigma^2 estimated as 0.0007795:  log likelihood = 567.94,  aic = -1131.89
```

Kadangi dreifo koeficientas 0.0026 nereikšmingas, apsiribosime atsitiktinio klaidžiojimo modeliu.

```
> (Yar010=arima(Ysh,c(0,1,0)))
Series: Ysh
ARIMA(0,1,0)(0,0,0)[12] model

sigma^2 estimated as 0.0007776:  log likelihood = 568.27,  aic = -1134.53
```

Deja šis modelis nėra priimtinas, nes komanda `tsdisplay(Yar010$res)` demonstruoja, kad likučiai yra AR(1) arba AR(2) procesas. Patikslinsime savo modelį:

```
> (Yar110=arima(Ysh,c(1,1,0)))

Coefficients:
         ar1
      0.3463
s.e.  0.0589

sigma^2 estimated as 0.000687:  log likelihood = 584.48,  aic = -1164.97

> tsdisplay(Yar110$res)
> (Yar210=arima(Ysh,c(2,1,0)))

Coefficients:
         ar1      ar2
      0.3760  -0.0914
s.e.  0.0621   0.0625

sigma^2 estimated as 0.0006815:  log likelihood = 585.55,  aic = -1165.1

> tsdisplay(Yar210$res)
```



3.21 pav. `Yar110` modelio likučiai nesudaro baltojo triukšmo (kairėje), o `Yar210` – sudaro (dešinėje)

Taigi procesą `Ysh` aprašysime modeliu su vienetine šaknimi `Yar210` (šio proceso pirmieji skirtumai yra stacionarus AR(2) procesas) $(\Delta Ysh)_t = 0.3760(\Delta Ysh)_{t\text{-}1} - 0.0914(\Delta Ysh)_{t\text{-}2} + w_t$.

2. Dabar iš duomenų išskirsime tiesinį trendą ir patikrinsime tiesinio modelio likučius. Kadangi likučiai, ko gero[17], sudaro AR(1) arba AR(2) procesą, Ysh_t modeliuosime pagal formulę $a_0 + \beta t + Xsh_t$ (čia Xsh_t yra ARIMA(1,0,0) arba ARIMA(2,0,0) procesas).

```
> (Yar100=arima(Ysh,c(1,0,0),xreg=TIMEsh))

Coefficients:
         ar1  intercept     xreg
      0.9783   104.0473  -0.0498
s.e.  0.0115    15.7594   0.0079

sigma^2 estimated as 0.0007507:  log likelihood = 573.5,  aic = -1138.99

> (Yar200=arima(Ysh,c(2,0,0),xreg=TIMEsh))

Coefficients:
         ar1      ar2  intercept     xreg
      1.3171  -0.3461   104.7444  -0.0502
s.e.  0.0586   0.0589    13.6401   0.0069

sigma^2 estimated as 0.0006632:  log likelihood = 589.72,  aic = -1169.45
```

Kadangi antrojo modelio AIC yra mažesnis ir jo likučiai aprašomi baltuoju triukšmu (surinkite `tsdisplay(Yar200$res)`), renkamės modelį `Yar200` – jis užrašomas pavidalu $Yar200_t = 104.74 - 0.0502t + u_t$; čia $u_t = 1.3171u_{t-1} - 0.3461u_{t-2} + w_t$.

Dabar išbrėšime abiejų modelių prognozę iki 1996.07 ir palyginsime su tikraisiais duomenimis.

```
par(mfrow=c(1,2))
plot(forecast(Yar210,19),include=50)
lines(Y,col=5)
plot(forecast(Yar200,19,xreg=TIMErest),include=50)
lines(Y,col=5)
```

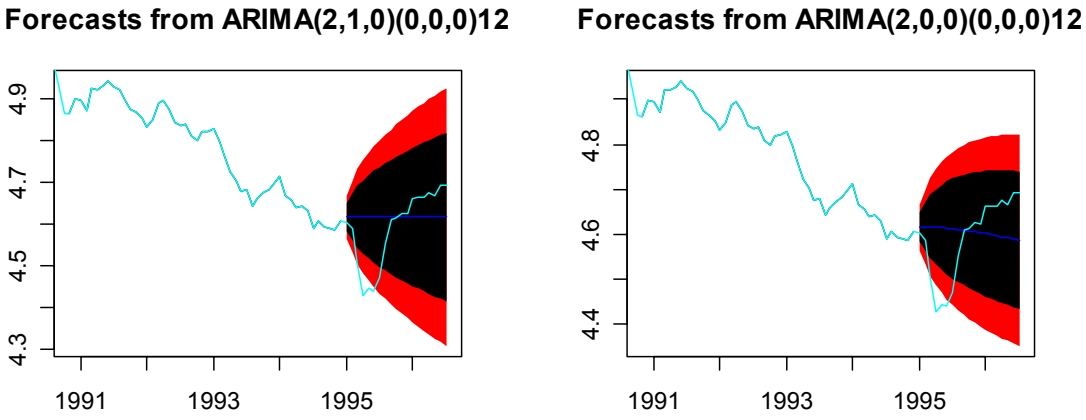
Kadangi „iš akies“ sunku pasakyti, kuris modelis tikslesnis (žr. 3.22 pav. žemiau), apskaičiuosime abiejų prognozių RMSE.

```
> (RMSE=sqrt( sum(( forecast(Yar210,19)$mean-Y[265:283]) ^2)/19))
[1] 0.0916146 # Modelis Yar210
> (RMSE=sqrt( sum(( forecast(Yar200,19,xreg=TIMErest)$mean-Y[265:283]) ^2)/19))
[1] 0.0971198 # Modelis Yar200
```

Matome, kad prognozės tikslumo prasme abu modeliai beveik vienodi.

Modelius galima klasifikuoti pagal turimų duomenų aprašymo tikslumą (tam vartojame R^2, AIC, RMSE modeliavimo srityje (angl. in sample) ir pan.) arba pagal prognozavimo tikslumą (tam vartojame RMSE arba jo analogus prognozavimo srityje (angl. out of sample)). Modelis geriausias viena prasme nebūtinai geriausias kita prasme.

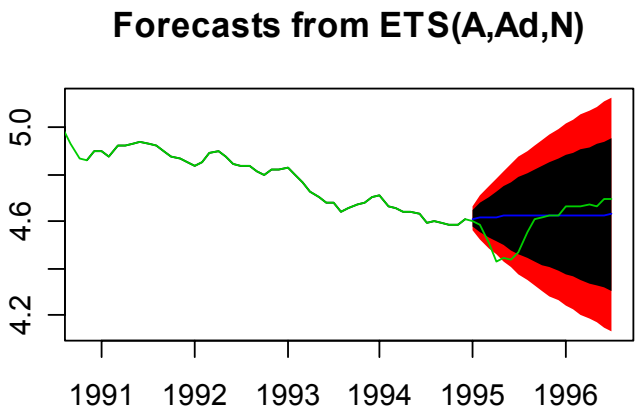
[17] „Ko gero“, nes jei taip iš tikro yra (t.y., jei likučiai nėra baltasis triukšmas), tai „tikrieji“ likučiai nesutampa su gautais (nes mažiausių kvadratų metodu tuomet pasitikėti negalima).



3.22 pav. Modelis `Yar210` (kairėje) ir `Yar200` (dešinėje); atkreipkite dėmesį į nenuspėjamai didelį jenos kurso kritimą 1995 m. balandžio –liepos mėnesiais

3. Išbandykime dar vieną modeliavimo ir prognozavimo metodą – eksponentinį glodinimą. Remsimės paketo `forecast` funkcija `ets`, kuri visiškai automatizuoja modelio parinkimą.

```
> fit=ets(Ysh)
> fit
ETS(A,Ad,N)

  Smoothing parameters:
    alpha = 0.99
    beta  = 0.2953
    phi   = 0.8

  Initial states:
    l = 5.7853
    b = -0.0938

  sigma:  0.0265
  AIC:    -435.4052
  AICc:   -435.1726
  BIC:    -417.5254
```




```
> plot(forecast(fit,h=19),include=50)
> lines(Y,col=2)
```

3.12 UŽDUOTIS. Apskaičiuokite šio modelio RMSE prognozavimo srityje.

3.13 UŽDUOTIS. Štai JAV tikrojo BVP (t.y., atsižvelgus į infliaciją, kitaip sakant, palyginamosiomis kainomis) ketvirtiniai duomenys (nuo 1960:1 iki 1991:4):

```
USrGDP=
structure(c(1866393, 1862336, 1864884, 1854409, 1868186, 1895647,
1921787, 1960383, 1987183, 2007661, 2023137, 2021439, 2051070,
2079003, 2112880, 2129300, 2184599, 2201774, 2226876, 2231877,
2274626, 2306239, 2347949, 2407683, 2455244, 2459868, 2486291,
2497049, 2512148, 2523094, 2552537, 2567353, 2602646, 2644356,
2660304, 2665683, 2705884, 2710696, 2725229, 2718340, 2710413,
2698334, 2733344, 2713339, 2781094, 2787417, 2805346, 2818369,
```

```
2868856, 2915945, 2948596, 2994836, 3069008, 3083446, 3080993,
3104396, 3075803, 3083541, 3056646, 3044662, 2976340, 3010689,
3066838, 3107038, 3167622, 3179418, 3190553, 3223865, 3271143,
3326442, 3373059, 3366265, 3389479, 3498190, 3525179, 3566701,
3567833, 3571136, 3593029, 3599729, 3615017, 3522348, 3523197,
3593973, 3643044, 3627851, 3646818, 3588877, 3544524, 3558679,
3542920, 3547827, 3570381, 3667579, 3722218, 3786104, 3859144,
3910574, 3931713, 3957947, 3983992, 4015133, 4066280, 4089400,
4143189, 4140547, 4164045, 4177728, 4208775, 4260960, 4302481,
4364952, 4393074, 4439785, 4467812, 4510466, 4538871, 4560198,
4572654, 4585960, 4605872, 4624273, 4627104, 4581619, 4552271,
4568030, 4588791, 4593793), .Tsp = c(1960, 1991.75, 4), class = "ts")
```

Sudarykite vieną ar kelis tinkamus `USrGDP` modelius ir su jais šią seką pratęskite vieneriems metams į priekį. Palyginkite gautą prognozę su tikrais duomenimis, kurių paieškokite internete.

3.14 UŽDUOTIS. 2 skyriuje nagrinėjome Kanados nedarbo duomenis `caemp` ir aprašėme juos kaip AR(2) procesą. Gal šią laikinę seką galima aprašyti procesu su vienetine šaknimi?

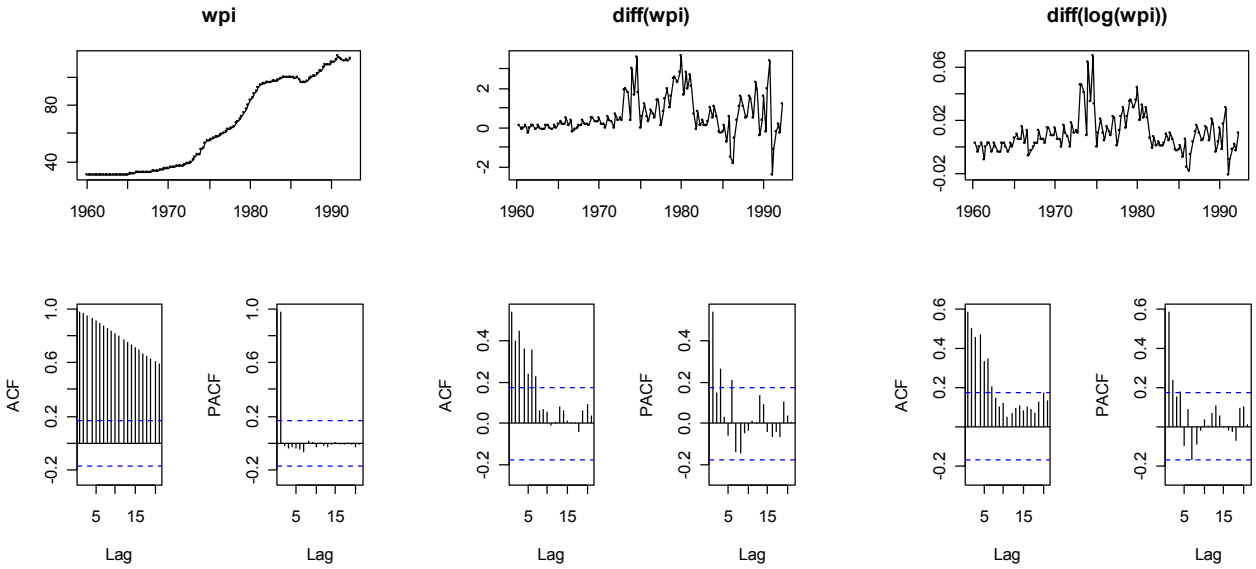
3.15 UŽDUOTIS. Išnagrinėkite `sheep` laikinę seką iš `fma` paketo. Ar tinkamai ją aprašo modelis $y_t = y_{t-1} + a_1(y_{t-1} - y_{t-2}) + a_2(y_{t-2} - y_{t-3}) + a_3(y_{t-3} - y_{t-4}) + w_t$? Įvardinkite šį modelį. Prognozuokite šios sekos reikšmes trims metus į priekį.

3.4 pavyzdys. Laikinę seką

```
wpi=structure(c(30.7, 30.8, 30.7, 30.7, 30.8, 30.5, 30.5, 30.6, 30.7,
30.6, 30.7, 30.7, 30.6, 30.5, 30.6, 30.7, 30.7, 30.6, 30.7, 30.7,
30.9, 31.2, 31.4, 31.6, 32.1, 32.2, 32.6, 32.4, 32.3, 32.3, 32.4,
32.5, 32.9, 33.1, 33.3, 33.4, 33.9, 34.4, 34.7, 35, 35.5, 35.7,
35.9, 35.9, 36.5, 36.9, 37.2, 37.2, 37.9, 38.3, 38.8, 39.2, 41.1,
43.1, 44.9, 45.3, 48.3, 50, 53.6, 55.4, 55.4, 56, 57.2, 57.8,
58.1, 59, 59.7, 60.2, 61.6, 63, 63.1, 63.9, 65.4, 67.4, 68.4,
70, 72.5, 75.1, 77.4, 80.2, 83.9, 85.6, 88.4, 90.4, 93.1, 95.2,
95.9, 95.8, 96.6, 96.7, 97.1, 97.2, 97.3, 97.6, 98.6, 99.1, 100.2,
100.8, 100.6, 100.3, 100.1, 100.2, 99.5, 100.1, 98.6, 96.8, 96.3,
96.7, 97.8, 99.4, 100.5, 101, 101.6, 103.2, 104.7, 105.2, 107.5,
109.4, 109, 109.4, 111, 110.8, 112.8, 116.2, 113.8, 112.7, 112.5,
112.7, 112.4, 113.6), .Dim = c(130, 1), .Dimnames = list(NULL,
"wpi"), .Tsp = c(1960, 1992.25, 4), class = "ts")
```

sudaro ketvirtiniai JAV didmeninės prekybos indeksai (angl. wholesale price index) nuo 1960 m. 1-ojo ketvirčio iki 1992 m. 2-ojo ketvirčio.

```
tsdisplay(wpi); tsdisplay(diff(wpi));tsdisplay(diff(log(wpi)))
```



3.23 pav. `wpi` (kairėje) yra akivaizdžiai nestacionarus procesas su, ko gero, vienetine šaknimi; `diff(wpi)` (viduryje) yra stacionarus vidurkių prasme, tačiau skirtumų kintamumas didėja kartu su `wpi`; logaritmų skirtumas (dešinėje) yra labiausiai panašus į stacionarų procesą (`diff(diff(log(wpi)))` būtų visai panašus į stacionarų, tačiau su dukart diferencijuotais procesais dirbama retai)

Laikinės sekos `dlwpi=diff(log(wpi))` modelį parinksime su `auto.arima`:

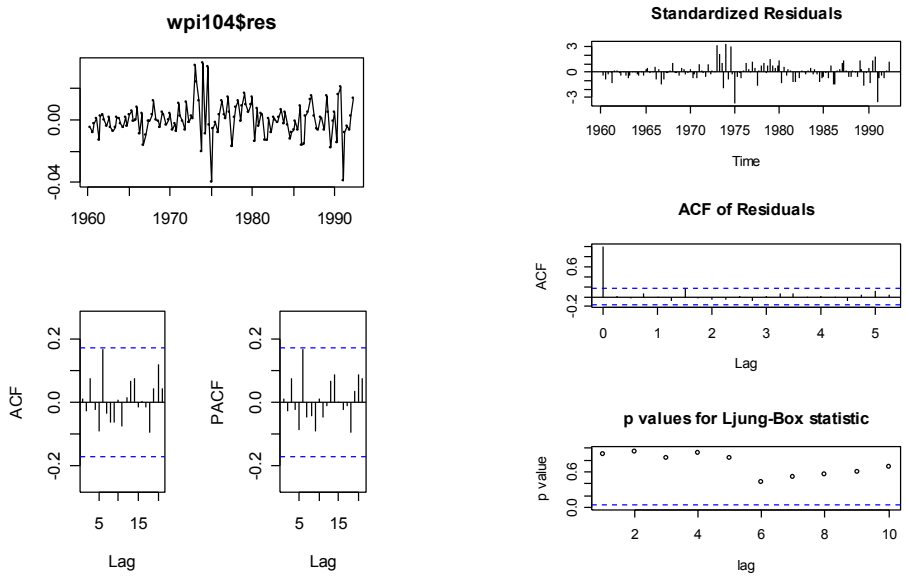
```
> auto.arima(dlwpi, d=0, D=0, max.p=5, max.q=5, max.P=0, max.Q=0, max.order=10,
alpha=0.05)

Coefficients:
         ar1      ar2     ar3      ma1     ma2  intercept
      1.7806  -1.2504  0.3537  -1.4330  0.8523     0.0098
s.e.  0.1323   0.1861  0.1020   0.1047  0.0803     0.0033

sigma^2 estimated as 0.0001163:  log likelihood = 400.58,  aic = -787.15
```

Ties šiuo modeliu būtų galima ir apsistoti, tačiau mes pateiksime samprotavimų grandinę, kuri atves prie dar geresnio modelio.

```
arima(dlwpi,order=c(1,0,0))$aic    # PACF grafiko pirmas stulpelis tikrai
[1] -775.1364                      # reikšmingas, todėl bandome AR(1) modelį

arima(dlwpi,order=c(2,0,0))$aic    # Reikšmingų stulpelių yra bent 4, todėl
[1] -780.6142                      # padidinkime AR eilę
arima(dlwpi,order=c(4,0,0))$aic
[1] -784.2706                      # Kol kas geriausiais AR(4) procesas

arima(dlwpi,order=c(1,0,1))$aic    # Kas būtų, jei įtrauktume MA narį?
[1] -785.25                        # Šis modelis kol kas geriausias
arima(dlwpi,order=c(1,0,4))$aic    # Mūsų duomenys ketvirtiniai, todėl gal būt
[1] -786.8855                      # verta atsižvelgti į praeitų metų atitinkamo
                                   # ketvirčio poveikį - gauname kol kas
                                   # geriausią modelį!
```

```
arima(dlwpi,order=c(1,0,4))
Series: dlwpi
ARIMA(1,0,4)(0,0,0)[4] model          # Konkretizuojame paskutinį modelį - kai
                                      # kurie koeficientai nereikšmingi
Coefficients:
         ar1      ma1      ma2     ma3     ma4  intercept
      0.7558  -0.3956  -0.0745  0.1057  0.3115     0.0098
s.e.  0.0878   0.1137   0.0917  0.0845  0.1106     0.0036
sigma^2 estimated as 0.0001167:  log likelihood = 400.44,  aic = -786.89

(wpi104=arima(dlwpi,order=c(1,0,4),fixed=c(NA,NA,0,0,NA,NA)))
                                      # Pašaliname nereikšmingus koeficientus
Coefficients:
         ar1      ma1  ma2  ma3     ma4  intercept
      0.7799  -0.4303    0    0  0.2888     0.0097
s.e.  0.0855   0.1058    0    0  0.1130     0.0036
sigma^2 estimated as 0.0001184:  log likelihood = 399.63,  aic = -789.26
```

Taigi tarp išnagrinėtųjų modelių `wpi104` yra geriausias (AIC prasme). Jo kokybę patikrinti galime su `tsdisplay(wpi104$res)` ir su `tsdiag(wpi104)`.



3.24 pav. Diagnostiniai grafikai teigia, kad modelis `wpi104` yra visai priimtinas

Taigi `wpi` elgesį aprašysime modeliu

$$(\Delta\log(wpi))_t = 0.0097 + 0.7799(\Delta\log(wpi))_{t\text{-}1} + w_t - 0.4303w_{t\text{-}1} + 0.2888w_{t\text{-}4}$$

(čia w_t yra baltasis triukšmas) arba

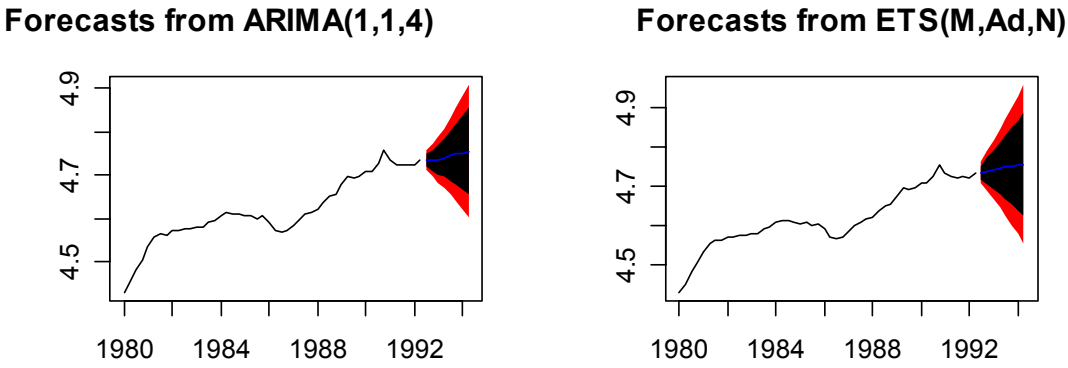
$$wpi_t = e^{0.0097} wpi_{t\text{-}1} \cdot \left(\frac{wpi_{t\text{-}1}}{wpi_{t\text{-}2}}\right)^{0.7799} e^{w_t - 0.4303w_{t\text{-}1} + 0.2888w_{t\text{-}4}}$$

Modelis `wpi104` jau leistų prognozuoti logaritmų skirtumų procesą. Norint prognozuoti patį `lwpi=log(wpi)`, modelį reikia užrašyti taip[18]:

```
(wpi114=arima(log(wpi),order=c(1,1,4),fixed=c(NA,NA,0,0,NA)))
Coefficients:
         ar1      ma1  ma2  ma3     ma4
      0.8732  -0.4909    0    0  0.2529
s.e.  0.0603   0.0913    0    0  0.1158
sigma^2 estimated as 0.0001224:  log likelihood = 397.32,  aic = -786.63
```

Savo modelius dar papildome eksponentinio glodinimo modeliu ir išbrėžiame du grafikus.

```
par(mfrow=c(1,2))
plot(forecast(wpi114,8),include=50)
plot(forecast(ets(log(wpi)),h=8),include=50) # Eksp. glodinimo modelis
```



3.25 pav. Du `lwpi` modeliai: ARIMA (kairėje) ir eksponentinio glodinimo (dešinėje)

Keletas pastabų.

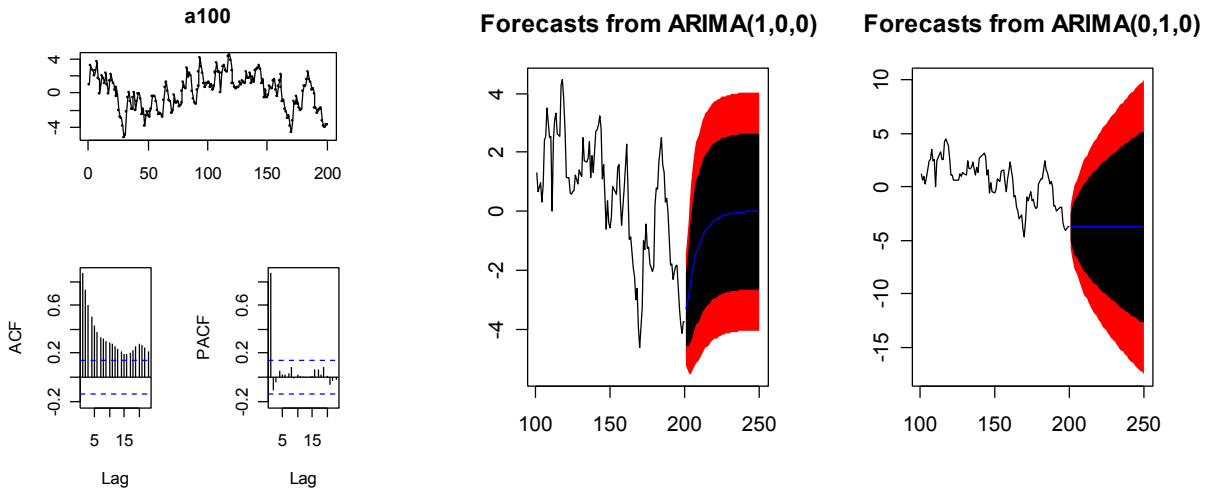
1. Laikinės sekos `wpi114` AR dalies koeficientas 0.8732 artimas vienetui, todėl galima pabandyti `lwpi` diferencijuoti du kartus (pabandykite).
2. Analizę galima pradėti ne diferencijavimu (t.y., stochastinio trendo pašalinimu), bet deterministinio trendo išskyrimu.
3. Mes atsižvelgėme į sezoniškumą, įtraukdami u_{t-4} narį. Sekančiame skyrelyje nagrinėsime kiek kitokį, vadinamąjį multiplikatyvųjį sezoninį modelį, užrašomą pavidalu $(1-a_1B)y_t=(1+b_1B)(1+b_4B^4)w_t$.

Šį skyrelį baigsime tokia bendro pobūdžio ***pastaba***. Jau matėme, kad dažnai tiriamąjį procesą patenkinamai aprašo keli modeliai, o pagrindinis skirtumas tarp stacionarių ir integruotų procesų yra prognozės pobūdis: stacionarių procesų prognozė artėja į vidurkį, o atsitiktinio klaidžiojimo – visą laiką lieka lygi paskutinei reikšmei.

```
a100=arima.sim(n = 200, list(ar = c(0.88)))
# Generavome stacionarų AR(1) procesą, tačiau jo koeficientas artimas 1
tsdisplay(a100)
par(mfrow=c(1,2))
```

[18] Plg. su `wpi104=arima(dlwpi,order=c(1,0,4),fixed=c(NA,NA,0,0,NA),include.mean=FALSE)` (diferencijuotiems procesams include.mean visuomet yra FALSE (tai, beje, neturi įtakos prognozei).

```
fit100=arima(a100,order=c(1,0,0)) # Vertiname modelį, tarę kad tai AR(1) seka
plot(forecast(fit100,50),include=100)
fit010=arima(a100,order=c(0,1,0)) # Vertiname modelį, tarę kad tai I(1) seka
plot(forecast(fit010,50),include=100)
```

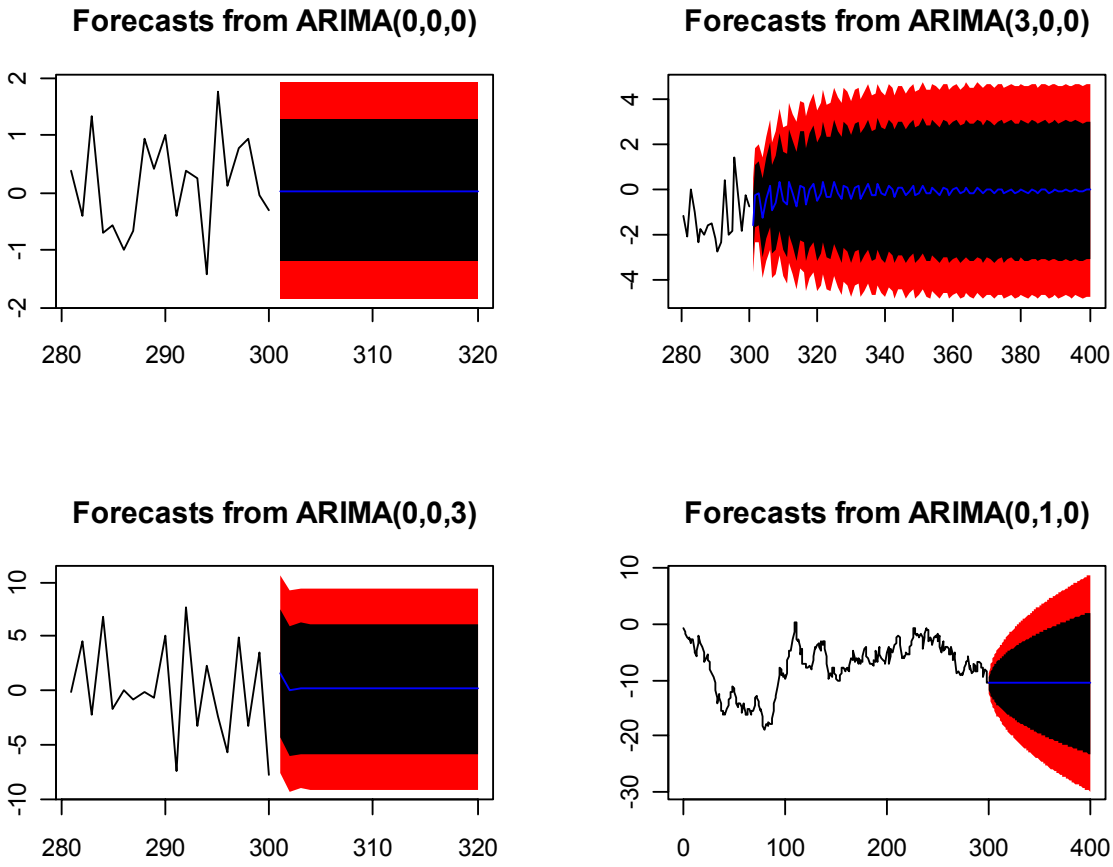


`a100` trajektorija panaši į atsitiktinį klaidžiojimą (kairėje) (beje, netgi `adf.test(a100)` testas neatmeta vienetinės šaknies hipotezės); antra vertus, modelių `fit100` ir `fit010` prognozės visai skirtingos (dešinėje)

Šitos pastabos moralas toks: proceso tipą reikia stengtis nustatyti kuo tiksliau (deja, kartais tai padaryti sunku...), nes klaidinga specifikacija gali duoti dideles paklaidas.

Padarysime kelias pastabas apie laikinių sekų prognozę – TS procesams prognozė artėja prie proceso trendo, o jos paklaida – prie proceso standarto; DS procesams be dreifo prognozė sutampa su paskutinia stebėta reikšme, o paklaidos standartas auga kaip kvadratinė šaknis.

```
opar=par(mfrow=c(2,2))
set.seed(1)
library(forecast)
# Generuojame baltąjį triukšmą
wn=arima.sim(n=300,list(order=c(0,0,0)))
fit.wn=arima(wn)
plot(forecast(fit.wn,20),include=20)
# Generuojame stacionarų AR(3) procesą
ar3=arima.sim(n=300,list(order=c(3,0,0),ar=c(0.2,-0.3,0.9)))
fit.ar3=arima(ar3,order=c(3,0,0))
plot(forecast(fit.ar3,100),include=20)
# Generuojame MA(3) procesą
ma3=arima.sim(n=300,list(order=c(0,0,3),ma=c(0.7,4,-2.1)))
fit.ma3=arima(ma3,order=c(0,0,3))
plot(forecast(fit.ma3,20),include=20)
# Generuojame atsitiktinį klaidžiojimą
rw=arima.sim(n=300,list(order=c(0,1,0)))
fit.rw=arima(rw,order=c(0,1,0))
plot(forecast(fit.rw,100),include=300)
par(opar)
```



3.27 pav. Baltojo triukšmo prognozė sutampa su proceso vidurkiu, AR(3) prognozė artėja prie vidurkio, MA(3) prognozė sutampa su vidurkiu po 3 žingsnių, atsitiktinio klaidžiojimo prognozė sutampa su paskutiniąja reikšme

****
****

Iš viso to, kas buvo anksčiau išdėstyta, aišku, kad vienetinės šaknies testavimas nėra lengvas (tai ypač liečia atvejį, kai procesas gali turėti trendą, žr., 3.2 lentelės rekomendacijas). Čia aprašysime gana paprastą procedūrą, kuri (kai $T > 100$) dažnai pateikia pakankamai tikslų atsakymą.

Procesą

$$\Delta y_t = \alpha + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + ... + \gamma_{p-1} \Delta y_{t-p+1} + \delta t + \varepsilon_t \qquad (\#)$$

(čia ε_t yra baltasis triukšmas) pavadinsime AR(p) procesu su determinuotoju trendu. Tyrimo tikslas – patikrinti hipotezę $H_0 : \rho = 0$ (kitaip sakant, nustatyti ar mūsų procesas turi vienetinę šaknį). Pirmiausiai nustatysime proceso eilę p[19], po to patikrinsime hipotezę $\delta = 0$ ir tada – hipotezę $H_0 : \rho = 0$.

[19] Vėlinių eilę galima nustatyti, bandant įvairias p reikšmes ir lyginant modelius pagal jų liekanų AIC arba SIC parametrų reikšmes arba (žr. [D], 347 psl.) remiantis klasikiniu t kriterijumi (pastarasis variantas aprašytas žemiau).

(1) Pasirinkite maksimalią vėlinio eilę $p_{\max}$ (pvz., $p_{\max} = 5$; proceso eilė p paprastai nėra žinoma iš anksto, ji turi būti tiek didelė, kad paklaidos ε_t sudarytų baltąjį triukšmą).

(2) MK metodu įvertinkite (#) modelio koeficientus. Jei nėra pagrindo atmesti hipotezės $\gamma_{p_{\max}-1} \neq 0$, pereikite prie (5) žingsnio su $\mathrm{AR}(p_{\max})$; priešingu atveju atlikite (3) žingsnį.

(3) MK metodu įvertinkite $\mathrm{AR}(p_{\max}-1)$ modelio

$$\Delta y_t = \alpha + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + \ldots + \gamma_{p_{\max}-2} \Delta y_{t-(p_{\max}-2)} + \delta t + \varepsilon_t$$

koeficientus ir patikrinkite hipotezę $\gamma_{p_{\max}-2} \neq 0$; jei jos nėra pagrindo atmesti, pereikite prie (5) žingsnio su $\mathrm{AR}(p_{\max}-1)$; priešingu atveju atlikite (4) žingsnį.

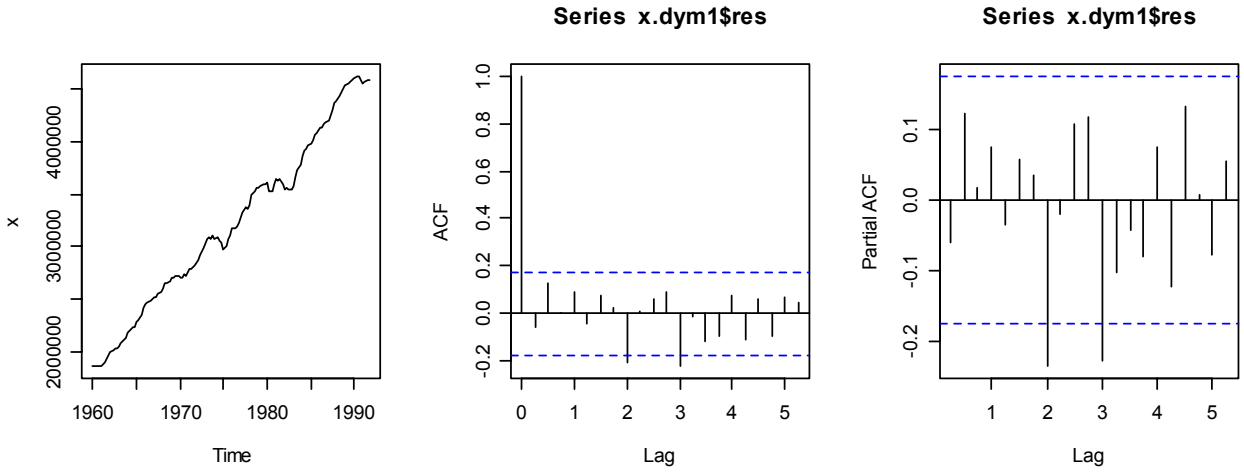
(4) AR modelio eilę mažinkite tol, kol aukščiausio vėlinio koeficientas taps reikšmingas (arba kol vėlinių iš viso nebeliks).

(5) MK metodu patikrinkite hipotezę $H_0 : \delta = 0$; jei ji teisinga, trendą pašalinkite iš modelio.

(6) Jei galutinis modelis yra su determinuotuoju trendu, Dickey-Fuller'io testo 5% kritinė reikšmė yra maždaug -3.45 (plg. 3.1 lentelę); taigi, jei ρ koeficiento t statistikos reikšmė yra mažesnė už šį skaičių, vienetinės šaknies hipotezę atmetame (taigi procesas stacionarus).

(7) Jei galutinis modelis determinuotojo trendo neturi, Dickey-Fuller'io 5% kritinė reikšmė yra maždaug -2.89; jei ρ koeficiento t statistikos reikšmė yra mažesnė už šį skaičių, vienetinės šaknies hipotezę atmetame (taigi procesas stacionarus).

3.5 pavyzdys. Panagrinėkime 3.13 užduotyje aptartus JAV realiojo BVP ketvirtinius duomenis (nuo 1960:1 iki 1991:4).



3.28 pav. `USrGDP` grafikas ir modelio `x.dyn1` liekanų ACF ir PACF grafikai (liekanos sudaro baltąjį triukšmą)

```
x=USrGDP
plot(x)
```

Panašu, kad x turi tiesinį trendą (nors gali būti, kad x auga dėl dreifo). Pasinaudosime dymlm paketu, kurio dymlm funkcija gerai pritaikyta darbui su vėliniais ir skirtumais. Pradėsime AR(5) procesu su determinuotoju trendu.

```
library(dynlm)
summary(dynlm(d(x) ~ L(x, 1) + L(d(x),1:4) + time(x)))
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.604e+07  5.292e+06  -3.031 0.003011 **
L(x, 1)       -9.504e-02  3.082e-02  -3.083 0.002558 **
L(d(x), 1:4)1  3.068e-01  8.965e-02   3.422 0.000858 ***
L(d(x), 1:4)2  1.471e-01  9.339e-02   1.575 0.117966
L(d(x), 1:4)3  5.694e-03  9.423e-02   0.060 0.951921
L(d(x), 1:4)4  9.933e-02  9.321e-02   1.066 0.288788
time(x)        8.275e+03  2.727e+03   3.034 0.002975 **
```

Koeficientas prie L(d(x),1:4)4 nereikšmingas, todėl AR eilę sumažinsime iki 4, o po to iki 3; tik tuomet, kai *p* taps lygiu 2, atitinkamas koeficientas taps reikšmingas:

```
x.dyn1 = dynlm(d(x) ~ L(x, 1) + L(d(x),1) + time(x))
summary(x.dyn1)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.241e+07  4.792e+06  -2.590   0.0108 *
L(x, 1)     -7.301e-02  2.786e-02  -2.621   0.0099 **
L(d(x), 1)   3.543e-01  8.447e-02   4.194 5.23e-05 ***
time(x)      6.407e+03  2.470e+03   2.594   0.0106 *
```

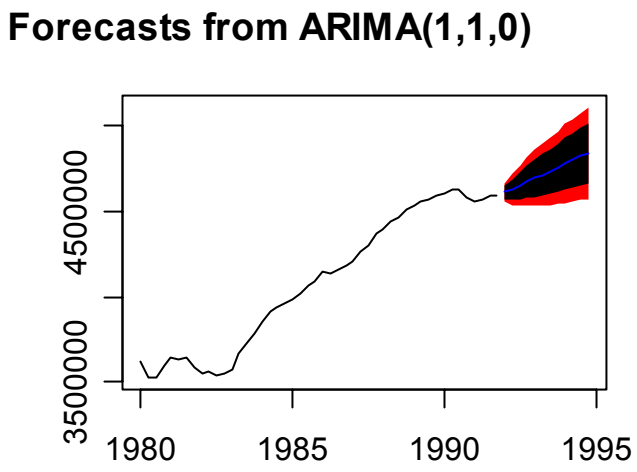
Kartu matyti, kad reikšmingas yra ir trendo koeficientas δ; kadangi koeficiento ρ *t*-statistika lygi -2.621 (ji nėra mažesnė už -3.45), neturime pagrindo atmesti vienetinės šaknies hipotezę (plg. [D, 356 psl.]).

Iki šiol mes nagrinėjome alternatyvųjį modelį (su tiesiniu trendu) $\Delta y_t = \alpha + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + ... + \gamma_{p-1} \Delta y_{t-p+1} + \delta t + \varepsilon_t$ ir nustatėme, kad nėra pagrindo atmesti nulinę hipotezę $\rho = 0$. Prisiminkime, kad nulinis modelis šiuo atveju yra vienetinės šaknies procesas su dreifu (žr. 3 – 14 psl., plg. [D, 346 psl., 355 psl. ir Table 12.7]) $\Delta y_t = \alpha + \gamma_1 \Delta y_{t-1} + \varepsilon_t$. Jo koeficientus nustatyti ir jį prognozuoti 12 ketvirčių į ateitį galima taip:

```
x.ur = arima(x, c(1,1,0), xreg=time(x)-1976) # Tai dreifo narys;
x.ur                                         # xreg vidurkis turi būti ≈ 0
```

```
Series: x
ARIMA(1,1,0)

Coefficients:
         ar1  time(x) - 1976
      0.3177             85290.74   # Tai dreifo koeficientas
s.e.  0.0840             14492.17

sigma^2 estimated as 781577340:  log likelihood = -1480.54
AIC = 2967.07   AICc = 2968.27   BIC = 2968.2
```

```
plot(forecast(x.ur,12,xreg=seq(1992,1994.75,by=0.25)-1976),include=48)
```



3.29 pav. Paskutinių 48 ketvirčių `USrGDP` grafikas ir jo 12 ketvirčių prognozė

3.16 UŽDUOTIS. Remdamiesi aukščiau aprašyta strategija, ištirkite dirbtines laikines sekas ...\DATA\Koop\Text\FIG.95.txt ir ...\DATA\Koop\Text\FIG96.txt.

3.17 UŽDUOTIS. Remdamiesi aukščiau aprašyta strategija, ištirkite, ar dirbtinė laikinė seka ...\DATA\Koop\Text\FIG98.txt turi vienetinę šaknį. O jos skirtumų seka?

3.18 UŽDUOTIS. Remdamiesi aukščiau aprašyta strategija, ištirkite, ar dirbtinė laikinė seka ...\DATA\Koop\Text\FIG98.txt turi vienetinę šaknį.

3.19 UŽDUOTIS. Remdamiesi aukščiau aprašyta strategija, ištirkite, ar abi laikinės sekos iš ...\DATA\Koop\Text\INCOME.txt turi vienetinę šaknį. Šiame faile pateikti JAV asmeninių pajamų ir vartojimo ketvirtiniai duomenys nuo 1954:1 iki 1994:4.

3.6. TS ir DS procesai

Jei laikinės sekos paklaidos (pašalinus trendą ir sezoninę dalį) sudaro stacionarų procesą $I(0)$, tai tokia laikinė seka vadinama TS seka (angl. Trend Stationary).
Jei laikinė seka nėra stacionari, bet jos skirtumai – stacionarūs, tai tokia laikinė seka vadinama DS seka (angl. Difference Stationary); ji žymima simboliu $I(1)$.

Paprasčiausias TS modelis yra užrašomas lygtimi $y_t = c_1 + c_2 t + w_t$. Papildę šią lygtį AR(1) nariu, gautume pakankamai įdomų modelį:

$$y_t = c_1 + a_1 y_{t-1} + c_2 t + w_t,\ t = 1,...,n.$$

Pateiksime jo kelis atskirus atvejus.

Proceso išraiška	Proceso pavadinimas	Proceso tipas	Koeficientų vertinimo proc.
$y_t = c_1 + a_1 y_{t-1} + c_2 t + w_t, \mid a_1 \mid < 1$	Laikinės sekos nuokrypiai nuo tiesinio trendo sudaro stacionarų AR(1) procesą	$I(0)$	`fit=arima(y, order=c(1,0,0), xreg=seq(1,n))` `plot(forecast(fit,m,xreg= seq(n+1,n+m),include=k)`
$y_t = c_1 + y_{t-1} + c_2 t + w_t$	Atsitiktinis klaidžiojimas su dreifu ir determinuotuoju trendu	$I(1)$	???
$y_t = c_1 + y_{t-1} + w_t$	Atsitiktinis klaidžiojimas su dreifu	$I(1)$	`fit=arima(y, order=c(0,1,0), xreg=seq(1,n))` `plot(forecast(fit,m,xreg= seq(n+1,n+m),include=k)`
$y_t = c_1 + c_2 t + w_t$	Determinuotasis trendas	$I(0)$	`fit=arima(y, order=c(0,0,0), xreg=seq(1,n))` `plot(forecast(fit,m,xreg= seq(n+1,n+m),include=k)`
$y_t = y_{t-1} + w_t$	Atsitiktinis klaidžiojimas	$I(1)$	`fit=arima(y, order=c(0,1,0))` `plot(forecast(fit,m),include=k)`

Kelis aukščiau pateiktus modelius pailiustruosime pastovaus augimo modeliu, aprašomo lygtimi

$$y_t = (1+R)y_{t-1}; \tag{3.1}$$

čia $R \times 100\%$ yra pastovaus „metinio“[20] augimo greitis procentais. Šios skirtuminės lygties sprendinį galima užrašyti dviem ekvivalenčiais būdais:

1.
$$y_t = y_0(1+R)^t; \tag{3.2a}$$

tai beje tas pat, kas

$$\log y_t = c_1 + c_2 t \tag{3.2b}$$

(čia $c_1 = \log y_0$, o $c_2 = \log(1+R)$), arba[21]

2.
$$(\log \frac{y_t}{y_{t-1}} = \log y_t - \log y_{t-1} =) \Delta \log y_t = c_2 \tag{3.3}$$

Aišku, kad realioje ekonomikoje augimo greitis nėra pastovus, todėl šias lygtis reikėtų papildyti atsitiktiniais elementais. Deja, jei vidutinis augimo greitis *r* yra teigiamas, lygtis $y_t = (1+r)y_{t-1} + w_t$ nusakys AR(1) procesą su koeficientu didesniu už 1 – tokie procesai nėra įdomūs (plg. 3-1 psl.). Perspektyvesnis variantas yra lygtis

$$\log y_t = c_1 + c_2 t + u_t; \tag{3.4a}$$

[20] Metinio, jei laikas matuojamas metais (ir, pvz., ketvirtinio, jei laikas matuojamas ketvirčiais). 4 skyriuje skaičių *R* pavadinsime paprastąja *y*-ko (metine) grąža.

[21] 4 skyriuje santykį $\log(y_t / y_{t-1})$ pavadinsime logaritmine *y*-ko grąža.

čia u_t yra koks nors stacionarus ARMA procesas. Šio TS proceso parametrai gali būti įvertinti (t.y. determinuotasis trendas išskirtas) arba su `lm` , arba su `arima` funkcijomis.

3.20 UŽDUOTIS. Nuskaitykite duomenis iš Data\Stewart\ASCII\jones.dat ir išnagrinėkite BVP vienam gyventojui laikinę seką. Kam lygus vidutinis augimo greitis c_2? ◀◀

Visai kitaip atsižvelgti į atsitiktinumą galima (3.3) lygtį perrašius pavidalu

$$\Delta \log y_t = c_2 + u_t . \qquad (3.4b)$$

Nežiūrint to, kad pradinės lygtys buvo ekvivalenčios, lygtys (3.4a) ir (3.4b) labai skiriasi – pastaroji yra DS seka.

3.6 pavyzdys. Štai lygtis, aprašanti JAV bendrojo vidaus produkto kitimą (pagal ketvirtinius duomenis nuo 1949 m. iki 1984 m.):

$$(\Psi_t =)\ \Delta \log y_t = (a_0 + a_1 \Delta \log y_{t-1} + w_t =)\ 0.005 + 0.406 \Delta \log y_{t-1} + w_t$$

Tai AR(1) modelis pirmiesiems skirtumams, taigi $\log y_t$ yra ARIMA(1,1,0) procesas su laisvuoju nariu. Antra vertus, šį procesą galima užrašyti (3.4b) pavidalu su $u_t = A_1 u_{t-1} + w_t$:

$$\Psi_t = c_2 + u_t = c_2 + A_1 u_{t-1} + w_t = c_2 + A_1(\Psi_{t-1} - c_2) + w_t =$$
$$= c_2(1 - A_1) + A_1 \Psi_{t-1} + w_t = a_0 + a_1 \Psi_{t-1} + w_t$$

taigi $c_2(1-0,406) = 0,005$, kitaip sakant, dreifas c_2 (vidutinis ketvirtinis prieaugis) lygus 0,008 (t.y., vidutinis metinis prieaugis lygus $4 \cdot 0,0084 = 0,036 = 3,6\%$).

Stacionaraus AR(p) proceso $y_t = a_0 + a_1 y_{t-1} + ... + a_p y_{t-p} + w_t$ vidurkis yra lygus $a_0 / (1 - a_1 - ... - a_p)$

Kiti autoriai vartojo tuos pačius JAV duomenis, tačiau nuo 1947:2 iki 1985:4, ir gavo modelį $\Delta \log y_t = 0,008 + w_t + 0,3 w_{t-1}$. Tai ARIMA(0,1,1) procesas su ketvirtiniu prieaugiu $c_2 = 0,008$ – rezultatas beveik sutampa su ankstesniu. ◀

3.21 UŽDUOTIS. Panašūs JAV duomenys yra pateikti 3.8 užduotyje. Pabandykite sudaryti savą modelius ar modelius. Prognozuokite šią seką metus į priekį. ◀

3.22 UŽDUOTIS. Nuskaitykite (su `yields=ts(scan(file.choose()))`) duomenis iš Data\Chan\Yields.dat. Tai mėnesinės 21-erių metų vienos Europos šalies pajamos iš trumpalaikių vyriausybinių obligacijų. Aprašykite šią laikinę seką kaip TS ir DS bei sudarykite atitinkamus modelius. Kurį iš jų pasirinktumėte? Prognozuokite `yields` 6 mėnesiams į priekį. ◀
Šiame skyrelyje, modeliuodami augimo procesą, visą laiką vartojome logaritmus.

Reikšmės ar jų logaritmai? Pastoviu greičiu augantys ekonominiai kintamieji paprastai turi būti logaritmuojami

3.23 UŽDUOTIS. Vieną paketo `Ecdat` duomenų rinkinio `Macrodat` stulpelių sudaro Japonijos BVP ketvirtiniai duomenys. Sudarykite jų ARIMA modelį ir apskaičiuokite vidutinį metinį prieaugį.

3.7 pavyzdys.

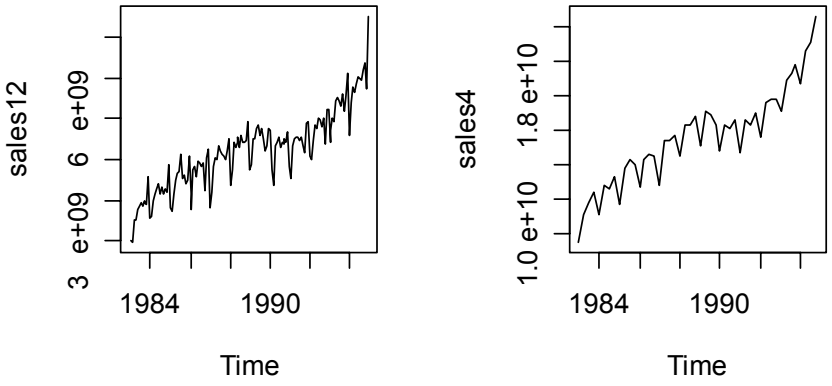
Failo Data\SAS forecasting ts\chpt1.sas 2-me psl. yra pateiktos Šiaurės Karolinos valstijos mažmeninės prekybos mėnesinės apimtys (nuo 1983 m. sausio mėn. iki 1994 m. gruodžio mėn.). Tie patys duomenys yra ir Data\Misc\ch1sales.txt faile:

```
1    3015233240    2    2955239916    3    3515214728    4    3542214765
5    3760171602    6    3861710890    7    3948119154    8    3857400765
.....................................................................
```

Nusiskaitykite iš ten šiuos duomenis (`sales12` yra mėnesiniai duomenys, o `sales4` – agreguoti ketvirtiniai duomenys (su jais bus lengviau dirbti)):

```
sales12=ts(matrix(scan(file.choose()),ncol=2,byrow=T)[,2],freq=12,start=1983.0)
sales4=ts(apply(matrix(sales12,ncol=3,byrow=T),1,sum),freq=4,start=1983.0)
par(mfrow=c(1,2))
plot(sales12)
plot(sales4)
```

Mūsų tikslas – prognozuoti ketvirtinius pardavimus `sales4` nuo 1995 m. 1-ojo ketvirčio iki 1998 m. 4 ketvirčio.



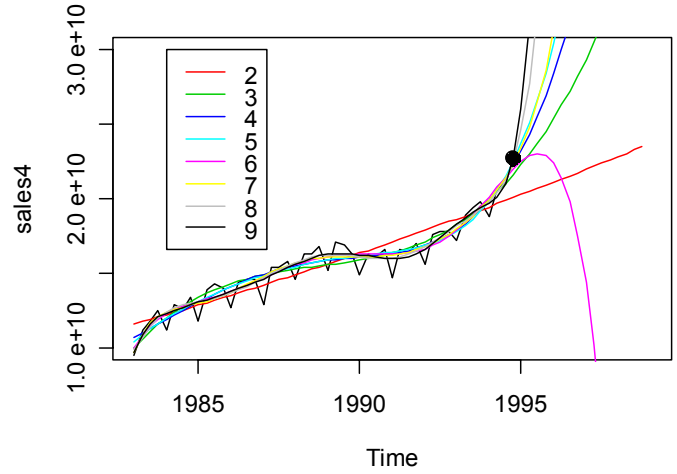
`sales4` seka turi akivaizdų trendą ir sezoninę dedamąją. Trendą išskirsime MK metodu. (Ortogonalaus) polinomo eilę parinksime pagal AIC minimumą:

```
TIM=as.numeric(time(sales4))
for(i in 2:9) cat("i=",i,AIC(lm(sales4~poly(TIM,i))),"\n")

i= 2 2138.748
i= 3 2116.184
i= 4 2112.903
i= 5 2114.1
i= 6 2112.269  # Mažiausia reikšmė
i= 7 2112.788
i= 8 2113.042
i= 9 2113.467
```

Formaliai žiūrint, geriausias yra 6-osios eilės modelis, tačiau tokios aukštos eilės modeliai retai vartojami prognozei (paprastai apsiribojama 3-iosios ar daugiausiai 4-osios eilės polinomais).

```
plot(sales4,xlim=c(1983,1999),ylim=c(1e+10,3.0e+10),ylab="sales4")
d= seq(1983,1998.75,by=0.25)
```

```
  for(degree in 2:9)
  {
     fm <- lm(sales4 ~ poly(TIM, degree))
     assign(paste("s", degree, sep="."), fm)
     lines(d, predict(fm, data.frame(TIM=d)), col = degree)
  }
legend(1984,3e+10,c("2","3","4","5","6","7","8","9"),lty=1,col=2:9)
points(1994.75, window(sales4,start=1994.75,end=1994.75),pch=16,cex=1.5)
anova(s.2,s.3,s.4,s.5,s.6,s.7,s.8,s.8)
```



3.30 pav. Aštuoni modeliai ir jų prognozės nuo 1995 m. 1-ojo ketvirčio. Aišku, kad 6-osios eilės modelis prognozavimui netinka, o 3-iosios ar 4-osios eilės – visai priimtini

```
> anova(s.2,s.3,s.4,s.5,s.6,s.7,s.8,s.9)

  Res.Df        RSS Df  Sum of Sq       F    Pr(>F)
1     45 5.3377e+19
2     44 3.1996e+19  1 2.1381e+19 34.5031 8.478e-07
3     43 2.8663e+19  1 3.3331e+18  5.3788   0.02586
4     42 2.8187e+19  1 4.7577e+17  0.7678   0.38641
5     41 2.6025e+19  1 2.1620e+18  3.4889   0.06951
6     40 2.5234e+19  1 7.9096e+17  1.2764   0.26565
7     39 2.4333e+19  1 9.0119e+17  1.4543   0.23529
8     38 2.3548e+19  1 7.8568e+17  1.2679   0.26722
```

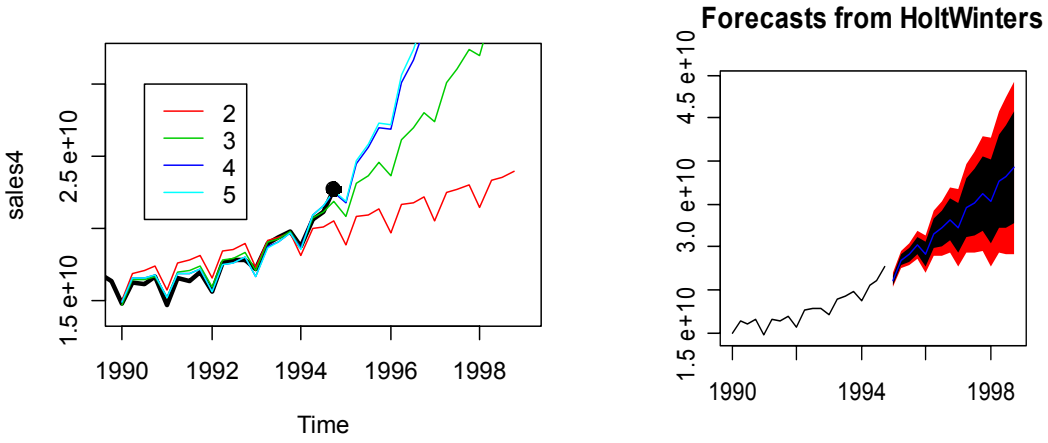
Priminsime, kad `anova` funkcija atlieka modelių palyginimą pagal F kriterijų. Tiksliau kalbant, 1-oje eilutėje yra pateikta 1-os eilės modelio $\texttt{sales4} = b_0 + b_1\texttt{poly(TIM,degree)1} + e$ liekanų kvadratų suma RSS(=RSS1). 2-oje eilutėje yra nagrinėjama hipotezė susijusi su 2-os eilės modeliu $\texttt{sales4} = b_0 + b_1\texttt{poly(TIM,degree)1} + b_2\texttt{poly(TIM,degree)2} + e : H_0 : b_2 = 0$. Ši hipotezė tikrinama pagal santykio $F = \dfrac{(\texttt{RSS1} - \texttt{RSS2})/1}{\texttt{RSS2}/(n-2-1)}$ didumą: jei $P(F_{1,n-3} > F) < 0.05$, H_0 atmetame ir tariame, kad 2-os eilės modelis geriau aprašo `sales4` priklausomybę nuo `TIM`. Iš pateiktos lentelės matyti, kad laipsnį verta kelti tik iki 4-ojo.

Pažymėsime, kad pateikta procedūra pateikia tik gana apytikslį uždavinio sprendimą. Reikalas tas, kad netgi „geriausias“ 4-ios eilės modelis yra blogas – jo paklaidos nebus baltasis triukšmas (jose tikrai bus likusi sezoninė komponentė). Išbandykime naują modelį su sezoniniais nariais.

```
library(lmtest)
plot(sales4,xlim=c(1990,1999), ylim=c(1.4e+10,3.2e+10),ylab="sales4",lwd=3)
d= seq(1990,1998.75,by=0.25)
seas=as.factor(rep(1:4,12)) # Sukuriame sezoninį faktorių
d.seas=as.factor(rep(1:4,9))
  for(degree in 2:5)
  {
     fm <- lm(sales4 ~ poly(TIM, degree)+seas)
     cat("degree=",degree,"AIC=",AIC(fm),"Durbin-Watson=",dwtest(fm)$stat,"\n")
      assign(paste("ss", degree, sep="."), fm)
     lines(d, predict(fm, data.frame(TIM=d,seas=d.seas)), col = degree)
  }
legend(1990.5,3e+10,c("2","3","4","5"),lty=1,col=2:5)
points(1994.75, window(sales4,start=1994.75,end=1994.75),pch=16,cex=1.5)
anova(ss.2,ss.3,ss.4,ss.5)

******************

degree= 2 AIC= 2110.911 Durbin-Watson= 0.2828222
degree= 3 AIC= 2062.872 Durbin-Watson= 0.5836363
degree= 4 AIC= 2041.486 Durbin-Watson= 0.8789683 # Geriausias (4-tos eilės)
degree= 5 AIC= 2043.348 Durbin-Watson= 0.8758298

******************

  Res.Df        RSS Df  Sum of Sq        F    Pr(>F)
1     42 2.6375e+19
2     41 9.2995e+18  1 1.7076e+19 116.9021 2.678e-13
3     40 5.7131e+18  1 3.5865e+18  24.5532 1.443e-05 # Geriausias (4-os eilės)
4     39 5.6967e+18  1 1.6395e+16   0.1122    0.7394

*****************************
> summary(fm4) # 4-os eilės modelio reziumė

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.442e+10  1.096e+08 131.499  < 2e-16 ***
poly(TIM, degree)1 1.703e+10  3.792e+08  44.908  < 2e-16 ***
poly(TIM, degree)2 6.608e+08  3.779e+08   1.748    0.088 .
poly(TIM, degree)3 4.164e+09  3.808e+08  10.934 1.38e-13 ***
poly(TIM, degree)4 1.894e+09  3.781e+08   5.011 1.14e-05 ***
seas2               1.635e+09  1.545e+08  10.584 3.67e-13 *** 2-ojo ketv. priedas
seas3               1.534e+09  1.550e+08   9.899 2.59e-12 *** 3 ketv. priedas
seas4               1.644e+09  1.558e+08  10.552 4.01e-13 *** 4 ketv. priedas
Residual standard error: 377900000 on 40 degrees of freedom
Multiple R-Squared: 0.9839,     Adjusted R-squared: 0.981
F-statistic: 348.5 on 7 and 40 DF,  p-value: < 2.2e-16
```

Apskritai kalbant, geriausias (4-osios eilės) modelis irgi nėra visai priimtinas – jo (liekanų) Durbino ir Watsono statistika lygi 0,8789683, taigi ji toli nuo 2 (iš čia išplaukia, kad paklaidos koreliuotos). Negana to, ši statistika artima 1, todėl gali būti, kad `sales4` yra aprašoma kaip seka su determinuotuoju trendu ir vienetinės šaknies proceso paklaidomis (jei taip, tai gautieji regresijos koeficientų įverčiai yra paslinkti). Kol kas nusprendžiame tik tiek, kad vietoje funkcijos `lm` reikia vartoti `arima`. Prie šio modelio sugrįšime truputį vėliau, o dabar dar aptarsime eksponentinio glodinimo metodą.

```
library(forecast)
plot(forecast(HoltWinters(sales4),h=16),include=20)
```



3.31 pav. `sales4` prognozė su polinominiais modeliais (kairėje) ir su Holto ir Winterso eksponentinio glodinimo modeliu (dešinėje)

Šio populiaraus metodo trūkumas yra tas, kad jame nedaroma jokių statistinių prielaidų apie į jį įeinančius kintamuosius, todėl 3.29 pav. išbrėžti 80% (juodas) ir 95% (raudonas) prognozės pasikliautties intervalai gali būti neteisingi (žr. [MWH, 373 p.]).

Atrodo, kad eksponentinė prognozė yra panaši į 3-iosios eilės modelį (patys palyginkite 3 ir 4 eilės modelius su eksponentiniu). Išbandykite dar ir tokį tam tikra prasme pranašesnį *Exponential smoothing state space* modelį:

```
fit <- ets(sales4)
plot(forecast(fit,h=16),include=20)
```

3.7. Tariamoji regresija

Kartais, nagrinėdami regresijos modelius $y_t = c_0 + c_1 x_t + \varepsilon_t$, stebime tariamąją (kitaip - netikrąją arba fiktyviąją, angl. spurious) regresiją: regresijos koeficientas prie kintamojo x reikšmingas, nors ekonominė logika sako, kad y arba iš vis neturėtų priklausyti nuo x, arba (reikšmingo) koeficiento ženklas yra „ne tas“. Tariamoji regresija gali atsirasti dėl kelių priežasčių:

- jei y, x ir regresinio modelio $y_t = c_0 + c_1 x_t + \varepsilon_t$ liekanos $\hat{\varepsilon}_t$ yra integruoti procesai $I(1)$, regresija tarp y ir x gali būti tik tariamoji (kasdienė IBM akcijų kaina nuo 1961v17 iki 1962xi02 (iš viso 369 dienos) ir kasdienis 369 dienų Vokietijos akcijų DAX indeksas nuo 1991 m. 130-osios dienos, matyt, niekaip nėra susiję, tačiau koeficientas c_1 yra „reikšmingas“; žr. 3.7 pavyzdį);
- jei y ir x yra TS procesai su tiesiniu trendu, tariamoji priklausomybė tarp y ir x gali atsirasti ne dėl jų tarpusavio ryšio, o dėl trendo (Žemės temperatūra ir Lietuvos žmonių atlyginimai didėja (DATA\misc), tačiau, kažin, ar tai susiję dydžiai; žr. xxx pavyzdį);
- koeficiento $\hat{c}_1$ ženklas gali būti neteisingas, jei pažeistos kai kurios klasikinio regresinio modelio sąlygos (pvz., jei x_t koreliuoja su ε_t arba jei prognoziniai kintamieji multikolinearūs; žr. xxx pavyzdi).

Aptarsime aukščiau išvardintas galimybes. Vienetinių šaknų klausimas natūraliai iškyla regresijos uždaviniuose. Nagrinėkime regresijos lygtį

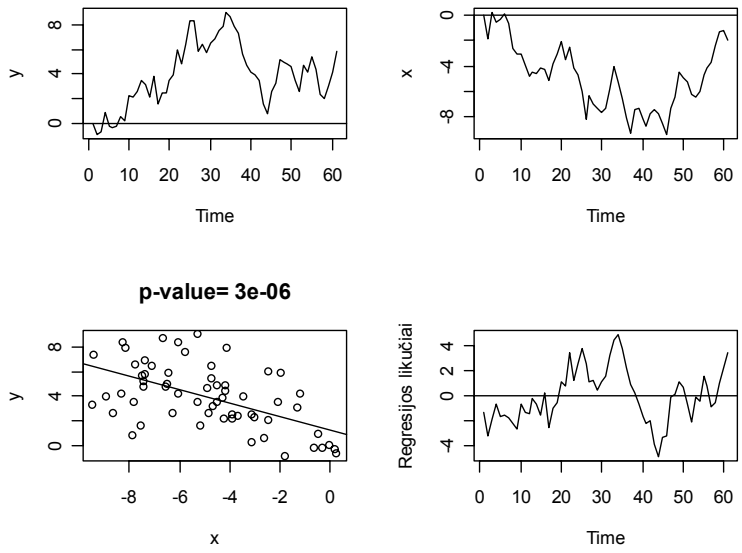
$$y_t = c_0 + c_1 x_t + e_t . \qquad (*)$$

Taikant klasikinę regresiją, yra daroma prielaida, kad abi sekos y_t ir x_t yra stacionarios, o paklaidos turi nulinį vidurkį ir baigtinę dispersiją. Antra vertus, jei lygties kintamieji yra integruoti, kartais (bet dažniau nei galėtume tikėtis) stebime vadinamąją tariamąją regresiją. Ši regresija turi didelį R^2, t statistikos atrodo reikšmingos, bet rezultatai neturi jokios ekonominės prasmės. Stebimasis reiškinys atsiranda dėl to, kad MK statistikos dabar nėra suderintos, o t statistika turi kitokį nei Student'o skirstinį.

Išnagrinėkime pamokantį pavyzdį. Generuokime dvi nepriklausomas atsitiktinių klaidžiojimų sekas $y_t = y_{t-1} + w_{yt}$ ir $x_t = x_{t-1} + w_{xt}$; čia w_{yt} ir w_{xt} yra du nepriklausomi baltieji triukšmai. Aišku, kad (*) lygtis yra beprasmė (kitaip sakant, c_1 turėtų būti lygus 0 - juk y_t ir x_t jokio ryšio neturi!), tačiau, atlikus 1000 Monte-Carlo bandymų, nesunku įsitikinti, kad hipotezė $H_0 : c_1 = 0$ bus atmetama, tarkime, 5% reikšmingumu, žymiai dažniau negu 5 kartus iš 100.

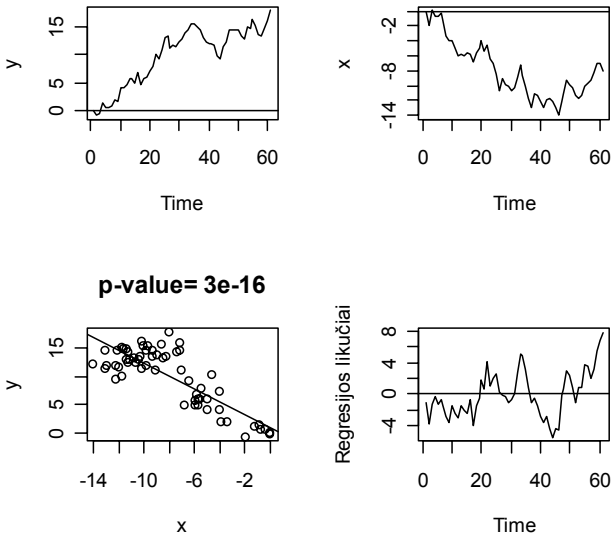
```
ilgis=50        # Atsitiktinio klaidžiojimo ilgis
kartai=1000     # Monte-Carlo bandymų kartų skaičius
set.seed(1)
p.value=numeric(kartai)    # Čia talpinsime hipotezės H0 p reikšmes
for(i in 1:kartai)         # Ciklas
{
y=ts(diffinv(rnorm(ilgis)))     # Generuojame ats. klaidžiojimą y
x=ts(diffinv(rnorm(ilgis)))     # Generuojame ats. klaidžiojimą x
p.value[i]=summary(lm(y~x))$coef[2,4]  # Hipotezės H0 p reikšmė
}
print(sum(ifelse(p.value<0.05,1,0))/kartai) # Regresijų, kai p<0,05, dalis

[1] 0.67   # 67 kartus iš 100 H0 atmesime
```

Vienas šio modeliavimo bandymų pavaizduotas 3.30 paveiksle. Ten `y` ir `x` yra dvi nepriklausomų atsitiktinių klaidžiojimų trajektorijos. Aišku, kad ryšio tarp šių kintamųjų nėra, tačiau jų sklaidos diagramoje išbrėžtos regresijos tiesės krypties koeficientas „didžiai reikšmingai" (p reikšmė lygi 0,000003) skiriasi nuo nulio. Ši tarytum reikšminga regresija ir vadinama tariamąja regresija. Derėtų atkreipti dėmesį ir į likučių grafiką – jis labiau panašus į atsitiktinio klaidžiojimo, o ne į stacionaraus proceso grafiką.



3.32 pav. Atsitiktinių klaidžiojimų `y` ir `x` grafikai (viršutinė eilutė), `x` ir `y` sklaidos diagrama su regresijos tiese (apačioje, kairėje) ir regresijos likučių grafikas (apačioje, dešinėje)

Dar įspūdingesnę „regresiją“ stebėtume, jei į modelį įtrauktume dreifą[22] (tam aukščiau pateiktoje programoje `diffinv(rnorm(ilgis)` pakeiskite į atitinkamai `diffinv(0.2+rnorm( ilgis)` ir `diffinv(-0,1+ rnorm(ilgis)` – dabar H_0 atmestume 76 kartus iš 100. Vienas šio modeliavimo bandymų pavaizduotas 3.31 paveiksle.



3.33 pav. Dviejų *nepriklausomų* klaidžiojimų su dreifu tariamoji regresija

Stebimo reiškinio esmė yra ta, kad lygtyje (*) tarę, jog $c_1 = 0$, gautume $y_t = c_0 + e_t$. Kadangi y yra integruotasis I(1) procesas, todėl ir paklaidos e turėtų būti tokios pat. Tačiau tai prieštarauja klasiki-

[22] Dauguma ekonominių reiškinių turi stochastinį trendą su dreifu.

nės MK regresijos sąlygoms (paklaidos turi sudaryti baltąjį triukšmą), todėl t ir F testai bei R^2 reikšmės yra nepatikimos. Galima įrodyti, kad, didinant imties dydį, problemos nedingsta – atvirkščiai, kuo didesnė imtis, tuo mažiau šansų priimti hipotezę $c_1 = 0$.

Kiek detaliau aptarsime tas rizikingas situacijas, kurios gali atsirasti, nagrinėjant (*) lygtį.

1 atvejis. Abi laikinės sekos, y_t ir x_t, yra stacionarios. Šiuo atveju klasikinis MK metodas yra visai priimtinas.

2 atvejis. Abi laikinės sekos yra integruotos, tačiau skirtingomis eilėmis. Šiuo atveju regresija yra beprasmė. Pavyzdžiui, jei $x_t = ax_{t-1} + w_{xt}, |a| < 1$, tai[23] tarę, kad $y_0 = x_0 = 0$, $y_t = y_{t-1} + w_{yt}$ gautume $e_t = \sum_{i=1}^{t} w_{yi} - c_1 \sum_{i=1}^{t} a^{t-i} w_{xi}$, taigi seka e_t turi stochastinį trendą (minus „beveik stacionarus“ priedas) ir todėl nestacionari. Panašiai regresinės analizės negalima taikyti ir tuomet, kai vienas kintamasis sudaro DS, o kitas –TS procesą[24]. Pvz., iš akcijų kurso (dažniausiai tai I(1) procesas) nedera išskirti polinominį trendą (tokios regresijos likučiai sudarys integruotą seką – blogai).

3 atvejis. Abi laikinės sekos yra integruotos ta pačia eile, o likučių seka turi stochastinį trendą. Šiuo atveju regresija yra tariamoji. Pvz., jei tartume, kad abi sekos aprašomos atsitiktiniu klaidžiojimu ir $c_0 = 0$, tai $e_t = \sum_{i=1}^{t} w_{yi} - c_1 \sum_{i=1}^{t} w_{xi}$, taigi $E(e_{t+i} \mid I_{\le t}) = e_t$ koks bebūtų $i \ge 1$, kitaip sakant, praeities įtaka nesilpnėja – toks ekonomikos modelis sunkiai įsivaizduojamas. Šiuo atveju dažnai rekomenduojama pereiti prie skirtumų lygties: $\Delta y_t = c_1 \Delta x_t + \Delta e_t$. Kadangi visos trys sekos, y_t, x_t ir e_t, turi vienetines šaknis, pirmieji skirtumai bus stacionarūs – grįžome prie 1-ojo atvejo. Žinoma, jei vienas trendas yra stochastinis, o kitas – determinuotasis, diferencijavimas nepagelbės (2-asis atvejis).

4 atvejis. Jei nestacionarios sekos yra integruotos ta pačia eile ir jų regresijos modelio paklaidų seka stacionari, tai sakome, kad tos sekos yra kointegruotos. Štai paprastas pavyzdys:

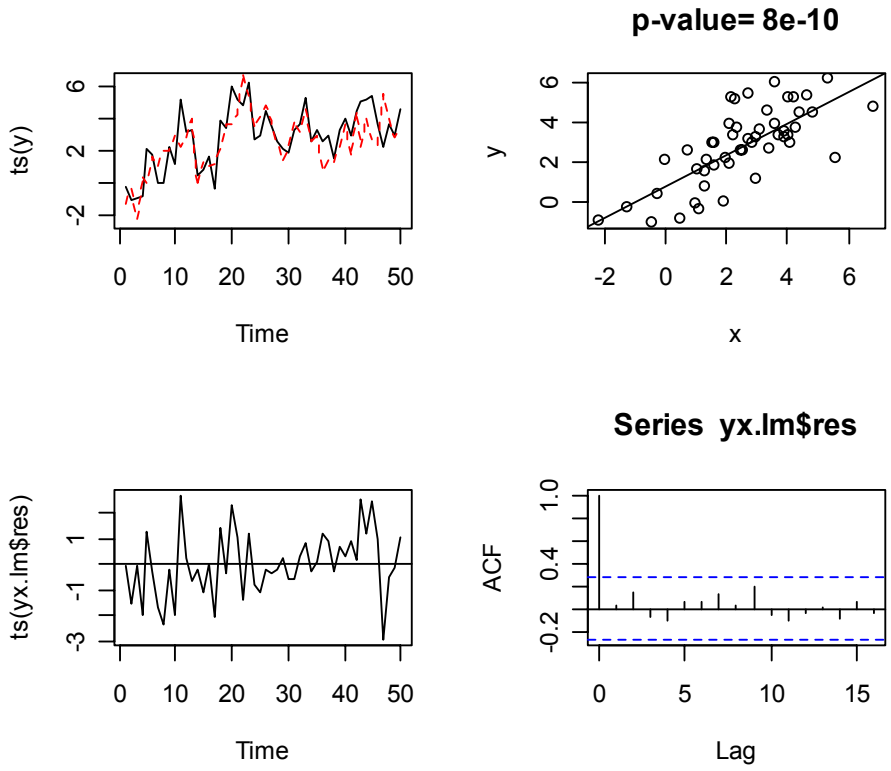
$$\begin{aligned} y_t &= \mu_t + w_{yt} \\ x_t &= \mu_t + w_{xt} \end{aligned},$$

čia μ_t yra abiem sekoms bendras atsitiktinis klaidžiojimas $\mu_t = \mu_{t-1} + w_t$, o w_{yt}, w_{xt} ir w_t yra trys tarpusavyje nepriklausomi baltieji triukšmai[25]. Šį kartą (*) modelio likučiai yra stacionarus procesas (žr. 3.32 pav, apačioje),

[23] **Užduotis**. y_t yra I(1) procesas, o x_t yra I(?) procesas.

[24] Priminsime, kad DS reiškia Difference Stationary, o TS – Trend Stationary.

[25] Jei atsitiktinį klaidžiojimą galimą įsivaizduoti kaip išgėrusio vyro kelią iš baro namo, tai šie du procesai vaizduoja šio žmogaus ir jo ištikimo šuns trajektorijas.



3.34 pav. Laikinių sekų y ir x grafikai bei jų sklaidos diagrama (viršuje); sprendžiant pagal (*) modelio likučių ir ACF grafikus (apačioje), likučiai sudaro baltąjį triukšmą

taigi y_t ir x_t yra kointegruoti procesai. Apie šiuos procesus daugiau kalbėsime ??? skyriuje.

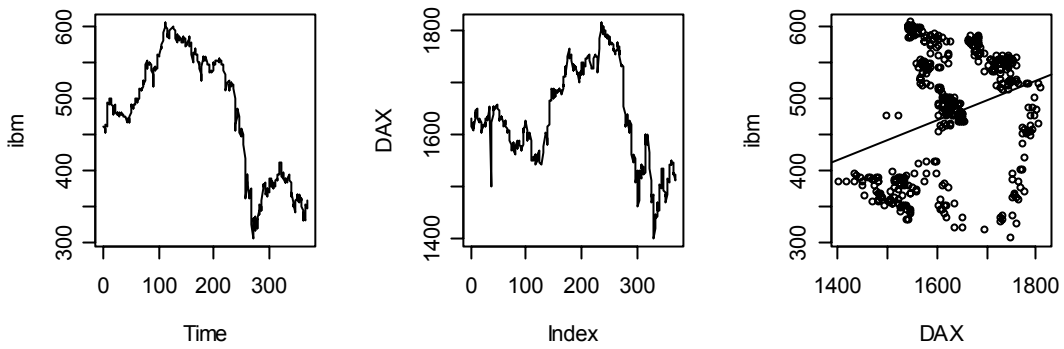
3.8 pavyzdys. Panagrinėkime kasdienių IBM akcijų kainą nuo 1961v17 iki 1962xi02 (iš viso 369 dienos) ir kasdienį 369-ių dienų Vokietijos akcijų DAX indeksą, pradedant 1991 m. 130-ąja diena.

```
library(waveslim); ?ibm
library(datasets); ?EuStockMarkets
DAX=EuStockMarkets[1:369,1]
par(mfrow=c(1,3))
plot(ibm); plot(DAX,type="l")
iD=lm(ibm~DAX)
plot(DAX,ibm); abline(iD)
summary(iD)
[...]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 31.99302   74.03495   0.432    0.666
DAX          0.27347    0.04527   6.040 3.78e-09 ***
```

Nors pagal duomenų prigimtį `ibm` ir `DAX` jokio ryšio turėti negali, regresijos koeficientas labai reikšmingas. Tai tariamosios regresijos pavyzdys, ją paaiškinti galime tuo, kad modelio liekanos

```
library(tseries)
adf.test(iD$res)
[...]
Dickey-Fuller = -1.9507, Lag order = 7, p-value = 0.5978
```

turi vienetinę šaknį.



3.35 pav. Laikinės sekos `ibm` ir `DAX` tikriausiai turi vienetines šaknis (kaip tai galima patikrinti?); kadangi `iD$res` taip pat turi vienetinę šaknį, regresija gali būti tik tariamoji

Norint įsitikinti tuo, kad regresija tik tariama (kitais kartais tai nėra taip akivaizdu), sudarysime modelių skirtumams:

```
> summary(lm(diff(ibm)~diff(DAX)))
[...]
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.278518   0.378917  -0.735    0.463
diff(DAX)    0.004781   0.023672   0.202    0.840
```

Dabar jokio ryšio tarp `ibm` ir `DAX` nėra, taigi anksčiau stebėta „priklausomybė“ buvo netikra.

Kaip nustatoma tariamoji regresija?

Brėžiant likučių korelogramas ir tikrinant likučių vienetinės šaknies hipotezę. Jei likučiai turi vienetinę šaknį, regresija gali būti tik tariamoji.

Kiti būdingi tariamosios regresijos simptomai:

- „Didelis“ determinacijos koeficientas R^2
- „Reikšmingos“ t- ir F- statistikų reikšmės
- Maža modelio likučių Durbin‘o ir Watson‘o statistikos DW reikšmė[26] (tiksliau kalbant, $R^2 >$DW)

Pateiksime kelis tariamos regresijos pavyzdžius (žr. halweb.uc3m.es/esp/Personal/personas/ jgonzalo/teaching/timeseriesMA/examplesspuriousregression.pdf).

1. Egipto kūdikių mirtingumo normos (Y, 1971-1990) regresija JAV visuminių pajamų prieš atskaitant mokesčius (angl. gross aggregate income) (I) ir Honduro pinigų kiekio (M) atžvilgiu[27]:

[26] Priminsime: modelio likučiai yra nekoreliuoti, jei DW≈2.
[27] Po koeficientais užrašytos t statistikų reikšmės.

$$\hat{Y} = 179.9 - 0.2952I - 0.0439M \qquad R^2 = 0.918,\ DW = 0.4752,\ F = 95.17$$
$$(16.63)\ \ (-2.32)\ \ (-4.26) \qquad cor(Y,I) = -0.9113,\ cor(Y,M) = -0.9445$$

2. JAV eksporto indekso (Y, 1960-1990 metiniai duomenys) regresija Australijos vyrų prognozuojamos gyvenimo trukmės (X) atžvilgiu.

$$\hat{Y} = -2943. + 45.7974X \qquad R^2 = 0.916,\ DW = 0.3599,\ F = 315.2$$
$$(16.70)\ \ (17.76)$$

3. Pietų Afrikos Respublikos gyventojų skaičiaus (Y, 1971-1990 metiniai duomenys) regresija visų JAV išlaidų mokslui (X) atžvilgiu.

$$\hat{Y} = 21698.7 + 111.58X \qquad R^2 = 0.974,\ DW = 0.3037,\ F = 696.96$$
$$(59.44)\ \ (26.40)$$

◀

3.24 UŽDUOTIS. aaaaaaaaaaabbbbbbbbbbbccccccccccddddddddddd

Pažymėsime, kad tariamoji regresija gali atsirasti ne tik stochastinio, bet ir determinuotojo trendo atveju. Tai būna tuomet, kai X ir Y tiesiogiai nėra susiję, bet jų tariamas ryšys atsiranda dėl trečio kintamojo įtakos jiems abiems[28] (tas trečias kintamasis dažnai būna tiesiog „progresas“, kurį galima sutapatinti su laiku).

3.9 pavyzdys. Duomenų rinkinyje HSEINV.RAW yra pateikti 1947-1988 metų JAV duomenys apie namų statybą.

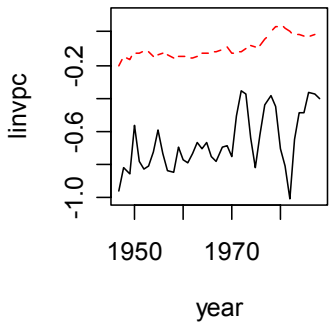
```
 1. year                    1947-1988
 2. inv                     real housing invest., millions $
 3. pop                     population, 1000s
 4. price                   housing price index; 1982 = 1
 5. linv                    log(inv)
 6. lpop                    log(pop)
 7. lprice                  log(price)
 8. t                       time trend: t=1,...,42
 9. invpc                   per capita invest., inv/pop
10. linvpc                  log(invpc)
11. lprice_1                lprice[t-1]
12. linvpc_1                linvpc[t-1]
13. gprice                  lprice - lprice_1
14. ginvpc                 linvpc - linvpc_1
```

```
year=hseinv[,1]
tt=hseinv[,8]
linvpc=hseinv[,10]
lprice=hseinv[,7]
hs1=lm(linvpc~lprice)
summary(hs1)
```

[28] Tarkime, kad $y_t = c_{y1} + c_{y2}t + w_{yt}$, o $x_t = c_{x1} + c_{x2}t + w_{xt}$; jei $c_{y2} = c_{x2} = 0$, tai regresijos $y_t = c_{yx,1} + c_{yx,2}x_t + e_t$ koeficientas $c_{yx,2}$ paprastai bus nereikšmingas, tačiau rezultatas bus priešingas, jei abu procesai turės tiesinį trendą.

```
[...]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.55023    0.04303 -12.788 1.03e-15 ***
lprice       1.24094    0.38242   3.245  0.00238 **

Residual standard error: 0.1554 on 40 degrees of freedom
Multiple R-Squared: 0.2084,     Adjusted R-squared: 0.1886
F-statistic: 10.53 on 1 and 40 DF,  p-value: 0.002376
```

Šioje (pastovaus elastingumo) lygtyje (ją galima interpretuoti kaip namų pasiūlos lygtį) `lprice` koeficientas yra reikšmingas ir statistiškai nesiskiria nuo vieno. Antra vertus, abu kintamieji (`linvpc` (juoda linija, pav. dešinėje) ir `lprice` (raudona trūki linija)) turi kylantį trendą, todėl ši (gal būt, tariama) priklausomybė gali atsirasti tik jo. Norėdami išanalizuoti šią galimybę, į modelį įtrauksime ir laiką `t`:




```
> hs2=lm(linvpc~lprice+tt)
> summary(hs2)

[...]
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.913059   0.135613  -6.733    5e-08 ***
lprice      -0.380961   0.678835  -0.561  0.57787
tt           0.009829   0.003512   2.798  0.00794 **

Residual standard error: 0.1436 on 39 degrees of freedom
Multiple R-Squared: 0.3408,     Adjusted R-squared: 0.307
F-statistic: 10.08 on 2 and 39 DF,  p-value: 0.000296
```

Dabar vaizdas visai kitas – koeficientas prie `tt` garantuoja maždaug 1% metinį investicijų (vienam gyventojui) prieaugį , o `lprice` koeficientas nereikšmingas. Taigi investicijų didėjimo priežastis yra „bendra ekonomikos pažanga“, bet ne kaina.

Pažymėsime, kad tokį patį rezultatą (koeficientą prie `lprice`) gautume, jei regresijos lygtį sudarytume nuokrypiams nuo trendo $\texttt{linvpc.t} = \texttt{linvpc} - \hat{a}_1 - \hat{b}_1 t$ ir $\texttt{lprice.t} = \texttt{lprice} - \hat{a}_2 - \hat{b}_2 t$:

```
linvpc.lm=lm(linvpc~tt)
linvpc.t=linvpc.lm$res
lprice.lm=lm(lprice~tt)
lprice.t=lprice.lm$res
summary(lm(linvpc.t~lprice.t))

[…]
Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  2.184e-18  2.189e-02 9.98e-17    1.000
lprice.t    -3.810e-01  6.703e-01   -0.568    0.573

Residual standard error: 0.1418 on 40 degrees of freedom
Multiple R-Squared: 0.008011,   Adjusted R-squared: -0.01679
F-statistic: 0.323 on 1 and 40 DF,  p-value: 0.573
```

Šios lygties išvada yra ta pati – `linvpc` nuo `lprice` nepriklauso.

3.25 UŽDUOTIS.

EXAMPLE 10.3
(Puerto Rican Employment and the Minimum Wage)

Annual data on the Puerto Rican employment rate, minimum wage, and other variables are used by Castillo-Freedman and Freedman (1992) to study the effects of the U.S. minimum wage on employment in Puerto Rico. A simplified version of their model is

$$\log(prepop_t) = \beta_0 + \beta_1 \log(mincov_t) + \beta_2 \log(usgnp_t) + u_t, \qquad \textbf{(10.16)}$$

where $prepop_t$ is the employment rate in Puerto Rico during year t (ratio of those working to total population), $usgnp_t$ is real U.S. gross national product (in billions of dollars), and *mincov* measures the importance of the minimum wage relative to average wages. In particular, *mincov* = (*avgmin*/*avgwage*)·*avgcov*, where *avgmin* is the average minimum wage, *avgwage* is the average overall wage, and *avgcov* is the average coverage rate (the proportion of workers actually covered by the minimum wage law).

Using data for the years 1950 through 1987 gives

$$\widehat{\log(prepop_t)} = \underset{(0.77)}{-1.05} - \underset{(.065)}{.154}\log(mincov_t) - \underset{(.089)}{.012}\log(usgnp_t)$$
$$n = 38, R^2 = .661, \bar{R}^2 = .641. \qquad \textbf{(10.17)}$$

The estimated elasticity of *prepop* with respect to *mincov* is −.154, and it is statistically significant with $t = -2.37$. Therefore, a higher minimum wage lowers the employment rate, something that classical economics predicts. The GNP variable is not statistically significant, but this changes when we account for a time trend in the next section.

EXAMPLE 10.9
(Puerto Rican Employment)

When we add a linear trend to equation (10.17), the estimates are

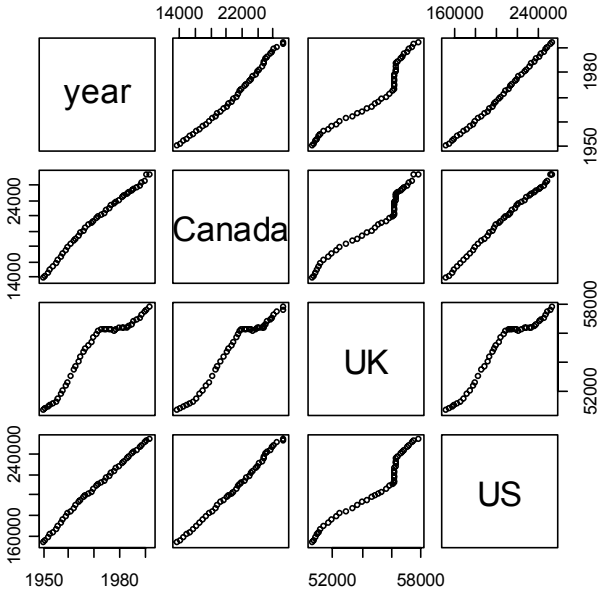
$$\widehat{\log(prepop_t)} = \underset{(1.30)}{-8.70} - \underset{(.044)}{.169}\log(mincov_t) + \underset{(0.18)}{1.06}\log(usgnp_t) - \underset{(.005)}{.032}\,t \qquad (10.38)$$
$$n = 38,\ R^2 = .847,\ \bar{R}^2 = .834.$$

The coefficient on log(*usgnp*) has changed dramatically: from −.012 and insignificant to 1.06 and very significant. The coefficient on the minimum wage has changed only slightly, although the standard error is notably smaller, making log(*mincov*) more significant than before.

The variable $prepop_t$ displays no clear upward or downward trend, but log(*usgnp*) has an upward, linear trend. (A regression of log(*usgnp*) on *t* gives an estimate of about .03, so that *usgnp* is growing by about 3% per year over the period.) We can think of the estimate 1.06 as follows: when *usgnp* increases by 1% *above* its long-run trend, *prepop* increases by about 1.06%.

3.10 pavyzdys. Surinkite `pop=read.table(file.choose(),header=TRUE)` ir nuvairuokite į Data\Stewart\ASCII\pop.dat. Šioje lentelėje yra pateikti trijų šalių gyventojų skaičiai.

```
> pop
   year Canada    UK     US
1  1950  13737 50614 152273
.........................
43 1992  27445 57848 255000

> pairs(pop)
```



3.36 pav. Trijų šalių gyventojų skaičiaus dinamika ir sklaidos diagramos

Kairiajame stulpelyje išbrėžti populiacijų grafikai, o kituose langeliuose – sklaidos diagramos. Akivaizdu, kad visų keturių kintamųjų koreliacijos koeficientai bus dideli

```
> cor(pop)
        year Canada    UK    US
year   1.000  0.995 0.956 0.999
Canada 0.995  1.000 0.977 0.998
UK     0.956  0.977 1.000 0.966
US     0.999  0.998 0.966 1.000
```

Regresijos lygčių koeficientai irgi bus reikšmingi, tačiau tai visai nereiškia, kad JAV gyventojų skaičių nusako Jungtinės karalystės ar Kanados gyventojų skaičius[29].

3.26 UŽDUOTIS. Nėra visai aišku ar tiesinių modelių *Šalis1=c(1)+c(2)Šalis2+paklaida* paklaidos yra aprašomos baltuoju triukšmu, stacionariu ARMA ar I(1) procesu. Pasirinkite dvi kurias nors šalis ir ištirkite jų modelį su `lm` ir `arima` funkcijomis. Ar reikšmingas *c(2)*? Ar turi likučiai vienetinę šaknį? Taigi *c(2)* reikšmingas iš tikrųjų ar tariamai?

3.11 pavyzdys. Importuokime duomenų rinkinį `meap93.raw` iš ...\DATA\Wooldridge_2ed. Šiame rinkinyje yra pateikti 1993 metais atliktų 408 Michigano valstijos vidurinių mokyklų tyrimo rezultatai. Stulpeliuose patalpinti tokie duomenys:

```
1. lnchprg                perc. of students in school lunch program
2. enroll                 school enrollment
3. staff                  staff per 1000 students (dėmesio kiekis vienam
                          mokiniui)
4. expend                 expend. per stud., $
```

[29] Populiacijos auga dėl „progreso“, kurį galima aproksimuoti visoms šalis bendru prognoziniu kintamuoju – laiku.

```
  5. salary                 avg. teacher salary, $
  6. benefits               avg. teacher benefits, $
  7. droprate               school dropout rate, perc
  8. gradrate               school graduation rate, perc
  9. math10                 perc studs passing MEAP math
 10. sci11                  perc studs passing MEAP science
 11. totcomp                annual teacher compensation (mokytojų kokybės
                            matas)
 12. ltotcomp               log(totcomp)
 13. lexpend                log of expend
 14. lenroll                log(enroll)
 15. lstaff                 log(staff)
 16. bensal                 benefits/salary
 17. lsalary                log(salary)
```

Čia `math10` yra procentas dešimtokų, gavusių teigimą pažymį per mokyklinį matematikos egzaminą. Be šio dydžio mums dar rūpi federalinės vyriausybės fondai, iš kurių finansuojami mokykliniai pietūs mokiniams iš skurdžių šeimų – dydis `lnchprg` žymi, koks mokinių procentas yra maitinamas nemokamai. Aišku, kad sotus mokinys turėtų mokytis geriau:

```
meap93=read.table(file.choose())
lnchprg=meap93[,1]
math10=meap93[,9]
summary(lm(math10~lnchprg))

[...]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 32.14271    0.99758  32.221   <2e-16 ***
lnchprg     -0.31886    0.03484  -9.152   <2e-16 ***

Residual standard error: 9.566 on 406 degrees of freedom
Multiple R-Squared: 0.171,      Adjusted R-squared: 0.169
F-statistic: 83.77 on 1 and 406 DF,  p-value: < 2.2e-16
```

Gauname nelauktą rezultatą – labiau remiamos mokyklos mokinių pažangumas yra žemesnis (koeficientas prie `lnchprg` yra neigiamas ir reikšmingas). Šį netikėtą rezultatą reikėtų aiškinti tuo, kad paklaidos narys, tikriausiai, koreliuotas su `lnchprg` ir todėl koeficiento įvertis paslinktas. Iš tikrųjų, į paklaidą įeina mokinių skurdo lygis, mokytojų kiekis, mokyklos aprūpinimas, lygis ir pan., kas koreliuoja su `lnchprg`.

3.27 UŽDUOTIS. Įvertinkite minėtų dydžių koreliaciją. Įtraukite daugiau kintamųjų į modelį ir patikslinkite jį.

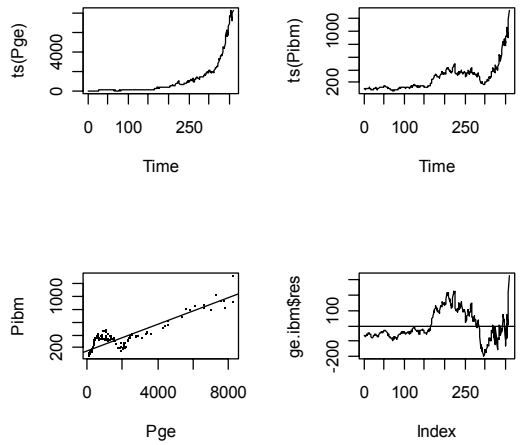
3.28 UŽDUOTIS. Laikraštis straipsnyje „TV Dooms Kids to Gloom“ pranešė apie vieno tyrimo rezultatus.

Lygindami televizijos ir depresijos plitimą „mes nustatėme, kad televizijos aparatų apskrityje skaičius beveik tiksliai atitinka vaikų sergančių depresija skaičių“.

Savais žodžiais paaiškinkite, kodėl nustatytas faktas neįrodo, kad televizija iššaukia depresiją. ◄

3.29 UŽDUOTIS. Iš paketo `Ecdat` pasiimkite `CRSPmon` duomenis – tai daugiamatė laikinė seka, kurios pirmoji komponentė yra General Electric, o antroji – IBM akcijų mėnesinės grąžos nuo 1969 m. sausio mėn. iki 1998 m. gruodžio mėn. Būtų visai neįtikėtina, jei IBM kompanijos politika ar sėkmė galėtų daryti įtaką GE akcijų grąžai, todėl regresija, jei ji ir būtų „reikšminga“, tai tik tariamai. Grąžos bus nagrinėjamos 4 skyriuje, todėl čia pateiksime reikalingas formules: jei akcijos kaina (kursas) laiko momentu t lygi P_t, tai jos grąža vadiname skaičių $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ (taigi $P_t = P_{t-1}(1+R_{t-1})$). Tarę, kad abiejų akcijų kaina (indeksas) 1969 m. sausio mėn. 1 d. buvo[30] 100, pagal paskutinę formulę grąžas paverskite akcijų kaina, atlikite jų regresinę analizę, su `help.search` susiraskite Durbin'o ir Watson'o testą ir išbrėžkite paveikslą panašų į 3.35 pav.

```
> print(summary(ge.ibm))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.551e+02  5.900e+00   26.29   <2e-16 ***
Pge         1.022e-01  2.895e-03   35.30   <2e-16 ***

Residual standard error: 93.44 on 358 degrees of freedom
Multiple R-Squared: 0.7768,     Adjusted R-squared: 0.7762
F-statistic:  1246 on 1 and 358 DF,  p-value: < 2.2e-16

> dwtest(ge.ibm)
        Durbin-Watson test

data:  ge.ibm
DW = 0.0691, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```



3.37 pav. GE ir IBM akcijų kursai (viršuje), jų sklaidos diagrama ir regresinio modelio likučių grafikas (apačioje)

◀

3.30 UŽDUOTIS. Internete susiraskite[31] dviejų skirtingų pramonės šakų Lietuvos firmų akcijų kurso duomenis ir ištirkite jų tarpusavio regresinį ryšį.

[30] Koeficientų *c(1)* ir *c(2)* reikšmės priklauso nuo mastelio, bet *c(2)* reikšmingumas ir R^2 – ne.
[31] Žr., pvz., http://market.lt.omxgroup.com/?market=XVSE&pg=mainlist

3.12 pavyzdys. Ar priklauso savaitinis santuokų skaičius Vilniuje s_t nuo savaitinio kritulių kiekio Vilniuje k_t? Vargu. Antra vertus, suminis santuokų skaičius $S_t = \sum_{i=1}^{t} s_i$ ir suminis kritulių kiekis $K_t = \sum_{i=1}^{t} k_i$, $t = 1,...,104$, yra (dėl bendro augančio trendo) smarkai koreliuoti (žr. 3.36 pav.).

Čia bus grafikas

3.38 pav.

Ryšys tarp S_t ir K_t vadinamas ilgalaikiu, o ryšys tarp jų skirtumų $s_t = \Delta S_t$ ir $k_t = \Delta K_t$ - trumpalaikiu[32]. Taigi šį kartą ilgalaikis ryšys yra tariamas (nors formaliais statistiniais metodais to nustatyti neįmanoma), o trumpalaikio ryšio iš vis nėra.

Surinkti duomenis, išbrėžti grafikus, sudaryti regresinius modelius

3.8. SARIMA (=Seasonal ARIMA) modeliai

Daugelis ekonominių procesų turi sezoninę komponentę. Įprastinė sezoninio veiksnio šalinimo procedūra pirmiausiai pašalina (trendą ir, po to,) sezoniškumo efektą, o paskui nagrinėja likusią proceso dalį (pvz., identifikuoja ją kaip stacionarų ARMA procesą). Antra vertus, dažnai gaunami geresni rezultatai, jei abi procedūros atliekamos vienu metu.

Štai du ketvirtinio sezoniškumo modeliai: $y_t = a_4 y_{t-4} + w_t$, $|a_4| < 1$ ir $y_t = w_t + b_4 w_{t-4}$. Nesunku įsitikinti, kad pirmuoju atveju $\rho_i = a_4^{i/4}$, jei i/4 yra sveikas skaičius, ir =0 kitais atvejais. Antruoju atveju, ACF turi vienintelį nenulinį stulpelį taške 4. Deja, stebimi procesai paprastai turi ir nesezoninę dalį, todėl ACF elgesys yra sudėtingesnis (be to, neužmirškime, kad empirinė ACF gali pastebimai skirtis nuo teorinės).

3.6 skyrelį baigėme modeliu $(\Delta Y_t =)\ y_t = a_1 y_{t-1} + w_t + b_1 w_{t-1} + b_4 w_{t-4}$. Iš principo, sezoninį efektą turėtų aprašyti ir toks modelis: $y_t = a_1 y_{t-1} + a_4 y_{t-4} + w_t + b_1 w_{t-1}$. Abu šie modeliai prideda sezoninį narį (w_{t-4} arba, atitinkamai, y_{t-4}), todėl jie vadinami *adityviaisiais*. Pateiksime du *multiplikatyviųjų* modelių variantus[33]:

$$(1 - a_1 L) y_t = (1 + b_1 L) \times (1 + b_4 L^4) w_t \quad \text{(žymėsime SARIMA(1,0,1)(0,0,1)}_4\text{)}$$

(t.y., $y_t = a_1 y_{t-1} + w_t + b_1 w_{t-1} + b_4 w_{t-4} + b_1 b_4 w_{t-5}$)

ir

$$(1 - a_1 L) \times (1 - a_4 L^4) y_t = (1 + b_1 L) w_t \quad \text{(žymėsime SARIMA(1,0,1)(1,0,0)}_4\text{).}$$

(t.y., ... – užrašykite). Dabar pademonstruosime jų taikymo galimybes.

---

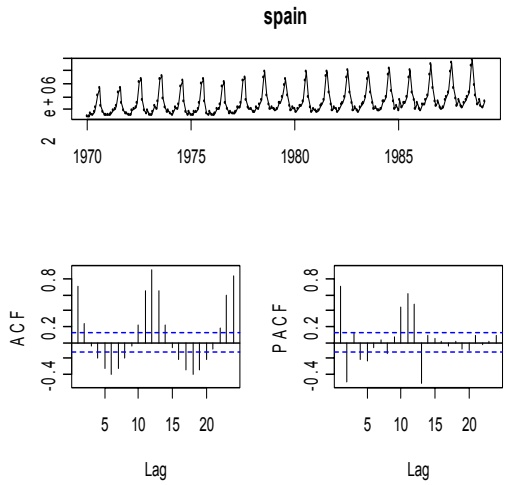
[32] Jei duomenys turi determinuotąjį trendą, trumpalaikiais vadinami ryšiai tarp modelių paklaidų.

[33] Procesas $(1-L)^d (1-L^s)^D \phi(L)\Phi(L^s) Y_t = \theta(L)\Theta(L^s) w_t$ žymimas $ARIMA(p,d,q)(P,D,Q)_s$ arba $SARIMA(p,d,q)(P,D,Q)_s$.

Surinkę

```
spain=ts(scan(file.choose()),start=1970,freq=12) # Nuvairuokite į
                                                 # Data\Enders\Spain.txt
tsdisplay(spain)
```

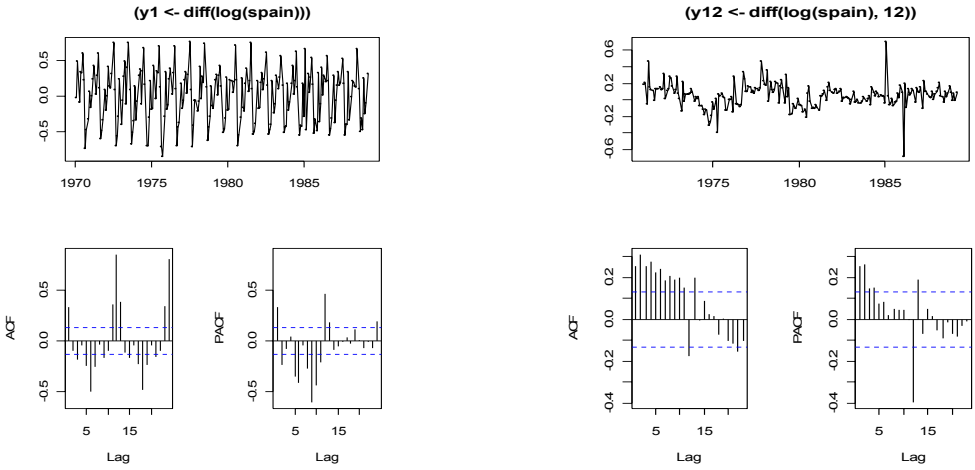
pamatysite mėnesinius Ispaniją aplankiusių turistų skaičius. Sekos `spain` vidurkis lėtai auga, o duomenys turi akivaizdų sezoniškumą.



3.39 pav. Ispaniją aplankiusių turistų skaičiaus pagrindiniai grafikai

Kadangi duomenų sklaidumas su metais didėja, juos pirmiausiai išlogaritmuosime, o po to diferencijuodami padarysime stacionariais.

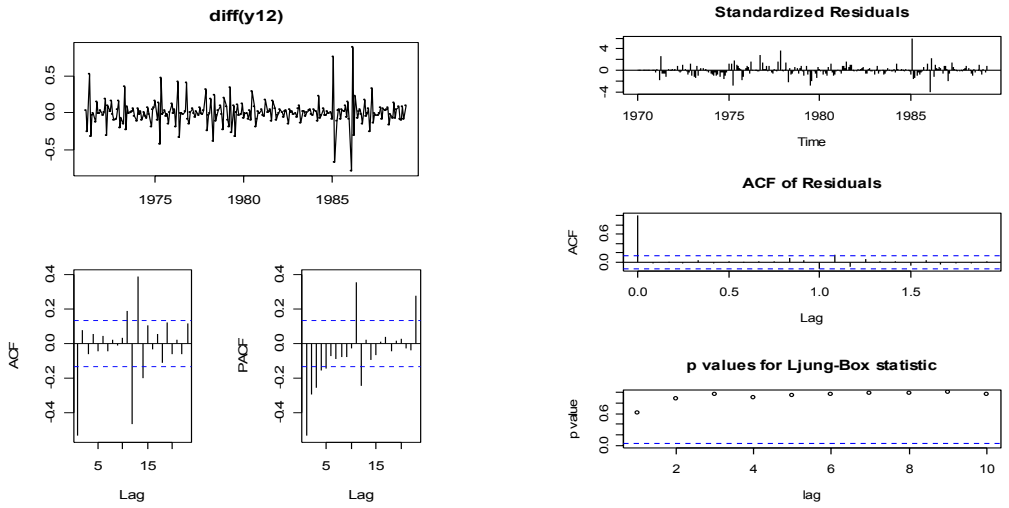
```
tsdisplay((y1 <- diff(log(spain))))      # = (1-L)y_t = y_t - y_t-1
tsdisplay((y12 <- diff(log(spain),12))) # = (1-L^12)y_t = y_t - y_t-12
```



3.40 pav. Laikinės sekos `spain` paprastųjų ir sezoninių skirtumų grafikai

Matome, kad sezoninių skirtumų seka žymiai panašesnė į stacionarią, todėl ateityje nagrinėsime laikinę seką `y12`. Vis dar stebimą šios sekos vidurkio svyravimą pašalinsime ją dar kartą (paprastai) diferencijuodami.

```
tsdisplay(diff(y12))
```



3.41 pav. diff(y12) grafikai (kairėje) ir MAR modelio (žr. žemiau) diagnostika (dešinėje)

Vienintelis ryškus `diff(y12)`'kos ACF pikas taške 1 ir tolygiai mažėjanti PACF siūlo MA(1) modelį, o reikšmingi pikai taško 12 aplinkoje gali atsirasti dėl adityvių arba multiplikatyvių sezoninių faktorių – mes pasiūlysime tris jų paaiškinimo variantus (čia `y=log(spain)`).

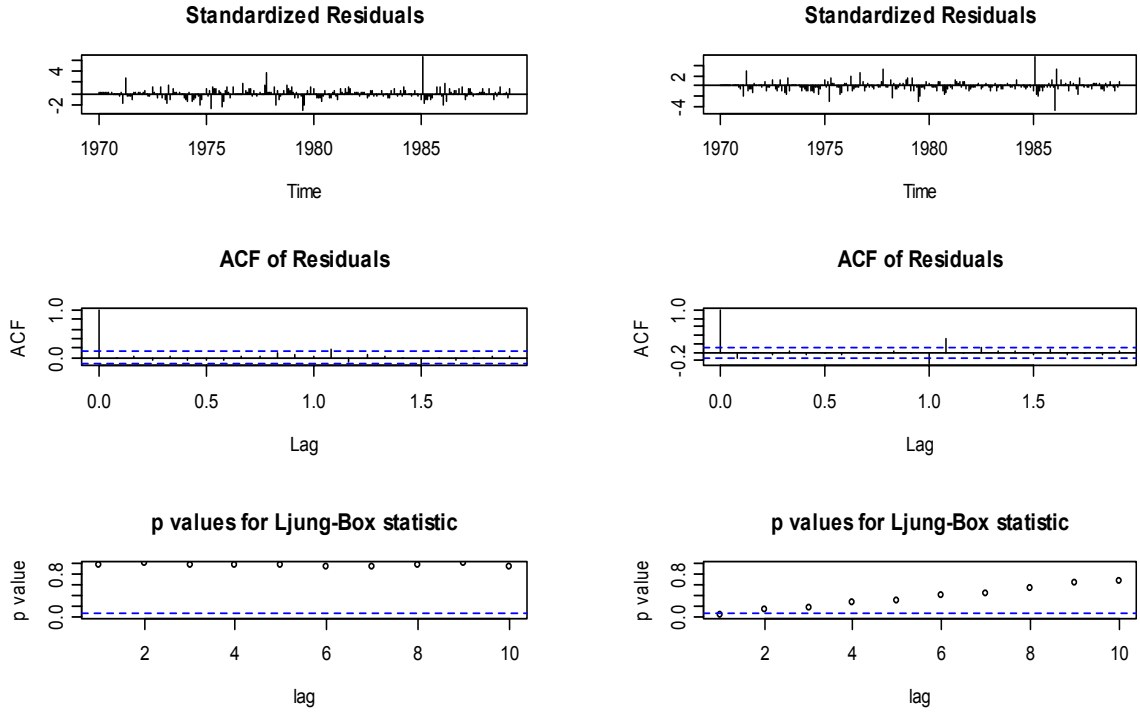
$(1-L^{12})(1-L)(1-a_{12}L^{12})y_t = (1+b_1L)w_t$ (multiplikatyvusis autoregresinis modelis - MAR)

$(1-L^{12})(1-L)y_t = (1+b_1L)(1+b_{12}L^{12})w_t$ (multiplikatyvusis slenkamojo vidurkio modelis - MMA)

$(1-L^{12})(1-L)y_t = (1+b_1L+b_{12}L^{12})w_t$ (adityvusis slenkamojo vidurkio modelis - AMA)

```
y=log(spain)

MAR=arima(y, order = c(0,1,1), seasonal = list(order=c(1,1,0)))
MAR
Series: y
ARIMA(0,1,1)(1,1,0)[12] model

Coefficients:
          ma1      sar1
      -0.7452  -0.4085
s.e.   0.0425   0.0627

sigma^2 estimated as 0.01378:  log likelihood = 156.16,  aic = -306.32
tsdiag(MAR) # Žr. 3.40 pav. (dešinėje)


MMA= arima(y, order = c(0,1,1), seasonal = list(order=c(0,1,1)))
MMA
Series: y
ARIMA(0,1,1)(0,1,1)[12] model

Coefficients:
          ma1     sma1
      -0.7354  -0.7267
s.e.   0.0455   0.0515
```

```
sigma^2 estimated as 0.01118:  log likelihood = 175.52,  aic = -345.04
tsdiag(MMA)
AMA=arima(y,   order=c(0,1,12),   fixed=c(NA,0,0,0,0,0,0,0,0,0,0,NA),   seasonal=
list(order=c(0,1,0)))
AMA
Series: y
ARIMA(0,1,12)(0,1,0)[12] model

Coefficients:
          ma1   ma2   ma3   ma4   ma5   ma6   ma7   ma8   ma9  ma10  ma11     ma12
      -0.6930     0     0     0     0     0     0     0     0     0     0  -0.2777
s.e.   0.1068     0     0     0     0     0     0     0     0     0     0   0.1047

sigma^2 estimated as 0.01522:  log likelihood = 145.08,  aic = -284.17
tsdiag(AMA)
```



3.42 pav. Modelių MMA (kairėje) ir AMA (dešinėje) diagnostiniai grafikai (MMA neabejotinai geresnis)

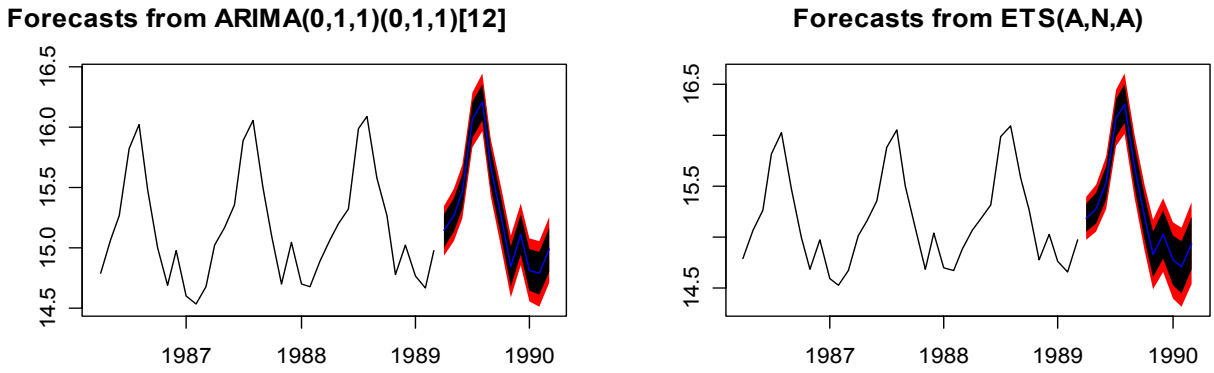
Sprendžiant pagal AIC reikšmes, geriausias tarp šių modelių (ir, galimas daiktas, tarp visų SARIMA modelių) yra MMA modelis, kurį trumpai žymėsime SARIMA$(0,1,1)(0,1,1)_{12}$. Šį modelį (užrašykite jį) jau galima panaudoti Ispanijos turistų skaičiaus logaritmui prognozuoti.

```
plot(forecast(MMA,12),include=36) # Žr. 3.41 pav. žemiau
```

Įdomu pastebėti, kad visiškai automatizuota eksponentinio glodinimo procedūra duoda praktiškai tą patį rezultatą:

```
y.ets=ets(y)
plot(forecast(y.ets,h=12),include=36) # Žr. 3.43 pav. žemiau
```

3.31 UŽDUOTIS. 1. `spain.p` modelį galima charakterizuoti ir pagal jo tikslumą (t.y., apskaičiuoti jo istorinių duomenų ME, MSE, MAE, MPE ir MAPE – žr. `summary(spain.p)`). Apskaičiuokite kai kurias iš šių charakteristikų MMA modeliui. **2**. Užrašykite skaitines abiejų modelių prognozės reikšmes.

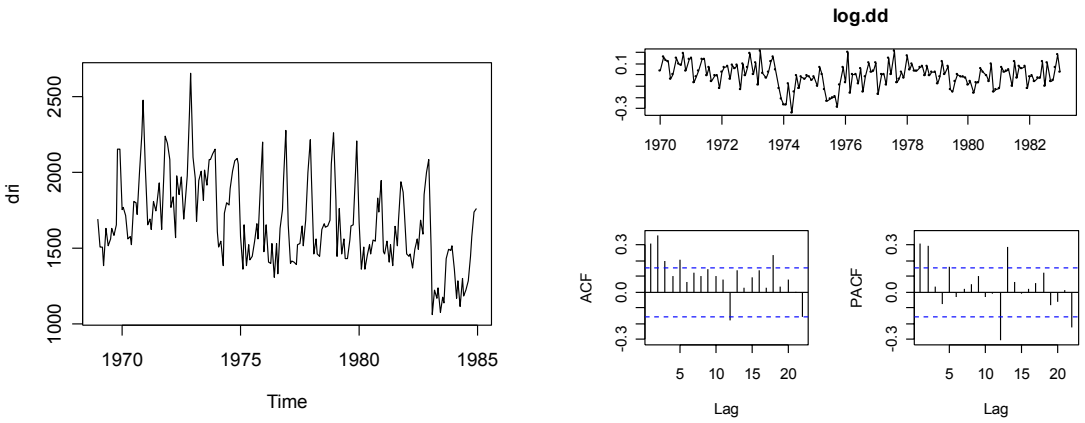


3.43 pav. Prognozė pagal MMA ir eksponentinio glodinimo modelius

3.32 UŽDUOTIS. `fma` pakete rasite `airpass` duomenis, kurių logaritmai (kodėl logaritmai?) paprastai aprašomi klasikiniu „oro linijų“ modeliu SARIMA$(0,1,1)(0,1,1)_{12}$. Sudarykite šį modelį. Palyginkite jį su eksponentinio glodinimo modeliu.

3.13 pavyzdys. Ištirsime ar saugos diržų privalomas naudojimas (nuo 1983 m. vasario) sumažino mirčių skaičių Didžiosios Britanijos keliuose. Surinkę `data(Seatbelts);Seatbelts`, pamatysite mums reikalingus mėnesinius duomenis (nuo 1969 m. sausio iki 1984 m. gruodžio).

```
dri=Seatbelts[,"drivers"] # Žuvusiųjų skaičius
dri
     Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969 1687 1508 1507 1385 1632 1511 1559 1630 1579 1653 2152 2148
1970 1752 1765 1717 1558 1575 1520 1805 1800 1719 2008 2242 2478
.....................
plot(dri)
```



3.44 pav. `dri` grafikas (kairėje) ir `log.dd` (žr. žemiau) grafikas (dešinėje)

Turimus duomenis natūralu išskaidyti į dvi grupes ir patikrinti hipotezę apie vidurkių lygybę. Deja, standartinis Stjudento testas čia netinka, nes duomenys grupėse yra priklausomi. Vis tik:

```
law=Seatbelts[,"law"] # 0 iki 1983m. vasario ir 1 - vėliau
t.test(dri~law)
t = 8.53, df = 33.732, p-value = 6.181e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 301.6697 490.4420
sample estimates:
mean in group 0 mean in group 1
       1717.751        1321.696
```

taigi, jei taikytume Stjudento testą, hipotezę apie vidurkių lygybę tektų neabejotinai atmesti.

Pabandysime atsižvelgti į duomenų priklausomumą. Preliminarų modelį sudarysime, nagrinėdami duomenis iki 1983 m. sausio 31 d.

```
log.d=window(log(dri),end=1983)
log.dd=diff(log.d,12)
tsdisplay(log.dd) # Žr. 3.42 pav.
```

Sezoniškai diferencijuotas procesas `log.dd` panašus į stacionarų, jo modelį parinksime su `auto.arima` funkcija.

```
auto.arima(log.d, d=0, D=1, max.p=2, max.q=2, max.P=2, max.Q=2, max.order=5, alpha=0.05)
Series: log.d
ARIMA(1,0,1)(0,1,2)[12] model

Coefficients:
         ar1      ma1     sma1     sma2
      0.9482  -0.6573  -0.8271  -0.1729
s.e.  0.0418   0.1058   0.1433   0.0914

sigma^2 estimated as 0.005362:  log likelihood = 173.54,  aic = -337.07
```

Dabar šį modelį praplėsime visoms laiko reikšmėms, atsižvelgdami į struktūrinį situacijos pasikeitimą, atsiradusį 1983m. vasarį (šį pasikeitimą naujajame modelyje atitiks naujas fiktyvusis kintamasis `law`).

```
(fit=arima(log(dri),order=c(1,0,1),seasonal=list(order=c(0,1,2)),xreg=law))
Series: log(dri)
ARIMA(1,0,1)(0,1,2)[12] model

Coefficients:
         ar1      ma1     sma1     sma2     xreg
      0.9471  -0.6618  -0.8533  -0.1343  -0.2430
s.e.  0.0391   0.0990   0.3148   0.0982   0.0479

sigma^2 estimated as 0.005274:  log likelihood = 202.45,  aic = -392.9
```

Nustatytasis SARIMA(1,0,1)(0,1,2)$_{12}$ modelis yra visai priimtinas (išbandykite `tsdiag(fit)`), regresijos koeficientas -0,2430 yra neabejotinai reikšmingas, `log(dri)` sumažėjo 0,2430, o pats `dri` dabar sudaro tik `exp(-0.2430)*100% = 0.7842715*100%=78%` ankstesnio lygio.
Mirčių autokeliuose skaičių galima prognozuoti su

```
plot(forecast(fit,h=12,xreg=rep(1,12)),include=24)
```

Trumpai aptarsime kitą šio uždavinio variantą. Nuo 1975 m. sausio iki 1982 m. sausio mirčių skaičius buvo gana reguliarus sezoninis procesas. Šią laikinę seką aprašysime multiplikatyviuoju SARIMA$(1,0,0)(1,0,0)_{12}$ modeliu. Tam galima vartoti klasikinę `arima` funkciją (išbandykite patys) arba `dynlm` funkciją iš `dynlm` paketo. Pastaroji funkcija skirta vertinti tiesinės regresijos modelį tuo atveju, kai lygties dešinėje yra vėluojančių narių

```
library(dynlm)
uk <- log10(dri)
dfm <- dynlm(uk ~ L(uk, 1) + L(uk, 12)) # Vartojame visus duomenis
dfm
# Vartosime tik dalį duomenų
dfm <- dynlm(uk ~ L(uk, 1) + L(uk, 12), start = c(1975, 1), end = c(1982, 12))
dfm
```

3.33 UŽDUOTIS. Vartodami komandas `library(datasets); data(USAccDeaths); acc=USAccDeaths`, nuskaitykite mėnesinius duomenis apie mirčių skaičių JAV greitkeliuose tarp 1973 m. ir 1978 m. Įsitikinkite, kad šiuos duomenis sėkmingai aprašo modelis $ddacc_t - \mathtt{mean}(ddacc_t) = (1+b_1L)(1+b_{12}L^{12})w_t$ (čia $ddacc_t = (1-L)(1-L^{12})acc_t$). Įvardinkite šį modelį. Apskaičiuokite jo koeficientus ir išbrėžkite 12 mėn prognozę.

3.34 UŽDUOTIS. Išnagrinėkite `elecnew` laikinę seką iš `forecast` paketo. Raskite ją tinkamai aprašantį modelį. Prognozuokite šios sekos reikšmes 24 mėnesius į priekį. Palyginkite su tikrais duomenimis iš www.eia.doe.gov .

3.35 UŽDUOTIS. Ištirkite `data(nottem);?nottem` duomenis ir sudarykite jų SARIMA modelį.

3.36 UŽDUOTIS. Žemiau pateikti duomenys yra Johnson&Johnson kompanijos vienos akcijos ketvirtinės pajamos (nuo 1960 m. iki 1980 m.):

```
jnj=structure(c(0.71, 0.63, 0.85, 0.44, 0.61, 0.69, 0.92, 0.55, 0.72, 0.77,
0.92, 0.6, 0.83, 0.8, 1, 0.77, 0.92, 1, 1.24, 1, 1.16, 1.3, 1.45, 1.25, 1.26,
1.38, 1.86, 1.56, 1.53, 1.59, 1.83, 1.86, 1.53, 2.07, 2.34, 2.25, 2.16, 2.43,
2.7, 2.25, 2.79, 3.42, 3.69, 3.6, 3.6, 4.32, 4.32, 4.05, 4.86, 5.04, 5.04, 4.41,
5.58, 5.85, 6.57, 5.31, 6.03, 6.39, 6.93, 5.85, 6.93, 7.74, 7.83, 6.12, 7.74,
8.91, 8.28, 6.84, 9.54, 10.26, 9.54, 8.729999, 11.88, 12.06, 12.15, 8.91, 14.04,
12.96, 14.85, 9.99, 16.2, 14.67, 16.02, 11.61), .Tsp = c(1960, 1980.75, 4),
class = "ts")
```

Pabandykite pagrįsti modelį $(1-L)(1-L^4)y_t = (1-0.681L)(1-0.315L^4)w_t$ (čia $y_t = \log(jnj_t)$). Iš naujo įvertinkite šį „oro linijų“ modelį pagal pirmuosius 76 ketvirčius ir pateikite 8 ketvirčių prognozę. Palyginkime su tikraisiais duomenimis.

3.37 UŽDUOTIS. [redacted] Nusiskaitykite duomenų failą US.txt iš Data/Enders direktorijos. Šios lentelės pirmajame stulpelyje yra pateikti duomenys apie JAV ketvirtinių (nuo 1960:1 iki 1991:4) M1 pinigų kiekį.

- Išbrėžkite `M1`, `lM1=log(M1)` ir `dlM1=diff(log(M1))` grafikus. Ką galite pasakyti apie jų trendus ir sezoniškumą? O apie laikinės sekos `dlM1` ACF ir PACF funkcijų elgesį taškuose 4, 8 ir 12? (Taikykite funkciją `tsdisplay` iš `forecast` paketo.)
- Sudarykite sezoninių `log(M1)` skirtumų `d4M1=diff(log(M1),4)` laikinę seką ir ištirkite jos ACF ir PACF funkcijas.

- Laikinę seką `d4M1` aprašykite AR(1) procesu: $\text{d4M1}_t = a_0 + a_1 \text{d4M1}_{t-1} + w_t$ ir, remdamiesi diagnostiniais grafikais, paaiškinkite, kodėl šis modelis nėra tinkamas.
- Laikinę seką `lM1` aprašykite modeliu SARIMA$(1,0,0)(0,1,1)_4$. Kodėl jis nepriimtinas?
- Sudarykite laikinę seką `dd4M1 = diff(d4M1)` ir modelį $\texttt{dd4M1}_t = (1+b_4 L^4)\, w_t$. Paaiškinkite, kuo jis pranašesnis už kitus modelius. ◀

EViews

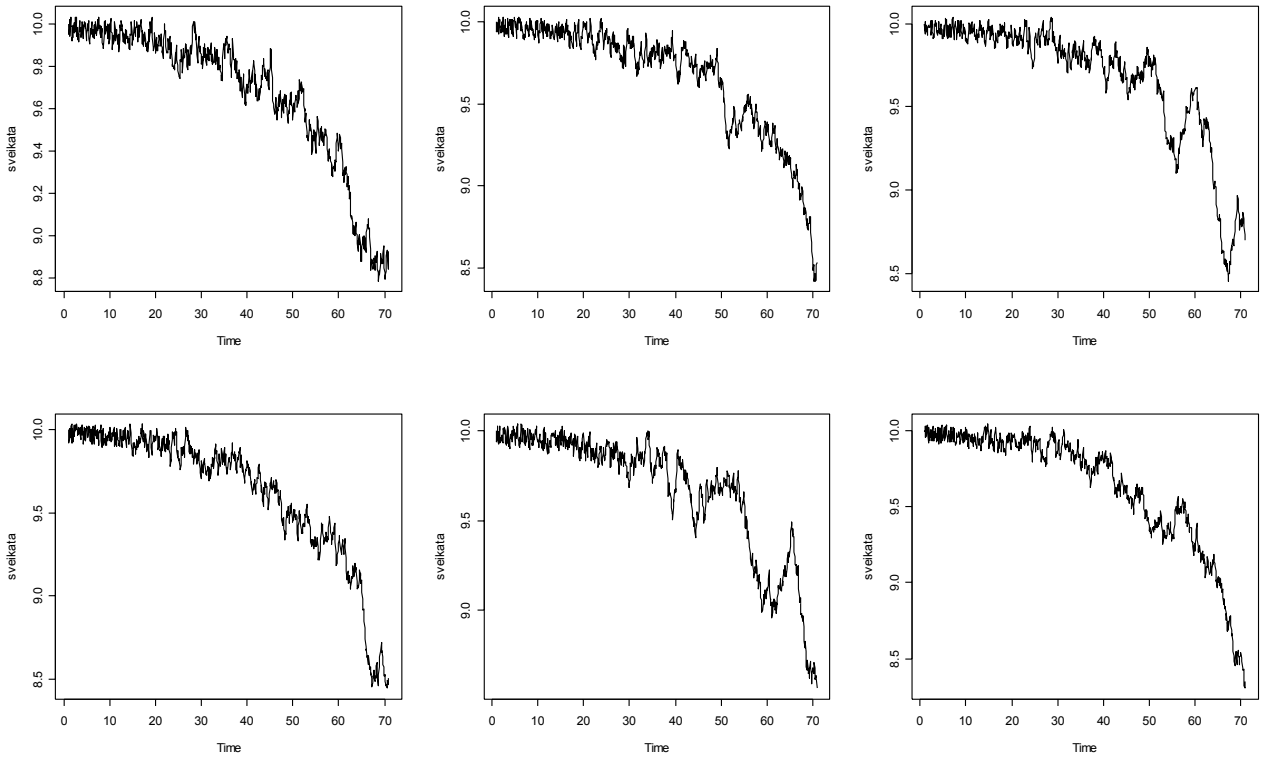
Žr. http://www.hkbu.edu.hk/~billhung/econ3600/application/app01/app01.html – tai Dickey-Fuller Unit Root (Stationarity) Test

Pamokantis pavyzdys

Žmogaus sveikata iki maždaug 30 metų yra daugmaž pastovi (ir gera). Jei ji kartais sutrinka (smukteli žemyn), tai žmogus paprastai pasveiksta ir sveikata vėl sugrįžta į stacionarią padėtį (šiame pavyzdyje lygią 10). Taigi sveikatą šiuo laikotarpiu galima aprašyti kaip stacionarų, pvz., AR(1) procesą. Vėliau sveikata ne tik po truputį blogėja, bet ir pradeda jaustis ligų kaupimosi reiškiniai – naują dieną žmogus pradeda ne bent kiek sveikesnis, bet vėl toje būsenoje, kurioje buvo vakar (vienetinės šaknies arba atsitiktinio klaidžiojimo požymis). Sveikata gali laikinai pagerėti, bet gali ir pablogėti (retos ekskursijos aukštyn, ir dažnesnės - žemyn). Tiesą sakant, į šį modelį dar reiktų įtraukti ir dreifo (irgi žemyn...) reiškinį. Reikia taip pat turėti galvoje, kad (gyvenimo) šokai turėtų būti nesimetriški - daugiau šansų sulaukti ligos ar traumos, negu netikėto sveikatos pagerėjimo.

Žemiau pateiktos šešios gyvenimo istorijos.

```
sv[i]-10 =                                      # AR procesas su vidurkiu 10
+ (1-exp(-(1:840)/200))[i] * (sv[i-1]-10)       # Autoregresijos koef artėja į 1
- i/1000000000                                  # Neigiamas ir stiprėjantis dreifas
+ sample(((-8):3)/100,1)     # i=1:12*70        # Nesimetriški šokai
```



3.45 pav. Šešios gyvenimo istorijos

3.34 UŽDUOTIS. Žemiau pateikta laikinė seka `a002x`:

```
a002x = structure(c(1.22, 3.38, 3.13, 1.63, 0.44, 0.68, -0.87, 1.14, 2.35, 1.29, 1.27,
0.02, -0.52, 3.36, -0.12, 3.38, -1.28, 2.95, 2.09, -1.21, 3.43, 4.07, 2.34, 4, 2.12, 4.3,
2.63, 1.22, 4.44, 2.63, 1.82, 0.93, 4.81, 4.38, 9.38, 6.17, 7.5, 5.39, 6.48, 3.95, 5.74,
6.01, 5.73, 6.4, 4.47, 5.73, 7.08, 11.21, 12.08, 15.94, 13.91, 12.8, 9.19, 7.08, 6.08,
5.58, 6.91, 8.71, 13.16, 14.27, 12.2, 13.39, 10.2, 12.56, 13.89, 12.73, 14.78, 12.73,
15.93, 14.43, 16.92, 18.29, 22.98, 23.36, 26.04, 19.51, 19.62, 16.95, 18.95, 20.42,
20.71, 23.75, 24.33, 21.16, 24.18, 23.69, 25.49, 24.34, 25.2, 27.21, 30.83, 29.87, 32.21,
31.85, 31.69, 31.25, 28.65, 31.08, 30.91, 35.81, 34.51, 33.95, 35.86, 33.81, 36.76,
35.98, 40.21, 37.96, 44.07, 43.01, 43.42, 42.96, 44.56, 46.37, 44.24, 43.34, 42.87,
43.28, 51.01, 49.89, 48.73, 44.66, 45.26, 47.21, 49.21, 48.28, 49.92, 52.45, 54.46, 56.8,
55.52, 53.79, 54.13, 52.69, 53.19, 55.39, 57.41, 59.73, 59.36, 60.48, 59.87, 62.13,
65.65, 67.57, 66.45, 68.83, 72.19, 71.97, 72.35, 73.44), .Tsp = c(1, 150, 1), class =
"ts")
```

Nubrėžkite jos grafiką. Su `lm` funkcija išskirkite jos kvadratinį (?) trendą ir apytiksliai nustatykite (ko gero) stacionarių liekanų struktūrą. Su `arima` funkcija patikslinkite proceso modelį. *Nuoroda*. Kvadratinį trendą įtraukti į `arima` funkciją galima taip: `arima(...,xreg=cbind(1:150, (1:150)^2))`.

4. Finansinės laikinės sekos ir jų charakteristikos

Pagrindinis finansinių laikinių eilučių teorijos ir praktikos objektas yra turto[1] (angl. asset) vertės kitimo analizė. Tai didžiai empirinis mokslas, bet, kaip įprasta, teorija padeda priimti sprendimus. Pagrindinis požymis, kuris skiria finansines nuo kitų laikinių eilučių, yra papildomas neapibrėžtumas. Pvz., svarbus akcijų grąžų (angl. returns) laikinės sekos parametras yra jos volatilumas[2], kuris, deja, nėra tiesiogiai matuojamas dydis. Atkreipsime dėmesį dvi priežastis, dėl kurių paprastai analizuojamos turto grąžos, o ne kainos. Pirma, eiliniam investuotojui grąža yra visa apimanti ir neprikausanti nuo mastelio investicijų sėkmingumo charakteristika ir, antra, kainų statistinės charateristikos yra sudėtingesnės negu grąžų. Kita vertus, finansinių laikinių eilučių teorija nagrinėja ne tik grąžas: ji tiria ir palūkanų normas, valiutų keitimo kursus, obligacijų pelningumą ar akcijų ketvirtinius dividendus.

Šiame skyriuje laikinių sekų reikšmes žymėsime raide P_t (P nuo Price (liet. kaina)). Dažniausiai nagrinėjamos dvi grąžų rūšys – paprastosios $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ (šie dydžiai panašūs į procesų skirtumus, bet nėra lygūs jiems) ir sudėtinės (kitaip logaritminės) $r_t = \ln\frac{P_t}{P_{t-1}} = \ln(1+R_t)$. Pažymėsime, kad tuomet, kai R_t maža, $r_t \approx R_t$ (prisiminkite ln skleidimą Teiloro eilute).

Paprastosios grąžos R_t ir tolygiai sukauptosios (kitaip logaritminės) grąžos r_t:

$$1+R_t = \frac{P_t}{P_{t-1}}, r_t = \ln\frac{P_t}{P_{t-1}}, r_t = \ln(1+R_t), R_t = e^{r_t} - 1$$

$$1+R_t[k] = (1+R_t)(1+R_{t-1})...(1+R_{t-k+1}) =$$

$$= (1+vidutinė\{R_t[k]\})^k = (1+(\left[\prod_{j=0}^{k-1}(1+R_{t-j})\right]^{1/k} - 1))^k : \text{ t.y.,}$$

jei turtas buvo kaupiamas k metų, tuomet vidutinė metinė grąža yra apibrėžiama taip: $\left[\prod_{j=0}^{k-1}(1+R_{t-j})\right]^{1/k} - 1$

$$r_t[k] = r_t + r_{t-1} + ... + r_{t-k+1} = k \times vidutinė\{r_t[k]\} =$$

$$= k \times \frac{r_t + r_{t-1} + ... + r_{t-k+1}}{k} \ (k \times \text{vidutinė metinė logaritminė grąža})$$

$$K_n = K_0 \exp(r \times n) \ (\textit{kapitalo K vertė po n metų})$$

Svarbi prielaida, daroma analizuojant grąžas, yra ta, kad paprastosios grąžos yra nepriklausomi vienodai pasiskirstę (n.v.p.) normalieji (Gauso) a.d. Deja, dėl šios prielaidos atsiranda problemų: 1) paprastosios grąžos visuomet didesnės už -1 (o normalieji a.d. yra neaprėžti iš apačios), 2) jei grąžos yra Gauso, tai kelių laikotarpių suminė grąža $R_t[k]$ nebebus tokia (nes Gauso dydžių sandauga

[1] Turtas – likvidinių ir nelikvidinių vertybių suma, kuria disponuoja ekonominis subjektas.
[2] Kartais jis suprantamas kaip grąžų standartinis nuokrypis.

turi kitokį skirstinį), 3) normalumo prielaida prieštarauja empiriniams faktams. Viena iš galimų išeičių – tarti, kad logaritminės grąžos yra n.v.p. normalieji a.d.

4.1 UŽDUOTIS. Parašykite keturias funkcijas:

`P2R(x)` - kainų laikinę seką `x` paverčiančią paprastosiomis grąžomis
`P2r(x)` - kainų laikinę seką `x` paverčiančią logaritminėmis grąžomis
`R2P(x, Po=1)` - paprastųjų grąžų seką `x` paverčiančią kainų seka (pradinė kaina `Po` lygi 1)
`r2P(x, Po=1)` - logaritminių grąžų seką `x` paverčiančią kainų seka (pradinė kaina `Po` lygi 1)◀

Finansinių laikinių eilučių pavyzdžių galima rasti

- ...\Data\Tsay direktorijoje

ir R paketuose:

• base
`euro`
`longley`
	
	Conversion Rates of Euro Currencies
	Longley's Economic Regression Data

• Ecdat	Data sets for econometrics
• forecast `data(package = "forecast")`	

• fSeries
`data(package="fSeries")`
`library(fSeries)`
`data(RS)`
`?RS`
	`1 'RS.txt' Monthly 91 day Treasury Bill rate,`
	`2 'R20.txt' Monthly Yield on 20 Year UK Gilts,`
	`3 'RSQ.txt' Quarterly 91 day Treasury Bill rate,`
	`4 'R20Q.txt' Quarterly Yield on 20 Year UK Gilts,`
	`5 'RSQREAL.txt' Quarterly real 91 day Treasury Bill rate,`
	`6 'FTAPRICE.txt' FTA All Share Price Index,`
	`7 'FTADIV.txt' FTA All Share Dividend Index,`
	`8 'FTARET.txt' FTA All Share Nominal Returns,`
	`9 'RPI.txt' UK Retail Price Index,`
	`10 'EXCHD.txt' Dollar/Sterling Exchange Rate,`
	`11 'EXCHQ.txt' Dollar/Sterling Exchange Rate,`
	`12 'SP500.txt' SP 500 Annual Data Index,`
	`13 'SP500R.txt' SP 500 Real Returns,`
	`14 'SP500D.txt' SP 500 Daily Data Index,`
	`15 'FT30.txt' FT 30 Index,`
	`16 'FTSE100.txt' FTSE 100 Index,`
	`17 'CTLD.txt' Courtaulds Share Price,`
	`18 'LGEN.txt' Legal and General Share Price,`
	`19 'PRU.txt' Prudential Share Price`
	`ir t.t.`

• lmtest
`bondyield`
`currencysubstitution`
`growthofmoney`
`jocci`
`moneydemand`
`unemployment`
`valueofstocks`
`wages`
	
	Bond Yield
	Currency Substitution
	Growth of Money Supply
	U.S. Macroeconomic Time Series
	Demand for Money
	Unemployment Data
	Value of Stocks
	Wages

• MASS `SP500`	 Returns of the Standard and Poors 500
• stats `EuStockMarkets`	 Daily Closing Prices of Major European Stock Indices, 1991-1998
• tseries	 Nelson-Plosser Macroeconomic Time Series

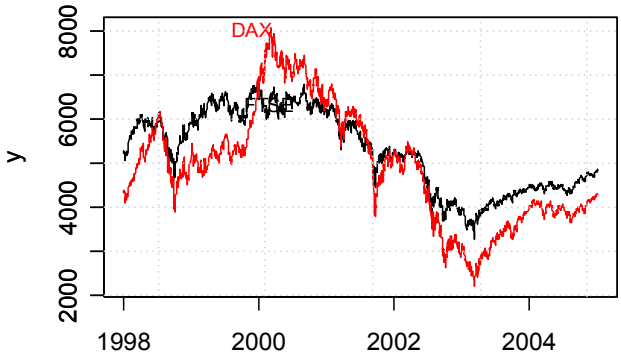
`NelPlo`
`USeconomic`
`tcm`
`tcmd`
	U.S. Economic Variables
	Monthly Yields on Treasury Securities
	Daily Yields on Treasury Securities

• urca
`denmark`
`ecb`
`finland`
`npext`
`nporg`
	
	Data set for Denmark, Johansen & Juselius (1990)
	Macroeconomic data of the Euro Zone
	Data set for Finland, Johansen & Juseliues (1990)
	Nelson & Plosser extended data set
	Nelson & Plosser original data set

Jei jūsų komputeris prijungtas prie interneto, tai R su `its` paketu kai kurių biržų indeksų duomenis gali atsisiųsti, pvz., šitaip:

```
library(its)
x1 <- priceIts(instrument = c("^ftse"), start = "1998-01-01",
                    quote = "Close")
x2 <- priceIts(instrument = c("^gdax"), start = "1998-01-01",
                    quote = "Close")
x <- union(x1,x2)
names(x) <- c("FTSE","DAX")
plot(x,lab=TRUE)
```

Įdomu tai, kad abiejų indeksų, Vokietijos DAX ir Didžiosios Britanijos FTSE, elgesys panašus į atsitiktinį klaidžiojimą. Antra vertus, indeksai klaidžioja „panašiai“, taigi, gali būti, kad jie kointegruoti (žr. ??? skyrių).



4.1 pav. FTSE ir DAX indeksų grafikai

4.1. Grąžų statistinės charakteristikos

4.1 pavyzdys. Panagrinėsime kasdienines (nuo 1990 m. sausio iki 1999 gruodžio) Alcoa firmos akcijų logaritmines grąžas (faile Data\Tsay\d-aa9099.dat jos pateiktos procentais).

- Apskaičiuosime jų empirinį vidurkį, dispersiją, asimetriją (angl. skewness), ekscesą (angl. (excess) curtosis), minimumą ir maksimumą.
- Logaritmines grąžas transformuosime į paprastąsias ir apskaičiuosime tas pačias charakteristikas.
- Ar logaritminių grąžų empirinis vidurkis reikšmingai skiriasi nuo nulio?

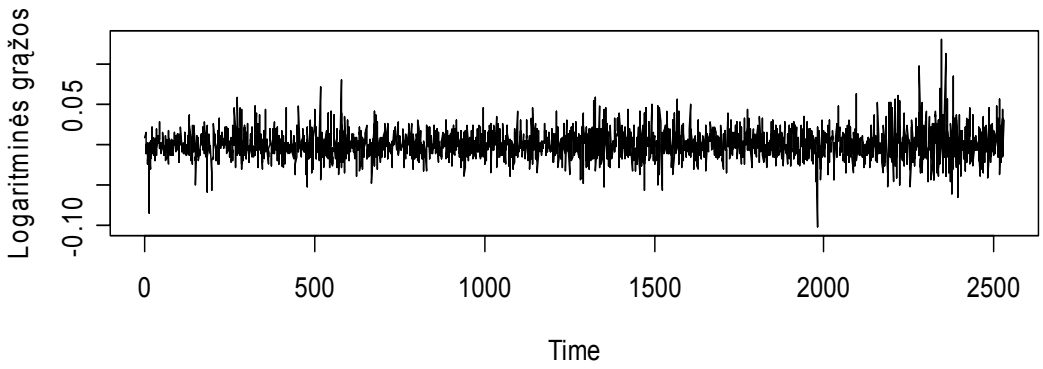
R

Surinkite

```
d.aa9099 = ts(read.table(file.choose()))
```

ir nuvairuokite į Data\Tsay direktorijos d-aa9099.dat failą (duomenų periodiškumas, gal būt, 5 (darbo) dienos, bet, kadangi, matyt, dar buvo išmestos ir švenčių dienos, frequency nenurodysime).

```
plot(d.aa9099/100, ylab="Logaritminės grąžos") # daliname iš 100
abline(0,0,col=2)
```



4.2 pav. Alcoa akcijų logaritmines grąžos

Štai funkcija, skirta skaičiuoti empirinėms laikinių eilučių charakteristikoms.

```
SUMM <- function(x)
{
vid <- mean(x)
disp <- var(x)
asim <- sum((x-mean(x))^3)/(length(x)*(sd(x))^3)
eksc <- sum((x-mean(x))^4)/(length(x)*(sd(x))^4)
cat(" mean     =",vid,"\n variance =",disp,"\n skewness =",asim,
"\n kurtosis =",eksc,"\n minimum  =",min(x),"\n maximum  =",max(x),"\n")
}
```

```
> SUMM(d.aa9099/100)
 mean     = 0.000678576
 variance = 0.0003282077
 skewness = 0.4639015
 kurtosis = 6.203343
 minimum  = -0.10237
 maximum  = 0.13152
```

Transformuosime į paprastąsias grąžas.

```
s.d.aa9099 <- exp(d.aa9099/100)-1 # s. = simple
```

```
> SUMM(s.d.aa9099)
 mean     = 0.0008434447
 variance = 0.0003317869
 skewness = 0.6024206
 kurtosis = 6.636359
 minimum  = -0.0973045
 maximum  = 0.1405607
```

Ar vidurkis reikšmingai skiriasi nuo 0? Pasiremsime tuo, kad grąžos yra n.a.d., todėl galime taikyti Student‘o kriterijų.

- Logaritminės grąžos

```
> t.test(d.aa9099) # Logaritminės gražos procentais

        One Sample t-test

data:  aa9099
t = 1.8833, df = 2527, p-value = 0.05978
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.002797263  0.138512453
sample estimates:
mean of x
0.0678576

> t.test(d.aa9099/100) # t testo rezultatas nepriklauso nuo mastelio

        One Sample t-test

data:  aa9099/100
t = 1.8833, df = 2527, p-value = 0.05978
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -2.797263e-05  1.385125e-03
sample estimates:
  mean of x
0.000678576
```

Abiem atvejais *p* reikšmės tos pačios, 6% reikšmingumo lygiu vidurkis nelygus nuliui (jis teigiamas, taigi ši akcija pelninga). Pažymėsime, kad Studento testas tokį nedidelį nuokrypį nuo nulio skelbia reikšmingu (ir) dėl to, kad stebinių labai daug – 2528.

- Paprastosios grąžos

```
> t.test(s.d.aa9099)

        One Sample t-test

data:  s.d.aa9099
t = 2.3282, df = 2527, p-value = 0.01998
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.0001330540 0.0015538354
sample estimates:
   mean of x
0.0008434447
```

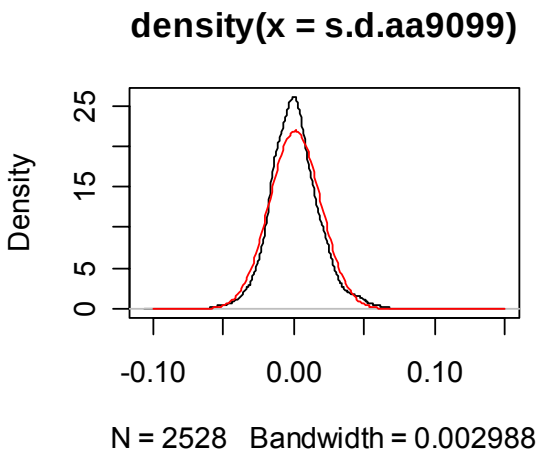
Paprastųjų grąžų vidurkio skirtumas nuo nulio yra reikšmingas (2% reikšmingumu).

Beje, `s.d.aa9099` skirstinys yra tikrai nenormalus (žr. `shapiro.test` rezultatus ir `density` grafiką žemiau), bet kadangi stebėjimų daug (virš 2000), *t* testo taikymas teisėtas.

```
> shapiro.test(s.d.aa9099)

        Shapiro-Wilk normality test
data:  s.aa9099
W = 0.9676, p-value < 2.2e-16
```

Tai rodo ir grafikas dešinėje.

```
plot(density(s.d.aa9099))
x=seq(-0.10,0.15,by=0.001)
lines(x,dnorm(x,mean(s.d.aa9099),
sd(s.d.aa9099)),col=2)
```



4.3 pav. s.d.aa9099 tankis

Atkreipsime dėmesį – paprastųjų grąžų tankio uodegos (žr. juodą kreivę dešinėje) yra „sunkesnės“ (jų grafikas yra aukščiau) už normaliąsias (raudona spalva).

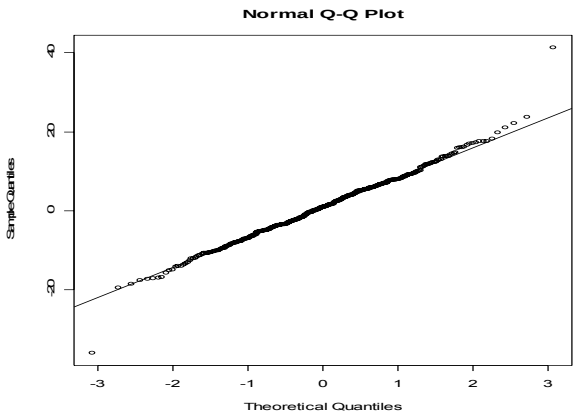
4.2 pavyzdys. Panagrinėsime Alcoa mėnesines logaritmines grąžas (procentais) nuo 1962 m. sausio iki 1999 m. gruodžio (iš viso 456 stebiniai).

- Ką galima pasakyti apie šios imties skirstinį ir pagrindines skaitines charakteristikas?
- Kam lygi metinė logaritminė grąža nurodytu laiko periodu?
- Kam lygi vidutinė metinė paprastoji grąža (angl. annualized (average) simple return) nurodytu laiko periodu?
- Investuotojas pirko Alcoa akcijų už vieną dolerį 1962 m. pradžioje. Kokia jų vertė 1999 m. pabaigoje?

Surinkite `m.aa6299 = ts(read.table(file.choose()), start = 1962, freq = 12)` ir nuvairuokite į Data\Tsay direktorijos m-aa6299.dat failą.

```
> SUMM(m.aa6299/100)
 mean =     0.008488684
 variance = 0.006432212
 skewness = 0.1133180
 kurtosis = 4.736475
 minimum =  -0.35987
 maximum =  0.41302
```

Nors kvantilių grafikas (žr. dešiniau) labai panašus į normalųjį:




```
qqnorm(m.aa6299)
qqline(m.aa6299)
```

tačiau, žiūrint formaliai, šie duomenys tikrai nėra normalieji

```
> shapiro.test(m.aa6299)
```

```
        Shapiro-Wilk normality test

data:  m.aa6299
W = 0.9858, p-value = 0.0002005 # Duomenys tikrai ne normalieji

help.search("jarque") # Ieškome Jarque-Bera normalumo testo
library(tseries)      # Jis yra „tseries“ pakete
jarque.bera.test(m.aa6299)

        Jarque Bera Test

data:  m.aa6299
X-squared = 59.6575, df = 2, p-value = 1.110e-13 # Duomenys tikrai nenormalūs
```

- Apskaičiuosime metinių logaritminių grąžų vidurkį. Kadangi $\ln\frac{P_{t+11}}{P_{t-1}} = r_t + ... + r_{t+12}$, tai metinių logaritminių grąžų r_y vidurkį skaičiuosime taip:

```
> r.y <- apply(matrix(m.aa6299/100,ncol=12,byrow=TRUE),1,sum) # Metinės grąžos
> mean(r.y) # Jų vidurkis
[1] 0.1018642
```

- Apskaičiuokime metinių (paprastųjų) grąžų vidurkį - jis lygus $\exp(\sum_{j=0}^{k-1} r_j / k) - 1$:

```
> exp(mean(r.y))-1
[1] 0.1072331 # Šis vidurkis beveik lygus ankstesniam
```

- Kuo virto 1 doleris, investuotas į Alcoa akcijas 1962 m. pradžioje, 1999 metų pabaigoje? Remsimės mėnesinėmis grąžomis (jų iš viso 456). Kadangi $P_t = P_{t-k}\prod_{j=0}^{k-1}(1+R_{t-j}) =$

$\$1 \cdot \prod_{j=0}^{456-1} e^{\ln(1+R_{456-j})} = \$1 \cdot \exp(r_1 + ... + r_{456})$, tai

```
> exp(sum(m.aa6299/100))
[1] 47.98267
```

kitais žodžiais, $1 virto beveik 48 doleriais.

4.2 UŽDUOTIS. Pakartokite ankstesnio pavyzdžio analizę su American Express akcijomis (failas m-axp7399.dat iš Data\Tsay direktorijos).

4.3 UŽDUOTIS. [redacted] Nuskaitykite metinius duomenis nuo 1880 m. iki 1987 m. iš Data\Stewart\ASCII\jones.dat (šiame faile pateikti JAV bendrasis vidaus produktas (BVP; tai stulpelis `GDP`) ir gyventojų skaičius (stulpelis `pop`)). Simboliu y_t pažymėkite BVP vienam gyventojui dydį. Ištirkite pastovaus augimo modelį $\log y_t = c_1 + c_2 t + w_t$ pagal 1880-1990 metų duomenis. Koks yra vidutinis metinis BVP vienam gyventojui augimo greitis? Ar sudarytas modelis tinkamai aprašo duomenis? Bet kuriuo atveju prognozuokite $\log y_t$ septyneriems metams į priekį.

4.4 UŽDUOTIS. [redacted] Nuskaitykite metinius duomenis nuo 1880 m. iki 1987 m. iš Data\Stewart\ASCII\jones.dat (šiame faile pateikti JAV bendrasis vidaus produktas (BVP; tai stulpelis `GDP`) ir gyventojų skaičius (stulpelis `pop`)). Stochastinio trendo modelis (jis skiriasi nuo 4.3

uždavinyje minėto) atrodo taip: $\Delta \log y_t = c_2 + w_t$. Įvertinkite šio modelio parametrą c_2 (BVP vienam gyventojui augimo greitį) su `arima` ir `lm` funkcijomis pagal 1880-1990 metų duomenis. Įsitikinkite, kad tai tiesiog empirinis vidurkis $\left(\sum_{t=2}^{n} \Delta \log y_t\right)/(n-1)$. Ar sudarytas modelis tinkamai aprašo duomenis? Koks bebūtų atsakymas, prognozuokite $\log y_t$ septyneriems metų į priekį.

4.5 UŽDUOTIS. [redacted] Nusiskaitykite metinius duomenis nuo 1880 m. iki 1987 m. iš Data\Stewart\ASCII\jones.dat (šiame faile pateikti JAV bendrasis vidaus produktas (BVP; tai stulpelis `GDP`) ir gyventojų skaičius (stulpelis `pop`)). Tarkime, kad žinome tik pirmą ir paskutinę BVP vienam gyventojui reikšmes, t.y., 0.5511 ir 3.6841. Stochastinio trendo modelyje įvertinkite augimo greitį. *Nuoroda*. Įdėmiai perskaitykite 4.4 uždavinio sąlygą.

4.6 UŽDUOTIS. [redacted] Data\Stewart\ASCII\fama.dat faile rasite kelių JAV kompanijų (IBM, Xerox ir Niu Jorko vertybinių popierių biržos NYSE) mėnesines (nuo 1963 m. liepos iki 1968 m. birželio) akcijų grąžas. Pagal 1963 m. liepos, rugpjūčio ir rugsėjo IBM kompanijos duomenis: -0.0040, 0.0259 ir 0.0163, apskaičiuokite šio ketvirčio IBM grąžą. Kokia būtų metinė grąža, jei tokia ketvirtinė grąža liktų ir toliau? Kokia būtų vidutinė mėnesinė grąža?

4.7 UŽDUOTIS. [redacted] 4.6 uždavinyje buvo minimi skaičiai -0.0040, 0.0259 ir 0.0163. Šiuos skaičius atitinkančios paprastosios ketvirtinė ir metinė grąžos yra $R_{ketv} = 0.0385$ ir $R_{met} = 0.1629$. Apskaičiuokite logaritmines grąžas $r_{mėn}$, r_{ketv} ir r_{met}.

4.8 UŽDUOTIS. Su `set.seed(1); x=ts(runif(52,-0.1,0.2),freq=4)` generuokite paprastųjų ketvirtinių grąžų seką. i) Atstatykite pagal ją kainas ($P_0 = 1$) ir ii) apskaičiuokite vidutinę metinę paprastąją grąžą. Kodėl grąžas generuojame iš nesimetriško intervalo [-0.1,0.2]?

4.9 UŽDUOTIS. Faile Data\Tsay\m-bnd.dat yra pateikti JAV vyriausybės obligacijų (angl. bonds) paprastųjų mėnesinių indeksų grąžos nuo 1942:1 iki 1999:12. Sudarykite AR ir MA modelius ketvirtajam stulpeliui (skolos apmokėjimo laikotarpis (angl. maturity) 5 metai). Kuris iš modelių tinkamiausias?

4.10 UŽDUOTIS. Faile Data\Tsay\d-hwp3dx8099.dat yra pateikti 5056 įrašai apie Hewlett-Packard'o akcijų, value-weighted indekso, equal-weighted indekso ir S&P500 indekso kasdienines logaritmines grąžas (procentais) nuo 1980 m. sausio iki 1999 m. gruodžio. Kiekvienai grąžų sekai patikrinkite hipotezę $H_0 : \rho_1 = \rho_2 = ... = \rho_{10} = 0$ su alternatyva H_1 : bent vienas ρ_i nelygus 0 (čia ρ_i yra proceso *i*-oji autokoreliacija). Ar skiriasi atsakymai kompanijos akcijoms ir rinkos indeksams?

4.2. Grąžų ypatingosios savybės

Daugumą finansinių laikinių eilučių nepavyksta aprašyti ARIMA modeliais. Čia[3] panagrinėsime kelis tai patvirtinančius empirinius faktus (šie faktai dažnai vadinamos ypatingosiomis (finansinių eilučių) savybėmis arba keistaisiais faktais (angl. stylized facts ≈ liet. nenatūralieji, rafinuotieji, keistieji faktai)). Aptarsime kelias ypatingasias grąžų savybes (žr. [L, 6 psl.]), susijusias su skirstinio uodegos ir kovariacijų elgesiu.

[3] Žr. taip pat `library(fBasics); ?StylizedFacts`.

Sunkiosios uodegos. Dideli kainų pokyčiai pasirodo žymiai dažniau nei tuo atveju, kai skirstinys yra normalusis. Toks efektas galetų būti paaiškinamas tuo, kad logaritmines grąžos skirstinys turi sunkią uodegą, t. y., kuriam nors baigtiniam skaičiui $a > 0, F_\Delta(u) \approx u^{-a}$, kai $u \to \infty$. Palyginimui, Wiener'io proceso W pokyčiai $W(t) - W(t-\Delta)$ turi normalųjį skirstinį, kurio uodega $1 - \Phi(u/\sqrt{\Delta}) = cu^{-u^2/(2\Delta)}$ laikoma lengva. Vienas pirmųjų sunkių uodegų efektą finansiniuose duomenyse pastebejo Mandelbrot'as (1963), del šios priežasties pasiūlęs vietoje $W(t)$ naudoti simetrinį α stabilųjį procesą $X_\alpha(t)$. Mat X_α pokyčių skirstiniai turi sunkias uodegas, kurioms $a = \alpha < 2$. Tačiau vėlesni tyrimai parodė, kad finansinių duomenų logaritmines grąžas geriau aproksimuoja tie skirstiniai su sunkiomis uodegomis, kurių skaičius $a \in (2,4)$. Taip pat pastebeta, kad uodegų charakteris keičiasi, pereinant nuo vieno logaritminės grąžos dažnio prie kito. Išsamiausiai kol kas ištirtos kasdienių duomenų logaritminės grąžos.

Asimetrija. To paties absoliutinio dydžio grąžas lydi nevienodo dydžio kintamumo reikšmės - kintamumas yra didesnis po neigiamos grąžos (t.y. po kainos kritimo). Tai paprastai aiškinama tuo, kad investuotojai „jautriau" reaguoja į neigiamą informaciją, nei į teigiamą informaciją. Del šios asimetrijos kovariacija tarp grąžos ir būsimų kintamumo reikšmių yra neigiama. Šis efektas dar vadinamas sverto efektu (angl. leverage effect).

Kintamumo klasterizacija. Tikėtina, kad finansinių duomenų didelio kintamumo ir mažo kintamumo periodai seka vienas kitą, t.y., stebima kintamumų klasterizacija.

Taylor'o efektas. Pačios grąžos r_t tarpusavy yra beveik nekoreliuotos, o jų absoliutinių dydžių laipsniai $|r_t|^\delta$ $(\delta > 0)$ turi nenulinę koreliaciją. Stipriausia koreliacija stebima absoliutinėms grąžoms, t. y., kai $\delta = 1$. Ši savybė pirmą kartą buvo pamineta Taylor'o (1986) ir dėl to kartais vadinama Taylor'o efektu.

Ilgalaikė atmintis. Koreliacijos tarp $|r_t|^\delta$ ir $|r_s|^\delta$ įvertis, didejant $|t-s|$, gęsta lėtai (panašiai kaip laipsnine funkcija). Tas pats teisinga ir koreliacijai tarp kintamumo įverčių. Dar sakoma, kad šie dydžiai pasižymi stipriu nuolatinumu (angl. persistency). Yra keletas hipotezių, bandančių pagrįsti grąžų kvadratų ar absoliutinių dydžių ilgalaikės atminties efektą. Daugelis jų remiasi įvairių nestacionarumų egzistavimu (trendų, šuolių buvimas ir pan., žr., pavyzdžiui, Lobato ir Savin (1998)). Vis dėlto, dar yra daug neaiškumų ir šio fenomeno paaiškinimas yra vienas aktualiausių šiuolaikinės finansų ekonometrijos uždavinių.

Suminis gausiškumas. Kuo dažnis Δ mažesnis, tuo logaritminių grąžų $r(t,\Delta)$ skirstinys (jis, apskritai kalbant, priklauso nuo Δ) darosi vis panašesnis į normalųjį skirstinį.

Vienas pagrindinių finansinių laikinių sekų analizės uždavinių yra modelių, kuo geriau atspindinčių minėtas ypatingąsias grąžų savybes, paieška.

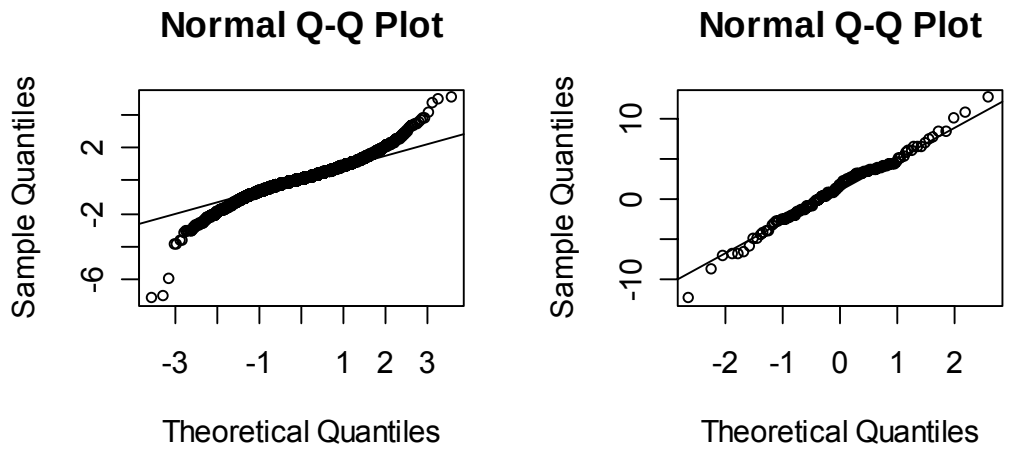
4.2.1. Sunkiosios uodegos

Nagrinėkime Standard and Poors 500 (kitaip S&P500 arba SP 500) indekso darbo dienų logaritmines grąžas $100\log(X_t / X_{t-1})$ nuo 1990 m. sausio 1 d. iki 1999 m. gruodžio ?? d.

```
library(MASS)
data(SP500)
opar=par(mfrow=c(1,2))
```

```
qqnorm(SP500)
qqline(SP500)
qqnorm(apply(matrix(SP500[1:2775],ncol=25,byrow=TRUE),1,sum))#Kodėl imame „sum“?
qqline(apply(matrix(SP500[1:2775],ncol=25,byrow=TRUE),1,sum))
par(opar)
```

Matome (žr. kvantilių grafiką 4.3 pav. kairėje), kad seka `SP500` turi žymiai daugiau didelių reikšmių, negu jų būtų Gauso (kitaip normaliuoju) atveju (tuomet sakome, kad skirstinys turi sunkias uodegas). Šis efektas labai būdingas finansinėms laikinėms sekomis, tačiau jis darosi vis mažiau pastebimas agreguojant duomenis (grafikas dešinėje) [L, 8 psl.] (duomenis čia grupavome po 25 dienas (mėnuo turi maždaug 25 darbo dienas)).

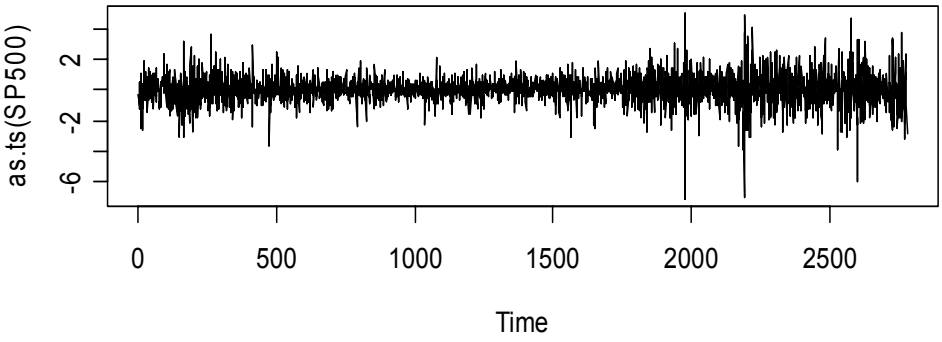


4.4 pav. Grąžų grafikai: pradinių (kairėje; jis turi daug „didelių“ reikšmių, taigi uodegos sunkios) ir agreguotų (dešinėje; imtis beveik normali)

4.11 UŽDUOTIS. Paketo `stats` duomenų rinkinyje `EuStockMarkets` yra pateikti pagrindinių Europos vertybinių popierių biržų kiekvienos 1991-1998 m. darbo dienos indeksai. Pakartokite ką tik atliktą analizę su Šveicarijos biržos indeksu `EuStockMarkets[, "SMI"]`.

4.2.2. Finansinių laikinių eilučių sklaidumas nėra pastovus

Su `plot(as.ts(SP500))` išbrėžę S&P500 indekso logaritminių grąžų grafiką (žr. 4.5 pav.), matome, kad paprastai jų sklaidumas (kitaip kintamumas – angl. volatility) nėra didelis, tačiau kartais grąžų reikšmės kinta labai audringai. Laikinės sekos, kurių besąlyginė (kitaip ilgalaikė) dispersija yra pastovi, bet yra laiko tarpų su santykinai didele dispersija, vadinamos sąlyginai heteroskedastiškomis arba ARCH (AutoRegressive Conditionally Heteroskedactic) sekomis. Jas smulkiau nagrinėsime 5 skyriuje.

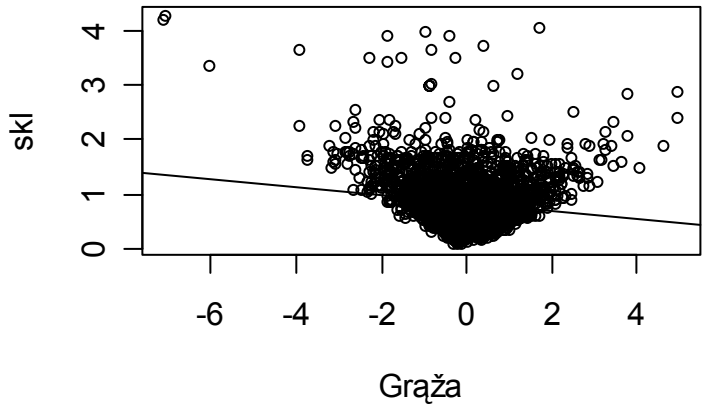


4.5 pav. S&P500 grafikas

4.2.3. Grąžų asimetrija [L, 7 psl.]

To paties modulio grąžas lydi nevienodo dydžio grąžos kintamumo reikšmės – kintamumas yra didesnis po neigiamos grąžos (t.y., po kainos kritimo). Tai paprastai aiškinama tuo, kad investuotojai jautriau reaguoja į „neigiamą“ informaciją negu į teigiamą. Šį teiginį galima pagrįsti įvairiais empiriniais faktais, štai vienas iš jų: koreliacija tarp grąžos ir būsimo sklaidumo yra neigiama (grąžai didėjant sklaidumas mažėja). Jei sklaidumą suprastume kaip penkių būsimų dienų grąžų standartinį nuokrypį, tai mūsų teiginį galėtume iliustruoti tokia programa.

```
require(MASS)
data(SP500)
x=SP500 # Kad būtų trumpiau, pasižymėsime viena raide
len=length(x)
skl=numeric(len-5)
for(i in 1:(len-5)) skl[i]=sd(x[i:(i+5)])
plot(x[1:(len-5)],skl,xlab="Grąža")
skl.lin=lm(skl~I(x[1:(len-5)]))
abline(skl.lin)
print(cor.test(x[1:(len-5)],skl))

  Pearson's product-moment correlation

data:  x[1:(len - 5)] and vol
t = -7.7913, df = 2773, p-value = 9.305e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1825775 -0.1097511 # Koreliacijos koeficientas neabejotinai neigiamas
sample estimates:
       cor
-0.1463626
```

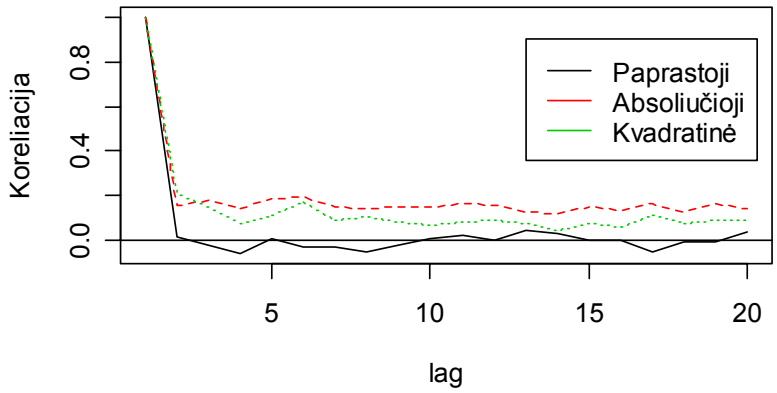


4.6 pav. Grąžos ir grąžų sklaidumo sklaidos diagrama ir regresijos tiesė

4.2.4. Taylor‘o efektas

Pačios grąžos r_t tarpusavyje yra (beveik) nekoreliuotos, tačiau jų absoliučiųjų dydžių laipsniai $|r_t|^\delta$ $(\delta > 0)$ turi nenulinę koreliaciją. Beje, iš čia išplaukia, kad grąžos yra (nekoreliuotos, bet) priklausomos – nepriklausomų a.d. funkcijos irgi būtų nepriklausomi a.d.

```
require(MASS)
data(SP500) # Returns of the Standard and Poors 500 Index in the 1990's
x=SP500
len=length(x)-19
corr=numeric(20)
corr1=numeric(20)
corr2=numeric(20)
   for(i in 1:20)   # Skaičiuosime koreliacijas tarp x(t) ir x(t-1),
                    # ..., x(t) ir x(t-20)
   {
   # Skaičiuosime koreliaciją tarp grąžų, jų modulių ir kvadratų
   corr[i]=cor(x[1:len],x[i:(len+(i-1))])
   corr1[i]=cor(abs(x[1:len]),abs(x[i:(len+(i-1))]))
   corr2[i]=cor(x[1:len]^2,x[i:(len+(i-1))]^2)
   }
plot(corr,xlab="lag",ylab="Koreliacija",type="l")
lines(corr1,col=2,lty=2)
lines(corr2,col=3,lty=3)
abline(0,0)
legend(13,0.9, c("Paprastoji","Absoliučioji","Kvadratinė"),lty=c(1,1,1),col =
c(1,2,3))
```



4.7 pav. Laikinės sekos SP, jos modulio ir kvadrato koreliacijos funkcijos

4.12 UŽDUOTIS. Pakartokite asimetrijos ir Taylor‘o efekto analizę su Šveicarijos biržos indeksu `EuStockMarkets[, "SMI"]`.

****

>Hi all,
>
>Does anybody know which is more commonly used in financial time series --
>log return or quotient return?
>
>Thanks a lot,
>
>M

****

I think you have the wrong question. The right
question is: Given what I'm doing, should I use
log returns or simple returns?

Since log returns are additive in time, it doesn't
stretch credibility too much to assume that the
distribution of log returns is Gaussian as the time
period gets large. With shorter periods the distribution
will be long-tailed, but is often not far from symmetric.
Hence for many modeling problems it makes sense to
use log returns.

Since simple returns are additive across assets,
it makes sense to use simple returns when going
from individual assets to a portfolio. Simple returns
are also better understood by investors.

Writing R functions to switch between the two is an
easy exercise left to the reader. Such functions should
be used as appropriate throughout a project.

Patrick Burns
****

Hi all,

In playing with the empirical finance models, we need the risk-free rate. I
am thinking of T-bill 3 month rate.

I've looked at a few webpages, e.g.

http://mortgage-x.com/general/indexes/t-bill_index_faq.asp

But they look complicated... is there a popular place that I can simply
download the T-bill 3 month historical data?

Is there a program in R that can automatically/streamingly pull stock and
T-bill rate data from popular website?

Thanks a lot!

****
2 good sources of info are

1. FRED database at the Federal Reserve bank of Kansas City.
2. www.economagic.com. They have a nice collection of free financial and
economic data.

```
*****************************

And IIRC Rmetrics has a function to access both:

TimeSeriesImport          package:fCalendar          R Documentation

Import Market Data from the Internet

Description:

     A collection and description of functions to import  financial and
     economic market data from the Internet. Download functions are
     available for economic and financial market data from
     Economagic's, from  Yahoo's, from the Federal Reserve's, and from
     the the forecasts.org Internet sites.

     The functions are:

       'economagicImport'  Economic series from Economagic's Web site,
       'yahooImport'       daily stock market data from Yahoo's Web site,
       'yahooSeries'       easy to use download from Yahoo,
       'keystatsImport'    key statistics from Yahoo's Web site,
       'fredImport'        time series from St. Louis FRED Web site,
       'forecastsImport'   monthly data from the Financial Forecast Center.


Hth, Dirk

*****************************

I use the US-fed site to get US-data from there:

library(zoo)

usfedyields<-function(mat) {
  ##from: http://www.federalreserve.gov/releases/h15/data.htm
  url<-
pas-
te("http://www.federalreserve.gov/releases/h15/data/Business day/H15 TCMNOM ",ma
t,".txt",sep="")
  raw<-read.csv(file=url,skip=7,colClasses=c("character","character"))
  date<-as.Date(raw[,1],format="%m/%d/%Y")
  yield<-as.numeric(raw[,2])
  return(zoo(yield,date))
}

y3 <-usfedyields("M3")

A more theoretical question:
Do you use the 3-month rate as the short rate? I don't know what model
you use, but if you use vasicek, CIR, some parametric model (Svensson,
...) the 3 month rate will differ from the short rate by a well
defined quantity. How do you deal with this? What do others use as
short rate?
Just tell me more! I am curious on literature as well; I just now
http://ideas.repec.org/p/wpa/wuwpfi/9808004.html

Best,
Thomas

*****************************
```

```
You could also use read.zoo.   With the same url as in the function below:

   read.zoo(url, skip = 7, header = TRUE, sep = ",", format = "%m/%d/%Y")
```

****

5. ARCH ir GARCH modeliai

5.1. ARCH procesai

Daugelis finansinių laikinių sekų y_t yra aprašomos ARIMA(0,1,0) procesu, kitaip sakant, jų grąžos[1] $r_t = \Delta y_t = y_t - y_{t-1}$ elgiasi kaip baltasis triukšmas. Tai visai priimtinas modelis, tačiau detalesnė analizė rodo, kad jį reikia patikslinti, atsižvelgiant į vadinamąsias grąžų ypatingąsias savybes (angl. stylized facts):

- dažnai po santykinai „ramių“ r_t periodų stebime „audringus“ (su dideliu r_t reikšmių sklaidumu), o juos vėl pakeičia „snudūriavimo“ laikotarpis (tai vadinama klasterizacijos (kitaip spietimosi) efektu);
- nors r_t elgiasi kaip nekoreliuoti atsitiktiniai dydžiai, tačiau r_t^2 yra smarkai koreliuoti (dažnai visos šios koreliacijos yra neneigiamos); tai reiškia, kad r_t yra nekoreliuotų, bet priklausomų dydžių seka;
- paprastai grąžų “uodegos“ yra sunkesnės negu Gauso skirstinio (taigi, lyginant su Gauso skirstiniu, grąžos turi per daug „didelių“ reikšmių).

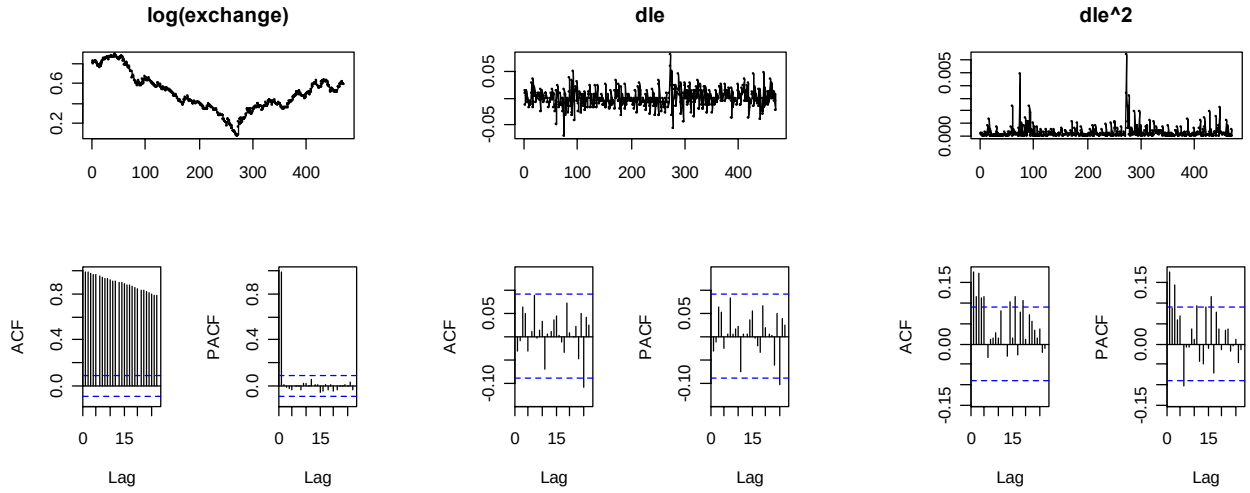
Šio skyrelio tikslas – sudaryti grąžų matematinius modelius, atspindinčius jų ypatingąsias savybes. Pasirodo, kad nors kiekviena grąžų trajektorija elgiasi gana nereguliariai, atsitiktinį gražų procesą vis dėlto galima aprašyti stacionariu modeliu.

5.1 pavyzdys. Nagrinėsime savaitinį USD\GBP kursą, 1980-1988 m. Surinkite `exchange=ts(scan(file.choose()))` ir nuvairuokite į Data/Chan/exchange.dat.

```
library(forecast)
tsdisplay(log(exchange))   # Nestacionarus procesas
dle=diff(log(exchange))    # Logaritmų skirtumų procesas
tsdisplay(dle)             # dle yra baltasis triukšmas
tsdisplay(dle^2)           # dle^2 nėra baltasis triukšmas
```

5.1 pav. matyti, kad laikinė seka $y_t = \texttt{dle}_t$ aprašoma baltuoju triukšmu (teisingiau sakant, baltąjį triukšmą sudaro centruotas procesas $r_t = y_t - \mu_t$, čia $\mu_t = \texttt{mean(dle)} = 0.00045$). Antra vertus, tai specialios struktūros baltasis triukšmas, nes jį sudaro tik nekoreliuoti, o ne nepriklausomi a.d. (jei r_t būtų nepriklausomi a.d., tai bet kokia jų funkcija, pvz., kvadratai, taip pat būtų nepriklausomi, taigi nekoreliuoti, ir todėl sudarytų baltąjį triukšmą). Ateityje skirsime baltąjį triukšmą plačiąja prasme (BT) (tai nekoreliuotų atsitiktinių dydžių (a.d.) su nuliniu vidurkiu ir pastovia dispersija seka) ir baltąjį triukšmą siaurąja prasme (NVPADS) (tai nepriklausomų vienodai pasiskirsčiusių (n.v.p.) a.d. su baigtine (ir pastovia) dispersija seka). Aišku, kad kiekviena NVPADS yra BT, todėl „siaurųjų“ baltųjų triukšmų yra mažiau. Iš grafikų aišku, kad r_t yra BT.

[1] Jei simboliu y_t žymime pradinio proceso logaritmą, tai $r_t = \Delta y_t$ yra proceso logaritminė grąža.



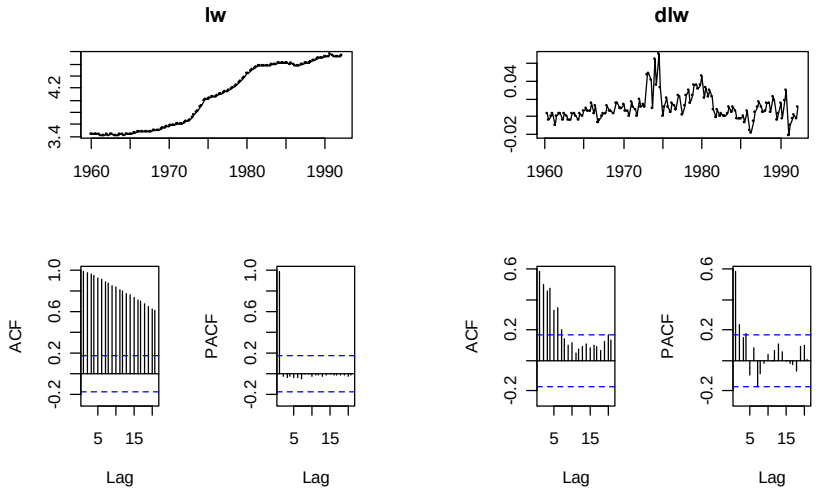
5.1 pav. `(exchange` ir) `log(exchange)` yra nestacionarus procesas su, gal būt, vienetine šaknimi (kairėje); taip ir yra – skirtumų procesas yra BT (viduryje); antra vertus, skirtumų kvadratai – jau ne (žr. dešinėje). Dešiniajame grafike matyti, kad ilgus `dle` ramybės intervalus pakeičia trumpi didelio sklaidumo periodai.

5.2 pavyzdys. [redacted] Surinkite komandą

```
wpi=ts(scan(file.choose(),skip=1),start=1960,end=1992.25,freq=4)
```

ir nuvairuokite į Data\Enders\WPI.txt – tai JAV ketvirtiniai didmeninių kainų indeksų duomenys.

```
lw=log(wpi)
tsdisplay(lw)    # Akivaizdžiai nestacionarus procesas
dlw=diff(lw)     # Vidurkis, gal būt, ir pastovus (žr. 5.2 pav. dešinėje)
tsdisplay(dlw)   # Logaritminės grąžos nėra BT (ir, tuo labiau, nėra NVPADS)
```

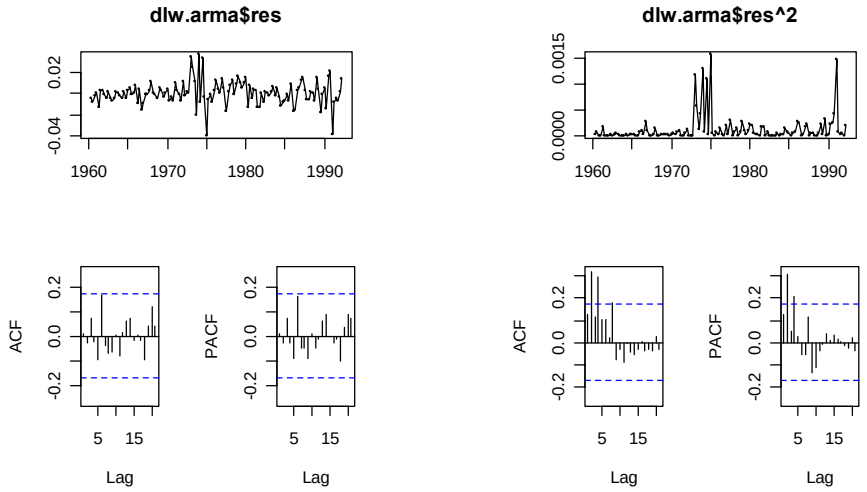


5.2 pav. `wpi` ir `dlw` (pastaroji laikinė seka nėra BT!) grafikai

Šį kartą logaritminės grąžos nesudaro BT, tačiau nesunku įsitikinti, kad tai stacionarus procesas, tiksliau kalbant, modelis su sezoniniu nariu MA(4)

```
(dlw.arma=arima(dlw,order=c(1,0,4),fixed=c(NA,NA,0,0,NA,NA)))
Coefficients:
         ar1      ma1  ma2  ma3     ma4  intercept  # intercept = dlw vidurkis
      0.7799  -0.4303    0    0  0.2888     0.0097
s.e.  0.0855   0.1058    0    0  0.1130     0.0036
sigma^2 estimated as 0.0001184:  log likelihood = 399.63,  aic = -789.26
```

visai neblogai aprašo (logaritminių) grąžų procesą (beje ketvirtinė infliacija tuo laikotarpiu buvo 0,97%, o metinė – 3,88%). Priminsime, kad žodžiai „neblogai aprašo“ reiškia, jog modelio liekanos sudaro BT (žr. 5.3 pav., grafiką kairėje)



5.3 pav. Modelio `dlw.arma` liekanos sudaro BT (grafikas kairėje), tačiau nesudaro NVPADS (žr. grafiką dešinėje). Dešiniajame grafike matyti, kad ilgus `dlw.arma$res` ramybės intervalus seka trumpi didelio sklaidumo periodai

Taigi $y_t = \texttt{dlw}_t$ yra stacionarus ARMA(1,4) procesas

$$y_t = (1-0.7799)\cdot 0.0097 + 0.7799 y_{t-1} - 0.4303 w_{t-1} + 0.2888 w_{t-4} + w_t ,$$

lygybė $\mu_t = (1-0.7799)\cdot 0.0097 + 0.7799 y_{t-1} - 0.4303 w_{t-1} + 0.2888 w_{t-4}$ apibrėžia proceso y_t (*sąlyginį*) *vidurkį* $E(y_t \mid \Omega_{t-1})$, šio modelio liekanos $r_t = y_t - \mu_t = w_t$ yra BT. ◀◀

Mes aptarėme du grąžų procesų pavyzdžius, abiem atvejais $r_t = y_t - E(y_t \mid \Omega_{t-1}) = y_t - \mu_t$ turi kelias ypatingąsias grąžų savybes. Paprastumo dėlei šį procesą vėl vadinsime (grynuoju) grąžų procesu ir pabandysime sudaryti jo modelius. Visi šie modeliai susiję su proceso *sąlyginės* dispersijos elgesiu, todėl pirmiausiai prisiminsime bendrą *sąlyginio* vidurkio apibrėžimą.

Momentu *t* prognozuojant atsitiktinio proceso ateitį $r_{t+1},...,r_{t+h},...$, paprastai remiamasi žiniomis apie ankstesnes proceso reikšmes, sukauptas vadinamojoje informacinėje aibėje $\Omega_t = \{r_t, r_{t-1}, r_{t-2},...\}$ arba, kas stacionariu atveju tas pat, $\Omega_t = \{w_t, w_{t-1}, w_{t-2},...\}$.

Sąlyginio vidurkio savybės

1. Konstantos sąlyginis vidurkis lygus jai pačiai: $E(c \mid \Omega_t) = c$.
2. Jei a.d. y yra lygus vienam iš informacinės aibės elementų, pvz., r_{t-1}, tai $E(y \mid \Omega_t) = E(r_{t-1} \mid \Omega_t) = r_{t-1}$.
3. Jei a.d. y nepriklauso nuo visų $r_t, r_{t-1},...$ (arba nuo $w_t, w_{t-1},...$), tai $E(y \mid \Omega_t) = Ey$.
4. $E(y) = E(E(y \mid \Omega_t))$.

Vienas natūralus grąžų modelis galėtų būti toks.

1. Sakykime, kad $r_t = x_{t-1} w_t$ (čia r_t yra grąžų procesas, w_t - (procesą inovuojančius) impulsus aprašanti NVPADS su pastovia dispersija σ^2, o x_t - prognozinis (nebūtinai atsitiktinis) kintamasis). Jei $x_t = x_{t-1} = x_{t-2} = ... = \text{const}$, tai $\{r_t\}$ yra NVPADS (taigi ir BT). Antra vertus, jei ne visi x_t lygūs, sąlyginė r_t dispersija $\sigma_t^2 = D(r_t \mid x_{t-1}) = x_{t-1}^2 \sigma^2$ nėra pastovi. Jei, pvz., x_t reikšmės yra teigiamai autokoreliuotos (t.y., tikėtina, kad po didelės x_t reikšmės didelė bus ir x_{t+1}), tai ir sąlyginė r_t dispersija bus tokia.

Pagrindinis šio modelio trūkumas yra tas, kad dažnai sunku rasti tinkamą prognozinį kintamąjį x_t. Dėl šios priežasties populiaresni yra kiti, vadinamieji ARCH (angl. AutoRegressive Conditional Heteroskedastic) klasės modeliai. Šiuo atveju proceso r_t dispersija priklausys ne nuo išorinio kintamojo x, bet nuo paties proceso r_t ankstesnių reikšmių.

2. **<u>Apibrėžimas</u>**. Tarkime, kad procesas r_t yra aprašomas lygtimi $r_t = \sigma_t w_t$; čia

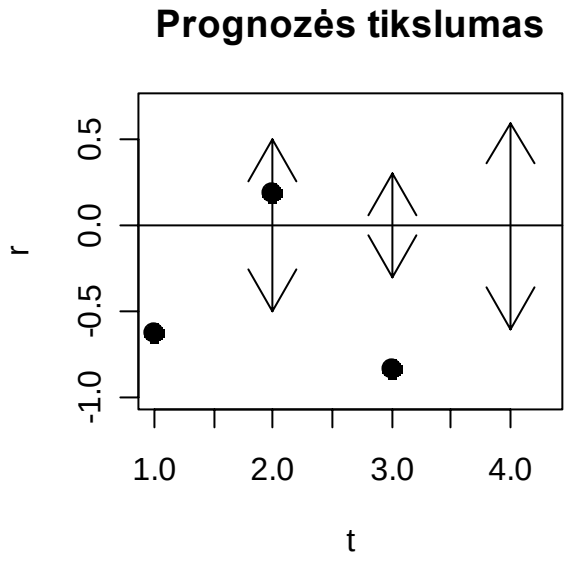
- inovacijos w_t sudaro NVPADS;
- grąžos r_t, fiksavus praeitį Ω_{t-1}, turi normalųjį skirstinį $\sigma_t \cdot \mathcal{N}(0,1)$ (5.1a)
- r_t sąlyginio skirstinio dispersija σ_t^2 priklauso nuo p ankstesnių w_t reikšmių:
  $\sigma_t^2 = D(r_t \mid \Omega_{t-1}) = \omega + \gamma_1 r_{t-1}^2 + ... + \gamma_p r_{t-p}^2$, $\omega > 0$, $\gamma_i \geq 0$, $\sum_{i=1}^{p} \gamma_i < 1$. (5.1b)

Šias sąlygas tenkinantis procesas r_t vadinamas ARCH(p) procesu. ◀◀

Nesunku įsitikinti, kad ARCH procesas sudaro nekoreliuotų a.d. seką su $Er_t \equiv 0$ ir $Dr_t \equiv \omega / (1 - \sum \gamma_i)$ (taigi tai BT). Antra vertus, nors sąlyginis vidurkis $E(r_t \mid \Omega_{t-1}) \equiv 0$ (dėl (5.1a)), tačiau sąlyginė dispersija $\sigma_t^2 = D(r_t \mid \Omega_{t-1})$ yra kintama ir aprašoma (5.1b) lygtimi. Taigi, jei kai

kurios iš paskutinių proceso reikšmių $r_{t-1}^2,...,r_{t-p}^2$ yra didelės, tai ir σ_t^2 bus didelė, kitaip sakant, daug šansų, kad r_t irgi bus didelė (vadinasi, ARCH procesas turi klasterizacijos savybę). Galima taip pat įrodyti, kad r_t besąlyginio skirstinio uodegos yra sunkesnės nei normaliojo (t.y., ARCH modeliu aprašomos grąžos r_t turi dar vieną iš ypatingųjų savybių). Taip pat pažymėsime, kad σ_t^2 yra r_t sąlyginės prognozės (kai remiamasi Ω_{t-1} duomenimis) paklaidos dispersija (ji dabar kinta-ma!).

Pats paprasčiausias ARCH tipo procesas yra ARCH(1) procesas: $r_t = \sigma_t w_t$, $\sigma_t^2 = \omega + \gamma_1 r_{t-1}^2$, $0 < \gamma_1 < 1$. Taip pat pastebėsime, kad kartais tariama, jog inovacijų iš (5.1a) skirstinys yra ne standartinis normalusis, o Student'o arba apibendrintasis paklaidų skirstinys [T1, p. 104; T2, p. 108] (R funkcijoje `garchFit` jie vadinami `conditional distribution`).



5.4 pav. ARCH(1) proceso grafikas. Laiko momentu $t = 1$ proceso reikšmė didelė, todėl σ_2^2 - irgi (ja nusakomas r_2 prognozės intervalas žymimas segmentu su strėlėmis). Momentu $t = 2$ proceso reikšmė maža, todėl ir r_3 prognozės intervalas $[-2\sigma_3, 2\sigma_3]$ – siauras ir t.t. ARCH(p), p>1, proceso volatilumas keičiasi (iš mažo į didelį ir atvirkščiai) rečiau.

Pradinių grąžų koreliacija nėra stipri, o dažniausiai jos iš viso nekoreliuotos. Dėl šios priežasties r_t sąlyginis vidurkis μ_t dažniausiai yra tiesiog proceso vidurkis (konstanta) arba koks nors paprastas stacionarus procesas (pvz., ARMA(p,q)) su, gal būt, keliais prognoziniais[2] kintamaisiais $(x_{1t},...,x_{kt})$:

$$r_t = \mu_t + w_t, \;\; \mu_t = a_0 + \sum_{i=1}^{p} a_i r_{t-i} + \sum_{i=1}^{q} b_i w_{t-i} + \sum_{i=1}^{k} c_i x_{it}; \qquad (5.2)$$

[2] Pvz., tai galėtų būti pirmadienį žymintis kintamasis (manoma, kad pirmadienis akcijų biržoje yra ypatinga diena).

čia antroji lygybė vadinama (grąžų sąlyginio) vidurkio lygtimi[3], o (5.1b) – volatilumo (kitaip – sąlyginės dispersijos) lygtimi. Mūsų analizės tikslas – nustatyti μ_t ir σ_t^2 pavidalą ir įvertinti jų parametrus.

Pastebėsime, kad visuomet vietoje bendrojo grąžų proceso galima nagrinėti grynąjį (pereikite nuo r_t prie $r_t - \mu_t$), kitaip sakant, užtenka nagrinėti lygtį $r_t = \sigma_t w_t$. ARCH tipo procesų yra labai daug, vienas nuo kito jie skiriasi σ_t^2 apibrėžimu.

(Grynasis) ARCH tipo procesas yra BT su tam tikru būdu kintančia sąlygine dispersija σ_t^2 (ji, taip pat, lygi sąlyginei vieno žingsnio prognozės paklaidos dispersijai)

Pateiksime kelių populiaresnių (grynųjų) ARCH tipo procesų apibrėžimus. Procesas $r_t = \sigma_t w_t$ vadinamas

- ARCH(p) procesu, jei

$$\sigma_t^2 = \omega + \gamma_1 r_{t-1}^2 + ... + \gamma_p r_{t-p}^2, \quad \omega > 0, \quad \gamma_i \geq 0, \sum_{i=1}^{p} \gamma_i < 1. \tag{5.3}$$

- GARCH(p,q) (G=Generalized) procesu, jei

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \ \omega > 0, \ \alpha_i \geq 0, \ \beta_i \geq 0, \ \sum \alpha_i + \sum \beta_i < 1.$$

- GARCHX(1,1) procesu[4] su egzogeniniu kintamuoju, jei

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

  (Pavyzdys. Finansų rinkos apimtis x_t dažnai pagelbsti prognozuojant volatilumą σ_t^2).

- APARCH(p,q) (Asymmetric Power ARCH) procesu, jei w_t yra bet koks a.d. su nuliniu vidurkiu ir vienetine dispersija, o

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

  čia $\delta > 0$, o $-1 < \gamma_i < 1$. Stacionarus sprendinys egzistuoja, jei $\omega > 0$ ir $\sum \alpha_i \kappa_i + \sum \beta_j < 1$; čia $\kappa_i = E(|w| + \gamma_i w)^{\delta}$. APARCH yra labai bendras modelis, jo atskiri atvejai yra

  1. Engle'o ARCH modelis ($\delta = 2, \gamma_i = 0, \beta_j = 0$)
  2. Bollerslev'o GARCH modelis ($\delta = 2, \gamma_i = 0$)
  3. Taylor'o ir Schwert'o TS-GARCH modelis ($\delta = 1, \gamma_i = 0$)
  4. Glosten'o, Jagannathan'o ir Runkle's GJR-GARCH modelis ($\delta = 2$)

[3] Toliau dažniausiai tarsime, kad vidurkis μ_t yra ARMA(0,0) procesas, tiksliau kalbant, kad $\mu_t \equiv 0$.

[4] Čia ir kitur (1,1) vien dėl patogumo.

5. Zakoian'o T-ARCH modelis ($\delta = 1$)
6. ir t.t.

R gali vertinti šio proceso parametrus (su `garchFit` funkcija), šios funkcijos sintaksė yra paaiškinta http://www.itp.phys.ethz.ch/econophysics/R/pdf/garch.pdf .

- EGARCH(1,1) (E=Exponential) procesu, jei (mes pateikiame du variantus)

$$h_t = \log\sigma_t^2 = \begin{cases} \omega + cg(r_{t-1}) + ah_{t-1} \\ \omega + \alpha\left|\frac{w_{t-1}}{\sigma_{t-1}}\right| + \gamma\frac{w_{t-1}}{\sigma_{t-1}} + ah_{t-1} \end{cases}$$

čia $g(x) = |x| + \psi x$ (šio modelio privalumas yra tas, kad šis modelis (kaip ir akcijų rinka) į šokus reaguoja asimetriškai). R neturi šio proceso parametrus vertinančių funkcijų, bet panašius modelius galima gauti, pasirinkus inovacijas su nesimetriškais tankiais (funkcija `garchFit`, opcijos `"dsged"` ir `"dsstd"`).

- TGARCH(1,1) (T=Threshold=slenkstis) procesu, jei

$$\sigma_t^2 = \omega + \alpha w_{t-1}^2 + \gamma w_{t-1}^2 D_{t-1} + \beta\sigma_{t-1}^2;$$

čia žymimasis kintamasis D_i lygus 1, jei $w_i < 0$ ir 0 priešingu atveju (dar vienas asimetrinės reakcijos modelis).

Dar vieną plačią klasę sudaro GARCH-M(1,1) (GARCH-in-mean) procesai. Šį kartą tariama, kad įprastinės tiesinės regresijos $y_t = \beta_0 + \beta_1 x_t + w_t$ paklaidas sudaro ne n.v.p.a.d., bet ARCH tipo procesas: $w_t \,|\, \Omega_{t-1} \sim \sigma_t \cdot \mathcal{N}(0,1)$, $\sigma_t^2 = \omega + \alpha w_{t-1}^2 + \beta\sigma_{t-1}^2$. Kitas variantas: $y_t = \beta_0 + \beta_1 x_t + \gamma\sigma_t^2 + w_t$. Abu šie modeliai gali būti naudingi grąžoms prognozuoti tuomet, kai rizika (kuri matuojama sąlygine dispersija) nėra pastovi.

5.2. ARCH modelio sudarymas

Grąžų proceso modelis sudaromi keturiais žingsniais[5].

1. Patikrinti, ar duomenys reikšmingai nukrypsta nuo BT; jei taip, apskaičiavus vidurkį ar sudarius paprastą ARMA tipo modelį, pašalinti sąlyginį vidurkį μ_t.
2. Jei vidurkio modelis teisingas, jo liekanos sudaro BT; ištirti, ar jis turi ARCH struktūrą. (Apibrėžimas. Sakome, kad $\{r_t - \mu_t\}$ turi ARCH struktūrą, jei tai BT, tačiau ne NVPADS; kitaip sakant, turi ARCH struktūrą, jei jo volatilumas nėra pastovus).

[5] Trumpas šių taisyklių variantas:
i) Ištirti grąžų ir jų kvadratų ACF ir PACF funkcijų grafikus
ii) Jei grąžų kvadratai nėra baltasis triukšmas (t.y., jei volatilumas nėra pastovus), sudaryti gražų sąlyginės dispersijos modelį.

3. Pasirinkus tarpinį liekanų volatilumo modelį, iš naujo sudaryti abu, vidurkio ir volatilumo, modelius (abi šios procedūros atliekamos vienu metu DT metodu).
4. Patikrinti sudarytą modelį ir, jei reikia, jį patikslinti.

Aptarsime visus žingsnius iš eilės.

1. Šis žingsnis aptartas 5.1 ir 5.2 pavyzdžiuose.

2. Bus paprasčiau, jei skirtumą $r_t - \mu_t$ (taigi `dlw.arma$res` 5.2 pavyzdyje) vėl pažymėsime simboliu r_t (kitaip sakant, tarsime, kad iš karto nagrinėjame BT). Bandysime išsiaiškinti, ar seka $\{r_t^2\}$ taip pat yra BT, tiksliau kalbant, ar seka $\{r_t\}$ yra NVPADS. Tai galima atlikti grafiškai (pvz., su funkcija `tsdisplay`, taikyta kvadratų sekai) arba skaitmeniškai, ištyrus regresijos lygtį $r_t^2 = \omega + a_1 r_{t-1}^2 + ... + a_p r_{t-p}^2 + w_t$ (ši lygtis panaši į (5.3) lygtį; skirtumas tik tas, kad dabar kairėje pusėje vietoje nestebimo dydžio σ_t^2 įrašytas r_t^2) . Grafinis `dlw.arma$res` tyrimas buvo atliktas 5.3 pav. (stebėjome ARCH efektą, kurio parametrą p dar reikės nustatyti).

Šią grafinę analizę galima papildyti skaitmenine. Tai galima padaryti keliais būdais: **1**) regresiniam modeliui $r_t^2 = \omega + \sum_{i=1}^{m} a_i r_{t-i}^2 + w_t$ patikrinti hipotezę $H_0 : a_1 = ... = a_m = 0$, *t.y.,* r_t^2 *yra baltasis triukšmas* w_t*, kitaip sakant,* r_t *nėra ARCH procesas* su alternatyva H_1 : *bent vienas koeficientas* a_i *nelygus nuliui (t.y.,* r_t *yra ARCH procesas)* (su F testu)) arba **2**) su Ljung'o ir Box'o testu; priminsime, kad šis testas tikrina hipotezę, kurią neformaliai galima užrašyti taip: H_0: *tiriamasis procesas sudaro baltąjį triukšmą* su alternatyva H_1: *yra ne taip* arba, kalbant tiksliau, hipotezę H_0: $\rho_1 = \rho_2 = ... = \rho_m = 0$ su alternatyva H_1: *bent vienas* ρ_i, $i = 1,...,m$, $\neq 0$; čia $\rho_i = corr(r_t^2, r_{t+i}^2)$. Galima įrodyti, kad tikslingiausia imti (tiksliau kalbant, testas galingiausias, kai) $m \approx 10 \log_{10} n$, o testo statistika pasirinkti sumą $Q(m) = n(n+2)\sum_{i=1}^{m} r_i^2 / (n-i)$ (Ljung'as ir Box'as įrodė, kad tuomet, kai teisinga hipotezė H_0, $Q(m)$ turi χ_m^2 skirstinį). ◀◀

Šiuos abu skaitinius metodus pritaikysime jau nagrinėtiems 5.1 ir 5.2 pavyzdžiams.

5.1 pavyzdys. Priminsime, kad tame pavyzdyje nagrinėjome savaitinio USD\GBP kurso logaritmų skirtumus `dle`. Šio skyriaus gale yra pateikta funkcija `lm.lag`, skirta regresijai su vėliniais $y_t = a_0 + a_1 y_{t-1} + ... + a_{lag} y_{t-lag} + w_t$ modeliuoti.

```
> lm.lag((dle-mean(dle))^2,4)
# Vietoje 4(=lag) galite išbandyti ir kitas reikšmes
# Nėra labai tikslių rekomendacijų dėl lag reikšmės
[...]
Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.689e-04  3.255e-05   5.189 3.18e-07 ***
embed(x, lag + 1)[, -1]1 1.439e-01  4.656e-02   3.091  0.00212 **
embed(x, lag + 1)[, -1]2 5.680e-02  4.661e-02   1.219  0.22363
embed(x, lag + 1)[, -1]3 1.371e-01  4.662e-02   2.941  0.00344 **
embed(x, lag + 1)[, -1]4 5.288e-02  4.658e-02   1.135  0.25688

Residual standard error: 0.0005425 on 460 degrees of freedom
Multiple R-Squared: 0.06162,    Adjusted R-squared: 0.05346
```

```
F-statistic: 7.551 on 4 and 460 DF,  p-value: 6.736e-06
```

Kadangi *p* reikšmė yra mažesnė už 0,05, nulinę hipotezę atmetame (t.y., `dle` turi ARCH struktūrą).

Tą patį rezultatą gauname ir su Ljung'o ir Box'o testu:

```
> Box.test(dle-mean(dle),lag=10*log(length(dle),10),type="Ljung")
[...]
X-squared = 24.857, df = 26.712, p-value = 0.5667 # dle-mean(dle) yra BT

> Box.test((dle-mean(dle))^2,lag=10*log(length(dle),10),type="Ljung")
[...]
X-squared = 76.3093, df = 26.712, p-value = 1.150e-06 # (dle-mean(dle))^2
                                                      # nėra BT
```

5.2 pavyzdys. Priminsime, kad tame pavyzdyje nagrinėjome JAV ketvirtinių didmeninių kainų indeksus. Patys indeksai nesudarė BT, tačiau ARMA(1,4) modelio liekanos `dlw.arma$res` jau buvo BT. Mus domina, ar šios liekanos turi ARCH struktūrą.

```
> lm.lag(dlw.arma$res,4)
[...]
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              0.0002223  0.0010018   0.222    0.825
embed(x, lag + 1)[, -1]1 0.0120818  0.0918240   0.132    0.896
embed(x, lag + 1)[, -1]2 -0.0317696 0.0915341  -0.347    0.729
embed(x, lag + 1)[, -1]3 0.0749032  0.0914637   0.819    0.414
embed(x, lag + 1)[, -1]4 -0.0269838 0.0917246  -0.294    0.769

Residual standard error: 0.0112 on 120 degrees of freedom
Multiple R-Squared: 0.007201,   Adjusted R-squared: -0.02589
F-statistic: 0.2176 on 4 and 120 DF,  p-value: 0.9282

> lm.lag(dlw.arma$res^2,4)
[...]
Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             5.589e-05  2.782e-05   2.009  0.04674 *
embed(x, lag + 1)[, -1]1 5.668e-02  8.925e-02   0.635  0.52662
embed(x, lag + 1)[, -1]2 2.399e-01  8.934e-02   2.685  0.00828 **
embed(x, lag + 1)[, -1]3 3.830e-02  8.935e-02   0.429  0.66892
embed(x, lag + 1)[, -1]4 2.106e-01  8.927e-02   2.360  0.01990 *

Residual standard error: 0.0002535 on 120 degrees of freedom
Multiple R-Squared: 0.151,      Adjusted R-squared: 0.1227
F-statistic: 5.334 on 4 and 120 DF,  p-value: 0.0005469
```

Matome, kad `dlw.arma$res` yra BT, tačiau `dlw.arma$res^2` – jau ne, kitaip sakant, `dlw.arma$res` turi ARCH struktūrą.

Tą patį rezultatą gauname ir su Ljung'o ir Box'o testu:

```
> Box.test(dlw.arma$res,lag=10*log(length(dlw.arma$res),10),type="Ljung")
[...]
X-squared = 14.0242, df = 21.106, p-value = 0.8723 # yra BT

> Box.test(dlw.arma$res^2,lag=10*log(length(dlw.arma$res),10),type="Ljung")
```

```
[...]
X-squared = 40.7046, df = 21.106, p-value = 0.006357 # nėra BT
```

◀◀

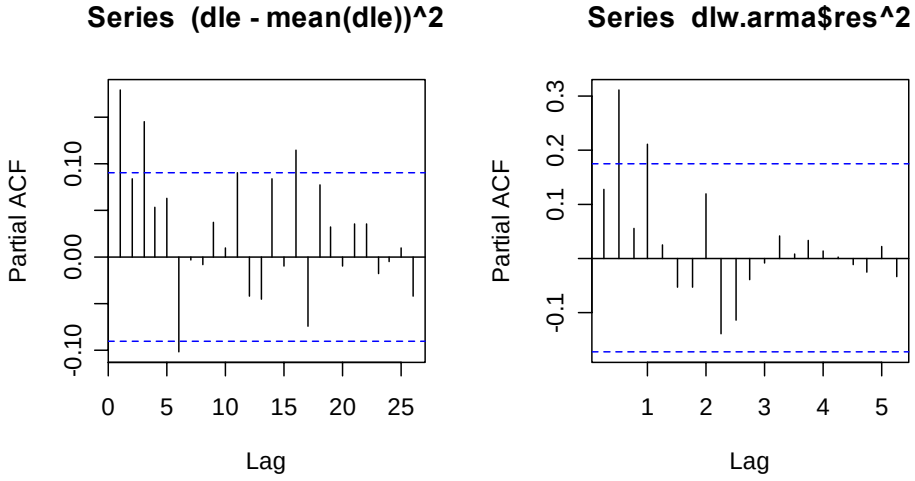
3. Nustačius, kad (grynasis) grąžų procesas turi ARCH struktūrą, toliau remiamasi tuo, kad volatilumo lygtis $\sigma_t^2 = \omega + a_1 r_{t-1}^2 + ... + a_p r_{t-p}^2$ yra labai panaši į AR(p) proceso lygtį $r_t^2 = \omega + a_1 r_{t-1}^2 + ... + a_p r_{t-p}^2 + w_t$. Prisiminę, kad AR proceso eilė yra nustatoma pagal reikšmingų stulpelių skaičių jo PACF funkcijos grafike, lygiai taip pat elgsimės ir dabar.

Grąžas aprašančio ARCH proceso eilė yra lygi reikšmingų stulpelių skaičiui grąžų kvadratų PACF funkcijos grafike

Priminsime 5.1 ir 5.2 pavyzdžių (grynųjų) grąžų kvadratų PACF grafikus (žr. 5.5 pav.). `dle` atveju reikšmingų stulpelių yra trys (arba, gal būt, šeši), o `dlw` atveju – keturi (tai ketvirtiniai duomenys). Nustatę proceso eilę, parametrų reikšmes parinksime su `garch` funkcija iš `tseries` paketo.

5.1 pavyzdys.

```
> library(tseries)
> dle.arch3=garch(dle-mean(dle),order=c(0,3)) # Vertiname ARCH(3) modelį
> summary(dle.arch3)
[...]
Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0 2.018e-04   1.443e-05   13.986  < 2e-16 ***
a1 6.605e-02   3.691e-02    1.789  0.07357 .
a2 3.077e-02   4.296e-02    0.716  0.47377
a3 1.677e-01   5.634e-02    2.978  0.00291 **
```



5.5 pav. 5.1 ir 5.2 pavyzdžių grąžų kvadratų PACF grafikai

```
Diagnostic Tests: # Tiriame, ar sudarėme gerą ARCH modelį

        Jarque Bera Test

data:  Residuals
X-squared = 100.7776, df = 2, p-value < 2.2e-16 # ARCH modelio liekanos
                                                # nėra normalios
        Box-Ljung test
```

```
data:  Squared.Residuals
X-squared = 0.3097, df = 1, p-value = 0.5779     # ARCH modelio liekanos
                                                 # yra NVPADS (gerai)
```

Matome, kad ARCH(3) modelis visai priimtinas. Patikrinsime dar šeštos eilės modelį.

```
> dle.arch6=garch(dle-mean(dle),order=c(0,6))
> summary(dle.arch6)
[...]
Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0 1.999e-04   2.146e-05    9.313   <2e-16 ***
a1 5.692e-02   3.342e-02    1.704   0.0885 .
a2 5.000e-02   5.480e-02    0.912   0.3615
a3 6.810e-02   4.463e-02    1.526   0.1270
a4 5.520e-02   5.751e-02    0.960   0.3372
a5 3.126e-02   5.254e-02    0.595   0.5518
a6 1.006e-14   4.814e-02 2.09e-13   1.0000
```

Matome, kad tai prastas modelis (beveik visi koeficientai nereikšmingi). Tai patvirtina ir AIC reikšmės:

```
> AIC(dle.arch3)
[1] -2510.947       # AIC reikšmė mažesnė
> AIC(dle.arch6)
[1] -2485.174
```

Taigi renkamės ARCH(3) modelį (jį dar galima smulkiai ištirti su `plot(dle.arch3)` funkcija).

Anksčiau minėjome, kad sudarius tarpinį modelį, galima sudaryti vidurkio ir volatilumo modelius vienu kartu. Tai galima atlikti su `garchFit` funkcija iš `fGarch` paketo. Deja, dirbti su šia funkcija nėra lengva, nes jos reikšmė yra S4 objektas (o ne S3 objektas, koks buvo visų iki šiol vartotų funkcijų atveju). Konkrečiai kalbant, sukurtas objektas dabar turi ne komponentes `$`, o slotus `@` (liet. padėtis, pozicija hierarchinėje sistemoje):

```
library(fGarch)
dle.ARCH3=garchFit(formula=~arma(0,0)+~garch(3,0),data = dle)
# Čia ~arma(0,0) yra (sąlyginio) vidurkio formulė,
# o garch(3,0) - (sąlyginės) dispersijos ARCH(3) modelio formulė
# Norėdami rasti modelio koeficientus, išbandykite komandą
# slotNames(dle.ARCH3) - pamatysite visų slotų vardus
# dle.ARCH3@fit # parametrų vertinimo rezultatai
dle.ARCH3@fit$matcoef
[...]
            Estimate   Std. Error    t value   Pr(>|t|)
mu     -0.0008705334 7.449339e-04 -1.1686049 0.24256286
omega   0.0002009387 2.108317e-05  9.5307647 0.00000000
alpha1  0.0696929345 4.031033e-02  1.7289098 0.08382523
alpha2  0.0277007396 4.793611e-02  0.5778679 0.56335333
alpha3  0.1670895253 6.654469e-02  2.5109369 0.01204112
```

arba, kas paprasčiau,

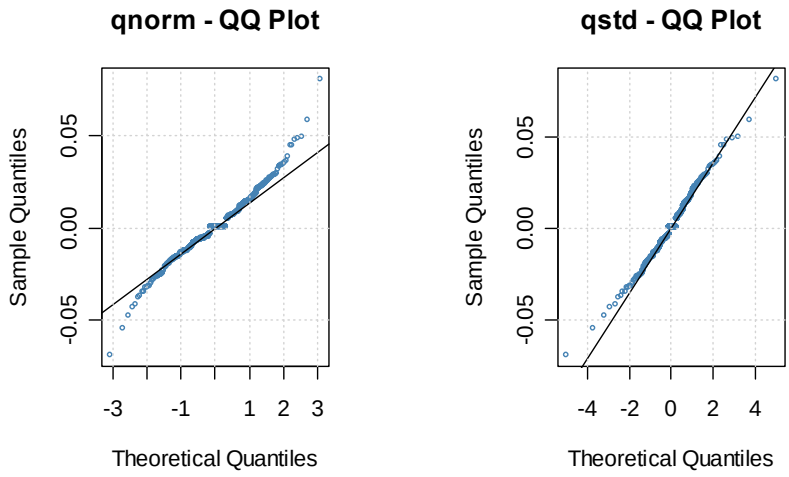
```
> summary(dle.ARCH3)
[...]
```

```
Conditional Distribution:
 dnorm

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -8.705e-04   7.449e-04   -1.169   0.2426
omega   2.009e-04   2.108e-05    9.531   <2e-16 ***
alpha1  6.969e-02   4.031e-02    1.729   0.0838 .
alpha2  2.770e-02   4.794e-02    0.578   0.5634
alpha3  1.671e-01   6.654e-02    2.511   0.0120 *

Standardized Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  102.6959  0
 Shapiro-Wilk Test  R    W      0.9778923 1.497607e-06
 Ljung-Box Test     R    Q(10)  7.810455  0.6473453
 Ljung-Box Test     R    Q(15)  10.44127  0.7911034
 Ljung-Box Test     R    Q(20)  12.40882  0.9012863
 Ljung-Box Test     R^2  Q(10)  4.470561  0.9236331
 Ljung-Box Test     R^2  Q(15)  10.15952  0.8095872
 Ljung-Box Test     R^2  Q(20)  24.08569  0.2386692
 LM Arch Test       R    TR^2   5.637804  0.933229

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
5.430175 5.474425 5.429951 5.447586
```

Abu, `garch` ir `garchFit`, funkcijų Jarque-Bera testai tvirtina, kad liekanos w_t nėra normaliosios, tą patį teigia ir 5.6 pav. (žr. žemiau), todėl ARCH koeficientus skaičiuosime iš naujo, tarę, kad liekanos turi Student'o skirstinį:

```
dle.ARCH3.st=garchFit(formula=~arma(0,0)+garch(3,0),cond.dist="dstd",data=dle)
summary(dle.ARCH3.st)
Conditional Distribution:
 dstd

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -1.226e-03   6.953e-04   -1.764 0.077774 .
omega   1.861e-04   2.665e-05    6.982 2.90e-12 ***
alpha1  7.968e-02   5.301e-02    1.503 0.132857
alpha2  7.048e-02   6.799e-02    1.037 0.299917
alpha3  1.794e-01   7.778e-02    2.307 0.021077 *
shape   6.575e+00   1.890e+00    3.479 0.000504 ***

Standardized Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  125.9182  0           # Liekanos nėra norm.,
 Shapiro-Wilk Test  R    W      0.9761292 6.017482e-07 # bet dabar jos ir
 Ljung-Box Test     R    Q(10)  7.65669   0.6623306   # neturi būti tokios
 Ljung-Box Test     R    Q(15)  10.17650  0.8084928
 Ljung-Box Test     R    Q(20)  12.11374  0.9121056
 Ljung-Box Test     R^2  Q(10)  3.750462  0.9579027
 Ljung-Box Test     R^2  Q(15)  7.983495  0.9244427
 Ljung-Box Test     R^2  Q(20)  20.73737  0.4127332
 LM Arch Test       R    TR^2   4.501894  0.9725866

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
5.485200 5.538299 5.484878 5.506092
```

Palyginus (su `plot(dle.ARCH3)` ir `plot(dle.ARCH3.st)` [6]) liekanų grafikus, aišku, kad Student'o skirstinys tinka geriau.



5.6 pav. Kairiame grafike matyti, kad, parametrus skaičiuojant su normalumo opcija, modelio liekanos nėra normalios (jų uodegos sunkesnės už normaliąsias); dešiniajame grafike empiriniai kvantiliai neblogai sutampa su teoriniais Studento kvantiliais (taškai yra (beveik) ant tiesės), todėl renkamės Student'o `cond. distribution`

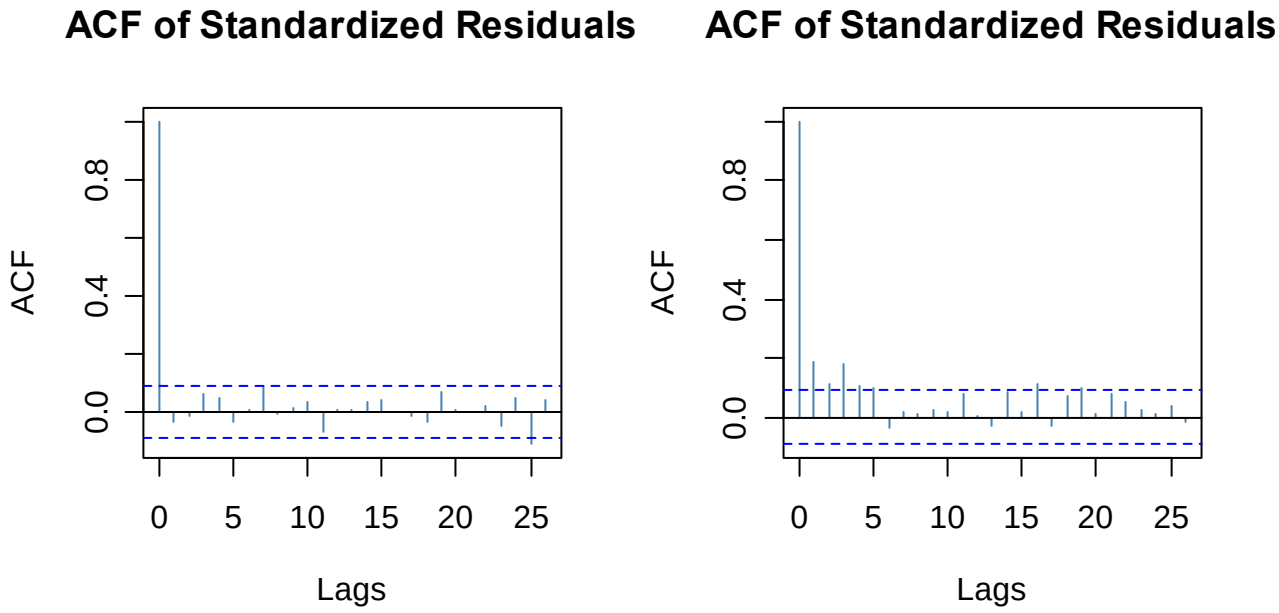
Galutinį modelį sudarysime pagal Student'o (su `(shape=)` 6.575 laisvės laipsniais) variantą:

$$dle_t = \mu + r_t = -0.001226 + r_t$$
$$r_t = \sigma_t w_t$$
$$\sigma_t^2 = 0.000186 + 0.0797 r_{t-1}^2 + 0.0705 r_{t-2}^2 + 0.1794 r_{t-3}^2$$

Ar tai geras modelis?

Tinkamai parinkus modelį, *standartinės paklaidos* r_t / σ_t turėtų sudaryti NVPADS. Tai galima patikrinti su Ljung'o ir Box'o testu arba grafiškai su `par(mfrow=c(1,2)); plot(dle.ARCH3.st, which=10); plot(dle.ARCH3.st, which=11)`.

[6] Kai funkcija `plot` prašo „Make a plot selection", pasirinkite skaičių 13.

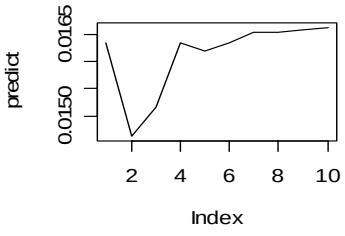


5.7 pav. Santykiai r_t / σ_t sudaro BT (kairėje), tačiau $(r_t / \sigma_t)^2$ - jau ne (dešinėje). Taigi standartinės paklaidos dar turi ARCH struktūrą, modelį reikėtų patikslinti

Prognozė. Vienas pagrindinių analizės tikslų yra `dle` prognozė. Funkcija `predict` prognozuoja 10 laiko momentų (šį kartą savaičių) į priekį.

```
> predict(dle.ARCH3.st)
```

	meanForecast	meanError	standardDeviation
1	-0.001226382	0.01662519	0.01635290
2	-0.001226382	0.01662519	0.01462719
3	-0.001226382	0.01662519	0.01517206
4	-0.001226382	0.01662519	0.01635484
5	-0.001226382	0.01662519	0.01618655
6	-0.001226382	0.01662519	0.01634348
7	-0.001226382	0.01662519	0.01654752
8	-0.001226382	0.01662519	0.01654486
9	-0.001226382	0.01662519	0.01658656
10	-0.001226382	0.01662519	0.01662595



Kadangi `dle` yra BT, todėl, kaip ir reikėjo tikėtis, vidurkio prognozė tapatingai lygi proceso vidurkiui μ = `mu=-1.226e-03`. Šaknies iš volatilumo prognozė `standardDeviation` gana greitai artėja į `dle` besąlyginį standartinį nuokrypį `sqrt(0.0001860819/(1-0.0796773883-0.0704799011-0.1794140446))` = `0.01666003`. Įdomu tai, kad pagal mūsų modelį, dviejų dienų prognozės dispersija yra pastebimai mažesnė negu kitomis dienomis.

5.2 pavyzdys. Panašiai elgdamiesi, sudarysime `dlw` modelį.

```
> dlw.arch4=garch(dlw.arma$res,order=c(0,4))
> summary(dlw.arch4)
[...]
Coefficient(s):
```

```
    Estimate  Std. Error  t value Pr(>|t|)
a0 2.702e-05   1.521e-05    1.777  0.07562 .
a1 1.962e-01   1.271e-01    1.543  0.12273
a2 2.202e-01   1.486e-01    1.482  0.13835
a3 7.901e-02   1.203e-01    0.657  0.51124
a4 5.165e-01   1.706e-01    3.028  0.00247 **

        Box-Ljung test

data:  Squared.Residuals
X-squared = 0.0451, df = 1, p-value = 0.8318

> dlw.ARCH4=garchFit(formula=~arma(1,4)+~garch(4,0),data = dlw)
> round(dlw.ARCH4@fit$matcoef,4)

        Estimate  Std. Error  t value Pr(>|t|)
mu        0.0021      0.0007   3.1487   0.0016
ar1       0.6004      0.1364   4.4006   0.0000
ma1      -0.1094      0.1415  -0.7731   0.4395
ma2      -0.0895      0.1170  -0.7646   0.4445
ma3       0.0235      0.0659   0.3566   0.7214
ma4       0.3198      0.0910   3.5150   0.0004
omega     0.0000      0.0000   2.9161   0.0035
alpha1    0.0958      0.0840   1.1402   0.2542
alpha2    0.2818      0.0895   3.1487   0.0016
alpha3    0.0000         NaN      NaN      NaN
alpha4    0.4015      0.1998   2.0099   0.0444
```

Ko gero, geresnį modelį gauname dviejų žingsnių procedūra su `garch` funkcija (atrodo, kad `garchFit` neleidžia pasirinkti ARMA modelio su nuliniais koeficientų apribojimais). Taigi

$$dlw_t = \mu + r_t = 22.59415 + 0.7799 dlw_{t-1} - 0.4303 w_{t-1} + 0.2888 w_{t-4} + r_t,$$
$$r_t = \sigma_t w_t,$$
$$\sigma_t^2 = 0.000027 + 0.1962 r_{t-1}^2 + 0.2202 r_{t-2}^2 + 0.0790 r_{t-3}^2 + 0.5165 r_{t-4}^2.$$

Kai kuriuos grafinės diagnostikos grafikus galima išbrėžti su `plot(dlw.arch4)`.

5.1 UŽDUOTIS. Modeliuokite ARCH(3) procesą $r_t = \sigma_t u_t$, $\sigma_t^2 = 10^{-6} + 0,1 r_{t-1}^2 + 0,2 r_{t-2}^2 + 0,6 r_{t-3}^2$ (su `library(fGarch); set.seed(5); rr=garchSim(model=list(alpha=c(0.1,0.2,0.6)),n=400)`) ir sudarykite tinkamą `rr` modelį.

5.2 UŽDUOTIS. [redacted] Iš ...\DATA\Tsay_fts2 importuokite duomenų rinkinį m-intc7303.txt (tai mėnesinės Intel firmos logaritminės grąžos nuo 1973 m. sausio iki 2003 m. gruodžio). Šiems duomenims sudarykite tinkamą ARCH modelį.

5.3 UŽDUOTIS. [redacted] Iš ...\DATA\Tsay_fts2 importuokite duomenų rinkinį exch-perc.txt (tai kas 10 minučių registruotos markės ir dolerio kurso logaritminės grąžos). Šiems duomenims sudarykite tinkamą ARCH modelį. ◀◀

Priminsime: „Pats paprasčiausias ARCH tipo procesas yra ARCH(1) procesas: $r_t = \sigma_t w_t$, $\sigma_t^2 = \omega + \gamma_1 r_{t-1}^2$, $0 < \gamma_1 < 1$. Taip pat pastebėsime, kad kartais tariama, jog inovacijų iš (5.1a) skirs-

tinys yra ne standartinis normalusis, o Student'o arba apibendrintasis paklaidų skirstinys [T1, p. 104; T2, p. 108] (R funkcijoje `garchFit` jie vadinami `conditional distribution`)."

5.4 UŽDUOTIS. Remdamiesi formule $r_t = (\omega + \gamma_1 r_{t-1}^2 + \gamma_2 r_{t-2}^2)^{1/2} w_t$, generuokite[7] ARCH(2) proceso su parametrais $\omega = 10^{-6}$, $\gamma_1 = 0.2$, $\gamma_2 = 0.6$ penkis šimtus reikšmių. Tarkite, kad inovacijos w_t turi Student'o su 3 laisvės laipsniais skirstinį. Pradinių reikšmių įtakai pašalinti, nagrinėkite tik reikšmes nuo 101 iki 500. Joms sudarykite tinkamą modelį.

5.3. GARCH modelio sudarymas

Finansines laikines sekas aprašantys ARCH procesai paprastai būna aukštos eilės. Antra vertus, šiuos procesus dažnai sėkmingai aprašo paprastas GARCH(1,1) procesas.

5.3 pavyzdys. Surinkite `MvsP=ts(read.table(file.choose()))` ir nuvairuokite į `Data\Misc\DEMvsGBP.dat` – ten pateikti Vokietijos markės ir Didžiosios Britanijos svaro kurso kasdienių logaritminių grąžų $100 \cdot (\ln(P_t) - \ln(P_{t-1}))$ reikšmės[8] (smulkiau - faile `Data\Misc\DEMvsGBP.txt`).

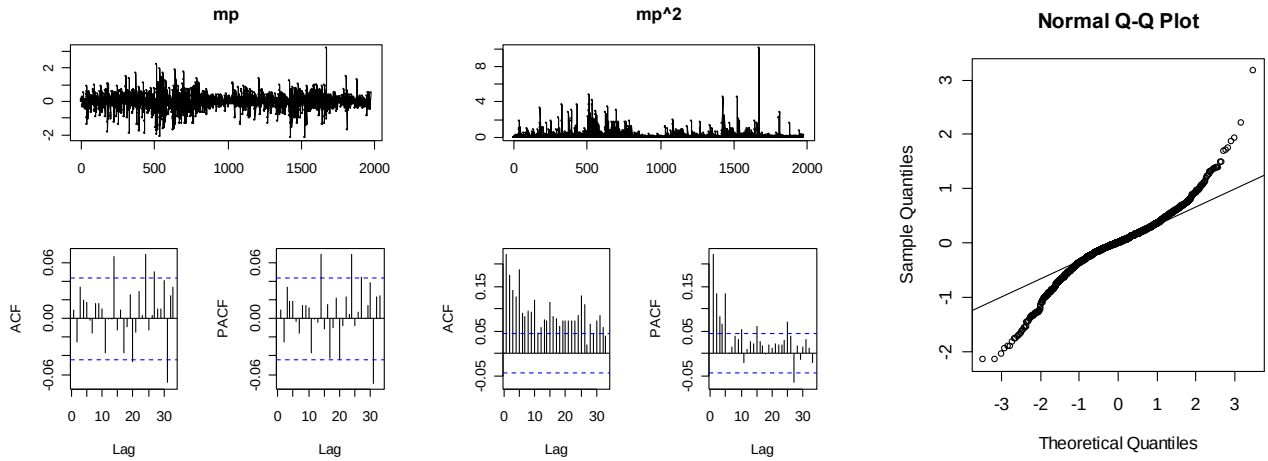
```
mp=MvsP[,1]          # MarkPound
dv=MvsP[,2]          # DummyVariable
tsdisplay(mp)        # Panašu į baltąjį triukšmą
```

(su `mp.a=arima(mp);tsdiag(mp.a)` nesunku įsitikinti, kad `mp` iš tikro yra baltasis triukšmas).

```
tsdisplay(mp^2)   # Aukštos eilės - gal ARCH(5) - arba GARCH procesas
```

Jei ARCH elgiasi panašiai kaip AR procesas (jo eilę nustatydavome pagal `mp^2` PACF grafiko reikšmingų stulpelių skaičių), tai GARCH elgiasi panašiai kaip ARMA – kadangi `mp^2` ACF ir PACF grafikai gęsta eksponentiškai, pirmiausiai išbandysime ARCH(5), o paskui GARCH(1,1) modelius.

[7] Šį procesą galima generuoti ir su `garchSim` funkcija, tačiau užduotis reikalauja parašyti tam tinkamą R programą.
[8] UŽDUOTIS. Išbrėžkite kurso P_t grafiką.



5.8 pav. `mp` yra baltasis triukšmas (kairėje), `mp^2` panašus į ARMA(1,1) procesą (viduryje); `mp` uodegos žymiai sunkesnės už normaliąsias (dešinėje)

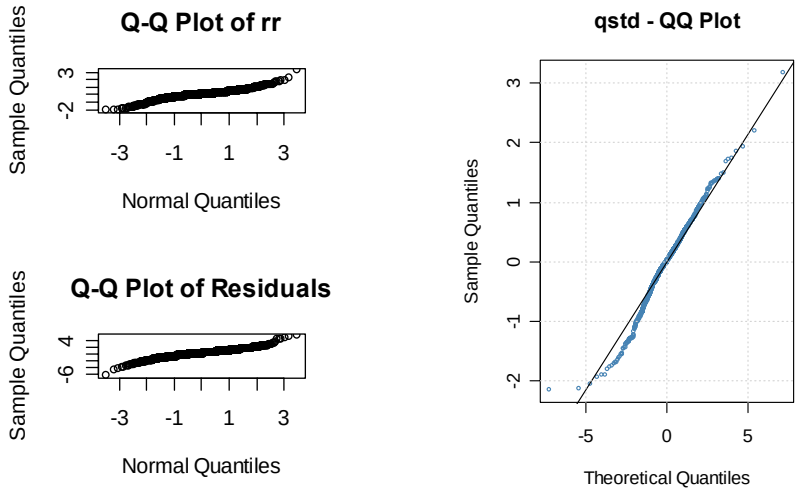
Pabandysime nustatyti proceso `mp` tipą. Tiksliau kalbant, kadangi nagrinėjamasis procesas $r_t = \sigma_t w_t$ privalo turėti nulinį vidurkį, iš `mp` atimsime jo vidurkį – pvz., taip:

```
> rr (= mp-mean(mp)) = lm(mp~1)$res    # Kodėl?
> rr.arch5=garch(rr,order=c(0,5))      # garch iš tseries paketo
> summary(rr.arch5)
[...]

Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0  0.079682    0.003984   20.001  < 2e-16 ***
a1  0.244400    0.022382   10.919  < 2e-16 ***
a2  0.154597    0.023688    6.526 6.74e-11 ***
a3  0.088706    0.023443    3.784 0.000154 ***
a4  0.078264    0.018121    4.319 1.57e-05 ***
a5  0.122537    0.023099    5.305 1.13e-07 ***

Diagnostic Tests:
        Jarque Bera Test

data:  Residuals
X-squared = 808.3238, df = 2, p-value < 2.2e-16  # Liekanos nėra normaliosios

        Box-Ljung test

data:  Squared.Residuals
X-squared = 0.063, df = 1, p-value = 0.8018      # Liekanos yra NVPADS

> AIC(rr.arch5) # 2249.605
> AIC(rr.arch6) # 2244.803 - geriausias AIC prasme tarp ARCH modelių
```

Tarp ARCH procesų geriausias (AIC prasme) yra 6 eilės modelis `rr.arch6`. Dabar sudarysime GARCH modelį.

```
> rr.garch11=garch(rr,order=c(1,1))
> summary(rr.garch11)
[...]
```

```
Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0  0.010550    0.001271    8.299   <2e-16 ***
a1  0.150958    0.013664   11.048   <2e-16 ***
b1  0.808929    0.015893   50.899   <2e-16 ***

Diagnostic Tests:
        Jarque Bera Test

data:  Residuals
X-squared = 1058.44, df = 2, p-value < 2.2e-16 # Liekanos nėra normalios


        Box-Ljung test

data:  Squared.Residuals
X-squared = 2.5497, df = 1, p-value = 0.1103

> AIC(rr.garch11) # 2220.217 - geriausias iš nagrinėtų modelių AIC prasme
> plot(rr.garch11)
```



5.9 pav. `rr.garch11` (standartinės?) liekanos nėra normalios (kairėje, apačioje), todėl išbandysime Student'o variantą ; `rr.GARCH11.st` grafike matyti, kad standartinių modelio paklaidų w_t skirstinys panašus į Student'o

Kadangi `rr.garch11` (standartinės) liekanos nėra normalios, išbandysime Student'o variantą.

```
> rr.GARCH11.st=garchFit(formula=~arma(0,0)+~garch(1,1), cond.dist="dstd",
trace=FALSE, data=mp)
> summary(rr.GARCH11.st)
[...]
Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002249    0.006955    0.323   0.7465
omega   0.002319    0.001167    1.987   0.0469 *
alpha1  0.124438    0.026958    4.616 3.91e-06 ***
beta1   0.884653    0.023517   37.617  < 2e-16 ***
shape   4.118422    0.401184   10.266  < 2e-16 ***
```

```
Standardized Residuals Tests:
                                Statistic p-Value
Jarque-Bera Test   R    Chi^2  1866.03   0          # Liekanos nėra normalios
Shapiro-Wilk Test  R    W      0.9505102 0
Ljung-Box Test     R    Q(10)  9.731061  0.4643966 # Liekanos sudaro BT
Ljung-Box Test     R    Q(15)  15.44447  0.4199002
Ljung-Box Test     R    Q(20)  17.70577  0.6067839
Ljung-Box Test     R^2  Q(10)  11.66537  0.3080764 # Liekanų kvadratai irgi BT
Ljung-Box Test     R^2  Q(15)  18.07763  0.2586062
Ljung-Box Test     R^2  Q(20)  22.31797  0.3235088
LM Arch Test       R    TR^2   13.6942   0.3206608 # Nėra pagrindo atmesti hi-
                                                   # potezę H0 : a1 = ... = am = 0

Information Criterion Statistics:
       AIC        BIC        SIC       HQIC
-0.9973742 -0.9832207 -0.9973870 -0.9921739

> plot(rr.GARCH11.st)
```

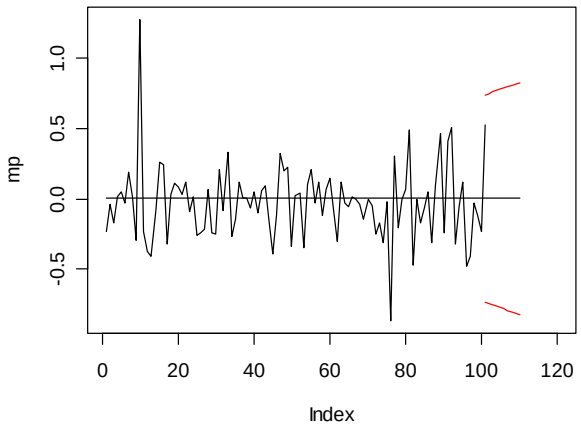
Student'o modelio `rr.GARCH11.st` liekanų grafike (žr. 5.9 pav. dešinėje) matyti, kad šis modelis tinkamiau aprašo paklaidų skirstinį. Taigi, `mp` aprašysime tokiu modeliu:

$$mp_t = 0.0022 + rr_t,$$
$$rr_t = \sigma_t w_t,$$
$$\sigma_t^2 = 0.0023 + 0.1244 rr_t + 0.8847 \sigma_{t-1}^2;$$

čia w_t turi Student'o skirstinį su (maždaug) `(shape≈)` 4 laisvės laipsniais.

`mp` prognozę 10-čiai dienų į priekį galima išbrėžti šitaip:

```
plot(mp[(length(mp)-100):length(mp)],xlim=c(1,120),ylab="mp",type="l")
lines(1:110,rep(0.0022,110))
lines(101:110,0.0022+2*predict(rr.GARCH11.st)$st,col=2)
lines(101:110,0.0022-2*predict(rr.GARCH11.st)$st,col=2)
```



5.10 pav. Vaizdumo dėlei brėžiame tik paskutines 100 `mp` reikšmių; ateinančioms 10-čiai dienų prognozuojame volatilumo didėjimą

5.5 UŽDUOTIS. Žemiau pateikti dirbtiniai duomenys `aa` (= aR_aRCH), generuoti pagal AR(1) dėsnį $y_t = 0,9y_{t-1} + e_t$ su paklaidomis e_t, sudarančiomis baltąjį triukšmą su papildoma ARCH(1) struktūra, nusakoma lygtimi $e_t = w_t(1 + 0,8e_{t-1}^2)^{1/2}$.

```
aa=structure(c(1.273, 0.2211, 0.5709, 0.2239, 1.56, 1.58, -1.627,
0.006158, 1.958, 0.3669, 1.122, -0.3134, -0.1258, -0.175, 0.7166,
-0.1395, -0.734, -0.04679, -0.09618, -0.1304, -0.2127, 0.4233,
-1.103, -0.3806, -0.7642, 0.6288, -0.619, -0.3402, -0.07623,
0.2941, 0.1112, -0.05489, 0.9988, -0.5094, -0.9521, -0.09007,
1.43, 1.722, 0.9146, -1.945, -0.09325, 0.8117, -0.3245, 1.092,
0.2093, 2.205, 2.316, 2.942, 1.732, 1.007, -0.3609, -1.781, -1.087,
-2.95, 0.9718, 1.883, -0.2679, -2.1, -0.686, -1.923, -4.39, -7.386,
-9.994, -10.77, -10.8, -9.8, -9.178, -7.006, -5.825, -6.793,
-7.697, -5.696, -4.589, -4.309, -3.43, -2.124, -0.9493, 1.136,
-0.8274, -3.028, -4.268, 0.5635, 0.4701, -0.8926, -0.5965, 1.418,
6.336, 11.84, 15.15, 14.33, 11.7, 11.2, 10.47, 7.722, 10.28,
6.087, 7.113, 8.623, 6.66, 6.305), .Tsp = c(1, 100, 1), class = "ts")
```

1. Ar `aa` proceso `tsdisplay` grafikas suderinamas su faktu, kad tai AR(1) procesas? **2**. Taikydami `arima` funkciją, gaukite tokį `aa` proceso modelį:

```
(ar.aa=arima(aa, ...)) # Vietoje daugtaškių įrašykite reikalingą tekstą
Coefficients:
         ar1
      0.9357
s.e.  0.0329
sigma^2 estimated as 2.913:  log likelihood = -196.39,  aic = 396.7
```

3. Ar `ar.aa` modelio likučiai `e=ar.aa$res` sudaro baltąjį triukšmą? O likučių kvadratai? **4**. Su `lm.lag` sudarykite modelį $e_t^2 = \omega + a_1 e_{t-1}^2$. Koks hipotezės apie likučių ARCH(1) struktūrą, t.y., $H_0 : a_1 = 0$, likimas? **5**. Su `garch` funkcija patikrinkite modelius

```
summary((e1=garch(e,order=c(1,0)))) # Dar kartą patikrinsime H0
summary(garch(e,order=c(4,0)))
summary(garch(e,order=c(1,1)))
```

Žodžiais įvardinkite šiuos modelius. Kuris jų tinkamiausias? **6**. Su `garch` funkcija raskite lygties $r_t = \sigma_t w_t$, $\sigma_t^2 = \omega + a_1 w_{t-1}^2$ koeficiento a_1 įvertį. **7**. Išbrėžkite `aa` proceso prognozę `aa.fit=aa-ar.aa$res` su $\pm$ modelio `e1` vieno sąlyginio standarto paklaidos juosta. 8. Atlikite panašią analizę su garchFit funkcija. ◀◀

5.4. TARCH modelio sudarymas

Pakete `tsDyn` yra funkcija `tarch`, kuri vertina TARCH(m) modelio (čia užrašysime jo paprastesnį variantą)

$$\sigma_t^2 = b_{0,0} + \sum_{j=1}^{m} b_{0,j}\sigma_{t-j}^2 \cdot \mathbf{1}_{r_t \le 0} + (b_{1,0} + \sum_{j=1}^{m} b_{1,j}\sigma_{t-j}^2) \cdot \mathbf{1}_{r_t > 0}$$

parametrus.

5.4 pavyzdys. Išnagrinėsime dirbtinį pavyzdį

```
n <- 1100
a <- c(0.1, 0.5, 0.2)  # ARCH(2) coefficients
e <- rnorm(n)
x <- double(n)
x[1:2] <- rnorm(2, sd = sqrt(a[1]/(1.0-a[2]-a[3])))
for(i in 3:n)  # Generate ARCH(2) process
{
   x[i] <- e[i]*sqrt(a[1]+a[2]*x[i-1]^2+a[3]*x[i-2]^2)
}
x <- ts(x[101:1100])

x.tarch <- tarch(x, m=2)
summary(x.tarch)

Estimated coefficients:
      Estimate  Std. Error  t value Pr(>|t|)
b0.0  0.091733    0.008271   11.090  < 2e-16 ***
b0.1  0.619998    0.069892    8.871  < 2e-16 ***
b0.2  0.225114    0.052712    4.271 1.95e-05 ***
b1.0  0.089128    0.008530   10.449  < 2e-16 ***
b1.1  0.641516    0.066550    9.640  < 2e-16 ***
b1.2  0.153870    0.038811    3.965 7.35e-05 ***
```

Taigi, nežiūrint to, kad generavome procesą su simetrinėmis inovacijomis (kurioje programos vietoje jos generuojamos?), mūsų procedūra nustatė asimetrišką sistemos reakciją. Gaila.

5.5. EViews‘o programa

EViews 6 yra ekonometrinei analizei skirtas komercinis produktas. Išvardinsime ARCH tipo modelius, kuriuos analizuoti ir kurių parametrus vertinti galima su EViews‘u:

- IGARCH
- GARCH-M
- GARCHX
- TARCH
- EGARCH
- PARCH

5.6. UŽDUOTYS

5.6 UŽDUOTIS. `Ecdat` pakete yra du duomenų rinkiniai: `CRSPday` ir `CRSPmon`. Pirmame iš jų kelių firmų kasdienės grąžos (iš viso 2528 dienos), o antrame – tų pačių firmų mėnesinės grąžos (iš viso 360 mėnesių). Sudarykite tinkamus ARCH tipo modelius `ge` kasdienėms ir mėnesinėms grąžoms.

5.7 UŽDUOTIS. `Ecdat` pakete yra du duomenų rinkiniai: `CRSPday` ir `CRSPmon`. Pirmame iš jų kelių firmų kasdienės grąžos (iš viso 2528 dienos), o antrame – tų pačių firmų mėnesinės grąžos (iš viso 360 mėnesių). Sudarykite tinkamus ARCH tipo modelius `ibm` kasdienėms ir mėnesinėms grąžoms.

PRIEDAS

Funkcija `lm.lag`

R regresinių modelių sintaksė nėra patogi tiesiogiai modeliuoti regresiją su vėluojančiais kintamaisiais, todėl patys parašysime tam skirtą funkciją[9]:

```
lm.lag = function(y,lag=1,x=y) summary(lm(embed(y,lag + 1)[,1] ~ embed(x,lag + 1)[,-1]))
```

(ji apskaičiuoja regresijos $y_t = a_0 + a_1 y_{t-1} + ... + a_{lag} y_{t-lag} + w_t$ arba $y_t = c_0 + c_1 x_{t-1} + ... + c_{lag} x_{t-lag} + w_t$ koeficientus). Štai vienas pavyzdys.

```
set.seed(7)
# Generuosime AR(3) procesą (t.y, ...(prisiminkite apibrėžimą))
ar3=arima.sim(n = 263, list(ar = c(0.486, -0.4858, 0.765)))
lm.lag(ar3,3) # ar3 proceso koef. apskaičiuosime maž. kvadratų metodu
[...]
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      0.02598    0.06327   0.411    0.682
embed(x, lag + 1)[, 2:(lag + 1)]1  0.48503    0.04192  11.569   <2e-16 ***
embed(x, lag + 1)[, 2:(lag + 1)]2 -0.47569    0.04215 -11.285   <2e-16 ***
embed(x, lag + 1)[, 2:(lag + 1)]3  0.74082    0.04184  17.708   <2e-16 ***
# Visi trys koeficientai reikšmingi ir beveik lygūs tikrosioms reikšmėms
Residual standard error: 1.011 on 256 degrees of freedom
Multiple R-Squared: 0.6032,     Adjusted R-squared: 0.5986
F-statistic: 129.7 on 3 and 256 DF,  p-value: < 2.2e-16

lm.lag(ar3,4) # Gal tiriamoji seka yra AR(4) procesas?

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      0.02686    0.06366   0.422    0.673
embed(x, lag + 1)[, 2:(lag + 1)]1  0.50475    0.06288   8.027 3.71e-14 ***
embed(x, lag + 1)[, 2:(lag + 1)]2 -0.48841    0.05196  -9.400  < 2e-16 ***
embed(x, lag + 1)[, 2:(lag + 1)]3  0.75351    0.05182  14.541  < 2e-16 ***
embed(x, lag + 1)[, 2:(lag + 1)]4 -0.02637    0.06267  -0.421    0.674
# Ketvirtasis koeficientas nėra reikšmingas
Residual standard error: 1.014 on 254 degrees of freedom
Multiple R-Squared: 0.6035,     Adjusted R-squared: 0.5972
F-statistic: 96.65 on 4 and 254 DF,  p-value: < 2.2e-16
```

[9] Išsiaiškinkite, ką daro `embed` funkcija.

LITERATŪRA

[D] Diebold F. Elements of Forecasting, 3rd Ed., 2003, Thomson South-Western

[C] Chan, Ngay H. Time Series Applications to Finance, 2002, John Wiley & Sons, Inc.

[E] Enders W. Applied Econometric Time Series, 1995, John Wiley & Sons, Inc.

[HS] Harris R., Sollis R. Applied Time Series. Modelling and Forecasting, 2003, John Wiley & Sons, Ltd.

[JD] Johnston J., DiNardo J., Econometric Methods, 4th Ed., 1997, The McGraw-Hill Companies, Inc.

[K] Koop G., Analysis of Economic Data, 2nd Ed., 2006, John Wiley & Sons, Ltd.

[MWH] Makridakis S., Wheelright S.C., Hyndman R.J. Forecasting – Methods and Applications, 3rd Ed., 1998, John Wiley & Sons

[L] Leipus R. Finansinės laiko eilutės, http://mif.vu.lt/~remis

[Pa] Patterson K., An Introduction to Applied Econometrics. A time Series Approach, 2000, Palgrave

[Pf] Pfaff B., Analysis of Integrated and Cointegrated Time Series with R, 2005, Springer

[PR] Pindyck R. S., Rubinfeld D.L. Econometric Models and Economic Forecasts, 4th Ed., 1998, Irwin/McGraw-Hill

[S] Stewart K. G., Introduction to Applied Econometrics, 2005 Brooks/Cole

[T1] Tsay R.S., Analysis of Financial Time Series, 2001, Wiley Inter-Science

[T2] Tsay R.S., Analysis of Financial Time Series, 2nd Ed., 2005, Wiley Inter-Science