

VILNIAUS UNIVERSITETAS
EKONOMIKOS FAKULTETAS
KIEKYBINIŲ METODŲ IR MODELIAVIMO KATEDRA

DOVYDAS ČEILUTKA

Ekonominės analizės ir planavimo magistro programa

MAGISTRO DARBAS

DETERMINISTINIŲ KOMPONENTŲ VAIDMUO
MODELIOJANT KOINTEGRUOTUS KINTAMUOSIUS
THE ROLE OF DETERMINISTIC COMPONENTS
WHILE MODELING COINTEGRATED VARIABLES

Leidžiama ginti _____
(parašas)

Katedros vedėjas
Prof. dr. **L. Čekanavičius**

Magistrantas _____
(parašas)

Darbo vadovas _____
(parašas)

Doc. dr. **A. Bartkus**

Įteikimo data:
Registracijos Nr.

Vilnius, 2016

TURINYS

Santrumpų sąrašas	3
Įvadas	4
1. Nestacionarūs procesai	7
1.1. Stacionarumas	8
1.2. Integruoti procesai	9
1.2.1. Vienetinių šaknų procesai	9
1.2.2. Integruotumo tikrinimas	10
1.3. Kointegruoti procesai	14
1.3.1. Kointegracijos rangas	15
1.3.2. Granger reprezentacijos teorema	15
2. Vektoriniai paklaidų korekcijos modeliai	17
2.1. Vektorinių paklaidos korekcijos modelių specifikacija	17
2.1.1. Vėlavimų eilės pasirinkimas	18
2.1.2. Deterministinių komponentų pasirinkimas	19
2.1.3. VECM be deterministinių komponentų	20
2.1.4. VECM su laisvuju nariu	21
2.1.5. VECM su tiesiniu trendu	21
2.1.6. VECM su tiesiniu trendu paklaidos korekcijos komponente	21
2.1.7. VECM su fiktyviais kintamaisiais	22
2.1.8. Deterministiniai komponentai kointegruotuose procesuose	22
2.2. Vektorinių paklaidos korekcijos modelių įvertinimas	26
2.2.1. Engle - Granger metodas	26
2.2.2. Johansen procedūra	27
2.2.3. Kointegracijos rango nustatymas su struktūriniu lūžiu	30
2.3. VECM diagnostika	32
2.3.1. Paklaidų autokoreliacijos tikrinimas	32
2.3.2. Paklaidų nenormalumo tikrinimas	32
2.3.3. Struktūrinių pokyčių tikrinimas	32
2.4. Struktūriniai VECM	33
2.4.1. Struktūrinių VECM įvertinimas	33
2.4.2. Impulso atsako funkcijos	34
3. Johansen procedūrų specifikacijos analizė naudojant Monte Carlo simuliacijas	35
3.1. Tyrimo aprašymas	35
3.2. Modelis 1	38
3.2.1. Johansen procedūra be deterministinių komponentų	40
3.2.2. Johansen procedūra su konstana	46
3.2.3. Johansen procedūra su trendu	52
3.2.4. Misspecifikacijos įtaka rezultatams	58
3.3. Modelis 2	67
3.3.1. Johansen procedūra be deterministinių komponentų	69
3.3.2. Johansen procedūra su konstana	75
3.3.3. Johansen procedūra su trendu	81
3.3.4. Misspecifikacijos įtaka rezultatams	87
3.4. Modelis 3	96

3.4.1.	Johansen procedūra be deterministinių komponentų	98
3.4.2.	Johansen procedūra su konstanta	104
3.4.3.	Johansen procedūra su trendu	109
3.4.4.	Misspecifikacijos įtaka rezultatams	115
3.5.	Monte Carlo simuliacijų rezultatai	124
3.5.1.	Modelis be deterministinių komponentų	125
3.5.2.	Modelis su laisvuojamu nariu	126
3.5.3.	Modelis su trendu	126
3.5.4.	Monte Carlo simuliacijų išvados	127
4.	Deterministinių komponentų vaidmuo modeliuojant pinigų paklausą	129
4.1.	Tiriamas makroekonominis modelis	129
4.2.	VECM specifikacijų ir įvertinimų palyginimų metodologija	130
4.3.	Deterministinių komponentų įtaka modeliuojant JAV pinigų paklausą	131
4.3.1.	Pinigų paklausos tyrimo modelis ir kintamųjų pasirinkimas	131
4.3.2.	Aprašomoji duomenų analizė	132
4.3.3.	Modelio specifikacija	136
4.3.4.	Rezultatų palyginimas	136
4.4.	Deterministinių komponentų įtaka modeliuojant Lietuvos pinigų paklausą	139
4.4.1.	Pinigų paklausos tyrimo modelis ir kintamųjų pasirinkimas	140
4.4.2.	Aprašomoji duomenų analizė	140
4.4.3.	Modelio specifikacija	144
4.4.4.	Rezultatų palyginimas	145
4.5.	Pinigų paklausos analizės išvados	150
	Išvados ir pasiūlymai	151
	Literatūros sąrašas	154
	Summary	159
	Priedas Nr. 1.	163
	Priedas Nr. 2.	180

Santrumpų sąrašas

- ADF - išplėstinis Dickey ir Fullerio testas
- AIC - Akaike informacinis kriterijus
- AK – atsitiktinis klaidžiojimas
- AKP – atsitiktinis klaidžiojimas su poslinkiu
- AR(p) - autoregresinis modelis, kurio eilė p
- ARIMA(p, d, q) - integruotas autoregresinis slankiųjų vidurkių modelis, kurio eilė (p, d, q)
- ARMA(p, q) - autoregresinis slankiųjų vidurkių modelis, kurio eilė (p, q)
- BVP - bendrasis vidaus produktas
- DV – fiktyvusis kintamasis (angl. Dummy variable)
- MA(q) - slankiųjų vidurkių modelis, kurio eilė q
- MKM - mažiausiųjų kvadratų metodas
- MTR - maksimali tikrinė reikšmė
- P - pėdsakas
- PIC – Phillipso informacinis kriterijus
- RBVP - realusis bendrasis vidaus produktas
- RSS – paklaidų kvadratų suma
- SIC - Szwartzo informacinis kriterijus
- SVAR – struktūrinis vektorinės autoregresijos modelis
- SVEC – struktūrinis vektorinis paklaidų (pusiausvyros) korekcijos modelis
- SVKI - suderintas vartotojų kainų indeksas
- VAR – vektorinės autoregresijos modelis
- VARMA – vektorinis autoregresinis slankiųjų vidurkių modelis
- VECM – vektorinis paklaidų (pusiausvyros) korekcijos modelis
- VKI - vartotojų kainų indeksas

Įvadas

Laiko eilučių analizėje dažniausiai sutinkami ne stacionarūs, o integruoti procesai (Maddala ir Kim, 1998). Yule (1926) nustatė, kad kartais modeliuojant laiko eilučių procesus turinčius tendą yra gaunama klaidinga regresija. Klaidingos regresijos problema detaliai ištirta Granger ir Newbold (1974). Šie tyrimai buvo pamatas leidęs atrasti kointegraciją. Kointegruojančių vektorių panaudojimą modeliuojant integruotus procesus tyrė Granger (1981), Granger ir Weiss (1983), Engle ir Granger (1985), o jų sąryšį su paklaidos korekcijų modeliais tyrė Stock (1987). Tačiau perversmą kointegracijos ir laiko eilučių ekonometrijos srityje sukėlė Engle ir Granger (1987) pasiūlytas dviejų pakopų regresijos pritaikymas kointegruotų kintamųjų modelių įvertinimui ir Johansen (1988) pasiūlytas maksimalaus tikėtimumo procedūros pritaikymas įvertinant VECM modelių parametrus. Nors ir anksčiau buvo bandoma modeliuoti integruotus kintamuosius naudojant Box ir Jenkins (1970) pasiūlytus ARIMA modelius, kurie pašalindavo stochastinius trendus kintamuosius diferencijuodami, tačiau kointegracija pirmą kartą leido modeliuoti integruotų kintamųjų sistemas ir jų tarpusavio dinaminis ryšius. Taip pat pirmą kartą buvo galima į modelius įtraukti integruotų kintamųjų lygio išraiškas, o ne tik skirtumines transformacijas, kurias galima iširti naudojant vektorinės autoregresijos modelius. Vektorinis paklaidų autokorekcijos modelis - pagrindinė kointegruotų kintamųjų analizavimo priemonė – gali būti įvairiai specifikuotas, į modelį įtraukiant ne tik kintamųjų vėlavimus bet ir įvairių formų deterministinius komponentus. Kointegruotų kintamųjų sistemose deterministiniai komponentai gali įgauti daug skirtingų išraiškų, todėl teisingas VECM specifikuojimas ypatingai svarbus procesas. VECM specifikuojimui ir diagnostikai yra sukurta keletas procedūrų Johansen (1995), Johansen ir kiti (2000), tačiau jie neapima viso deterministinių komponentų įtraukimą į modelius spektro, taip pat nepateikia atsakymų, kokią įtaką turi deterministiniai komponentai ir imties dydis Johansen procedūros pateikiamiems rezultatams, todėl VECM specifikuojimas išlieka aktualia problema modeliuojant kointegruotus kintamuosius su deterministiniais komponentais.

Temos aktualumas ir naujumas. Deterministinių komponentų įtraukimas į kointegruotų kintamųjų modelius yra ypatingai svarbus modeliuojant makroekonominis kintamuosius. Daugelis makroekonominis kintamųjų turi deterministinių komponentų, o neteisingai specifikuojant modelį ne tik gaunami klaidingi parametrai, tačiau kai kuriais atvejais ir neteisingai nustatomas kointegruojančių vektorių skaičius. Siekiant patikimai įvertinti VECM modelius ir panaudoti juos prognozavimui, būtina teisingai specifikuoti VECM, pasirinkti tinkamas įvertinimo procedūras, suprasti misspecifikacijos įtaką rezultatams bei, panaudoti tinkamiau-

sius kointegracijos testus tiriamiems atvejams.

Mokslinio tyrimo objektas. Mokslinio tyrimo objektas - Johansen procedūrų specifikuojimas, įtraukiant įvairias deterministinių komponentų kombinacijas, įvertinant VECM modelius su deterministiniais komponentais, VECM įvertinimo procedūrų pasirinkimo įtaka kointegruojančių vektorių skaičiaus nustatymo ir VECM parametrų įvertinimo rezultatams priklausomai nuo imties dydžio.

Darbo tikslas ir uždaviniai. Darbo tikslas - naudojant Monte Carlo simuliacijas sumodeliuoti VECM modelius su įvairiais deterministiniais komponentais, ištirti skirtingai specifikuotų Johansen procedūrų nustatomą kointegruojančių vektorių skaičių priklausomai nuo imties dydžio ir parengti metodologiją įvertinti deterministinių komponentų įtakai šioms procedūroms. Darbo tikslui pasiekti buvo sprendžiami šie uždaviniai:

1. Išanalizuoti mokslines publikacijas nestacionarių (integruotų ir kointegruotų) procesų analizės tema.
2. Išanalizuoti VECM modelių specifikuojimo ir įvertinimo procedūras.
3. Atlikti Monte Carlo simuliacijas su įvairiai specifikuotomis Johansen procedūromis įvertinant įvairių VECM modelių kointegruojančių vektorių skaičių priklausomai nuo imties dydžio.
4. Išanalizuoti Johansen procedūros specifikuojimo įtaką VECM modelių įvertinimui modeliuojant JAV ir Lietuvos pinigų paklaudas.
5. Pateikti išvadas dėl deterministinių komponentų įtakos modeliuojant kointegruotus kintamuosius.

Mokslinė darbo problema. Mokslinė darbo problema - įvairiai specifikuotų Johansen procedūrų pasirinkimo ir VECM specifikuojimo įtaka kointegruojančių vektorių skaičiaus nustatymui ir VECM parametrų įvertinimui. Nors galimos VECM specifikuojimo yra gerai ištirtos, teoriniu lygmeniu nėra vieningo susitarimo dėl deterministinių komponentų ir imties dydžio įtakos jų įvertinimo procedūroms. Empiriniu lygmeniu Johansen procedūros specifikuojamos į paklaidos korekcijos komponentą įtraukiant konstantą arba trendą, o į paklaidos nekoreguojantį komponentą įtraukiama konstanta. Kitokių formų deterministinių kintamųjų įtraukimas labai retas. Ieškoma atsakymo į klausimus: kokią įtaką turi neteisinga VECM specifikuojimo; kokiais atvejais į modelius ir įvertinimo procedūras reikia įtraukti deterministinius komponentus; kokią įtaką įvertinimo procedūroms daro imties dydis; kokia turėtų

būti deterministinių komponentų pasirinkimo procedūra ir nuo kokių kintamųjų sąvybių ji priklauso? Šiame darbe VECM su įvairiai specifikuotomis deterministinių kintamųjų kombinacijomis bus analizuojami empiriniu aspektu.

Darbo struktūra. Ši magistro darbą sudaro šeši pagrindiniai skyriai:

1. Įvadas. Darbo įvade pateikiama informaciją apie temos aktualumą ir naujumą, mokslinio tyrimo objektą, darbo tikslą ir uždavinius bei mokslinę darbo problemą.
2. Nestacionarių procesų mokslinės literatūros analizė. Šioje dalyje analizuojama mokslinė literatūra nestacionarių procesų tema: apibrėžiama stacionarumo svarba, išanalizuojami vienetinių šaknų procesai, integruotumo testai, kointegracijos rangas ir pristatoma Granger reprezentacijos teorema.
3. Vektoriniai paklaidų korekcijos modeliai. Šiame skyriuje pristatomi VECM specifikuotais variantais ir tvarka, VECM įvertinimo procedūros, VECM diagnostika bei struktūriniai VECM modeliai.
4. Johansen procedūrų specifikuotų analizė naudojant Monte Carlo simuliacijas. Šiame skyriuje aprašomi trys skirtingai specifikuoti VECM modeliai ir pateikiami Monte Carlo simuliacijų rezultatai parodantys deterministinių komponentų įtaką kointegruojančių vektorių skaičiaus nustatymui priklausomai nuo imties dydžio.
5. JAV ir Lietuvos pinigų paklausos analizė. Šioje darbo dalyje įvertinami JAV ir Lietuvos pinigų paklausos modeliai naudojant įvairiai specifikuotas Johansen procedūras panaudojant Monte Carlo simuliacijų rezultatus apie Johansen procedūrų elgseną.
6. Išvados ir pasiūlymai. Šiame skyriuje pateikiamos darbo išvados bei tolesnių tyrimų kryptys ir pasiūlymai.

1. Nestacionarūs procesai

Ekonometrinėje analizėje tyrimo objektai dažniausiai yra diskretieji stochastiniai procesai pvz. pinigų paklausa, bendrasis vidaus produktas, palūkanų normos ir t.t. Tiriant tokius kintamuosius įprasta turėti tik vieną atsitiktinio dydžio įvertį konkrečiu laiko periodu, o ne jų derinį (Enders, 2010). Kad būtų galima tirti tokius kintamuosius būtina sumažinti tiriamų parametrų skaičių. Tam yra taikomi apribojimai. Maddala ir Kim (1998) išskiria dvi apribojimų rūšis:

1. Stacionarumas.
2. Asimptotinis nepriklausomumas.

Asimptotinis nepriklausomumas (angl. asymptotic independence) yra apribojimai proceso atminčiai, kuriuos tyrė Spanos (1986). Stacionarumo įtaką parametrų skaičiui parodo stochastinis procesas X_t , kurio parametrai Φ , kai t poaibis $(t_1, \dots, t_n) \in T$. Be stacionarumo sąlygos, X_t jungtinis skirstinys yra charakterizuojamas tokiu parametrų vektoriumi:

$$\Phi = [\mu_{t_i}, cov(X_{t_i}, X_{t_j})] \text{ kai } i, j = 1, \dots, n \quad (1)$$

Šio vektorius dimensijos yra $n + \frac{n(n+1)}{2} \times 1$. Su stacionarumo sąlyga parametrų skaičius sumažėja taip:

$$\Phi = [\mu_{t_i}, \sigma^2, \gamma] \quad (2)$$

Šio vektorius dimensijos yra $(n + 1) \times 1$. Stacionarumo sąlyga leidžia žymiai sumažinti nežinomų parametrų skaičių (Maddala ir Kim, 1998). Toks parametrų skaičiaus sumažinimas yra būtinas tiriant laiko eilučių modelius.

Įprastai integruoti (nestacionarūs) kintamieji buvo tiriami juos diferencijuojant, tačiau būtinybė diferencijuoti kintamuosius apsunkina prognozavimą, modelių tinkamumo tikrinimą ir kitus ekonominius tyrimus. Kointegracija ir vektoriniai paklaidų korekcijos modeliai išsiskiria iš kitų laiko eilučių tyrimų tuo, kad leidžia modeliuoti integruotus kintamuosius jų nediferencijuojant. Kadangi stacionarumo sąlyga vis tiek yra būtina stochastinis trendas yra eliminuojamas sudarant tokią tiesinę kintamųjų kombinaciją, kad kintamųjų stochastiniai trendai vienas kitą neutralizuoja.

1.1. Stacionarumas

Formaliai stacionarumas yra skirstomas į griežtą ir negriežtą stacionarumą. Laiko eilutė yra griežtai stacionari, jei jos jungtinis skirstinys X_{t_1}, \dots, X_{t_n} yra toks pat kaip ir $X_{t_1+\tau}, \dots, X_{t_n+\tau}$ visiems t_1, \dots, t_n , ir τ . Stacionaraus proceso skirstinys nepriklauso nuo t , o tik nuo τ (Maddala ir Kim, 1998).

Laiko eilučių analizėje šią stacionarumo koncepciją yra sunku patikrinti, nes ji yra apibrėžta kaip skirstinio funkcija, o skirstinys beveik visada yra nežinomas. Dėl šios priežasties dažniausiai naudojama negriežto stacionarumo (kitai vadinama antros eilės stacionarumo arba kovariacijos stacionarumo) sąlyga teigianti, kad stochastinis procesas yra negriežtai stacionarus, jei jo vidurkis ir dispersija nepriklauso nuo laiko (Lütkepohl, 2007). Stochastinis procesas X_t , $t \in T$ yra stacionarus l eile, jei visiems (t_1, t_2, \dots, t_n) iš T ir visiems τ jungtiniai momentai yra $E(X_{t_1}^{l_1}, \dots, X_{t_n}^{l_n}) = E(X_{t_1+\tau}^{l_1}, \dots, X_{t_n+\tau}^{l_n})$, kur $l_1 + \dots + l_n \leq l$.

Kai $l = l_1 = 1$:

$$E(X_t) = E(X_{t+\tau}) = \mu \quad (3)$$

kur μ yra konstanta. Toks procesas X_t yra stacionarus pirmąja eile.

Kai $l = 2$:

$$E(X_t) = E(X_{t+\tau}) = \mu \quad (4)$$

$$E(X_t^2) = E(X_{t+\tau}^2) = \sigma^2 \quad (5)$$

kur μ ir σ yra konstantos. Toks procesas X_t yra stacionarus antrąja eile.

Daugiamatnių modelių kontekste stacionarumas yra apibrėžiamas taip:

k dimensijų modelis z_t yra silpnai stacionarus, jei:

$$E(z_t) = \mu \quad (6)$$

$$Cov(z_t) = E[(z_t - \mu)(z_t - \mu)'] = \Sigma_z \quad (7)$$

kur μ yra k dimensijų konstantų vektorius ir Σ_z yra $k \times k$ dimensijų teigiamai apibrėžta matrica. Modelio z_t matematinės vilties vektorius ir kovariacijų matrica nepriklauso nuo laiko, jei jis yra silpnai stacionarus.

k dimensijų modelis z_t yra griežtai stacionarus, jei jo jungtinis skirstinys $(z_{t_1}, \dots, z_{t_m})$ yra toks pat kaip ir $z_{t_1+j}, \dots, z_{t_m+j}$ visiems t_1, \dots, t_m , ir j .

1.2. Integruoti procesai

Laiko eilučių analizėje dažniausiai sutinkami ne stacionarūs, o integruoti procesai (Maddala ir Kim, 1998). Ekonominėje analizėje daugelis kintamųjų turi nėra stacionarūs (Lütkepohl, 2007). Integruotais vadinami tokie procesai, kurių charakeringosios šaknys yra ant arba už vienetinio apskritimo ribų t.y. jie turi vienetinę šaknį. Integruoti kintamieji mokslinėje literatūroje yra žymimi $I(d)$, kur d yra integruotumo eilė.

Formalų integruotumo apibrėimą pateikia Pfaff (2008): laiko eilutė be deterministinių komponentų, turinti stacionarią ir invertabilią Box - Jenkins (ARMA) reprezentaciją po diferencijavimo d kartų yra integruota d eile ir žymima $x_t I(d)$.

1.2.1. Vienetinių šaknų procesai

Vienetinių šaknų procesai taip vadinami todėl, kad jie turi stochastinį trendą. Skirtingai nei stacionariuose procesuose, vienetinių šaknų procesuose šokų poveikis niekada neišnyksta - paklaidų įtaka procesui yra kaupiamoji.

Atsitiktinio klaidžiojimo procesas. Atsitiktinio klaidžiojimo procesas puikiai atspindi pagrindines integruotų procesų savybes. Atsitiktinio klaidžiojimo procesas be poslinkio turi tokią formą:

$$y_t = y_{t-1} + \varepsilon_t \quad (8)$$

$$y_t = y_0 + \sum_{s=1}^t \varepsilon_s \quad (9)$$

kur ε_t yra n.i.d. procesas. Jei y_0 toks procesas turi tokius parametrus:

$$E(y_t) = 0 \quad (10)$$

$$var(y_t) = t\sigma^2 \quad (11)$$

Toks procesas pažeidžia stacionarumo sąlygą, nes jo dispersija auga bėgant laikui. N.i.d. sąlyga yra labai svarbi, nes, jei tokio proceso paklaidos bus autokoreliuotos, toks procesas nebus atsitiktinio klaidžiojimo, tačiau vistiek bus vienetinių šaknų procesas.

Atitiktinio klaidžiojimo su poslinkiu procesas.

$$y_t = \mu + y_{t-1} + \varepsilon_t \quad (12)$$

$$y_t = y_0 + \mu t + \sum_{s=1}^t \varepsilon_s \quad (13)$$

kur ε_t yra baltasis triukšmas. Atitiktinio klaidžiojimo su poslinkiu procesas turi ir deterministinių ir stochastinių tendą. Abu šie trendai gali būti pašalinti diferencijuojant tokį procesą.

1.2.2. Integruotumo tikrinimas

Box ir Jenkins (1970) integruotumo eilei nustatyti paklausė išanalizuoti kintamųjų autokorelogramas. Stacionarių kintamųjų autokorelogramos yra eksponentiškai krinantčios, tuo tarpu integruotų kintamųjų korelogramos mažėja palengva.

Dickey-Fuller testas. Dickey (1976), Fuller (1976) Dickey ir Fuller (1979) ir sukūrė ir pristatė vienetinių šaknų testus, kurie yra vadinami Dickey-Fuller testais. Jie leidžia formaliai įvertinti ar kintamieji yra integruoti ir nustatyti jų integruotumo eilę.

$$\Delta y_t = \gamma y_{t-1} + \varepsilon_t \quad (14)$$

$$\Delta y_t = a_0 + \gamma y_{t-1} + \varepsilon_t \quad (15)$$

$$\Delta y_t = a_0 + a_1 t + \gamma y_{t-1} + \varepsilon_t \quad (16)$$

Šios lygtys skiriasi tik deterministinių komponentų specifikacija. Lygtis (14) deterministinių komponentų neturi, lygtis (15) įvertina konstantą (dar vadinamą poslinkiu), o lygtis (16) įtraukia ir poslinkį ir linijinį tendą.

Lygtyse (15) ir (16) parametrai a_0 ir a_1 yra skirtingai interpretuojami nulinės ir alternatyviosios hipotezės atveju. Lygtyje (15) nulinės hipotezės atveju, kai $\gamma = 0$, a_0 yra trendo koeficientas, o alternatyviosios hipotezės atveju, kai $\gamma \neq 0$, y_t yra stacionarus su konstanta $a_0/(1 - \gamma)$. Lygtyje (16) nulinės hipotezės atveju, kai $\gamma = 0$, parametrai a_0 reprezentuoja tendą, o a_1 reprezentuoja kvadratinį tendą.

Papildytasis Dickey-Fuller testas. Įprastasis Dickey-Fuller testas netinka autoregresiniams procesams tirti, todėl Dickey ir Fuller (1981) paklausė papildytąjį Dickey-Fuller testo variantą, įtraukiant autoregresinius vėlavimus į testo lygtis.

$$\Delta y_t = \gamma y_{t-1} + \sum_{i=2}^p \beta_i \Delta y_{t-i+1} + \varepsilon_t \quad (17)$$

$$\Delta y_t = a_0 + \gamma y_{t-1} + \sum_{i=2}^p \beta_i \Delta y_{t-i+1} + \varepsilon_t \quad (18)$$

$$\Delta y_t = a_0 + a_1 t + \gamma y_{t-1} + \sum_{i=2}^p \beta_i \Delta y_{t-i+1} + \varepsilon_t \quad (19)$$

Nuo įpratinio Dickey-Fuller testo lygčių (17), (18) ir (19) skiriasi tik papildomais kintamojo vėlavimais.

Dickey-Fuller nėra vieninteliai integruotumo testai. Deterministinių komponentų egzistavimas tiriamose laiko eilutėse smarkiai sumažina Dickey-Fuller testų galią. Dėl šios ir kitų Dickey-Fuller testų trūkumų yra sukurta nemažai kitų vienetinių šaknų testų.

Phillips-Perron testas. Phillips ir Perron (1988) ir Perron (1988) paklausė neparametrinius vienetinių šaknų testus, kurie leidžia tirti kintamuosius turinčius siplną priklausomumą ir paklaidų proceso heterogeniškumą.

$$y_t = \mu + \alpha y_{t-1} + \varepsilon_t \quad (20)$$

$$y_t = \mu + \beta \left(t - \frac{1}{2}T\right) + \alpha y_{t-1} + \varepsilon_t \quad (21)$$

Testo statistikos lygčiai (20) yra apibrėžiamos taip:

$$Z(\hat{\alpha}) = T(\hat{\alpha} - 1) - \hat{\lambda} / \bar{m}_{yy} \quad (22)$$

$$Z(\tau_{\hat{\alpha}}) = (\hat{s} / \hat{\sigma}_{Tl}) t_{\hat{\alpha}} - \hat{\lambda}'_{Tl} / \bar{m}_{yy}^{\frac{1}{2}} \quad (23)$$

$$Z(\tau_{\hat{\mu}}) = (\hat{s} / \hat{\sigma}_{Tl}) t_{\hat{\mu}} + \hat{\lambda}' \sigma_{Tl} m_y / \bar{m}_{yy}^{\frac{1}{2}} m_{yy}^{\frac{1}{2}} \quad (24)$$

kur $\bar{m}_{yy}^{\frac{1}{2}} = T^{-2} \sum (y_t - \bar{y})^2$, $m_{yy} = T^{-2} \sum y_t^2$, $m_y = T^{-\frac{3}{2}} \sum y_t$, $\hat{\lambda} = 0.5(\sigma_{Tl}^2 - \hat{s}^2)$, kur \hat{s}^2 yra imties paklaidų dispersija, $\hat{\lambda}' = \hat{\lambda} / \hat{\sigma}_{Tl}^2$, $t_{\hat{\alpha}}$ yra parametro įverčio $\hat{\alpha}$ t reikšmė, o $t_{\hat{\mu}}$ yra parametro įverčio $\hat{\mu}$ t reikšmė. Ilgojo laikotarpio dispersija yra įvertinama taip:

$$\hat{\sigma}_{Tl}^2 = T^{-1} \sum_{t=1}^T \hat{\varepsilon}_t^2 + 2T^{-1} \sum_{s=1}^l w_{sl} \sum_{t=s+1}^T \hat{\varepsilon}_t \hat{\varepsilon}_{t-s} \quad (25)$$

kur $w_{sl} = 1 - s/(l + 1)$.

Lygčiai (21) testo statistikos yra apibrėžiamos taip:

$$Z(\tilde{\alpha}) = T(\tilde{\alpha} - 1) - \tilde{\lambda}/M \quad (26)$$

$$Z(t_{\tilde{\alpha}}) = (\tilde{s}/\tilde{\sigma}_{Tl})t_{\tilde{\alpha}} - \tilde{\lambda}'_{Tl}/M^{\frac{1}{2}} \quad (27)$$

$$Z(t_{\tilde{\mu}}) = (\tilde{s}/\tilde{\sigma}_{Tl})t_{\tilde{\mu}} - \tilde{\lambda}'\tilde{\sigma}_{Tl}m_y/M^{\frac{1}{2}}(M + m_y^2)^{\frac{1}{2}} \quad (28)$$

$$Z(t_{\tilde{\beta}}) = (\tilde{s}/\tilde{\sigma}_{Tl})t_{\tilde{\beta}} - \tilde{\lambda}'\tilde{\sigma}_{Tl}(\frac{1}{2}m_y - m_{ty})/(M/12)^{\frac{1}{2}}\tilde{m}_{yy}^{\frac{1}{2}} \quad (29)$$

kur m_y , \tilde{m}_{yy} , $\tilde{\lambda}$, $\tilde{\lambda}'$ ir $\tilde{\sigma}_{Tl}$ reiškia tą patį, ką ir lygtyje (20), o $m_{ty} = T^{-5/2} \sum ty_t$, $t_{\tilde{\mu}}$ yra $\tilde{\mu}$ t santykis, $t_{\tilde{\beta}}$ yra $\tilde{\beta}$ t santykis, $t_{\tilde{\alpha}}$ yra $\tilde{\alpha}$ t santykis, $M = (1 - T^{-2})m_{yy} - 12m_{ty}^2 + 12(1 + T^{-1})m_{ty}m_y - (4 + 6T^{-1} + 2T^{-2})m_y^2$.

Kritinės Z statistikų reikšmės yra tokios pat, kaip ir Dickey-Fuller testų atveju. Phillips-Perron testo pranašumas prieš Dickey-Fuller testą yra tada, kai tiriamo kintamojo paklaidos nėra normaliai pasiskirsčiusios. Kita vertus šiame teste būtina teisingai nustatyti vėlavimų eilę l naudojamą ilgojo laikotarpio dispersijoms suskaičiuoti.

Elliott-Rothenberg-Stock testas. Dickey-Fuller ir Phillips-Perron testų trūkumas yra jų maža galia, kai duomenis generuojantis procesas yra pirmosios eilės autoregresinis procesas su aukštu autoregresijos parametru. Elliott ir kiti (1992) pristatė Elliott-Rothenberg-Stock testą, kuris sprendžia šią problemą. Šis naudoja metodą panašų į diferencijavimą, kai yra pasirenkama konstanta α , kuri yra artima 1 ir kintamojo dalinis skirtumas \tilde{y}_t yra gaunamas iš y_t atimant αy_{t-1} .

$$\tilde{y}_t = a_0 + a_2 t - \alpha a_0 - \alpha a_2(t - 1) + e_t, \quad \text{kai } t = 2, \dots, T \quad (30)$$

Kai $t = 1$ tokiu būdu gauti kintamojo dalinio skirtumo nėra įmanoma, todėl \tilde{y}_t yra tiesiog y_t . Tada galima (30) lygtį perrašyti taip:

$$\tilde{y}_t = (1 - \alpha)a_0 + a_2[(1 - \alpha)t + \alpha] + e_t \quad (31)$$

Pirmame Elliott-Rothenberg-Stock testo etape reikia įvertinti (31) lygtį, naudojant mažiausiųjų kvadratų metodą.

Antrame Elliott-Rothenberg-Stock testo etape gauti a_0 ir a_2 įverčiai yra panaudojami pašalinant trendą iš y_t .

$$\Delta y_t^d = \gamma y_{t-1}^d + \sum_{i=1}^p c_i \Delta y_{t-i}^d + \varepsilon_t \quad (32)$$

kur $y_t^d = y_t - \hat{a}_0 - \hat{a}_2 t$. Elliott ir kiti (1992) siūlo naudoti SC informacinį kriterijų pasirenkant vėlavimų eilę p .

Elliott (1999) siūlo naudoti $\alpha = (1 - 10)/T$ visoms modelio specifikacijoms - su deterministiniu trendu ir be jo.

Kwiatkowski-Phillips-Schmidt-Shin testas. Kwiatkowski ir kiti (1992) pristatė Lagrange daugiklių (LD) vientinių šaknų testą, kuris mokslinėje literatūroje yra vadinamas KPSS. Šis testas skiriasi nuo DF, PP ir ERS testų tuo, kad jo nulinė hipotezė yra stacionarumas, o alternatyvioji hipotezė yra integruotumas. Kointegruotų kintamųjų analizėje galima panaudoti šį testą formaliai parodant, kad tikrai visi kintamieji yra integruoti, nors jų tiesinė kombinacija yra stacionari.

KPSS testas naudoja tokį modelį:

$$y_t = \delta t + \mu_t + \varepsilon_t \quad (33)$$

kur ε_t yra stacionarus procesas, o μ_t yra atsitiktinio klaidžiojimo procesas, turintis tokią formą $\mu_t = \mu_{t-1} + u_t$.

Zivot-Andrews testas. Zivot ir Andrews (1992) pristatė vienetinių šaknų testą skirtą nustatyti stacionariems procesams turintiems struktūrinį lūžį. Šio testo nulinė hipotezė teigia, kad kintamasis yra nestacionarus ir neturi struktūrinio lūžio, o alternatyvioji hipotezė teigia, kad kintamasis yra stacionarus ir turi struktūrinį lūžį, todėl testo specifikacija yra asimetriška.

Nulinės hipotezės testinės Zivot-Andrews lygtys yra užrašomos taip:

$$y_t = \mu + dD(T_B)_t + y_{t-1} + e_t, \quad (34)$$

$$y_t = \mu + y_{t-1} + (\mu_2 - \mu_1)DU_t + e_t, \quad (35)$$

$$y_t = \mu + dD(T_B)_t + y_{t-1} + dD(T_B) + e_t \quad (36)$$

Alternatyviosios hipotezės testinės lygtys yra užrašomos taip:

$$y_t = \mu_1 + \beta t + (\mu_2 - \mu_1)DU_t + e_t, \quad (37)$$

$$y_t = \mu + \beta_1 t + (\beta_2 - \beta_1)DT_t^* + e_t, \quad (38)$$

$$y_t = \mu + \beta_1 t + (\mu_2 - \mu_1)DU_t y_{t-1} + (\beta_2 - \beta_1)DT_t^* + e_t \quad (39)$$

Zivot-Andrews testas yra ypatingai naudingas įvertinant fiktyviųjų deterministinių komponentų įtaką nustatant kintamųjų stacionarumą.

1.3. Kointegruoti procesai

Engle ir Granger (1987), Granger (1981) kointegruotais vadina tokius integruotus kintamuosius, kurie turi tokią nenulinę tiesinę kombinaciją, kuri yra integruota žemesne eile nei patys kintamieji. Jei visi vektoriaus elementai laiko eilutėje x_t yra integruoti ir tampa neintegruoti juos diferencijuojant, tačiau jų tiesinė kombinacija $\beta'x_t$ yra stacionari, tai laikoma, kad laiko eilutė x_t yra kointegruota su kointegravimo vektoriumi β . Tuo atveju, kai yra keletas kointegravimo vektorių β tampa matrica. Lygtis $\beta'x_t = 0$ yra interpretuojama kaip ilgojo laikotarpio pusiausvyra, todėl kointegracijos buvimas suponuoja, kad nuokrypiai nuo pusiausvyros yra stacionarūs, su baigtine dispersija, nors pačios laiko eilutės yra nestacionarios ir turi begalines dispersijas. Engle ir Granger (1987) pristatė reprezentavimo teoremą, kuri sujungia slankiojo vidurkio, autoregresijos ir paklaidos korekcijos reprezentacijas kointegruotoms sistemoms ir parodė, kad vektorinė diferencijuotų kintamųjų autoregresija nėra suderinama su šiomis reprezentacijomis. Tai parodė, kad negalima kointegruotų procesų modeliuoti naudojant diferencijuotų kintamųjų VAR modelius, nes tokie modeliai neatsižvelgia į kintamųjų kointegruotumą. Nors Engle ir Granger (1987) pristatė paprastą ir asimptotiškai efektyvią dviejų pakopų įvertinimo procedūrą, tačiau būtent Johansen (1988), Johansen ir Juselius (1990) pristatyta VECM ir maksimalaus tikėtinumo procedūra jo įvertinimui tapo de facto standartu kointegruotų kintamųjų analizėje.

K -dimensijos procesas y_t yra kointegruotas eile (d, b) , jei egzistuoja tiesinė kombinacija $z_t := \beta'y_t$, kur $\beta = (\beta_1, \dots, \beta_K)' \neq 0$, kad $z_t \sim I(d - b)$. Sutrumpintai toks procesas žymimas taip:

$$y_t \sim CI(d, b) \quad (40)$$

kur $x_t \sim I(d)$, $y_t \sim I(d)$, $d > 0$, $b > 0$.

Jei tiesinė integruotų kintamųjų kombinacija yra neintegruota, tokie kintamieji yra kointegruoti.

1.3.1. Kointegracijos rangas

Kai du ar daugiau kintamųjų turi tuos pačius deterministinius ir stochastinius trendus, galima rasti jų linijinę kombinaciją, kuri neutralizuoja abu trendus. Gautas kointegracijos santykis neturi trendo, net jei abu kintamieji turi trendus. Kointegruotame VAR modelyje šis atvejis gali būti analizuojamas įtraukiant trendą į kointegravimo erdvę. Kitais atvejais linijinė kintamųjų kombinacija pašalina stochastinį, bet ne deterministinį trendą, taigi vėl reikia įtraukti linijinį trendą į kointegravimo erdvę. Panašūs argumentai gali būti panaudojami intervenciniam fiktyviam kintamajam: intervencija gali turėti įtaką keletai kintamųjų tokiu pat dydžiu, taigi intervencija išnyksta linijinėje jų kombinacijoje, taigi reikia įtraukti intervencinio kintamojo nereikia. Yra tik keletas pavyzdžių, kurie parodo, kad deterministinių ir stochastinių komponentų vaidmuo yra komplikuoatas. Taip pat svarbu suvokti, kad modelis su neteisingai suformuluotais deterministiniais komponentais gali generuoti klaidingus parametrų įverčius, nes asimptotinis kointegracijos testų skirstinys nėra neveikiamas šių komponentų specifikacijos. Taip pat šios savybės gali būti nepageidautinos prognozavime, nes kartais gali išlaikyti kvadratinius trendus. Bendru atveju prognozavimas yra daug labiau įtakojamas deterministinių VAR komponentų specifikavimo, nei stochastinių. Kitas svarbus aspektas yra išskistyti laisvąjį narį π į komponentus, kurie sukelia augimą sistemoje ir tuos, kurie parodo kointegracijos sistemos vidurkius. Analizuojant tokią VAR išraišką: $\Delta x_t = \Phi_1 \Delta x_{t-1} + \alpha(\beta' x_{t-1}) + \pi + \varepsilon_t$ Pagal kointegruotumo apibrėžimą kintamieji gali būti kointegruotais tik tada, kai jie abu yra integruoti – t.y. turintys stochastinį trendą.

1.3.2. Granger reprezentacijos teorema

Engle ir Granger (1987) pristatyta Granger reprezentacijos teorema leidžia atskirti integruotų kintamųjų integruotas ir neintegruotas dalis. Granger reprezentacijos teorema teigia, kad jei turime kointegruotų kintamųjų vektorių y_t , galima juos užrašyti naudojant VECM išraišką (Johansen, 1995).

$$\Delta y_t = \alpha \beta' y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t, \quad t = 1, 2, \dots, \quad (41)$$

Kur $y_t = 0$, kai $t \leq 0$, u_t yra baltasis triukšmas, kai $t = 1, 2, \dots$, ir $u_t = 0$, kai $t \leq 0$

Mes apibrėžiame $C(z)$ taip:

$$C(z) := (1 - z)I_K - \alpha\beta'z - \sum_{i=1}^{p-1} \Gamma_i(1 - z)z^i \quad (42)$$

Taip pat nustatome tokius apribojimus:

1. $\det C(z) = 0 \Rightarrow |z| > 1$ arba $z = 1$.
2. Vienetinių šaknų skaičius, $z = 1$, yra lygiai $K - r$.
3. α ir β yra $(K \times r)$ dydžio matricos, kur $rk(\alpha) = rk(\beta) = r$.

Tada y_t galima užrašyti

$$y_t = \Xi \sum_{i=1}^t u_i + \Xi^*(L)u_t + y_0^* \quad (43)$$

kur

$$\Xi = \beta_{\perp} [\alpha'_{\perp} (I_K - \sum_{i=1}^{p-1} \Gamma_i) \beta_{\perp}]^{-1} \alpha'_{\perp} \quad (44)$$

kur $\Xi^*(L)u_t = \sum_{j=0}^{\infty} \Xi^* u_{t-j}$ yra $I(0)$ procesas, o y_0^* yra pradinių reikšmių vektorius.

Granger reprezentacijos teorema yra labai svarbi kointegruotų procesų įvertinimui, nes ji atskiria proceso y_t integruotą $I(1)$ ir stacionarią $I(0)$ dalis Lütkepohl (2007).

Lygtis (43) yra daugiamatė Beveridge-Nelson išskaidymo versija. Pirmasis lygties elementas $\sum_{i=1}^t u_i$ susideda iš K atsitiktinio klaidžiojimo procesų. Ξ yra matrica, kurios rangas yra $K - r$. Šių elementų sandauga sistemai duoda $K - r$ stochastinių trendų, kurie yra vienas iš labiausiai y_t reikšmes nulemenčių elementų. Todėl, procesas y_t yra integruotas $I(1)$, jei dešinėje lygties (43) pusėje ši sandauga generuoja stochastinių trendų. Tokiu atveju $\Xi \neq 0$, tada $rank(\Xi) \neq 0$, nes $\Xi^*(L)u_t$ yra $I(0)$ procesas. Jei $r = K$, tada $rank(\Xi) = 0$ ir $\Xi = 0$, todėl procesą y_t nulemia tik stacionari dalis $\Xi^*(L)u_t$ ir todėl procesas y_t yra stacionarus.

Lygtis (44) implikuoja, kad $((K - r) \times (K - r))$ dydžio matrica $\alpha'_{\perp} (I_K - \sum_{i=1}^{p-1} \Gamma_i) \beta_{\perp}$ yra invertabili. Su tokia sąlyga $rk(\Xi) = K - r$, o y_t turi $K - r$ stochastinių trendų.

$\Xi^*(L)u_t$ stacionarumo įrodymas yra pateikiamas Saikkonen (2005) ir Lütkepohl (2007).

2. Vektoriniai paklaidų korekcijos modeliai

Vienmačiai paklaidos korekcijos modeliai buvo nagrinėjami Davidson ir kiti (1978) ir Salmon (1982), kuriuose buvo nagrinėjama kintamųjų priklausomybė nuo jų pusiausvyrų. Šių modelių vektorinė forma puikiai tinka kointegracijos analizei. Šiame darbe bus analizuojami tik daugiamačiai vektoriniai paklaidos korekcijos modeliai. Vektorinis paklaidos korekcijos modelis, kaip jį pristatė Johansen ir Juselius (1990) gali būti užrašomas taip:

$$\begin{aligned}\Delta y_t &= \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \\ \Delta y_t &= \alpha \beta' y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t\end{aligned}\tag{45}$$

Kur $\Gamma_i := -(A_{i+1} + \dots + A_p)$, $i = 1, \dots, p-1$. Lygtį (45) galima užrašyti ir kitokia forma, kurioje paklaidos korekcijos komponentas turėtų p vėlavimą.

$$\Delta y_t = \Pi y_{t-p} + \sum_{i=1}^{p-1} D_i \Delta y_{t-i} + u_t\tag{46}$$

Kur $D_i = -(I_K - A_1 - \dots - A_i)$, $i = 1, \dots, p-1$.

Tsay (2013) parodo, kad nors lygtis (45) ir lygtis (46) yra pačios populiariausios VECM reprezentacijos jos nėra vienintelės.

VECM yra pagrindinis ir dažniausiai naudojamas kointegruotų procesų parametrizavimo ir specifikuojamo modelis, suteikiantis galimybę į vektorinės autoregresijos modelį (VAR) įtraukti paklaidos korekcijos komponentą ir todėl mokslinėje literatūroje VECM modelis dar vadinamas kointegruotu vektorinės autoregresijos (CVAR) pavadinimu (Lütkepohl, 2007).

2.1. Vektorinių paklaidos korekcijos modelių specifikuojama

Lütkepohl (2007) teigia, kad specifikuojant VECM turi būti nustatyta vėlavimų eilė, kointegracijos rangas ir, jei reikia, kiti apribojimai, tačiau įprasta pirmiausia nustatyti vėlavimų eilę, tada kointegracijos rangą. Kointegracijos rangas neturi įtakos vėlavimų eilės nustatymui, tačiau vėlavimų eilę dalyvauja daugelyje kointegracijos rango nustatymo procedūrų. Asteriou ir Hall (2007) pateikia tokią VECM specifikuojamo procedūrą:

1. Nustatyti kintamųjų integruotumo eilę

2. Nustatyti modelio vėlavimų eilę
3. Pasirinkti deterministinių komponentų įtraukimo į modelį būdą
4. Nustatyti kointegruojančių vektorių skaičių

2.1.1. Vėlavimų eilės pasirinkimas

Enders (2010) teigia, kad teisingas vėlavimų eilės pasirinkimas VECM modeliuose yra ypatingai svarbus: jei vėlavimų eilė bus pasirinkta per maža, modelis bus neteisingai parametrizuotas; jei vėlavimų eilė bus pasirinkta per didelė bus be reikalo prarandami laisvės laipsniai.

Makroekonominuose modeliuose vėlavimų eilės pasirinkimui įtakos taip pat turi ir a priori informacija apie makroekonominio modelio cikliškumą Enders (2008).

Juselius (2006) siūlo VECM vėlavimų eilei nustatyti galima naudoti Akaike, Hannan–Quinn ir Schwarz informacinius kriterijus. Lütkepohl (2007) prie šių kriterijų taip pat siūlo naudoti Final Prediction Error informacinį kriterijų:

1. Akaike information criterion (AIC)
2. Hannan–Quinn information criterion (HQ)
3. Schwarz information criterion (SC)
4. Final Prediction Error (FPE)

$$AIC(m) = \ln |\tilde{\Sigma}_u(m)| + \frac{2mK^2}{T} \quad (47)$$

$$HQ(m) = \ln |\tilde{\Sigma}_u(m)| + \frac{2 \ln \ln T}{T} mK^2 \quad (48)$$

$$SC(m) = \ln |\tilde{\Sigma}_u(m)| + \frac{\ln T}{T} mK^2 \quad (49)$$

$$FPE(m) = \left[\frac{T + Km + 1}{T - Km - 1} \right]^K \det \tilde{\Sigma}_u(m) \quad (50)$$

Visi informaciniai kriterijai gali būti panaudojami vėlavimų eilės pasirinkimui. Pažymėjus informacinių kriterijų AIC, HQ, SC parinktas vėlavimų eilės atitinkamai $\hat{p}(AIC)$, $\hat{p}(HQ)$, $\hat{p}(SC)$, gauname tokią informacinių kriterijų palyginomo lygtį Lütkepohl (2007):

$$\hat{p}(SC) \leq \hat{p}(HQ) \leq \hat{p}(AIC), \text{ kai } T \geq 16 \quad (51)$$

Paulsen (1984) ir Tsay (1984) parodo, kad HQ ir SC kriterijai yra suderinti ir išlaiko savo išlaiko savo savybę nustatyti teisingą vėlavimų skaičių didelėse imtyse ir modeliuose su integruotais kintamaisiais.

Lütkepohl (2007) pabrėžia, kad nors SC ir HQ yra suderinti kriterijai, o AIC ir FPE nėra suderinti, tai nereiškia, kad AIC ir FPE kriterijai yra prastesni. Ši savybė ypač svarbi mažose imtyse, nes mažose imtyse suderinamumas nėra svarbus ir DGP gali neturėti baigtinės VAR reprezentacijos.

Nors ir nėra tikslios taisyklės, kokią vėlavimų eilę naudoti, jei informaciniai kriterijai siūlo skirtingas vėlavimų eiles, praktikoje geriau naudoti šiek tiek per trumpą nei per ilgą vėlavimų eilę (Juselius, 2006).

2.1.2. Deterministinių komponentų pasirinkimas

Makroekonominėje analizėje dauguma kintamųjų turi deterministinius komponentus, dažniausiai teigiamą deterministinį trendą (Maddala ir Kim, 1998).

Neteisinga modelio specifikacija, įtraukiant per didelį kiekį deterministinių komponentų, gerokai sumažina kointegracijos rango testų galią Doornik ir kiti (1998) ir Hubrich ir kiti (2001). Johansen (1995) paklausė naudoti tikėtinumo santykio testus, įvertinti deterministinių komponentų įtraukimo į modelį hipotezes. Šie testai leidžia panaudoti maksimalaus tikėtinumo įverčius ir atitinkamą tikėtinumo santykio maksimumą visiems deterministinių komponentų įtraukimo į modelius atvejams.

Tiesinis trendas yra bendriausias deterministinių komponentų atvejis (neskaitant fiktyvių kintamųjų): $\mu_0 + \mu_1 t$. Jo tikrinimui galima sudaryti tokias hipotezes:

$$H_0 : \beta' \mu_1 = 0 \quad (52)$$

$$H_1 : \beta' \mu_1 \neq 0$$

Lygtis (52) lygina VECM modelį su trendu su VECM modeliu be trendo. Jei H_0 atmetame liekame su modeliu su trendu lygtyje apibrėžtu modeliu, jei H_0 atmesti negalime, pasirenkame modelio specifikaciją be trendo.

Kita praktikoje dažnai vertinama hipotezė yra tokia:

$$H_0 : \mu_1 = 0 \quad (53)$$

$$H_1 : \mu_1 \neq 0, \beta' \mu_1 = 0$$

$$(54)$$

Lygtis (53) lygina VECM modelį su trendu esančiu tik VAR dalyje su VECM modeliu be trendo. Jei H_0 atmetame pasirenkame modelį su trendu, jei H_0 atmesti

negalime, pasirenkame modelį be trendo.

Pažymima, kad naudojant tikėtino santykio testus specifikuoti VECM, į modeliavimo procedūrą įtraukiama papildoma paklaida, nes testai skirti VECM su nustatytais kointegravimo rangais.

Juselius (2006) išskiria penkis deterministinių komponentų įtraukimo į VECM atvejus:

1. VECM be deterministinių komponentų
2. VECM su laisvuju nariu paklaidos korekcijos komponente
3. VECM su laisvuju nariu
4. VECM su trendu paklaidos korekcijos komponente
5. VECM su trendu

Lütkepohl (2007) išskiria atvejus su poslinkio, impulso ir sezoniniais fiktyviaisiais kintamaisiais.

Johansen ir kiti (2000) pateikia detaliają analizę modelių su lūžiais deterministiniuose komponentuose.

Bendruoju atveju VECM su deterministinėmis komponentėmis gali būti užrašoma taip:

$$\Delta y_t = \Delta x_t + \mu_t \quad (55)$$

Kur x_t yra VECM reprezentaciją turintis kintamasis, kurio vidurkis yra nulis, o μ_t yra deterministiniai komponentai galintys turėti įvairias formas. Savo analizėje Juselius (2006) bendrąją deterministinių komponentų formą laiko atvejį, kai $\mu_t = \mu_0 + \mu_1 t$, o $x_t = \alpha \beta' y_{t-1}$.

2.1.3. VECM be deterministinių komponentų

VECM modelis be deterministinių komponentų turi tokią formą:

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (56)$$

Remiantis Johansen įvertinimo procedūra Johansen (1988), Johansen (1994) ir Johansen (1995) gauname ir įvertiname maksimalios tikrinės reikšmės ir pėdsako statistikas.

2.1.4. VECM su laisvuoju nariu

Jei deterministinę komponentą sudaro tik laisvasis narys $\mu_t = \mu_0$, kur μ_0 yra $(K \times 1)$ vektorius. Tada VECM turi tokią formą:

$$\Delta y_t = \Pi^o y_{t-1}^o + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (57)$$

Kur $\beta'^o[\beta' : \tau']$, $\tau' := \beta' \mu_0$, $y_{t-1}^o := \begin{bmatrix} y_{t-1} \\ 1 \end{bmatrix}$, $\Pi^o := [\Pi : v_0]$, o $v_0 := -\Pi \mu_0 = \alpha \tau'$. Toks VECM gali būti įvertinamas taip pat, kaip ir VECM modelis be deterministinių komponentų, tik vietoj y_{t-1} naudosime y_{t-1}^o .

2.1.5. VECM su tiesiniu trendu

Jei deterministinę komponentą sudaro laisvasis narys ir tiesinis trendas $\mu_t = \mu_0 + \mu_1 t$, kur μ_0 ir μ_1 yra $(K \times 1)$ vektoriai. Tada VECM būtų tokios formos:

$$\Delta y_t = v + \Pi^+ y_{t-1}^+ + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (58)$$

Kur, $v := -\Pi \mu_0 + (I_K - \Gamma_1 - \dots - \Gamma_{p-1}) \mu_1$, $\eta' := -\beta' \mu_1$, $\Pi^+ := \alpha[\beta' : \eta']$, $y_t^+ := \begin{bmatrix} y_t \\ t \end{bmatrix}$. Toks VECM gali būti įvertinamas taip pat kaip ir VECM modelis be deterministinių komponentų, tik vietoj y_{t-1} naudosime y_{t-1}^+ .

2.1.6. VECM su tiesiniu trendu paklaidos korekcijos komponente

Lygties (58) modelyje esantys deterministiniai komponentai nėra apriboti, taigi jie gali dalyvauti ir paklaidos korekcijos komponente. Formaliai toks atvejis reiškia, kad $\mu_1 \neq 0$ ir $\Pi \mu_1 = \alpha \beta' \mu_1 = 0$, kur $\beta' \mu_1 \neq 0$. Tokia situacija reiškia, kad trendas μ_1 yra nenulis ir yra ortogonalus kointegruojančiam vektoriui. Tokiu atveju VECM įgauna tokią formą:

$$\Delta y_t = v + \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (59)$$

Reiktų atkreipti dėmesį, kad Π negali turėti pilno rango, nes $\mu_1 \neq 0$, o $\Pi \mu_1 = 0$, taigi $rk(\Pi) < K$.

2.1.7. VECM su fiktyviais kintamaisiais

Fiktyviusius kintamuosius galima išskirti į tris grupes:

1. Sezoniniai fiktyvūs kintamieji
2. Poslinkio fiktyvūs kintamieji
3. Impulso fiktyvūs kintamieji

2.1.8. Deterministiniai komponentai kointegruotuose procesuose

Deterministiniai komponentai yra viena iš sudedamųjų VECM modelio dalių. Deterministiniai komponentai VAR procese su vienetinėmis šaknimis turi iš esmės turėti kitokį poveikį, nei stabiliam VAR procese, pavyzdžiui laisvasis narys atsitiktinio klaidžiojimo procese generuoja linijinį trendą, o stabiliam AR procese jis generuoja pastovų vidurkį. Siekiant įvertinti parodyti VECM deterministinių komponentų specifikavimo būdus naudosime tokią modelio formą:

$$y_t = \mu_t + x_t \quad (60)$$

kur x_t yra $VECM(p)$, kurio vidurkis yra nulis su galimai kointegruotais kintamaisiais, o μ_t yra deterministinis komponentas. Galimos keletas deterministinio komponento μ_t formų:

1. $\mu_t = \mu_0$, tokiu atveju deterministinis komponentas yra konstanta;
2. $\mu_t = \mu_0 + \mu_1 t$, kur μ_0 ir μ_1 yra fiksuoti K -dimensijų parametru vektoriai;
3. Sezoniniai fiktyvūs kintamieji
4. Poslinkio fiktyvūs kintamieji
5. Impulso fiktyvūs kintamieji

Lygties (60) užrašymas suponuoja, kad tiesiogiai stebimo y_t vidurkis priklauso tik nuo deterministinio komponento μ_t . Stacionarų x_t komponentą iš lygties (60) galima užrašyti taip:

$$\begin{aligned} \Delta x_t &= \Pi x_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta x_{t-i} + u_t \\ \Delta x_t &= \alpha \beta' x_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta x_{t-i} + u_t \end{aligned} \quad (61)$$

Kur $\Pi = \alpha\beta'$. Įvertiname pirmąjį variantą, kai $\mu_t = \mu_0$. Tokiu atveju iš lygties (60) gauname $x_t = y_t - \mu_0$ ir $\Delta x_t = \Delta y_t$. Iš to gauname:

$$\begin{aligned}\Delta y_t &= \alpha\beta'(y_{t-1} - \mu_0) + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \\ &= \alpha\beta'^o \begin{bmatrix} y_{t-1} \\ 1 \end{bmatrix} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \\ &= \Pi^o y_{t-1}^o + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t\end{aligned}\quad (62)$$

Kur $\beta'^o := [\beta' : \tau']$, $\tau' := -\beta'\mu_0$, $y_{t-1}^o := \begin{bmatrix} y_{t-1} \\ 1 \end{bmatrix}$, $\Pi^o := [\Pi : v_0]$, o $v_0 := -\Pi\mu_0 = \alpha\tau'$. Taigi pirmuoju atveju, kai $\mu_t = \mu_0$ konstantą galima įtraukti į paklaidos korekcijos komponentą. Kitas būdas yra įtraukti μ_0 tokiu būdu:

$$\begin{aligned}\Delta y_t &= v_0 + \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \\ \Delta y_t &= v_0 + \alpha\beta' y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t\end{aligned}\quad (63)$$

Kur $v_0 = \alpha\tau'$. Šis apribojimas yra reikalingas, kad nebūtų generuojamas linijinis trendas y_t kintamųjų vidurkiuose. Iš (60) lygties aišku, kad konstanta negali generuoti linijinio trendo.

Antruoju atveju, kai $\mu_t = \mu_0 + \mu_1 t$, gauname $x_t = y_t - \mu_0 - \mu_1 t$ ir $\Delta x_t = \Delta y_t - \mu_1$. Iš to gauname:

$$\Delta y_t - \mu_1 = \alpha\beta'(y_{t-1} - \mu_0 - \mu_1(t-1)) + \sum_{i=1}^{p-1} \Gamma_i (\Delta y_{t-i} - \mu_1) + u_t \quad (64)$$

$$\begin{aligned}\Delta y_t &= v + \alpha[\beta' : \eta'] \begin{bmatrix} y_{t-1} \\ t-1 \end{bmatrix} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \\ &= v + \Pi^+ y_{t-1}^+ + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t\end{aligned}\quad (65)$$

Kur, $v := -\Pi\mu_0 + (I_K - \Gamma_1 - \dots - \Gamma_{p-1})\mu_1$, $\eta' := -\beta'\mu_1$, $\Pi^+ := \alpha[\beta' : \eta']$,

$y_t^+ := \begin{bmatrix} y_t \\ t \end{bmatrix}$ Šiuo atveju laisvasis narys v nėra apribotas ir gali įgauti visas reikšmes iš \mathbb{R}^K , priklausomai nuo pradinių kintamųjų μ_0, μ_1 ir kitų parametru. Šią lygtį galima perrašyti neįtraukiant trendo komponento į paklaidos korekcijos komponentą, tokiu atveju modelis įgautų tokį pavidalą:

$$\Delta y_t = v_0 + v_1 t + \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (66)$$

Tokiu pavidalu užrašytas modelis gali generuoti kvadratinius trendus kintamųjų vidurkiuose. Kitas galimas variantas, kai μ_1 yra ortogonalus kointegracijos matricai taip, kad $\beta' \mu_1 = 0$ ir iš to seka, kad $\eta = 0$ ir nebelieka trendo komponento paklaidos korekcijos komponente. Tokia situacija gali nutikti ir tada, kai $\mu_1 \neq 0$ ir kintamųjų vidurkiai turi linijinius trendus. Tokiu atveju šiuos trendus generuotų v . Tokiu atveju gautume modelį:

$$\Delta y_t = v + \alpha \beta' y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (67)$$

$$\Delta y_t = v + \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (68)$$

Kur v nebūtų apribotas. Toks modelis aprašo situaciją, kai linijinis trendas yra kintamuosiuose, bet ne VECM. Svarbu pastebėti, kad tokiu atveju kointegracijos rangas turi būti mažesnis už K . Jei procesas turi kointegravimo rangą K , jis yra stabilus ir todėl negali generuoti trendo kai tik intercept yra įtrauktas į modelį. Formaliai kointegravimo matrica β , kurios rangas yra K nėra singuliari, todėl $\beta' \mu_1$ negali būti nulis, jei μ_1 nėra nulis. Papildomas privalumas užrašant lygtį $y_t = \mu_t + x_t$ forma yra tas, kad Beveridge-Nelson y_t lygties reprezentacija yra gaunama pridendant deterministinį komponentą prie Beveridge-Nelson x_t reprezentacijos.

Hendry ir Juselius (2001) išskiria penkis deterministinio komponento atvejus. Jei turime tokį supaprastintą modelį tik su vienu kintamųjų vektoriaus vėlavimu:

$$\Delta x_t = \alpha \beta' x_{t-1} + \pi + \delta t + \varepsilon_t \quad (69)$$

Perrašius π ir δ iš (69) lygties tokiu pavidalu:

$$\begin{aligned}\pi &= \alpha\mu + \gamma \\ \delta &= \alpha\rho + \tau\end{aligned}\tag{70}$$

Ištačius (70) lygtį į (69) lygtį gauname:

$$\begin{aligned}\Delta x_t &= \alpha\beta'x_{t-1} + \alpha\mu + \gamma + \alpha\rho t + \tau t + \varepsilon_t \\ \Delta x_t &= \alpha(\beta' : \mu : \rho) \begin{bmatrix} x_{t-1} \\ 1 \\ t \end{bmatrix} + (\gamma + \tau t) + \varepsilon_t \\ \Delta x_t &= \alpha \begin{bmatrix} \beta \\ \mu' \\ \rho' \end{bmatrix}' x_{t-1}^* + (\gamma + \tau t) + \varepsilon_t\end{aligned}\tag{71}$$

Kur $x_{t-1}^* = (x'_{t-1}, 1, t)'$. μ ir ρ galime pasirinkti taip, kad pusiausvyros paklaida $(\beta^*)'x_t^* = v_t$ turi vidurkį lygų nuliui (kur $(\beta^*) = (\beta', \mu, \rho)'$), taigi trendas iš (71) lygties gali būti išreikštas:

$$E[\Delta x_t] = \gamma + \tau t\tag{72}$$

Taigi, jei $\gamma \neq 0$ kintamasis x_t turi trendą, o jei $\tau \neq 0$ kintamojo pokyčiai Δx_t turi trendą, o pats kintamasis x_t turi kvadratinį trendą.

Iš (71) lygties Hendry ir Juselius (2001) suformuoja penkis deterministinio komponento įtraukimo į modelį atvejus:

1. δ ir π yra neapriboti
2. $\tau = 0$, o γ , μ ir ρ yra neapriboti
3. $\delta = 0$, o π yra neapribotas
4. $\gamma = 0$ ir $\delta = 0$, o $\mu \neq 0$
5. $\pi = 0$ ir $\delta = 0$

Pirmuoju atveju $\gamma \neq 0$, $\mu \neq 0$, $\rho \neq 0$ ir $\tau \neq 0$. μ yra laisvasis narys paklaidos korekcijos komponente, todėl jis veikia tik kintamojo x_t vidurkį. ρ yra paklaidos korekcijos komponento trendas. γ yra kintamojo pokyčių Δx_t laisvasis narys. Lygtis (70) parodo skirtumą tarp γ ir ρ - γ yra priešpadaugintas iš paklaidų korekcijos

parametrų matricos α ir todėl dalyvauja tik paklaidos korekcijos komponente, o ρ dalyvauja likusiame komponente, bet nedalyvauja paklaidos korekcijos komponente. τ yra kintamojo pokyčių Δx_t trendas.

2.2. Vektorinių paklaidos korekcijos modelių įvertinimas

Vektorinių paklaidos korekcijos modelių įvertinimo metodų mokslinėje literatūroje aprašoma nemažai. Savo įvertinimo procedūras pateikė Johansen (1988) ir Engle ir Granger (1987), Lucas (1998) ir Boswijk ir Lucas (2002) pristatė testus nenormaliesiems procesams, Gonzalo ir Lee (1998) ir Breitung ir Hassler (2002) tyrė aukštesne nei pirma eile integruotus ir ilgiosios atminties procesus. Phillips (1991) pristatė savo beparametrį trumpojo laikotarpio parametrų įvertinimą, Stock ir Watson (1988) pristatė įvertinimo procedūrą paremtą pagrindiniais komponentais, o Bossaerts (1988) rėmėsi kanoninėmis koreliacijomis. Hubrich ir kiti (2001) pateikė išsamią VECM įvertinimo modelių analizę. Šiame darbe išskirsime tik šias pagrindines daugiamačiams modeliams pritaikomas procedūras.

1. Engle - Granger metodas
2. Johansen procedūra

Toliau bus aptarti kiekvienas iš šių VECM įvertinimo metodų.

2.2.1. Engle - Granger metodas

Engle ir Granger (1987) pristatė savo dviejų pakopų VECM įvertinimo metodą, kuris dviejų kintamųjų y_{1t} ir y_{2t} , kurie abu yra $I(1)$ VECM gali būti užrašytas tokia forma:

$$y_{1t} = \beta y_{2t} + u_{1t} \quad (73)$$

$$y_{2t} = \alpha y_{1t} + u_{2t} \quad (74)$$

kur u_t yra $I(0)$. Tokio modelio ypatybes Maddala ir Kim (1998) ir Enders (2008) apibendrino taip:

1. β unikalumas. Nors lygtyje (74) aprašytas modelis turi unikalų β , modeliuose su daugiau kintamųjų β nėra unikalus.
2. $\hat{\beta}$ super-suderintas. Maddala ir Kim (1998) parodo, kad parametro β MKM įvertis $\hat{\beta}$ artėja prie tikrosios reikšmės T greičiu (super-suderinamumas), o ne įprastu \sqrt{T} greičiu (suderinamumas), nes $\sum y_{2t}^2$ yra $O_p(T^2)$, jei y_{2t} yra $I(1)$. Ši savybė galioja, jei modelis turi didesnę kiekį $I(1)$ kintamųjų.

3. Nors $\hat{\beta}$ yra super-suderintas Banerjee ir kiti (1986) ir Banerjee ir kiti (1993) pasinaudodami Monte Carlo simuliacijomis parodo, kad egzistuoja didelis poslinkis, kai imties dydis nėra didelis.
4. Normalizacija. Lygtys (74) yra normalizuota y_1t atžvilgiu. Perrašius lygtį (74) normalizuojant y_2t atžvilgiu gautume (74) lygtį. $\hat{\alpha}$ yra super-suderintas α įvertis ir tuo pačiu super-suderintas β įvertis, nes $\alpha = 1/\beta$. Jei R^2 yra artimas 1, tada $\hat{\beta} \simeq 1/\hat{\alpha}$. Hendry (1986) teigia, kad tai galioja daugumai ekonominių kintamųjų. Ng ir Perron tyrimas parodė, kad parametrų įverčiai gali būti beveik be poslinkių normalizavus juos pagal vieną iš kintamųjų ir turėti didelį poslinkį normalizavus juos pagal kitą kintamąjį Ng ir Perron (1997). Taip nutinka, kai vienas iš kintamųjų yra beveik stacionarus ir siūloma mažiau integruotą kintamąjį naudoti kaip priklausomą kintamąjį.
5. Dviejų pakopų įvertinimo procedūra turi didelį trūkumą, nes pirmoje pakopoje įvertintos paklaidos \hat{u}_t antroje pakopoje yra naudojamos įvertinti tokios formos regresiją $\Delta\hat{u}_t = a_1\hat{u}_{t-1} + \dots$

2.2.2. Johansen procedūra

Johansen (1988) paklausė procedūrą VECM parametrams įvertinti ir nustatyti kointegruojančių vektorių skaičių. Procedūra generuoja dvi testo statistikas - pėdsako statistiką ir maksimalios tikrinės reikšmės statistiką.

Pėdsako statistika įvertina H_0 , kuri teigia, kad yra ne daugiau nei r kointegruojančių vektorių, prieš H_1 , kuri teigia, kad yra daugiau nei r kointegruojančių vektorių. Maksimalios tikrinės reikšmės statistika įvertina H_0 , kuri teigia, kad yra lygiai r kointegruojančių vektorių, prieš H_1 , kuri teigia, kad yra $r + 1$ kointegruojančių vektorių. Johansen ir Juselius (1990) rekomenduoja naudoti maksimalios tikrinės reikšmės testą.

Jei procesas y_t turi normalųjį skirstinį, tai $u_t \sim N(0, \Sigma_u)$, o VECM yra išreikštas tokia forma:

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + u_t \quad (75)$$

kur $\Pi = \alpha\beta'$. Lygtis (75) gali būti įvertintas naudojant Johansen maksimalaus tikėtimumo procedūrą. Jei imties dydis yra T logaritmuota tikėtimumo funkcija yra:

$$\ln l = -\frac{KT}{2} \ln 2\pi - \frac{T}{2} \ln |\Sigma_u| - \frac{1}{2} \text{tr}[(\Delta Y - \alpha\beta'Y_{-1} - \Gamma\Delta X)' \Sigma_u^{-1} (\Delta Y - \alpha\beta'Y_{-1} - \Gamma\Delta X)] \quad (76)$$

Įvertinant VECM naudojant Johansen procedūrą laikoma, kad $rk(\Pi) = r$, taigi matrica Π gali būti išreikšta dviejų vektorių α ir β , kurie abu yra $(K \times r)$, o $rk(\alpha) = rk(\beta) = r$ sandauga $\Pi = \alpha\beta'$. Jei pažymėsime $M := I_T - \Delta X'(\Delta X\Delta X')^{-1}\Delta X$, $R_0 := \Delta Y M$, o $R_1 := Y_{-1}M$, o $S_{ij} := R_i R_j' / T$, $i = 0, 1, \dots$, $\lambda_1 \geq \dots \geq \lambda_K$ yra tikrinės $S_{11}^{-1/2} S_{10} S_{00}^{-1} S_{01} S_{11}^{-1/2}$ reikšmės, o v_1, \dots, v_K yra atitinkami ortogoniniai tikriniai vektoriai. Lygtis (76) maksimizuojama kai:

$$\begin{aligned} \beta &= \tilde{\beta} := [v_1, \dots, v_r]' S_{11}^{-1/2} \\ \alpha &= \tilde{\alpha} := \Delta Y M Y_{-1}' \tilde{\beta} (\tilde{\beta}' Y_{-1} M Y_{-1}' \tilde{\beta})^{-1} = S_{01} \tilde{\beta} (\tilde{\beta}' S_{11} \tilde{\beta})^{-1} \\ \Gamma &= \tilde{\Gamma} := (\Delta Y - \tilde{\alpha} \tilde{\beta}' Y_{-1}) \Delta X' (\Delta X \Delta X')^{-1} \\ \Sigma_u &= \tilde{\Sigma}_u := (\Delta Y - \tilde{\alpha} \tilde{\beta}' Y_{-1} - \tilde{\Gamma} \Delta X) (\Delta Y - \tilde{\alpha} \tilde{\beta}' Y_{-1} - \tilde{\Gamma} \Delta X)' / T \end{aligned} \quad (77)$$

$$\max \ln l = -\frac{KT}{2} \ln 2\pi - \frac{T}{2} \left[\ln |S_{00}| + \sum_{i=1}^r \ln(1 - \lambda_i) \right] - \frac{KT}{2} \quad (78)$$

Sprendiniai $\tilde{\alpha}$ ir $\tilde{\beta}$ nėra unikalūs, bet kokiai nevienetinei matricai Q , nes $\tilde{\alpha}Q^{-1}$ ir $\tilde{\beta}Q'$ yra dar vienas sprendinių rinkinys. Taip pat verta pastebėti, kad, procedūra galioja net kai $r = K$.

Asimptotinės įverčių sąvybės.

$$\sqrt{T} \text{vec}([\tilde{\alpha}\tilde{\beta}' : \tilde{\Gamma}] - [\Pi : \Gamma]) \xrightarrow{d} N(0, \Sigma_{co}) \quad (79)$$

kur

$$\Sigma_{co} = \left(\begin{bmatrix} \beta & 0 \\ 0 & I_{Kp-K} \end{bmatrix} \Omega^{-1} \begin{bmatrix} \beta' & 0 \\ 0 & I_{Kp-K} \end{bmatrix} \otimes \Sigma_u \right) \quad (80)$$

ir

$$\sqrt{T} \text{vech}(\tilde{\Sigma}_u - \Sigma_u) \xrightarrow{d} N(0, D_K^+(\Sigma_u \otimes \Sigma_u) D_K'^+) \quad (81)$$

Silpnojo egzogeniškumo testas. Po to, kai yra nustatomas Π matricos rangas ir kointegruojančių vektorių skaičius, reikia įvertinti silpnojo egzogeniškumo sąlygą (Asteriou ir Hall, 2007). $\Pi = \alpha\beta'$, kur α parodo prisitaikymo koeficientų įverčius, o β parodo ilgojo laikotarpio koeficientų matrica. Asteriou ir Hall (2007) silpnąjį egzogeniškumą apibrėžia taip: kintamasis Z yra silpnai egzogeniškas, jei jis priklauso tik nuo savo vėlavimų ir yra nepriklausomas nuo kitus kintamuosius generuojančių parametrų.

Linijiniai kointegruojančiojo vektoriaus apribojimai. Svarbi Johansen procedūros sąlybė yra galimybė gauti α ir β matricų koeficientų įverčius ir jiems pritaikyti linijinius apribojimus. β matricos parametrai leidžia patikrinti įvairias kintamųjų tarpusavio sąveikų hipotezes (Asteriou ir Hall, 2007).

Johansen procedūra su deterministiniais komponentais.

$$y_t = \mu_t + x_t \quad (82)$$

Kur x_t yra VECM reprezentaciją turintis kintamasis, kurio vidurkis yra nulis, o μ_t yra deterministinis komponentas. Bendruoju atveju deterministinis komponentas gali turėti konstantas, polinominius trendus, sezoninius ir kitokius fiktyvius kintamuosius. Išreiškus Δy_t tokiu pavidalu:

$$\begin{aligned} \Delta y_t &= \alpha[\beta' : \eta'] \begin{bmatrix} y_{t-1} \\ D_{t-1}^{co} \end{bmatrix} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + CD_t + u_t \\ &= \Pi^+ y_{t-1}^+ + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + CD_t + u_t \end{aligned} \quad (83)$$

Kur D_{t-1}^{co} talpina visus paklaidos korekcijos komponente esančius deterministinius komponentus, o D_t talpina visus likusius (nedalyvaujančius kointegravimo santykiuose) deterministinius komponentus, o η' ir C yra atitinkamos parametrų matricos. $\Pi^+ := \alpha[\beta' : \eta'] = \alpha\beta^{+'}$ ir $y_t^+ := \begin{bmatrix} y_t \\ D_t^{co} \end{bmatrix}$

Reiktų atkreipti dėmesį, kad konkretus deterministinis komponentas patenka arba į D_t^{co} arba į D_t , bet ne į abu.

Tolesnis įvertinimas vyksta kaip ir be deterministinių komponentų:

$$\begin{aligned}
Y_{-1}^+ &:= [y_0^+, \dots, y_{t-1}^+] \\
\Gamma^+ &:= [\Gamma_1, \dots, \Gamma_{p-1}, C] \\
\Delta X^+ &:= [\Delta X_0^+, \dots, \Delta X_{t-1}^+] \\
\Delta X_{t-1}^+ &:= \begin{bmatrix} \Delta y_{t-1} \\ \vdots \\ \Delta y_{t-p+1} \\ D_t \end{bmatrix}
\end{aligned} \tag{84}$$

Gauname tokią mažiausių kvadratų išraišką:

$$[\hat{\Pi}^+ : \hat{\Gamma}^+] = [\Delta Y Y_{-1}^{+'} : \Delta Y \Delta X^{+'}] \begin{bmatrix} Y_{-1}^+ Y_{-1}^{+'} & Y_{-1}^+ \Delta X^{+'} \\ \Delta X^+ Y_{-1}^{+'} & \Delta X^+ \Delta X^{+'} \end{bmatrix} \tag{85}$$

Asimptotinė teorija reikalauja dėmesio, nes konvergavimo tempas priklauso nuo konkrečių deterministinių komponentų. Pavyzdžiui, jei įtrauksime linijinį trendą, susieto parametro konvergavimo tempas nebus \sqrt{T} i-tasis D_t komponentas gali būti įtrauktas į paklaidos korekcijos komponentą, jei i-tasis koeficientų matricos C stulpelis (C_i) tenkina apribojimą $C_i = \alpha \eta_i$ ir nulinė hipotezė yra $\alpha' C_i = 0$. Taigi, yra $K - r$ apribojimų kiekvienam komponentui, kuris yra apribotas kointegracijos santykiuose. Juos nesunku patikrinti, nes didžiausio tikėtimumo įverčiai ir maksima gali būti surasti tiek apribotam, tiek ir neapribotam modeliui tiesiog specifikuojant D_t^o ir D_t . Jei m deterministinių komponentų yra apriboti kointegracijos santykiuose, tikėtimumo santykio (LR) statistika turi asimptotinį $\chi^2(m(K - r))$ skirstinį.

2.2.3. Kointegracijos rango nustatymas su struktūriniu lūžiu

Lütkepohl ir kiti (2004) paklausė VECM įvertinimo procedūrą, kai duomenis generuojantis procesas turi struktūrinį lūžį. Struktūriniai lūžiai gali turėti įtakos atliekant kintamųjų kointegravimo testus. Struktūriniai lūžiai paveikia šių testų rezultatus tokiais atvejais:

1. Kai kintamieji nėra kointegruoti ir struktūrinis lūžis paveikia visus kintamuosius vienodai.
2. Kai kintamieji yra kointegruoti ir struktūrinis lūžis paveikia ne visus kintamuosius arba paveikia juos ne vienoda apimtimi.

Pirmuoju atveju, kai struktūriniai lūžiai paveikia nekointegruotus kintamuosius, struktūrinis lūžis supanašina nekointegruotų laiko eilučių dinamiką ir įprasti kointegracijos testai gali nustatyti, kad kintamieji yra kointegruoti.

Antruoju atveju, kai struktūriniai lūžiai paveikia kointegruotus kintamuosius, tačiau struktūrinio lūžio poveikis yra skirtingas, įprasti kointegracijos testai gali nustatyti, kad kintamieji nėra kointegruoti.

Lütkepohl ir kiti (2004) procedūra yra tokia:

1. Išskaidyti duomenis generuojantį procesą į deterministinę ir stochastinę dalis;
2. Išmatuoti deterministinę proceso dalį;
3. Išmatuoti stochastinę proceso dalį;

$$y_t = \mu_0 + \mu_1 t + \delta d_{t\tau} + x_t \quad (86)$$

kur $d_{t\tau}$ yra fiktyvus kintamasis, kuris įgauna reikšmę 0, kai $t < \tau$ ir 1, kai $t \geq \tau$, o x_t yra kointegruotų kintamųjų VECM be deterministinių komponentų.

Struktūrinio lūžio data τ nėra žinoma ir yra išreikšta kaip fiksuota imties dydžio dalis.

$$\tau = [T\lambda], \text{ kur } 0 < \underline{\lambda} \leq \lambda \leq \bar{\lambda} < 1 \quad (87)$$

kur $\underline{\lambda}$ ir $\bar{\lambda}$ yra realieji skaičiai. Struktūrinis lūžis negali įvykti nei pačioje imties pradžioje, nei pačioje imties pabaigoje.

Struktūrinio lūžio datos nustatymas. Struktūrinis lūžis nustatomas tokios regresijos pagalba

$$y_t = \nu_0 + \nu_1 t + \delta d_{t\tau} + \sum_{i=1}^p A_i y_{t-1} + \varepsilon_{t\tau} \quad (88)$$

kur A_i yra $(K \times K)$ koeficientų matrica. Pfaff (2008) pastebi, kad į 88 taip pat galima įtraukti ir kitus deterministinius komponentus, pvz. sezoninį fiktyvų komponentą ir pan.

Struktūrinio lūžio data tada yra nustatoma taip

$$\hat{\tau} = \arg \min_{\tau < \Upsilon} \det \left(\sum_{t=p+1}^T \hat{\varepsilon}_{t\tau} \hat{\varepsilon}'_{t\tau} \right) \quad (89)$$

kur $\Upsilon = [T\underline{\lambda}, T\bar{\lambda}]$.

Po to, kai yra nustatoma struktūrinio lūžio data, modelis yra pertvarkomas į tokią formą:

$$\hat{x}_t = y_t - \hat{\mu}_0 - \hat{\mu}_1 t - \hat{\delta} d_{t\hat{\tau}} \quad (90)$$

Po kointegruojančių vektorių skaičius yra nustatomas naudojant Johansen procedūrą, tačiau naudojant specialias Lütkepohl ir kiti (2004) pateiktas kritines reikšmes.

2.3. VECM diagnostika

VECM diagnostika yra svarbi VECM specifikuojimo procedūra. Ši procedūra yra alternatyvi vienamačių Box - Jenkins modelių diagnostikos procedūroms ir VAR modelių procedūroms. Lütkepohl (2007) siūlo tokią VECM diagnostikos procedūrą:

1. Paklaidų autokoreliacijos tikrinimas
2. Paklaidų nenormalumo tikrinimas
3. Struktūrinių pokyčių tikrinimas

2.3.1. Paklaidų autokoreliacijos tikrinimas

Brüggemann ir kiti (2006) parodo, kad portmanteau ir Lagrange daugiklių testai gali būti naudojami VECM paklaidų autokoreliacijų tikrinimui. Taip pat reiktų išanalizuoti paklaidų autokorelogramas.

2.3.2. Paklaidų nenormalumo tikrinimas

Paklaidų nenormalumui tikrinti tinka tokie patys testai, kaip ir VAR modeliuose. Mokslinėje literatūroje dažniausiai rekomenduojamas Jarque–Bera testas.

2.3.3. Struktūrinių pokyčių tikrinimas

Chow testai. Chow testas skirtas VECM tikrinimui yra toks pati kaip ir VAR atveju. Tokiu atveju duomenų imtis yra suskirstoma į dvi dalis: imtį iki struktūrinio

lūžio ir imtį po struktūrinio lūžio. Tada abi šios imtys yra įvertinamos atskirai. Jei įvertintų parametrų reikšmės yra panašios, laikoma, kad struktūrinio lūžio nėra, tačiau, jei testų rezultatai skiriasi (galime atmesti H_0 , kuri teigia, kad rezultatų skirtumas yra 0) kintamasis turi struktūrinį lūžį. Formaliai Chow testo metodiką aprašo Enders (2010).

Prognozavimo testai. Lütkepohl (2007) išskiria prognozavimo testą, kaip būdą struktūriniam pokyčiams VECM nustatyti. Šis testas gali būti naudojamas kartu su Chow testu.

2.4. Struktūriniai VECM

VECM modeliai, kaip ir VAR modeliai, turi struktūrinės formos. Skirtingai nei VAR modeliuose VECM modeliuose ilgojo laikotarpio ryšiai tarp kintamųjų yra gaunami iš paklaidos korekcijos komponento.

Kadangi galima VECM modelį paversti VAR modeliu, struktūrinio VECM analizė yra tokia pat kaip ir struktūrinio VAR analizė (Lütkepohl, 2007).

2.4.1. Struktūrinių VECM įvertinimas

Lütkepohl (2007) ir Pfaff (2008) VECM modelio struktūrinę formą užrašo taip:

$$\Delta y_t = \Pi y_{t-1} + \Gamma_1 \Delta y_{t-1} + \cdots + \Gamma_{p-1} y_{t-p+1} + B \varepsilon_t \quad (91)$$

kur $\Pi = \alpha \beta'$, $B \varepsilon_t = u_t$ ir $\varepsilon_t \sim N(0, I_K)$. Beveridge-Nelson slankiuųjų vidurkių išskaidymas leidžia išskaidyti šį procesą taip:

$$y_t = \Xi \sum_{i=1}^t u_i + \sum_{j=0}^{\infty} \Xi_j^* u_{t-j} + y_0^* \quad (92)$$

kur $\Xi \sum_{i=1}^t u_i$ yra bendrasis trendas, o $\sum_{j=0}^{\infty} \Xi_j^* u_{t-j}$ yra stacionari ir į 0 artėjanti dalis, kai j artėja į 0. y_0^* reprezentuoja pradinės sistemos reikšmes.

Ξ matrica yra apibrėžtas 44 lygtyje. Ξ matricos rangas yra $K - r$, kur r yra kointegruojančių vektorių skaičius. Nustatę Π matricos rangą remdamiesi 91 galime teigti, kad ne daugiau nei r struktūrinių paklaidų turi ilgalaikį poveikį sistemai. Tai reiškia, kad ne daugiau nei r Ξ matricos elementų gali būti apriboti ir prilyginti nuliui. Matrica ΞB parodo ilgojo laikotarpio struktūrinių šokų poveikį, tuo tarpu vienalaikius poveikius parodo B matrica. Modeliui reikia suteikti bent $\frac{K(K-1)}{2}$ apribojimų skaičių. $\Pi = \alpha \beta'$ suteikia modeliui $r(K - r)$ apribojimų. Likę apribojimai

gali būti suteikiami abiems matricoms, tačiau ne mažesnis nei $\frac{r(r-1)}{2}$ apribojimų skaičius privalo būti suteiktas vienalaikių poveikių matricai B .

Juselius (2006) struktūrinius VECM užrašo tokia forma:

$$A_0 \Delta y_t = a \beta' x_{t-1} + A_1 \Delta y_{t-1} \tilde{\Phi} D_t + u_t, \quad u_t \sim IN(0, \Sigma) \quad (93)$$

Šio struktūrinio VECM redukuota forma:

$$\Delta y_t = \alpha \beta' x_{t-1} + \Gamma_1 \Delta y_{t-1} + \Phi D_t + \varepsilon_t, \quad \varepsilon_t \sim IN(0, \Omega) \quad (94)$$

Redukuotos formos parametrus galima užrašyti taip: $\lambda_{RF} = \{\alpha, \beta, \Gamma_1, \Phi, \Omega\}$, o struktūrinės formos parametrus taip: $\lambda_{SF} = \{A_0, a, \beta, A_1, \tilde{\Phi}, \Sigma\}$. Juselius (2006) toliau skirsto parametrus į redukuotus ilgojo laikotarpio $\lambda_{RF}^L = \beta$, struktūrinius ilgojo laikotarpio $\lambda_{SF}^L = \beta$, redukuotus trumpojo laikotarpio $\lambda_{RF}^S = \{\alpha, \Gamma_1, \Phi, \Omega\}$ ir struktūrinius trumpojo laikotarpio $\lambda_{SF}^S = \{A_0, a, A_1, \tilde{\Phi}, \Sigma\}$. Redukuotų ir struktūrinių parametrų sąryšis: $\Gamma_1 = A_0^{-1} A_1$, $\alpha = A_0^{-1} a$, $\varepsilon_t = A_0^{-1} u_t$, $\Phi = A_0^{-1} \tilde{\Phi}$, $\Omega = A_0^{-1} \Sigma A_0'^{-1}$. λ_{RF}^S gali būti įvertinti be apribojimų, tuo tarpu, kad būtų galima įvertinti λ_{SF}^S sistemai reikia suteikti $p(p-1)$ apribojimų.

2.4.2. Impulso atsako funkcijos

Vektorinių paklaidos korekcijos modelių analizėje β vektorius parodo kintamųjų ilgojo laikotarpio pusiausvyrą.

$$\beta_1 y_{1t} + \beta_2 y_{2t} + \dots + \beta_K y_{Kt} = 0 \quad (95)$$

Įprastai 95 yra normalizuojama pirmojo kintamojo atžvilgiu (kai $\beta_1 \neq 0$):

$$y_{1t} = -\frac{\beta_2}{\beta_1} y_{2t} - \dots - \frac{\beta_K}{\beta_1} y_{Kt} \quad (96)$$

Lütkepohl (2007) pastebi, kad jei y_2 pasikeis per 1, ilguoju laikotarpiu y_1 pasikeis ne $-\frac{\beta_2}{\beta_1}$ dydžiu, bet dar ir priklausomai nuo atitinkamo VAR(p) arba VECM modelio. Šokas y_2 kintamajame taip pat paveikia ir kitus kintamuosius, kurie savo ruožtu paveikia y_1 , todėl ilgojo laikotarpio vienietinis pokytis y_2 kintamajame nėra $-\frac{\beta_2}{\beta_1}$.

3. Johansen procedūrų specifikacijos analizė naudojant Monte Carlo simuliacijas

2.2 skyriuje aprašytų VECM įvertinimo procedūrų pagrindinis tikslas yra teisingai įvertinti VECM modelių Π matricos rangą - t.y. nustatyti ar kintamieji yra kointegruoti ir jei taip, kiek jie turi kointegruojančių vektorių. Teisingas kointegruojančių vektorių skaičiaus nustatymas yra būtina prielaida teisingam modelio parametrų įvertinimui. Jei kointegruojančių vektorių skaičius yra nustatomas neteisingai, kointegracijos parametrai yra įvertinami neteisingi, todėl neteisingai įvertinama ir ilgojo laikotarpio pusiausvyra. Klaidingas ilgojo laikotarpio pusiausvyros nustatymas veda ne tik prie klaidingų prognozių, bet ir prie klaidingo kintamųjų tarpusavio ryšio nustatymo. Šiame skyriuje bus pristatytos Monte Carlo simuliacijos ir jų rezultatai tiriant kokią įtaką turi deterministiniai komponentai Johansen procedūrų nustatamam kointegruojančių vektorių skaičiui.

3.1. Tyrimo aprašymas

Pagrindinis šio tyrimo tikslas yra išsiaiškinti, kokią įtaką Johansen procedūra nustatamam kointegruojančių vektorių skaičiui daro neteisinga VECM deterministinių komponentų specifikacija. Šiam tikslui pasiekti buvo modeliuojami sintetiniai VECM procesai, pagal iš anksto paruoštus parametrų rinkinius, kuriems paskui buvo atliekama Johansen procedūra. Siekiant kuo geriau ištirti kokią įtaką Johansen procedūros nustatamam kointegruojančių vektorių skaičiui daro VECM deterministinių komponentų specifikacija, kai kurie modelių parametrai buvo apriboti. Buvo tiriami tik dvimačiai ir trimačiai modeliai su ketvirta vėlavimų eile. Buvo tiriami procesai, kai imties dydis yra tarp 20 ir 200 ir kiekvienas procesas buvo simuliuojamas 1000 kartų, o tada įvertinama, kiek kointegruojančių vektorių nustatė konkreti Johansen procedūra. Atvejis kai kintamieji yra stacionarūs gali būti įvertinamas VAR modeliais. Vienetinių šaknų testai leidžia nustatyti ar VAR modelio kintamieji yra stacionarūs. Tuo tarpu atvejis, kai $rank(\Pi) = 0$ yra dažnas praktikoje, todėl bus tiriami ir nekointegruoti VAR modeliai.

Akivaizdu, kad tiriami modeliai turės konkrečią deterministinių komponentų specifikaciją ir tik viena iš Johansen procedūros specifikacijų bus teisinga. Kadangi, praktikoje kartais sudėtinga nustatyti, kurią Johansen procedūros specifikaciją naudoti buvo ištirta ir neteisingos Johansen procedūros naudojimo įtaka kointegruojančių vektorių skaičiaus nustatymui.

Imties dydis daro didelę įtaką testų rezultatams. Nuo šiol bus laikoma, kad mažos imtys yra tada, kai stebėjimų skaičius yra mažesnis už 50, vidutinės imtys yra kai stebėjimų skaičius yra tarp 50 ir 100, o didelės imtys, kai stebėjimų skaičius

yra didesnis už 100.

Naudojama programinė įranga. Monte Carlo simuliacijoms bus naudojama atvirojo kodo statistinė programavimo aplinka **R**, kurios versija: Smooth Sidewalk 3.1.3 (2015-03-09). **R** statistinė programavimo aplinka buvo naudojama kartu su RStudio (versija 0.98.1103) grafine vartotojo sąsaja, veikiančia x86 64 architektūros OS X El Capitan operacinėje sistemoje.

Naudojami paketai. **R** statistinėje programavimo ypatingai svarbią vietą užima paketai. Atliekant Monte Carlo simuliacijas buvo naudojami šie paketai:

1. vars - paketas skirtas VAR, SVAR ir SVEC modelių įvertinimui, diagnostikai, priežastingumo analizei, prognozės paklaidos dispersijos išskaidymui ir impulso atsako funkcijų analizei. Paketo autorius yra Bernhard Pfaff (Pfaff, 2008).
2. urca - paketas skirtas vienetinių šaknų ir kointegravimo testams atlikti. Paketo autorius yra Bernhard Pfaff (Pfaff, 2008).
3. graphics - paketas skirtas grafiniam duomenų atvaizdavimui. Paketo autorius yra R Core Team, prie paketo kūrimo prisidėjo daug kitų autorių (R Core Team, 2015).
4. Hmisc - paketas turintis daug naudingų funkcijų duomenų analizei, grafiniam atvaizdavimui, lentelių sudarymui, duomenų importavimui ir pan. Paketo autorius yra Frank E Harrell Jr., prie paketo kūrimo prisidėjo daug kitų autorių (Harrell ir kiti, 2015).

Tiriami modeliai. Siekiant ištirti dažniausiai pasitaikančius deterministinių modelių įtraukimo į VECM atvejus buvo sudaryti ir tiriami trys modeliai. Modeliai skiriasi deterministinių komponentų skaičiumi ir tipu, todėl puikiai tinka deterministinių komponentų vaidmens įvertinimo tyrimams.

1. Modelis 1: VECM modelis be deterministinių komponentų
2. Modelis 2: VECM modelis su laisvuju nariu
3. Modelis 3: VECM modelis su laisvuju nariu ir trendu

Bendrieji modelių parametrai:

- 4 eilės autoregresinis procesas
- integruoti procesai (Π nėra pilno rango)

- imties dydis tarp 20 ir 200

Tyrimo metu buvo susidurta su modelio parametrų absoliučių reikšmių svarbos problema. Tais atvejais, kai atsitiktinai sugeneruotos parametrų reikšmės būdavo artimos nuliui, atitinkamo komponento vaidmuo VECM modelyje tapdavo nereikšminga - t.y. modeliai turėdami baigtinę imtį beveik niekuo nesiskirdavo nuo modelių be atitinkamų komponentų. Siekiant išspręsti šią problemą visos parametrų reikšmės būdavo atsitiktinai generuojamos, tačiau galimi parametrų intervalai buvo apriboti - tokiu būdu kintamųjų forma būdavo panašiausia į realių makroekonominių kintamųjų formą. Atsitiktiniai šokai buvo n.i.d turintys standartinį nuokrypį lygų 1 ir vidurkį lygų 0.

- deterministinis trendas: nuo -0.55 iki -0.05 ir nuo 0.05 iki 0.55
- laisvasis narys: nuo -100 iki -10 ir nuo 10 iki 100
- stochastinis trendas: nuo -5.5 iki -0.5 ir nuo 0.5 iki 5.5
- autoregresiniai parametrai: nuo -0.9 iki 0.9

Modelių sudarymo procedūra buvo tokia:

1. Sukuriamas modelio parametrų objektas, kuriame parenkamas modelio imties dydis, dimensių skaičius, Π matricos rangas (ir tuo pačiu kointegruojančių vektorių skaičius), ARIMA eilės kiekvienai modelio lygčiai bei laisvojo nario ir trendo indikatoriai.
2. Atliekami parametrų tinkamumo testai, tikrinantys Π matricos rangą, imties dydį ir integruotumo eilę.
3. Sukuriamas rezultatų objektas, kuris bus naudojamas gražinti rezultatams.
4. Sukuriamos modelio paklaidos. Paklaidos generuojamos iš normaliojo skirstinio ir išsaugomas paklaidų objektas.
5. Sukuriami ARMA parametrai. ARMA parametrai generuojami iš anksčiau aprašytų intervalų ir išsaugomi.
6. Simuliuojami ARMA procesai naudojant anksčiau sugeneruotas paklaidas.
7. Pridedamas stochastinis trendas. Priklausomai nuo modelio dimensių skaičiaus, sukuriama tiek skirtingų stochastinių trendų, kiek reikia siekiant sukurti tokį VECM modelį, kad Π matricos rangas būtų lygus parametrų vektoriuje nustatytam skaičiui. Sukurti stochastiniai trendai pridedami prie modelio lygčių padauginėti iš parametro reikšmės atsitiktinai sugeneruotis iš anksčiau pateikto vektoriaus.

8. Pridedami deterministiniai komponentai. Sukuriami deterministinių komponentų parametrai, tada jie pridedami prie modelio lygčių.

Tiriamos VECM įvertinimo procedūros. Nustatant VECM kointegruojančių vektorių skaičių buvo naudojamos trys skirtingai specifikuotos Johansen procedūros. Visos šios procedūros yra pateikiamos urca pakete. Visos Johansen procedūros turėjo ilgojo laikotarpio specifikaciją, kuri yra aprašoma lygtyje (97).

$$\Delta \mathbf{X}_t = \mathbf{\Gamma}_1 \Delta \mathbf{X}_{t-1} + \cdots + \mathbf{\Gamma}_{k-1} \Delta \mathbf{X}_{t-k+1} + \mathbf{\Pi} \mathbf{X}_{t-1} + \boldsymbol{\mu} + \boldsymbol{\Phi} \mathbf{D}_t + \boldsymbol{\varepsilon}_t \quad (97)$$

kur $\mathbf{\Gamma}_i = -(\mathbf{I} - \mathbf{\Pi}_1 - \cdots - \mathbf{\Pi}_i)$, $(i = 1, \dots, k-1)$ ir $\mathbf{\Pi} = -(\mathbf{I} - \mathbf{\Pi}_1 - \cdots - \mathbf{\Pi}_k)$.
Buvo tiriamos tokios Johansen procedūros:

1. Johansen procedūra be deterministinių komponentų. Ši procedūra neturi deterministinių komponentų nei paklaidos korekcijos komponente, nei VAR komponente. Procedūra yra iškviečiama su parametru `ecdet = "none"`.
2. Johansen procedūra su laisvuju nariu. Ši procedūra turi laisvąjį narį paklaidos korekcijos komponente, tačiau neturi deterministinių komponentų VAR komponente. Procedūra yra iškviečiama su parametru `ecdet = "const"`.
3. Johansen procedūra su trendu. Ši procedūra turi trendo komponentą paklaidos korekcijos komponente ir laisvąjį narį VAR komponente. Procedūra yra iškviečiama su parametru `ecdet = "trend"`.

3.2. Modelis 1

Modelis 1 yra trimatis VECM modelis be deterministinių komponentų. Bus tiriamos trys šio modelio specifikacijos: su nekointegruotais kintamaisiais, su vienu kointegruojančiu vektoriumi ir su dviem kointegruojančiais vektoriais. Lygtis 98 aprašo modelio specifikaciją su skirtingais stochastiniais trendais, lygtis 99 aprašo modelio specifikaciją su vienu kointegruojančiu vektoriumi, o lygtis 100 aprašo modelio specifikaciją su dviem kointegruojančiais vektoriais.

$$\begin{aligned} x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + \varepsilon_{xt} \\ y_t &= \alpha_y m_t + \sum_{i=1}^4 b_i y_{t-i} + \varepsilon_{yt} \\ z_t &= \alpha_z n_t + \sum_{i=1}^4 c_i z_{t-i} + \varepsilon_{zt} \end{aligned} \quad (98)$$

kur l_t , m_t ir n_t yra atsitiktinai sugeneruoti vienetinės šaknies procesai.

$$\begin{aligned} x_t &= \alpha_x l1_t + \sum_{i=1}^4 a_i x_{t-i} + \varepsilon_{xt} \\ y_t &= \alpha_y l2_t + \sum_{i=1}^4 b_i y_{t-i} + \varepsilon_{yt} \\ z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + \varepsilon_{zt} \end{aligned} \quad (99)$$

kur $l1_t$ ir $l2_t$ yra atsitiktinai sugeneruoti vienetinės šaknies procesai, o $l_t = l1_t + l2_t$.

$$\begin{aligned} x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + \varepsilon_{xt} \\ y_t &= \alpha_y l_t + \sum_{i=1}^4 b_i y_{t-i} + \varepsilon_{yt} \\ z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + \varepsilon_{zt} \end{aligned} \quad (100)$$

kur l_t yra atsitiktinai sugeneruotas vienetinės šaknies procesas.

Modelis 4 turi tokią VECM reprezentaciją:

$$\begin{pmatrix} \Delta x_t \\ \Delta y_t \\ \Delta z_t \end{pmatrix} = \Pi \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} + \Gamma_1 \begin{pmatrix} \Delta x_{t-1} \\ \Delta y_{t-1} \\ \Delta z_{t-1} \end{pmatrix} + \Gamma_2 \begin{pmatrix} \Delta x_{t-2} \\ \Delta y_{t-2} \\ \Delta z_{t-2} \end{pmatrix} + \quad (101)$$

$$+ \Gamma_3 \begin{pmatrix} \Delta x_{t-3} \\ \Delta y_{t-3} \\ \Delta z_{t-3} \end{pmatrix} + \Gamma_4 \begin{pmatrix} \Delta x_{t-4} \\ \Delta y_{t-4} \\ \Delta z_{t-4} \end{pmatrix} + \begin{pmatrix} \varepsilon_{xt} \\ \varepsilon_{yt} \\ \varepsilon_{zt} \end{pmatrix} \quad (102)$$

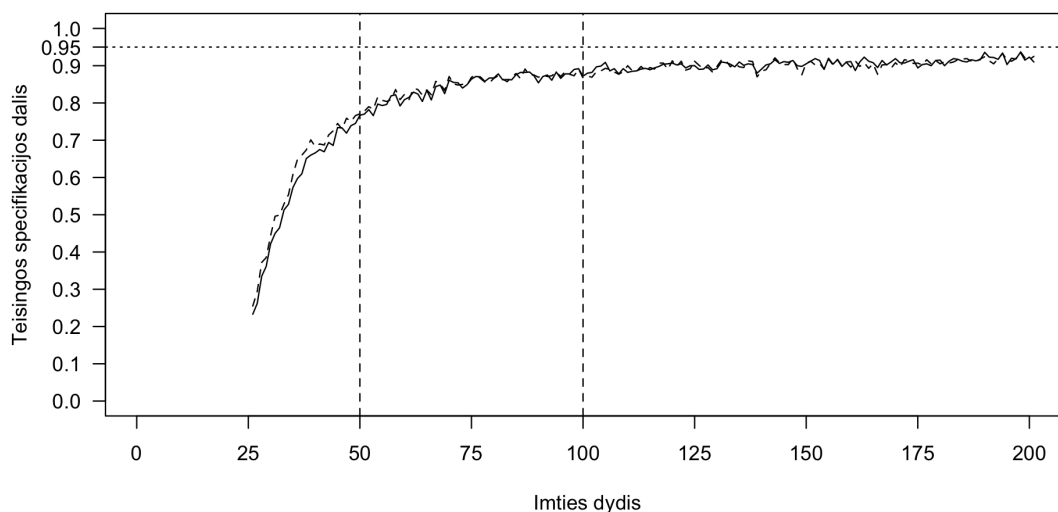
kur

$$\Pi = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} \\ \beta_{12} & \beta_{22} & \beta_{32} \\ \beta_{13} & \beta_{23} & \beta_{33} \end{pmatrix} \quad (103)$$

3.2.1. Johansen procedūra be deterministinių komponentų

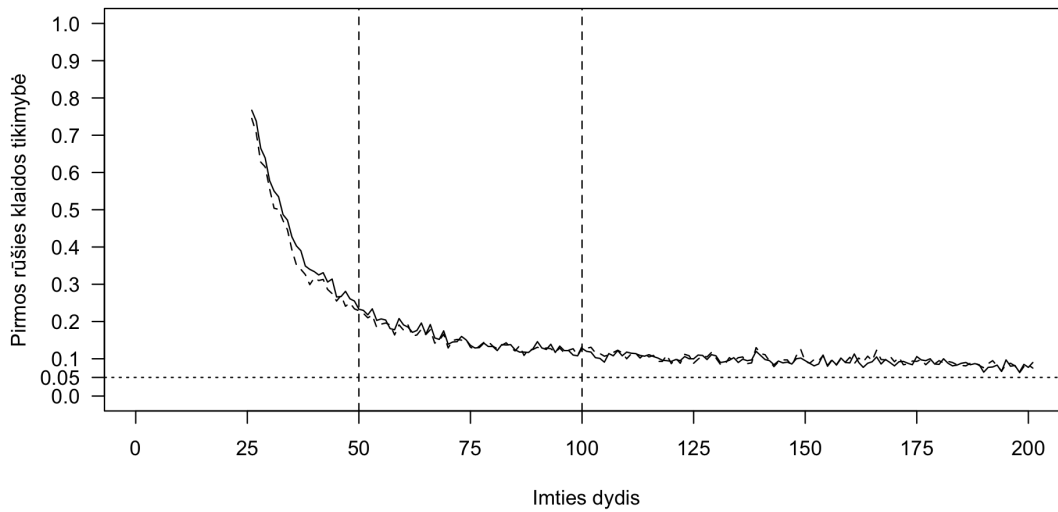
Johansen procedūra be deterministinių komponentų yra tinkamiausia šiam modeliui, nes jame nėra deterministinių komponentų.

Nekointegruoti kintamieji. Pirmiausia Johansen procedūra be deterministinių komponentų bus atlikta nekointegruotai modelio specifikacijai. Paveikslėlis 1 parodo kokia dalis testų nustato kointegruojančių vektorių skaičių teisingai. Šiuo atveju tiek pėdsako, tiek maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, tačiau maksimalios tikrinės reikšmės procedūra yra šiek tiek pranašesnė mažose imtyse. Mažose imtyse testo galia labai silpna tarp 20% ir 80%, o vidutinėse ir didelėse imtyse stipri - tarp 80% ir 90%.



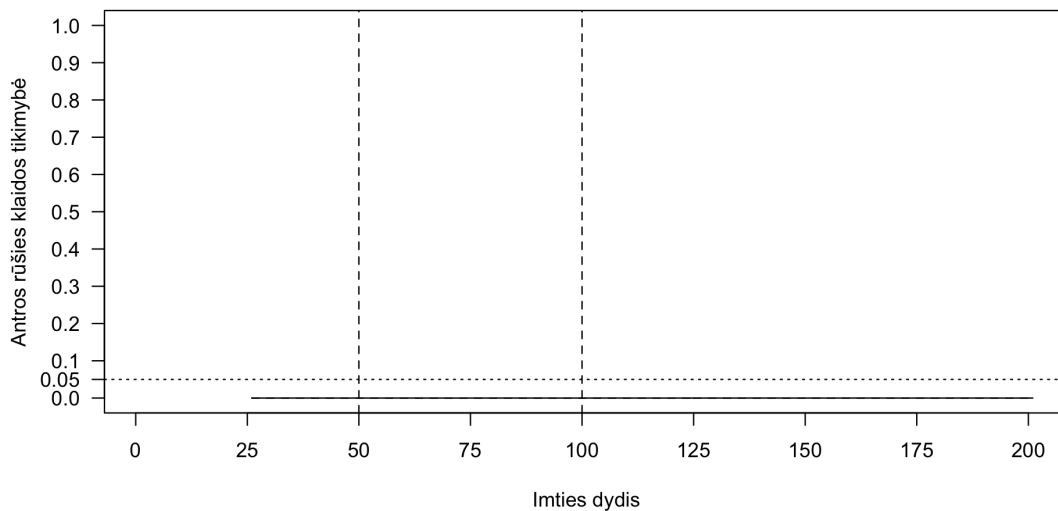
1 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 2 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus. Mažose imtyse klaidos tikimybė yra labai didelė, tačiau sparčiai mažėjanti didėjant imties dydžiui. Stebina tai, kad net ir didelėse imtyse klaidos tikimybė yra šiek tiek didesnė nei turėtų būti pagal reikšmingumo lygį.



2 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

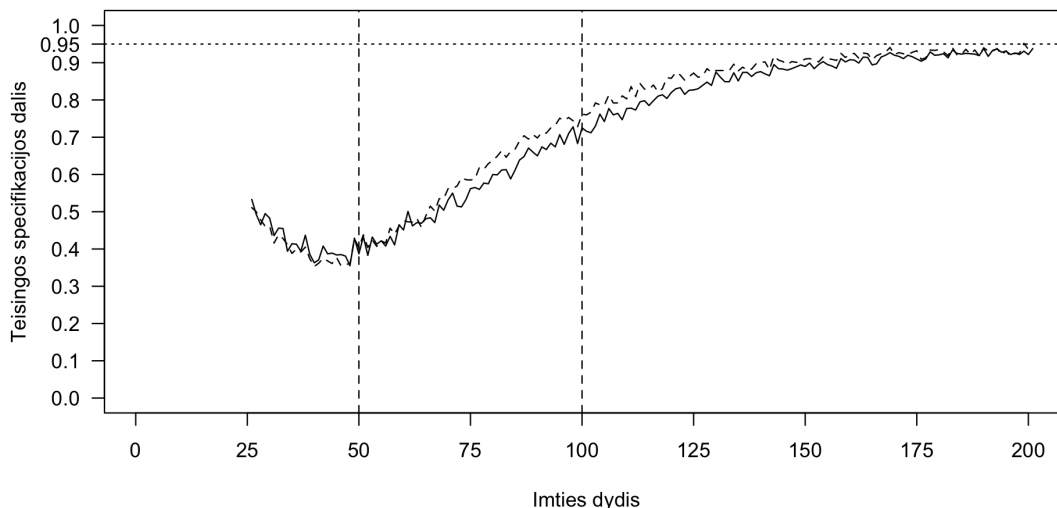
Paveikslėlis 3 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Kadangi kointegruojančių vektorių nėra, tai ir antros rūšies klaidos padaryti neįmanoma, todėl tikimybė yra 0.



3 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

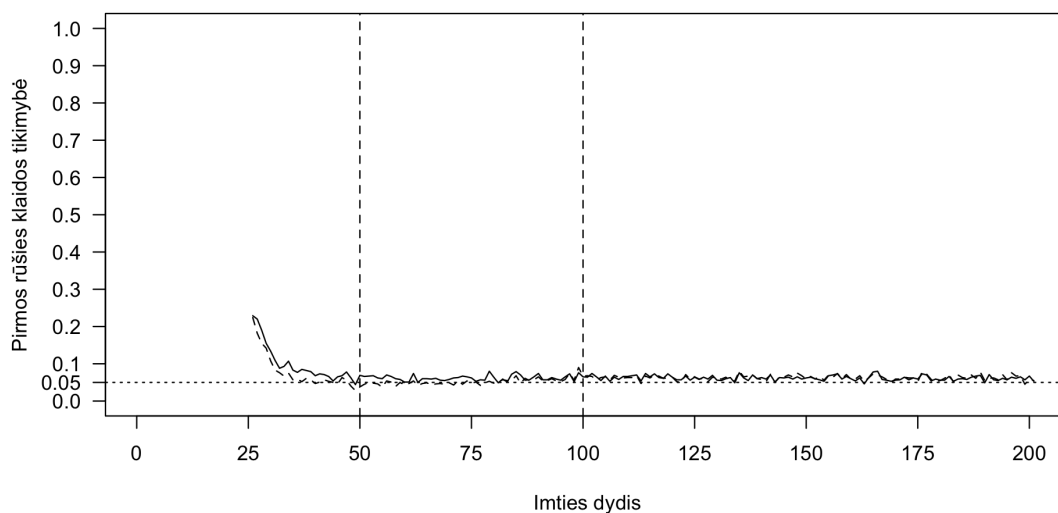
Vienas kointegruojantis vektorius. Tuo atveju, kai visos trys modelio lygtys turi tą patį stochastinį trendą, pėdsako ir maksimalios tikrinės reikšmės testai

pateikia gan panašius rezultatus. Paveikslėlis 4 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra ganėtinai silpnas mažose imtyse tarp 30% ir 55%. Vidutinėse imtyse testas reikšmingai stiprėja iki 80%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



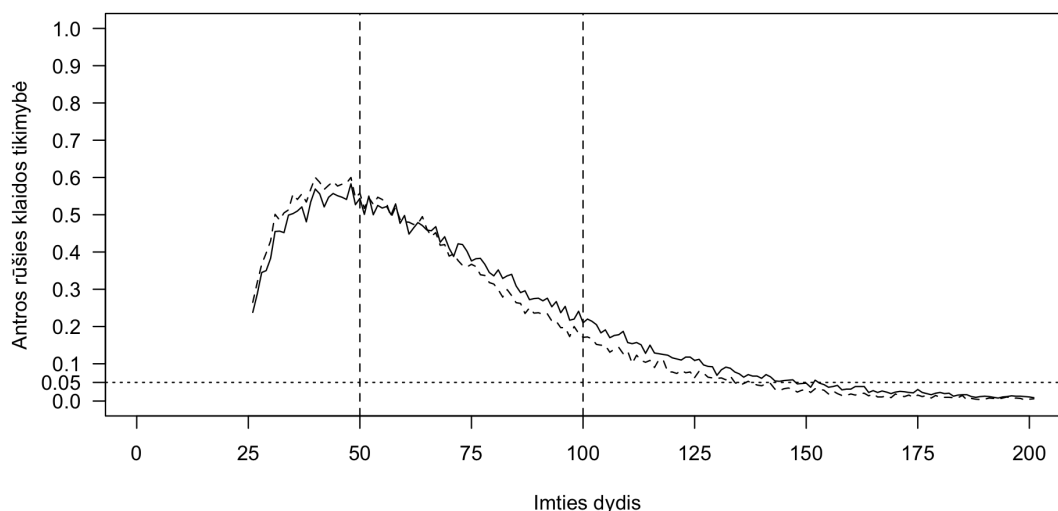
4 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 4 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Kai imtys yra vidutinės arba didelės tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos padaro šią klaidą maždaug 5% atvejų, kaip ir tikėtasi pagal reikšmingumo lygmenį. Mažose imtyse šios klaidos tikimybė yra šiek tiek didesnė - apie 10%.



5 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

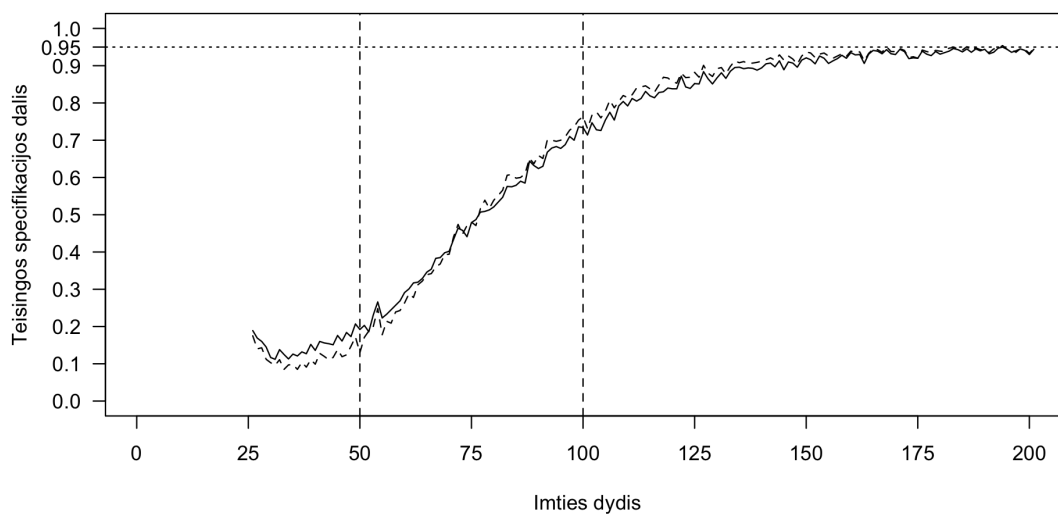
Paveikslėlis 6 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 70%. Vidutinėse imtyse šios klaidos tikimybė mažėja, o didelėse imtyse priartėja prie pat 0.



6 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

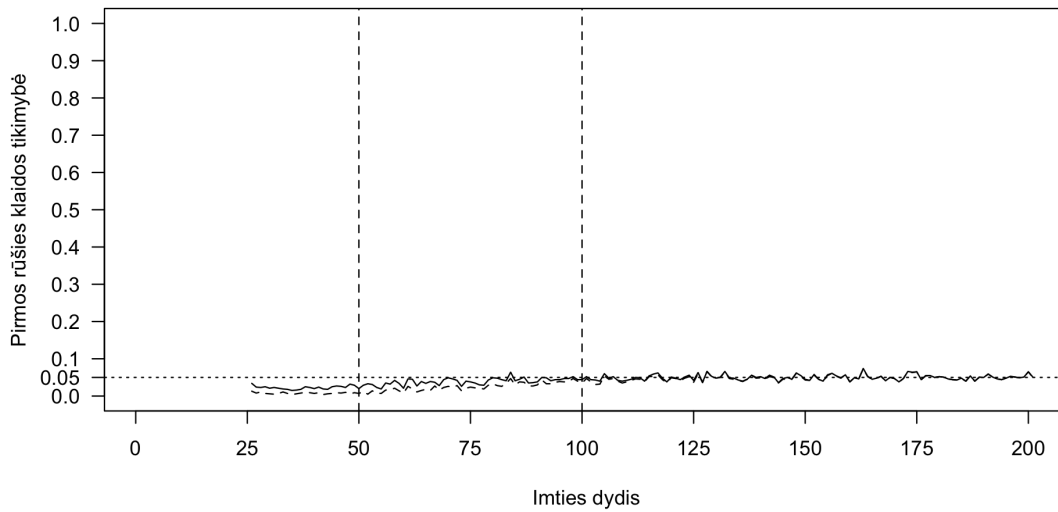
Du kointegruojantys vektoriai. Tuo atveju, kai modelis turi du kointegruojančius vektorius testų galia yra šiek tiek silpnesnė nei vieno kointegruojančio

vektoriaus atveju. Paveikslėlis 7 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testo galia mažose imtyse yra ypatingai maža - tarp 10% ir 30%. Vidutinėse imtyse testas reikšmingai stiprėja iki 75%, o didelėse imtyse pasiekia 95%. Pėdsako testas yra pranašesnis mažose imtyse, tačiau maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



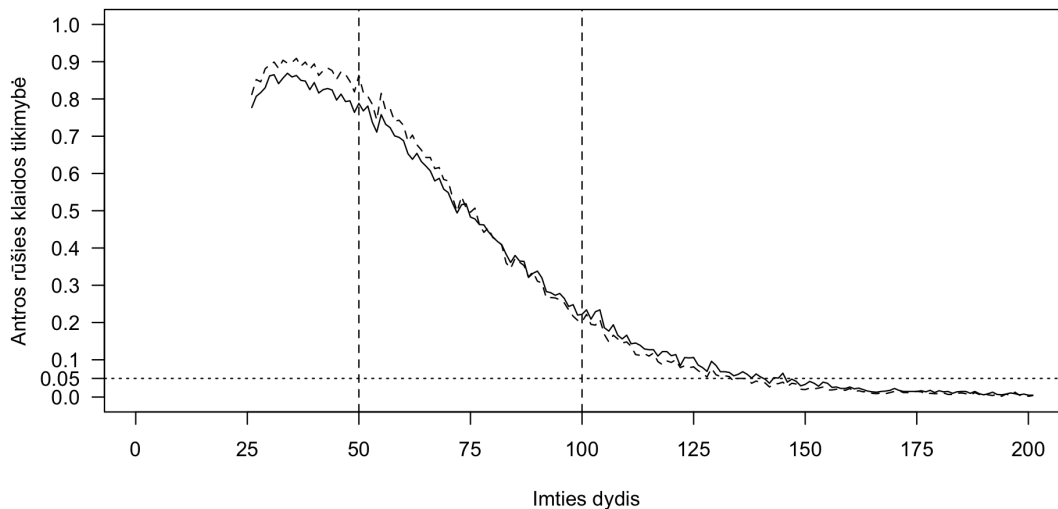
7 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 7 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji turi du kointegruojančius vektorius. Tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos padaro šią klaidą maždaug 5% atvejų, kaip ir tikėtasi pagal reikšmingumo lygmenį, kai imtys yra didelės. Mažose ir vidutinėse imtyse testai pirmos rūšies klaidą padaro rečiau - tarp 0% ir 5%.



8 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 9 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 80%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie pat 0.

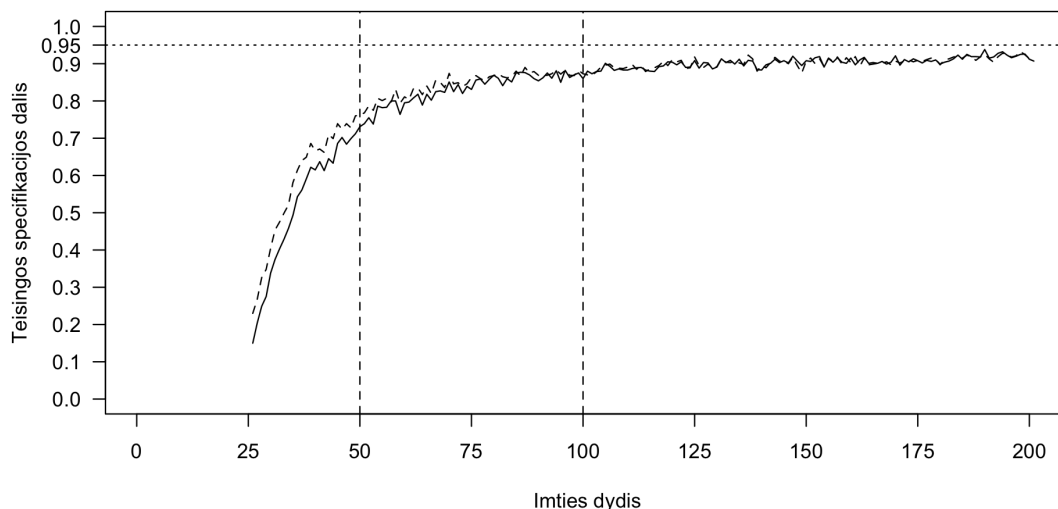


9 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.2.2. Johansen procedūra su konstanta

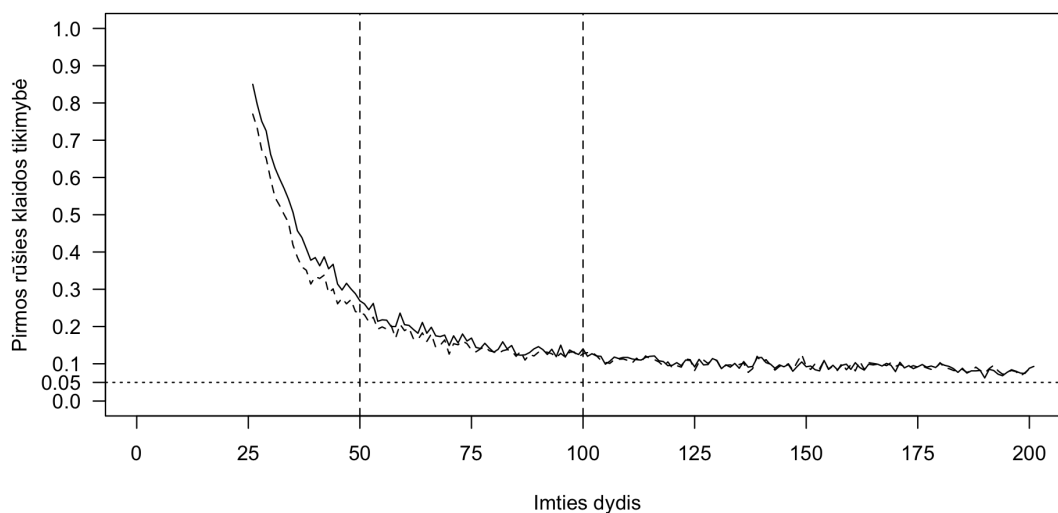
Johansen procedūra su konstanta nėra pati tinkamiausia šiam modeliui, nes jame nėra deterministinių komponentų. Ši misspecifikacija neturėtų daryti lemiamos įtakos testų rezultatams, tačiau galima tikėtis, kad šio testo galia bus silpnesnė nei teisingai specifikuotos procedūros.

Nekointegruoti kintamieji. Paveikslėlis 10 parodo kokia dalis Johansen procedūros su konstanta testų nustato kointegruojančių vektorių skaičių teisingai. Rezultatai panašūs kaip ir Johansen procedūros be deterministinių komponentų atveju: tiek pėdsako, tiek maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, kai imtys yra vidutinės arba didelės, tačiau, kai imtys yra mažos, maksimalios tikrinės reikšmės testas pateikia tikslesnius rezultatus. Mažose imtyse testo galia yra labai silpna tarp 15% ir 80%, o vidutinėse ir didelėse imtyse stipri - tarp 80% ir 95%.



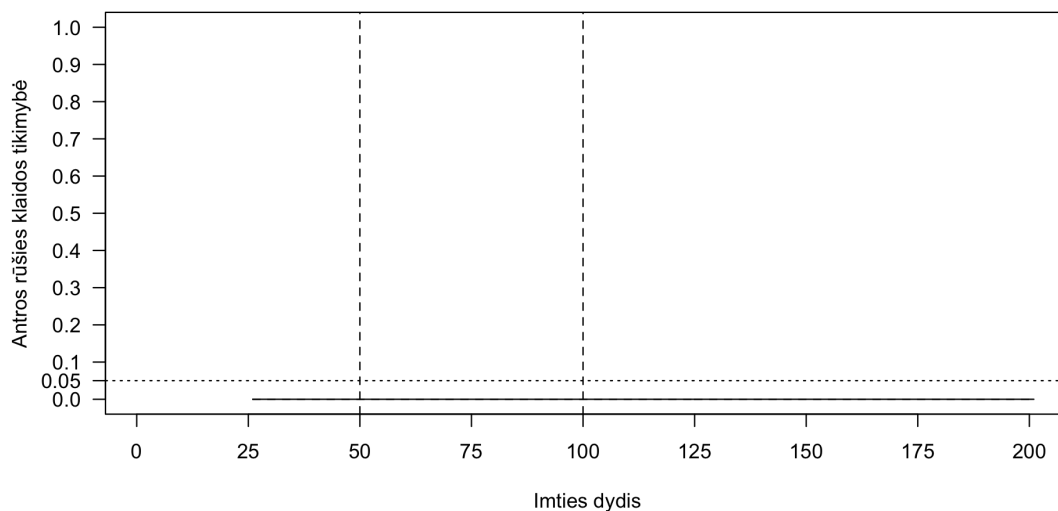
10 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 11 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, kai imtys yra vidutinio arba didelio dydžio. Kai imtys yra mažos maksimalios tikrinės reikšmės testas yra pranašesnis. Mažose imtyse klaidos tikimybė yra ypač didelė iki 85%, tačiau sparčiai mažėjanti didėjant imties dydžiui.



11 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

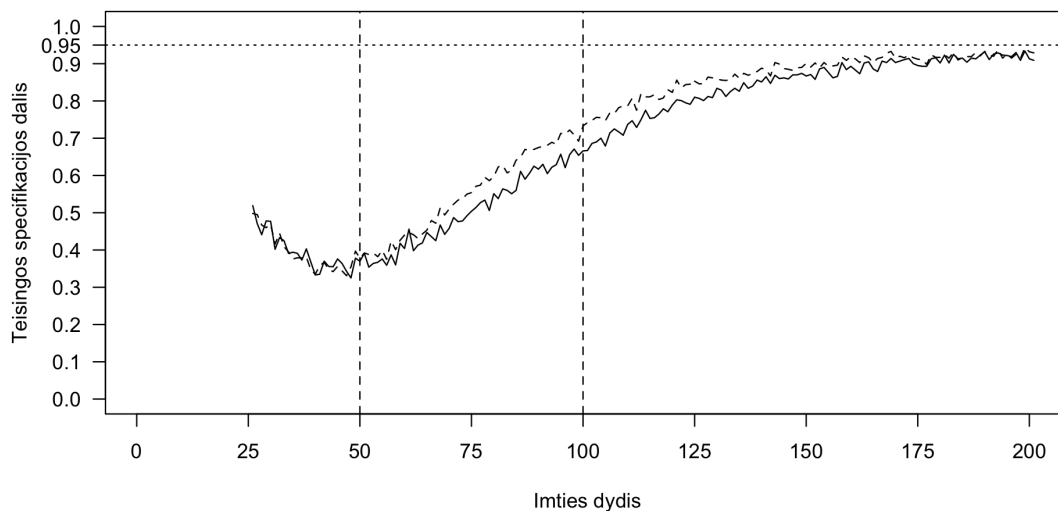
Paveikslėlis 12 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su laisvuju nariu. Kaip ir praeitu atveju šios klaidos tikimybė yra 0.



12 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

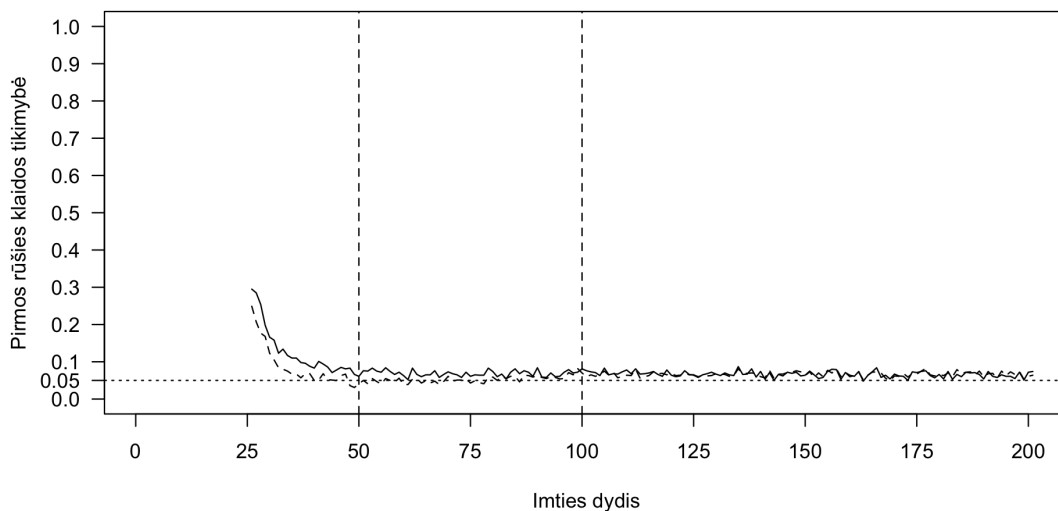
Vienas kointegruojantis vektorius. Kai modelis turi bendrą stochastinį trendą, testų rezultatai yra panašūs į Johansen procedūros atvejį be deterministinių komponentų. Paveikslėlis 13 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra labai silpnas mažose imtyse tarp 30% ir

50%. Vidutinėse imtyse testas šiek tiek sustipėja stiprėja iki 70%, o didelėse imtyse pasiekia 90%. Maksimalios tikrinės reikšmės testas reikšmingai tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 60 ir 160.



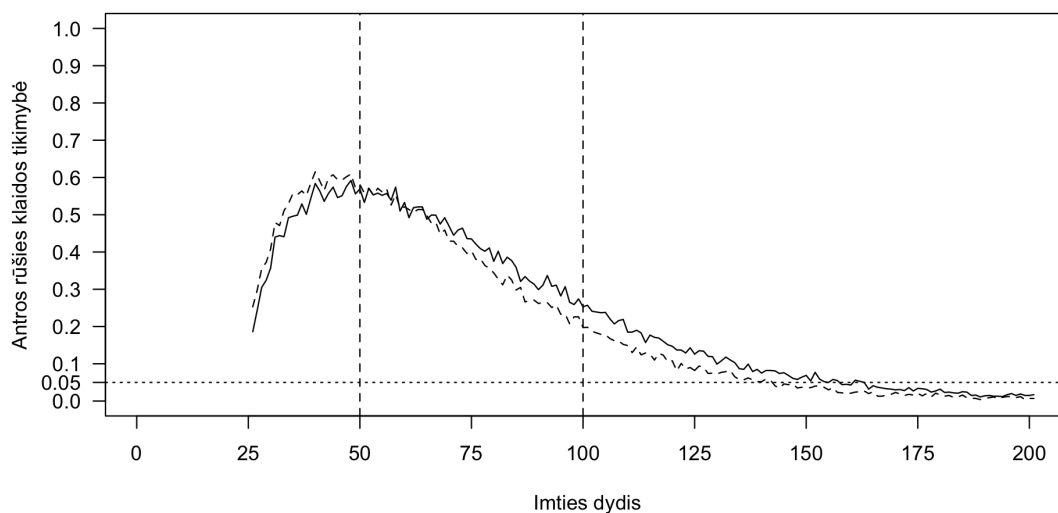
13 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 14 pardodo, pirmos rūšies klaidos tikimybę. Ji svyruoja tarp 5% ir 10% vidutinėse ir didelėse imtyse. Kai imtys yra mažos klaidos tikimybės intervalas yra tarp 5% ir 35%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testas yra ženkliai geresnis nei pėdsako testas.



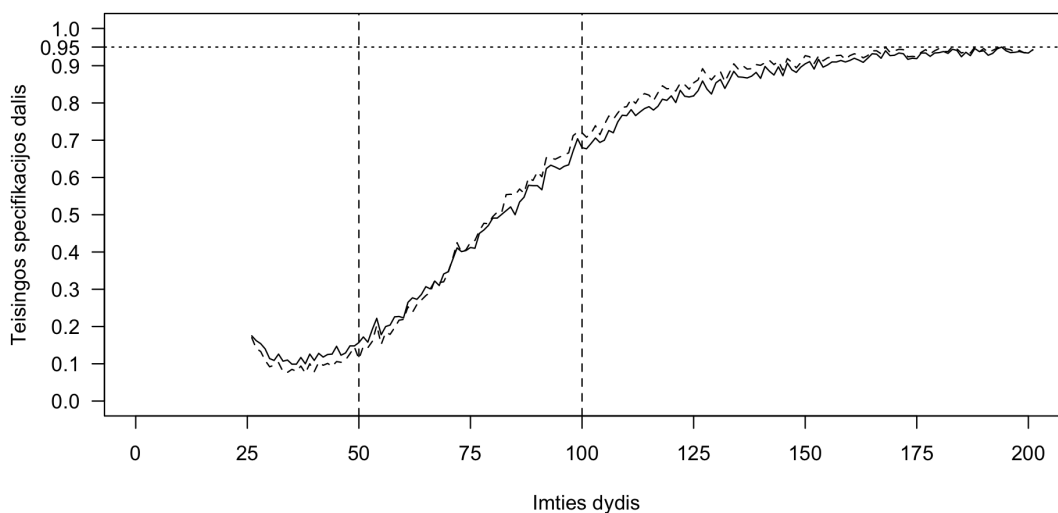
14 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 15 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji turi vieną kointegruojantį vektorių. Šios klaidos tikimybė yra didelė mažose ir vidutinėse imtyse, net iki 65%. Vidutinėse imtyse, didėjant imties dydžiui, šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Kai imties dydis yra tarp 25 ir 40, didėjant imties dydžiui, šios klaidos tikimybė sparčiai auga, po to pradeda sparčiai mažėti. Mažose imtyse pėdsako testas yra pranašesnis, o vidutinėse ir didelėse imtyse maksimalios tikrinės reikšmės testas yra tikslesnis.



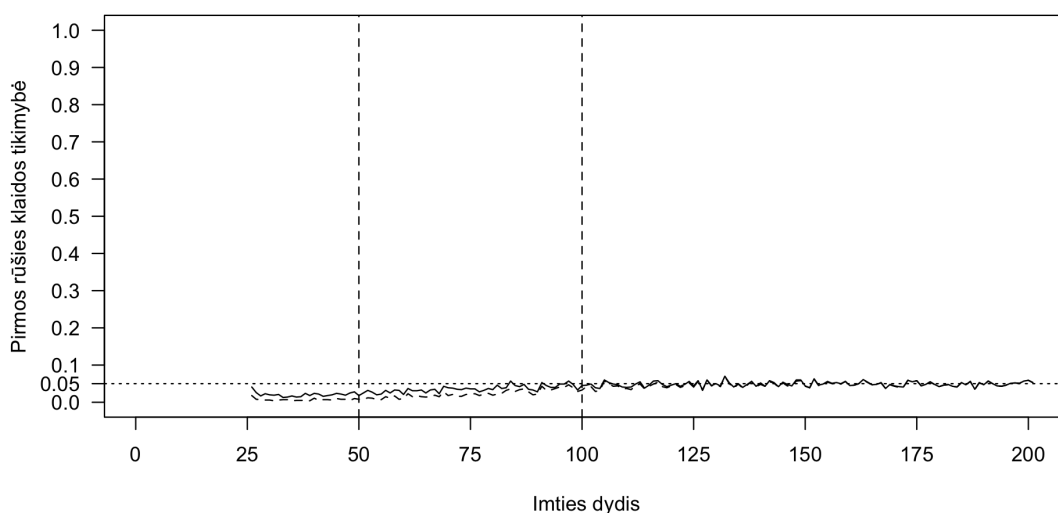
15 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Kai modelis turi du kointegruojančius vektorius, testų rezultatai skiriasi nuo kitų dviejų atvejų. Paveikslėlis 16 parodo, kiek procentų atvejų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra labai silpnas mažose imtyse tarp 10% ir 20%. Vidutinėse imtyse testas reikšmingai stiprėja iki 70%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



16 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

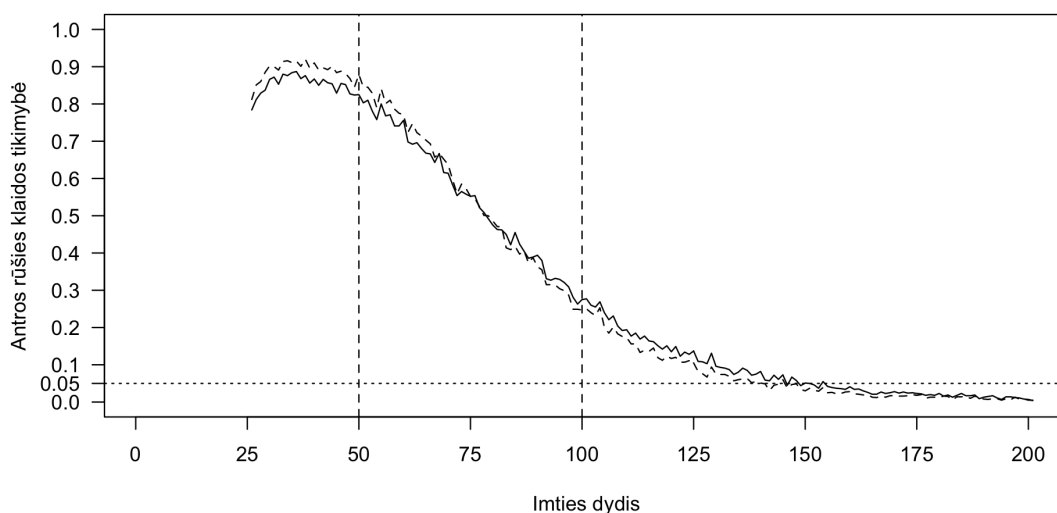
Paveikslėlis 17 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5% vidutinėse ir didelėse imtyse. Mažose imtyse šios klaidos tikimybė yra mažesnė nei 5%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testai yra pranašesni už pėdsako testus.



17 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 18 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač

didelė mažose imtyse, kur pasiekia net 90%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Mažose imtyse pėdsako testai yra ženkliai pranašesni už maksimalios tikrinės reikšmės testus, tuo tarpu, kai imties dydis yra tarp 75 ir 125 maksimalios tikrinės reikšmės testai yra tikslesni.

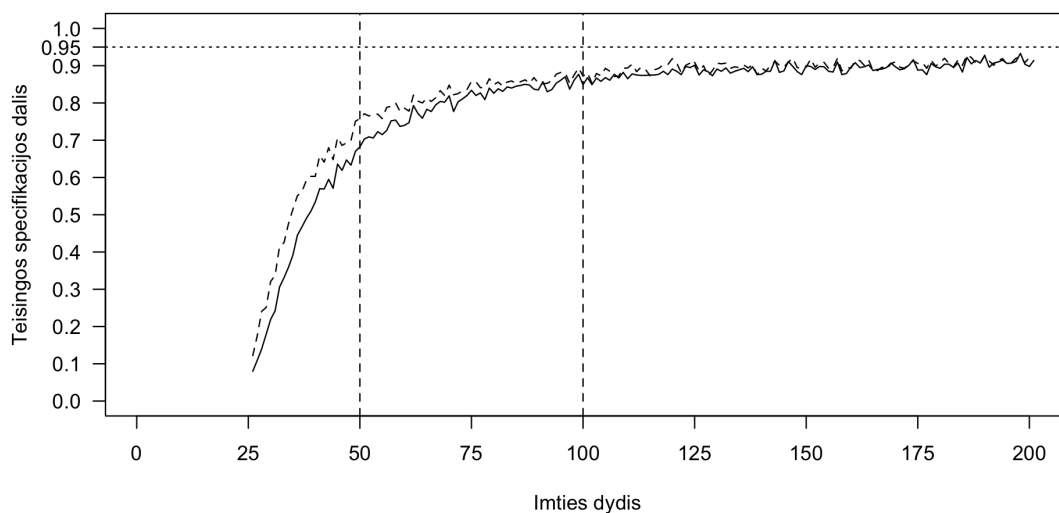


18 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.2.3. Johansen procedūra su trendu

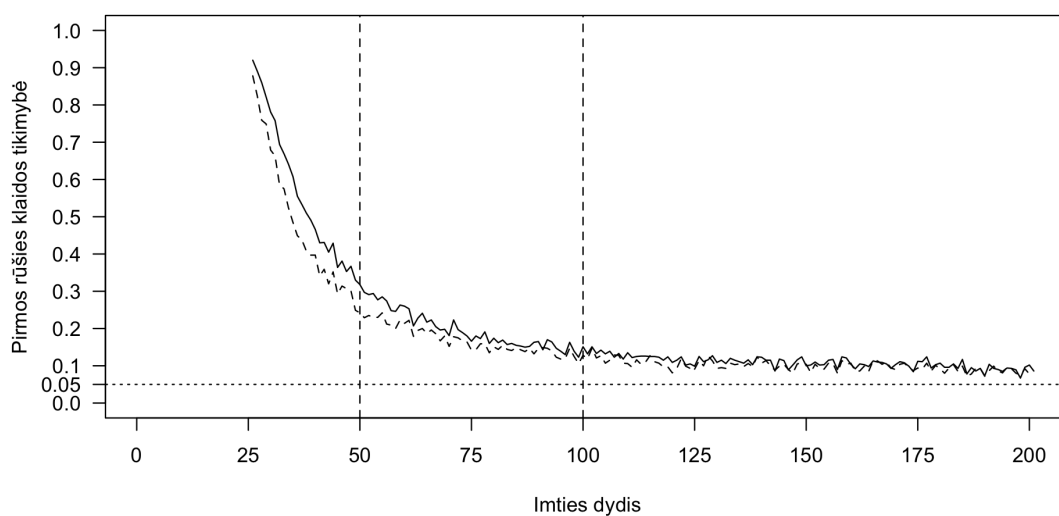
Johansen procedūra su trendu nėra tinkama šiam modeliui, nes jame nėra deterministinių komponentų.

Nekointegruoti kintamieji. Paveikslėlis 19 parodo kokia dalis Johansen procedūros su trendu testų nustato kointegruojančių vektorių skaičių teisingai. Grafiko forma panaši į dviejų anskčiau tirtus atvejus. Testas yra ypač silpnas mažose imtyse - jo galia yra intervale tarp 5% ir 70%. Vidutinėse imtyse testas reikšmingai stiprėja nuo 70% iki 85%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra nuo 25 iki 125.



19 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

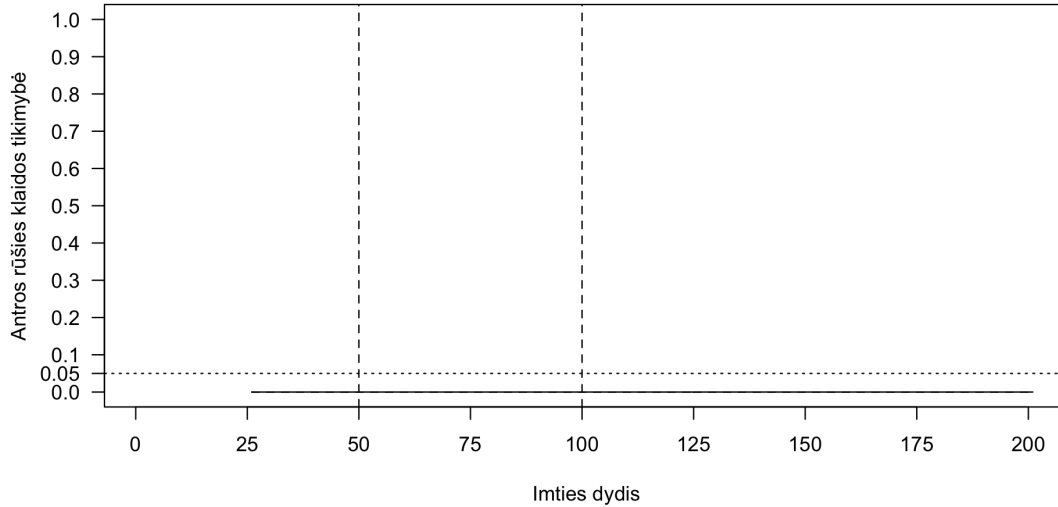
Paveikslėlis 20 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu. Klaidos tikimybė mažose imtyse labai didelė, siekianti 95%. Didėjant imties dydžiui klaidos tikimybė mažėja ir didelėse imtyse pasiekia 10%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testas yra tikslesnis už pėdsako testą.



20 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

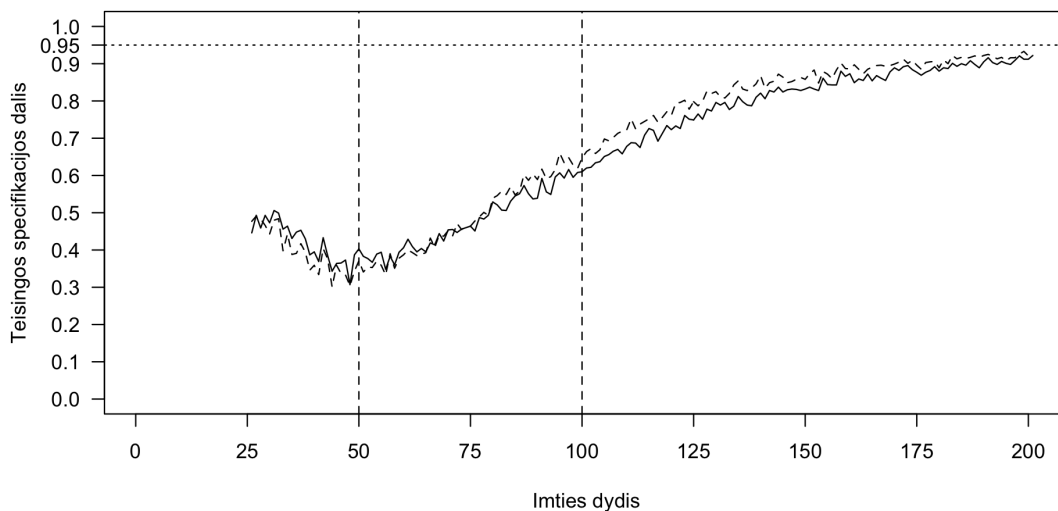
Paveikslėlis 20 parodo antros rūšies klaidos tikimybę naudojant Johansen pro-

cedūrą su trendu. Ši tikimybė yra 0.



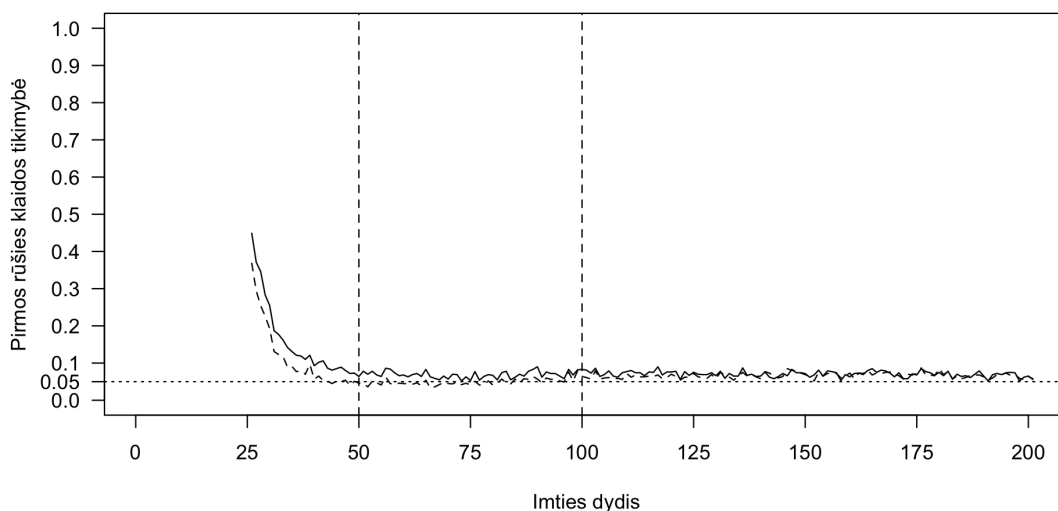
21 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Vienas kointegruojantis vektorius. Kai modelis turi vieną kointegruojantį vektorių, Johansen procedūros rezultatai yra panašūs į kitus du Johansen procedūros specifikavimo atvejus, tačiau testų galia yra šiek tiek silpnesnė. Paveikslėlis 22 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas galia mažose imtyse yra vidutinė ir svyruoja tarp 30% ir 50%, vidutinėse imtyse - tarp 30% ir 60%, o didelėse imtyse tarp 60% ir 95%. Pėdsako testas yra tikslesnis mažose imtyse, o maksimalios tikrinės reikšmės testas yra tikslesnis didelėse imtyse.



22 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

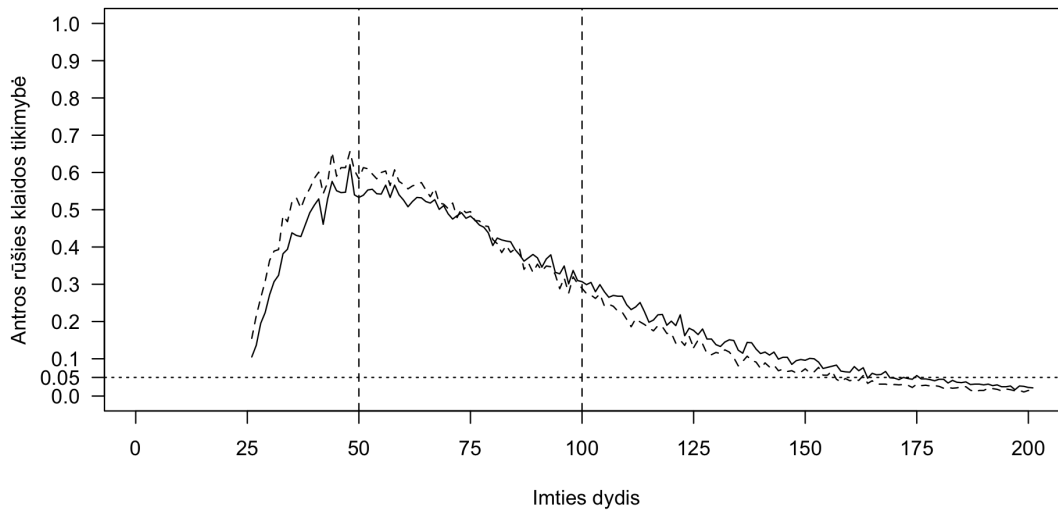
Paveikslėlis 23 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5%, kai imtys yra vidutinės arba didelės. Mažose imtyse klaidos tikimybė siekia 50% tačiau sparčiai krenta didėjant imties dydžiui. Maksimalios tikrinės reikšmės testas yra pranašesnis mažose ir vidutinėse imtyse.



23 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

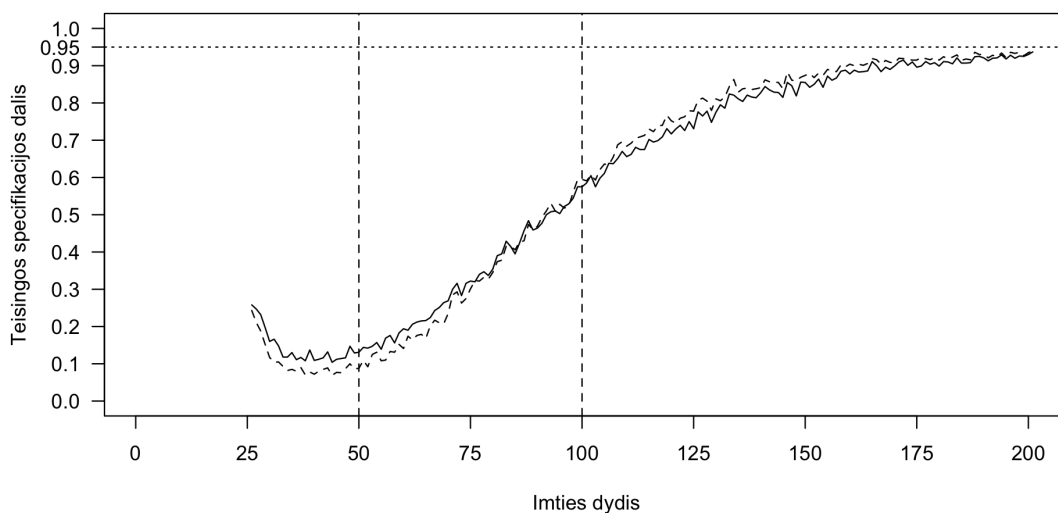
Paveikslėlis 24 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti ir turi vieną kointegruojantį

vektorių. Šios klaidos tikimybė yra didelė mažose imtyse, ir siekia iki 60%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Mažose imtyse didėjant imties dydžiui klaidos tikimybė didėja, o vidutinėse ir didelėse imtyse mažėja.



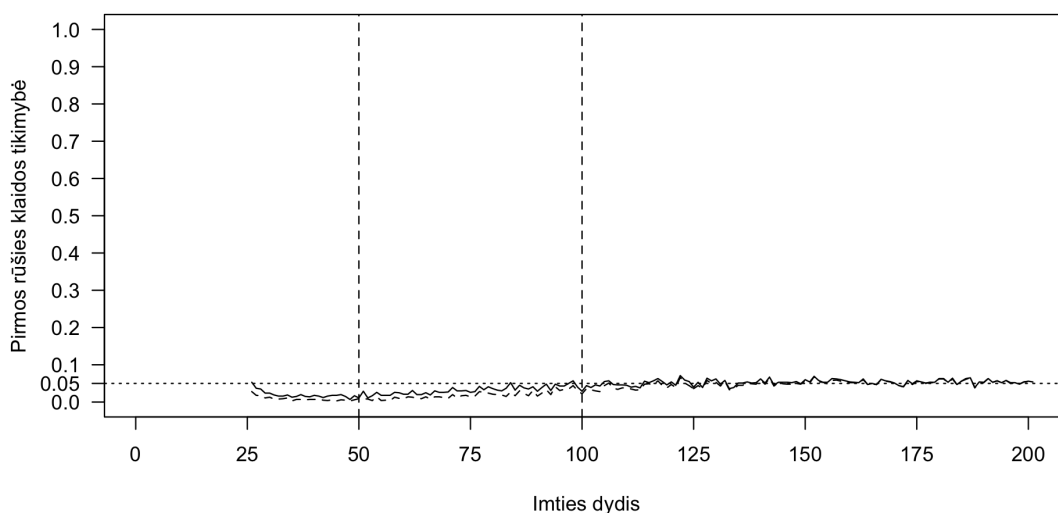
24 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Paveikslėlis 25 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai, kai kintamieji turi du kointegruojančius vektorius. Testas yra labai silpnas mažose imtyse tarp 5% ir 25%. Vidutinėse imtyse testas reikšmingai stiprėja iki 60%, o didelėse imtyse pasiekia 95%. Pėdsako testas yra pranašesnis mažose imtyse, o maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių didelėse imtyse.



25 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

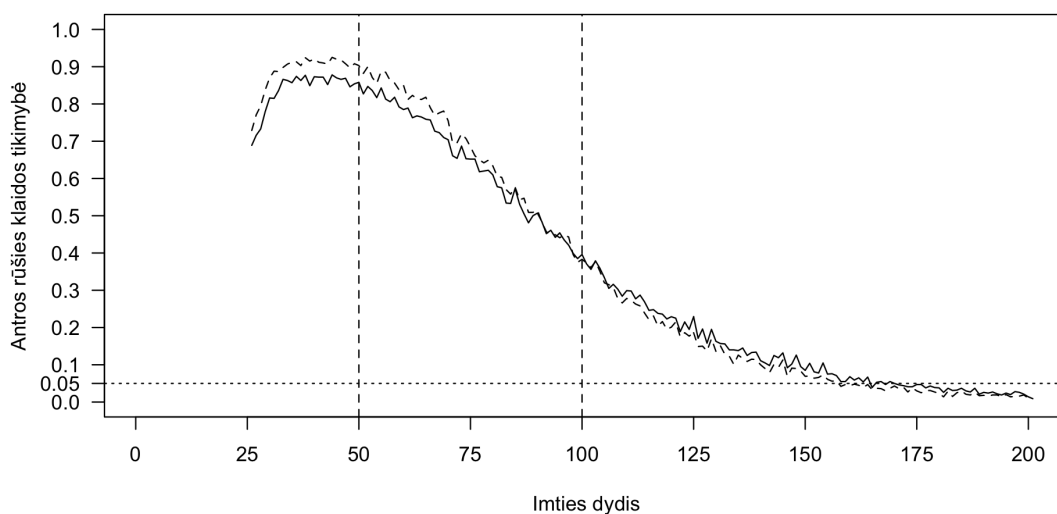
Paveikslėlis 26 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5%, kai imtys yra vidutinės ir didelės. Mažose imtyse šios klaidos tikimybės intervalas yra tarp 0% ir 5%.



26 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 27 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu, kai kintamieji yra kointegruoti ir turi du kointegruojančius vektorius. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 95%. Vidutinėse imtyse

šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Maksimalios tikrinės reikšmės testas ir šiuo atveju yra mažiau tikslus už pėdsako testą, išskyrus dideles imtis.

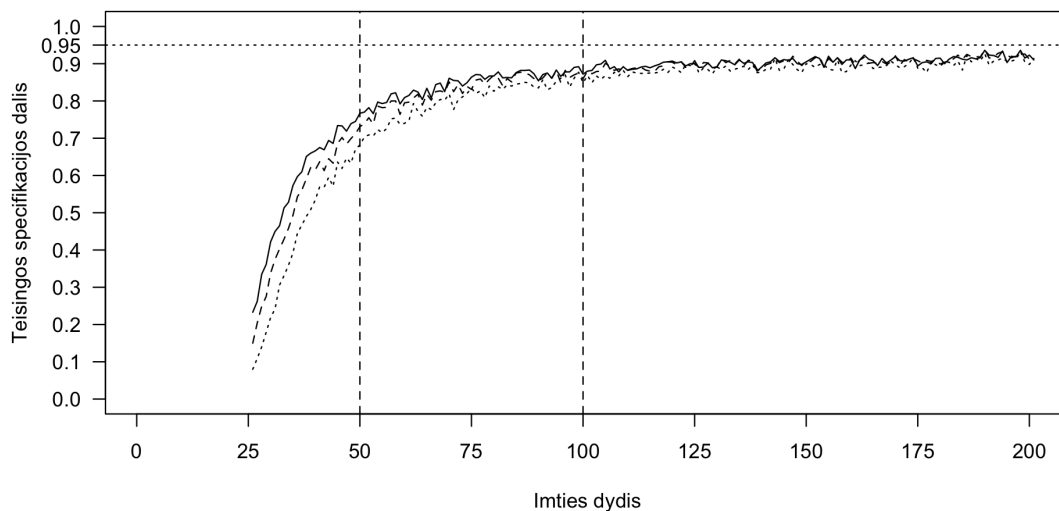


27 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

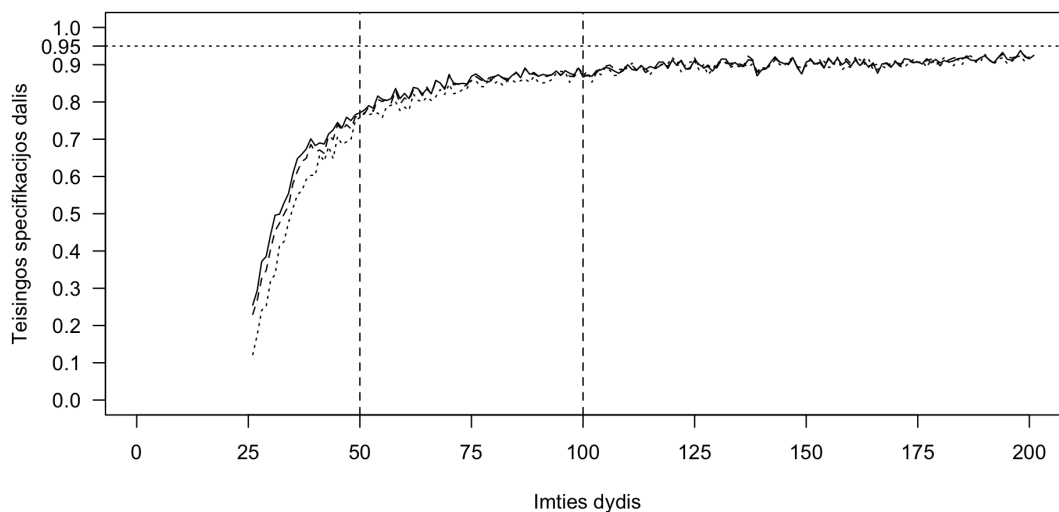
3.2.4. Misspecifikacijos įtaka rezultatams

Neteisingai specifikuota Johansen procedūra įtraukiant per daug deterministinių komponentų nedaro esminės įtakos testų rezultatams. Misspecifikuoti testai yra silpnesni, tačiau didelėse imtyse konverguoja prie teisingai specifikuotų Johansen procedūrų rezultatų.

Nekointegruoti kintamieji. Kai imtys yra mažos visų tipų Johansen procedūros turi vidutinę tikimybę padaryti pirmos rūšies klaidą. Vidutinėse ir didelėse imtyse testų galia yra didelė. Paveikslėlis 28 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 29 - naudojant maksimalios tikrinės reikšmės statistiką. Teisingai specifikuotas testas yra tiksliausias, procedūra su konstanta šiek tiek silpnesnė, o procedūra su trendu pati silpniausia.

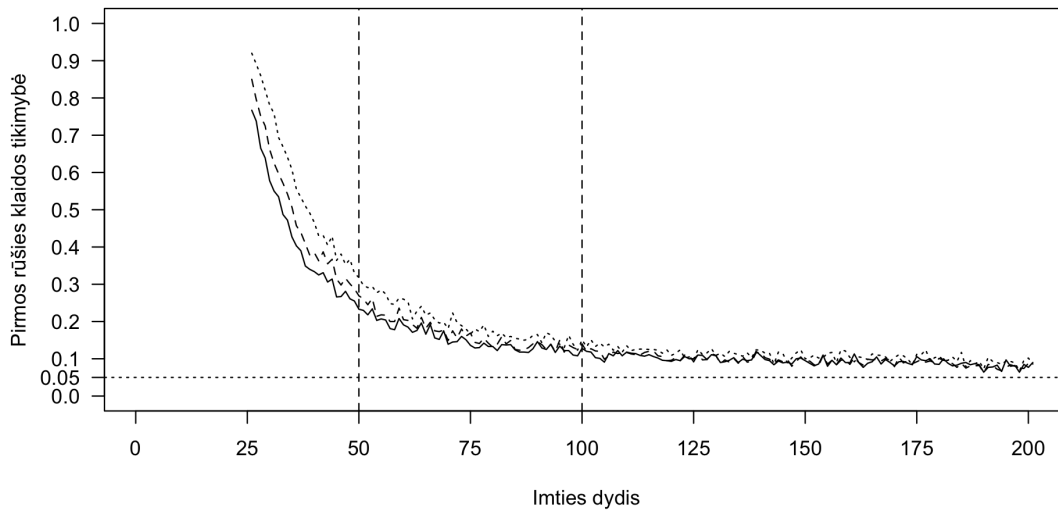


28 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

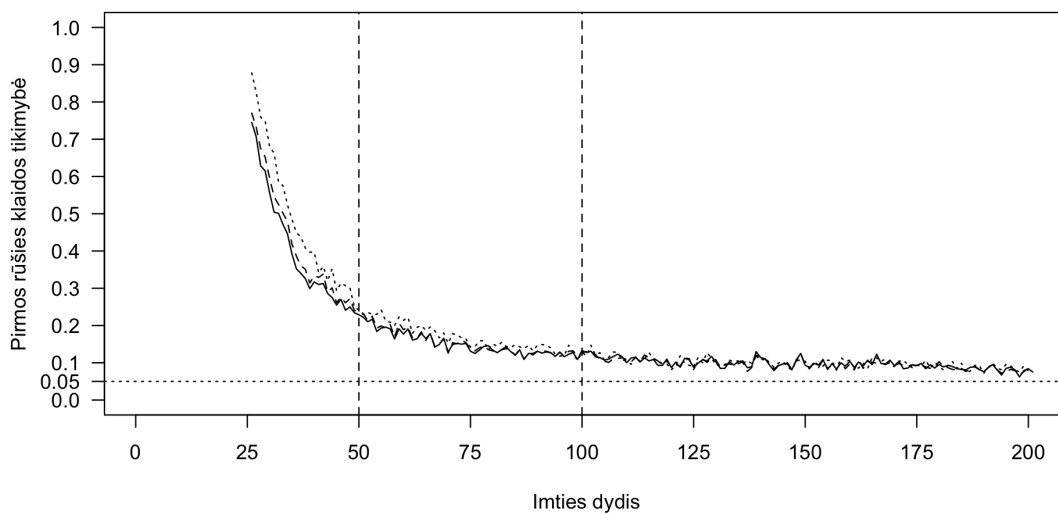


29 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 30 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 31 - naudojant maksimalios tikrinės reikšmės statistiką. Klaidos tikimybė mažiausia teisingai specifiukuoto testo atveju, šiek tiek didesnė testo atveju su konstanta ir didžiausia testo atveju su trendu.

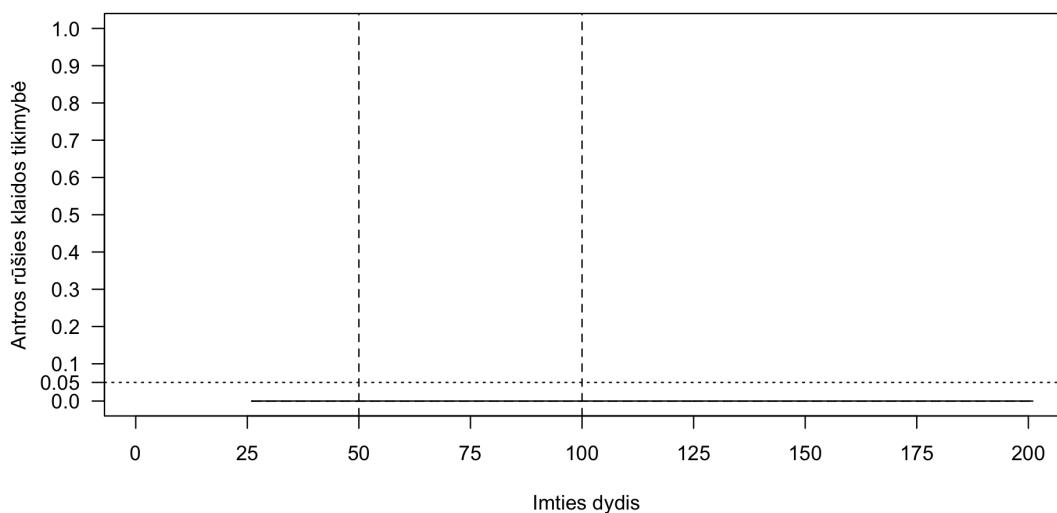


30 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

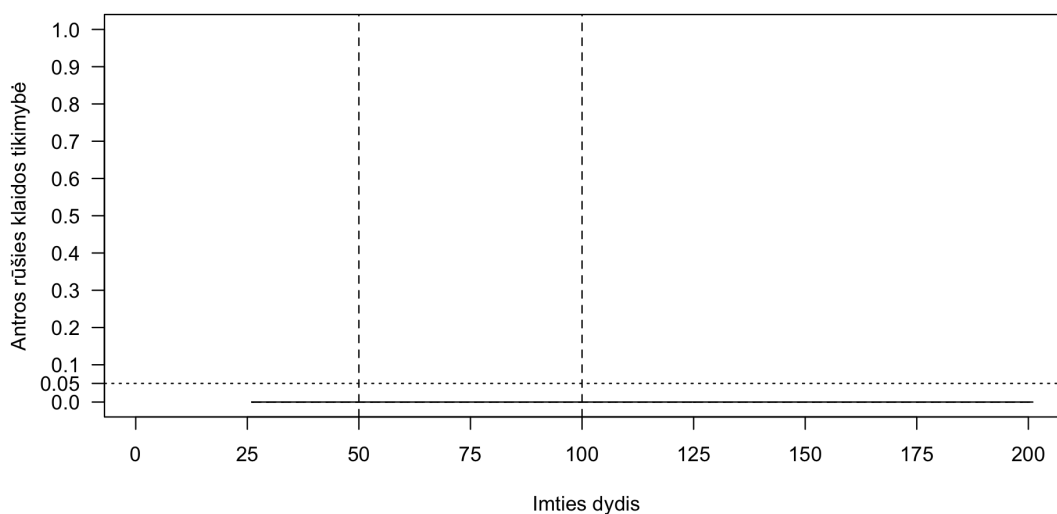


31 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 32 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 33 - naudojant maksimalios tikrinės reikšmės statistiką. Visais atvejais klaidos tikimybė yra 0.

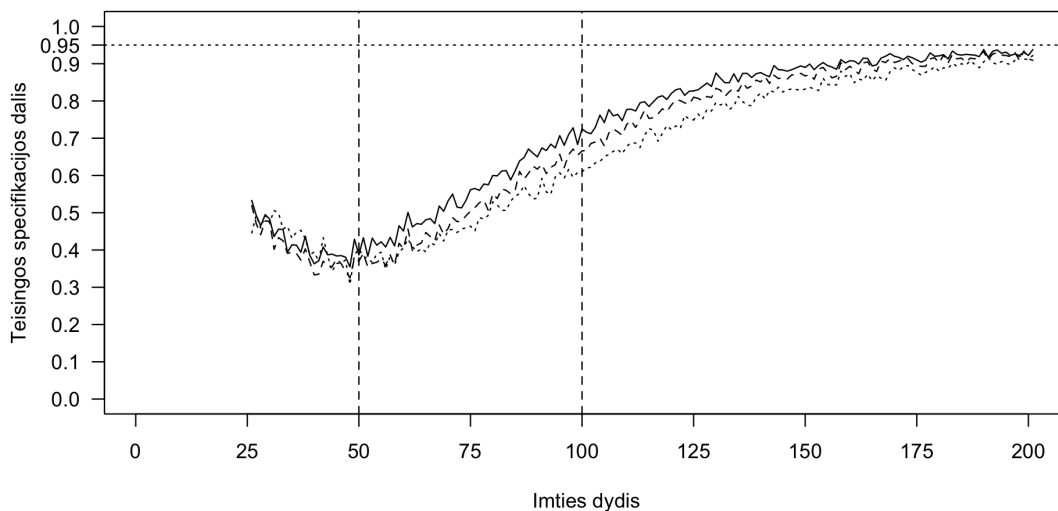


32 pav.: Antros rūšies klaida naudojant pėdsako statistiką

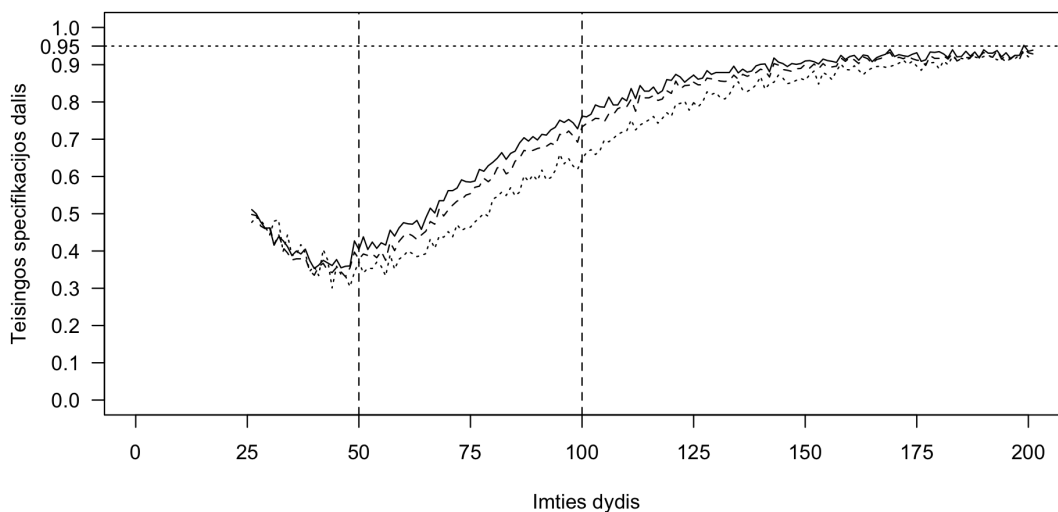


33 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Vienas kointegruojantis vektorius. Paveikslėlis 34 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 35 - naudojant maksimalios tikrinės reikšmės statistiką. Teisingai specifikuoti testai yra tiksliausi, procedūros su konstantomis yra šiek tiek silpnesnės, o procedūros su trendais pačios silpniausios, kai imtys yra vidutinės arba didelės. Mažose imtyse visų specifikacijų procedūros yra vienodos galios.

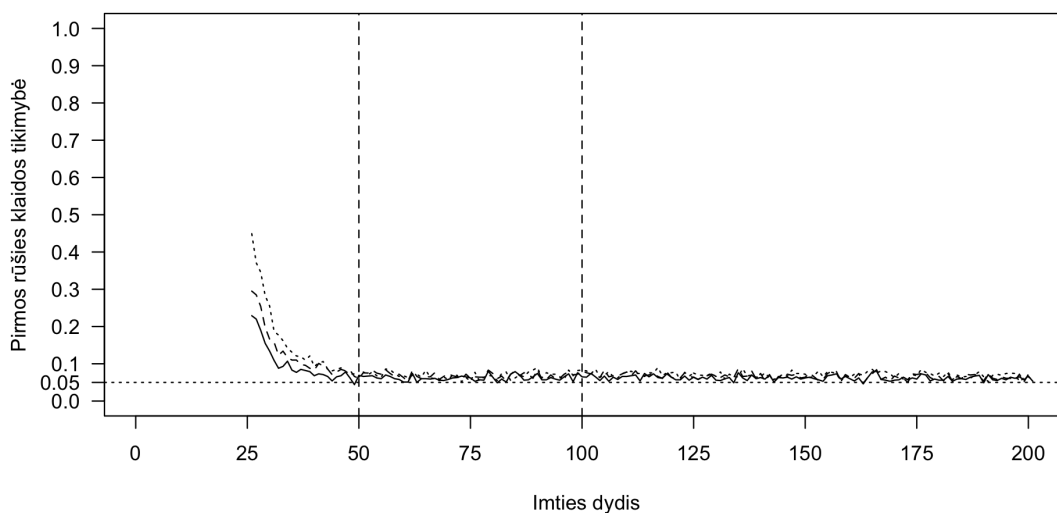


34 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

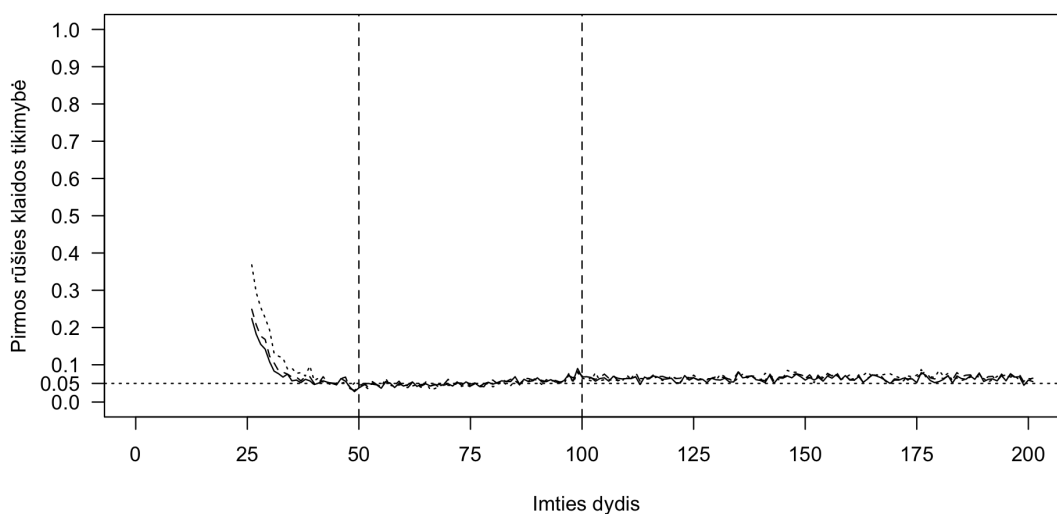


35 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 36 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 37 - naudojant maksimalios tikrinės reikšmės statistiką. Visų specifikacijų procedūros pateikia beveik identiškus rezultatus - klaidos tikimybė yra apie 5%, kai imtys yra didelės arba vidutinės. Mažose imtyse tinkamai specifikuotos procedūros yra tiksliausios.

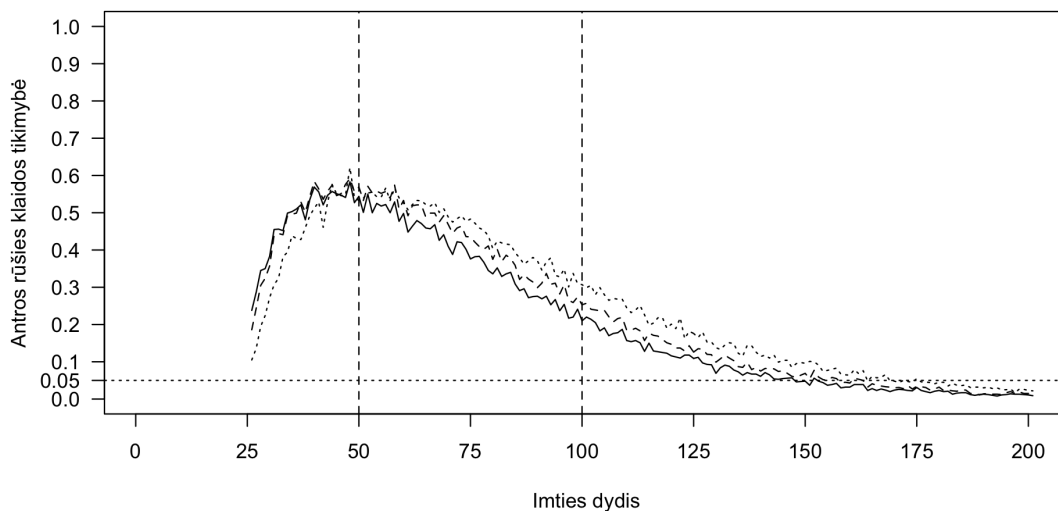


36 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

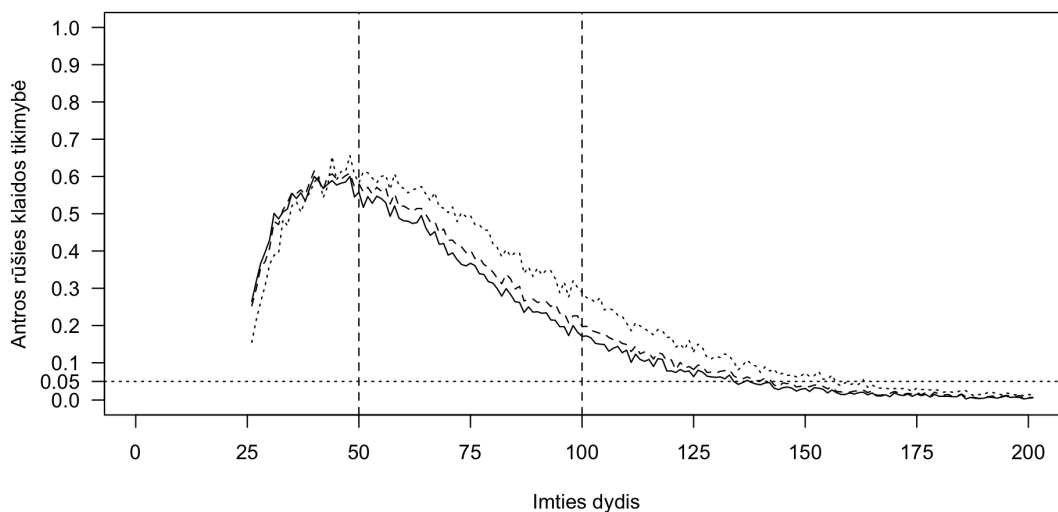


37 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 38 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 39 - naudojant maksimalios tikrinės reikšmės statistiką. Mažose imtyse mažiausią tikimybę padaryti antros rūšies klaidą turi testas su trendu, o testas be deterministinių komponentų ir testas su konstanta turi panašų tikslumą. Vidutinėse ir didelėse imtyse tiksliausi yra testai be deterministinių komponentų, testai su laisvuju nariu yra mažiau tikslūs, o testai su trendu dažniausiai klysta.



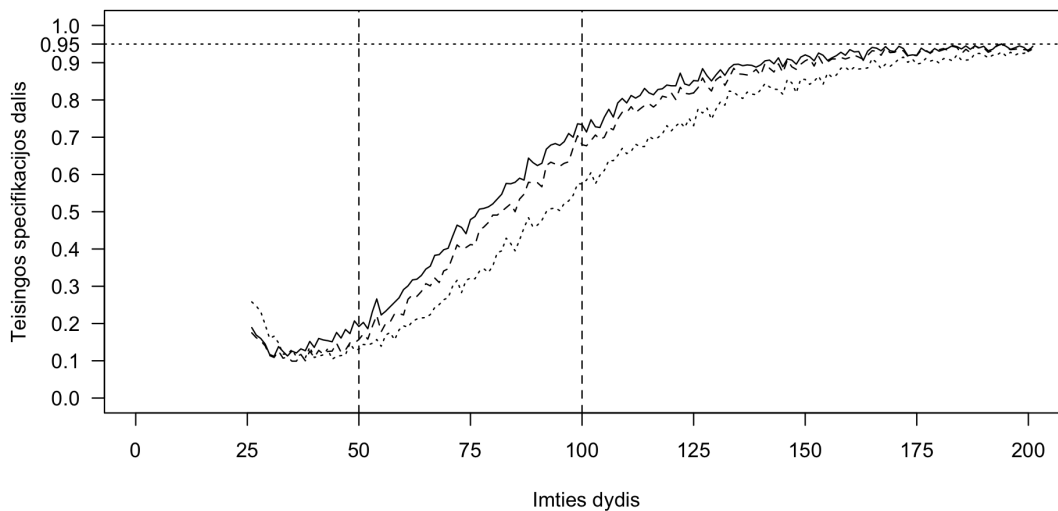
38 pav.: Antros rūšies klaida naudojant pėdsako statistiką



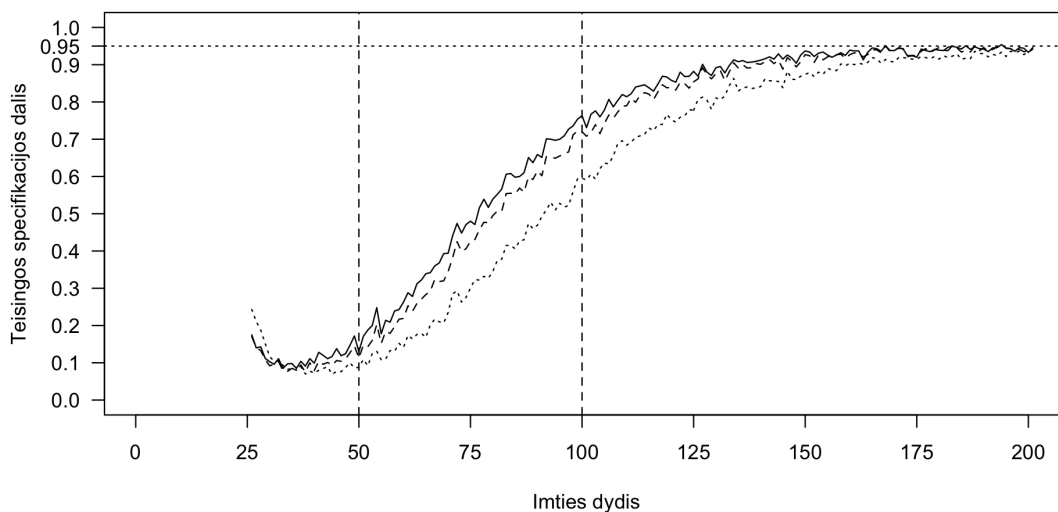
39 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Du kointegruojantys vektoriai. Paveikslėlis 40 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 41 - naudojant maksimalios tikrinės reikšmės statistiką, tiriant antros rūšies klaidas. Teisingai specifikuoti testai yra tiksliausi, procedūros su konstantomis yra šiek tiek silpnesnės, o procedūros su trendais pačios silpniausios. Rezultatai ypač išsiskiria, vidutinio dydžio imtyse, kada testai be deterministinių komponentų yra ženkliai tikslesni už testus su trendais. Testai su laisvuju nariu yra mažiau tikslūs už testus be deter-

ministinių komponentų, bet tikslesni už testus su trendais.

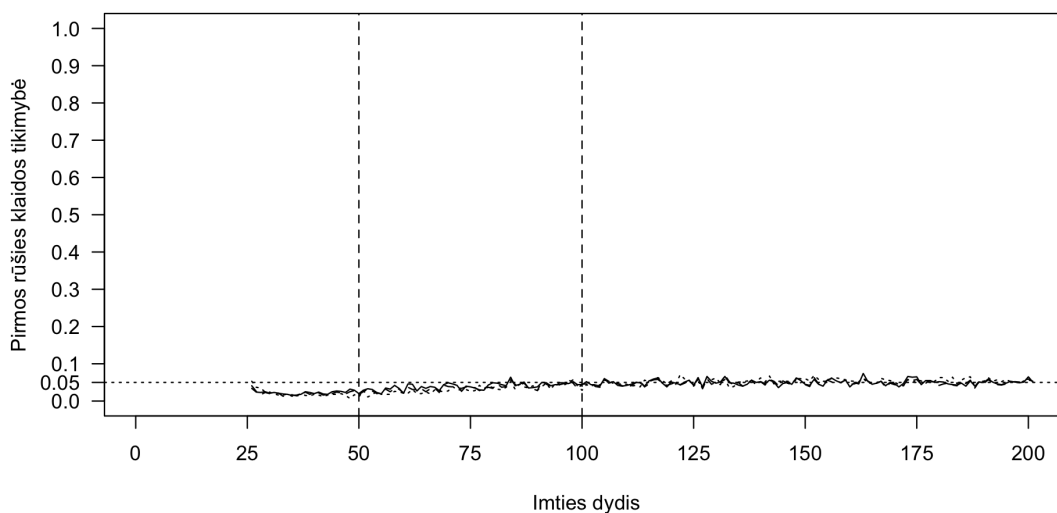


40 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

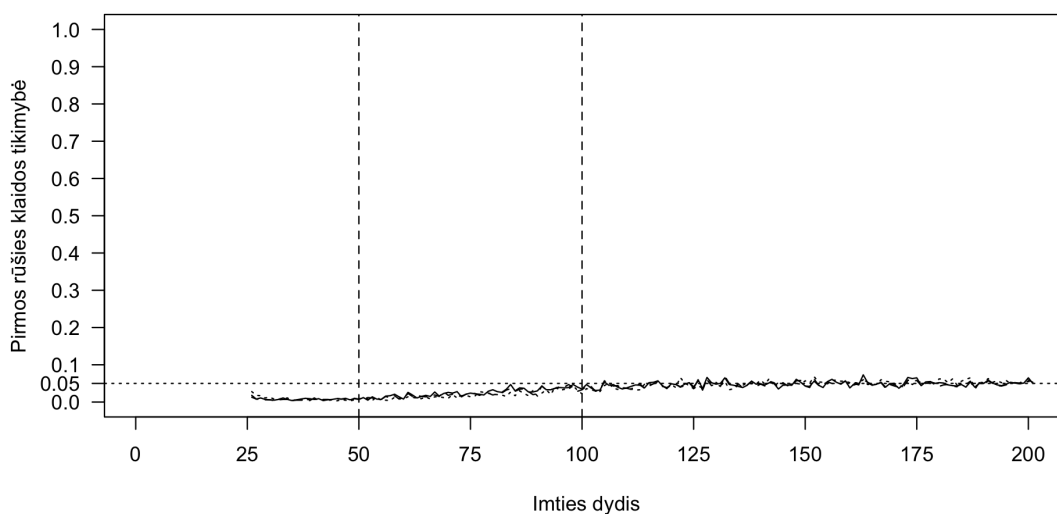


41 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 42 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 43 - naudojant maksimalios tikrinės reikšmės statistiką. Visų specifikacijų procedūros pateikia beveik identiškus rezultatus - klaidos tikimybė yra apie 5%, tačiau mažose imtyse ši tikimybė yra šiek tiek mažesnė.

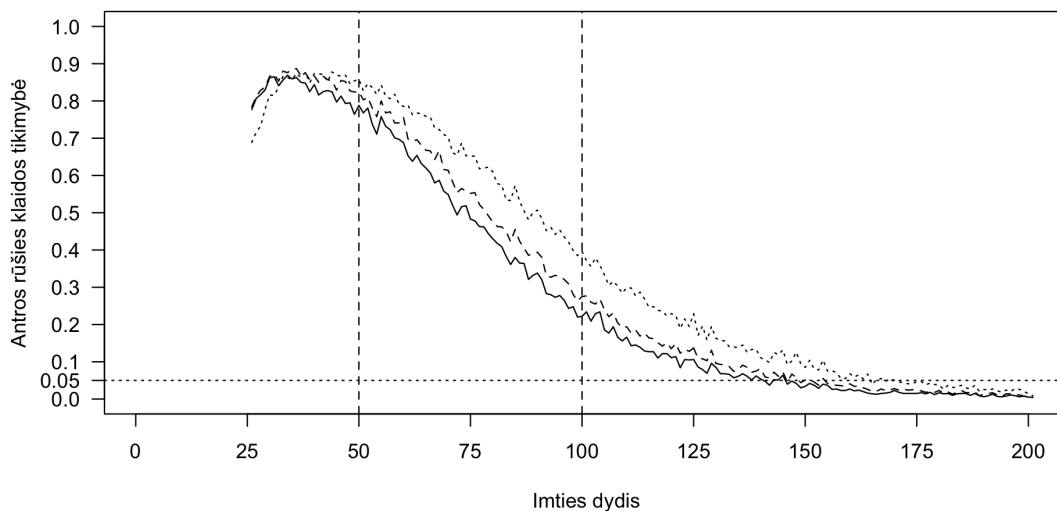


42 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

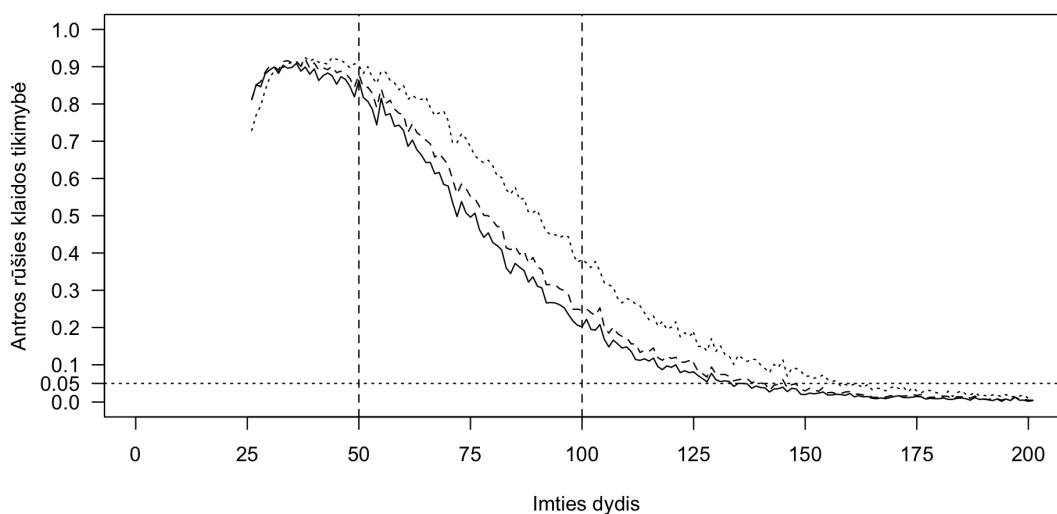


43 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 44 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 45 - naudojant maksimalios tikrinės reikšmės statistiką. Ypač mažose imtyse (< 30) tiksliausias yra testas su trendu, tuo kitais atvejais tiksliausias yra testas be deterministinių komponentų, o testas su laisvu nariu yra tikslesnis už testą su trendu, bet mažiau tikslus už testą be deterministinių komponentų.



44 pav.: Antros rūšies klaida naudojant pėdsako statistiką



45 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

3.3. Modelis 2

Modelis 2 yra trimatis VECM modelis su laisvuju nariu. Bus tiriamos trys šio modelio specifikacijos: su neintegruotais kintamaisiais, su vienu kointegruojančiu vektoriumi ir su dviem kointegruojančiais vektoriais. Lygtis 104 aprašo modelio specifikaciją su skirtingais stochastiniais trendais, lygtis 105 aprašo modelio specifikaciją su vienu kointegruojančiu vektoriumi, o lygtis 106 aprašo modelio specifikaciją su dviem kointegruojančiais vektoriais.

$$\begin{aligned}
x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + \varepsilon_{xt} \\
y_t &= \alpha_y m_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + \varepsilon_{yt} \\
z_t &= \alpha_z n_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + \varepsilon_{zt}
\end{aligned} \tag{104}$$

kur l_t , m_t ir n_t yra atsitiktinai sugeneruoti vienetinės šaknies procesai.

$$\begin{aligned}
x_t &= \alpha_x l1_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + \varepsilon_{xt} \\
y_t &= \alpha_y l2_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + \varepsilon_{yt} \\
z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + \varepsilon_{zt}
\end{aligned} \tag{105}$$

kur $l1_t$ ir $l2_t$ yra atsitiktinai sugeneruoti vienetinės šaknies procesai, o $l_t = l1_t + l2_t$.

$$\begin{aligned}
x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + \varepsilon_{xt} \\
y_t &= \alpha_y l_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + \varepsilon_{yt} \\
z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + \varepsilon_{zt}
\end{aligned} \tag{106}$$

kur l_t yra atsitiktinai sugeneruotas vienetinės šaknies procesas.

Modelis 5 turi tokią VECM reprezentaciją:

$$\begin{pmatrix} \Delta x_t \\ \Delta y_t \\ \Delta z_t \end{pmatrix} = \Pi \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} + \Gamma_1 \begin{pmatrix} \Delta x_{t-1} \\ \Delta y_{t-1} \\ \Delta z_{t-1} \end{pmatrix} + \Gamma_2 \begin{pmatrix} \Delta x_{t-2} \\ \Delta y_{t-2} \\ \Delta z_{t-2} \end{pmatrix} + \tag{107}$$

$$+ \Gamma_3 \begin{pmatrix} \Delta x_{t-3} \\ \Delta y_{t-3} \\ \Delta z_{t-3} \end{pmatrix} + \Gamma_4 \begin{pmatrix} \Delta x_{t-4} \\ \Delta y_{t-4} \\ \Delta z_{t-4} \end{pmatrix} + \begin{pmatrix} \varepsilon_{xt} \\ \varepsilon_{yt} \\ \varepsilon_{zt} \end{pmatrix} \tag{108}$$

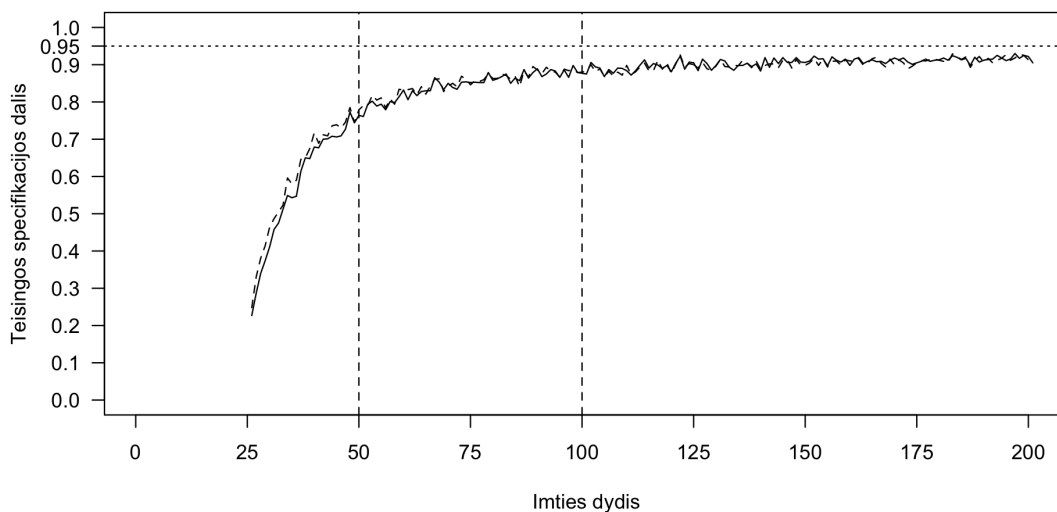
kur

$$\Pi = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{e1} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{e2} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{e3} \end{pmatrix} \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} \\ \beta_{12} & \beta_{22} & \beta_{32} \\ \beta_{13} & \beta_{23} & \beta_{33} \\ e_x & e_y & e_z \end{pmatrix} \quad (109)$$

3.3.1. Johansen procedūra be deterministinių komponentų

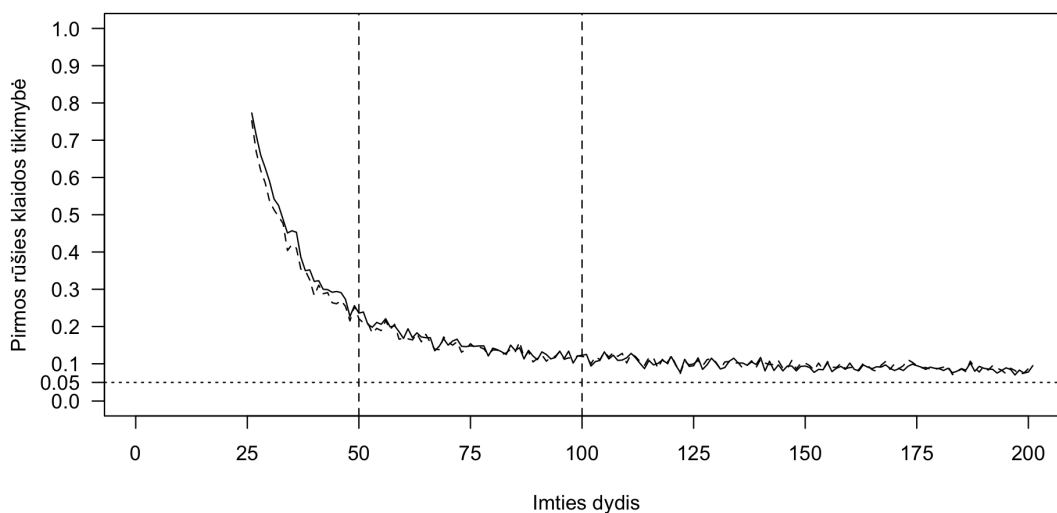
Johansen procedūra be deterministinių komponentų nėra tinkama šiam modeliui, nes kointegracijos komponente yra konstanta.

Nekointegruoti kintamieji. Pirmiausia Johansen procedūra be deterministinių komponentų bus atlikta nekointegruotai modelio specifikacijai. Paveikslėlis 46 parodo kokia dalis testų nustato kointegruojančių vektorių skaičių teisingai. Šiuo atveju tiek pėdsako, tiek maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, tačiau maksimalios tikrinės reikšmės procedūra yra šiek tiek pranašesnė mažose imtyse. Mažose imtyse testo galia labai silpna tarp 20% ir 80%, o vidutinėse ir didelėse imtyse stipri - tarp 80% ir 90%.



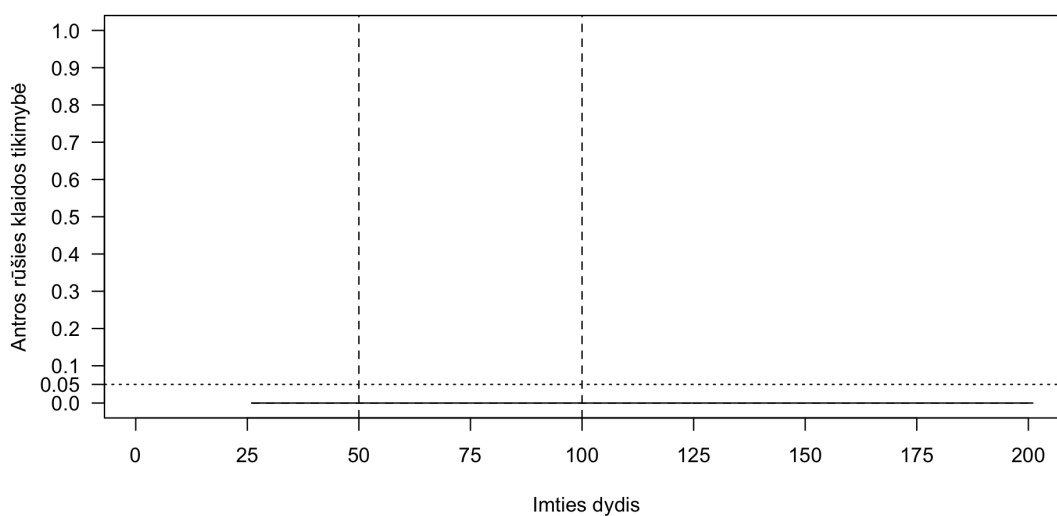
46 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 47 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus. Mažose imtyse klaidos tikimybė yra labai didelė, tačiau sparčiai mažėjanti didėjant imties dydžiui.



47 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

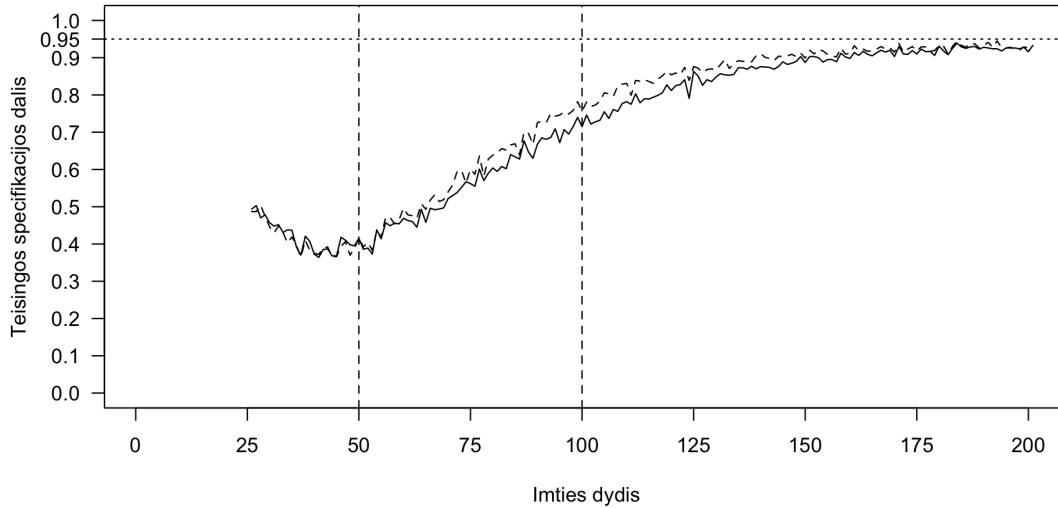
Paveikslėlis 48 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Kadangi kointegruojančių vektorių nėra, tai ir antros rūšies klaidos padaryti neįmanoma, todėl tikimybė yra 0.



48 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

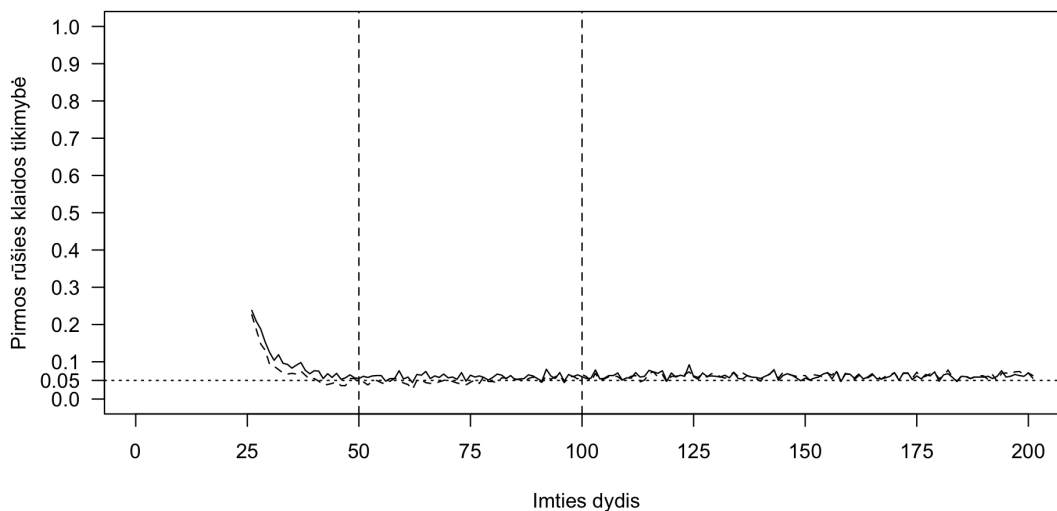
Vienas kointegruojantis vektorius. Tuo atveju, kai visos trys modelio lygtys turi tą patį stochastinį trendą, pėdsako ir maksimalios tikrinės reikšmės testai

pateikia gan panašius rezultatus. Paveikslėlis 49 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra ganėtinai silpnas mažose imtyse tarp 30% ir 55%. Vidutinėse imtyse testas reikšmingai stiprėja iki 80%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



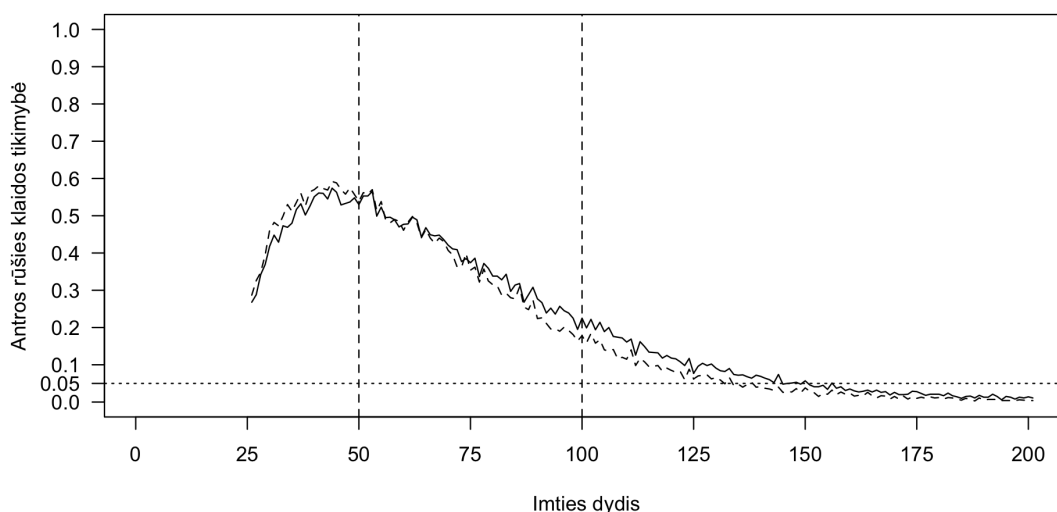
49 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 49 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Kai imtys yra vidutinės arba didelės tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos padaro šią klaidą maždaug 5% - 10% atvejų. Mažose imtyse šios klaidos tikimybė yra šiek tiek didesnė.



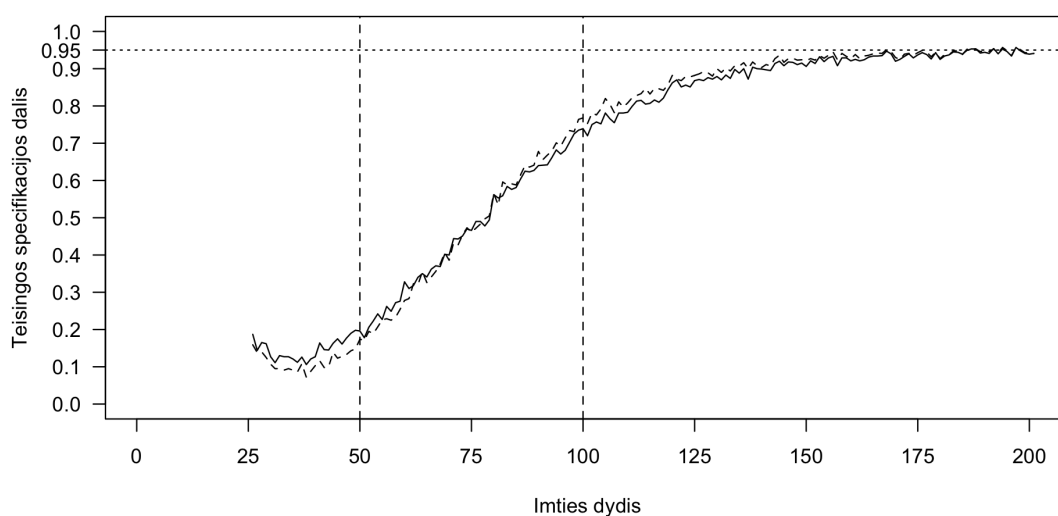
50 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 51 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 60%. Vidutinėse imtyse šios klaidos tikimybė mažėja, o didelėse imtyse priartėja prie pat 0. Pėdsako testas yra pranašesnis mažose imtyse, tačiau maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



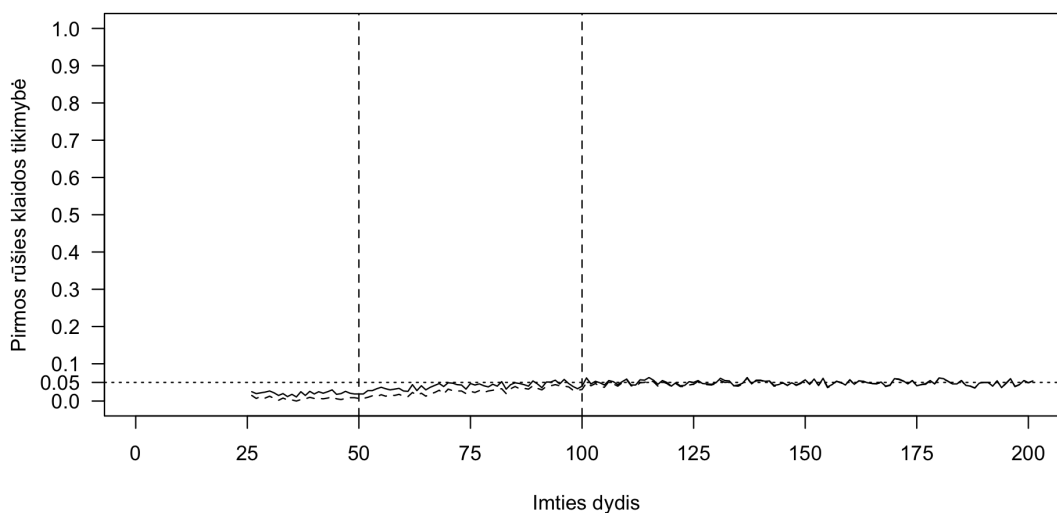
51 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Tuo atveju, kai modelis turi du kointegruojančius vektorius testų galia yra šiek tiek silpnesnė nei vieno kointegruojančio vektoriaus atveju. Paveikslėlis 52 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testo galia mažose imtyse yra ypatingai maža - tarp 5% ir 20%. Vidutinėse imtyse testas reikšmingai stiprėja iki 75%, o didelėse imtyse pasiekia 95%. Pėdsako testas yra pranašesnis mažose imtyse, tačiau maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125.



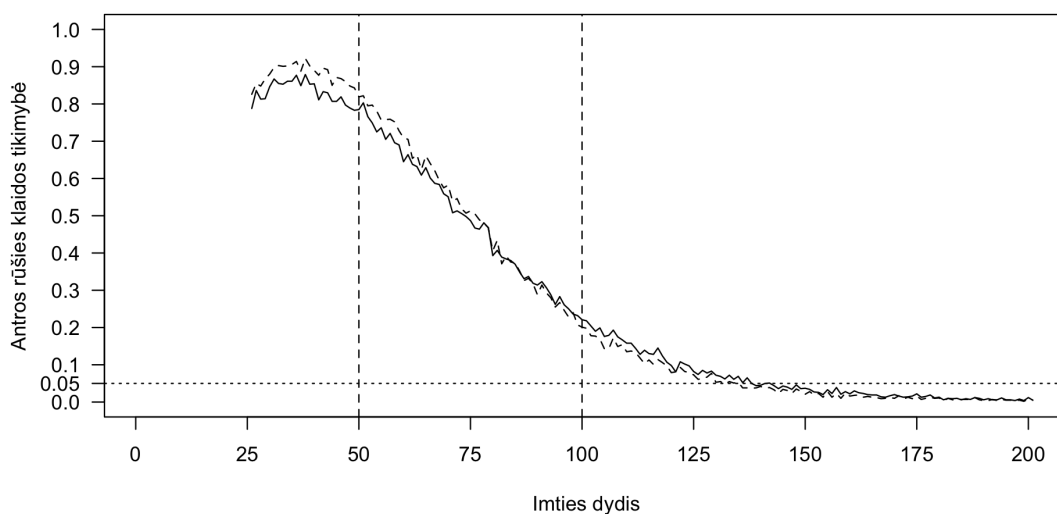
52 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 52 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji turi du kointegruojančius vektorius. Tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos padaro šią klaidą maždaug 5% atveju, kaip ir tikėtasi pagal reikšmingumo lygmenį, kai imtys yra didelės. Mažose ir vidutinėse imtyse testai pirmos rūšies klaidą padaro rečiau - tarp 0% ir 5%.



53 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 54 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 95%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie pat 0.

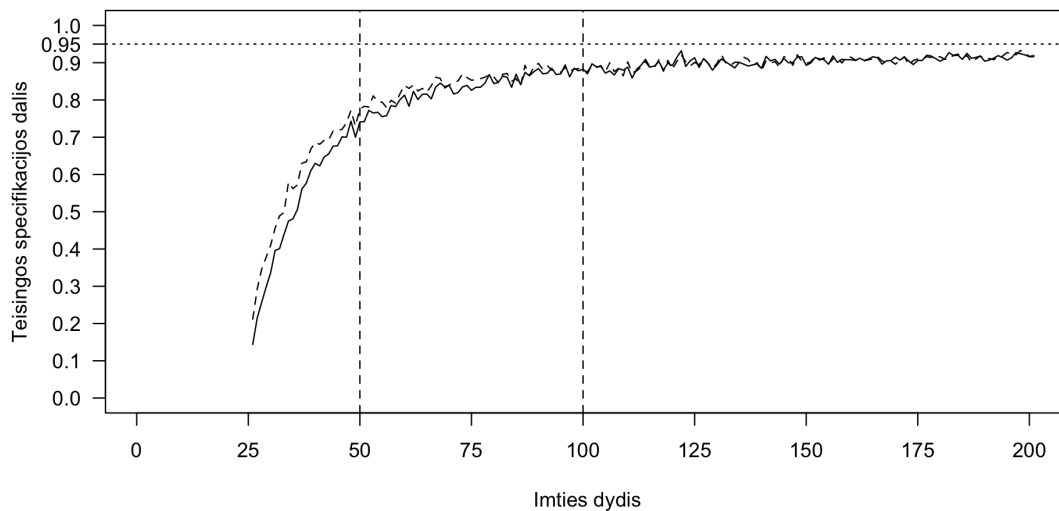


54 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.3.2. Johansen procedūra su konstanta

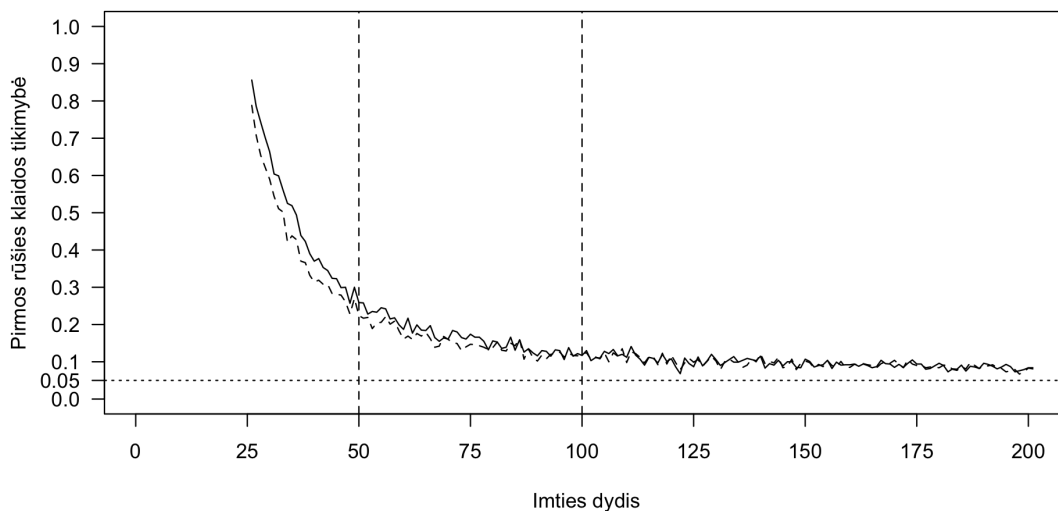
Johansen procedūra su konstanta yra pati tinkamiausia šiam modeliui.

Nekointegruoti kintamieji. Paveikslėlis 55 parodo kokia dalis Johansen procedūros su konstanta testų nustato kointegruojančių vektorių skaičių teisingai. Rezultatai panašūs kaip ir Johansen procedūros be deterministinių komponentų atveju: tiek pėdsako, tiek maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, kai imtys yra vidutinės arba didelės, tačiau, kai imtys yra mažos, maksimalios tikrinės reikšmės testas pateikia tikslesnius rezultatus. Mažose imtyse testo galia yra labai silpna tarp 15% ir 75%, o vidutinėse ir didelėse imtyse stipri - tarp 80% ir 95%.



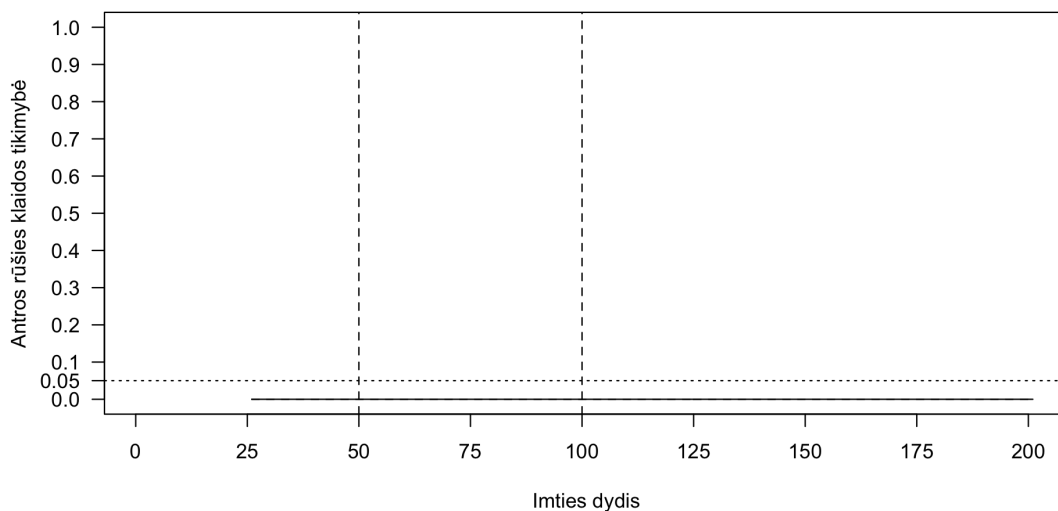
55 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 56 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, kai imtys yra vidutinio arba didelio dydžio. Kai imtys yra mažos maksimalios tikrinės reikšmės testas yra pranašesnis. Mažose imtyse klaidos tikimybė yra ypač didelė iki 85%, tačiau sparčiai mažėjanti didėjant imties dydžiui.



56 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

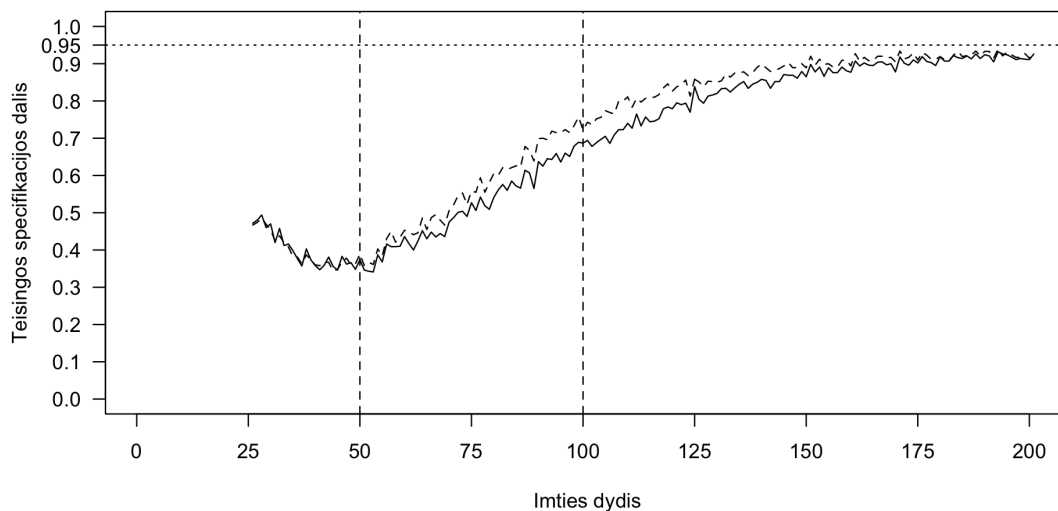
Paveikslėlis 57 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su laisvuju nariu. Kaip ir praeitu atveju šios klaidos tikimybė yra 0.



57 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

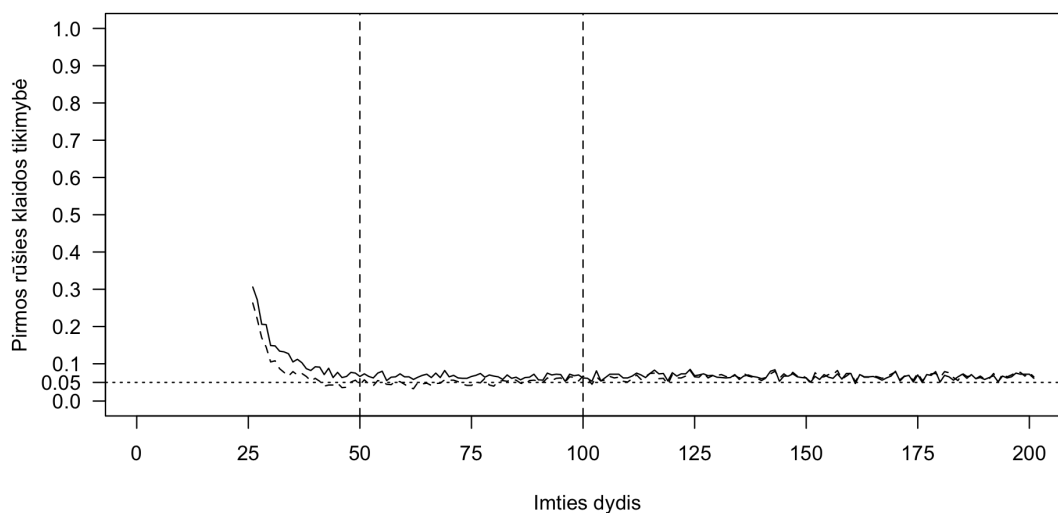
Vienas kointegruojantis vektorius. Kai modelis turi bendrą stochastinį trendą, testų rezultatai yra panašūs į Johansen procedūros atvejį be deterministinių komponentų. Paveikslėlis 58 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra labai silpnas mažose imtyse tarp 30% ir

50%. Vidutinėse imtyse testas šiek tiek sustipėja stiprėja iki 75%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės testas reikšmingai tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 60 ir 160.



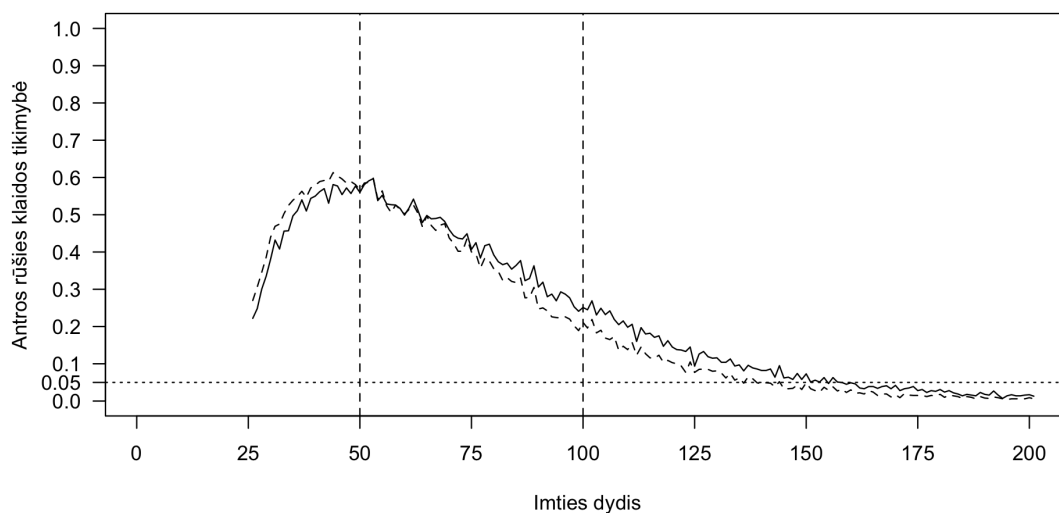
58 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 59 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja tarp 5% ir 10% vidutinėse ir didelėse imtyse. Kai imtys yra mažos klaidos tikimybės intervalas yra tarp 5% ir 35%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testas yra ženkliai geresnis nei pėdsako testas.



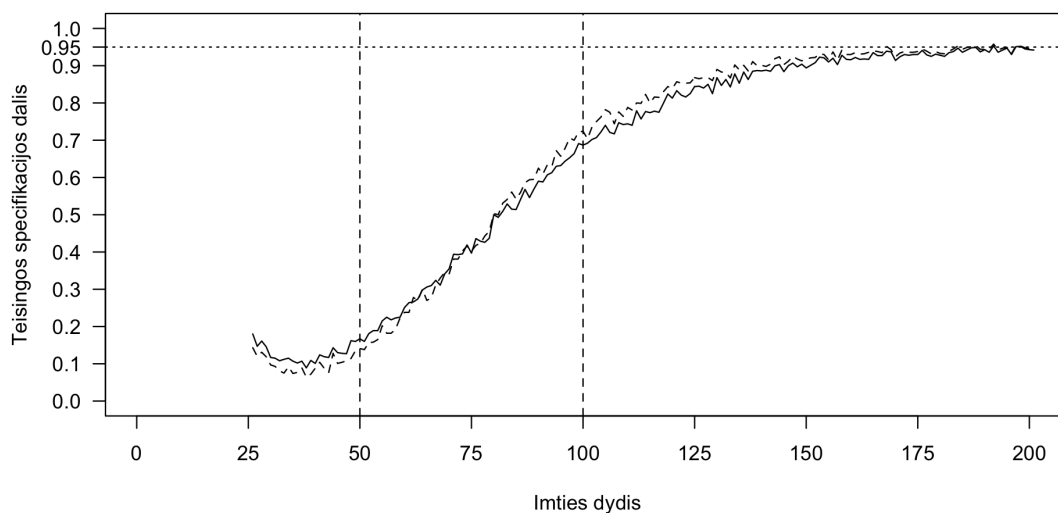
59 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 60 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji turi vieną kointegruojantį vektorių. Šios klaidos tikimybė yra didelė mažose ir vidutinėse imtyse, net iki 65%. Vidutinėse imtyse, didėjant imties dydžiui, šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Kai imties dydis yra tarp 25 ir 40, didėjant imties dydžiui, šios klaidos tikimybė sparčiai auga, po to pradeda sparčiai mažėti. Mažose imtyse pėdsako testas yra pranašesnis, o vidutinėse ir didelėse imtyse maksimalios tikrinės reikšmės testas yra tikslesnis.



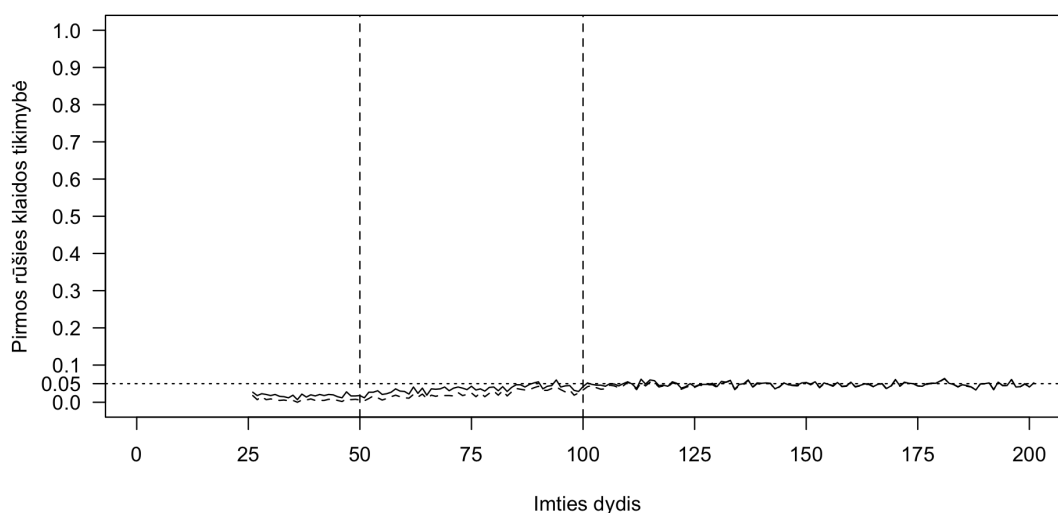
60 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Kai modelis turi du kointegruojančius vektorius, testų rezultatai skiriasi nuo kitų dviejų atvejų. Paveikslėlis 61 parodo, kiek procentų atvejų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra labai silpnas mažose imtyse tarp 5% ir 20%. Vidutinėse imtyse testas reikšmingai stiprėja iki 70%, o didelėse imtyse pasiekia 95%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra tarp 75 ir 125, o kai imtys yra mažose tikslesnis yra pėdsako testas.



61 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

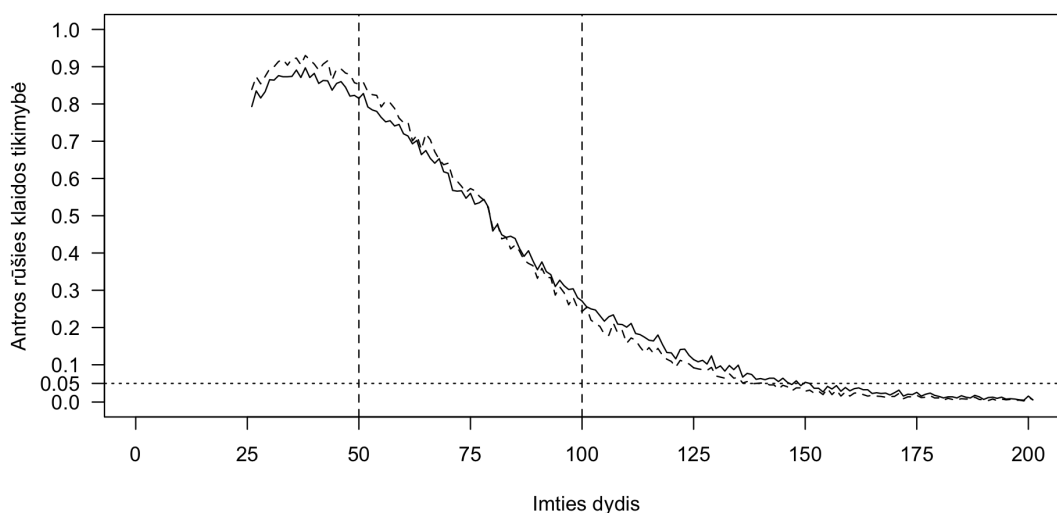
Paveikslėlis 62 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5% vidutinėse ir didelėse imtyse. Mažose imtyse šios klaidos tikimybė yra mažesnė nei 5%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testai yra pranašesni už pėdsako testus.



62 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 63 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač

didelė mažose imtyse, kur pasiekia net 95%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Mažose imtyse pėdsako testai yra ženkliai pranašesni už maksimalios tikrinės reikšmės testus, tuo tarpu, kai imties dydis yra tarp 75 ir 125 maksimalios tikrinės reikšmės testai yra tikslesni.

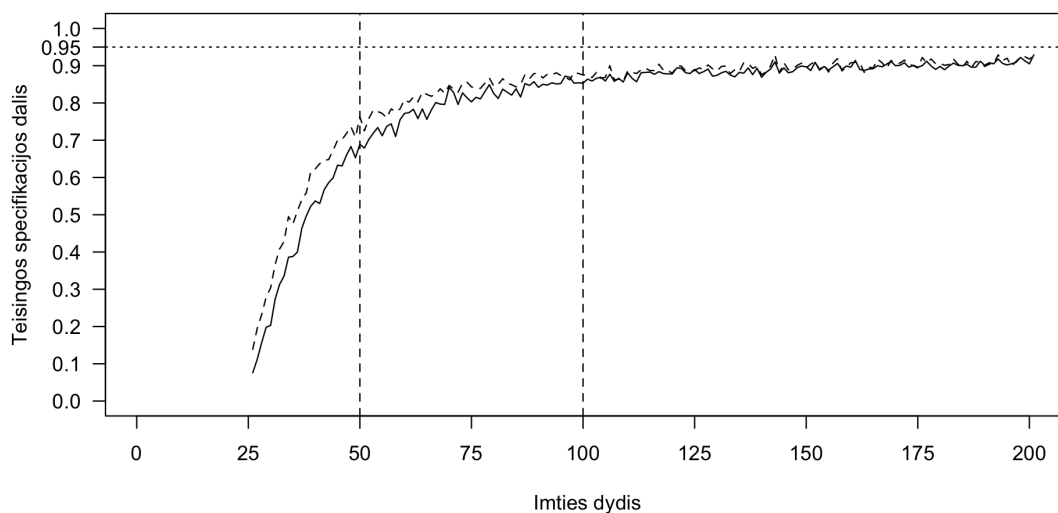


63 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.3.3. Johansen procedūra su trendu

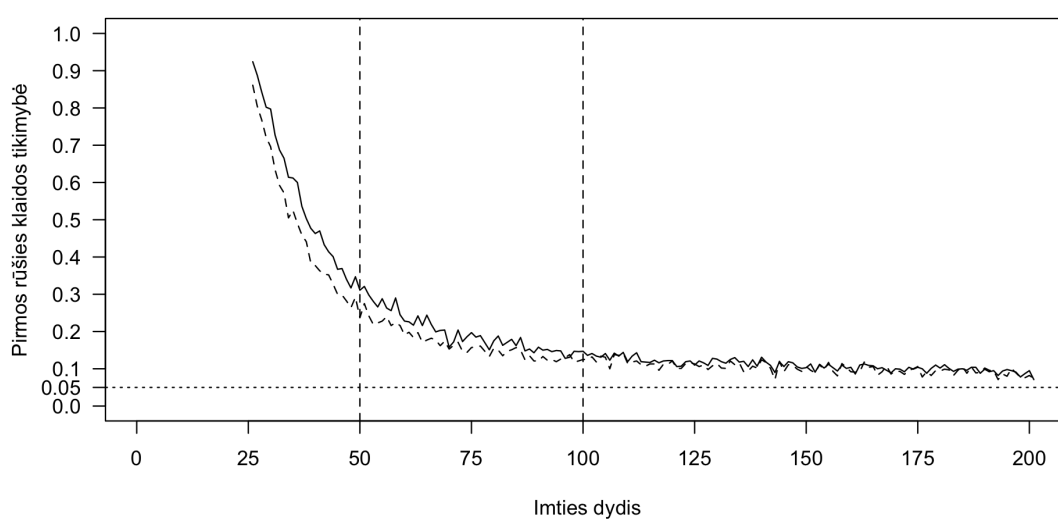
Johansen procedūra su trendu nėra pati tinkamiausia šiam modeliui, nes jame nėra trendo komponento, tuo tarpu procedūroje nėra konstantos komponento.

Nekointegruoti kintamieji. Paveikslėlis 64 parodo kokios dalis Johansen procedūros su trendu testų nustato kointegruojančių vektorių skaičių teisingai. Grafiko forma panaši į dviejų anksčiau tirtus atvejus. Testas yra ypač silpnas mažose imtyse - jo galia yra intervale tarp 5% ir 70%. Vidutinėse imtyse testas šiek tiek stiprėja nuo 70% iki 85%, o didelėse imtyse pasiekia 90%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, mažose ir vidutinėse imtyse.



64 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

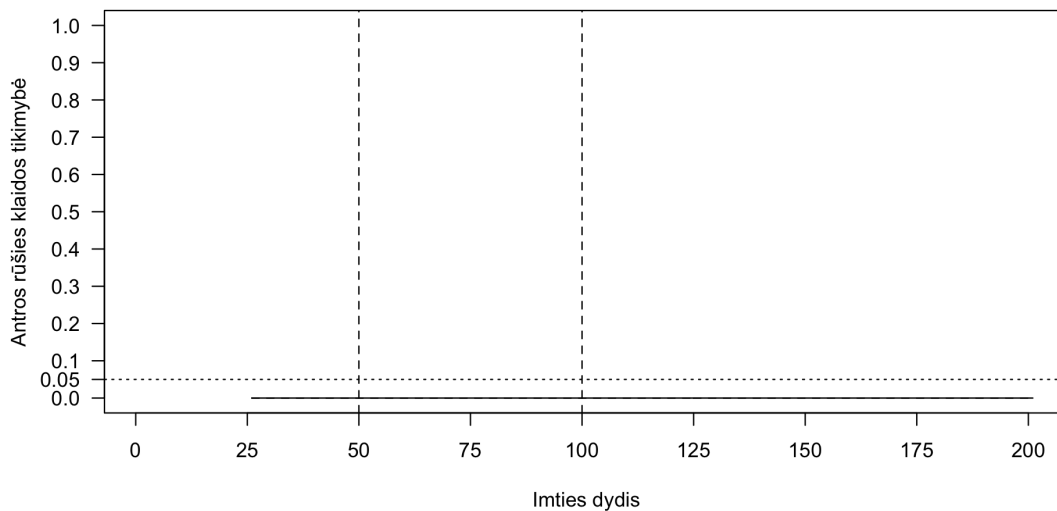
Paveikslėlis 65 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu. Klaidos tikimybė mažose imtyse labai didelė, siekianti 95%. Didėjant imties dydžiui klaidos tikimybė mažėja ir didelėse imtyse pasiekia 10%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testas yra tikslesnis už pėdsako testą.



65 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

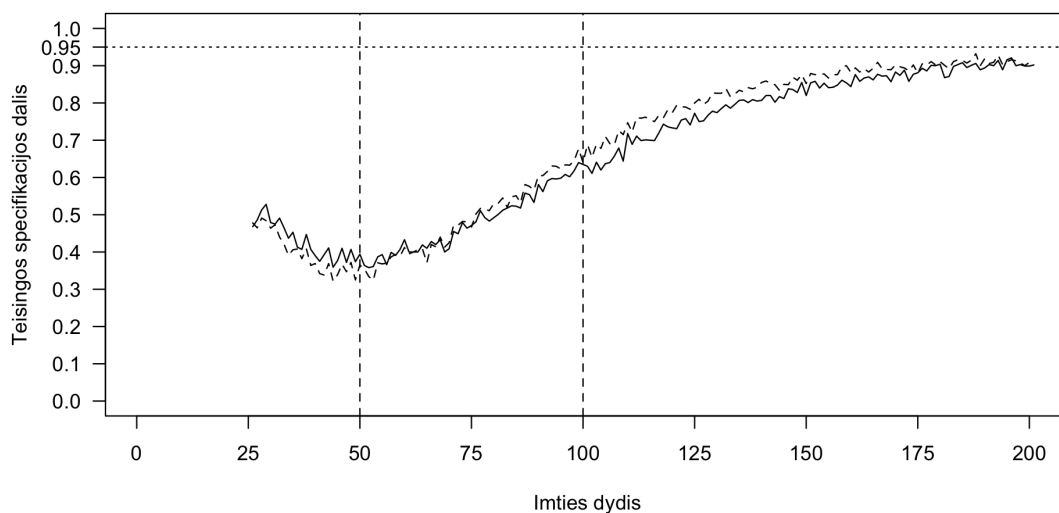
Paveikslėlis 65 parodo antros rūšies klaidos tikimybę naudojant Johansen pro-

cedūrą su trendu. Ši tikimybė yra 0.



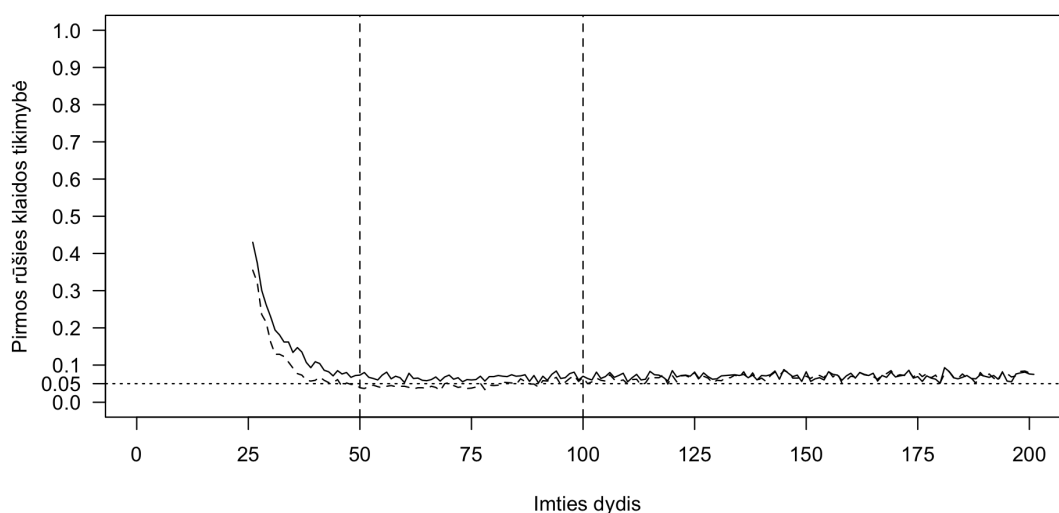
66 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Vienas kointegruojantis vektorius. Kai modelis turi vieną kointegruojantį vektorių, Johansen procedūros rezultatai yra panašūs į kitus du Johansen procedūros specifikavimo atvejus, tačiau testų galia yra šiek tiek silpnesnė. Paveikslėlis 67 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas galia mažose imtyse yra vidutinė ir svyruoja tarp 30% ir 60%, vidutinėse imtyse - tarp 35% ir 65%, o didelėse imtyse tarp 60% ir 95%. Pėdsako testas yra tikslesnis mažose imtyse, o maksimalios tikrinės reikšmės testas yra tikslesnis didelėse imtyse.



67 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

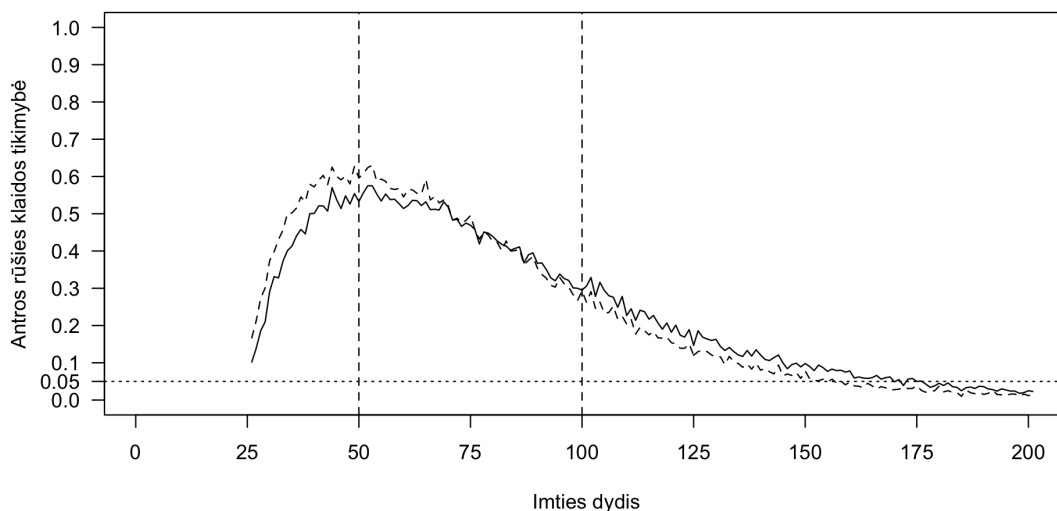
Paveikslėlis 68 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja tarp 5% ir 10%, kai imtys yra vidutinės arba didelės. Mažose imtyse klaidos tikimybė siekia 45%, tačiau sparčiai krenta didėjant imties dydžiui. Maksimalios tikrinės reikšmės testas yra pranašesnis mažose ir vidutinėse imtyse.



68 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

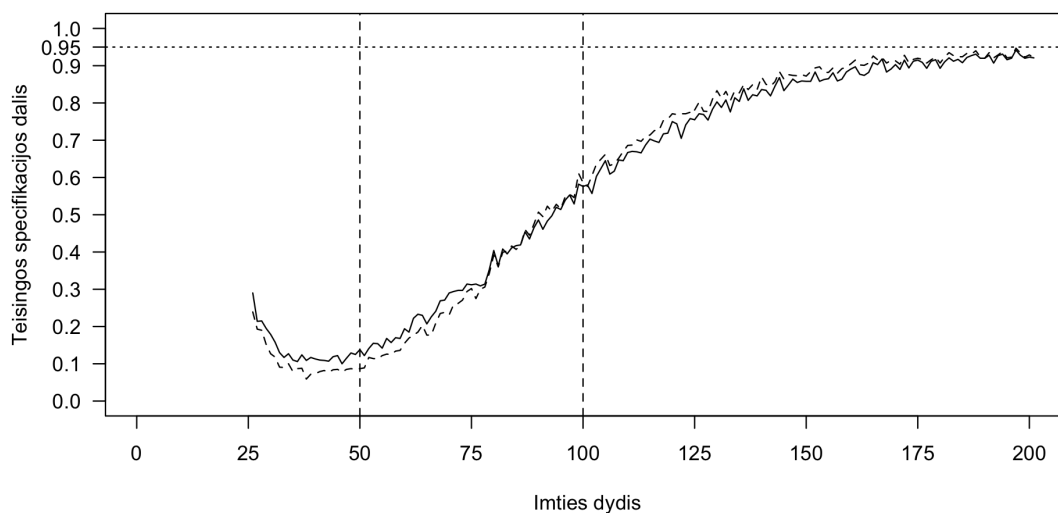
Paveikslėlis 69 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti ir turi vieną kointegruojantį

vektorių. Šios klaidos tikimybė yra didelė mažose imtyse, ir siekia iki 65%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Mažose imtyse didėjant imties dydžiui klaidos tikimybė didėja, o vidutinėse ir didelėse imtyse mažėja.



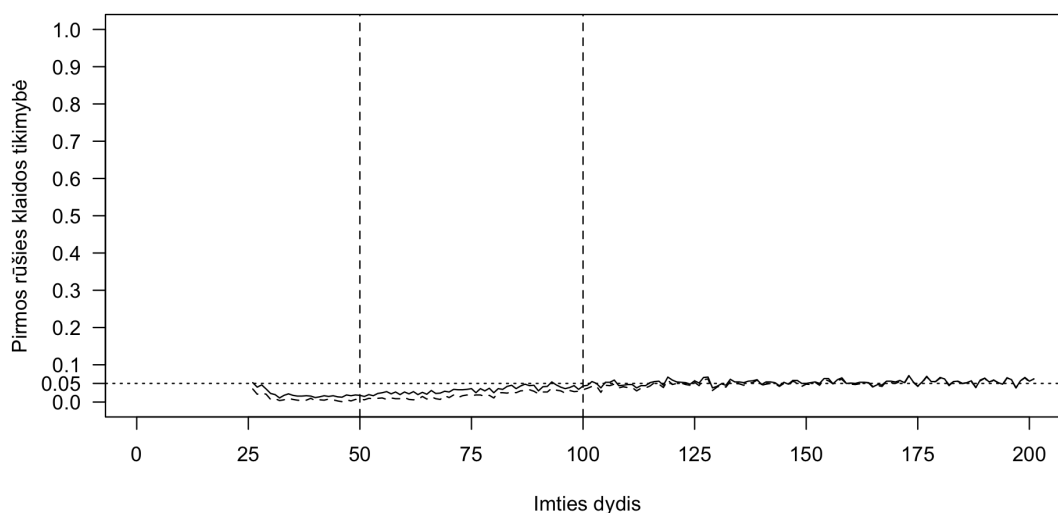
69 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Paveikslėlis 70 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai, kai kintamieji turi du kointegruojančius vektorius. Testas yra labai silpnas mažose imtyse tarp 5% ir 30%. Vidutinėse imtyse testas reikšmingai stiprėja iki 65%, o didelėse imtyse pasiekia 95%. Pėdsako testas yra pranašesnis mažose ir vidutinėse imtyse, o maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių didelėse imtyse.



70 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

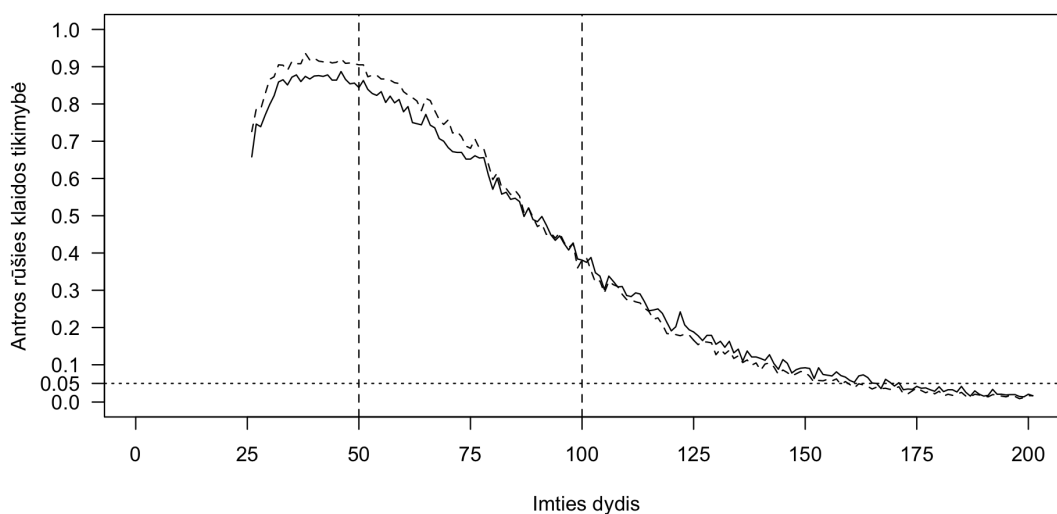
Paveikslėlis 71 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5%, kai imtys yra vidutinės ir didelės. Mažose imtyse šios klaidos tikimybės intervalas yra tarp 0% ir 5%.



71 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 72 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu, kai kintamieji yra kointegruoti ir turi du kointegruojančius vektorius. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 95%. Vidutinėse imtyse

šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Maksimalios tikrinės reikšmės testas ir šiuo atveju yra mažiau tikslus už pėdsako testą, išskyrus dideles imtis.



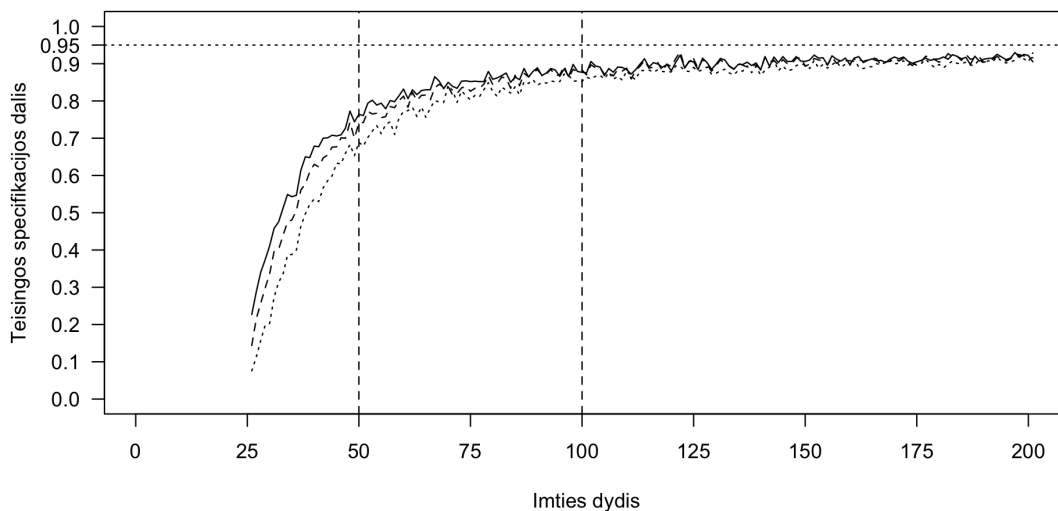
72 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.3.4. Misspecifikacijos įtaka rezultatams

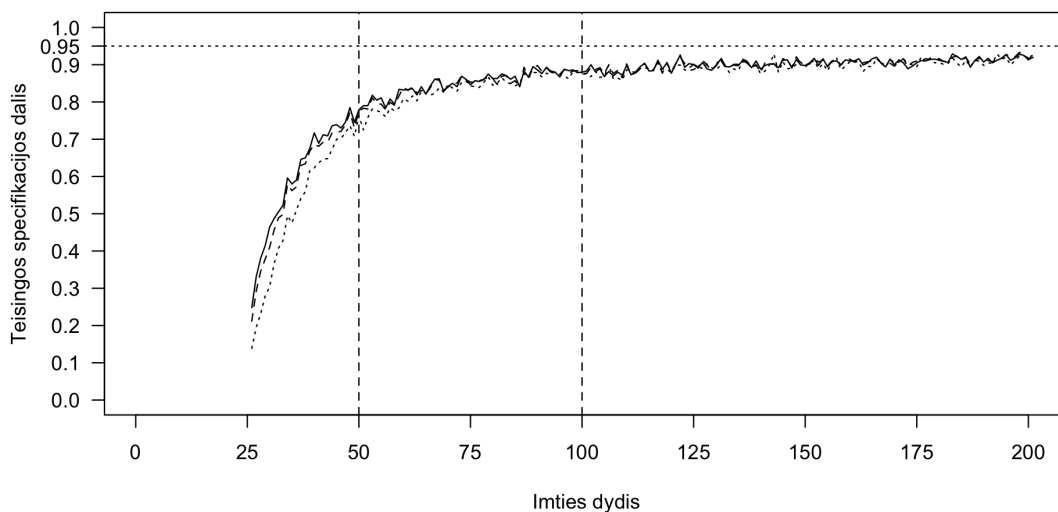
Neteisingai specifikuota Johansen procedūra įtraukiant per daug deterministinių komponentų nedaro esminės įtakos testų rezultatams. Procedūros be deterministinių komponentų yra net pranašesnės už teisingai specifikuotas procedūras. Tuo tarpu procedūros su trendo deterministiniu komponentu, bet be konstantos komponento yra tik šiek tiek silpnesnės. Didelėse imtyse visų tipų procedūrų rezultatai konverguoja.

Kadangi procedūros be deterministinių komponentų yra pačios tiksliausios nustatant kointegruojančių vektorių skaičių, siūloma naudoti jas nustatant kointegruojančių vektorių skaičių, o tada naudoti Johansen procedūrą su konstanta įvertinant kointegruojančio vektoriaus parametrų reikšmes.

Nekointegruoti kintamieji. Kai imtys yra mažos visų tipų Johansen procedūros turi vidutinę tikimybę padaryti pirmos rūšies klaidą. Vidutinėse ir didelėse imtyse testų galia yra didelė. Paveikslėlis 73 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 74 - naudojant maksimalios tikrinės reikšmės statistiką. Testas be deterministinių komponentų yra tiksliausias, procedūra su konstanta šiek tiek silpnesnė, o procedūra su trendu pati silpniausia.

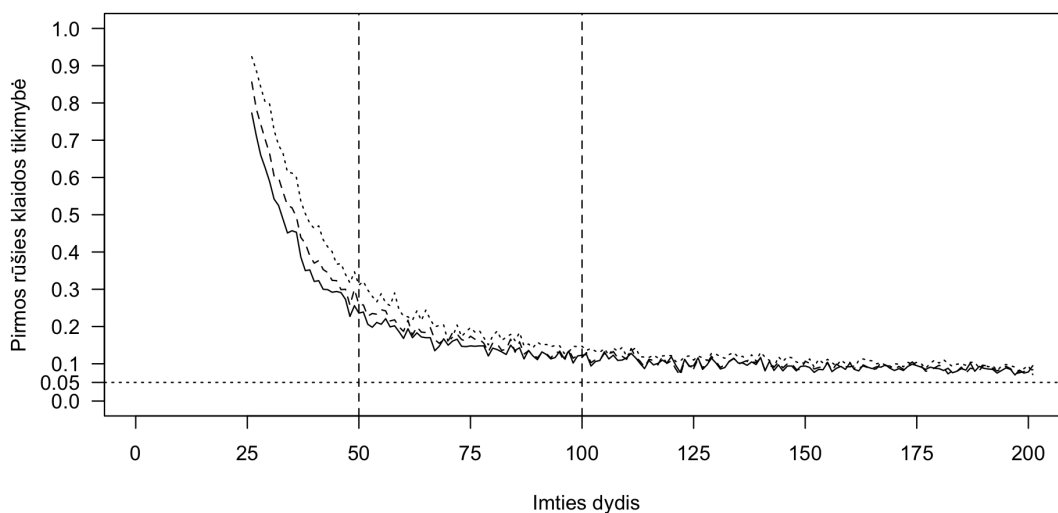


73 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

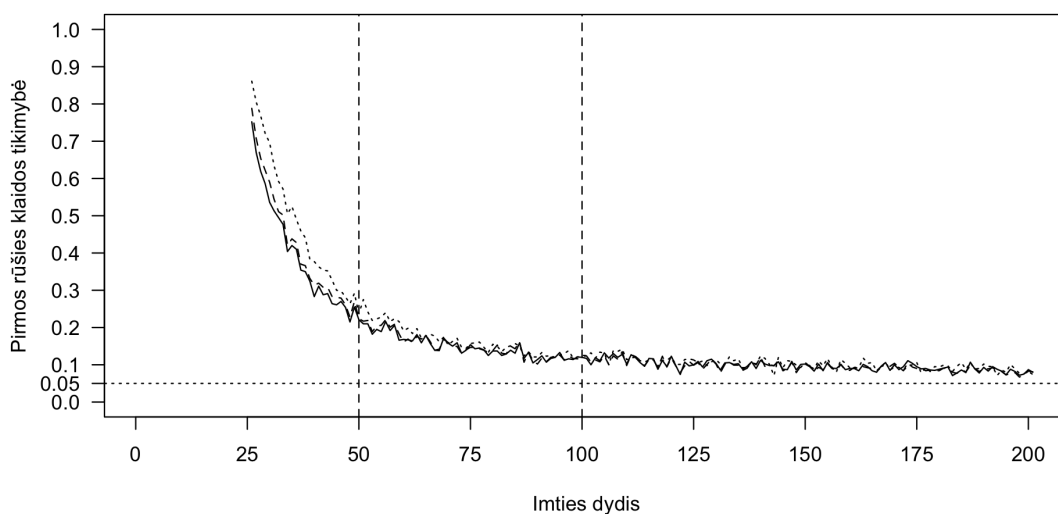


74 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 75 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 76 - naudojant maksimalios tikrinės reikšmės statistiką. Klaidos tikimybė mažiausia testo be deterministinių komponentų atveju, šiek tiek didesnė testo atveju su laisvuju nariu ir didžiausia testo atveju su trendu.

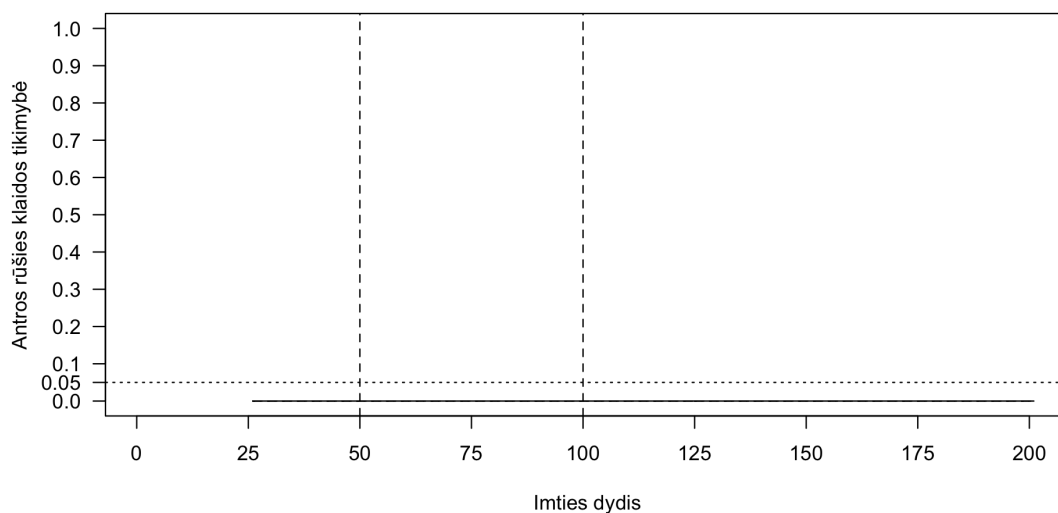


75 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

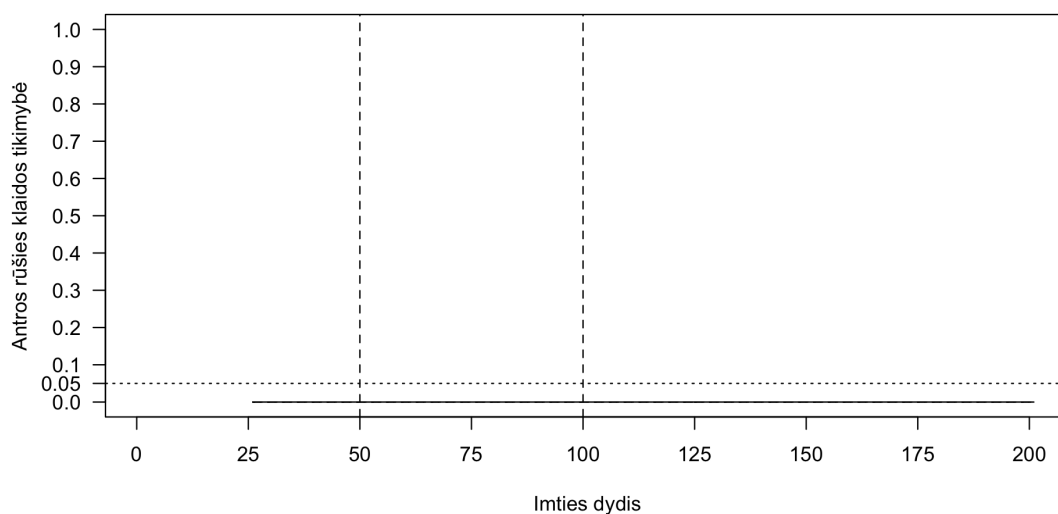


76 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 77 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 78 - naudojant maksimalios tikrinės reikšmės statistiką. Visais atvejais klaidos tikimybė yra 0.

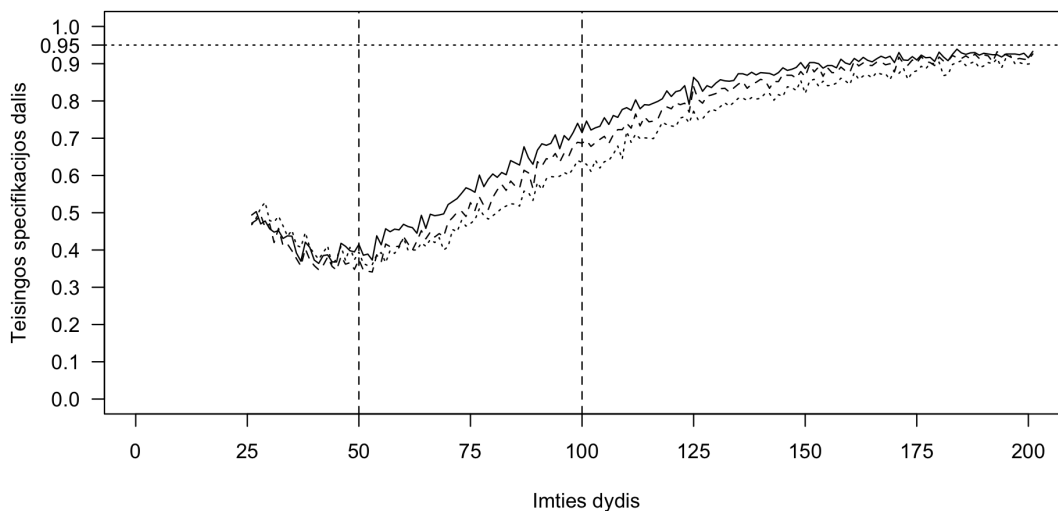


77 pav.: Antros rūšies klaida naudojant pėdsako statistiką

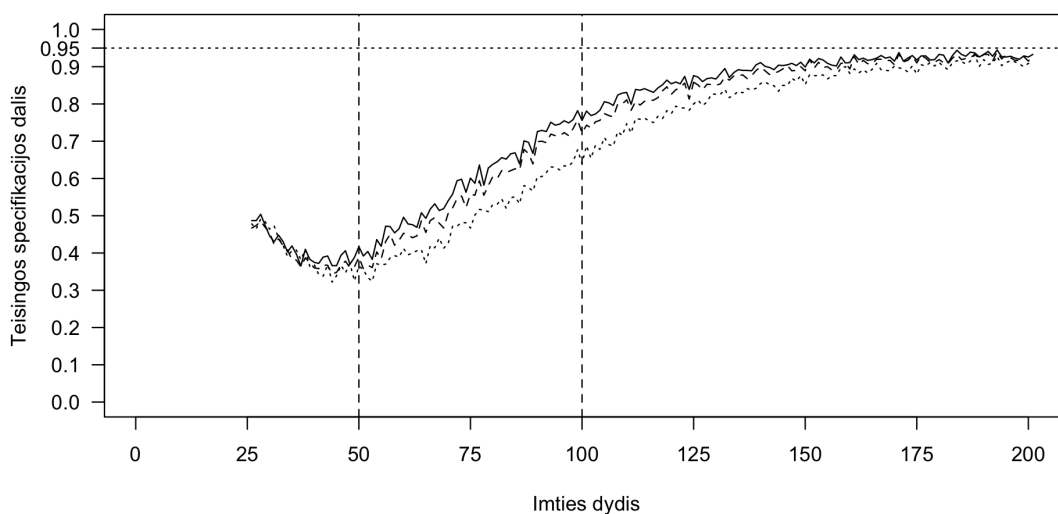


78 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Vienas kointegruojantis vektorius. Paveikslėlis 79 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 80 - naudojant maksimalios tikrinės reikšmės statistiką. Testai be deterministinių komponentų yra tiksliausi, procedūros su konstantomis yra šiek tiek silpnesnės, o testai su trendais yra patys silpniausi.

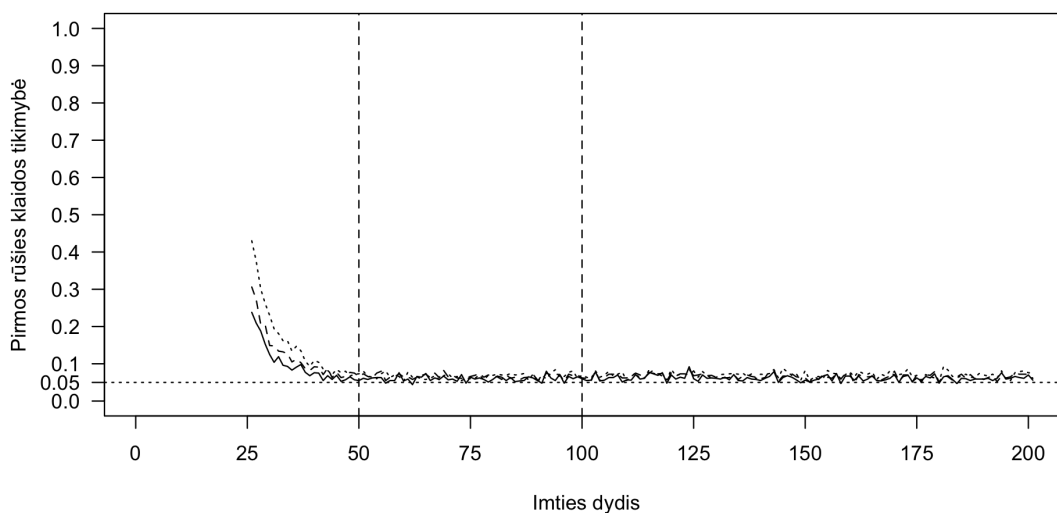


79 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

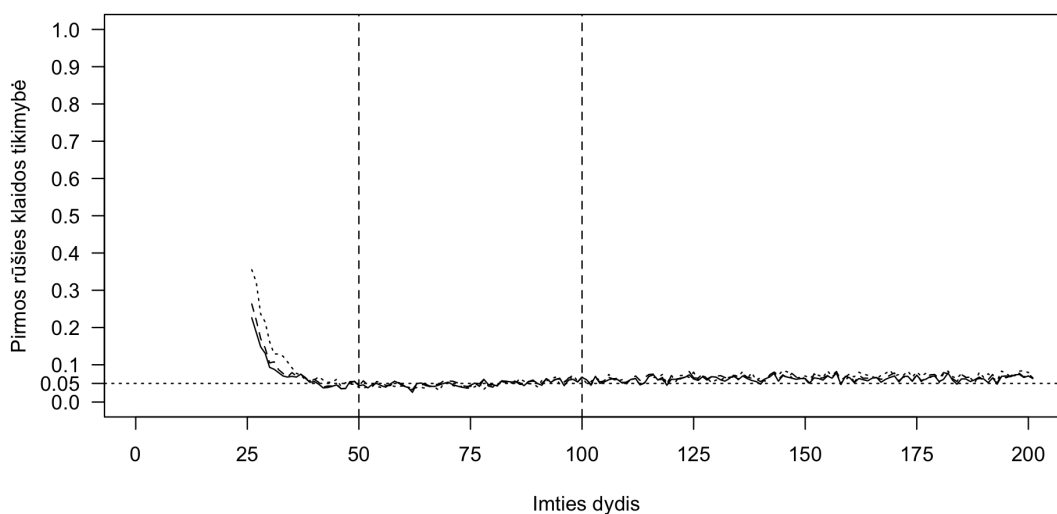


80 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 81 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 82 - naudojant maksimalios tikrinės reikšmės statistiką. Visų specifikacijų procedūros pateikia beveik identiškus rezultatus - klaidos tikimybė yra apie 5%, kai imtys yra didelės arba vidutinės. Mažose imtyse procedūros be deterministinių komponentų yra tiksliausios.

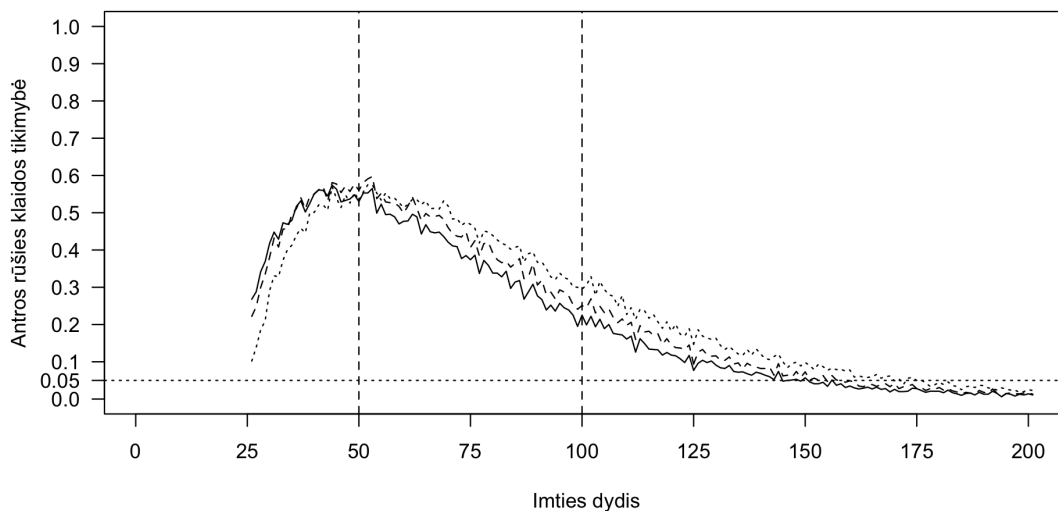


81 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

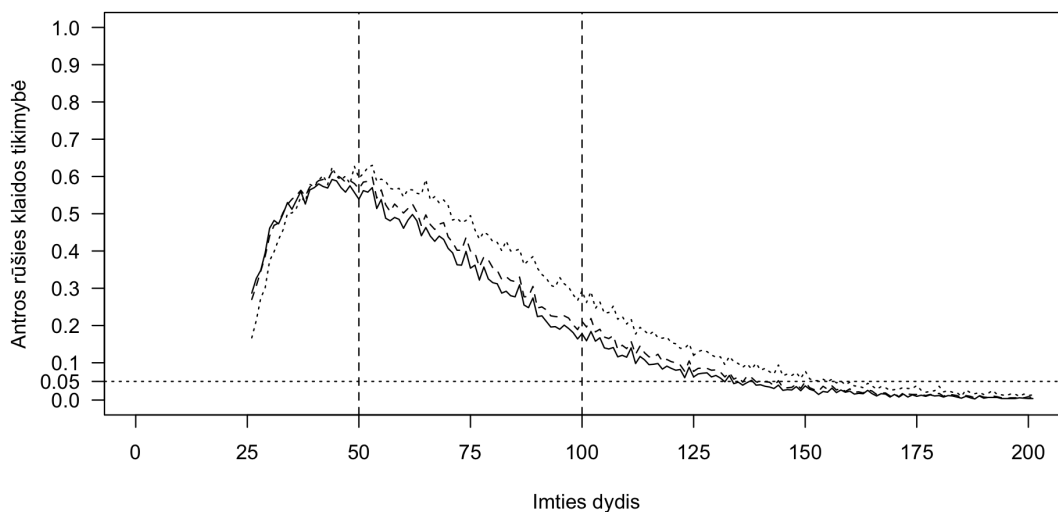


82 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 83 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 84 - naudojant maksimalios tikrinės reikšmės statistiką. Mažose imtyse mažiausią tikimybę padaryti antros rūšies klaidą turi testas su trendu, o testas be deterministinių komponentų ir testas su konstanta turi panašų tikslumą. Vidutinėse ir didelėse imtyse tiksliausi yra testai be deterministinių komponentų, testai su laisvuju nariu yra mažiau tikslūs, o testai su trendu dažniausiai klysta.



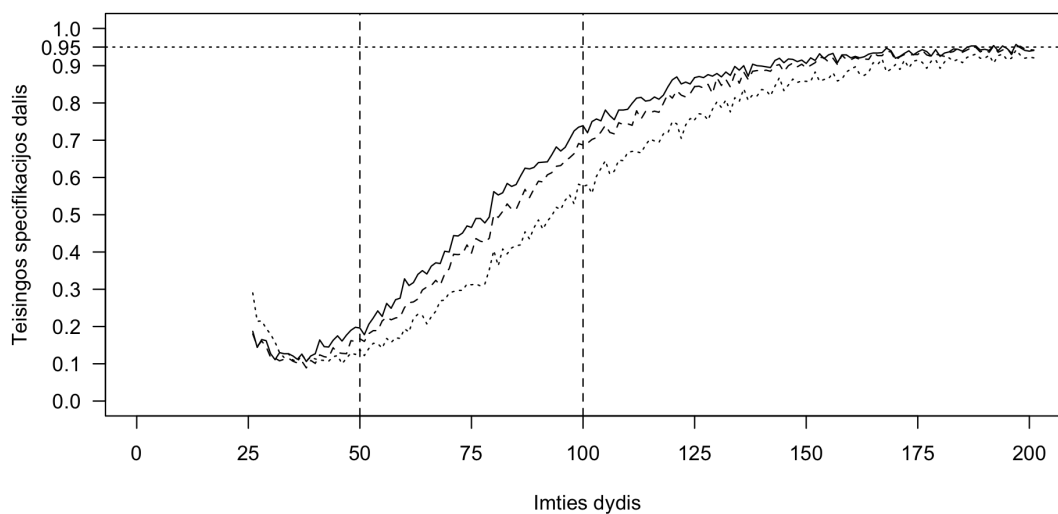
83 pav.: Antros rūšies klaida naudojant pėdsako statistiką



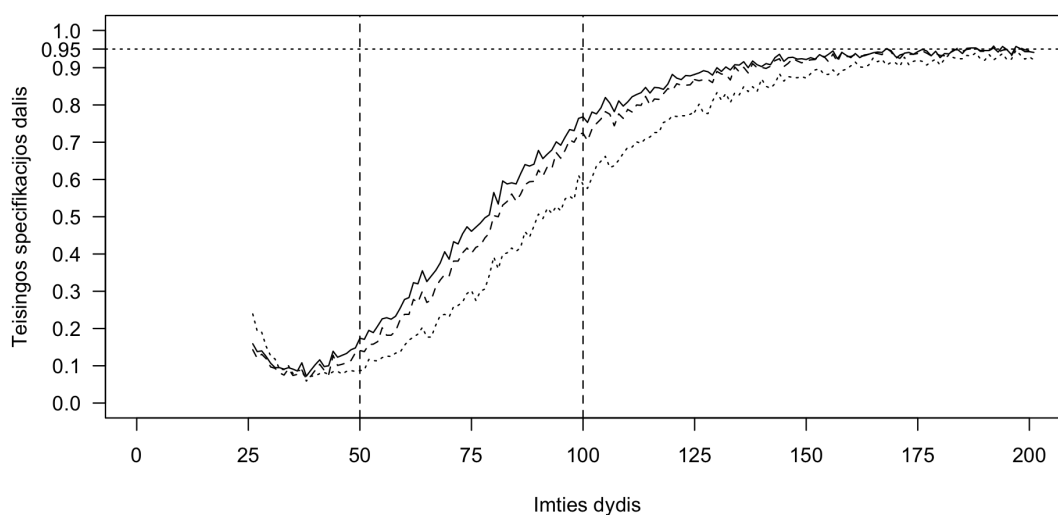
84 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Du kointegruojantys vektoriai. Paveikslėlis 85 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 86 - naudojant maksimalios tikrinės reikšmės statistiką, tiriant antros rūšies klaidas. Testai be deterministinių komponentų yra patys stipriausi, procedūros su konstantomis yra šiek tiek silpnesnės, o procedūros su trendais pačios silpniausios. Rezultatai ypač išsiskiria, vidutinio dydžio imtyse, kada testai be deterministinių komponentų yra ženkliai tikslesni už testus su trendais. Testai su laisvuju nariu yra mažiau tikslūs

už testus be deterministinių komponentų, bet tikslesni už testus su trendais.

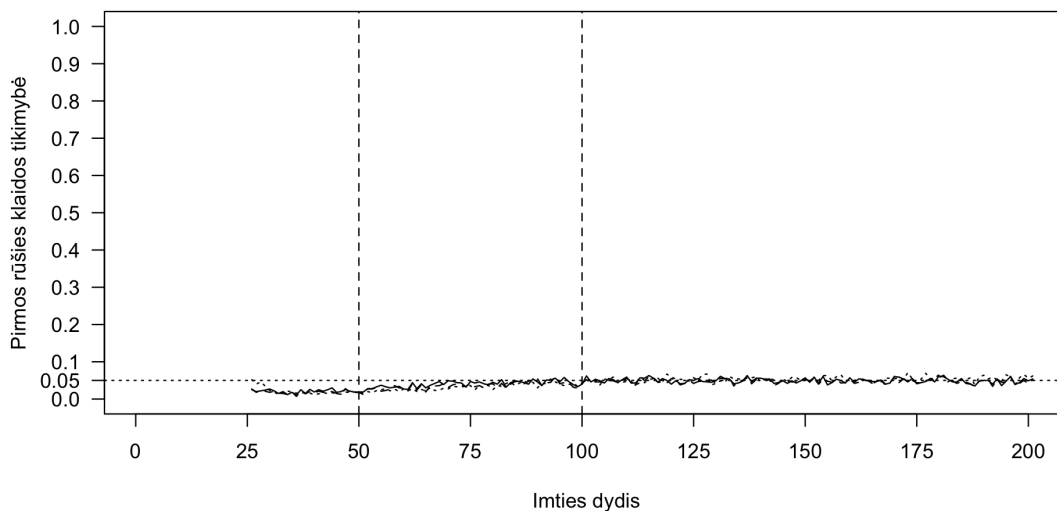


85 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

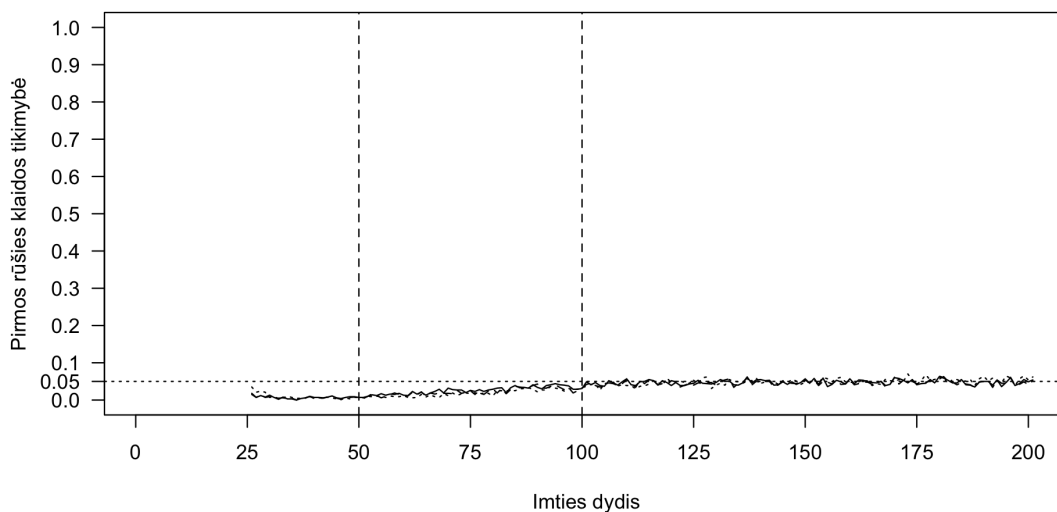


86 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 87 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 88 - naudojant maksimalios tikrinės reikšmės statistiką. Visų specifikacijų procedūros pateikia beveik identiškus rezultatus - klaidos tikimybė yra apie 5%, tačiau mažose imtyse ši tikimybė yra šiek tiek mažesnė.

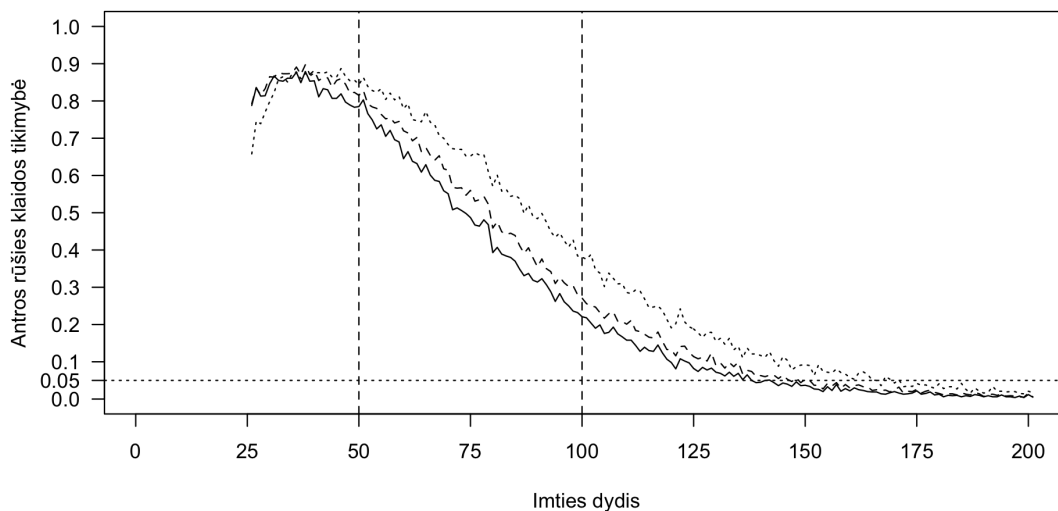


87 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

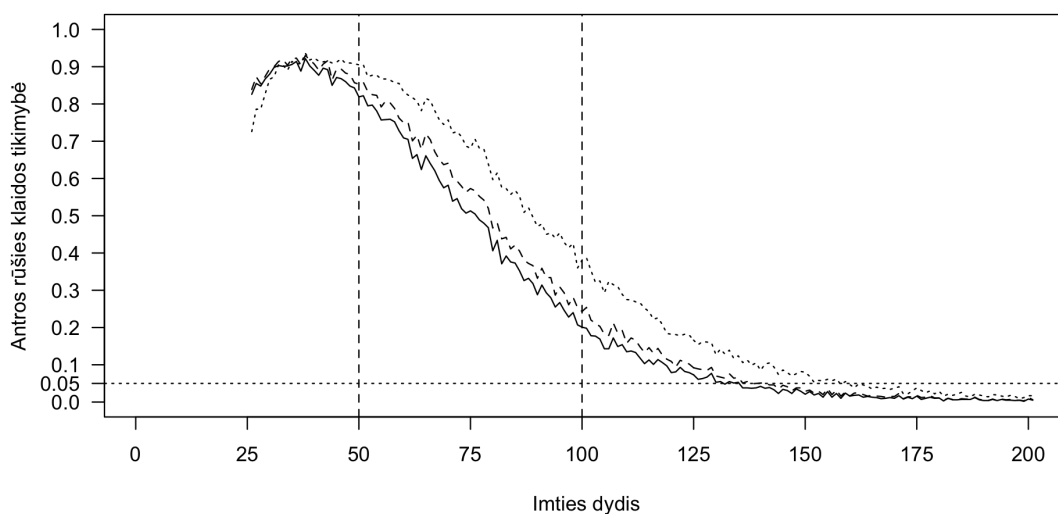


88 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 89 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 90 - naudojant maksimalios tikrinės reikšmės statistiką. Ypač mažose imtyse (< 30) tiksliausias yra testas su trendu, tuo kitais atvejais tiksliausias yra testas be deterministinių komponentų, o testas su laisvu nariu yra tikslesnis už testą su trendu, bet mažiau tikslus už testą be deterministinių komponentų.



89 pav.: Antros rūšies klaida naudojant pėdsako statistiką



90 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

3.4. Modelis 3

Modelis 3 yra trimatis VECM modelis su konstanta ir trendu. Bus tiriamos trys šio modelio specifikacijos: su neintegruotais kintamaisiais, su vienu kointegruojančiu vektoriumi ir su dviem kointegruojančiais vektoriais. Lygtis 110 aprašo modelio specifikaciją su skirtingais stochastiniais trendais, lygtis 111 aprašo modelio specifikaciją su vienu kointegruojančiu vektoriumi, o lygtis 112 aprašo modelio specifikaciją su dviem kointegruojančiais vektoriais.

$$\begin{aligned}
x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + d_x t + \varepsilon_{xt} \\
y_t &= \alpha_y m_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + d_y t + \varepsilon_{yt} \\
z_t &= \alpha_z n_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + d_z t + \varepsilon_{zt}
\end{aligned} \tag{110}$$

kur l_t , m_t ir n_t yra atsitiktinai sugeneruoti vienetinės šaknies procesai.

$$\begin{aligned}
x_t &= \alpha_x l1_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + d_x t + \varepsilon_{xt} \\
y_t &= \alpha_y l2_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + d_y t + \varepsilon_{yt} \\
z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + d_z t + \varepsilon_{zt}
\end{aligned} \tag{111}$$

kur $l1_t$ ir $l2_t$ yra atsitiktinai sugeneruoti vienetinės šaknies procesai, o $l_t = l1_t + l2_t$.

$$\begin{aligned}
x_t &= \alpha_x l_t + \sum_{i=1}^4 a_i x_{t-i} + e_x + d_x t + \varepsilon_{xt} \\
y_t &= \alpha_y l_t + \sum_{i=1}^4 b_i y_{t-i} + e_y + d_y t + \varepsilon_{yt} \\
z_t &= \alpha_z l_t + \sum_{i=1}^4 c_i z_{t-i} + e_z + d_z t + \varepsilon_{zt}
\end{aligned} \tag{112}$$

kur l_t yra atsitiktinai sugeneruotas vienetinės šaknies procesas.

Modelis 6 turi tokią VECM reprezentaciją:

$$\begin{pmatrix} \Delta x_t \\ \Delta y_t \\ \Delta z_t \end{pmatrix} = \Pi \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} + \Gamma_1 \begin{pmatrix} \Delta x_{t-1} \\ \Delta y_{t-1} \\ \Delta z_{t-1} \end{pmatrix} + \Gamma_2 \begin{pmatrix} \Delta x_{t-2} \\ \Delta y_{t-2} \\ \Delta z_{t-2} \end{pmatrix} + \tag{113}$$

$$+ \Gamma_3 \begin{pmatrix} \Delta x_{t-3} \\ \Delta y_{t-3} \\ \Delta z_{t-3} \end{pmatrix} + \Gamma_4 \begin{pmatrix} \Delta x_{t-4} \\ \Delta y_{t-4} \\ \Delta z_{t-4} \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} + \begin{pmatrix} \varepsilon_{xt} \\ \varepsilon_{yt} \\ \varepsilon_{zt} \end{pmatrix} \tag{114}$$

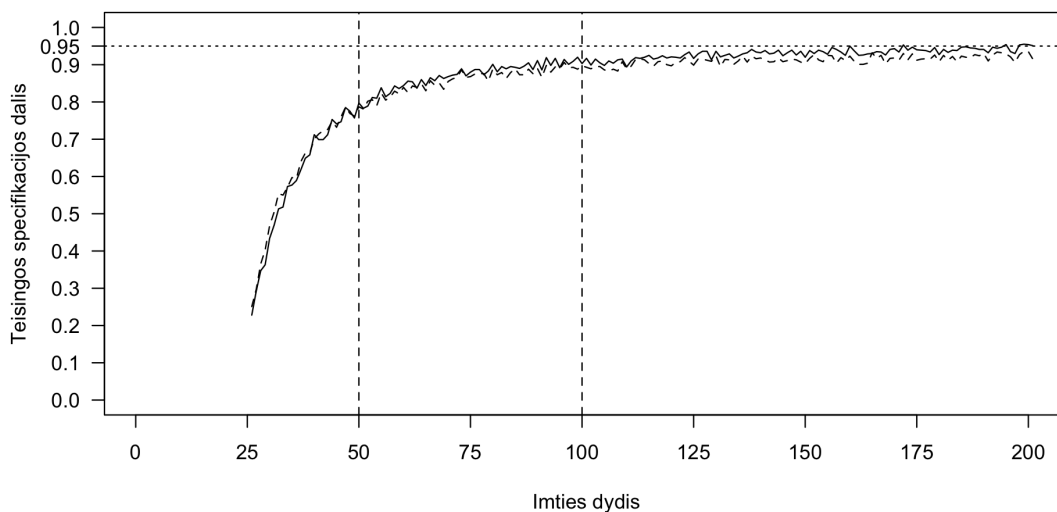
kur

$$\Pi = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{e1} & \alpha_{d1} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{e2} & \alpha_{d2} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{e3} & \alpha_{d3} \end{pmatrix} \begin{pmatrix} \beta_{11} & \beta_{21} & \beta_{31} \\ \beta_{12} & \beta_{22} & \beta_{32} \\ \beta_{13} & \beta_{23} & \beta_{33} \\ e_x & e_y & e_z \\ d_x & d_y & d_z \end{pmatrix} \quad (115)$$

3.4.1. Johansen procedūra be deterministinių komponentų

Johansen procedūra be deterministinių komponentų nėra tinkama šiam modeliui, nes kointegracijos komponente yra laisvasis narys ir trendas.

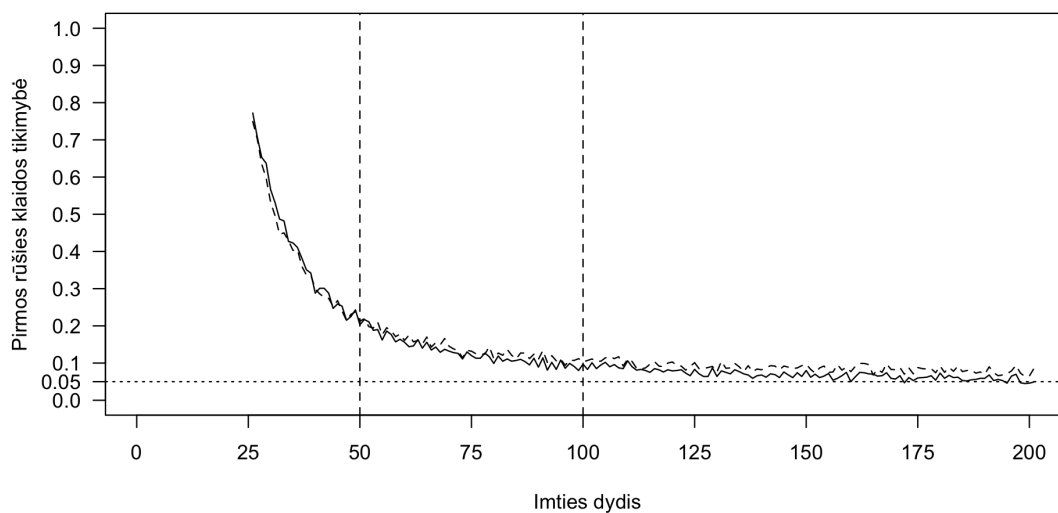
Nekointegruoti kintamieji. Pirmiausia Johansen procedūra be deterministinių komponentų bus atlikta nekointegruotai modelio specifikacijai. Paveikslėlis 91 parodo kokia dalis testų nustato kointegruojančių vektorių skaičių teisingai. Šiuo atveju tiek pėdsako, tiek maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, tačiau maksimalios tikrinės reikšmės procedūra yra šiek tiek mažiau tiksli vidutinėse ir didelėse imtyse. Mažose imtyse testo galia labai silpna tarp 20% ir 80%, o vidutinėse ir didelėse imtyse stipri - tarp 80% ir 90%.



91 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

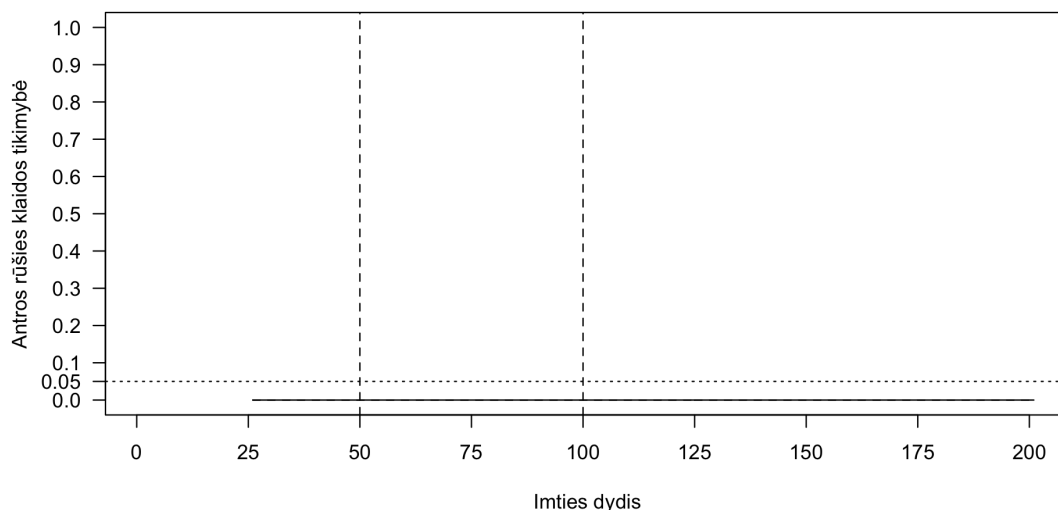
Paveikslėlis 92 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia beveik identiškus rezultatus, tačiau vidutinėse ir didelėse imtyse maksimalios tikrinės reikšmės statistika pateikia prastesnius rezultatus. Mažose

imtyse klaidos tikimybė yra labai didelė, tačiau sparčiai mažėjanti didėjant imties dydžiui.



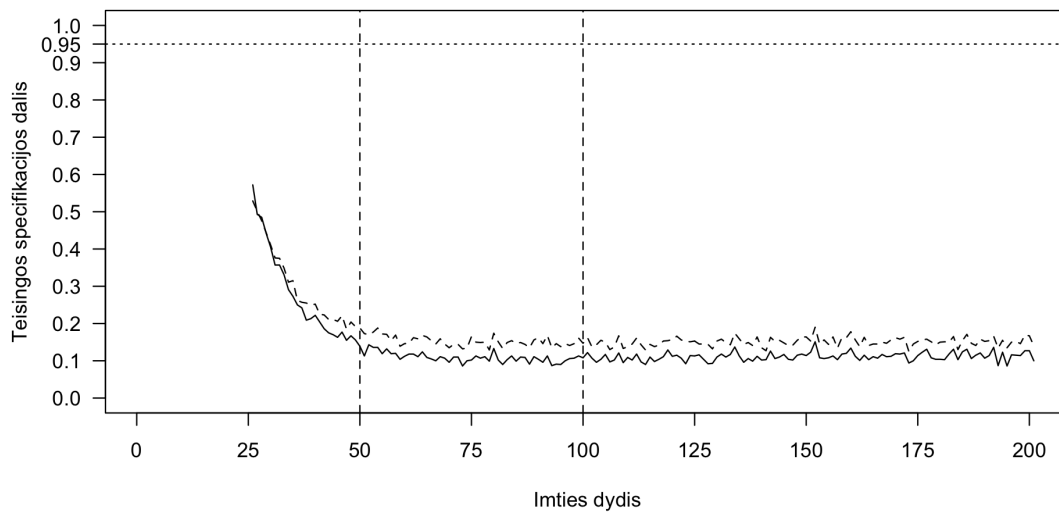
92 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 93 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų. Kadangi kointegruojančių vektorių nėra, tai ir antros rūšies klaidos padaryti neįmanoma, todėl tikimybė yra 0.



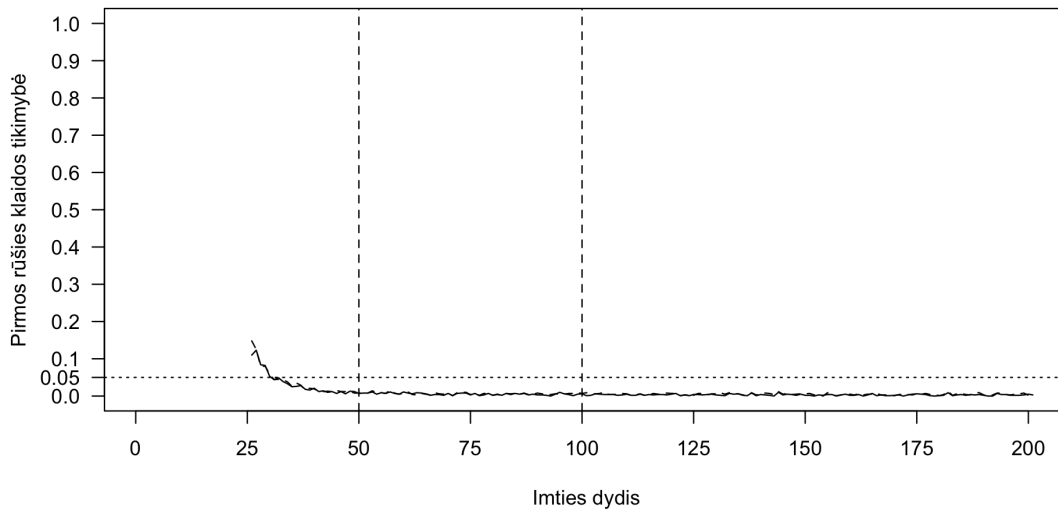
93 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Vienas kointegruojantis vektorius. Šiuo atveju visos trys modelio lygtys turi tą patį stochastinį trendą. Paveikslėlis 94 parodo, kiek procentų atvejų testai nustatė kointegruojančių vektorių skaičių teisingai. Šiuo atveju Johansen procedūra be deterministinių komponentų visiškai netinka tiriamam modeliui, nes ji teisingai specifikuoja modelį tik nuo 10% iki 60% atvejų, o svarbiausia didėjant imties dydžiui testo rezultatai tik prastėja. Maksimalios tikrinės reikšmės testas šiuo atveju pateikia šiek tiek geresnius rezultatus, nei pėdsako testas.



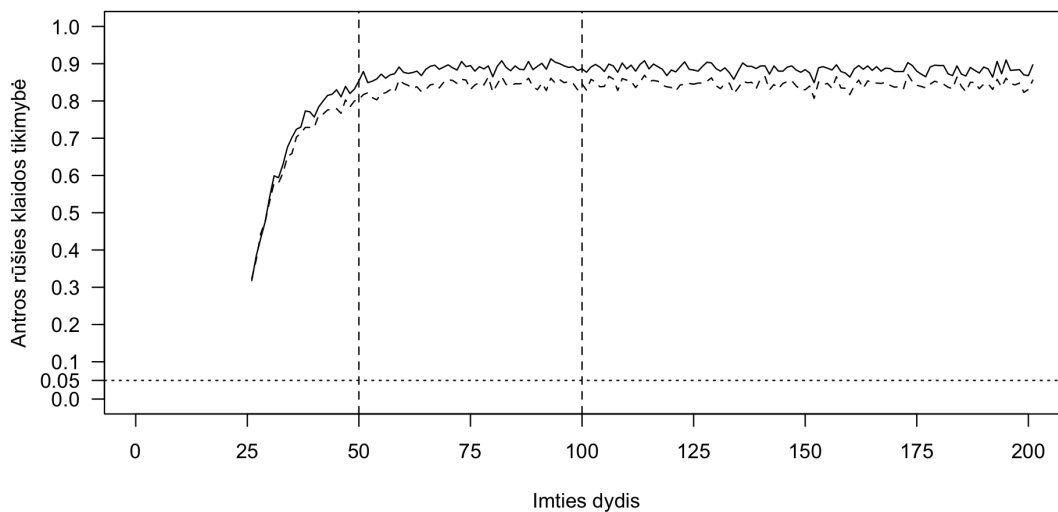
94 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 94 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Tiek pėdsako tiek maksimalios tikrinės reikšmės testai beveik niekada nepadaro šios klaidos - beveik visuose imties dydžiuose šios klaidos tikimybė yra arti 0.



95 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

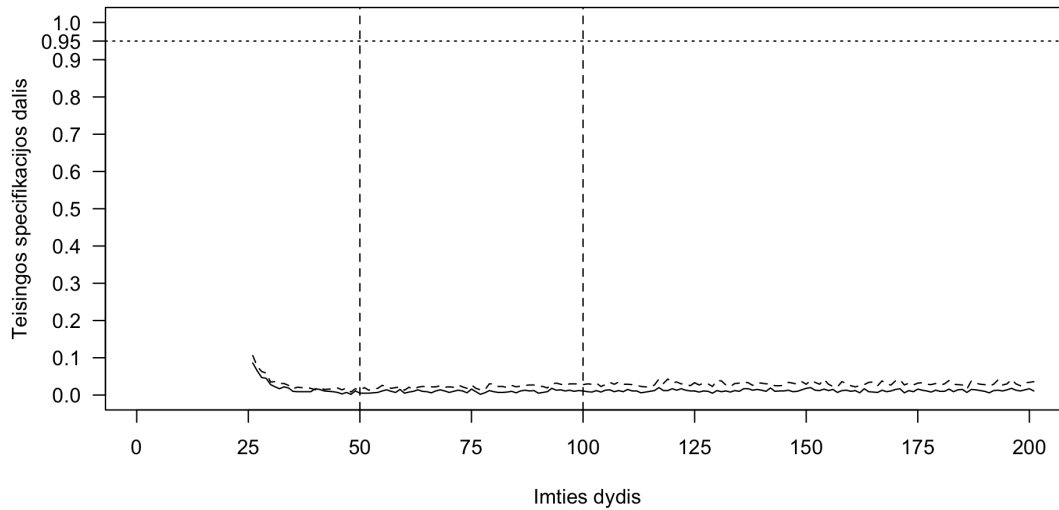
Paveikslėlis 96 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač didelė ir padaro šio tipo procedūrą visiškai netinkama naudoti, kai kintamieji yra kointegruoti ir turi deterministinį trendą ir laisvąjį narį.



96 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

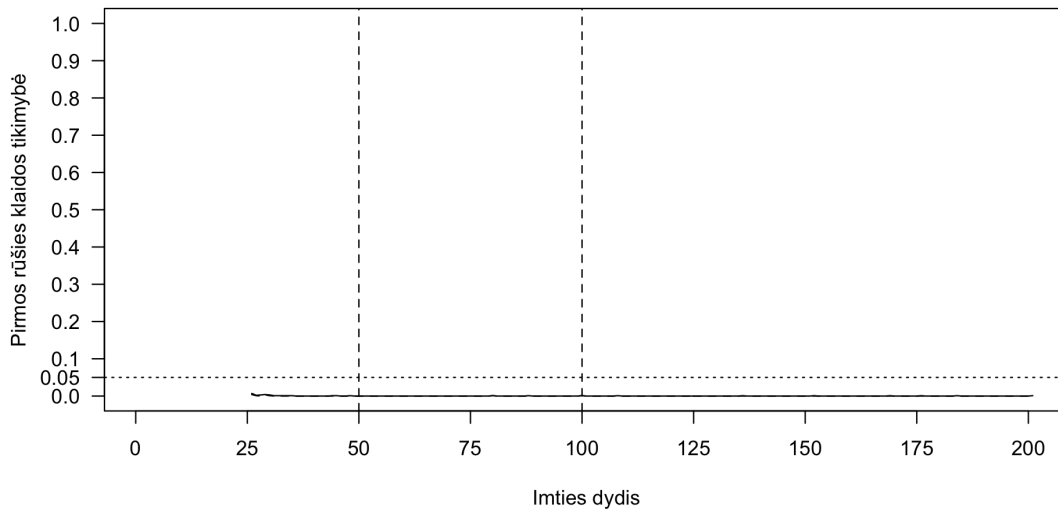
Du kointegruojantys vektoriai. Paveikslėlis 97 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai, tuo atveju, kai modelis turi

du kointegruojančius vektorius. Kaip ir specifikacijos su vienu kointegruojančiu vektoriumi atveju ši procedūra nėra tinkama. Tikimybė teisingai nustatyti kointegruojančių vektorių skaičių yra artima 0 nepriklausomai nuo imties dydžio.



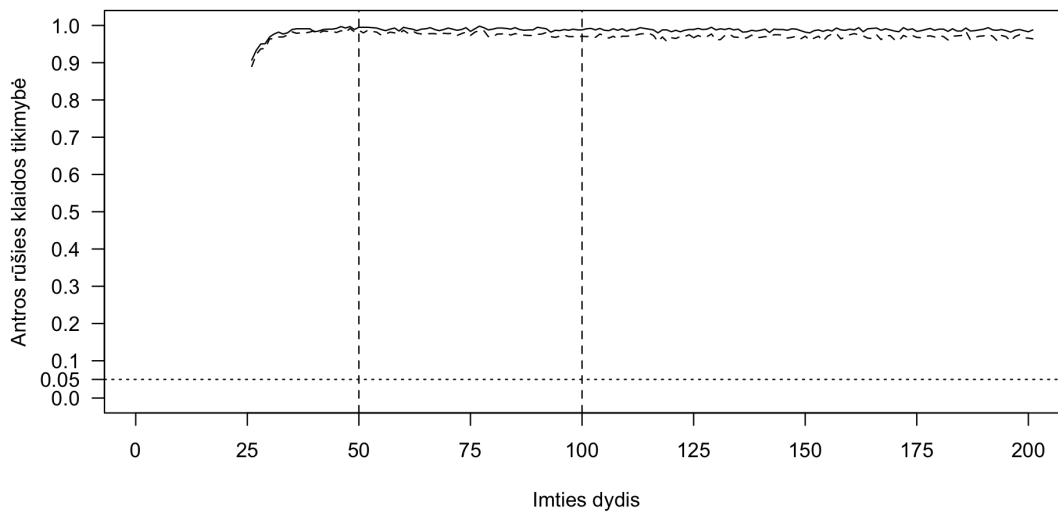
97 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 97 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji turi du kointegruojančius vektorius. Tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos atveju klaidos tikimybė yra arti 0.



98 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 99 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą be deterministinių komponentų, kai kintamieji yra kointegruoti. Šios klaidos tikimybė yra ypač didelė ir yra arti 100% nepriklausomai nuo imties dydžio.

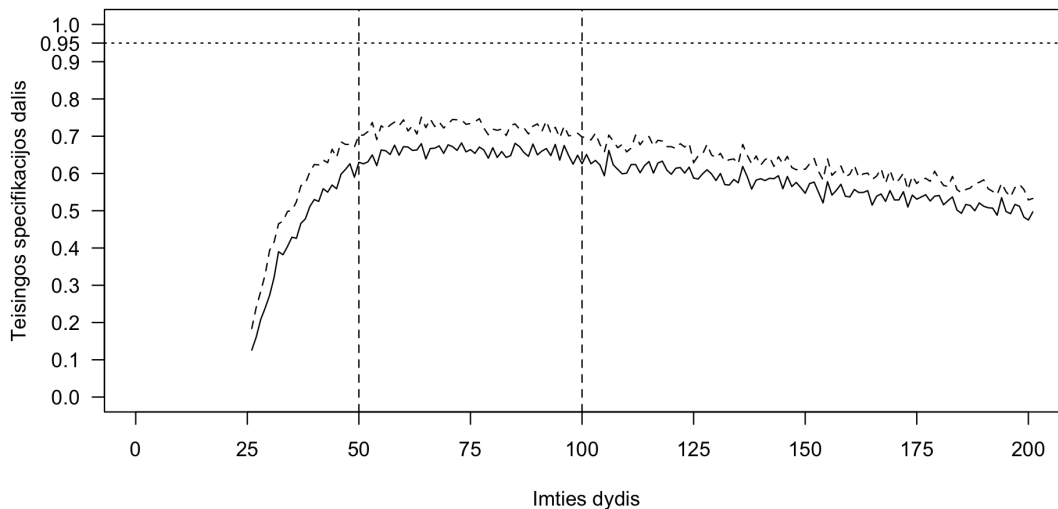


99 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.4.2. Johansen procedūra su konstanta

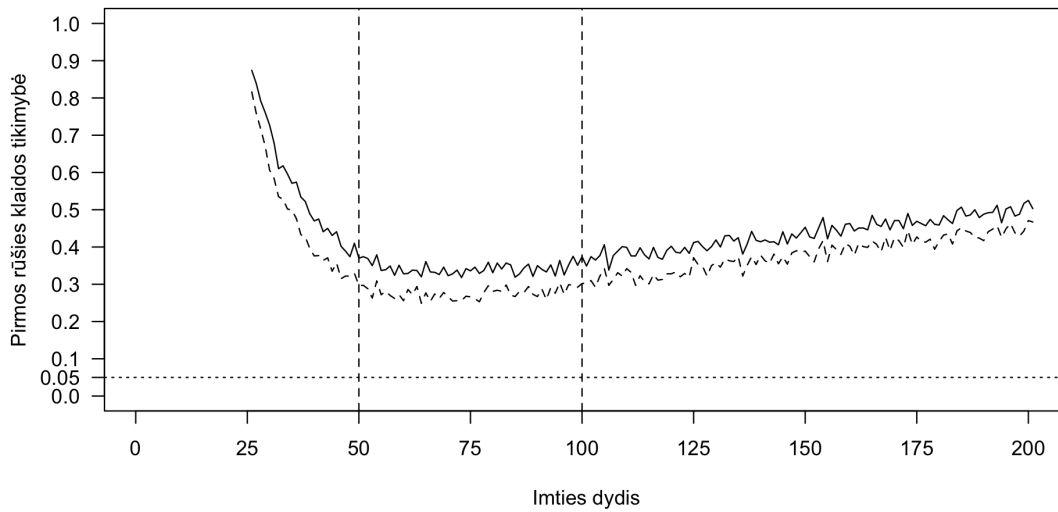
Johansen procedūra nėra tinkama šiam modeliui, nes modelyje yra deterministinis trendas.

Nekointegruoti kintamieji. Paveikslėlis 100 parodo kokia dalis Johansen procedūros su konstanta testų nustato kointegruojančių vektorių skaičių teisingai. Rezultatai parodo, kad šio tipo procedūros taip pat negalima naudoti tiriant modelius su deterministiniu trendu ir laisvuju nariu. Tiek pėdsako tiek maksimalios tikrinės reikšmės testai nustato kointegruojančių vektorių skaičių teisingai tarp 10% ir 75% atveju, tačiau didžiausia problema yra tai, kad testų patikimumas mažėja didėjant imties dydžiui. Maksimalios tikrinės reikšmės statistika taip pat yra daug pranašesnė už pėdsako statistiką visose imtyse.



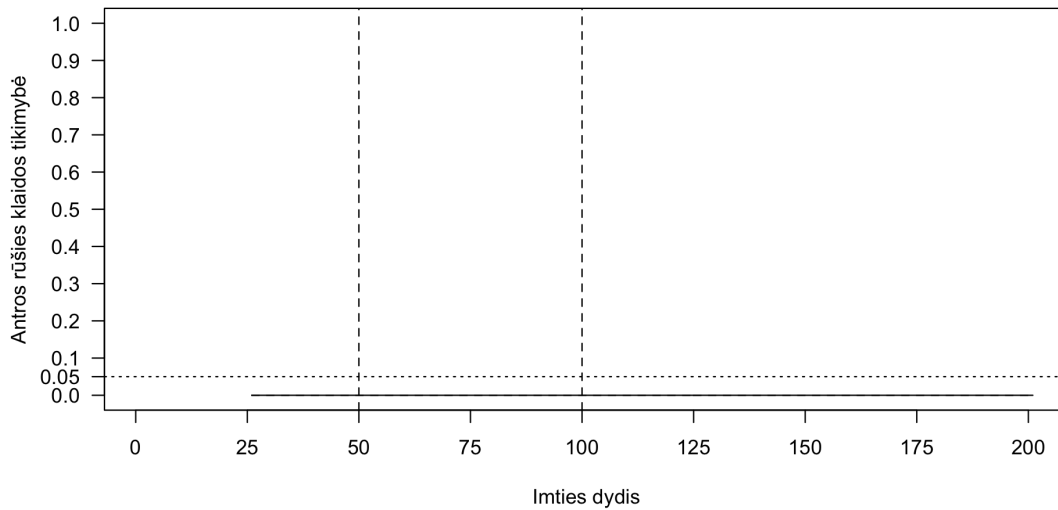
100 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 101 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta. Pėdsako ir maksimalios tikrinės reikšmės statistikos pateikia skirtingus rezultatus - maksimalios tikrinės reikšmės atveju šios klaidos tikimybė yra mažesnė visuose imties intervaluose. Mažose imtyse, didėjant imties dydžiui, šios klaidos tikimybė mažėja, tačiau vidutinėse ir dideliuose imtyse klaidos tikimybė didėja didėjant imties dydžiui.



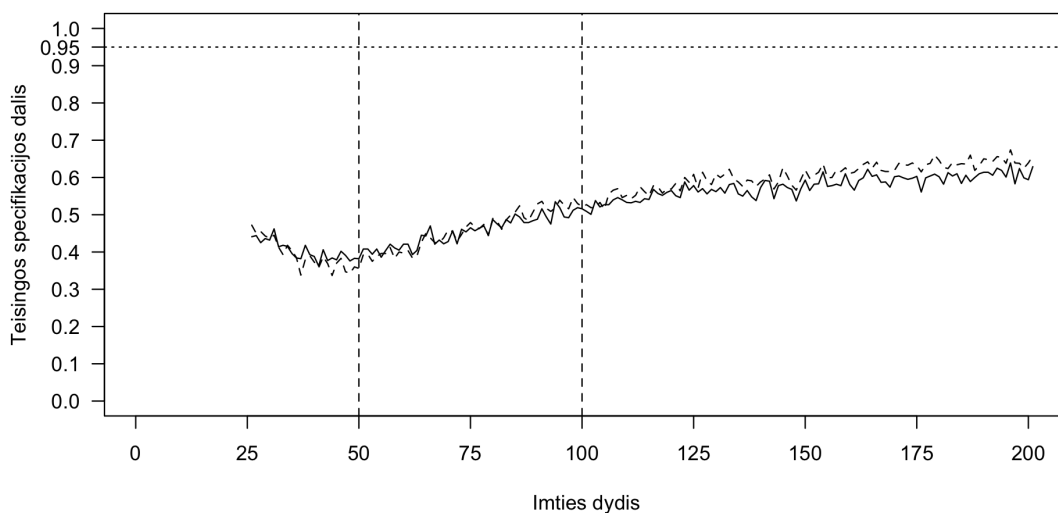
101 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 102 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su laisvuju nariu. Šios klaidos tikimybė yra 0.



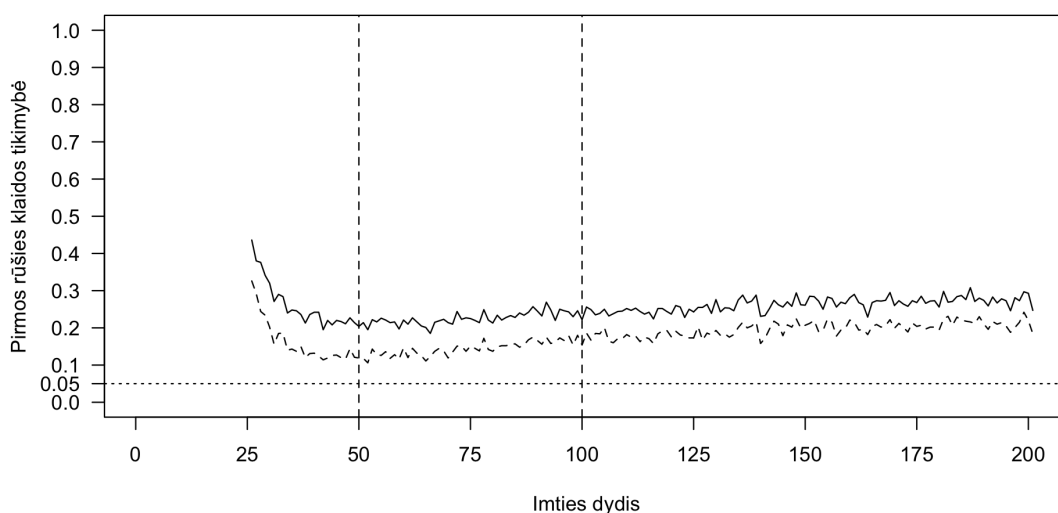
102 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Vienas kointegruojantis vektorius. Paveikslėlis 103 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Šis testas ir šiuo atveju pateikia nepatenkinamus rezultatus - teisingo specifikuojimo tikimybė mažose imtyse siekia tik 30% - 40% ir pakyla iki 65% didėjant imties dydžiui.



103 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

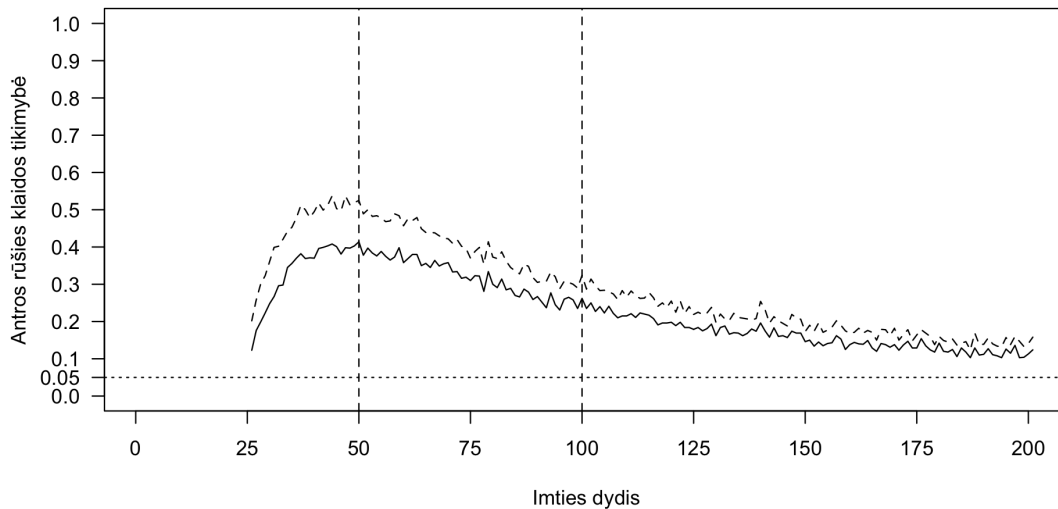
Paveikslėlis 104 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja tarp 15% ir 45% mažose imtyse, tačiau po truputį didėja didėjant imties dydžiui ir pasiekia 30%. Šiuo atveju maksimalios tikrinės reikšmės testas yra daug tikslesnis už pėdsako testą.



104 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

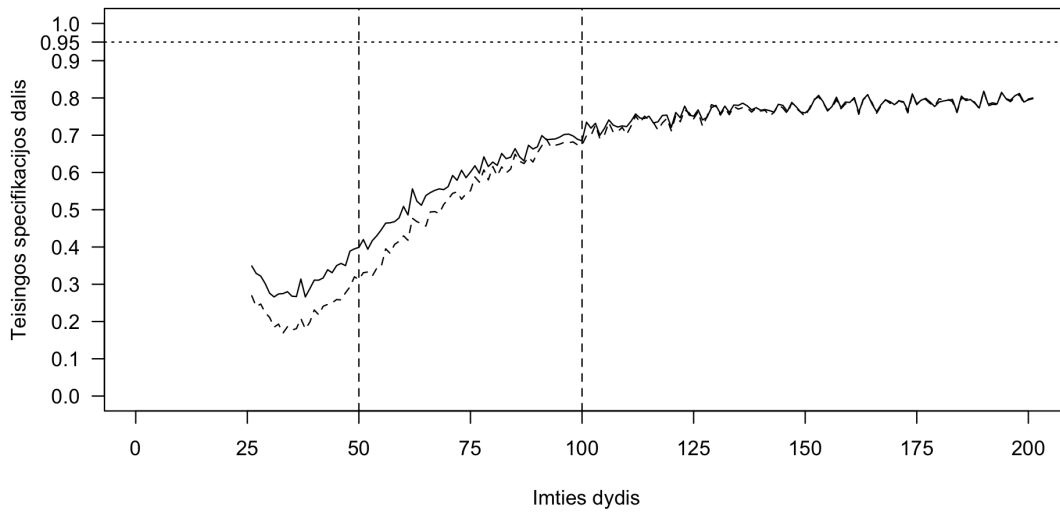
Paveikslėlis 105 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji turi vieną kointegruojantį vektorių. Šios klaidos

tikimybė yra vidutinio dydžio mažose ir vidutinėse imtyse. Vidutinėse imtyse, didėjant imties dydžiui, šios klaidos tikimybė palaipsniui mažėja, o didelėse imtyse priartėja prie 15%. Kai imties dydis yra tarp 25 ir 40, didėjant imties dydžiui, šios klaidos tikimybė sparčiai auga, po to pradeda palaipsniui mažėti. Visuose imties intervaluose pėdsako testas yra pranašesnis už maksimalios tikrinės reikšmės testą.



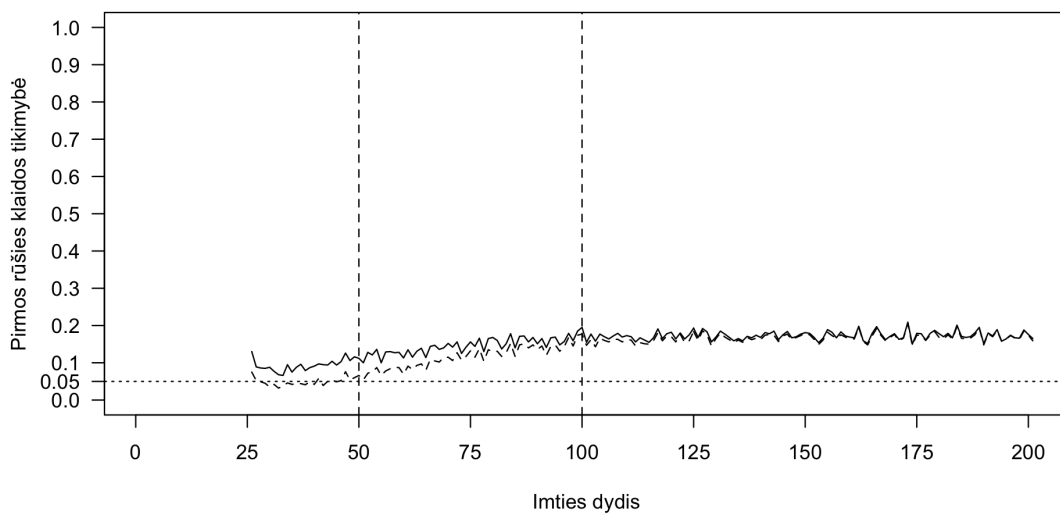
105 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Kai modelis turi du kointegruojančius vektorius, testų rezultatai yra panašūs į Johansen procedūros su vienu kointegruojančiu vektoriumi atvejį. Paveikslėlis 106 parodo, kiek procentų atvejų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas yra labai silpnas mažose imtyse tarp 15% ir 40%. Vidutinėse imtyse testas reikšmingai stiprėja iki 70%, o didelėse imtyse pasiekia 80%. Mažose ir vidutinėse imtyse pėdsako testas yra tikslesnis už maksimalios tikrinės reikšmės testą.



106 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

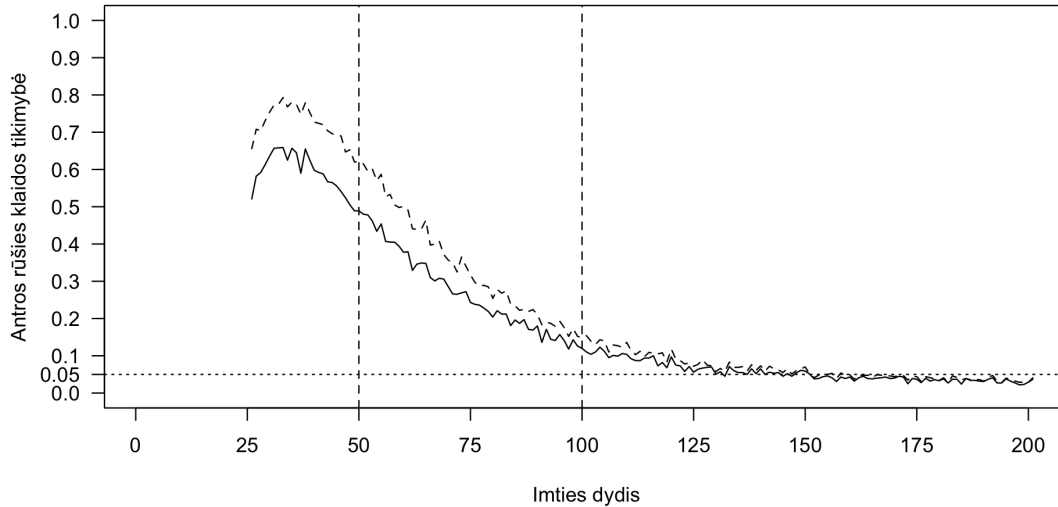
Paveikslėlis 107 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja tarp 5% ir 25%. Mažose imtyse maksimalios tikrinės reikšmės testas šią klaidą padaro rečiau nei pėdsako testas.



107 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 108 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti ir turi du kointegruojančius vektorius. Šios klaidos tikimybė yra ypač didelė mažose imtyse, kur pasiekia net

80%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 5%. Mažose ir vidutinėse imtyse pėdsako testai yra ženkliai pranašesni už maksimalios tikrinės reikšmės testus.

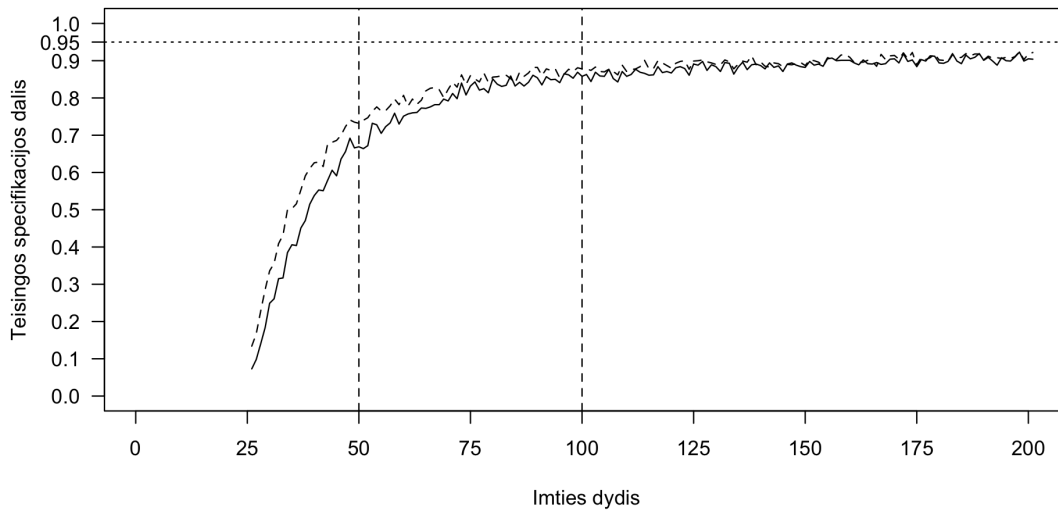


108 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

3.4.3. Johansen procedūra su trendu

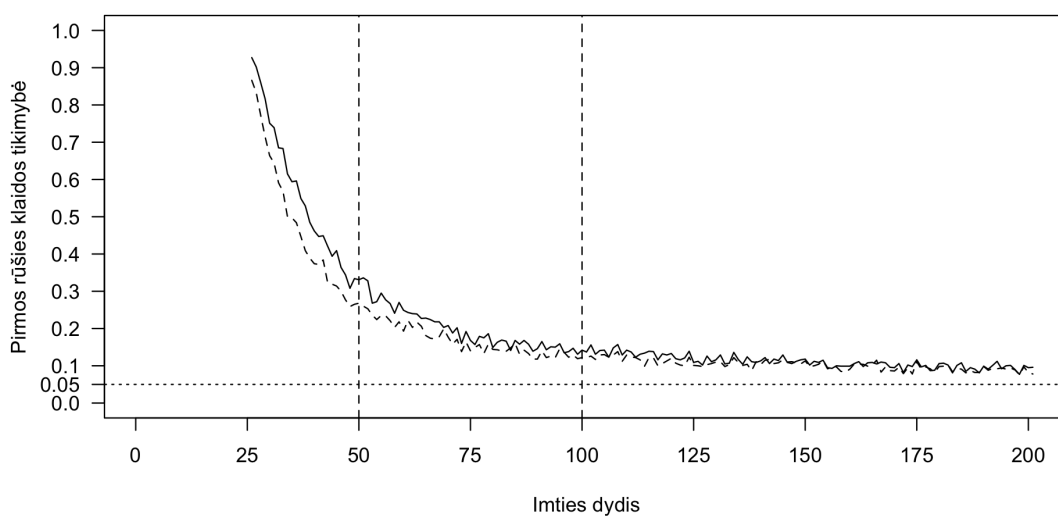
Johansen procedūra su trendu yra pati tinkamiausia šiam modeliui, nors modelyje yra ne tik trendas bet ir laisvasis narys, tačiau deterministinio trendo komponentą specifikuoti yra svarbiau.

Nekointegruoti kintamieji. Paveikslėlis 109 parodo kokia dalis Johansen procedūros su trendu testų nustato kointegruojančių vektorių skaičių teisingai. Testas yra ypač silpnas mažose imtyse - jo galia yra intervale tarp 5% ir 70%. Vidutinėse imtyse testas šiek tiek stiprėja nuo 70% iki 85%, o didelėse imtyse pasiekia 90%. Maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių, kai imties dydis yra nuo 25 iki 125.



109 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

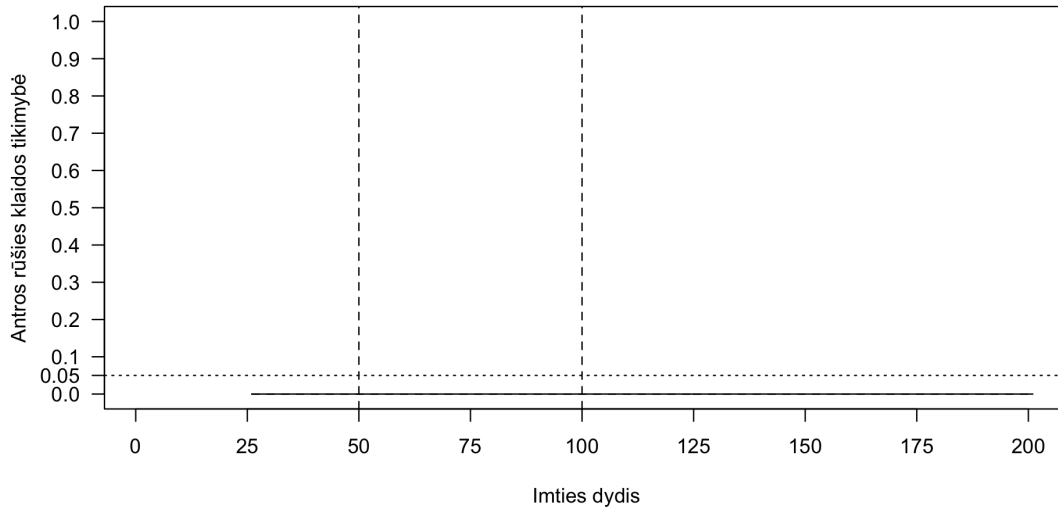
Paveikslėlis 110 parodo pirmos rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu. Klaidos tikimybė mažose imtyse labai didelė, siekianti 95%. Didėjant imties dydžiui klaidos tikimybė mažėja ir didelėse imtyse pasiekia 10%. Mažose ir vidutinėse imtyse maksimalios tikrinės reikšmės testas yra tikslesnis už pėdsako testą.



110 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

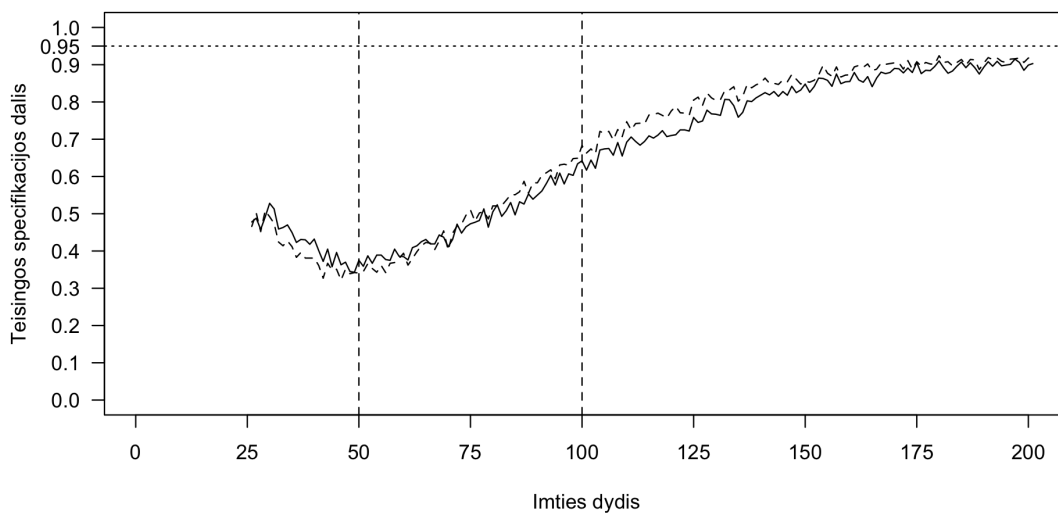
Paveikslėlis 110 parodo antros rūšies klaidos tikimybę naudojant Johansen pro-

cedūrą su trendu. Ši tikimybė yra 0.



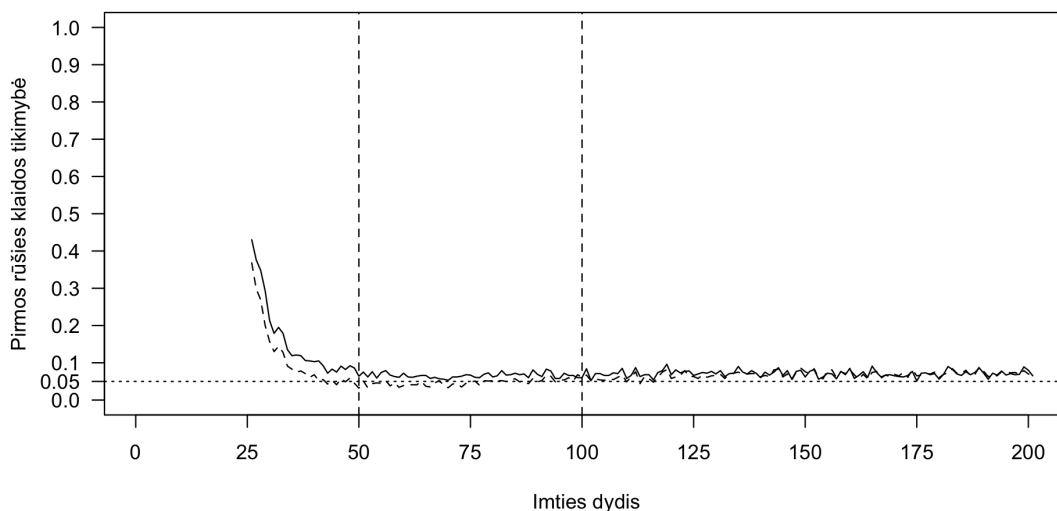
111 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Vienas kointegruojantis vektorius. Paveikslėlis 112 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai. Testas galia mažose imtyse yra vidutinė ir svyruoja tarp 30% ir 50%, vidutinėse imtyse - tarp 30% ir 60%, o didelėse imtyse tarp 60% ir 95%. Pėdsako testas yra tikslesnis mažose imtyse, o maksimalios tikrinės reikšmės testas yra tikslesnis didelėse imtyse.



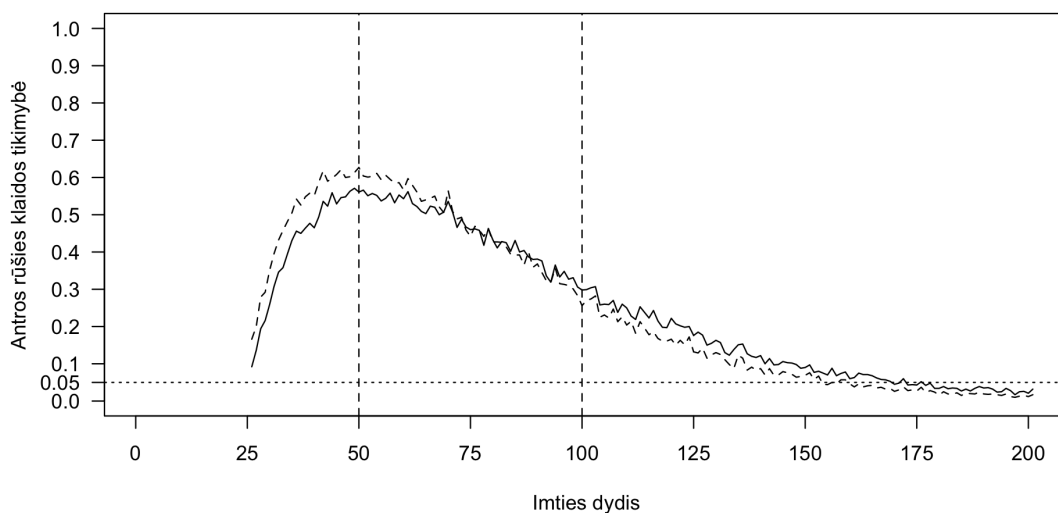
112 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 113 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5%, kai imtys yra vidutinės arba didelės. Mažose imtyse klaidos tikimybė siekia 50% tačiau sparčiai krenta didėjant imties dydžiui. Maksimalios tikrinės reikšmės testas yra pranašesnis mažose ir vidutinėse imtyse.



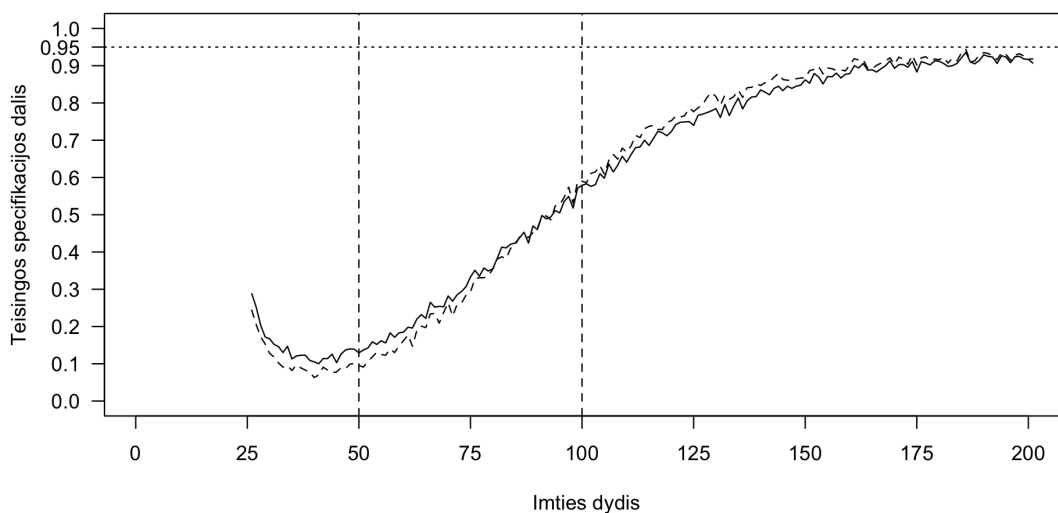
113 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 114 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su konstanta, kai kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Šios klaidos tikimybė yra didelė mažose imtyse, ir siekia iki 65%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Mažose imtyse didėjant imties dydžiui klaidos tikimybė didėja, o vidutinėse ir didelėse imtyse mažėja.



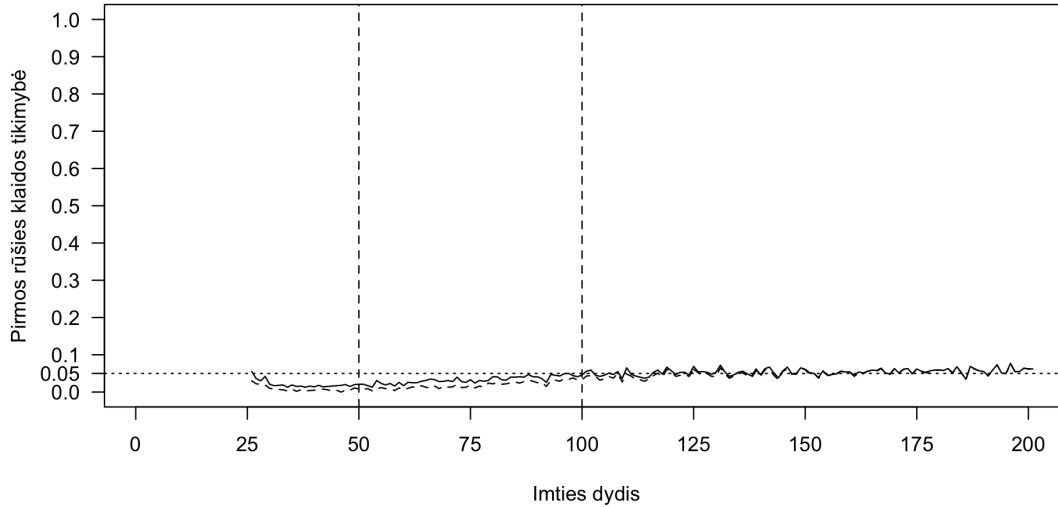
114 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Du kointegruojantys vektoriai. Paveikslėlis 115 parodo, kiek procentų testai nustatė kointegruojančių vektorių skaičių teisingai, kai kintamieji turi du kointegruojančius vektorius. Testas yra labai silpnas mažose imtyse tarp 5% ir 30%. Vidutinėse imtyse testas reikšmingai stiprėja iki 60%, o didelėse imtyse pasiekia 95%. Pėdsako testas yra pranašesnis mažose imtyse, o maksimalios tikrinės reikšmės šiek tiek tiksliau nustato kointegruojančių vektorių skaičių didelėse imtyse.



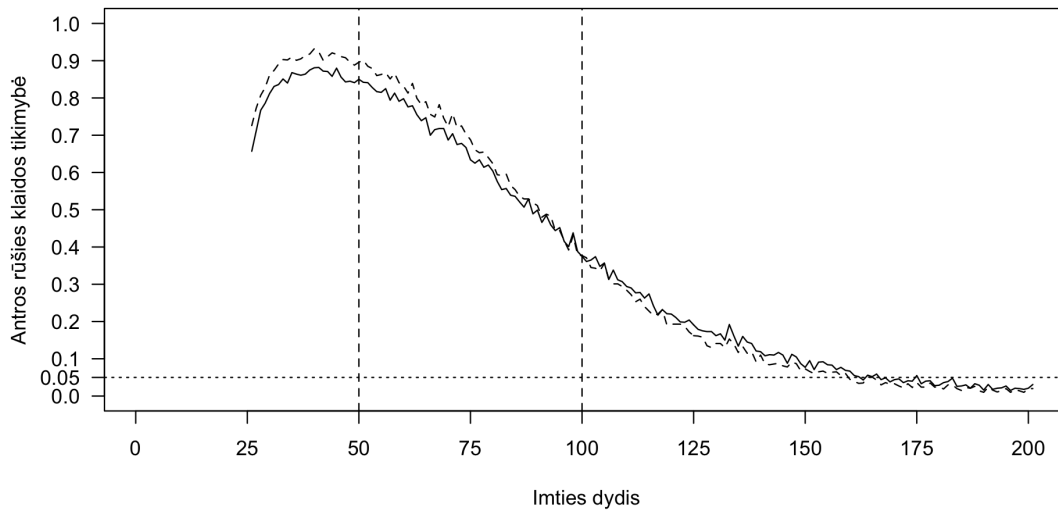
115 pav.: Teisingai nustatomas vektorių skaičius palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 116 parodo, pirmos rūšies klaidos tikimybę. Ji svyruoja apie 5%, kai imtys yra vidutinės ir didelės. Mažose imtyse šios klaidos tikimybės intervalas yra tarp 0% ir 5%.



116 pav.: Pirmos rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

Paveikslėlis 117 parodo antros rūšies klaidos tikimybę naudojant Johansen procedūrą su trendu, kai kintamieji yra kointegruoti ir turi du kointegruojančius vektorius. Šios klaidos tikimybė yra ypač didelė mažose imtyse, net iki 95%. Vidutinėse imtyse šios klaidos tikimybė sparčiai mažėja, o didelėse imtyse priartėja prie 0. Maksimalios tikrinės reikšmės testas ir šiuo atveju yra mažiau tikslus už pėdsako testą, išskyrus dideles imtis.

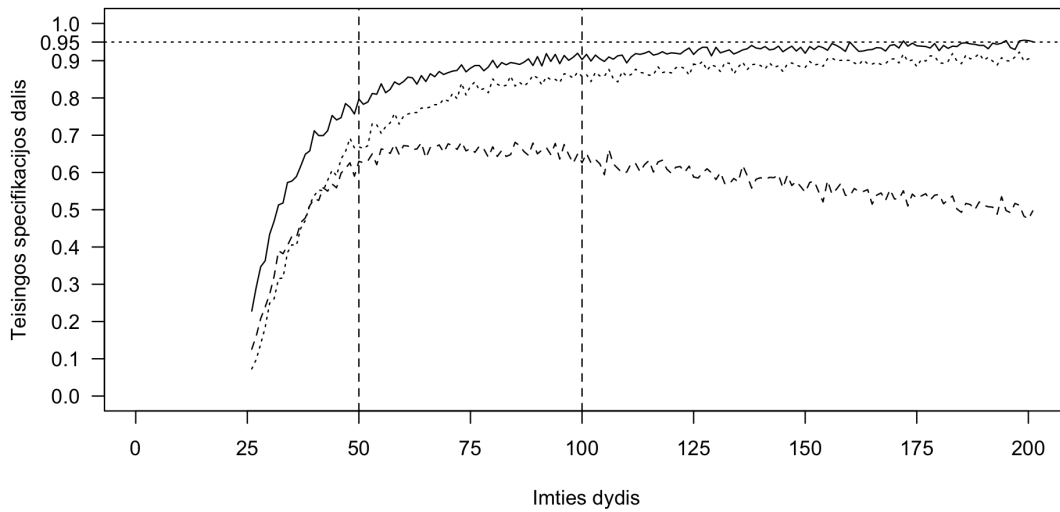


117 pav.: Antros rūšies klaida palyginant pėdsako (vientisa linija) ir maksimalios tikrinės reikšmės (brūkšniuota linija) testus

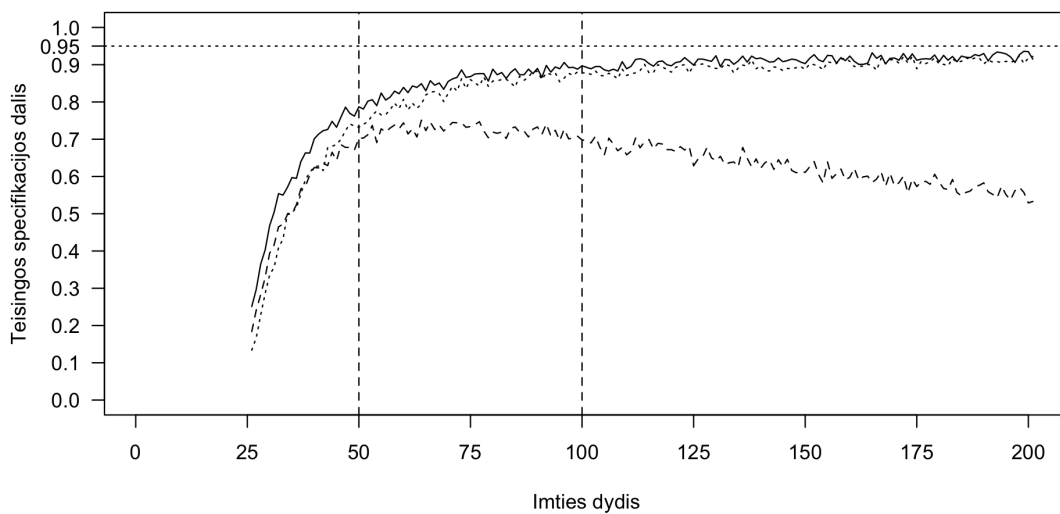
3.4.4. Misspecifikacijos įtaka rezultatams

Nors visos tirtos Johansen procedūros misspecifikuoja modelio deterministinius komponentus, misspecifikacijos įtaka labai skirtinga. Johansen procedūros be deterministinių komponentų visiškai netinka nustatant kointegruojančių vektorių skaičių šiame modelyje - šios procedūros antros rūšies klaidos tikimybė, kai modelyje nėra kointegruojančių vektorių siekia net 95% ne tik mažose, bet ir vidutinėse bei didelėse imtyse. Johansen procedūra su konstanta taip pat nėra tinkama - pirmos rūšies klaidos tikimybė, kai modelyje nėra kointegruojančių vektorių yra auganti didėjant imties dydžiui. Bendroju atveju vienintelė procedūra su trendu yra tinkama nustatant kointegruojančių vektorių skaičių tokiuose modeliuose visose trijose specifikacijose.

Nekointegruoti kintamieji. Kai imtys yra mažos visų tipų Johansen procedūros turi vidutinę tikimybę padaryti pirmos rūšies klaidą. Vidutinėse ir didelėse imtyse testų galia yra didelė. Paveikslėlis 118 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 119 - naudojant maksimalios tikrinės reikšmės statistiką. Testas be deterministinių komponentų yra tiksliausias, procedūra su trendu šiek tiek silpnesnė, tuo tarpu procedūra su konstanta yra visiškai netinkama, nes didėjant imties dydžiui jos tikslumas mažėja.

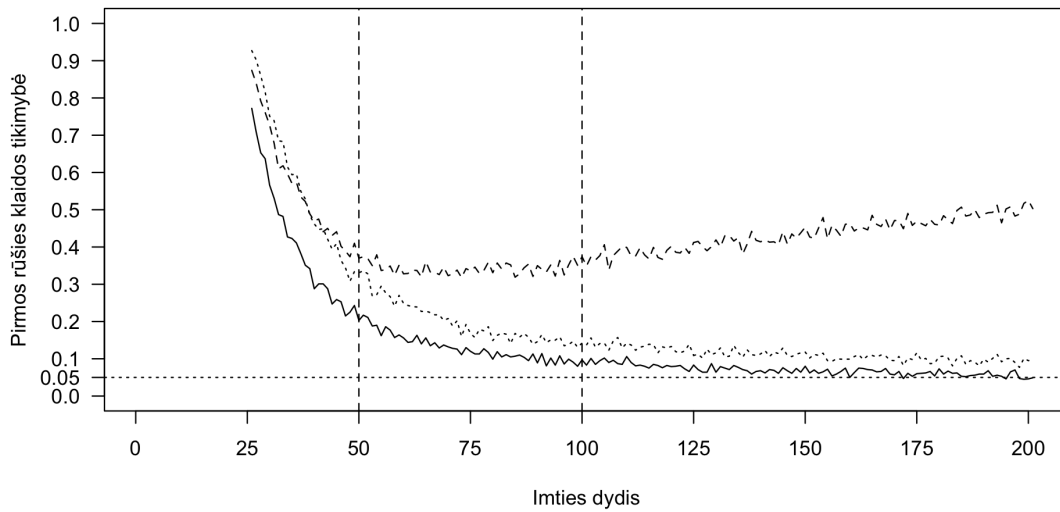


118 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

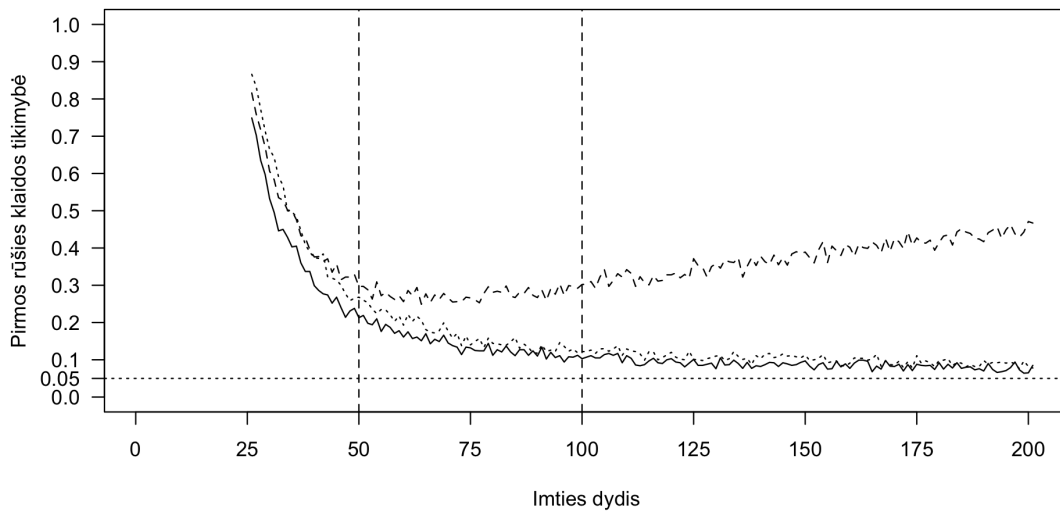


119 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 120 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 121 - naudojant maksimalios tikrinės reikšmės statistiką. Klaidos tikimybė mažiausia testo be deterministinių komponentų atveju, šiek tiek didesnė testo atveju su trendu, o testo su laisvuju nariu atveju didėjanti, kai didėja imties dydis.

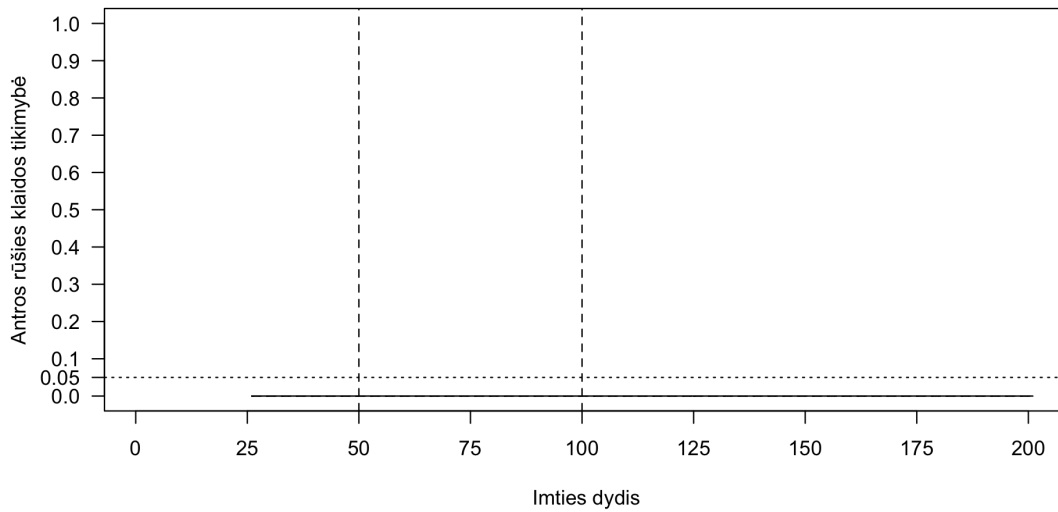


120 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

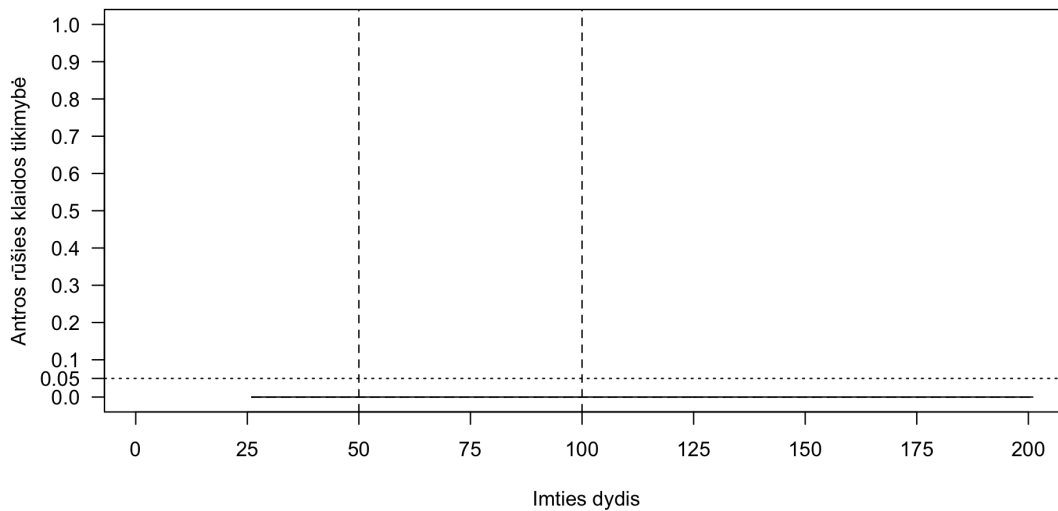


121 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 122 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 123 - naudojant maksimalios tikrinės reikšmės statistiką. Visais atvejais klaidos tikimybė yra 0.

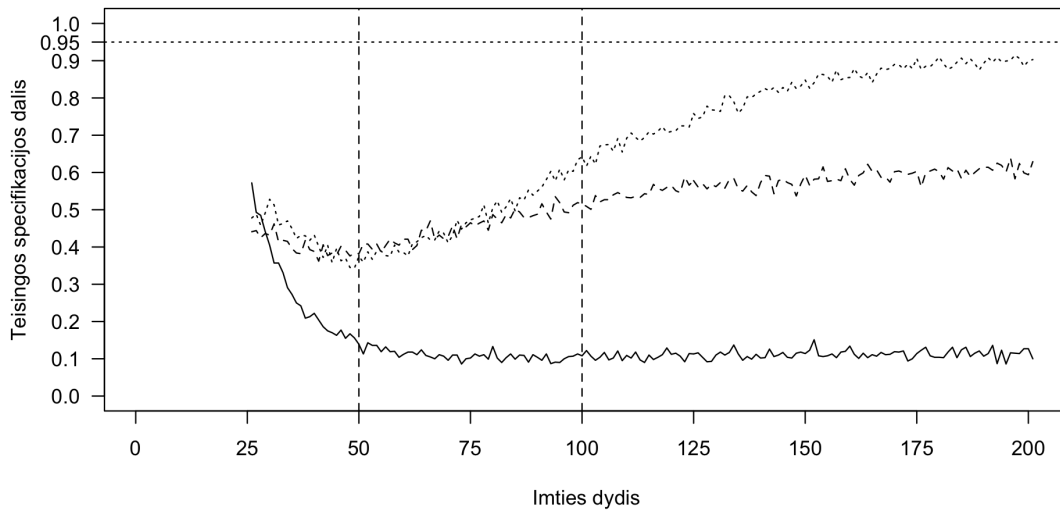


122 pav.: Antros rūšies klaida naudojant pėdsako statistiką

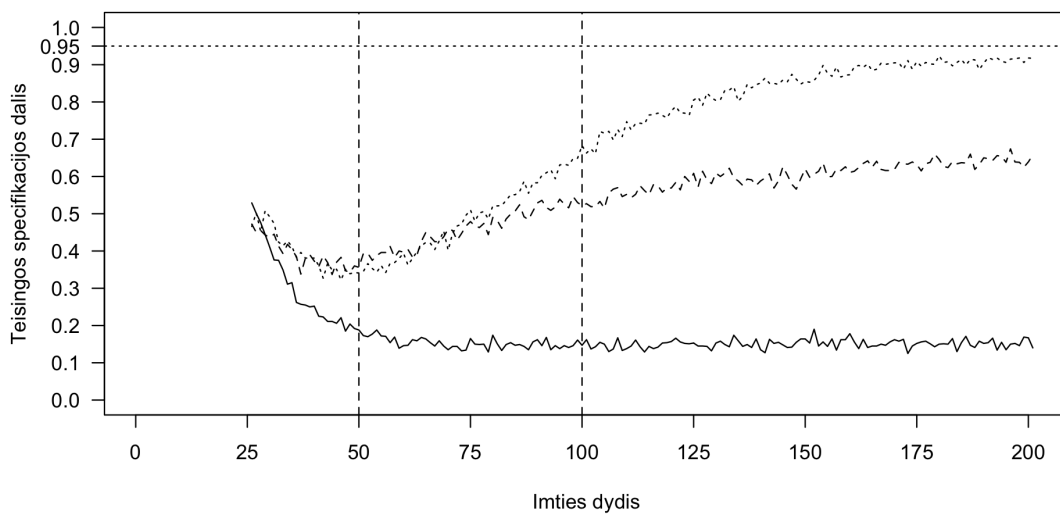


123 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Vienas kointegruojantis vektorius. Paveikslėlis 124 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 125 - naudojant maksimalios tikrinės reikšmės statistiką. Johansen procedūra be deterministinių komponentų yra visiškai netinkama šiam modeliui - tikimybė teisingai specifikuoti modelį artėja prie 10% imčiai didėjant. Johansen procedūra su konstanta mažose ir vidutinėse imtyse yra beveik tokia pati tiksli kaip ir testas su trendu, tačiau didelėse imtyse Johansen procedūra su trendu yra tikslesnė.

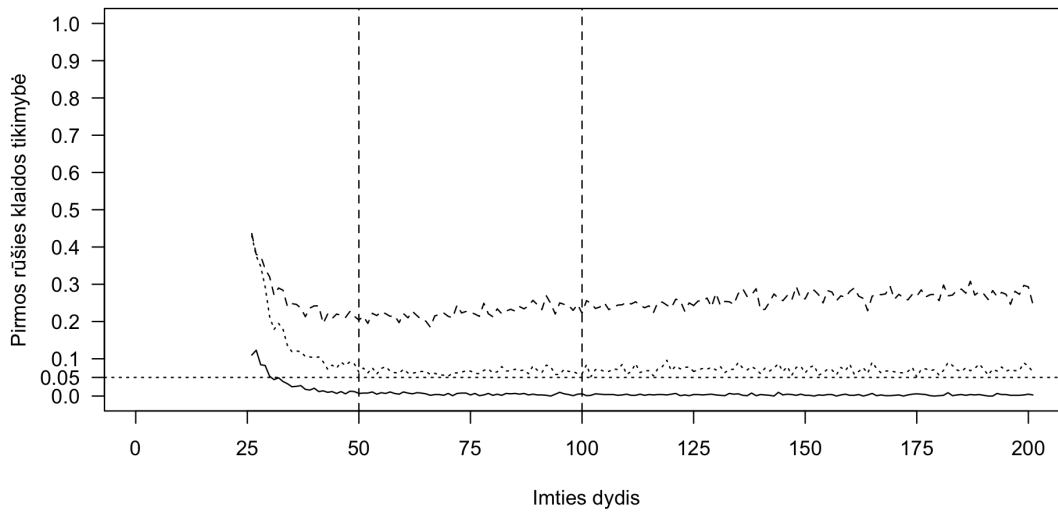


124 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

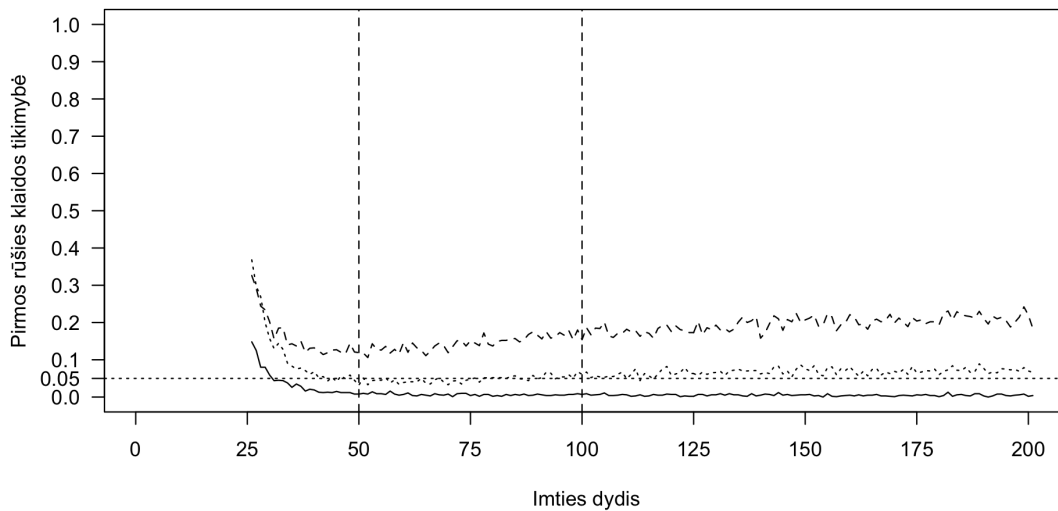


125 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 126 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 127 - naudojant maksimalios tikrinės reikšmės statistiką. Johansen procedūra be deterministinių komponentų pirmos rūšies klaidos tikimybė yra arti 0%, išskyrus labai mažas imtis, procedūros su laisvuju nariu tarp 25% ir 50%, o procedūros su trendu apie 10%, kai imtys yra vidutinės arba didelės ir šiek tiek didesnė mažose imtyse.

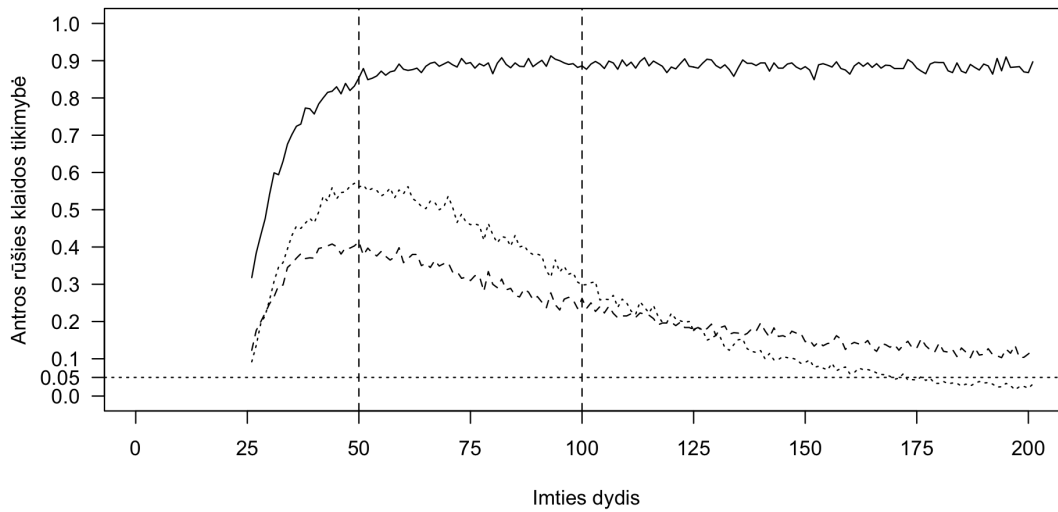


126 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

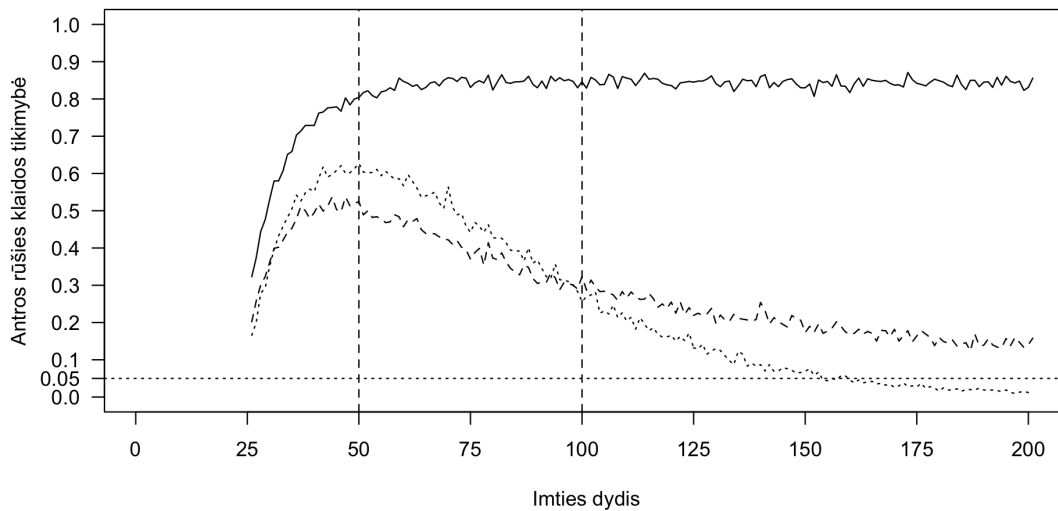


127 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 128 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 129 - naudojant maksimalios tikrinės reikšmės statistiką. Johansen testas be deterministinių komponentų šią klaidą padaro apie 90% atvejų, todėl jis nėra tinkamas. Johansen procedūra su konstanta mažiau linkusi padaryti šią klaidą mažose ir vidutinėse imtyse. Didelėse imtyse šios klaidos tikimybė yra mažesnė testo su deterministiniu trendu, nei testo su laisvuojamu nariu atveju.

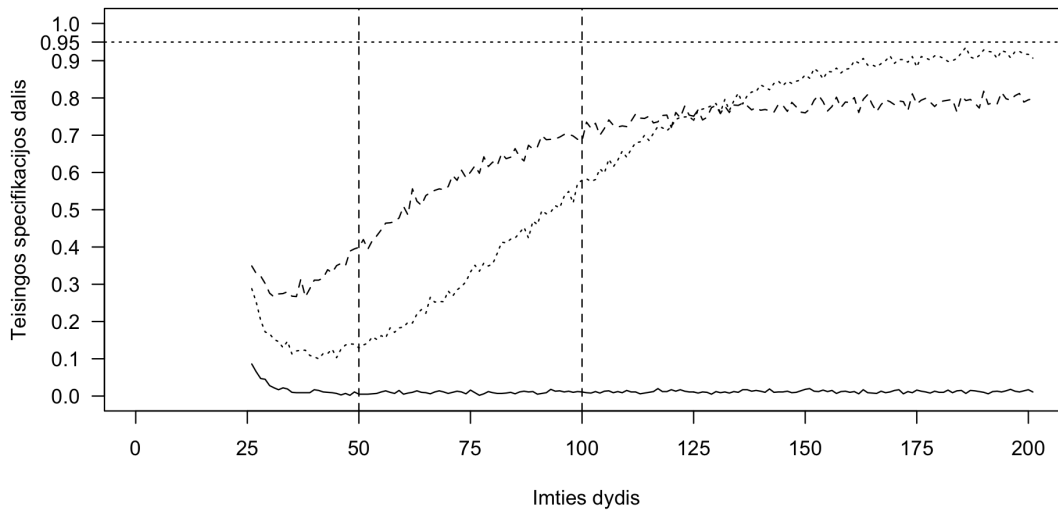


128 pav.: Antros rūšies klaida naudojant pėdsako statistiką

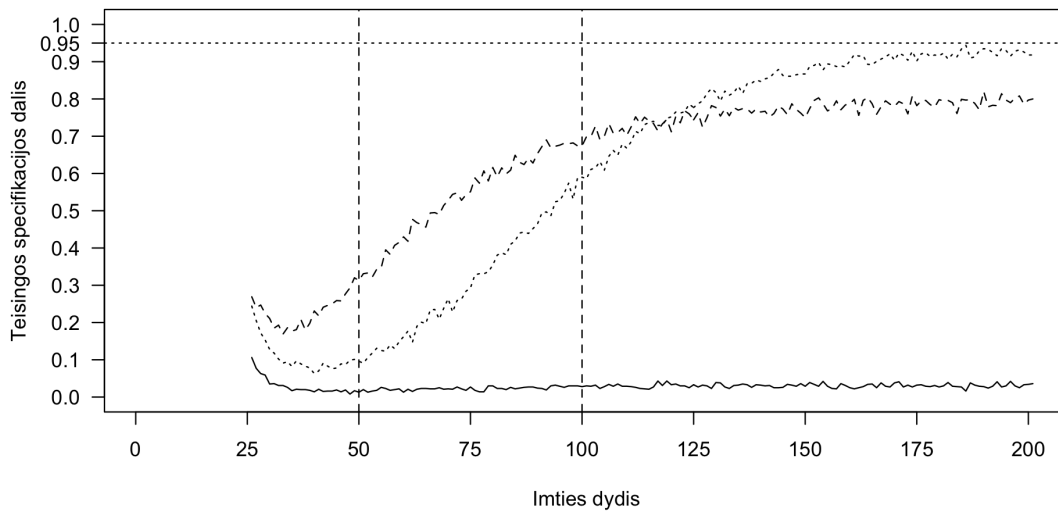


129 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

Du kointegruojantys vektoriai. Paveikslėlis 130 palygina visų specifikacijų Johansen procedūras naudojant pėdsako statistiką, o paveikslėlis 131 - naudojant maksimalios tikrinės reikšmės statistiką, tiriant antros rūšies klaidas. Šiuo atveju testų rezultatai labiausiai diverguoja. Testas be deterministinių komponentų yra visiškai netinkamas, testas su laisvuju nariu yra tinkamesnis mažose ir vidutinėse imtyse, tačiau labai didelėse imtyse (> 125) tinkamesnis yra testas su trendu.

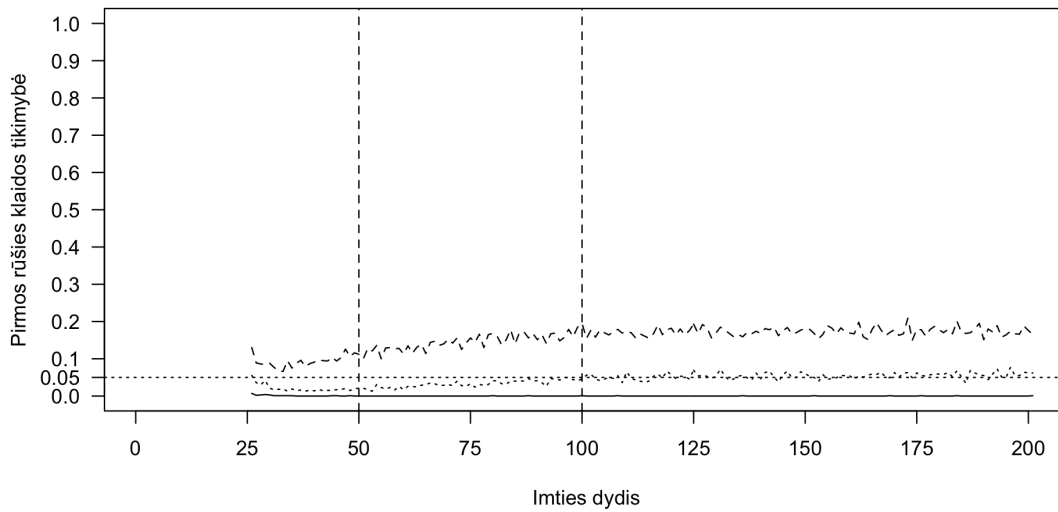


130 pav.: Teisingai nustatomas vektorių skaičius naudojant pėdsako statistiką

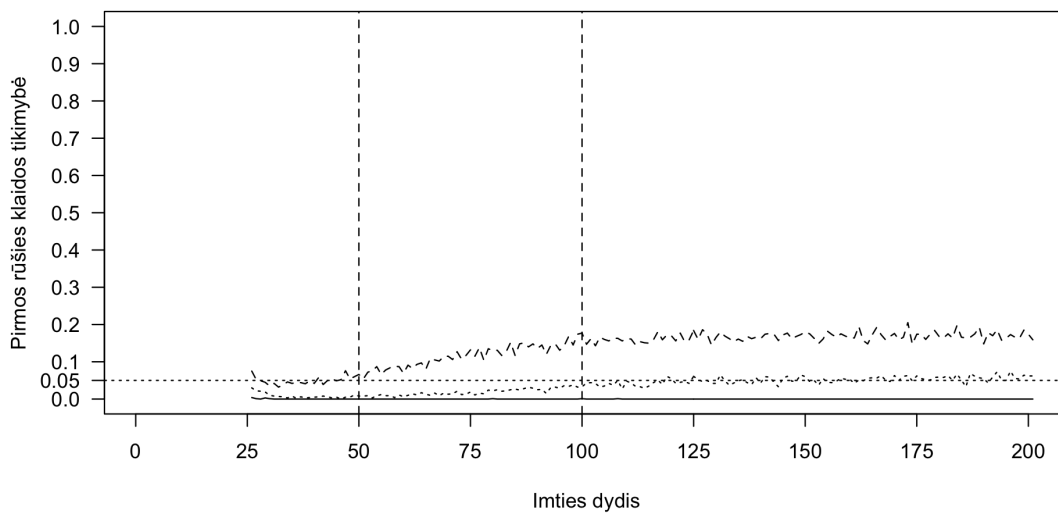


131 pav.: Teisingai nustatomas vektorių skaičius naudojant maksimalios tikrinės reikšmės statistiką

Paveikslėlis 132 parodo pirmos rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 133 - naudojant maksimalios tikrinės reikšmės statistiką. Johansen procedūra be deterministinių komponentų pirmos rūšies klaidos tikimybė yra tarp 0%, procedūros su laisvuju nariu tarp 5% ir 25%, o procedūros su trendu apie 5%.



132 pav.: Pirmos rūšies klaida naudojant pėdsako statistiką

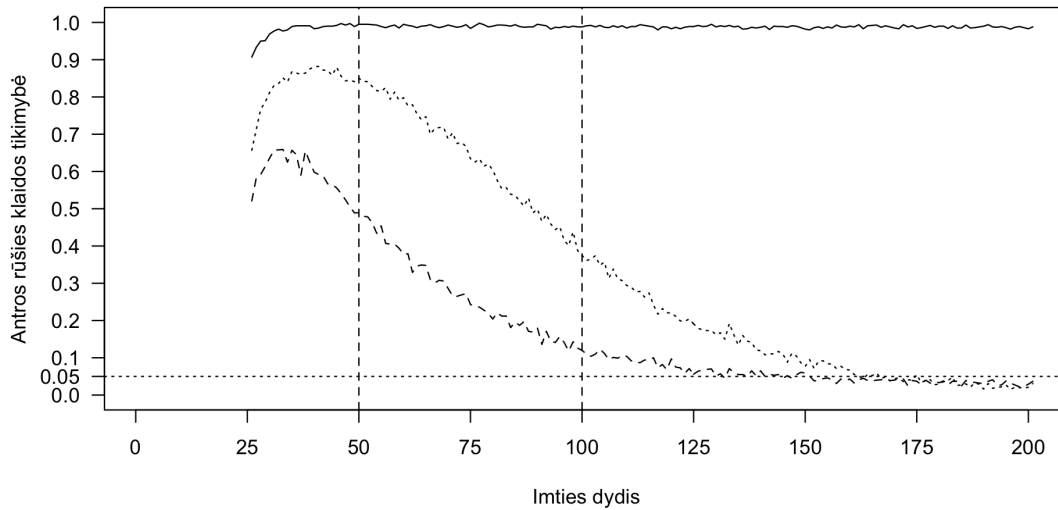


133 pav.: Pirmos rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

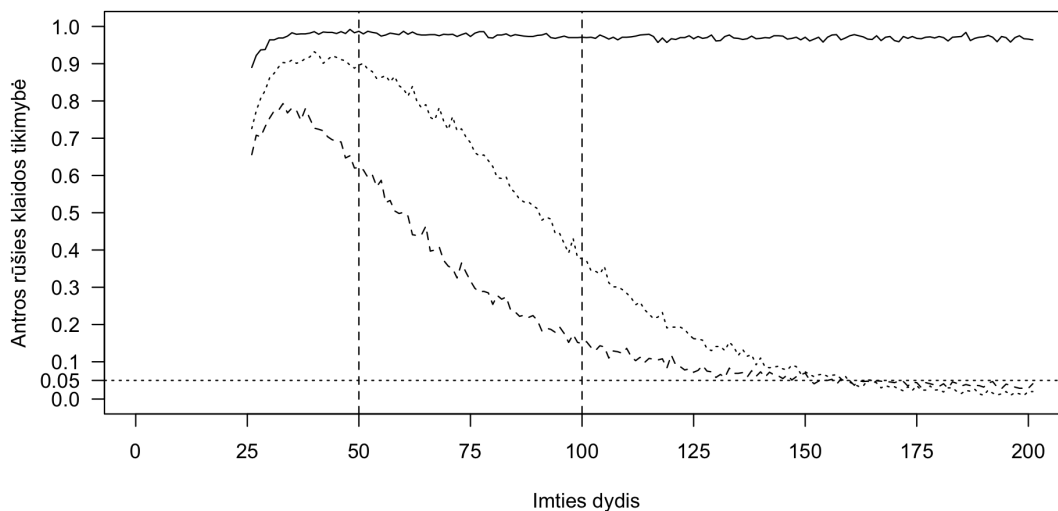
Paveikslėlis 134 parodo antros rūšies klaidos tikimybę visų specifikacijų Johansen procedūroms naudojant pėdsako statistiką, o paveikslėlis 135 - naudojant maksimalios tikrinės reikšmės statistiką. Ypač mažose imtyse (< 30) tiksliausias yra testas su trendu, tuo kitais atvejais tiksliausias yra testas be deterministinių komponentų, o testas su laisvuju nariu yra tikslesnis už testą su trendu, bet mažiau tikslus už testą be deterministinių komponentų.

Johansen testas be deterministinių komponentų šią klaidą padaro apie 90% atvejų, todėl jis nėra tinkamas. Johansen procedūra su konstanta mažiau linkusi

padaryti šią klaidą mažose ir vidutinėse ir didelėse imtyse. Labai didelėse imtyse šios klaidos tikimybė yra vienoda testo su deterministiniu trendu ir su laisvuoju nariu atvejais - 5%.



134 pav.: Antros rūšies klaida naudojant pėdsako statistiką



135 pav.: Antros rūšies klaida naudojant maksimalios tikrinės reikšmės statistiką

3.5. Monte Carlo simuliacijų rezultatai

Monte Carlo simuliacijos parodė, kad deterministinių komponentų specifikacija yra labai svarbi modeliuojant kointegruotus kintamuosius. Testų nustatomas koin-

tegruojančių vektorių skaičius skiriasi priklausomai nuo Johansen procedūros specifikacijos. Tyrimas taip pat atskleidė, kad deterministinių komponentų specifikacijos įtaka labai priklauso nuo imties dydžio. Kai kuriais atvejais pėdsako testai duodavo geresnius rezultatus, tačiau kitais atvejais tikslesni buvo maksimalios tikrinės reikšmės testai.

3.5.1. Modelis be deterministinių komponentų

Lentelėje 1 pateikiami VECM modelio be deterministinių komponentų tikimybės teisingai nustatyti kointegruojančių vektorių skaičių. Labai mažose imtyse tiksliausia yra Johansen procedūra be deterministinių komponentų naudojant pėdsako statistiką. Mažose imtyse tiksliausia yra Johansen procedūra be deterministinių komponentų: naudojant tiek maksimalios tikrinės reikšmės tiek pėdsako testus rezultatas yra toks pat. Vidutinėse imtyse vėl tiksliausia yra Johansen procedūra be deterministinių komponentų naudojant pėdsako statistiką. Didelėse ir labai didelėse imtyse didžiausią tikimybę nustatyti teisingą kointegruojančių vektorių skaičių turi Johansen procedūra be deterministinių komponentų naudojant maksimalios tikrinės reikšmės statistiką.

1 lentelė.: Tikimybė teisingai nustatyti kointegruojančių vektorių skaičių

	Johansen procedūra be deterministinių komponentų		Johansen procedūra su laisvuju nariu		Johansen procedūra su trendu	
	MTR statistika	P statistika	MTR statistika	P statistika	MTR statistika	P statistika
Labai mažos imtys	0.31	0.32	0.3	0.28	0.28	0.26
Mažos imtys	0.38	0.38	0.36	0.34	0.34	0.33
Vidutinės imtys	0.46	0.47	0.44	0.43	0.41	0.41
Didelės imtys	0.79	0.77	0.77	0.74	0.71	0.69
Labai didelės imtys	0.94	0.93	0.93	0.92	0.93	0.92

3.5.2. Modelis su laisvuju nariu

Lentelėje 2 pateikiami VECM modelio su laisvuju nariu tikimybės teisingai nustatyti kointegruojančių vektorių skaičių. Visuose imties dydžiuose tiksliausia yra Johansen procedūra be deterministinių komponentų naudojant maksimalios tikrinės reikšmės statistiką. Labai mažose imtyse tokią pačią galią turi ir Johansen procedūra be deterministinių komponentų naudojant pėdsako statistiką, o labai didelėse imtyse dar ir Johansen procedūra su laisvuju nariu naudojant tiek maksimalios tikrinės reikšmės tiek pėdsako statistikas. Šis rezultatas yra netikėtas, nes teisingai specifiukuota t.y. Johansen procedūra su laisvuju nariu tik labai didelėse imtyse turi tokią pačią galią kaip Johansen procedūra be deterministinių komponentų naudojant maksimalios tikrinės reikšmės statistiką.

2 lentelė.: Tikimybė teisingai nustatyti kointegruojančių vektorių skaičių

	Johansen procedūra be deterministinių komponentų		Johansen procedūra su laisvuju nariu		Johansen procedūra su trendu	
	MTR statistika	P statistika	MTR statistika	P statistika	MTR statistika	P statistika
Labai mažos imtys	0.30	0.30	0.27	0.27	0.29	0.28
Mažos imtys	0.36	0.35	0.34	0.33	0.34	0.31
Vidutinės imtys	0.45	0.44	0.43	0.42	0.39	0.39
Didelės imtys	0.81	0.78	0.77	0.75	0.71	0.69
Labai didelės imtys	0.93	0.93	0.93	0.93	0.92	0.92

3.5.3. Modelis su trendu

Lentelėje 3 pateikiami VECM modelio su trendu tikimybės teisingai nustatyti kointegruojančių vektorių skaičių. Labai mažose, mažose ir vidutinėse imtyse tiksliausia yra Johansen procedūra su laisvuju nariu: tiek naudojant maksimalios tikrinės reikšmės tiek pėdsako statistikas. Kai imtys yra didelės ir labai didelės tiksliausia yra Johansen procedūra su trendu naudojant maksimalios tikrinės reikšmės

statistiką. Šis rezultatas yra netikėtas, nes teisingai specifikuota t.y. Johansen procedūra su trendu tik didelėse ir labai didelėse imtyse yra pranašesnė už Johansen procedūrą be su laisvuju nariu.

3 lentelė.: Tikimybė teisingai nustatyti kointegruojančių vektorių skaičių

	Johansen procedūra be deterministinių komponentų		Johansen procedūra su laisvuju nariu		Johansen procedūra su trendu	
	MTR statistika	P statistika	MTR statistika	P statistika	MTR statistika	P statistika
Labai mažos imtys	0.3	0.3	0.31	0.31	0.28	0.28
Mažos imtys	0.29	0.28	0.36	0.36	0.33	0.32
Vidutinės imtys	0.32	0.3	0.48	0.48	0.39	0.39
Didelės imtys	0.36	0.35	0.64	0.63	0.71	0.69
Labai didelės imtys	0.36	0.35	0.66	0.64	0.92	0.9

3.5.4. Monte Carlo simuliacijų išvados

Monte Carlo simuliacijos parodė, kad modeliai su deterministiniais komponentais yra sunkiau teisingai įvertinami nei modeliai be deterministinių komponentų. Ypač sudėtingas modelių su deterministiniais komponentais vertinimas mažose imtyse - Johansen procedūrų su deterministiniais komponentais galia yra ypatingai silpna, kai imtys yra mažos.

Lentelėje 4 pateikiami rekomenduojami naudoti Johansen procedūrų tipai, priklausomai nuo modelio tipo ir imties dydžio. Visais atvejais maksimalios tikrinės reikšmės testai yra tikslesni už pėdsako testus, išskyrus modelius be deterministinių komponentų labai mažose ir vidutinėse imtyse, tačiau net ir šiais atvejais skirtumas yra labai nežymus (0.1 procentinio punkto). Siekiant atsakyti į klausimą ar tikrai egzistuoja skirtumas tarp šių dviejų testų tipų buvo atliktas papildomas tyrimas naudojant 10000 iteracijų vietoj 1000 naudotų pagrindiniame tyrime. Šis tyrimas parodė, kad skirtumas yra, tačiau jis sudaro tik mažiau nei 0.1 procentinio

punkto. Rekomenduojami Johansen procedūrų tipai yra skirti tik maksimaliai teisingam kointegruojančių vektorių skaičiaus nustatymui, parametrų įvertinimui turi būti naudojamos teisingai specifikuotos procedūros.

4 lentelė.: Monte Carlo simuliacijų rekomenduojami Johansen procedūros tipai

	Modelis be deterministinių komponentų	Modelis su laisvuju nariu	Modelis su trendu
Labai mažos imtys	Johansen procedūra be deterministinių komponentų naudojant pėdsako statistiką	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra su laisvuju nariu naudojant MTR statistiką
Mažos imtys	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra su laisvuju nariu naudojant MTR statistiką
Vidutinės imtys	Johansen procedūra be deterministinių komponentų naudojant pėdsako statistiką	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra su laisvuju nariu naudojant MTR statistiką
Didelės imtys	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra su trendu naudojant MTR statistiką
Labai didelės imtys	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra be deterministinių komponentų naudojant MTR statistiką	Johansen procedūra su trendu naudojant MTR statistiką

4. Deterministinių komponentų vaidmuo modeliuojant pinigų paklausą

3 skyriuje atlikta sintetinių VECM analizė parodė, kad ne visada teisinga Johansen procedūros specifikacija nustato kointegruojančių vektorių skaičių tiksliausiai. Šiame skyriuje 3 skyriuje gauti rezultatai bus pritaikyti praktiškai tiriant JAV ir Lietuvos pinigų paklausas. Tokiu būdu bus pademonstruota kaip pasirinkti Johansen procedūros specifikaciją ir testo tipą tiriant realius makroekonominis kintamuosius.

Charakteringoji VECM ypatybė yra tiek pokyčio, tiek lygio išraiškų įtraukimas į vieną modelį leidžiantis nagrinėti tiek trumpalaikius, tiek ir ilgalaikius poveikius. Kaip teigia Hendry ir Juselius (2000), koeficientų interpretavimas dinaminėse efektų kontekste yra sudėtingas. Ši išvada taip pat galioja ir trendui bei konstantai ir kitiems deterministiniams komponentams. Dėl šios priežasties deterministinių kintamųjų vaidmens analizei modeliuojant kointegruotus kintamuosius buvo pasitikta modeliuoti makroekonominis kintamuosius - JAV ir Lietuvos pinigų paklausas.

Pinigų paklausos makroekonominiai modeliai teigia, kad pinigų paklausos augimas viršijantis realų produktyvumo augimą sukelia infliaciją ilguoju laikotarpiu ir kartais trumpuoju laikotarpiu Juselius (2006).

4.1. Tiriamas makroekonominis modelis

Mokslinėje literatūroje yra išskirta daug skirtingų makroekonominių modelių tinkamų pinigų paklausai modeliuoti. Šiame tyrime bus naudojamas supaprastintas Clarida ir kiti (2000) pristatytas trijų lygčių kanoninis naujasis keinsistinis modelis.

$$y_t = E_t[y_{t+1}] - \frac{1}{\Phi} r_t + u_t^{IS}, \quad \phi > 0, \quad (116)$$

$$\pi_t = \beta E_t[\pi_{t+1}] + k y_t + u_t^\pi, \quad 0 < \beta < 1, \quad k > 0, \quad (117)$$

$$r_t = \phi_\pi E_t[\pi_{t+1}] + \phi_y E_t[y_{t+1}] + u_t^{MP}, \quad \phi_\pi > 0, \quad \phi_y \geq 0 \quad (118)$$

Lygtis (116) yra naujoji keinsistinė IS kreivė, lygtis (117) yra naujoji keinsistinė Phillips kreivė, o lygtis (118) yra palūkanų normų taisyklė. Paklaidos šiame modelyje įgauna nepriklausomo AR(1) proceso formą.

$$u_t^{IS} = \rho_{IS} u_{t-1}^{IS} + e_t^{IS}, \quad -1 < \rho_{IS} < 1 \quad (119)$$

$$u_t^\pi = \rho_\pi u_{t-1}^\pi + e_t^\pi, \quad -1 < \rho_\pi < 1 \quad (120)$$

$$u_t^{MP} = \rho_{MP} u_{t-1}^{MP} + e_t^{MP}, \quad -1 < \rho_{MP} < 1 \quad (121)$$

kur e^{IS} , e^π ir e^{MP} yra baltojo triukšmo procesai.

Šiame modelyje naujoji keinsistinė Phillips kreivė yra naudojama modeliuoti infliacijai.

Kadangi šio skyriaus tikslas yra pademonstruoti Johansen procedūros pasirinkimą ir pritaikymą įvertinant VECM modelius, o šiame skyrelyje aprašyto trijų lygčių kanoninio naujojo keinsistinio modelio tiesiogiai išmatuoti nėra galimybės, šis modelis buvo supaprastintas į VECM modelį su trim kintamaisiais: palūkanu norma, realiuoju M1 pinigų kiekiu ir realiuoju BVP. Tikslesnė modelio specifikacija bus sudaryta tyrimo metu, nustatant autoregresinių parametrų vėlavimo eiles, deterministinių komponentų specifikacijas ir kointegruojančių vektorių skaičių.

4.2. VECM specifikacijų ir įvertinimų palyginimų metodologija

Šiame skyriuje nagrinėjami VECM specifikacijos ir įvertinimo analizei taikytini tyrimo metodai ir priemonės.

VECM modelių specifikacijos. Egzistuoja įvairūs deterministinių komponentų įtraukimo į VECM būdai. Ištiriamos tokios VECM specifikacijos:

1. VECM be deterministinių komponentų
2. VECM su laisvuoju nariu
3. VECM su trendu

Būtina įvertinti ar visos VECM specifikacijos yra tinkamos konkretiems modeliuojamiems kintamiesiems. Kai kurios specifikacijos gali prieštarauti makroekonomikos teorijai ir todėl jų įvertinimas neturėtų prasmės. Taip pat žinoma, kad modelio specifikacija keičia ne tik parametrų įverčius, bet ir jų prasmę. Paklaidos korekcijos komponente esantys deterministiniai komponentai keičia prasmę priklausomai nuo to ar jie yra kointegruoti ar ne.

VECM modelių palyginimo metodologija. VECM specifikacijos sudaromos remiantis žiniomis apie duomenis generuojantį procesą. Pinigų paklausos prognozavimo atveju modelis specifikuojamas remiantis makroekonomikos teorija. Modelių specifikuojimas bus pradedamas nuo paprasčiausio VECM modelio be deterministinių komponentų. Tada bus specifikuojami modeliai su trendu ir laisvuoju nariu, bet be fiktyvių kintamųjų. Paskui bus specifikuojami VECM su įvairiais fiktyviais kintamaisiais. Fiktyvūs kintamieji įvertins įvairius ilgalaikius ir trumpalaikius šokus ir intervencijas, kurios reikšmingai įtakoja VECM įvertinimą.

VECM modelių įvertinimai. VECM įvertinimui bus naudojamos įvairios Johansen procedūros specifikacijos. Ši procedūra pasirinkta dėl gerai ištirtų deterministinių komponentų įtakos įvertinimui, galimybės įvertinti keletą kointegruojančių vektorių ir galimybės tiksliai nustatyti įverčius mažose imtyse. Engle - Granger dviejų pakopų metodas nebus naudojamas dėl trūkumų aprašytų 2.2 skyriuje. Įvertinami skirtingai specifikuoti VECM modeliai. Įvertinami skirtumai tarp skirtingai specifikuotų VECM parametrų įverčių. Atliekama modelių diagnostika.

Deterministinių komponentų vaidmens nustatymo metodologija. Palyginami Johansen procedūros rezultatai ir nustatomas tinkamiausias modelis pinigų paklausos tyrimui. Kadangi pradedama nuo modelio be deterministinių komponentų ir modelis yra plečiamas palaipsniui palyginama, kokią įtaką rezultatams turi kiekvienas naujai įtrauktas deterministinis komponentas. Jei naujai įtrauktas komponentas nėra reikšmingas įvertinama, ar kitaip specifikavus modelį šis kintamasis gali tapti reikšmingas. Galiausiai pasirenkamas tinkamiausias modelis JAV pinigų paklausos įvertinimui ir pateikiamos išvados, kokį vaidmenį turi deterministiniai komponentai kointegruotų procesų analizėje.

4.3. Deterministinių komponentų įtaka modeliuojant JAV pinigų paklausą

Kointegruotų kintamųjų analizėje pinigų paklausa yra vienas geriausiai ištirtų modelių. Pinigų paklausos tyrimus darė Johansen ir Juselius (1990), Anderson ir kiti (2002), Ahking (2002), Adam ir Hendry (2000) ir Doornik ir kiti (1998). Tikslus pinigų paklausos prognozavimas yra labai svarbus šalies monetarinės politikos formavimui. JAV pinigų paklausa buvo pasirinkta vertinimui dėl gero kintamojo ištyrimo makroekonominio požiūriu.

4.3.1. Pinigų paklausos tyrimo modelis ir kintamųjų pasirinkimas

Ekonomikos literatūroje yra pateikiama daug pinigų paklausos modelių aprašymų. Šiame skyriuje bus tiriamas nesudėtingas VECM modelis, kuriame pinigų paklausa bus vertinama kaip kainų lygio, palūkanų normos ir bendrojo vidaus produkto funkcija. Pagrindinis kintamasis, kuris bus naudojamas kaip pinigų paklausos matas yra M1 pinigų paklausa. Pinigų paklausa taip pat gali būti vertinama ir M2 arba M3. Palūkanų normos pasirinkimas nėra trivialus, nes kaip palūkanų normą galima naudoti FED palūkanų normą, trumpojo laikotarpio valstybės išdo obligacijas, vidutinio laikotarpio valstybės išdo obligacijas arba ilgojo laikotarpio valstybės išdo obligacijas. Šiuo atveju tyrimui pasirinkta FED palūkanų norma.

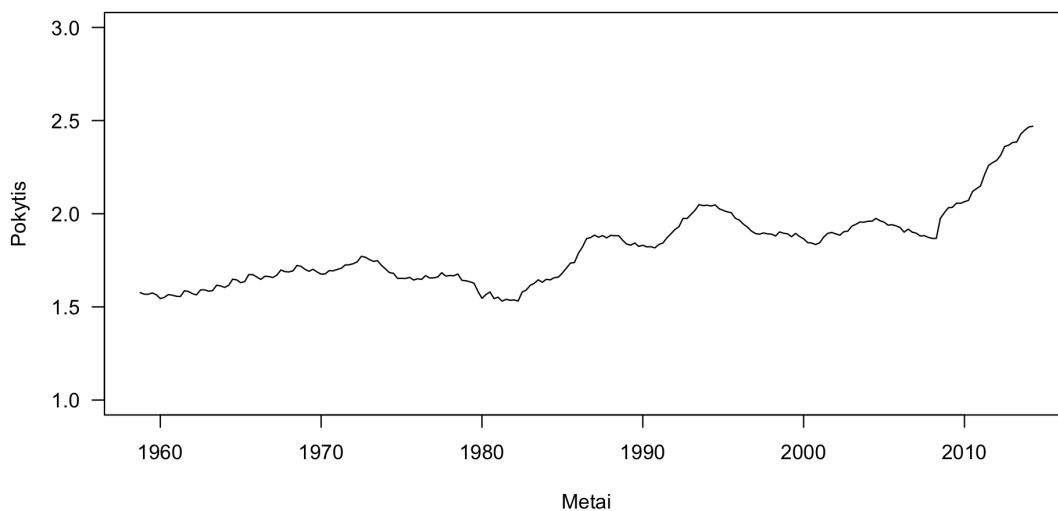
Tyrimo naudojami šie kintamieji:

1. reali M1 pinigų pasiūla
2. FED palūkanų norma
3. realus JAV BVP

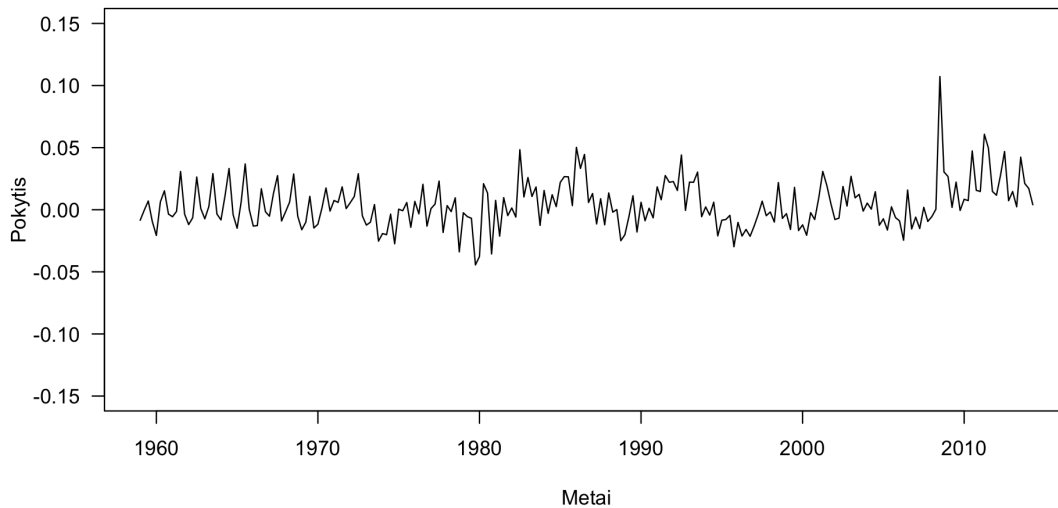
4.3.2. Aprašomoji duomenų analizė

Reali JAV M1 pinigų pasiūla.. Reali M1 pinigų pasiūla yra suskaičiuojama nominalią M1 pinigų pasiūlą padalijant iš atitinkamo periodo vartotojų kainų indekso (VKI). Tada kintamasis yra logaritmuojamas.

Grafinė analizė parodo, kad M1 yra nestacionarus kintamasis. Formaliai integruotumas patikrinamas naudojant papildytąją Dickey-Fuller testą.



136 pav.: Realios JAV M1 pinigų pasiūlos 1983 m. kainomis logaritmas



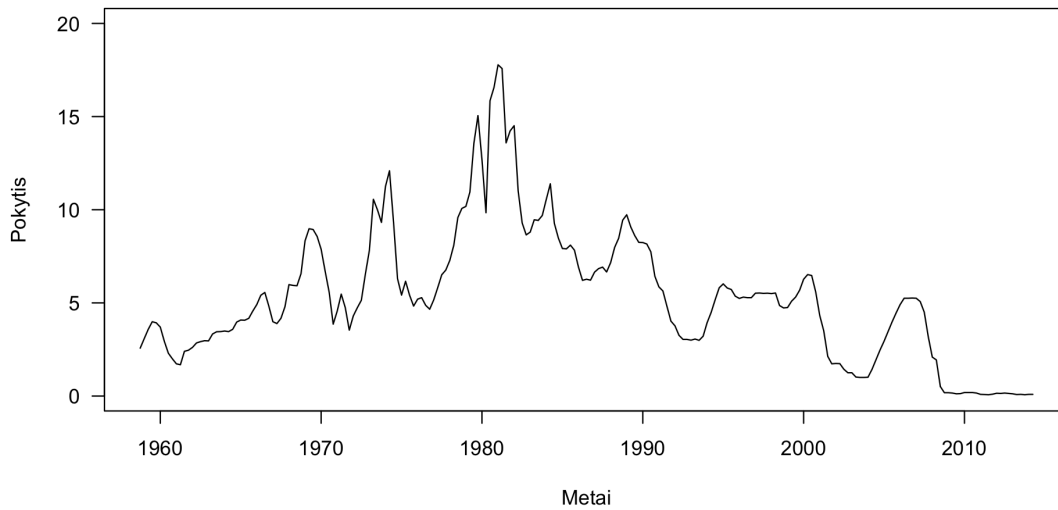
137 pav.: Diferencijuotas realios JAV M1 pinigų pasiūlos 1983 m. kainomis logaritmas

Papildytasis Dickey-Fuller testas parodė, kad kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi vienetinės šaknies. Diferencijuojant kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintamasis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad kintamasis yra stacionarus.

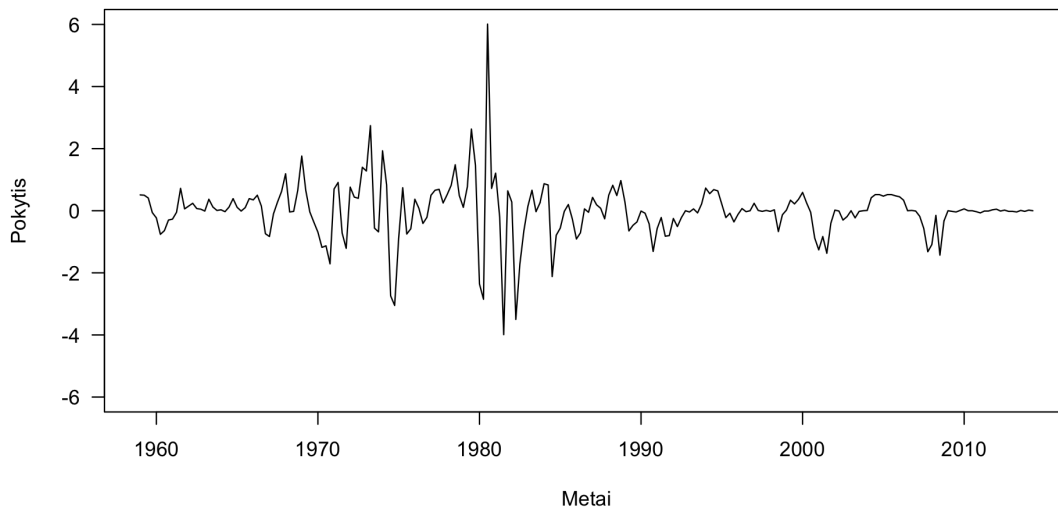
Grafinė kintamojo dinamikos analizė leidžia daryti prielaidą, kad kintamasis turi laisvąjį narį ir deterministinį trendą.

FED palūkanų norma. FED palūkanų norma yra pateikiama nemodifikuota.

Grafinė analizė parodo, kad FED yra nestacionarus kintamasis. Formaliai integruotumas patikrinamas naudojant papildytąjį Dickey-Fuller testą.



138 pav.: JAV FED palūkanų norma



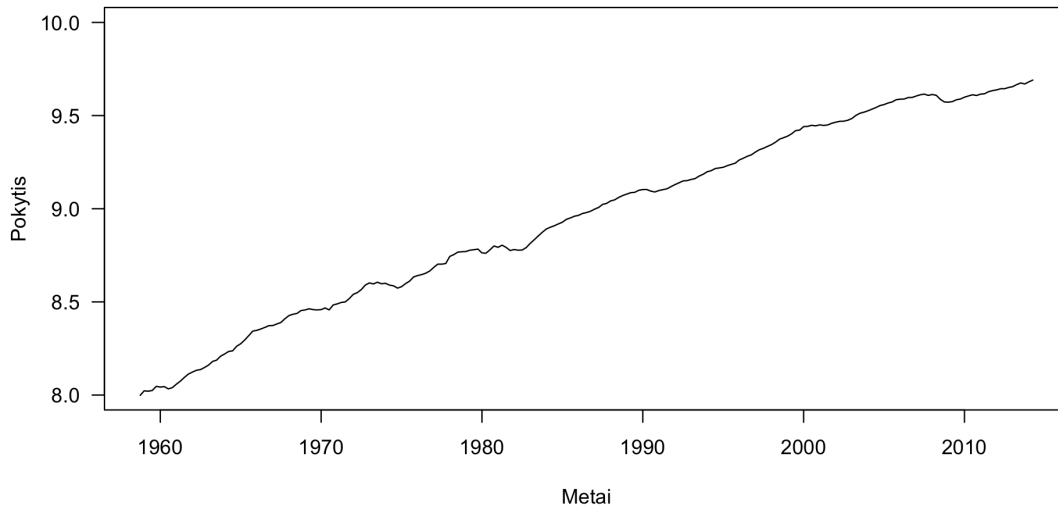
139 pav.: Diferencijuota JAV FED palūkanų norma

Papildytasis Dickey-Fuller testas parodė, kad kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi vienetinės šaknies. Diferencijuotą kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintamasis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad kintamasis yra stacionarus.

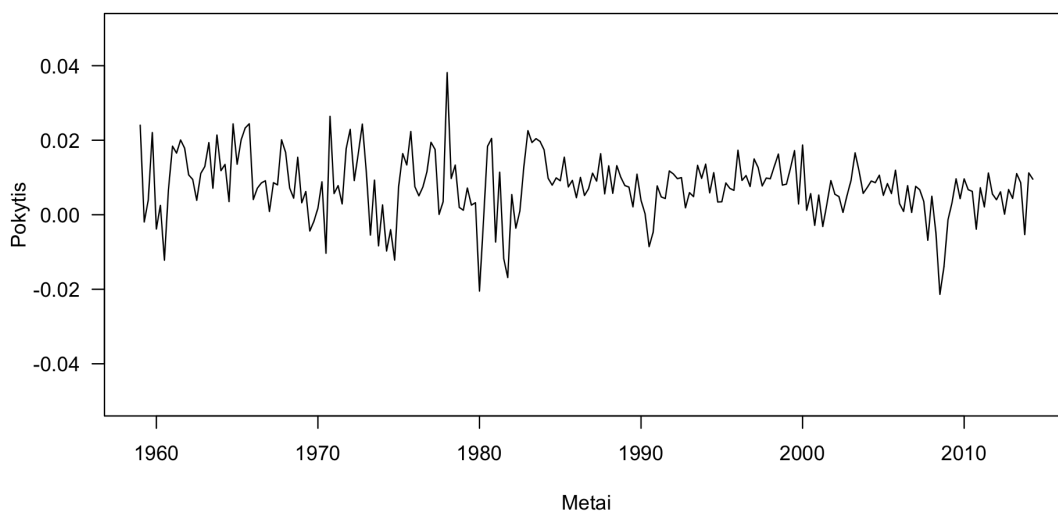
Grafinė kintamojo dinamikos analizė leidžia daryti prielaidą, kad kintamasis turi laisvąjį narį, bet deterministinio trendo greičiausiai neturi.

Realus JAV BVP. Realus JAV BVP yra suskaičiuojamas nominalų JAV BVP padalijant iš atitinkamo periodo vartotojų kainų indekso. Tada kintamasis yra logaritmuojamas.

Grafinė analizė parodo, kad BVP yra nestacionarus kintamasis. Formaliai integruotumas patikrinamas naudojant papildytąją Dickey-Fuller testą.



140 pav.: Realaus JAV BVP 1983 m. kainomis logaritmas



141 pav.: Diferencijuotas realaus JAV BVP 1983 m. kainomis logaritmas

Papildytasis Dickey-Fuller testas parodė, kad kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi

vienetinės šaknies. Diferenciuojant kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintamasis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad kintamasis yra stacionarus.

Grafinė kintamojo dinamikos analizė leidžia daryti prielaidą, kad kintamasis turi laisvąjį narį ir deterministinį trendą.

4.3.3. Modelio specifikacija

Siekiant nustatyti ar kintamieji yra kointegruoti pirma reikia sudaryti VECM modelį.

Vėlavimų eilės pasirinkimas. VECM vėlavimų eilės įvertinimui buvo apskaičiuoti šie informaciniai kriterijai: AIC, HQ, SC ir FPE. Kriterijai AIC ir FPE indikuoja VECM(12), HQ indikuoja VECM(6), o SC indikuoja VECM(5). Vėlavimų eilė buvo pasirinkta įvertinus kintamųjų lygio išraiškų VAR(5), VAR(6) ir VAR(12) modelių autokoreliacijas. Modelis VECM(5) turi žymiai daugiau reikšmingų autokoreliacijų nei VECM(8). Įvertinis modelių VECM(6), VECM(7), VECM(8), VECM(9), VECM(10), VECM(11) ir VECM(12) korelogramas ir taip pat atsižvelgus į tai, kad modeliuojami ketvirtiniai duomenys buvo pasirinktas VECM(8) modelis.

Deterministinių komponentų pasirinkimas. Kadangi visi trys kintamieji turi laisvuosius narius ir du iš trijų kintamųjų turi deterministinius trendus daroma prielaida, kad VECM modelis turi konstantą ir trendą kointegruojančiame komponente ir konstantą VAR dalyje.

Tiriamas VECM modelis. Lygtis (122) parodo tiriamojo VECM modelio specifikaciją.

$$y_t = \delta_1 + \delta_2 t + \sum_{p=1}^8 \alpha_p y_{t-p} + \sum_{p=1}^8 \beta_p r_{t-p} + \sum_{p=1}^8 \gamma_p m_{t-p} + u_t \quad (122)$$

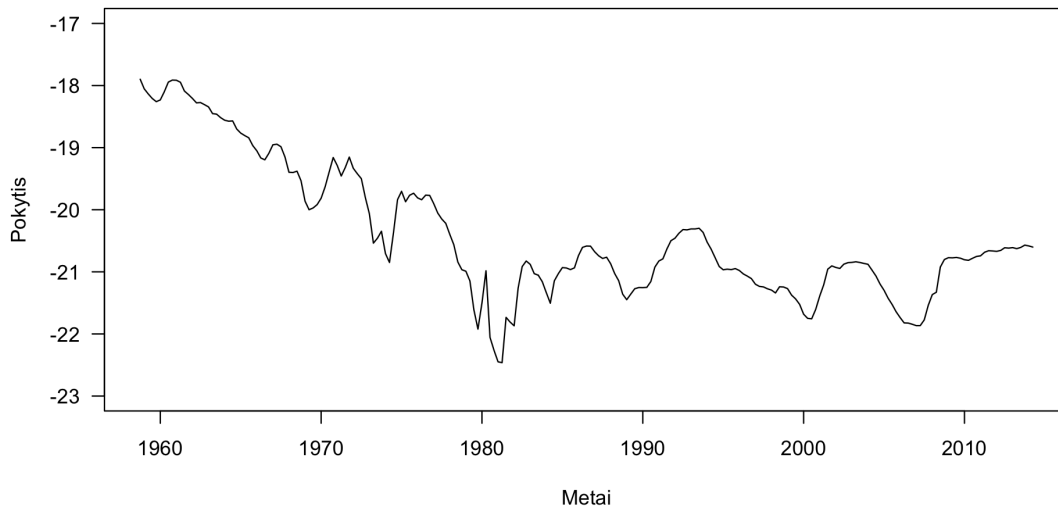
$$r_t = \delta_3 + \delta_4 t + \sum_{p=1}^8 a_p r_{t-p} + \sum_{p=1}^8 b_p y_{t-p} + \sum_{p=1}^8 c_p m_{t-p} + v_t \quad (123)$$

$$m_t = \delta_5 + \delta_6 t + \sum_{p=1}^8 \rho_p r_{t-p} + \sum_{p=1}^8 \zeta_p y_{t-p} + \sum_{p=1}^8 \eta_p m_{t-p} + w_t \quad (124)$$

4.3.4. Rezultatų palyginimas

Johansen procedūra be deterministinių komponentų. Įvertinus VECM(8) naudojant Johansen procedūrą be deterministinių komponentų tiek pagal maksima-

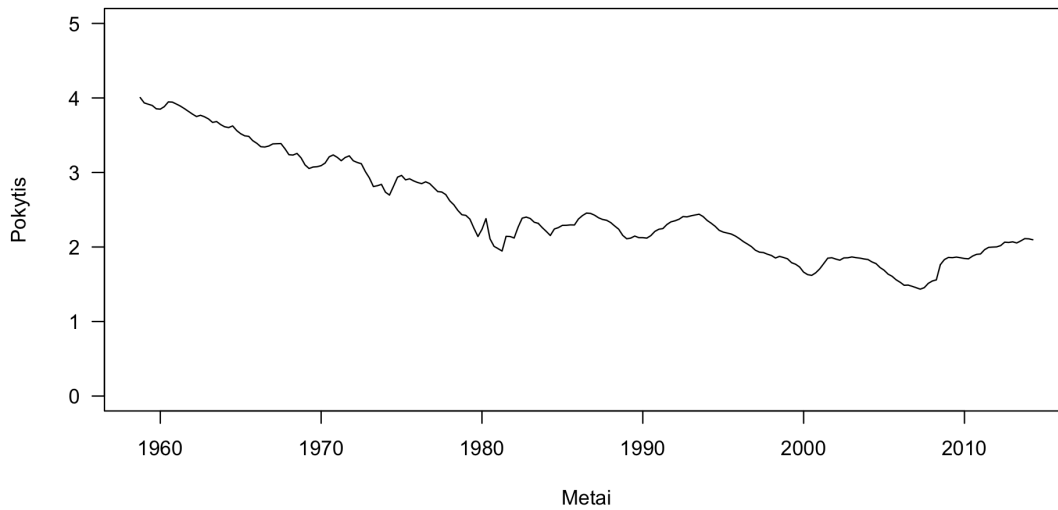
lios tikrinės reikšmės, tiek pagal pėdsako statistikas negalime atmesti H_0 teigiančios, kad $rk(\Pi) = 0$.



142 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 142 parodo VECM modelio paklaidos korekcijos komponentą. Aki-vaizdu, kad šis komponentas nėra stacionarus.

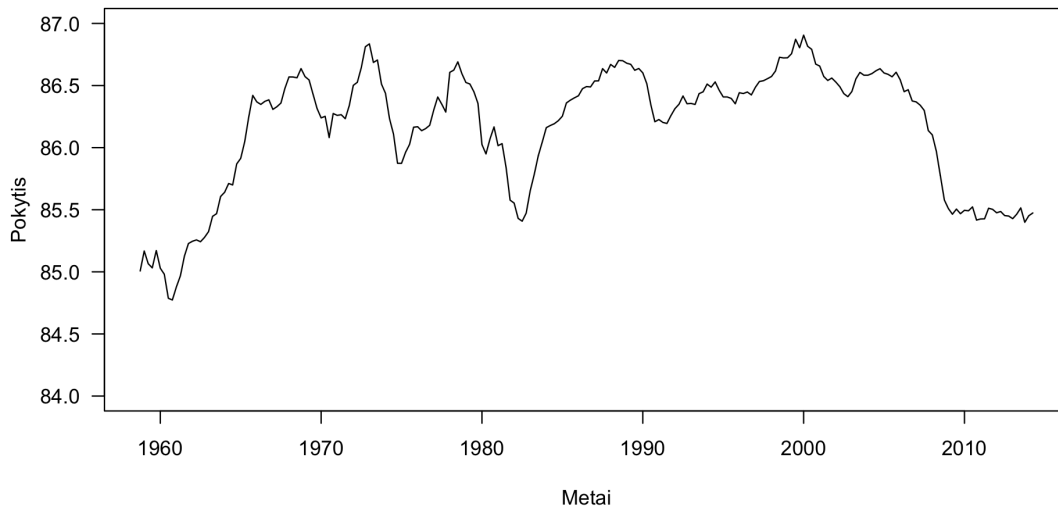
Johansen procedūra su laisvuju nariu. Įvertinus VECM(8) su laisvuju nariu tiek pėdsako, tiek ir maksimalios tikrinės reikšmės statistikos indikuoja, kad $rk(\Pi) = 1$ t.y. kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Įvertinti normalizuoto kointegruojančio vektoriaus parametrai: $\beta = (1, -0.58, 0.02, -9.48)$.



143 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 143 parodo VECM modelio paklaidos korekcijos komponentą. Aki-vaizdu, kad šis komponentas nėra stacionarus, todėl taip specifikuotas modelis nėra tinkamas.

Johansen procedūra su trendu. Įvertinus VECM(8) su trendu ir laisvuoju nariu tiek pagal maksimalios tikrinės reikšmės, tiek pagal pėdsako statistikas negalime atmesti H_0 teigiančios, kad $rk(\Pi) = 0$. Kadangi pagal pradinę prielaidą ši procedūra yra tinkama tokiam modeliui įvertinti, galima daryti išvadą, kad kintamieji nėra kointegruoti.



144 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 144 parodo VECM modelio paklaidos korekcijos komponentą. Ir šiuo atveju laikoma, kad šis komponentas nėra stacionarus, todėl taip specifikuotas modelis nėra tinkamas.

VECM specifikuojamo įtaka modelio įvertinimui. Įvertinus VECM modelį naudojant tris skirtingai specifikuotas Johansen procedūras du iš trijų įvertinimų - VECM be deterministinių komponentų ir VECM su trendu ir laisvuju nariu - nurodo, kad kintamieji nėra kointegruoti, o VECM su laisvuju nustato, kad kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Kadangi nei viena Johansen procedūros specifikacija negali sukurti tokių β vektoriaus parametrų su kuriais modelis būtų stacionarus laikome, kad kintamieji yra nekointegruoti ir Johansen procedūra su laisvuju nariu padarė I rūšies klaidą. Šios klaidos tikimybė labai didelėse imtyse modeliuose neturinčiuose kointegruojančių vektorių su laisvojo nario ir trendo deterministiniais komponentais yra labai didelė (virš 50%) ir didėjanti didėjant imties dydžiui.

4.4. Deterministinių komponentų įtaka modeliuojant Lietuvos pinigų paklausą

Lietuvos pinigų paklausa buvo pasirinkta vertinimui dėl kintamojo aktualumo ir Lietuvos ekonomikos, monetarinės politikos ir politinės kultūros skirtumų nuo JAV.

4.4.1. Pinigų paklausos tyrimo modelis ir kintamųjų pasirinkimas

Pagrindinis kintamasis, kuris bus naudojamas kaip pinigų paklausos matas yra M1 pinigų pasiūla. Palūkanų normos pasirinkimas kaip ir JAV tyrime nėra trivialus. Šiuo atveju palūkanų normų matu yra pasirinktas LIBOR.

$y_t = \text{realaus BVP logaritmas}$

$r_t = \text{LIBOR}$

$m_t = \text{realios pinigų pasiūlos (M1) logaritmas}$

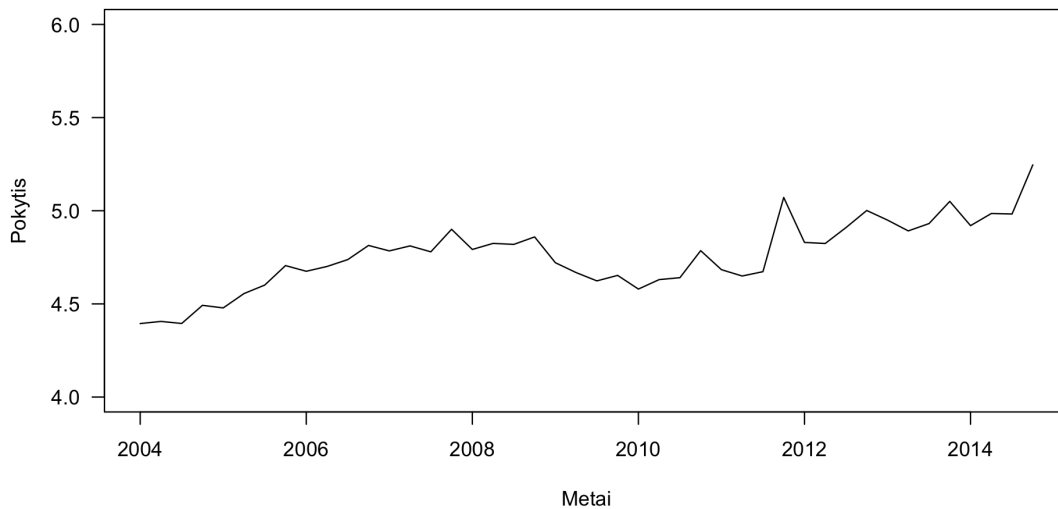
$\delta = \text{deterministiniai komponentai}$

(125)

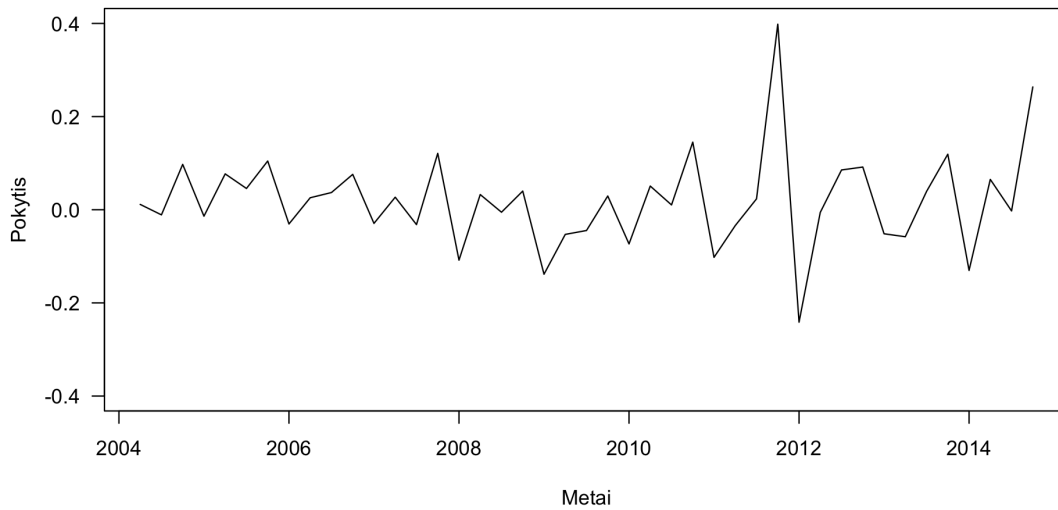
4.4.2. Aprašomoji duomenų analizė

Realioji Lietuvos M1 pinigų pasiūla. Realioji M1 pinigų pasiūla yra suskaičiuojama nominalią M1 pinigų pasiūlą padalijant iš atitinkamo periodo vartotojų kainų indekso (VKI). Tada kintamasis yra logaritmuojamas.

Grafinė analizė parodo, kad M1 gali būti nestacionarus kintamasis.



145 pav.: Realios Lietuvos M1 pinigų pasiūlos 2010 m. kainomis logaritmas

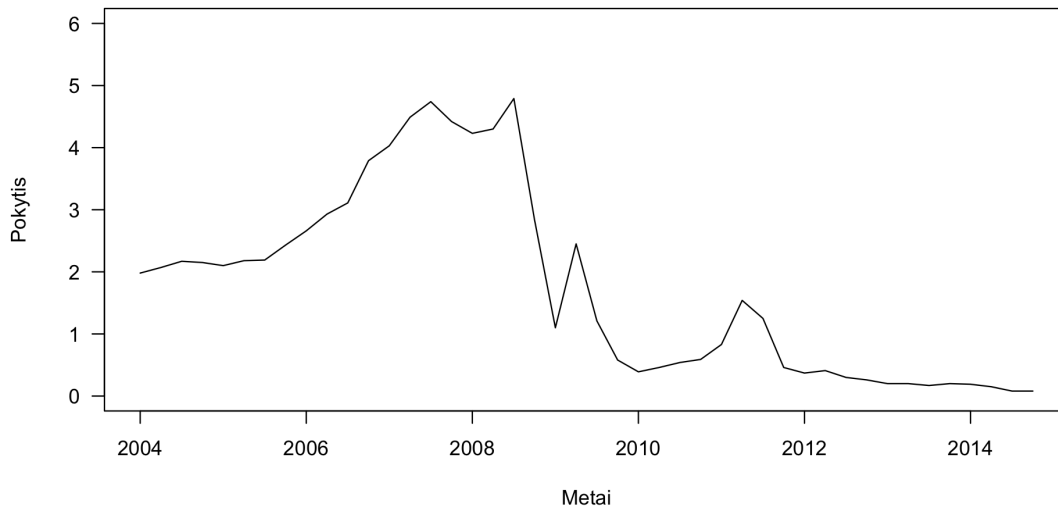


146 pav.: Diferencijuotas realios Lietuvos M1 pinigų pasiūlos 2010 m. kainomis logaritmas

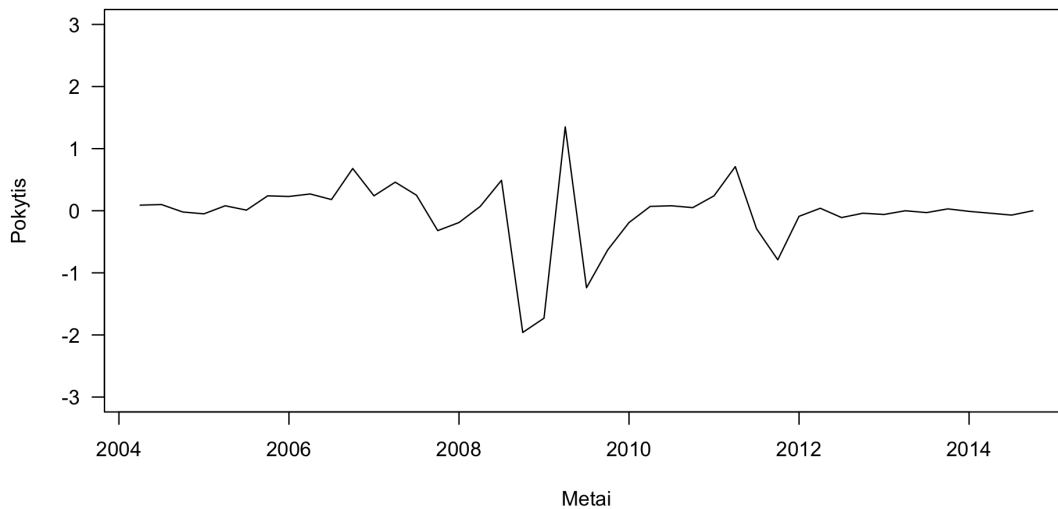
Formaliai stacionarumas patikrinamas naudojant papildytąją Dickey-Fuller testą. Naudojama antroji Dickey-Fuller lygtis. Papildytasis Dickey-Fuller testas parodė, kad negalime atmesti H_0 teigiančios, kad kintamasis turi vienetinę šaknį. KPSS testo pagalba patikrinsime ar kintamasis tikrai turi vienetinę šaknį. KPSS testo reikšmė yra tarp 10% ir 5% kritinės reikšmės. Šiuo atveju laikysime, kad kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi vienetinės šaknies. Diferencijuojant kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintamasis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad diferencijuotas kintamasis yra stacionarus ir kad kintamasis turi laisvąjį narį ir deterministinį trendą.

LIBOR palūkanų norma. LIBOR palūkanų norma yra pateikiama nemodifikuota.

Grafinė analizė parodo, kad LIBOR yra nestacionarus kintamasis.



147 pav.: LIBOR palūkanų norma



148 pav.: Diferencijuota LIBOR palūkanų norma

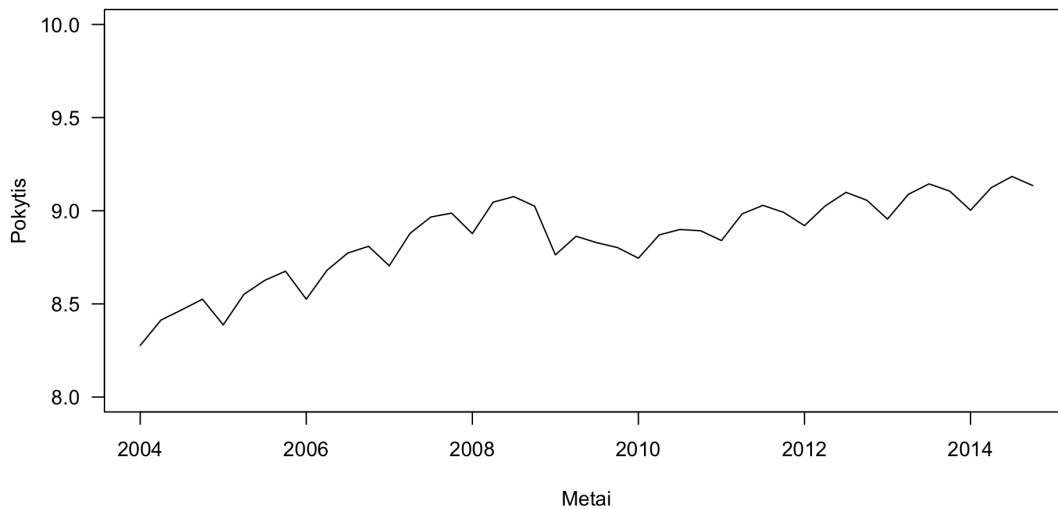
Formaliai stacionarumas patikrinamas naudojant papildytą Dickey-Fuller testą. Naudojama pirmoji Dickey-Fuller lygtis. Papildytasis Dickey-Fuller testas parodė, kad negalime atmesti H_0 teigiančios, kad kintamasis turi vienetinę šaknį. KPSS testo pagalba patikrinsime ar kintamasis tikrai turi vienetinę šaknį. KPSS testo reikšmė leidžia mums atmesti H_0 , todėl kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi vienetinės šaknies. Diferencijuotą kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintama-

sis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad diferencijuotas kintamasis yra stacionarus.

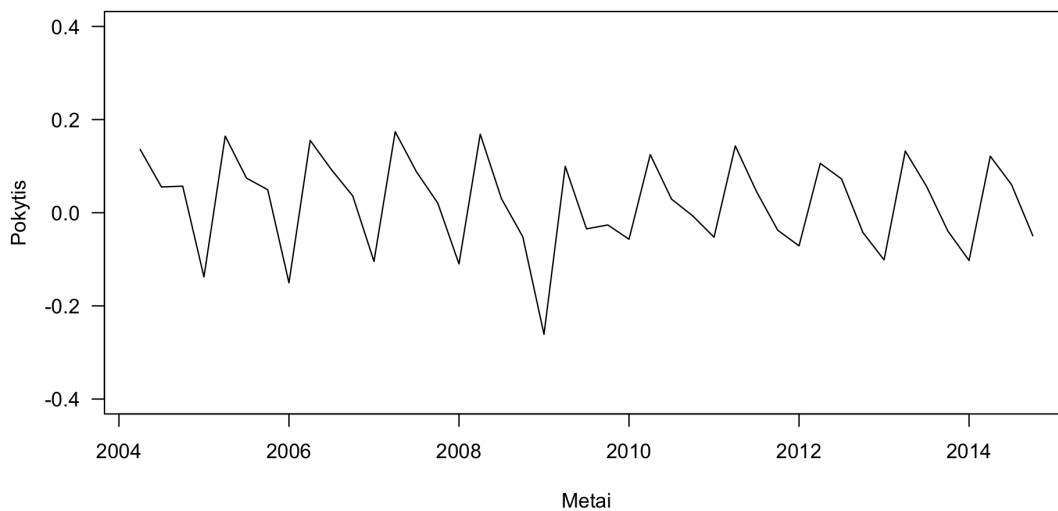
Grafinė kintamojo dinamikos analizė leidžia daryti prielaidą, kad kintamasis turi laisvąjį narį ir deterministinį trendą.

Realus Lietuvos BVP. Realus Lietuvos BVP yra suskaičiuojamas nominalų Lietuvos BVP padalijant iš atitinkamo periodo vartotojų kainų indekso. Tada kintamasis yra logaritmuojamas. Po to yra pašalinama sezoniškumo įtaka.

Grafinė analizė parodo, kad BVP gali būti stacionarus kintamasis su trendu ir struktūriniu lūžiu trende.



149 pav.: Realiojo Lietuvos BVP 2010 m. kainomis logaritmas



150 pav.: Diferencijuotas realusis Lietuvos BVP 2010 m. kainomis logaritmas

Formaliai integruotumas patikrinamas naudojant papildytąjį Dickey-Fuller testą. Papildytasis Dickey-Fuller testas parodė, kad kintamasis nėra stacionarus. Atlikus KPSS testą nustatyta, kad kintamasis yra integruotas. Integravimo eilei nustatyti kintamasis diferencijuojamas ir vėl tikrinama, ar jis neturi vienetinės šaknies. Diferenciuojant kintamąjį ir vėl atlikus ADF testą nustatyta, kad kintamasis yra integruotas pirma eile. Grafinė analizė taip pat parodo, kad diferencijuotas kintamasis yra stacionarus.

Grafinė kintamojo dinamikos analizė leidžia daryti prielaidą, kad kintamasis turi laisvąjį narį ir deterministinį trendą.

4.4.3. Modelio specifikacija

Siekiant nustatyti ar kintamieji yra kointegruoti pirma reikia sudaryti VECM modelį.

Vėlavimų eilės pasirinkimas. VECM eilės įvertinimui buvo apskaičiuoti šie informaciniai kriterijai: AIC, HQ, SC ir FPE. Šiuo atveju turime gerokai mažiau informacijos apie kintamuosius, nei tiriant JAV pinigų paklausą, todėl naudosime nustatant VECM su ilga vėlavimų eile galime prarasti labai daug laisvės laipsnių. Siekiant įvertinti skirtingai specifikuotus modelius įvertinami du modeliai: VECM(4) ir VECM(6).

Deterministinių komponentų pasirinkimas. Kadangi visi trys kintamieji turi laisvuosius narius ir deterministinius trendus daroma prielaida, kad VECM

modelis turi konstantą ir trendą kointegruojančiame komponente ir laisvąjį narį VAR dalyje.

Tiriamas VECM modelis. Lygtys (126) ir (129) parodo tiriamų VECM modelių specifikacijas.

$$y_t = \delta_1 + \delta_2 t + \sum_{p=1}^4 \alpha_p y_{t-p} + \sum_{p=1}^4 \beta_p r_{t-p} + \sum_{p=1}^4 \gamma_p m_{t-p} + u_t \quad (126)$$

$$r_t = \delta_3 + \delta_4 t + \sum_{p=1}^4 a_p r_{t-p} + \sum_{p=1}^4 b_p y_{t-p} + \sum_{p=1}^4 c_p m_{t-p} + v_t \quad (127)$$

$$m_t = \delta_5 + \delta_6 t + \sum_{p=1}^4 \rho_p r_{t-p} + \sum_{p=1}^4 \zeta_p y_{t-p} + \sum_{p=1}^4 \eta_p m_{t-p} + w_t \quad (128)$$

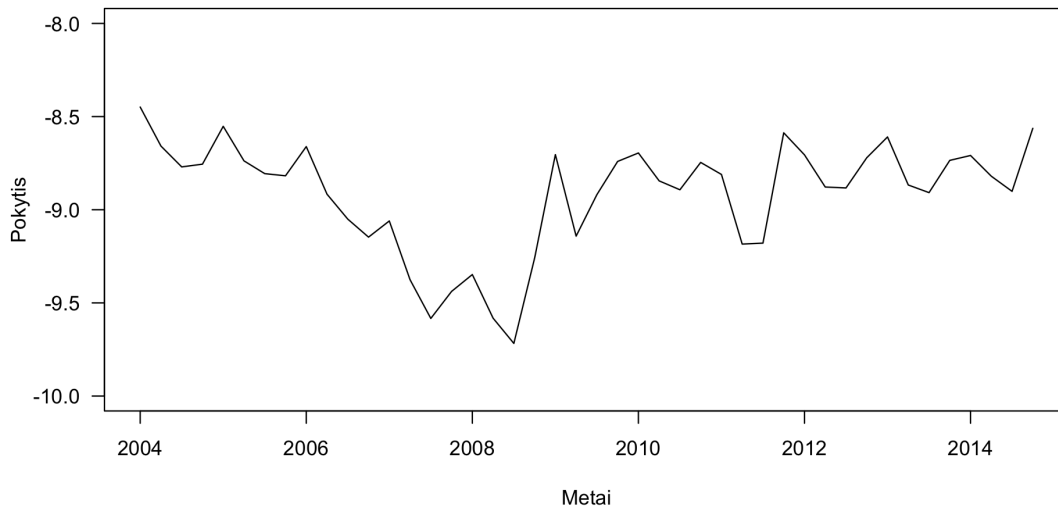
$$y_t = \delta_1 + \delta_2 t + \sum_{p=1}^6 \alpha_p y_{t-p} + \sum_{p=1}^6 \beta_p r_{t-p} + \sum_{p=1}^6 \gamma_p m_{t-p} + u_t \quad (129)$$

$$r_t = \delta_3 + \delta_4 t + \sum_{p=1}^6 a_p r_{t-p} + \sum_{p=1}^6 b_p y_{t-p} + \sum_{p=1}^6 c_p m_{t-p} + v_t \quad (130)$$

$$m_t = \delta_5 + \delta_6 t + \sum_{p=1}^6 \rho_p r_{t-p} + \sum_{p=1}^6 \zeta_p y_{t-p} + \sum_{p=1}^6 \eta_p m_{t-p} + w_t \quad (131)$$

4.4.4. Rezultatų palyginimas

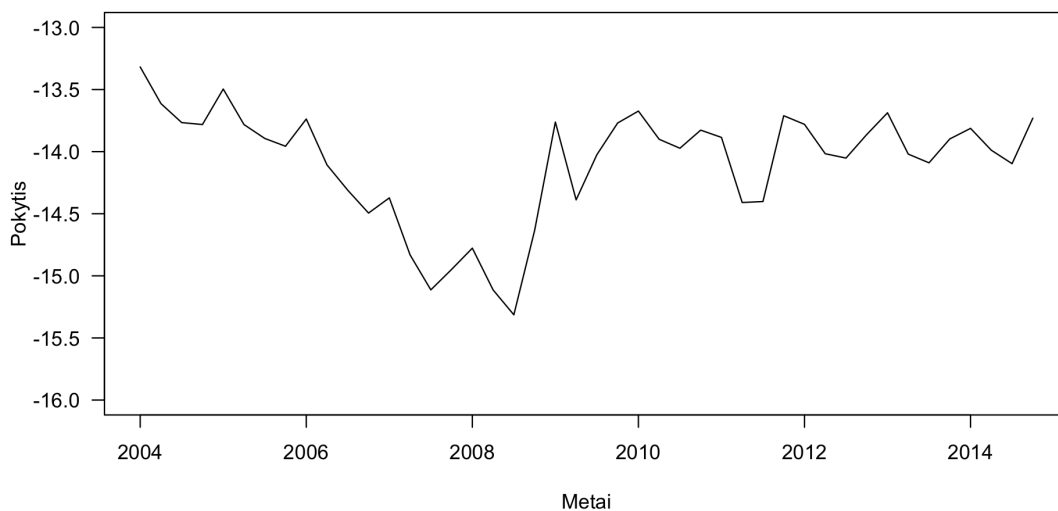
Johansen procedūra be deterministinių komponentų. Įvertinus VECM(4) naudojant Johansen procedūrą be deterministinių komponentų pagal maksimalios tikrinės reikšmės statistiką galime atmesti H_0 teigiančią, kad $rk(\Pi) = 0$. Pėdsako testas nurodo, kad kintamieji nėra kointegruoti.



151 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 151 parodo VECM(4) modelio paklaidos korekcijos komponentą. Akivaizdu, kad šis komponentas nėra stacionarus.

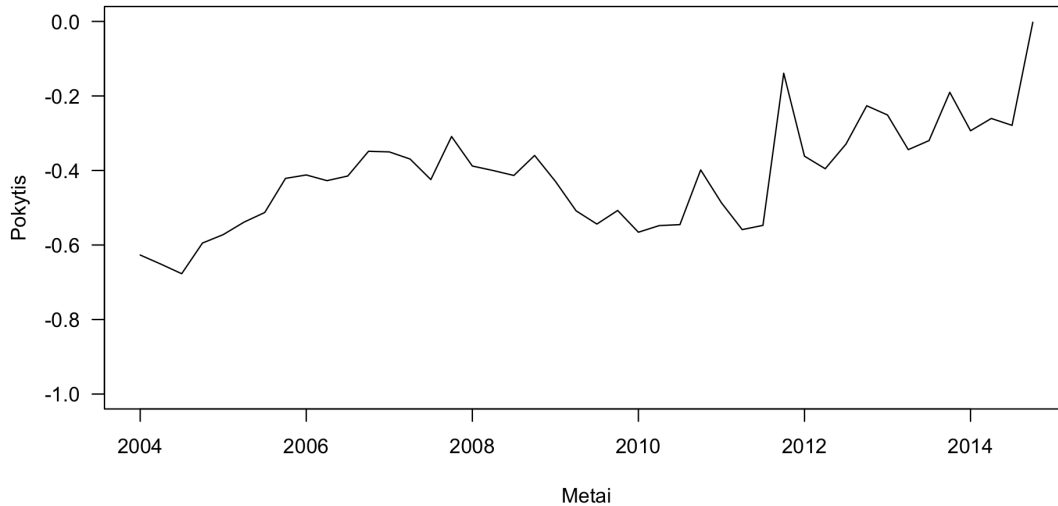
Įvertinus VECM(6) naudojant Johansen procedūrą be deterministinių komponentų tiek pėdsako tiek maksimalios tikrinės reikšmės statistikos rodo, kad $rk(\Pi) = 0$.



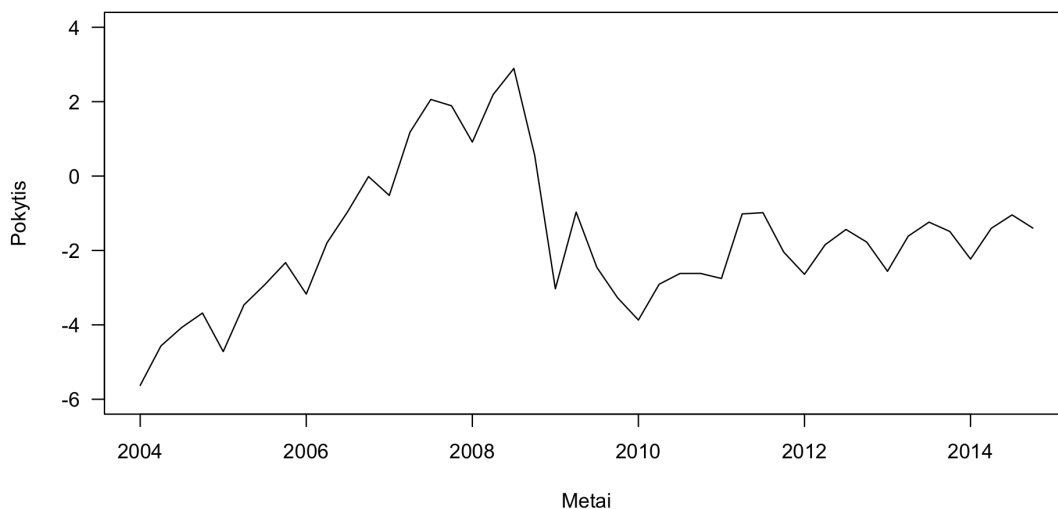
152 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 152 parodo VECM(6) modelio paklaidos korekcijos komponentą. Tikėtina, kad šis komponentas nėra stacionarus.

Johansen procedūra su laisvuju nariu. Įvertinus VECM(4) naudojant Johansen procedūrą su laisvuju nariu tiek pėdsako, tiek ir maksimalios tikrinės reikšmės statistikos indikuoja, kad $rk(\Pi) = 2$ t.y. yra du kointegruojantys vektoriai. Įvertinus VECM(6) naudojant Johansen procedūrą su laisvuju nariu gaunami analogiški rezultatai: $rk(\Pi) = 2$ ir du kointegruojantys vektoriai.



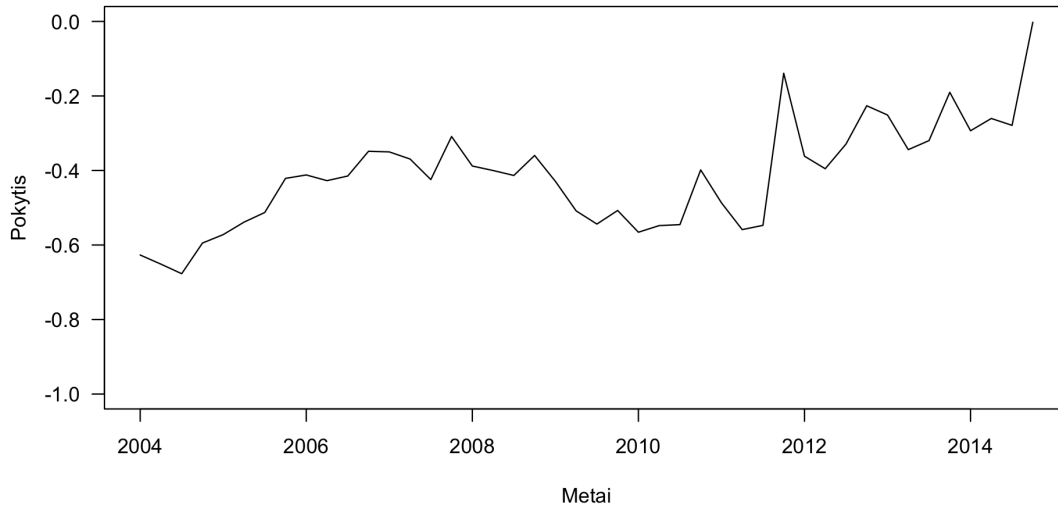
153 pav.: Paklaidos korekcijos komponentas



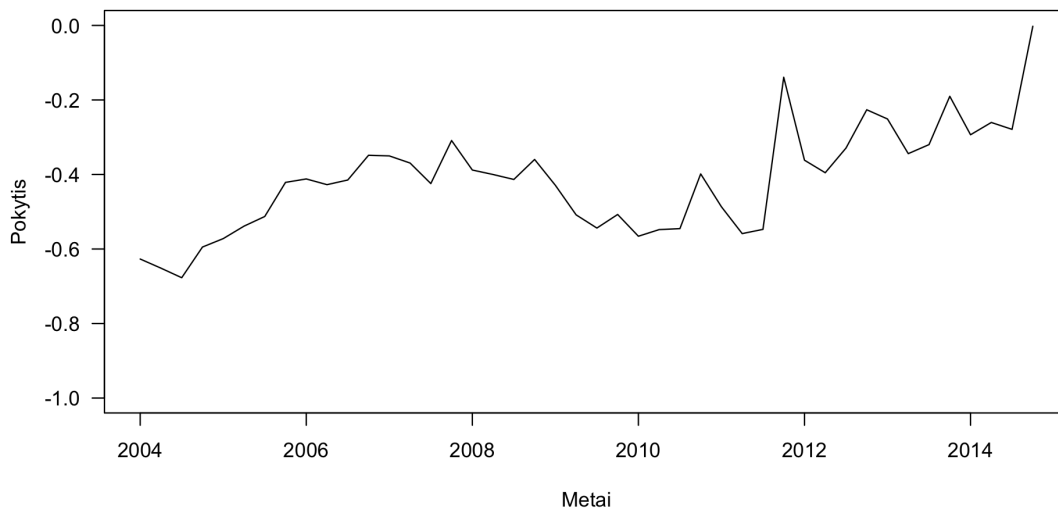
154 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 153 parodo VECM(4) modelio paklaidos korekcijos komponentą sudarytą su pirmuoju kointegruojančiu vektoriumi, paveikslėlis 154 su antruoju koin-

tegruojančiu vektoriumi.



155 pav.: Paklaidos korekcijos komponentas

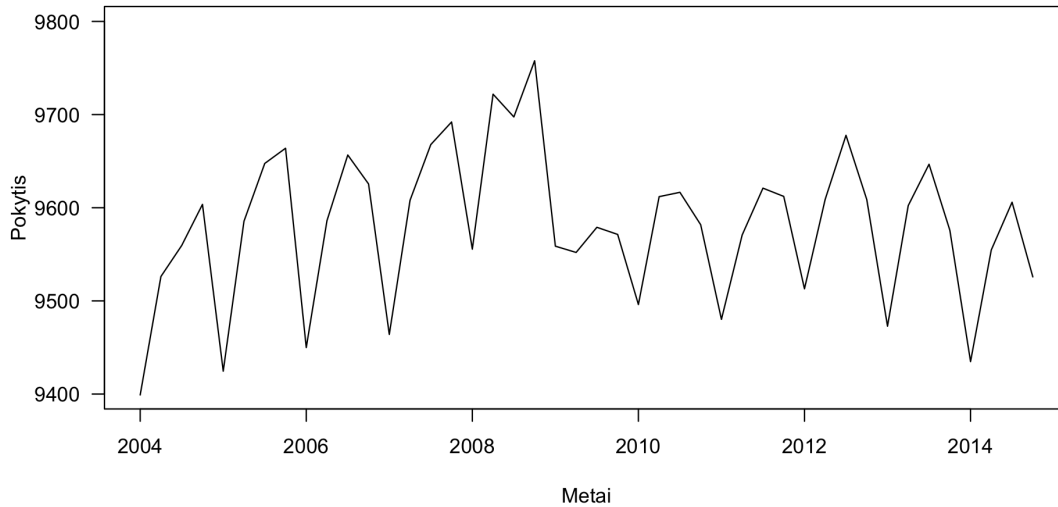


156 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 155 parodo VECM(6) modelio paklaidos korekcijos komponentą sudarytą su pirmuoju kointegruojančiu vektoriumi, paveikslėlis 156 su antruoju kointegruojančiu vektoriumi.

Johansen procedūra su trendu. Įvertinus VECM(4) naudojant Johansen procedūrą su trendu tiek pagal maksimalios tikrinės reikšmės, tiek pagal pėdsako

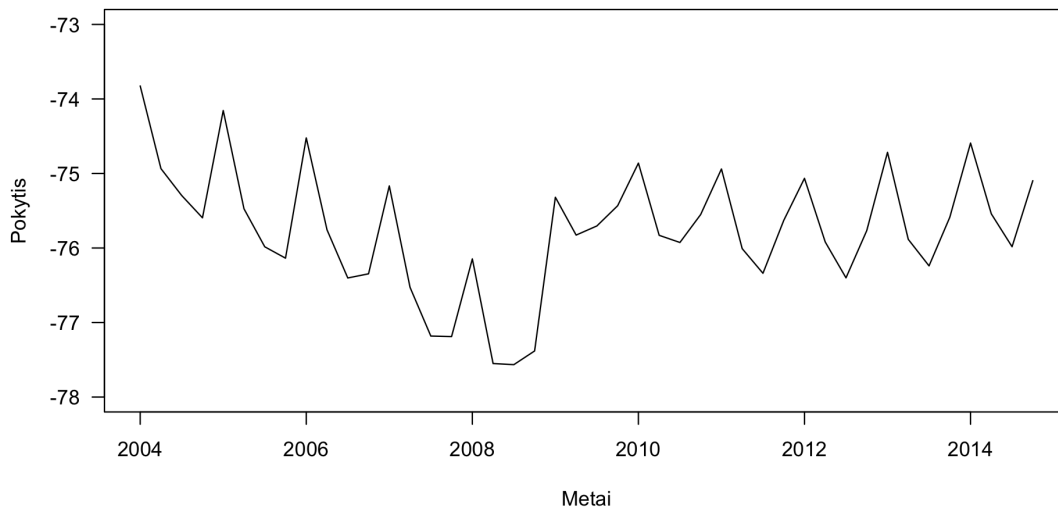
statistikas galime atmesti H_0 teigiančią, kad $rk(\Pi) = 0$ ir priimti H_1 , kuri teigia, kad kintamieji turi vieną kointegruojantį vektorių.



157 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 157 parodo VECM(4) modelio paklaidos korekcijos komponentą.

Įvertinus VECM(6) naudojant Johansen procedūrą su trendu naudojantis maksimalios tikrinės ir pėdsako statistikos testais negalime atmesti H_0 teigiančios, kad $rk(\Pi) = 0$.



158 pav.: Paklaidos korekcijos komponentas

Paveikslėlis 158 parodo VECM(6) modelio paklaidos korekcijos komponentą.

Formaliai patikrinus šį komponentą naudojant ADF testą nustatyta, kad jis yra stacionarus.

VECM specifikuojamo įtaka modelio įvertinimui. Įvertinus VECM(4) ir VECM(6) modelius naudojant tris skirtingai specifikuotas Johansen procedūras gauti labai skirtingi rezultatai. VECM(4) modeliui tiek Johansen procedūros be deterministinių komponentų, tiek su trendo komponentu nurodė 1 kointegruojantį vektorių. Procedūra su laisvuju nariu nurodė du kointegruojančius vektorius. Atlikus paklaidos korekcijos komponento grafinę analizę galima teigti, kad modelis turi 1 kointegruojantį vektorių. VECM(6) modeliui tiek Johansen procedūros be deterministinių komponentų, tiek su trendo komponentu nurodė, kad kintamieji nėra kointegruoti. Johansen procedūra su laisvuju nariu nurodė du kointegruojančius vektorius. Kadangi imtis yra labai maža VECM(6) modelis yra laikomas pertekliniu labai sumažinančiu laisvės laipsnių skaičių ir tuo pačiu Johansen procedūrų galia, todėl lygtyje (126) aprašytas VECM(4) modelis su trendu ir laisvuju nariu yra laikomas teisingai specifikuotu. Atižvelgiant į tyrimo rezultatus galima teigti, kad kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių. Johansen procedūra su trendu parametrizuoja β vektorių taip, kad paklaidos korekcijos komponentas yra stacionarus. Tokie rezultatai suponuoja, kad Johansen procedūra su laisvuju nariu padaro I rūšies klaidą, tuo tarpu Johansen procedūra be deterministinių komponentų ir Johansen procedūra su trendu teisingai specifikuoja VECM modelį. Pirmos rūšies klaidos tikimybė mažose ir vidutinėse imtyse modeliuose turinčiuose vieną kointegruojantį vektorių su laisvojo nario ir trendo deterministiniais komponentais yra vidutinė - tarp 20% ir 30%, tačiau liekanti šiame intervale didėjant imties dydžiui.

4.5. Pinigų paklausos analizės išvados

Ištyrus Lietuvos ir JAV pinigų paklausos modelius naudojant įvairiai specifikuotas Johansen procedūras nustatyta, kad teisinga deterministinių komponentų specifikacija šiose procedūrose lemia rezultatus tiek labai didelėse imtyse (JAV atvejis), tiek vidutinėse imtyse (Lietuvos atvejis). Nustatyta, kad JAV pinigų paklausos modelis nėra kointegruotas, tuo tarpu Lietuvos pinigų paklausos modelis yra kointegruotas ir turi vieną kointegruojantį vektorių. Abiem atvejais VECM modeliai buvo specifikuoti su laisvuju nariu ir trendu paklaidos korekcijos komponente ir konstanta VAR komponente. 3 skyriuje atlikta sintetinių modelių analizė padėjo sprendžiant apie kointegruojančių vektorių skaičių, kai skirtingai specifikuotos Johansen procedūros pateikė skirtingus rezultatus.

Išvados ir pasiūlymai

Tyrimų išvados.

1. Išanalizavus mokslinę literatūrą nestacionarių: integruotų ir kointegruotų kintamųjų analizės temomis nustatyta, kad deterministinių komponentų vaidmuo modeliuojant kointegruotus kintamuosius nėra vienareikšmiškas. Kai kintamieji turi stochastinį trendą ir deterministinius komponentus: laisvuosius narius, trendus, ir fiktyvius kintamuosius, kointegruotų kintamųjų modeliai tampa sunkiau įvertinami.
2. VECM modelių mokslinės literatūros analizė parodė, kad VECM modeliuose kointegruojančių vektorių skaičiaus nustatymas yra paskutinis žingsnis VECM specifikacijos procedūrose, kai kiti modelio elementai yra žinomi. Rekomenduojama tokia VECM specifikavimo procedūros eiga:
 - (a) Nustatyti kintamųjų integruotumo eilę.
 - (b) Nustatyti modelio vėlavimų eilę.
 - (c) Pasirinkti deterministinių komponentų įtraukimo į modelį būdą.
 - (d) Nustatyti kointegruojančių vektorių skaičių.
3. Ištyrus vektorinių paklaidų korekcijos modelių įvertinimo procedūras buvo nustatyta, kad Johansen maksimalaus tikėtinumo procedūra teoriniu aspektu yra pranašesnė už Engle - Granger dviejų pakopų įvertinimo procedūrą. Mokslinė literatūra palyginanti šias dvi procedūras taip pat pranašesne laiko Johansen maksimalaus tikėtinumo procedūrą. Kitos įvertinimo procedūros yra mažiau populiaros, mokslinėje literatūroje ir praktikoje vartojamos retai.
4. Johansen procedūrų specifikacijos analizė naudojant Monte Carlo simuliacijas parodė, kad VECM modeliuose esantys deterministiniai komponentai daro nevienareikšmišką įtaką Johansen procedūrų rezultatams: priklausomai nuo imties dydžio taip pat specifikuotam modeliui tinkamiausios gali būti skirtingos Johansen procedūros. Remiantis šių tyrimų rezultatais buvo sudarytos rekomendacijos, kurią Johansen procedūros specifikaciją pasirinkti, priklausomai nuo VECM modelio deterministinių komponentų ir imties dydžio.
5. Monte Carlo simuliacijų analizė parodė, kad Johansen procedūros naudojančios maksimalios tikrinės reikšmės testus nustatant kointegruojančių vektorių skaičių beveik visada yra pranašesnės už pėdsako statistiką naudojančias Johansen procedūras. Išimtys yra tik tais atvejais, kai įvertinami VECM modeliai neturintys deterministinių komponentų naudojant deterministinių komponentų neturinčias Johansen procedūras labai mažose ir vidutinėse imtyse.

6. Sintetinių VECM modelių analizė parodė, kad Johansen procedūrų galia yra labai maža, kai imtys yra mažos. Taip pat buvo nustatyta, kad deterministinių komponentų įtraukimas į Johansen procedūras labai smarkiai sumažina jų galią: Johansen procedūros su laisvaisiais nariais ir Johansen procedūros su deterministiniais trendais daug dažniau daro II rūšies klaidą, kai imtys yra mažos arba vidutinės. Didėjant imties dydžiui procedūrų galia konverguoja ir labai didelėse imtyse visų tipų procedūros teisingai nustato kointegruojančių vektorių skaičių maždaug 95% atveju.
7. Johansen procedūrų specifikacijos analizė naudojant Monte Carlo simuliacijas parodė, kad dėl nevienodo imties dydžio įtakos įvairiai specifiкуotų Johansen procedūrų galiai tam tikrais atvejais geriausia naudoti misspecifikuotas Johansen procedūras, įtraukiant per mažai deterministinių komponentų, siekiant nustatyti teisingą kointegruojančių vektorių skaičių ir naudoti teisingai specifiкуotas procedūras įvertinant modelio parametrų reikšmes. Sudaryta rekomenduojamų Johansen procedūrų pasirinkimo, priklausomai nuo imties dydžio ir VECM specifikacijos, lentelė.
8. JAV pinigų paklausos analizėje nustatyta, kad tinkamos Johansen procedūros specifikacijos pasirinkimas turi lemiamą įtaką nustatant kointegruojančių vektorių skaičių. Neteisingai specifiкуotos procedūros nustatė, kad kintamieji yra kointegruoti ir turi vieną kointegruojantį vektorių, tuo tarpu teisingai specifiкуotos procedūros nustatė, kad kintamieji nėra kointegruoti. Papildoma paklaidos korekcijos komponentų analizė atskleidė, kad kintamieji nėra kointegruoti ir neteisingai specifiкуotos Johansen procedūros padarė I rūšies klaidą.
9. Lietuvos pinigų paklausos analizėje, gautos tokios pat išvados kaip ir JAV pinigų paklausos analizėje - deterministinių komponentų specifikacija Johansen procedūroje turi lemtingą įtaką nustatant kointegruojančių vektorių skaičių. Johansen procedūros su laisvaisiais nariais nustatė, kad kintamieji yra kointegruoti ir turi du kointegruojančius vektorius, tačiau Monte Carlo simuliacijų metu gautos išvados panaudotos įvertinant I rūšies klaidos tikimybę ir nustatyta, kad kintamieji turi vieną kointegruojantį vektorių. Teisingai specifiкуotos Johansen procedūros nustatė teisingą kointegruojančių vektorių skaičių.

Tolesnių tyrimų kryptys ir pasiūlymai. Šiame darbe atlikti tyrimai gali būti praplėsti ir pritaikyti platesniam panaudojimui, atlikus papildomus kointegruotų kintamųjų tyrimus. Siekiant detaliau ištirti deterministinių komponentų įtaką kointegruotų kintamųjų modelių įvertinimo procedūroms būtų galima ištirti:

1. Keturmačius VECM modelius. Siekiant įvertinti papildomos dimensijos įtaką dabartinių tyrimų rezultatams galima tirti keturmačius VECM modelius. Daugeliu atveju makroekonominių kintamųjų modeliai yra keturmačiai, o trimatųjų modelių išvados nebūtinai galios keturių dimensijų modeliams.
2. VECM vėlavimų eilės įtaką dabartinių tyrimų rezultatams. Galima tirti dabartinio modelio rezultatus priklausomai nuo modelio vėlavimų eilės. Dabartiniame tyrime visi modeliai buvo ketvirtos eilės, kuri buvo pasirinkta dėl ketvirtinių duomenų populiarumo makroekonominiuose tyrimuose, tačiau yra daug atvejų, kai reikia aštunto, dvyliktos arba šešioliktos eilės modelių.
3. Fiktyvių kintamųjų įtaukimą į VECM modelius. Šiame tyrime nebuvo tiriami VECM modeliai su fiktyviais kintamaisiais. Šį tyrimą galima būtų praplėsti įtraukiant į dabartinius VECM modelius impulso ir poslinkio fiktyvius kintamuosius.
4. Lütkepohl - Saikkonen - Trenkler testą. Dabartinį tyrimą galima praplėsti tiriant kaip kointegracijos rangą nustato Lütkepohl - Saikkonen - Trenkler testas. Šis testas yra skirtas tirti VECM modeliams su struktūriniais lūžiais, todėl puikiai tinka tiriant modelius su fiktyviais kintamaisiais. Taip pat svarbu nustatyti šio testo elgseną, kai modelyje iš tikrųjų fiktyvių kintamųjų nėra.

Literatūros sąrašas

- [1] Charleen Adam ir Scott Hendry. The m1 vector-error-correction model: Some extensions and applications. *Money, Monetary Policy, and Transmission Mechanisms*, pages 151–80, 2000.
- [2] Francis W Ahking. Model mis-specification and johansen's co-integration analysis: an application to the us money demand. *Journal of Macroeconomics*, 24 (1):51–66, 2002.
- [3] Richard G Anderson, Dennis L Hoffman, ir Robert H Rasche. A vector error-correction forecasting model of the us economy. *Journal of Macroeconomics*, 24 (4):569–598, 2002.
- [4] Dimitrios Asteriou ir Stephen G Hall. *Applied econometrics: A modern approach using eviews and microfit revised edition*. Palgrave Macmillan, 2007.
- [5] Anindya Banerjee, Juan J Dolado, David F Hendry, ir Gregor W Smith. Exploring equilibrium relationships in econometrics through static models: Some monte carlo evidence*. *Oxford Bulletin of Economics and Statistics*, 48(3): 253–277, 1986.
- [6] Anindya Banerjee, Juan J Dolado, John W Galbraith, ir David Hendry. Co-integration, error correction, and the econometric analysis of non-stationary data. *OUP Catalogue*, 1993.
- [7] Peter Bossaerts. Common nonstationary components of asset prices. *Journal of Economic Dynamics and Control*, 12(2):347–364, 1988.
- [8] H Peter Boswijk ir André Lucas. Semi-nonparametric cointegration testing. *Journal of econometrics*, 108(2):253–280, 2002.
- [9] G.E.P. Box ir G.M. Jenkins. *Time Series Analysis Forecasting and Control*. Holden-Day, San Francisco, 1970.
- [10] Jörg Breitung ir Uwe Hassler. Inference on the cointegration rank in fractionally integrated processes. *Journal of Econometrics*, 110(2):167–185, 2002.
- [11] Ralf Brüggemann, Helmut Lütkepohl, ir Pentti Saikkonen. Residual autocorrelation testing for vector error correction models. *Journal of Econometrics*, 134 (2):579–604, 2006.
- [12] Richard Clarida, Jordi Gali, ir Mark Gertler. Monetary policy rules and macroeconomic stability: Evidence and some theory. *Quarterly Journal of Economics*, 115:147–180, 2000.

- [13] James EH Davidson, David F Hendry, Frank Srba, ir Stephen Yeo. Econometric modelling of the aggregate time-series relationship between consumers' expenditure and income in the united kingdom. *The Economic Journal*, pages 661–692, 1978.
- [14] David A Dickey ir Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.
- [15] David A Dickey ir Wayne A Fuller. Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica: Journal of the Econometric Society*, pages 1057–1072, 1981.
- [16] David Alan Dickey. Estimation and hypothesis testing in nonstationary time series. 1976.
- [17] Jurgen A. Doornik, David F. Hendry, ir Bent Nielsen. Inference in cointegrating models: Uk m1 revisited. *Journal of economic surveys*, 12(5):533–572, 1998.
- [18] Graham Elliott. Efficient tests for a unit root when the initial observation is drawn from its unconditional distribution. *International Economic Review*, 40(3):767–784, 1999.
- [19] Graham Elliott, Thomas J Rothenberg, ir James H Stock. Efficient tests for an autoregressive unit root, 1992.
- [20] Walter Enders. *Applied econometric time series*. John Wiley & Sons, 2008.
- [21] Walter Enders. *Applied econometric time series*. John Wiley & Sons, 2010.
- [22] RF Engle ir CWH Granger. Dynamic model specification with equilibrium constraints: Cointegration and error corrections, paper presented to the econometric society fifth world congress. *MIT*, 306:277–304, 1985.
- [23] Robert F Engle ir Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276, 1987.
- [24] Wayne A Fuller. *Introduction to statistical time series*. John Wiley, 1976.
- [25] Jesus Gonzalo ir Tae-Hwy Lee. Pitfalls in testing for long run relationships. *Journal of Econometrics*, 86(1):129–154, 1998.
- [26] Clive WJ Granger. Some properties of time series data and their use in econometric model specification. *Journal of econometrics*, 16(1):121–130, 1981.

- [27] Clive WJ Granger ir Paul Newbold. Spurious regressions in econometrics. *Journal of econometrics*, 2(2):111–120, 1974.
- [28] Clive WJ Granger ir Andrew A Weiss. Time series analysis of error-correction models. *Studies in Econometrics, Time Series, and Multivariate Statistics*, pages 255–278, 1983.
- [29] Frank E Harrell, with contributions from Charles Dupont, ir many others. *Hmisc: Harrell Miscellaneous*, 2015. URL <http://CRAN.R-project.org/package=Hmisc>. R package version 3.17-0.
- [30] David F Hendry. Econometric modelling with cointegrated variables: an overview. *Oxford Bulletin of Economics and Statistics*, 48(3):201–212, 1986.
- [31] David F Hendry ir Katarina Juselius. Explaining cointegration analysis: Part i. *The Energy Journal*, pages 1–42, 2000.
- [32] David F Hendry ir Katarina Juselius. Explaining cointegration analysis: Part ii. *The Energy Journal*, pages 75–120, 2001.
- [33] Kirstin Hubrich, Helmut Lütkepohl, ir Pentti Saikkonen. A review of systems cointegration tests. *Econometric Reviews*, 20(3):247–318, 2001.
- [34] Søren Johansen. Statistical analysis of cointegration vectors. *Journal of economic dynamics and control*, 12(2):231–254, 1988.
- [35] Søren Johansen. The role of the constant and linear terms in cointegration analysis of nonstationary variables. *Econometric reviews*, 13(2):205–229, 1994.
- [36] Soren Johansen. Likelihood-based inference in cointegrated vector autoregressive models. *OUP Catalogue*, 1995.
- [37] Søren Johansen ir Katarina Juselius. Maximum likelihood estimation and inference on cointegration—with applications to the demand for money. *Oxford Bulletin of Economics and statistics*, 52(2):169–210, 1990.
- [38] Søren Johansen, Rocco Mosconi, ir Bent Nielsen. Cointegration analysis in the presence of structural breaks in the deterministic trend. *The Econometrics Journal*, 3(2):216–249, 2000.
- [39] Katarina Juselius. *The cointegrated VAR model: methodology and applications*. Oxford University Press, 2006.

- [40] Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, ir Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1):159–178, 1992.
- [41] RE Lucas. On the mechanics of economic development. *ECONOMETRIC SOCIETY MONOGRAPHS*, 29:61–70, 1998.
- [42] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer, 2007.
- [43] Helmut Lütkepohl, Pentti Saikkonen, ir Carsten Trenkler. Testing for the cointegrating rank of a var process with level shift at unknown time. *Econometrica*, pages 647–662, 2004.
- [44] Gangadharrao S Maddala ir In-Moo Kim. *Unit roots, cointegration, and structural change*. Number 4. Cambridge University Press, 1998.
- [45] Serena Ng ir Pierre Perron. Estimation and inference in nearly unbalanced nearly cointegrated systems. *Journal of Econometrics*, 79(1):53–81, 1997.
- [46] Jostein Paulsen. Order determination of multivariate autoregressive time series with unit roots. *Journal of time series analysis*, 5(2):115–127, 1984.
- [47] Pierre Perron. Trends and random walks in macroeconomic time series: Further evidence from a new approach. *Journal of economic dynamics and control*, 12(2):297–332, 1988.
- [48] B. Pfaff. *Analysis of Integrated and Cointegrated Time Series with R*. Springer, New York, second edition, 2008. URL <http://www.pfaffikus.de>. ISBN 0-387-27960-1.
- [49] Bernhard Pfaff. Var, svar and svec models: Implementation within R package vars. *Journal of Statistical Software*, 27(4), 2008. URL <http://www.jstatsoft.org/v27/i04/>.
- [50] Peter CB Phillips. Error correction and long-run equilibrium in continuous time. *Econometrica: Journal of the Econometric Society*, pages 967–980, 1991.
- [51] Peter CB Phillips ir Pierre Perron. Testing for a unit root in time series regression. *Biometrika*, 75(2):335–346, 1988.
- [52] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL <http://www.R-project.org/>.

- [53] Pentti Saikkonen. Stability results for nonlinear error correction models. *Journal of Econometrics*, 127(1):69–81, 2005.
- [54] Mark Salmon. Error correction mechanisms. *The Economic Journal*, pages 615–629, 1982.
- [55] Airs Spanos. Statistical foundation of economic modelling, 1986.
- [56] James H Stock. Asymptotic properties of least squares estimators of cointegrating vectors. *Econometrica: Journal of the Econometric Society*, pages 1035–1056, 1987.
- [57] James H Stock ir Mark W Watson. Testing for common trends. *Journal of the American statistical Association*, 83(404):1097–1107, 1988.
- [58] R.S. Tsay. *Multivariate Time Series Analysis: With R and Financial Applications*. Wiley Series in Probability and Statistics. Wiley, 2013. ISBN 9781118617755. URL <https://books.google.lt/books?id=A4QVAgAAQBAJ>.
- [59] Ruey S Tsay. Order selection in nonstationary autoregressive models. *The Annals of Statistics*, pages 1425–1433, 1984.
- [60] G Udny Yule. Why do we sometimes get nonsense-correlations between time-series?—a study in sampling and the nature of time-series. *Journal of the royal statistical society*, pages 1–63, 1926.
- [61] Eric Zivot ir DW Andrews. Further evidence on the great crash, the oil-price shock, and the unit-root. *Journal of Business & Economic Statistics*, 10(0):3, 1992.

THE ROLE OF DETERMINISTIC COMPONENTS WHILE MODELING COINTEGRATED VARIABLES

Dovydas Čeilutka

Master Thesis

Economic Analysis and Planning Programme

Vilnius University, Faculty of Economics, Management Department

Supervisor – doc. dr. A. Bartkus

Vilnius, 2016

SUMMARY

162 pages, 158 pictures, 4 tables, 61 references.

Research problem. The research problem - there are multiple ways to specify the Johansen procedure to estimate the VECM models, however the impact of the specification of the procedure to the estimation results is not clear. The questions like what is the impact of the misspecification to the number of cointegrated vectors of the VECM model; in which cases the Johansen procedure should include deterministic terms; what is the impact of deterministic terms in relation to the sample size; how should procedure of specifying the VECM models should look like? In this thesis synthetic and real VECM models will be estimated using different specifications of Johansen procedure.

Relevance of the thesis. Deterministic component inclusion to the cointegrated models is critically important in estimating macroeconomic variables. Most of the macroeconomic variables have deterministic terms and misspecification provides biased estimates and incorrect estimation of cointegrated relationships. In order to correctly estimate VECM models and use them in forecasting it is necessary to correctly specify the VECMs and choose appropriate estimation procedures, take into account misspecification problems and the power of the cointegration tests.

Object of the research. The object of the thesis is the specification of the Johansen procedures including various deterministic terms, estimating the VECM models with deterministic components, the impact of the choice of the appropriate VECM estimation procedure and estimation of VECM parameters based on sample size.

Aim and objectives of the thesis. The aim of the research is to use simulated VECM models with various deterministic terms in order to estimate the results of the Johansen procedures based on deterministic terms and sample size and provide suggestions on the choice of Johansen procedure for different VECM models. In order to achieve the aim these objectives were completed:

1. Analyze scientific studies on nonstationary (integrated and cointegrated) processes.
2. Analyze VECM specification and estimation procedures.
3. Simulate VECM models using Monte Carlo method and estimate them using Johansen procedures.
4. Analyze the impact of Johansen procedure specification on VECM estimation while modeling the money demand of USA and Lithuania.
5. Provide conclusions regarding the impact of deterministic terms while modeling cointegrated variables.

General conclusions and suggestions.

1. Analysis of the current papers on cointegration suggests VECM models are the most popular models for specifying the relations between potentially cointegrated variables while the Johansen procedure is the main tool for estimating the VECM models.
2. The Monte Carlo study suggests that deterministic terms in the VECM models play a major role on the number of cointegrated vectors estimated by Johansen procedures.
3. The analysis of sintetic VECMs proves that the role of deterministic components varies based on the sample size: deterministic terms have much larger impact on the power of Johansen procedures if the samples are small and in very large sampls it makes virtually no difference.

4. The Monte Carlo study proves that the Johansen procedure is very weak when samples are small. Also the power of the procedure decreases as more deterministic terms are added to the procedure.
5. It was determined that in some cases it is optimal to use misspecified Johansen procedures, with less deterministic terms, because of the combined impact of sample size and deterministic terms renders correctly specified procedures extremely weak. This only applies to determining the cointegration rank - the parameters of the VECM models should then be estimated using the properly specified Johansen procedures.
6. The analysis of the money demand of the US determined that correct specification of the Johansen procedure makes a difference while estimating the number of cointegrated vectors. Misspecified procedures found one cointegrated vector, while correctly specified procedures determined that the variables are not cointegrated.
7. The analysis of the money demand of Lithuania showed that, as in the case of the US, correct specification of the Johansen procedure makes a difference while estimating the number of cointegrated vectors. Misspecified procedures found two cointegrated vectors, while correctly specified procedures found one cointegrated vector.

Structure of the thesis. This master thesis consists of six major sections:

1. Introduction. In the introduction the information about the relevance of the thesis, the object of the research and the aim and the objectives of the thesis are stated.
2. Analysis of non-stationary processes. In this section the scientific papers regarding the nonstationary processes are analysed: the importance of stationarity is stated, the integrated and cointegrated processes are compared.
3. VECM models. In this section of the thesis the specification and estimation of the VECM models are analysed.
4. The analysis of specification of Johansen procedures using Monte Carlo method. In this section three synthetic VECM models are analysed and the results of their estimation taking into account the sample size by various Johansen procedures are provided.
5. The analysis of money demand of the US and Lithuania. In this section of the thesis several models of money demand are estimated by various types

of Johansen procedures. The results are compared using the insights from the Monte Carlo study and then the final models are constructed for money demand for each country.

6. Conclusions and suggestions. In this section the conclusions and suggestions are provided.

Priedas Nr. 1.

Sintetinių VECM tyrimams naudotas R kodas

```
#####
# Bibliotekos
#####

install.packages('vars')
install.packages('urca')
install.packages('graphics')
install.packages('Hmisc')
install.packages('xlsx')

library(vars)
library(urca)
library(graphics)
library(Hmisc)
library(xlsx)

toBibtex(citation('vars'))
toBibtex(citation('urca'))
toBibtex(citation('graphics'))
toBibtex(citation('Hmisc'))

#####
# Funkcijos
#####

source("../R/dgp.r") # Duomenis generuojanti funkcija
source("../R/money.r")
source("../R/export.r")
source("../R/defaults.r")

#####
# j.test funkcija
#####

j.test <- function (ss, args) {
  args$ss <- ss
  x <- dgp(args)
  t1 <- ca.jo(x$vecm, type="trace", K = args$arima[1], ecdet="none")
  t2 <- ca.jo(x$vecm, type="trace", K = args$arima[1], ecdet="const")
  t3 <- ca.jo(x$vecm, type="trace", K = args$arima[1], ecdet="trend")
  e1 <- ca.jo(x$vecm, type="eigen", K = args$arima[1], ecdet="none")
  e2 <- ca.jo(x$vecm, type="eigen", K = args$arima[1], ecdet="const")
}
```

```

e3 <- ca.jo(x$vecm, type="eigen", K = args$arima[1], ecdet="trend")
t1.h0 <- (t1@cval[, 2] > t1@teststat)
t2.h0 <- (t2@cval[, 2] > t2@teststat)
t3.h0 <- (t3@cval[, 2] > t3@teststat)
e1.h0 <- (e1@cval[, 2] > e1@teststat)
e2.h0 <- (e2@cval[, 2] > e2@teststat)
e3.h0 <- (e3@cval[, 2] > e3@teststat)
result = c(
  t.none = length(t1.h0[t1.h0==FALSE]),
  t.const = length(t2.h0[t2.h0==FALSE]),
  t.trend = length(t3.h0[t3.h0==FALSE]),
  e.none = length(e1.h0[e1.h0==FALSE]),
  e.const = length(e2.h0[e2.h0==FALSE]),
  e.trend = length(e3.h0[e3.h0==FALSE])
)
result <- result - args$rank
return (result)
}

```

```

#####
# jj.test funkcija
#####

```

```

jj.test <- function(ss, it, args) {
  params <- rep(ss, it)
  temp <- sapply(params, j.test, args = args)
  result <- data.frame(
    t.none.e0 = sum(temp[1,] == 0)/length(temp[1,]), #1
    t.none.e1 = sum(temp[1,] > 0)/length(temp[1,]), #2
    t.none.e2 = sum(temp[1,] < 0)/length(temp[1,]), #3
    t.const.e0 = sum(temp[2,] == 0)/length(temp[2,]), #4
    t.const.e1 = sum(temp[2,] > 0)/length(temp[2,]), #5
    t.const.e2 = sum(temp[2,] < 0)/length(temp[2,]), #6
    t.trend.e0 = sum(temp[3,] == 0)/length(temp[3,]), #7
    t.trend.e1 = sum(temp[3,] > 0)/length(temp[3,]), #8
    t.trend.e2 = sum(temp[3,] < 0)/length(temp[3,]), #9
    e.none.e0 = sum(temp[4,] == 0)/length(temp[4,]), #10
    e.none.e1 = sum(temp[4,] > 0)/length(temp[4,]), #11
    e.none.e2 = sum(temp[4,] < 0)/length(temp[4,]), #12
    e.const.e0 = sum(temp[5,] == 0)/length(temp[5,]), #13
    e.const.e1 = sum(temp[5,] > 0)/length(temp[5,]), #14
    e.const.e2 = sum(temp[5,] < 0)/length(temp[5,]), #15
    e.trend.e0 = sum(temp[6,] == 0)/length(temp[6,]), #16
    e.trend.e1 = sum(temp[6,] > 0)/length(temp[6,]), #17
    e.trend.e2 = sum(temp[6,] < 0)/length(temp[6,]) #18
  )
  return(result)
}

```

```
}

```

```
#####
# parametru objektas
#####

```

```
args <- list(
  dim = 2, # dimensiju skaičius
  rank = 0, # \Pi matricos rangas (kointegruojančių vektorių skaičius)
  arima = c(4, 1, 0), # arima komponentas
  dc = c(0, 0), # 1 - įjungta, 0 - išjungta
  dv = c(0, 0) # fiktyvūs kintamieji
)
```

```
#####
# tyrimo kodas
#####

```

```
models <- list()
models$m2040000 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 0)
models$m2041000 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 1)
models$m2040100 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 1)
models$m2041100 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 0)
args$rank <- 1
models$m2140000 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 0)
models$m2141000 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 1)
models$m2140100 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 1)
models$m2141100 <- data.frame(t(sapply(20:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 0)
args$rank <- 0
args$dim <- 3
models$m3040000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 0)
models$m3041000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 1)
models$m3040100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 1)
models$m3041100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 0)
args$rank <- 1
```

```

models$m3140000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 0)
models$m3141000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 1)
models$m3140100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 1)
models$m3141100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 0)
args$rank <- 2
models$m3240000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 0)
models$m3241000 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(0, 1)
models$m3240100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))
args$dc <- c(1, 1)
models$m3241100 <- data.frame(t(sapply(25:200, jj.test, it=1000, args = args)))

#####
# grafinės analizės kodas
#####

x0 <- models$m3040000
x1 <- models$m3140000
x2 <- models$m3240000

name = list(
  model = 6,
  type = "c0j0e0te"
)

d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 1])),
  x2 = c(rep(NA, 25), unlist(x0[, 10])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c0j0e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 2])),
  x2 = c(rep(NA, 25), unlist(x0[, 11])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c0j0e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 3])),
  x2 = c(rep(NA, 25), unlist(x0[, 12])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c1j0e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 1])),
  x2 = c(rep(NA, 25), unlist(x1[, 10])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c1j0e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 2])),
  x2 = c(rep(NA, 25), unlist(x1[, 11])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c1j0e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 3])),
  x2 = c(rep(NA, 25), unlist(x1[, 12])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c2j0e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 1])),
  x2 = c(rep(NA, 25), unlist(x2[, 10])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c2j0e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 2])),
  x2 = c(rep(NA, 25), unlist(x2[, 11])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c2j0e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 3])),
  x2 = c(rep(NA, 25), unlist(x2[, 12])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c0j1e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 4])),
  x2 = c(rep(NA, 25), unlist(x0[, 13])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c0j1e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 5])),
  x2 = c(rep(NA, 25), unlist(x0[, 14])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```



```

name$type = "c0j1e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 6])),
  x2 = c(rep(NA, 25), unlist(x0[, 15])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c1j1e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 4])),
  x2 = c(rep(NA, 25), unlist(x1[, 13])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c1j1e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 5])),
  x2 = c(rep(NA, 25), unlist(x1[, 14])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c1j1e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 6])),
  x2 = c(rep(NA, 25), unlist(x1[, 15])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2j1e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 4])),
  x2 = c(rep(NA, 25), unlist(x2[, 13])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c2j1e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 5])),
  x2 = c(rep(NA, 25), unlist(x2[, 14])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2j1e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 6])),
  x2 = c(rep(NA, 25), unlist(x2[, 15])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c0j2e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 7])),
  x2 = c(rep(NA, 25), unlist(x0[, 16])))

```

```

export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c0j2e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 8])),
  x2 = c(rep(NA, 25), unlist(x0[, 17])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c0j2e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 9])),
  x2 = c(rep(NA, 25), unlist(x0[, 18])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c1j2e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 7])),
  x2 = c(rep(NA, 25), unlist(x1[, 16])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c1j2e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 8])),
  x2 = c(rep(NA, 25), unlist(x1[, 17])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c1j2e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 9])),
  x2 = c(rep(NA, 25), unlist(x1[, 18])))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c2j2e0te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 7])),
  x2 = c(rep(NA, 25), unlist(x2[, 16])))
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

name$type = "c2j2e1te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 8])),
  x2 = c(rep(NA, 25), unlist(x2[, 17])))
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

name$type = "c2j2e2te"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 9])),

```

```
x2 = c(rep(NA, 25), unlist(x2[, 18]))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)
```

```
name$type = "c0e0t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 1])),
  x2 = c(rep(NA, 25), unlist(x0[, 4])),
  x3 = c(rep(NA, 25), unlist(x0[, 7])))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)
```

```
name$type = "c0e1t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 2])),
  x2 = c(rep(NA, 25), unlist(x0[, 5])),
  x3 = c(rep(NA, 25), unlist(x0[, 8])))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)
```

```
name$type = "c0e2t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 3])),
  x2 = c(rep(NA, 25), unlist(x0[, 6])),
  x3 = c(rep(NA, 25), unlist(x0[, 9])))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)
```

```
name$type = "c1e0t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 1])),
  x2 = c(rep(NA, 25), unlist(x1[, 4])),
  x3 = c(rep(NA, 25), unlist(x1[, 7])))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)
```

```
name$type = "c1e1t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 2])),
  x2 = c(rep(NA, 25), unlist(x1[, 5])),
  x3 = c(rep(NA, 25), unlist(x1[, 8])))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)
```

```
name$type = "c1e2t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 3])),
  x2 = c(rep(NA, 25), unlist(x1[, 6])),
```

```

x3 = c(rep(NA, 25), unlist(x1[, 9]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2e0t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 1])),
  x2 = c(rep(NA, 25), unlist(x2[, 4])),
  x3 = c(rep(NA, 25), unlist(x2[, 7]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c2e1t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 2])),
  x2 = c(rep(NA, 25), unlist(x2[, 5])),
  x3 = c(rep(NA, 25), unlist(x2[, 8]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2e2t"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 3])),
  x2 = c(rep(NA, 25), unlist(x2[, 6])),
  x3 = c(rep(NA, 25), unlist(x2[, 9]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c0e0e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 10])),
  x2 = c(rep(NA, 25), unlist(x0[, 13])),
  x3 = c(rep(NA, 25), unlist(x0[, 16]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c0e1e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 11])),
  x2 = c(rep(NA, 25), unlist(x0[, 14])),
  x3 = c(rep(NA, 25), unlist(x0[, 17]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c0e2e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x0[, 12])),

```

```

x2 = c(rep(NA, 25), unlist(x0[, 15])),
x3 = c(rep(NA, 25), unlist(x0[, 18]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c1e0e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 10])),
  x2 = c(rep(NA, 25), unlist(x1[, 13])),
  x3 = c(rep(NA, 25), unlist(x1[, 16]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c1e1e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 11])),
  x2 = c(rep(NA, 25), unlist(x1[, 14])),
  x3 = c(rep(NA, 25), unlist(x1[, 17]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c1e2e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x1[, 12])),
  x2 = c(rep(NA, 25), unlist(x1[, 15])),
  x3 = c(rep(NA, 25), unlist(x1[, 18]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2e0e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 10])),
  x2 = c(rep(NA, 25), unlist(x2[, 13])),
  x3 = c(rep(NA, 25), unlist(x2[, 16]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Teisingos specifikacijos dalis', min = 0, max = 1, h = 0.95, name=name)

```

```

name$type = "c2e1e"
d.x <- data.frame(
  x1 = c(rep(NA, 25), unlist(x2[, 11])),
  x2 = c(rep(NA, 25), unlist(x2[, 14])),
  x3 = c(rep(NA, 25), unlist(x2[, 17]))
d.x[c(26, 36, 51, 101, 201), ]
export(d.x, xlab = 'Imties dydis', ylab = 'Pirmos rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)

```

```

name$type = "c2e2e"
d.x <- data.frame(

```

```
x1 = c(rep(NA, 25), unlist(x2[, 12])),
x2 = c(rep(NA, 25), unlist(x2[, 15])),
x3 = c(rep(NA, 25), unlist(x2[, 18]))
export(d.x, xlab = 'Imties dydis', ylab = 'Antros rūšies klaidos tikimybė', min = 0, max = 1, h = 0.05, name=name)
```

```
#####
# lentelių generavimas
#####
```

```
x0 <- models$m3041100
x1 <- models$m3141100
x2 <- models$m3241100
```

```
tbl <- data.frame(
  d0ec0 <- unlist(x0$e.none.e0)[c(1, 11, 26, 76, 176)],
  d0ec1 <- unlist(x1$e.none.e0)[c(1, 11, 26, 76, 176)],
  d0ec2 <- unlist(x2$e.none.e0)[c(1, 11, 26, 76, 176)],
  d0tc0 <- unlist(x0$t.none.e0)[c(1, 11, 26, 76, 176)],
  d0tc1 <- unlist(x1$t.none.e0)[c(1, 11, 26, 76, 176)],
  d0tc2 <- unlist(x2$t.none.e0)[c(1, 11, 26, 76, 176)],
  d1ec0 <- unlist(x0$e.const.e0)[c(1, 11, 26, 76, 176)],
  d1ec1 <- unlist(x1$e.const.e0)[c(1, 11, 26, 76, 176)],
  d1ec2 <- unlist(x2$e.const.e0)[c(1, 11, 26, 76, 176)],
  d1tc0 <- unlist(x0$t.const.e0)[c(1, 11, 26, 76, 176)],
  d1tc1 <- unlist(x1$t.const.e0)[c(1, 11, 26, 76, 176)],
  d1tc2 <- unlist(x2$t.const.e0)[c(1, 11, 26, 76, 176)],
  d2ec0 <- unlist(x0$e.trend.e0)[c(1, 11, 26, 76, 176)],
  d2ec1 <- unlist(x1$e.trend.e0)[c(1, 11, 26, 76, 176)],
  d2ec2 <- unlist(x2$e.trend.e0)[c(1, 11, 26, 76, 176)],
  d2tc0 <- unlist(x0$t.trend.e0)[c(1, 11, 26, 76, 176)],
  d2tc1 <- unlist(x1$t.trend.e0)[c(1, 11, 26, 76, 176)],
  d2tc2 <- unlist(x2$t.trend.e0)[c(1, 11, 26, 76, 176)]
)
```

```
colnames(tbl) <- c('d0ec0', 'd0ec1', 'd0ec2', 'd0tc0', 'd0tc1', 'd0tc2',
  'd1ec0', 'd1ec1', 'd1ec2', 'd1tc0', 'd1tc1', 'd1tc2',
  'd2ec0', 'd2ec1', 'd2ec2', 'd2tc0', 'd2tc1', 'd2tc2')
```

```
rownames(tbl) <- c('Labai mažos imtys (25)',
  'Mažos imtys (35)',
  'Vidutinės imtys (50)',
  'Didelės imtys (100)',
  'Labai didelės imtys (200)')
```

```
tbl <- data.frame(
  d0e = round((tbl$d0ec0 + tbl$d0ec1 + tbl$d0ec2)/3, digits=2),
  d0t = round((tbl$d0tc0 + tbl$d0tc1 + tbl$d0tc2)/3, digits=2),
  d1e = round((tbl$d1ec0 + tbl$d1ec1 + tbl$d1ec2)/3, digits=2),
  d1t = round((tbl$d1tc0 + tbl$d1tc1 + tbl$d1tc2)/3, digits=2),
```

```

d2e = round((tbl$d2ec0 + tbl$d2ec1 + tbl$d2ec2)/3, digits=2),
d2t = round((tbl$d2tc0 + tbl$d2tc1 + tbl$d2tc2)/3, digits=2)
)

```

```
write.xlsx(tbl, './file.xlsx')
```

```
#####
# dgp funkcija generuoja duomenis pagal specifikuotus parametrus #
#####

```

```
dgp <- function(args) {
```

```
#####
# Testo režimas #
#####

```

```
dgp.test <- F
```

```
if (dgp.test) {
```

```
  warning('Testo režimas.')
```

```
  args <- list(
```

```
    ss = 1000, # imties dydis
```

```
    dim = 4, # dimensijų skaičius
```

```
    rank = 3, # \Pi matricos rangas (kointegruojančių vektorių skaičius)
```

```
    arima = c(4, 1, 0), # arima komponentas
```

```
    dc = c(0, 1), # deterministiniai komponentai (skaičiai reiškia sd)
```

```
    dv = c(0, 0, 0) # fiktyvūs kintamieji (skaičiai reiškia sd)
```

```
  )
```

```
}
```

```
#####
# Parametrų diagnostika #
#####

```

```
if (args$dim <= args$rank) {
```

```
  stop('Matricos \Pi rangas negali būti didesnis arba lygus dimensijų skaičiui.')
```

```
}
```

```
if (args$ss < args$arima[[1]]*2) {
```

```
  stop('Imties dydis negali būti mažesnis nei ar eilė padauginta iš 2')
```

```
}
```

```
if (args$arima[2] > 1) {
```

```
  stop('Šiuo metu galima į modelį įdėti ne daugiau nei 1 eilė integruotus kintamuosius.')
```

```
}
```

```
#####
# Rezultatų vektorius #

```

```
#####

result <- list(
  args = args,
  params = list (
    ar = c(),
    ma = c(),
    const = c(),
    trend = c(),
    idv = list(
      x = c(),
      sp = c(),
      ep = c()
    ),
    pdv = c(),
    pdvt = c(),
    cv = c(), # nenormalizuotas \beta vektorius
    beta = c(), # normalizuotas \beta vektorius
    st = c() # stochastinis trendas
  )
)

#####
# Sukuriamos modelio paklaidos #
#####

dgp.err <- lapply(rep.int(0, args$dim), rnorm, n = args$ss)
dgp.vecm <- dgp.err
result$serr <- dgp.err

#####
# Pridedamas impulso fiktyvus kintamasis
#####

if (args$dv[1]) {
  result$params$idv$x <- sample(c(-55:-5, 5:55), args$dim, T)
  result$params$idv$sp <- sample(1:args$ss, args$dim)
  dgp.vecm <- lapply(1:args$dim, function (it) {
    dgp.vecm[[it]][result$params$idv$sp[[it]]] <- result$params$idv$x[[it]]
    return(dgp.vecm[[it]])
  })
}

#####
# Pridedamas poslinkio fiktyvus kintamasis
#####
```



```

if (args$dv[2]) {
  result$params$pdv$x <- sample(c(-55:-5, 5:55), args$dim, T)
  result$params$pdv$sp <- sample(1:args$ss, args$dim)
  dgp.pdv <- lapply(result$params$pdv$x, rep.int, time=args$ss)
  dgp.pdv <- lapply(1:args$dim, function (it) {
    dgp.pdv[[it]][1:result$params$pdv$sp[[it]]] <- 0
    return(dgp.pdv[[it]])
  })
  dgp.vecm <- mapply("+", dgp.vecm, dgp.pdv, SIMPLIFY = F)
  result$pdv <- dgp.pdv
}

#####
# Sukuriami ARMA parametrai #
#####

if (args$arima[1]) {
  result$params$ar <- lapply(
    1:args$dim,
    function (dim) sample(-90:90, args$arima[1])/100/args$arima[1]
  )
}

if (args$arima[3]) {
  result$params$ma <- lapply(
    1:args$dim,
    function (dim) sample(-90:90, args$arima[3])/100/args$arima[3]
  )
}

#####
# Simuliuojami ARMA procesai #
#####

dgp.arma <- lapply(
  1:args$dim,
  function(dim) {
    arima.sim(
      n = args$ss,
      model = list(
        ar = result$params$ar[[dim]],
        ma = result$params$ma[[dim]]
      ),
      innov = dgp.vecm[[dim]]
    )
  }
)

```

```

result$arma <- data.frame(
  matrix(
    unlist(dgp.arma),
    ncol = args$dim,
    byrow = FALSE
  )
)

#####
# Priedamas bendras stochastinis trendas #
#####

if (args$rank < args$dim) {
  dgp.st <- lapply(rep.int(0, args$dim), rnorm, n = args$ss)
  dgp.st <- lapply(dgp.st, cumsum)
  if (args$rank > 0) {
    dgp.stc <- Reduce("+", dgp.st[1: (args$dim - args$rank)])
    dgp.st[(args$dim - args$rank + 1): args$dim] <- rep(list(dgp.stc), args$rank)
  }
  result$params$st <- dgp.st
  result$params$cv <- sample(c(-55:-5, 5:55), args$dim, T)/10
  result$params$beta <- result$params$cv/result$params$cv[1]
  dgp.cv <- mapply("*", result$params$st, result$params$cv, SIMPLIFY = F)
  dgp.vecm <- mapply("+", dgp.arma, dgp.cv, SIMPLIFY = F)
}

#####
# Priedami deterministiniai komponentai #
#####

if (args$dc[1]) {
  result$params$const <- sample(c(-100:-10, 10:100), args$dim, T)
  dgp.vecm <- mapply("+", dgp.vecm, result$params$const, SIMPLIFY = F)
}

if (args$dc[2]) {
  result$params$trend <- sample(c(-55:-5, 5:55), args$dim, T)/100
  dgp.trend <- lapply(result$params$trend, function(it) {
    return (ts(it*(1:args$ss)))
  })
  dgp.vecm <- mapply("+", dgp.vecm, dgp.trend, SIMPLIFY = F)
}

#####
# Sukuriama rezultatų duomenų bazė #
#####

```

```

result$vecm <- data.frame(
  matrix(
    unlist(dgp.vecm),
    ncol = args$dim,
    byrow = FALSE
  )
)
return(result)
}

```

```

#####
# defaults funkcija
#####

```

```
defaults <- function (name) {
```

```

#####
# Pirminiai nustatymai dgp funkcijai
#####

```

```

if (name == 'dgp') {
  args <- list(
    ss = 50, # imties dydis
    dim = 2, # dimensijų skaičius
    rank = 0, # \Pi matricos rangas (kointegruojančių vektorių skaičius)
    arima = c(4, 1, 0), # arima komponentas
    dc = c(0, 0), # 1 - įjungta, 0 - išjungta
    dv = c(0, 0) # fiktyvūs kintamieji (skaičiai reiškia sd)
  )
  return (args)
}

```

```

#####
# Nustatymai money funkcijai
#####

```

```

if (name == 'money') {
  args <- list (
    country = 'usa',
    freq = 4,
    log = T
  )
  return (args)
}

```

```
#####
# Jei parametrai nėra teisingi
#####

return (F)
}

#####
# export funkcija
#####

export <- function (d.x, xlab, ylab, name="", min = 0, max = 1, h = 0) {
  if (name == "") {
    filename <- paste(format(Sys.time(), "%Y%m%d%H%M%S"), '.png', sep = "")
  } else {
    filename = paste("m", name$model, name$type, ".png", sep="")
  }
  png(filename = filename, width = 1920, height = 1080, pointsize = 11, res = 230)
  plot.ts(d.x, xlab = xlab, ylab = ylab, ylim=c(min, max), plot.type="single", lty=1:5, las = 1)
  abline(h=h, v=c(50, 100), lty = c(3, 2, 2))
  axis(side = 2, at=h, las = 1)
  axis(side = 2, at=c(0.1, 0.3, 0.5, 0.7, 0.9), las = 1)
  axis(side = 1, at=c(25, 75, 125, 175))
  dev.off()
}
```

Priedas Nr. 2.

Pinigų paklausos tyrimams naudotas R kodas

```
#####
# money funkcija
#####

money <- function (args) {

  #####
  # Duomenų importavimas
  #####

  raw.data <- read.delim(paste("./data/", args$country, ".txt", sep = ""))

  #####
  # Pavertimas į time series formatą
  #####

  start.date <- as.numeric(substr(as.character(raw.data$DATE), 1, 4)[1])
  start.q <- as.numeric(substr(as.character(raw.data$DATE), 7, 7)[1])

  result <- list (
    M1 = ts(raw.data$M1/raw.data$CPI, start = c(start.date, start.q), frequency = args$freq),
    IR = ts(raw.data$IR, start = c(start.date, start.q), frequency = args$freq), #palūkanų norma
    # CPI = ts(raw.data$CPI, start = c(start.date, start.q), frequency = args$freq), #VKI
    Y = ts(raw.data$Y, start = c(start.date, start.q), frequency = args$freq) #BVP
  )

  #####
  # Pavertimas į Log
  #####

  if (args$log) {
    result$M1 <- log(result$M1)
    result$Y <- log(result$Y)
  }

  return (result)

}

usa.list <- money (def.args)

plot.ts(usa.list$M1, xlab = "Metai", ylab = "Pokytiis")
```

```

# export(usa.list$M1, xlab = "Metai", ylab = "Pokytis")
summary(ur.df(usa.list$M1, type="drift"))
# H0
summary(ur.kpss(usa.list$M1, type = "tau"))
# HA

plot.ts(diff(usa.list$M1), xlab = "Metai", ylab = "Pokytis")
# export(diff(usa.list$M1), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(usa.list$M1), type="drift"))
# HA

plot.ts(usa.list$IR, xlab = "Metai", ylab = "proc.")
# export(usa.list$IR, xlab = "Metai", ylab = "proc.")
summary(ur.df(usa.list$IR, type="none"))
# H0
summary(ur.kpss(usa.list$IR, type = "tau"))
# HA

plot.ts(diff(usa.list$IR), xlab = "Metai", ylab = "proc.")
# export(diff(usa.list$IR), xlab = "Metai", ylab = "proc.")
summary(ur.df(diff(usa.list$IR), type="none"))
# HA

plot.ts(usa.list$Y, xlab = "Metai", ylab = "Pokytis")
# export(usa.list$Y, xlab = "Metai", ylab = "Pokytis")
summary(ur.df(usa.list$Y, type="drift"))
# H0
summary(ur.kpss(usa.list$Y, type="tau"))
# HA

plot.ts(diff(usa.list$Y), xlab = "Metai", ylab = "Pokytis")
# export(diff(usa.list$Y), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(usa.list$Y), type="none"))
# HA
summary(ur.kpss(diff(usa.list$Y), type="tau"))
# HA

usa.list$YMR <- cbind(usa.list$Y, usa.list$M1, usa.list$IR)

VARselect(usa.list$YMR, lag.max = 15)

# VECM(8) atvejis

# Be deterministinių komponentų

usa.list$vecm8nd <- ca.jo(usa.list$YMR, type = "eigen", ecdet = "none", K=8, spec = "transitory")
summary(usa.list$vecm8nd)

```

```

# 0 kointegruojančių vektorių
usa.list$vecm8nd.rls <- cajorls(usa.list$vecm8nd)
usa.list$vecm8nd.rls$beta
acf(residuals(usa.list$vecm8nd.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'usa'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 20)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="none", K=8, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# Su konstanta

usa.list$vecm8c <- ca.jo(usa.list$YMR, type = "eigen", ecdet = "const", K=8, spec = "transitory")
summary(usa.list$vecm8c)
# 1 kointegruojantis vektorius
usa.list$vecm8c.rls <- cajorls(usa.list$vecm8c)
usa.list$vecm8c.rls$beta
acf(residuals(usa.list$vecm8c.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'usa'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 20)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="const", K=8, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# Su trendu

usa.list$vecm8t <- ca.jo(usa.list$YMR, type = "eigen", ecdet = "trend", K=8, spec = "transitory")
summary(usa.list$vecm8t)
# 0 kointegruojančių vektorių
usa.list$vecm8t.rls <- cajorls(usa.list$vecm8t)
usa.list$vecm8t.rls$beta

```

```

acf(residuals(usa.list$vecm8t.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'usa'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 20)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="trend", K=8, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

def.args <- defaults('money')
def.args$country <- "lt"
lt.list <- money (def.args)

plot.ts(lt.list$M1, xlab = "Metai", ylab = "Pokytis")
# export(lt.list$M1, xlab = "Metai", ylab = "Pokytis")
summary(ur.df(lt.list$M1, type="drift"))
# H0
summary(ur.kpss(lt.list$M1, type = "tau"))
# H0 / HA

plot.ts(diff(lt.list$M1), xlab = "Metai", ylab = "Pokytis")
# export(diff(lt.list$M1), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(lt.list$M1), type="none"))
# HA

plot.ts(lt.list$IR, xlab = "Metai", ylab = "proc.")
# export(lt.list$IR, xlab = "Metai", ylab = "proc.")
summary(ur.df(lt.list$IR, type="none"))
# H0
summary(ur.kpss(lt.list$IR, type = "mu"))
# HA

plot.ts(diff(lt.list$IR), xlab = "Metai", ylab = "Pokytis")
# export(diff(lt.list$IR), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(lt.list$IR), type="none"))
# HA

plot.ts(lt.list$Y, xlab = "Metai", ylab = "mlrd. EUR")
# export(lt.list$Y, xlab = "Metai", ylab = "mlrd. EUR")
summary(ur.df(lt.list$Y, type="drift"))
# H0

```



```

summary(ur.kpss(lt.list$Y, type="tau"))
# HA

plot.ts(diff(lt.list$Y), xlab = "Metai", ylab = "Pokytis")
# export(diff(lt.list$Y), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(lt.list$Y), type="none"))
# HA
summary(ur.kpss(diff(lt.list$Y), type="tau"))
# HA

lt.list$YMR <- cbind(lt.list$Y, lt.list$M1, lt.list$IR)
VARselect(lt.list$YMR, lag.max = 4)

#####
# Sezoniškumo pašalinimas
#####

lt.list$Yde <- decompose(lt.list$Y, type="multiplicative")
lt.list$Yd <- lt.list$Y / lt.list$Yde$seasonal

plot.ts(lt.list$Yd, xlab = "Metai", ylab = "Pokytis")
# export(lt.list$Yd, xlab = "Metai", ylab = "Pokytis")
summary(ur.df(lt.list$Yd, type="drift"))
# H0 (nestacionarus)
summary(ur.kpss(lt.list$Yd, type="tau"))
# HA (integruotas)

plot.ts(diff(lt.list$Yd), xlab = "Metai", ylab = "Pokytis")
# export(diff(lt.list$Yd), xlab = "Metai", ylab = "Pokytis")
summary(ur.df(diff(lt.list$Yd), type="none"))
# HA

lt.list$YMRd <- cbind(lt.list$Yd, lt.list$M1, lt.list$IR) #suformuojama integruotų kintamųjų matrica
plot.ts(lt.list$YMRd)
VARselect(lt.list$YMRd, lag.max = 8) #FPE siūlo VECM(4)
# Juselius siūlo įtraukti mažiau lag'ų

# VECM(4) atvejis

acf(residuals(VAR(lt.list$YMRd, p=4))) #IR (kaip ir VAR modelyje) yra autokoreliuotas
str(acf(residuals(VAR(lt.list$YMRd, p=4))))
# Be deterministinių komponentų

lt.list$vecm4nd <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "none", K=4, spec = "transitory")
summary(lt.list$vecm4nd)
# 0 / 1 kointegruojančių vektorių
lt.list$vecm4nd <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "none", K=4, spec = "transitory")

```

```

summary(lt.list$vecm4nd)
lt.list$vecm4nd.rls <- cajorls(lt.list$vecm4nd)
lt.list$vecm4nd.rls$beta
acf(residuals(lt.list$vecm4nd.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="none", K=4, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# Su konstanta

lt.list$vecm4c <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "const", K=4, spec = "transitory")
summary(lt.list$vecm4c)
# 2 kointegruojantys vektoriai
lt.list$vecm4c <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "const", K=4, spec = "transitory")
summary(lt.list$vecm4c)
# 2 kointegruojantys vektoriai
lt.list$vecm4c.rls <- cajorls(lt.list$vecm4c)
lt.list$vecm4c.rls$beta
acf(residuals(lt.list$vecm4c.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="const", K=8, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# Su trendu

lt.list$vecm4t <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "trend", K=4, spec = "transitory")
summary(lt.list$vecm4t)

```

```

# 0 kointegruojančių vektorių
lt.list$vecm4t <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "trend", K=4, spec = "transitory")
summary(lt.list$vecm4t)
# 0 kointegruojančių vektorių
lt.list$vecm4t.rls <- cajorls(lt.list$vecm4t)
lt.list$vecm4t.rls$beta
acf(residuals(lt.list$vecm4t.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="trend", K=4, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# VECM(6) atvejis

# Be deterministinių komponentų

lt.list$vecm6nd <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "none", K=6, spec = "transitory")
summary(lt.list$vecm6nd)
# 1 kointegruojantis vektorius
lt.list$vecm6nd <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "none", K=6, spec = "transitory")
summary(lt.list$vecm6nd)
# 1 kointegruojantis vektorius
lt.list$vecm6nd.rls <- cajorls(lt.list$vecm6nd)
lt.list$vecm6nd.rls$beta
acf(residuals(lt.list$vecm6nd.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="none", K=6, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

```

```

# Su konstanta

lt.list$vecm6c <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "const", K=6, spec = "transitory")
summary(lt.list$vecm6c)
# 2 kointegruojantys vektoriai
lt.list$vecm6c <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "const", K=6, spec = "transitory")
summary(lt.list$vecm6c)
# 2 kointegruojantys vektoriai
lt.list$vecm6c.rls <- cajorls(lt.list$vecm6c)
lt.list$vecm6c.rls$beta
acf(residuals(lt.list$vecm6c.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="const", K=6, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))

# Su trendu

lt.list$vecm6t <- ca.jo(lt.list$YMRd, type = "eigen", ecdet = "trend", K=6, spec = "transitory")
summary(lt.list$vecm6t)
# 1 kointegruojantis vektorius
lt.list$vecm6t <- ca.jo(lt.list$YMRd, type = "trace", ecdet = "trend", K=6, spec = "transitory")
summary(lt.list$vecm6t)
# 1 / 2 kointegruojantys vektoriai
lt.list$vecm6t.rls <- cajorls(lt.list$vecm6t)
lt.list$vecm6t.rls$beta
acf(residuals(lt.list$vecm6t.rls$rlm))
y.args <- defaults('money')
y.args$country <- 'lt'
y <- data.frame(money(y.args))
y
VARselect(y, lag.max = 8)
plot.ts(y)
y0 <- ca.jo(y, type="eigen", ecdet="trend", K=6, spec="longrun")
summary(y0)
y1 <- cajorls(y0, r=1)
y1$beta

```

```
y2 <- (y$M1 + y1$beta[2]*y$IR + y1$beta[3]*y$Y)
ts.plot(y2)
summary(ur.df(y2, type="drift", lags=8))
```