LIETUVOS EDUKOLOGIJOS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

MATEMATIKOS KATEDRA

INESA ŽAKIENĖ

HORVICO IR TOMPSONO ĮVERTINIO DISPERSIJOS VERTINIMAS

Estimation of the variance of the Horvitz & Thompson estimator

Matematikos magistro diplominis darbas

Darbo vadovas: doc. dr. D. Pumputis

Vilnius, 2012

Turinys

ĮVADAS

Baigtinių populiacijų statistika, kitaip vadinama imčių metodais, yra gana jauna matematinės statistikos mokslo šaka, pradėjusi sparčiai vystytis XX a. viduryje. Jos pradininku laikomas J. Neymanas, 1934 m. parašęs straipsnį “Atsitiktinės imtys”, kuriame atliekant įvairius skaičiavimus imčių pagrindu, pagrindžiami atsitiktinių imčių privalumai.

Susidomėjimas baigtinių populiacijų statistika Lietuvoje pradėjo augti nuo 1990 m. Imčių teoriją ir metodus pirmieji pradėjo nagrinėti D. Krapavickaitė ir A. Plikusas. Baigtinių populiacijų metodai nuo 1996 m. Lietuvoje plačiai taikomi:

- Valstybinėje statistikoje
- Demografiniuose tyrimuose
- Viešosios nuomonės tyrimuose
- Ekologiniuose tyrimuose
- Draudimų kompanijose ir bankuose.

Klasikinis baigtinių populiacijų statistikos uždavinys - tai populiacijos parametrų vertinimas, pavyzdžiui, kintamojo reikšmių suma, populiacijos vidurkis, populiacijos dispersija ir t.t. Minėti parametrai dažniausiai yra nežinomi, todėl jiems vertinti konstruojami įvairūs įvertiniai. Kaip pavyzdį, galime imti gerai žinomą universalų populiacijos sumos įvertinį, kurį 1952 m. pasiūlė D. Horvicas ir D. Tompsonas. Tai vienas iš svarbiausių imčių teorijos rezultatų. Todėl natūralu, jog šis įvertinys nuolat tyrinėjamas, analizuojamas. Viena iš nagrinėjamų aktualių problemų - Horvico ir Tompsono įvertinio dispersijos vertinimas. Šios dispersijos prireikia sprendžiant įvairius uždavinius.

Šiame darbe, remiantis J. Deville & C. Särndal [14] pasiūlytu imties plano svorių kalibravimo metodu, sukonstruoti aštuoni nauji Horvico ir Tompsono įvertinio dispersijos įvertiniai, kurie tarpusavyje palyginti empiriškai. Darbe taip pat pateikiamos sukonstruotų įvertinių apytikslės dispersijos ir šių dispersijų įvertiniai.

Darbo tikslas

Pagrindinis šio darbo tikslas - sukonstruoti naujus Horvico ir Tompsono įvertinio dispersijos įvertinius ir ištirti jų tikslumą.

Darbo uždaviniai

Darbo tikslui pasiekti iškelti šie uždaviniai:

1. Pasiūlyti įvertinių pavidalą ir naudojant skirtingas atstumo funkcijas ir kalibravimo lygtis išvesti įvertinių svorius.

2. Naudojant Teiloro ištiesinimo metodą, apskaičiuoti sukonstruotų įvertinių apytiksles dispersijas ir pasiūlyti šių dispersijų įvertinius.

3. Taikant matematinį modeliavimą sukonstruotus įvertinius palyginti tarpusavyje ir su standartiniu įvertiniu.

Tyrimo metodai

Iškeltų uždavinių sprendimui, diplominiame darbe, taikomi šie metodai:

- Lagranžo neapibrėžtųjų daugiklių metodas

- Matematinis modeliavimas

- Teiloro ištiesinimo metodas

- Sluoksniavimo metodai

Mokslinis darbo naujumas

Darbe, naudojant skirtingas atstumo funkcijas ir kalibravimo lygtis, išvedami įvertinių svoriai. Naudojant juos, sukonstruojami nauji Horvico ir Tompsono įvertinio dispersijos įvertiniai. Šie įvertiniai, taikant matematinį modeliavimą, palyginami tarpusavyje ir su standartiniu įvertiniu. Be to, darbe pateikiamos sukonstruotų įvertinių apytikslės dispersijos ir pasiūlyti šių dispersijų įvertiniai.

1 PAGRINDINĖS SĄVOKOS IR METODAI

1.1 Populiacija ir imtis

Tyrime nagrinėjama objektų aibė vadinama ***populiacija***. Iš N elementų sudarytą baigtinę populiaciją žymėsime $\mathcal{U} = \{u_1, u_2, ..., u_N\}$. ***Imtis*** - tai vienas ar daugiau ėmimo vienetų, paimtų iš populiacijos, siekiant gauti informaciją apie populiaciją. Imtį iš baigtinės populiacijos žymėsime $\mathbf{i} = \{u_{i_1}, u_{i_2}, ..., u_{i_n}\}$. Imties elementų skaičius n vadinamas ***imties dydžiu***. Darbe nagrinėsime tikimybines imtis, apibrėžiamas 1.2 poskyryje.

Turėdami baigtinę populiaciją $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ laikysime, kad kievienam jos elementui apibrėžta tam tikra kiekybinio požymio reikšmė. Matuodami šio požymio reikšmes, gausime tam tikrą dydį y, kurį vadinsime ***kintamuoju*** arba ***tyrimo kintamuoju***. Šio kintamojo reikšmes žymėsime $y_1, y_2, ..., y_N$, o imties $\mathbf{i}$ elementų kintamojo y reikšmes žymėsime $y_{i_1}, y_{i_2}, ..., y_{i_n}$. Kartais, vertinant sudėtingesnius parametrus, pavyzdžiui dviejų sumų santykį, nepakanka vieno kintamajojo. Tada apibrėžiamas kintamasis x, jo reikšmes žymėsime $x_1, x_2, ..., x_N$ ir atitinkamai imties elementų reikšmes $x_{i_1}, x_{i_2}, ..., x_{i_n}$. Šis kintamasis laikomas antruoju tyrimo kintamuoju, o kartais, kai jo reikšmes gauname iš pagalbinių šaltinių - ***papildomu kintamuoju***.

Atliekant tyrimą dažniausiai mus domina ne pats kintamasis, o kintamojo reikšmių funkcija - ***populiacijos parametras*** θ:

$$\theta = \theta(y_1, y_2, ..., y_N).$$

Baigtinei populiacijai galima apibrėžti nemažai parametrų, kurie ją apibūdina, pavyzdžiui vidurkis, dispersija, suma, dviejų sumų santykis ir pan.

1.2 Tikimybinis ėmimas. Tikimybinė imtis

Tikimybinis ėmimas ([5], p.27) - tai imties išrinkimo būdas, kuris, tiesiogiai renkant elementus iš populiacijos, tenkina tokias sąlygas:

a) Nusakomos visos galimos skirtingos imtys $\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_V$, kurias iš turimos populiacijos galima išrinkti taikant imčių išrinkimo procedūrą. Vadinasi, galima nusakyti, kurie populiacijos elementai priklauso atitinakamai imčiai $\mathbf{i}_1, \mathbf{i}_2$ ir t.t.

b) Kiekvienai galimai imčiai $\mathbf{i}_v$ priskiriama žinoma imties išrinkimo tikimybė

$$\mathbf{P}(\mathbf{i}_v) = p(\mathbf{i}_v) > 0, \quad v = 1, 2, ..., V,$$

taip, kad

$$\sum_{v=1}^{V} p(\mathbf{i}_v) = 1.$$

c) Kiekvienas populiacijos elementas priklauso bent vienai galimai imčiai.

d) Imtis išrenkama taikant tokią atsitiktinę procedūrą, kai kiekviena iš galimų imčių $\mathbf{i}_v$ išrenkama su nurodyta tikimybe $p(\mathbf{i}_v)$, $v = 1, ..., V$.

Imtis, kuri gaunama, taikant imties išrinkimo procedūrą, tenkinančią visus minėtus reikalavimus, vadinama ***tikimybine imtimi***.

Galimos imtys $\mathbf{i}_1, ..., \mathbf{i}_V$ ir tikimybės, su kuriomis jos gali būti išrinktos $p(\mathbf{i}_1), ..., p(\mathbf{i}_V)$, apibrėžia tikimybinį skirstinį, vadinamą ***imties planu***. Tikimybės $p(\mathbf{i}_v)$, $v = 1, 2, ..., V$, vadinamos ***imties plano tikimybėmis***.

1.3 Įvertiniai ir jų tikslumo matai

Parametro θ ***įvertiniu*** vadinama funkcija, kuri nurodo, kaip, turint imtį, įvertinti parametrą, ir jį žymėsime $\hat{\theta} = \hat{\theta}(y_{i_1}, y_{i_2}, ..., y_{i_n})$. Skaitinę įvertinio reikšmę, apskaičiuotą konkrečiai imčiai, vadinsime ***įverčiu***.

Norint įvertinti baigtinės populiacijos parametrą θ, renkama tikimybinė imtis. Tarkime, kad pagal pasirinktą imties planą yra V galimų imčių. Parametro θ įvertinys $\hat{\theta}$ yra atsitiktinis dydis. Pažymėkime $\hat{\theta}_1$ jo reikšmę, gautą, apskaičiavus įvertį imčiai $\mathbf{i}_1$, $\hat{\theta}_2$ - jo reikšmę, gautą apskaičiavus įvertį imčiai $\mathbf{i}_2$,..., ir $\hat{\theta}_V$ - reikšmę, gautą apskaičiavus įvertį imčiai $\mathbf{i}_V$. Žinodami visas galimas įvertinio reikšmes ir tikimybes, su kuriomis tos reikšmės įgyjamos, galime apskaičiuoti ***įvertinio vidurkį***, imties plano tikimybių atžvilgiu:

$$\mathbf{E}\hat{\theta} = \sum_{i=1}^{V} \hat{\theta}_i p(\mathbf{i}_i).$$

Skirtumas $\mathbf{E}\hat{\theta} - \theta$ vadinamas ***įvertinio*** $\hat{\theta}$ ***poslinkiu***. Jį pažymėsime

$$\mathbf{POSL}(\hat{\theta}) = \mathbf{E}\hat{\theta} - \theta.$$

Įvertinio poslinkis parodo, kiek vidutiniškai parametro įvertinys yra nutolęs nuo tikrosios parametro reikšmės. Parametro θ įvertinys $\hat{\theta}$ vadinamas ***nepaslinktuoju***, jei jo vidurkis imties plano tikimybių atžvilgiu sutampa su tikrąja to parametro reikšme arba jei jo poslinkis lygus nuliui:

$$\mathbf{E}\hat{\theta} = \theta, \text{ arba } \mathbf{POSL}(\hat{\theta}) = 0.$$

Įvertinio nepaslinktumas yra svarbi jo savybė. Patariama taikyti tik nepaslinktuosius, arba turinčius pakankamai mažą poslinkį, įvertinius.

Įvertinio $\hat{\theta}$ sklaidą apie jo vidutinę reikšmę apibūdina ***įvertinio dispersija*** $\mathbf{D}\hat{\theta}$:

$$\mathbf{D}\hat{\theta} = \mathbf{E}(\hat{\theta} - \mathbf{E}\hat{\theta})^2 = \sum_{k=1}^{V} (\hat{\theta}_k - \mathbf{E}\hat{\theta})^2 p(\mathbf{i}_k).$$

Įvertinio dispersija - tai paklaida, atsirandanti dėl imties atsitiktinumo, kuri dar vadinama ***imties paklaida***. Ji atsiranda todėl, kad vietoj visos populiacijos tiriame imtį. Įvertinio reikšmė, apskaičiuota konkrečiai imčiai, priklauso tik nuo imties elementų kintamojo reikšmių, o ne nuo visos populiacijos elementų kintamojo reikšmių.

Įvertinio $\hat{\theta}$ ***vidutine kvadratine paklaida*** (**VKP**) vadinsime jo nuokrypio nuo tikrosios reikšmės kvadrato vidurkį imties plano tikimybių atžvilgiu:

$$\mathbf{VKP}(\hat{\theta}) = \mathbf{E}(\hat{\theta} - \theta)^2$$

arba

$$\mathbf{VKP}(\hat{\theta}) = \mathbf{D}\hat{\theta} + (\mathbf{POSL}(\hat{\theta}))^2.$$

Dažnai būna sunku įvertinti dispersijos $\mathbf{D}\hat{\theta}$ dydį. Todėl dažniau naudojamas santykinis įvertinio paklaidos matas - ***variacijos koeficientas***:

$$\widehat{\mathbf{cv}}(\hat{\theta}) = \frac{\sqrt{\mathbf{D}\hat{\theta}}}{\mathbf{E}\hat{\theta}}, \quad \mathbf{E}\hat{\theta} \neq 0.$$

Kuo šis matas mažesnis, tuo įvertinys tikslesnis.

Įvertinio tikslumas priklauso nuo daugelio dalykų: nuo paties įvertinio, imties plano, imties dydžio, populiacijos kintamojo skirstinio.

1.4 Horvico ir Tompsono įvertinys

Nagrinėkime baigtinę populiaciją $\mathcal{U} = \{u_1, u_2, ..., u_N\}$, kurioje apibrėžtas tyrimo kintamasis y: $y_1, y_2, ..., y_N$ ir papildomas kintamasis x: $x_1, x_2, ..., x_N$, o populiacijos vertinamasis parametras - kintamojo y reikšmių suma

$$t_y = \sum_{k=1}^{N} y_k.$$

1952 m. Horvicas ir Tompsonas [3] pasiūlė populiacijos sumos įvertinį

$$\hat{t}_\pi = \sum_{k=1}^{\nu} \frac{y_k}{\pi_k}, \tag{1}$$

čia π_k, $k = 1, 2, ..., N$ - k- ojo populiacijos elemento priklausymo imčiai tikimybė;
ν - efektyvusis imties dydis, t.y skirtingų elementų skaičius n dydžio imtyje. Šis įvertinys tapo vienu iš svarbiausių imčių teorijos rezultatų. (1) įvertinys yra universalus, kadangi tinka bet kokiam imties planui.

Horvico ir Tompsono sumos įvertinys dažnai literatūroje sutinkamas tokiame pavidale

$$\hat{t}_\pi = \sum_{k=1}^{\nu} d_k y_k, \tag{2}$$

čia svoris $d_k = \frac{1}{\pi_k}$ yra populiacijos elementų skaičius, kuriam atstovauja elementas y_k, būdamas išrinktu į imtį.

Pažymėkime π_k, $k = 1, 2, ..., N$ - k-ojo populiacijos elemento priklausymo imčiai tikimybę $\pi_k = \mathbf{P}\{k \in \mathbf{i}\}$; $\pi_{kl} = \mathbf{P}(k \in \mathbf{i}, l \in \mathbf{i})$, $k, l = 1, 2, ..., N$, - k-ojo ir l-ojo populiacijos elementų tikimybę kartu priklausyti imčiai.

1 teiginys (žr. [5], 94 psl). Pasirinkus bet kokį imties planą,

a) populiacijos sumos įvertinys

$$\hat{t}_\pi = \sum_{k=1}^{\nu} \frac{y_k}{\pi_k}$$

yra nepaslinktasis;

b) šio įvertinio dispersija yra

$$\mathbf{D}\hat{t}_\pi = \sum_{k=1}^{N} \frac{1-\pi_k}{\pi_k} y_k^2 + \sum_{k=1}^{N} \sum_{\substack{l=1 \\ l \neq k}}^{N} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_k \pi_l} y_k y_l; \tag{3}$$

c) įvertinio $\hat{t}_\pi$ dispersijos įvertinys

$$\widehat{\mathbf{D}}\hat{t}_\pi = \sum_{k=1}^{\nu} \frac{1-\pi_k}{\pi_k^2} y_k^2 + \sum_{k=1}^{\nu} \sum_{\substack{l=1 \\ l \neq k}}^{\nu} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_k \pi_l} \frac{y_k y_l}{\pi_{kl}}, \quad \pi_{kl} > 0 \tag{4}$$

yra nepaslinktasis.

Siekdami gauti tikslesnius įvertinius, J. Deville & C. Särndal [14] pasiūlė svorių kalibravimo idėją. Remiantis ja, kintamojo y sumos t_y ***kalibruotuoju įvertiniu*** vadinamas toks įvertinys

$$\hat{t}_y = \sum_{k \in i} \omega_k y_k,$$

kurio svoriai tenkina sąlygas:

1. ω_k kaip galima mažiau skiriasi nuo svorių d_k minimizuodami funkciją

$$L(\omega_k, d_k, k \in \mathbf{i}) = \sum_{k \in \mathbf{i}} \frac{(\omega_k - d_k)^2}{d_k q_k},$$

   čia q_k - laisvai pasirenkami svoriai.

2. Svoriai ω_k tenkina lygtį

$$\sum_{k \in \mathbf{i}} \omega_k \mathbf{x}_k = \mathbf{t}_x,$$

   vadinamą kalibravimo lygtimi.

Populiacijos kintamojo suma yra vienas paprastesnių vertinamų parametrų. Svorių kalibravimo idėja dažniau naudojama vertinant sudėtingesnius populiacijos parametrus, tokius kaip kovariacija, santykis ir pan. Apie tai rašoma, atitinkamai, [8] ir [4],[7] straipsniuose.

1.5 Paprastasis atsitiktinis ėmimas

Paprastoji atsitiktinė imtis, arba ***paprastoji atsitiktinė negrąžintinė imtis***, - tai tokia n skirtingų elementų imtis iš N dydžio baigtinės populiacijos, kai bet kuris n skirtingų elementų rinkinys turi vienodą tikimybę būti išrinktas. Tai vienas paprasčiausių ir dažnai taikomų ėmimo būdų.

Iš N dydžio populiacijos $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ galima išrinkti C_N^n tokių imčių. Tikimybė, kad kiekvienas n skirtingų elementų rinkinys $\mathbf{i} = \{u_{i_1}, ..., u_{i_n}\}$ bus išrinktas iš N

dydžio populiacijos, lygi $\frac{1}{C_N^n}$. Taigi, paprastosios atsitiktinės imties plano tikimybės $p(\mathbf{i}) = \frac{1}{C_N^n}$ visoms galimoms $\mathbf{i}$.

Tikimybė, kad k-asis populiacijos elementas priklausys kuriai nors n dydžio paprastajai atsitiktinei imčiai yra

$$\pi_k = \mathbf{P}(\mathbf{i} : k \in \mathbf{i}) = \frac{n}{N}, \quad k = 1, 2, ..., N.$$

Tikimybė, kad du populiacijos elementai k ir l priklauso paprastajai atsitiktinei imčiai,

$$\pi_{kl} = \mathbf{P}(k \in \mathbf{i}, l \in \mathbf{i}) = \mathbf{P}(k \in \mathbf{i})\mathbf{P}(l \in \mathbf{i} | k \in \mathbf{i}) = \frac{n}{N}\frac{n-1}{N-1}, \quad k, l = 1, ..., N, \quad k \neq l.$$

Paprastosios atsitiktinės negrąžintinės imties atveju (1) galime užrašyti

$$\hat{t}_\pi = \frac{N}{n}\sum_{k=1}^{n} y_k,$$

kuris vadinamas standartiniu Horvico ir Tompsono įvertiniu.

Paprastosios atsitiktinės imties išrinkimo būdų yra daug: išrinkimas pagal apibrėžimą, nuoseklus išrinkimas, išrinkimas naudojant atsitiktinių skaičių lentelę ar atsitiktinių skaičių generatorių ir pan.

1.6 Sluoksninis ėmimas. Sluoksniavimo metodai

Tarkime turime N dydžio baigtinę populiaciją $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ iš kurios norime išrinkti n elementų dydžio imtį, o $x : x_1, x_2, ..., x_N$ - populiacijoje $\mathcal{U}$ apibrėžtas papildomas kintamasis, be to, $x_1 < x_2 < ... < x_N$. Esant tokioms sąlygoms, galime taikyti sluoksninį ėmimą. ***Sluoksniniu ėmimu*** arba ***sluoksnine imtimi*** vadinami tokie imties planai, kai suskaidžius populiaciją į kelias, bendrų elementų neturinčias, dalis, žymimas $U_1, U_2, ..., U_H$ ir vadinamas sluoksniais, imtys renkamos iš kiekvieno sluoksnio atskirai, nepriklausomai nuo kitų. Sluoksnių skaičių žymėsime raide H. Skirtingiems sluoksniams gali būti taikomi skirtingi imčių planai ir parametrų vertinimo būdai. Populiacijos skaidymas į sluoksnius, kuriuos sudaro panašūs elementai, leidžia išvengti blogų imčių ir tikėtis tikslesnių įverčių.

Taikant sluoksninės imties planą, iškyla du pagrindiniai klausimai: kaip nustatyti sluoksnių ribas, žymimas k_0, k_1, ..., k_{H-1}, k_H ir kaip paskirstyti imtį į sluoksnius. Nustatyti sluoksnių ribas padeda įvairūs sluoksniavimo metodai, kuriuos trumpai aptarsime.

- **Šaknies iš f metodas**. Taikant šį metodą, daroma prielaida, kad kiekviename sluoksnyje kintamasis x turi tolygųjį skirstinį. Jo santykinių dažnių funkciją žymėsime $f(x)$. T. Danelius ir J. Hodges [15] parodė, kad sluoksnių ribos $k_h^{(f)}$ parenkamos taip, kad sumos

  $$\sum_{i \in U_h} \sqrt{f(x_i)}, \quad h = 1, 2, ..., H$$

  būtų apytiksliai lygios.

- **Geometrinis metodas**. Šį metodą pasiūlė P. Gunning ir J. M. Horgan [2]. Taikant šį metodą, daroma prielaida, kad papildomas kintamasis x kiekviename sluoksnyje yra tolygiai pasiskirstęs. Tada apytiksliai optimalios sluoksnių ribos gaunamos pagal formulę:

  $$k_h = k_0 r^h, \quad r = \left(\frac{k_H}{k_0}\right)^{\frac{1}{H}}, \quad h = 1, 2, ..., H.$$

  Šis metodas vadinamas ***geometriniu metodu***, kadangi sluoksnių ribos sudaro geometrinę progresiją.

- **Laipsnių metodas**. Šis metodas, aprašomas [10] straipsnyje, yra paprastas ir efektyvus. Taikant laipsnių metodą, sluoksnių ribos $k_h^{(p)}$ parenkamos taip, kad sumos

  $$\sum_{i \in U_h} x_i^{\alpha} = const, \quad h = 1, 2, ..., H$$

  būtų apytiklsiai lygios. Deja, šis metodas nėra teoriškai pagrįstas. Jo privalumai parodomi modeliuojant. Daugelis eksperimentų rodo, kad $\alpha \in [0.5; 0.7]$.

Nustačius sluoksnių ribas, lieka atsakyti į klausimą, kaip optimaliai paskirstyti imties dydį į turimus sluoksnius. Atsakymą mums duoda ***Neimano optimaliojo imties paskirstymo principas***, kuris teigia, jog optimalusis imties dydžio paskirstymas į sluoksnius yra po

$$n_h = n \frac{N_h s_h}{\sum_{h=1}^{H} N_h s_h}, \quad h = 1, 2, ..., H,$$

elementų, čia n - imties dydis; N_h - h-ojo sluoksnio dydis; s_h - h-ojo sluoksnio standartinis nuokrypis.

1.7 Teiloro ištiesinimo metodas

Svarbią imčių teorijos dalį sudaro populiacijos parametrų įvertinių dispersijos vertinimas. Dispersijos įvertiniai naudojami parametrų įvertinių tikslumui nustatyti. Tačiau kartais šie įvertiniai yra sudėtingi, nes sukonstruoti atsižvelgiat į sudėtingą imties planą bei naudojantis papildoma informacija. Todėl, vertinant tokių įvertinių dispersijas, tenka naudoti sudėtingas formules, o kartais iš viso neturime dispersijos įvertinio išraiškos. Tad natūralu, jog ieškoma metodų, kurie padėtų išspręsti susidariusią problemą. Vieną tokių metodų aptarsime šiame poskyryje.

Nagrinėkime q tyrimo kintamųjų $y^{(1)}, y^{(2)}, ..., y^{(q)}$, apibrėžtų baigtinėje populiacijoje $\mathcal{U} = \{u_1, u_2, ..., u_N\}$, ir populiacijos parametrą

$$\theta = \theta(y^{(1)}, y^{(2)}, ..., y^{(q)}).$$

Imsime tik tokius parametro θ įvertinius, kurie yra išreikšti sumų

$$t_i = \sum_{k=1}^{N} y_k^{(i)}, \quad i = 1, 2, ..., q,$$

įvertinių $\hat{t}_1, ..., \hat{t}_q$ funkcijomis $\hat{\theta} = g(\hat{t}_1, ..., \hat{t}_q)$, kai $\hat{t}_i$ turi pavidalą

$$\hat{t}_i = \sum_{k \in \mathbf{i}} d_k y_k^{(i)}.$$

Tarkime, kad parametro θ įvertinys $\hat{\theta} = g(\hat{t}_1, ..., \hat{t}_q)$ yra glodžioji sumų funkcija, t.y funkcija, tolydi taško $(t_1, ..., t_q)$ aplinkoje ir turi tame taške dalines išvestines. Tokiam įvertiniui taikome Teiloro formulę taške $(t_1, ..., t_q)$, imdami tik pirmosios eilės narius:

$$\hat{\theta} = g(\hat{t}_1, ..., \hat{t}_q) \approx a_0 + \sum_{i=1}^{q} a_i (\hat{t}_i - t_i), \tag{5}$$

čia

$$a_0 = g(t_1, ..., t_q),$$
$$a_i = \left.\frac{\partial g(\hat{t}_1, ..., \hat{t}_q)}{\partial \hat{t}_i}\right|_{(\hat{t}_1, ..., \hat{t}_q) = (t_1, ..., t_q)}, \quad i = 1, ..., q.$$

(5) įvertinys vadinamas ***Teiloro ištiesintuoju įvertiniu***.

Parametro įvertinio $\hat{\theta}$ (5) skleidinio pirmosios eilės narių dispersiją vadinsime šio įvertinio apytiksle dispersija ir žymėsime

$$A\mathbf{D}\hat{\theta} = \mathbf{D}\left(\sum_{i=1}^{q} a_i (\hat{t}_i - t_i)\right).$$

Toks apytikslės dispersijos skaičiavimo metodas vadinamas ***Teiloro ištiesinimo*** metodu.

2 NAUJI HORVICO IR TOMPSONO DISPERSIJOS ĮVERTINIAI

2.1 Įvertinių svorių kalibravimas

Nagrinėkime baigtinę N dydžio populiaciją $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ ir joje apibrėžtą tyrimo kintamąjį $y : y_1, y_2, ..., y_N$ bei papildomą kintamąjį $x : x_1, x_2, ..., x_N$. Vertinamasis populiacijos parametras - Horvico ir Tompsono įvertinio dispersija, kurios standartinis įvertinys užrašomas dviem pavidalais.

Pirmąjį pavidalą gavo Yates & Grundy [14] parodę, kad, fiksavus imties dydį, (2) įvertinio dispersiją galima užrašyti

$$\mathbf{D}_{YG}(\widehat{Y}_{HT}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j - \pi_{ij})\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2,$$

o dispersijos įvertinį

$$\widehat{\mathbf{D}}_{YG}(\widehat{Y}_{HT}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \tag{6}$$

čia $d_{ij} = \frac{\pi_i\pi_j - \pi_{ij}}{\pi_{ij}}$ žymi svorius.

Antrąjį pavidalą gavo Wu & Sitter [15] parodę, kad (2) įvertinio dispersiją galima užrašyti

$$\mathbf{D}_{MC} = \sum_{i=1}^{N}\sum_{j>1}\varphi(y_i, y_j),$$

o dispersijos įvertinį

$$\widehat{\mathbf{D}}_{MC} = \sum_{i=1}^{n}\sum_{j>1} d_{ij}\varphi(y_i, y_j), \tag{7}$$

čia $d_{ij} = \frac{1}{\pi_{ij}}$ - svoriai, $\varphi(y_i, y_j) = (\pi_i\pi_j - \pi_{ij})\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2$.

Taigi, (6) ir (7) išraiškos yra laikomos Horvico ir Tompsono įvertinio dispersijos standartiniais įvertiniais.

Remiantis J. Deville & C. Särndal [14] pasiūlyta svorių kalibravimo idėja, modifikuosime (6) įvertinio svorius ir sukonstruosime keturis naujus kalibruotuosius Horvico ir Tompsono įvertinio dispersijos įvertinius pavidalo

$$\widehat{\mathbf{D}}^{(k)}(\widehat{Y}_{HT}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}^{(k)}\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \quad k = 1, ..., 4 \tag{8}$$

čia $\omega_{ij}^{(k)}$ - kalibruotieji įvertinio svoriai, tenkinantys sąlygas:

1. minimizuoja atstumo funkciją

$$L_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{(\omega_{ij}^{(k)} - d_{ij})^2}{d_{ij}q_{ij}}, \quad k = 1,4, \tag{9}$$

$$L_2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{q_{ij}}\left(\frac{\omega_{ij}^{(k)}}{d_{ij}} - 1\right)^2, \quad k = 2,3, \tag{10}$$

2. tenkina kalibravimo lygtį

$$\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}^{(k)}\left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 - \sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j - \pi_{ij})\left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2\right) = 0, \quad k = 1,2,3,4, \tag{11}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}^{(k)} = 0, \quad k = 2,4. \tag{12}$$

Tokiu pat būdu sukonstruosime keturis kalibruotuosius Horvico ir Tompsono įvertinio dispersijos įvertinius pavidalo

$$\widehat{\mathbf{D}}_{MC}^{(k)} = \sum_{i=1}^{n}\sum_{j>1}\omega_{ij}^{(k)}\varphi(y_i, y_j), \quad \varphi(y_i, y_j) = (\pi_i\pi_j - \pi_{ij})\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \quad k = 1, ..., 4, \tag{13}$$

kurie gaunami modifikuojant (7) įvertinio svorius. Dydžiai ω_{ij}, esantys (13) formulėje, tenkina sąlygas:

1. minimizuoja atstumo funkciją

$$L_3 = \sum_{i=1}^{n}\sum_{j>i}\frac{(\omega_{ij}^{(k)} - d_{ij})^2}{d_{ij}q_{ij}}, \quad k = 1,2, \tag{14}$$

$$L_4 = \sum_{i=1}^{n}\sum_{j>i}\frac{1}{q_{ij}}\left(\frac{\omega_{ij}^{(k)}}{d_{ij}} - 1\right)^2, \quad k = 3,4, \tag{15}$$

2. tenkina kalibravimo lygtį

$$\frac{1}{N^*}\sum_{i=1}^{n}\sum_{j>i}\omega_{ij}^{(k)} - 1 = 0, \quad N^* = \frac{N(N-1)}{2}, \quad k = 1,4, \tag{16}$$

$$\sum_{i=1}^{n}\sum_{j>i}\omega_{ij}^{(k)}\varphi(\hat{y}_i, \hat{y}_j) - \sum_{i=1}^{N}\sum_{j>i}\varphi(\hat{y}_i, \hat{y}_j) = 0, \quad k = 1,2,3,4. \tag{17}$$

2 teiginys. Kalibruotųjų įvertinių $\widehat{\mathbf{D}}^{(k)}(\hat{Y}_{HT})$, $k = 1,3$ svoriai $\omega_{ij}^{(k)}$, tenkinantys kalibravimo lygtį (11) ir minimizuojantys atstumo funkciją L_p, $p = 1,2$, apibrėžtą (9, 10) lygybėmis, yra atitinkamai lygūs:

$$\omega_{ij}^{(1)} = \frac{\lambda^{(1)}}{2}d_{ij}q_{ij}\left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 + d_{ij},$$

$$\omega_{ij}^{(3)} = \frac{\lambda^{(3)}}{4} d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 + d_{ij}$$

čia $\lambda^{(k)}$ - Lagranžo daugikliai.

Įrodymas. Imkime atvejį, kai $k = 1,\ p = 1$ ir taikykime Lagranžo neapibrėžtųjų daugiklių metodą. Sudarykime Lagranžo funkciją

$$\Lambda = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(\omega_{ij}^{(1)} - d_{ij})^2}{d_{ij} q_{ij}} - \frac{\lambda^{(1)}}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{ij}^{(1)} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 - \sum_{i=1}^{N} \sum_{j=1}^{N} (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \right).$$

Funkcijos Λ dalinės išvestinės svorių $\omega_{ij}^{(1)}$ atžvilgiu

$$\frac{\partial \Lambda}{\partial \omega_{ij}^{(1)}} = \frac{\omega_{ij}^{(1)} - d_{ij}}{d_{ij} q_{ij}} - \frac{\lambda^{(1)}}{2} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2.$$

Prilyginę jas nuliui, išreiškiame kalibruotojo įvertinio $\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$ svorius

$$\omega_{ij}^{(1)} = \frac{\lambda^{(1)}}{2} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 + d_{ij}. \tag{18}$$

Matome, kad šie svoriai priklauso nuo Lagranžo daugiklio $\lambda^{(1)}$. Norėdami jį rasti, (18) įstatome į (11). Sutvarkę narius gauname, kad

$$\lambda^{(1)} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^4} \tag{19}$$

Įrašę (19) išraišką į (18) lygybes, gausime kalibruotojo įvertinio $\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$ svorių išraiškas.

Atvejis, kai $k = 3,\ p = 2$ įrodomas analogiškai.

Įrodyta.

3 teiginys. Kalibruotųjų įvertinių $\widehat{\mathbf{D}}^{(k)}(\widehat{Y}_{HT}),\ k = 2,4$ svoriai $\omega_{ij}^{(k)}$, tenkinantys kalibravimo lygtis (11), (12) ir minimizuojantys atstumo funkciją L_p, $p = 1,2$, apibrėžtą (9, 10) lygybėmis, yra atitinkamai lygūs:

$$\omega_{ij}^{(2)} = \frac{\lambda_1^{(2)}}{4} d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 + \frac{\lambda_2^{(2)}}{2} d_{ij}^2 q_{ij} + d_{ij},$$

$$\omega_{ij}^{(4)} = \frac{\lambda_1^{(4)}}{2} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 + \lambda_2^{(4)} d_{ij} q_{ij} + d_{ij}$$

čia $\lambda^{(k)}$ - Lagranžo daugikliai.

Įrodymas. Imkime atvejį, kai $k = 4$, $p = 1$ ir taikykime Lagranžo neapibrėžtųjų daugiklių metodą. Sudarykime Lagranžo funkciją

$$\Lambda = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(\omega_{ij}^{(4)} - d_{ij})^2}{d_{ij} q_{ij}} - \frac{\lambda_1^{(4)}}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{ij}^{(4)} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 - \right.$$
$$\left. - \sum_{i=1}^{N} \sum_{j=1}^{N} (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 \right) - \lambda_2^{(4)} \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_{ij}^{(4)}.$$

Funkcijos Λ dalinės išvestinės svorių $\omega_{ij}^{(4)}$ atžvilgiu

$$\frac{\partial \Lambda}{\partial \omega_{ij}^{(4)}} = \frac{\omega_{ij}^{(4)} - d_{ij}}{d_{ij} q_{ij}} - \frac{\lambda_1^{(4)}}{2} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 - \lambda_2^{(4)}.$$

Prilyginę jas nuliui, išreiškiame kalibruotojo įvertinio $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$ svorius

$$\omega_{ij}^{(4)} = \frac{\lambda_1^{(4)}}{2} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 + \lambda_2^{(4)} d_{ij} q_{ij} + d_{ij}. \tag{20}$$

Matome, kad šie svoriai priklauso nuo Lagranžo daugiklių $\lambda_1^{(4)}$ ir $\lambda_2^{(4)}$. Rasime šiuos daugiklius.

Pirmiausia rasime $\lambda_1^{(4)}$. Tuo tikslu, (20) įstatome į (11). Turime

$$\frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\lambda_1^{(4)}}{2} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^4 + \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_2^{(4)} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 + \right.$$
$$\left. + \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 - \sum_{i=1}^{N} \sum_{j=1}^{N} (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 \right) = 0. \tag{21}$$

Tada, (20) įstatome į (12) ir pertvarkę lygtį, išsireiškiame $\lambda_1^{(4)}$:

$$\lambda_1^{(4)} = -\frac{2}{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} + \lambda_2^{(4)} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \right) \tag{22}$$

Ieškome $\lambda_2^{(4)}$. Tuo tikslu, (22) įrašome į (21) lygtį. Sutvarkę narius turime

$$\lambda_2^{(4)} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2}{\left(\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2 \right)^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^4}.$$

$$\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2+\frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}\cdot\right.$$

$$\left.\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^4-\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2\right). \tag{23}$$

Įrašę (23) į (22) gauname galutinę $\lambda_1^{(4)}$ išraišką. Įstatę (22) ir (23) išraiškas į (20) lygybes, gausime kalibruotojo įvertinio $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$ svorių išraiškas.

Atvejis, kai $k=2,\ p=2$ įrodomas analogiškai.

Įrodyta.

4 teiginys. Kalibruotųjų įvertinių $\widehat{\mathbf{D}}_{MC}^{(k)}$, $k=2,3$ svoriai $\omega_{ij}^{(k)}$, tenkinantys kalibravimo lygtį (17) ir minimizuojantys atstumo funkciją L_p, $p=3,4$, apibrėžtą (14, 15) lygybėmis, yra atitinkamai lygūs:

$$\omega_{ij}^{(2)}=\frac{\lambda^{(2)}}{2}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij}+d_{ij},$$

$$\omega_{ij}^{(3)}=\frac{\lambda^{(3)}}{2}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}^2q_{ij}+d_{ij}$$

čia $\lambda^{(k)}$ - Lagranžo daugikliai.

Įrodymas. Imkime atvejį, kai $k=2,\ p=3$ ir taikykime Lagranžo neapibrėžtųjų daugiklių metodą. Sudarykime Lagranžo funkciją

$$\Lambda=\sum_{i=1}^{n}\sum_{j>i}\frac{(\omega_{ij}^{(2)}-d_{ij})^2}{d_{ij}q_{ij}}-\lambda^{(2)}\left(\sum_{i=1}^{n}\sum_{j>i}\omega_{ij}^{(2)}\varphi(\hat{y}_i,\hat{y}_j)-\sum_{i=1}^{N}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)\right).$$

Funkcijos Λ dalinės išvestinės svorių $\omega_{ij}^{(2)}$ atžvilgiu

$$\frac{\partial\Lambda}{\partial\omega_{ij}^{(2)}}=\frac{2(\omega_{ij}^{(2)}-d_{ij})}{d_{ij}q_{ij}}-\lambda^{(2)}\varphi(\hat{y}_i,\hat{y}_j).$$

Prilyginę jas nuliui, išreiškiame kalibruotojo įvertinio $\widehat{\mathbf{D}}_{MC}^{(2)}$, svorius

$$\omega_{ij}^{(2)}=\frac{\lambda^{(2)}}{2}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij}+d_{ij}. \tag{24}$$

Matome, kad šie svoriai priklauso nuo Lagranžo daugiklio $\lambda^{(2)}$. Norėdami jį rasti, (24) įstatome į (17). Sutvarkę narius gauname, kad

$$\lambda^{(2)}=\frac{2}{\sum_{i=1}^{n}\sum_{j>i}\varphi^2(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij}}\left(\sum_{i=1}^{N}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)-\sum_{i=1}^{n}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}\right). \tag{25}$$

Įrašę (25) išraišką į (24) lygybes, gausime kalibruotojo įvertinio $\widehat{\mathbf{D}}_{MC}^{(2)}$ svorių išraiškas.

Atvejis, kai $k=3,\ p=4$ įrodomas analogiškai.

Įrodyta.

5 teiginys. Kalibruotųjų įvertinių $\widehat{\mathbf{D}}_{MC}^{(k)}$, $k=1,4$ svoriai $\omega_{ij}^{(k)}$, tenkinantys kalibravimo lygtis (16), (17) ir minimizuojantys atstumo funkciją L_p, $p=3,4$, apibrėžtą (14, 15) lygybėmis, yra atitinkamai lygūs:

$$\omega_{ij}^{(1)} = \frac{\lambda_1^{(1)}}{2N^*} d_{ij}q_{ij} + \frac{\lambda_2^{(1)}}{2}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij} + d_{ij},$$

$$\omega_{ij}^{(4)} = \frac{\lambda_1^{(4)}}{2N^*} d_{ij}^2 q_{ij} + \lambda_2^{(4)}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}^2 q_{ij} + d_{ij}$$

čia $\lambda^{(k)}$ - Lagranžo daugikliai.

Įrodymas. Imkime atvejį, kai $k=1,\ p=3$ ir taikykime Lagranžo neapibrėžtųjų daugiklių metodą. Sudarykime Lagranžo funkciją

$$\Lambda = \sum_{i=1}^{n}\sum_{j>i}\frac{(\omega_{ij}^{(1)}-d_{ij})^2}{d_{ij}q_{ij}} - \lambda_1^{(1)}\left(\frac{1}{N^*}\sum_{i=1}^{n}\sum_{j>i}\omega_{ij}^{(1)} - 1\right) - \lambda_2^{(1)}\left(\sum_{i=1}^{n}\sum_{j>i}\omega_{ij}^{(1)}\varphi(\hat{y}_i,\hat{y}_j) - \sum_{i=1}^{N}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)\right).$$

Funkcijos Λ dalinės išvestinės svorių $\omega_{ij}^{(1)}$ atžvilgiu

$$\frac{\partial\Lambda}{\partial\omega_{ij}^{(1)}} = 2\frac{\omega_{ij}^{(1)}-d_{ij}}{d_{ij}q_{ij}} - \frac{\lambda_1^{(1)}}{N^*} - \lambda_2^{(1)}\varphi(\hat{y}_i,\hat{y}_j).$$

Prilyginę jas nuliui, išreiškiame kalibruotojo įvertinio $\widehat{\mathbf{D}}_{MC}^{(1)}$ svorius

$$\omega_{ij}^{(1)} = \frac{\lambda_1^{(1)}}{2N^*} d_{ij}q_{ij} + \frac{\lambda_2^{(1)}}{2}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij} + d_{ij}. \tag{26}$$

Matome, kad šie svoriai priklauso nuo Lagranžo daugiklių $\lambda_1^{(1)}$ ir $\lambda_2^{(1)}$. Rasime šiuos daugiklius.

Pirmiausia rasime $\lambda_1^{(1)}$. Tuo tikslu, (26) įstatome į (16) ir išsireiškiame $\lambda_1^{(1)}$:

$$\lambda_1^{(1)} = \frac{2N^*}{\sum_{i=1}^{n}\sum_{j>i}d_{ij}q_{ij}}\left(N^* - \frac{\lambda_2^{(1)}}{2}\sum_{i=1}^{n}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij} - \sum_{i=1}^{n}\sum_{j>i}d_{ij}\right) \tag{27}$$

Ieškome $\lambda_2^{(1)}$. Tuo tikslu, (26) įrašome į (17) lygtį. Turime

$$\frac{\lambda_1^{(1)}}{2N^*}\sum_{i=1}^{n}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij} + \frac{\lambda_2^{(1)}}{2}\sum_{i=1}^{n}\sum_{j>i}\varphi^2(\hat{y}_i,\hat{y}_j)d_{ij}q_{ij} + \sum_{i=1}^{n}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j)d_{ij} - \sum_{i=1}^{N}\sum_{j>i}\varphi(\hat{y}_i,\hat{y}_j) = 0. \tag{28}$$

Įstatome (27) į (28) ir sutraukę panašiuosius išsireiškiame $\lambda_2^{(1)}$

$$\lambda_2^{(1)} = \frac{2\sum_{i=1}^n \sum_{j>i} d_{ij}q_{ij}}{\sum_{i=1}^n \sum_{j>i} d_{ij}q_{ij} \cdot \sum_{i=1}^n \sum_{j>i} \varphi^2(\hat{y}_i,\hat{y}_j) d_{ij}q_{ij} - \left(\sum_{i=1}^n \sum_{j>i} \varphi(\hat{y}_i,\hat{y}_j) d_{ij}q_{ij}\right)^2}\cdot$$

$$\cdot\left(\frac{1}{\sum_{i=1}^n \sum_{j>i} d_{ij}q_{ij}}\left(\sum_{i=1}^n \sum_{j>i} d_{ij} \cdot \sum_{i=1}^n \sum_{j>i} \varphi(\hat{y}_i,\hat{y}_j) d_{ij}q_{ij} - N^* \sum_{i=1}^n \sum_{j>i} \varphi(\hat{y}_i,\hat{y}_j) d_{ij}q_{ij}\right) - \right.$$
$$\left. - \sum_{i=1}^n \sum_{j>i} \varphi(\hat{y}_i,\hat{y}_j) d_{ij} + \sum_{i=1}^n \sum_{j>i} \varphi(\hat{y}_i,\hat{y}_j)\right). \tag{29}$$

Įrašę (29) į (27) gauname galutinę $\lambda_1^{(1)}$ išraišką. Įstatę (27) ir (29) išraiškas į (26) lygybes, gausime kalibruotojo įvertinio $\widehat{\mathbf{D}}_{MC}^{(1)}$ svorių išraiškas.

Atvejis, kai $k = 4,\ p = 4$ įrodomas analogiškai.

Įrodyta.

2.2 Įvertinių dispersijos vertinimas

Darbe sukonstruotų įvertinių kalibruotieji svoriai, kaip matėme, turi sudėtingas išraiškas, todėl apskaičiuoti tikslią įvertinio dispersiją yra sudėtinga. Šiam procesui palengvinti buvo sukurti metodai, kuriuos naudojant randama ne tiksli, bet apytikslė įvertinio dispersija. Vienas tokių metodų - Teiloro ištiesinimo metodas, aprašytas 1.7 skyrelyje.

Šiame poskyryje, taikydami minėtą Teiloro ištiesinimo metodą, rasime sukonstruoto Horvico ir Tompsono įvertinio dispersijos įvertinio $\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$ apytikslę dispersiją ir pasiūlysime šios dispersijos įvertinį.

6 teiginys. Horvico ir Tompsono įvertinio dispersijos įvertinio

$$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}) = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \omega_{ij}^{(2)} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \tag{30}$$

kurio kalibruotieji svoriai $\omega_{ij}^{(2)}$ tenkina (11), (12) kalibravimo lygtis ir minimizuoja (10) atstumo funkciją, apytikslė dispersija yra

$$A\mathbf{D}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})) = \sum_{k=1}^{N^*} \left(\frac{1-\widetilde{\pi}_k}{\widetilde{\pi}_k}\right)\widetilde{V}_k^2 + \sum_{k=1}^{N^*}\sum_{\substack{l=1\\ l\neq k}}^{N^*} \left(\frac{\widetilde{\pi}_{kl} - \widetilde{\pi}_k\widetilde{\pi}_l}{\widetilde{\pi}_k\widetilde{\pi}_l}\right)\widetilde{V}_k\widetilde{V}_l,$$

o dispersijos įvertinys

$$\widehat{\mathbf{D}}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})) = \sum_{k=1}^{n^*} \left(\frac{1-\widetilde{\pi}_k}{\widetilde{\pi}_k^2} \right) \widetilde{V}_k^2 + \sum_{k=1}^{n^*} \sum_{\substack{l=1 \\ l \neq k}}^{n^*} \left(\frac{\widetilde{\pi}_{kl} - \widetilde{\pi}_k \widetilde{\pi}_l}{\widetilde{\pi}_k \widetilde{\pi}_l \widetilde{\pi}_{kl}} \right) \widetilde{V}_k \widetilde{V}_l.$$

Teiginio formuluotėje naudojami dydžiai N^*, n^*, $\widetilde{\pi}_k$, $\widetilde{\pi}_{kl}$, $\widetilde{V}_k$ palaipsniui apibrėžiami įrodyme.

Įrodymas. (30) įvertinys, įrašius svorių $\boldsymbol{\omega}_{ij}^{(2)}$ išraišką, yra

$$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}) = \frac{1}{2(\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} (\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j})^4 - (\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} (\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j})^2)^2)} \cdot$$

$$\cdot \Bigg[\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \cdot \sum_{i=1}^N \sum_{j=1}^N (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 -$$

$$- \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 +$$

$$+ \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 -$$

$$- \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^N \sum_{j=1}^N (\pi_i \pi_j - \pi_{ij}) \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 -$$

$$- \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^4 +$$

$$+ \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 -$$

$$- \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \left(\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \right)^2 +$$

$$+ \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2 \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \cdot \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^4 \Bigg].$$

Pažymėkime

$$\hat{t}_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij}, \quad \hat{t}_2 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2, \quad \hat{t}_3 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^4,$$

$$\hat{t}_4 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2, \quad \hat{t}_5 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}, \quad \hat{t}_6 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j} \right)^2 \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2,$$

$$\hat{t}_7 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2, \quad \hat{t}_8 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 q_{ij} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j} \right)^2,$$

$$t_9 = \sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j - \pi_{ij})\left(\frac{x_i}{\pi_i} - \frac{x_j}{\pi_j}\right)^2.$$

Tada įvertinį $\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$ galime užrašyti

$$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}) = \frac{\hat{t}_6\hat{t}_1 t_9 - \hat{t}_6\hat{t}_1\hat{t}_2 + \hat{t}_6\hat{t}_5\hat{t}_4 - \hat{t}_8 t_9\hat{t}_4 - \hat{t}_8\hat{t}_5\hat{t}_3 + \hat{t}_8\hat{t}_2\hat{t}_4 - \hat{t}_7\hat{t}_4^2 + \hat{t}_7\hat{t}_1\hat{t}_3}{2(\hat{t}_1\hat{t}_3 - \hat{t}_4^2)}. \tag{31}$$

Taško $(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8) = (t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)$ aplinkoje įvertinį (31) aproksimuojame tiesine Teiloro eilutės dalimi ir gauname

$$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}) \approx a_0 + a_1(\hat{t}_1 - t_1) + a_2(\hat{t}_2 - t_2) + a_3(\hat{t}_3 - t_3) + a_4(\hat{t}_4 - t_4) + a_5(\hat{t}_5 - t_5) + a_6(\hat{t}_6 - t_6) +$$
$$+ a_7(\hat{t}_7 - t_7) + a_8(\hat{t}_8 - t_8),$$

čia

$$a_0 = \widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8) = \frac{t_6t_1t_9 - t_6t_1t_2 + t_6t_5t_4 - t_8t_9t_4 - t_8t_5t_3 + t_8t_2t_4 - t_7t_4^2 + t_7t_1t_3}{2(t_1t_3 - t_4^2)},$$

$$a_1 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_1}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_6t_2t_4^2 - t_6t_9t_4^2 - t_3t_6t_5t_4 - t_3t_8t_2t_4 + t_3t_8t_9t_4 + t_8t_5t_3^2}{2(t_1t_3 - t_4^2)^2},$$

$$a_2 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_2}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_8t_4 - t_1t_6}{2(t_1t_3 - t_4^2)},$$

$$a_3 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_3}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_8t_5t_4^2 + t_6t_1^2t_2 - t_6t_1^2t_9 - t_1t_6t_5t_4 - t_1t_8t_2t_4 + t_1t_8t_9t_4}{2(t_1t_3 - t_4^2)^2},$$

$$a_4 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_4}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{1}{2(t_1t_3 - t_4^2)^2}(t_6t_5t_4^2 + t_6t_5t_1t_3 + t_8t_2t_4^2 + t_8t_2t_1t_3 -$$
$$- t_8t_9t_4^2 - t_8t_9t_1t_3 - 2t_4t_6t_1t_2 + 2t_4t_6t_1t_9 - 2t_4t_8t_5t_3),$$

$$a_5 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_5}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_8t_3 - t_6t_4}{2(t_1t_3 - t_4^2)},$$

$$a_6 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_6}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_1t_9 + t_5t_4 - t_1t_2}{2(t_1t_3 - t_4^2)},$$

$$a_7 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_7}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{1}{2},$$

$$a_8 = \frac{\partial\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})}{\partial\hat{t}_8}\bigg|_{(\hat{t}_1,\hat{t}_2,\hat{t}_3,\hat{t}_4,\hat{t}_5,\hat{t}_6,\hat{t}_7,\hat{t}_8)=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8)} = \frac{t_2t_4 - t_4t_9 - t_5t_3}{2(t_1t_3 - t_4^2)},$$

$$t_1=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})d_{ij}q_{ij},\quad t_2=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2,$$

$$t_3=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^4,\quad t_4=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2,$$

$$t_5=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij}),\quad t_6=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2,$$

$$t_7=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2,\quad t_8=\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi_i\pi_j-\pi_{ij})d_{ij}q_{ij}\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2.$$

Tuomet $\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$ įvertinio apytikslė dispersija

$$A\mathbf{D}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}))=\mathbf{D}\left(\sum_{h=1}^{8}a_h\hat{t}_h\right)=\mathbf{D}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}u_{ij}\right)=\mathbf{D}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\pi_{ij}}V_{ij}\right),\qquad(32)$$

čia

$$u_{ij}=a_1d_{ij}q_{ij}+a_2\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2+a_3d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^4+a_4d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2+a_5+$$

$$+a_6d_{ij}q_{ij}\left(\frac{x_i}{\pi_i}-\frac{x_j}{\pi_j}\right)^2\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2+a_7\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2+a_8d_{ij}q_{ij}\left(\frac{y_i}{\pi_i}-\frac{y_j}{\pi_j}\right)^2,$$

$$V_{ij}=(\pi_i\pi_j-\pi_{ij})u_{ij}.$$

Imkime visas galimas populiacijos $\mathcal{U}$ elementų poras $m=(ij)$, $i,j=1,2,...,N$, $m=1,2,...,N^*$, $N^*=N^2$. Tokiu būdu sudarome naują N^* dydžio elementų populiaciją $\mathcal{U}^*=\{u_1^*,u_2^*,...,u_{N^*}^*\}$, kurią vadinsime sintetine populiacija. Iš jos išrenkame $n^*=n^2$ dydžio imtį $\mathbf{i}^*=\{m=(ij),i,j\in\mathbf{i}\}$.

Tada (32) galima užrašyti

$$A\mathbf{D}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}))=\mathbf{D}\left(\sum_{m=1}^{n^*}d_m^*V_m^*\right)=\mathbf{D}\left(\sum_{m=1}^{n^*}\widetilde{d}_m\widetilde{V}_m\right),$$

čia $\widetilde{d}_m$ - svoriai populiacijoje $\mathcal{U}^*$, $\widetilde{V}_m=\frac{1}{\widetilde{d}_m}d_m^*V_m^*$, $d_m^*=\frac{1}{\pi_m^*}=\frac{1}{\pi_{ij}}$, $V_m^*=V_{ij}$.

Remiantis Horvico ir Tompsono įvertinio dispersijos išraiška (3) turime, kad $\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$ įvertinio apytikslė dispersija yra

$$A\mathbf{D}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT}))=\sum_{k=1}^{N^*}\left(\frac{1-\widetilde{\pi}_k}{\widetilde{\pi}_k}\right)\widetilde{V}_k^2+\sum_{k=1}^{N^*}\sum_{\substack{l=1\\l\neq k}}^{N^*}\left(\frac{\widetilde{\pi}_{kl}-\widetilde{\pi}_k\widetilde{\pi}_l}{\widetilde{\pi}_k\widetilde{\pi}_l}\right)\widetilde{V}_k\widetilde{V}_l,$$

o pagal (4), dispersijos įvertinys

$$\widehat{\mathbf{D}}(\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})) = \sum_{k=1}^{n^*}\left(\frac{1-\widetilde{\pi}_k}{\widetilde{\pi}_k^2}\right)\widetilde{V}_k^2 + \sum_{k=1}^{n^*}\sum_{\substack{l=1\\ l\neq k}}^{n^*}\left(\frac{\widetilde{\pi}_{kl}-\widetilde{\pi}_k\widetilde{\pi}_l}{\widetilde{\pi}_k\widetilde{\pi}_l\widetilde{\pi}_{kl}}\right)\widetilde{V}_k\widetilde{V}_l,$$

čia $\widetilde{\pi}_k = \frac{n^*}{N^*}$, $\widetilde{\pi}_{kl} = \frac{n^*(n^*-1)}{N^*(N^*-1)}$.

Įrodyta.

Kitų kalibruotųjų Horvico ir Tompsono įvertinio dispersijos įvertinių apytikslės dispersijos ir jų įvertiniai gaunami analogiškai.

3 MATEMATINIS MODELIAVIMAS

Matematinis modeliavimas yra taikomosios matematikos dalis, skirta įvairių sričių uždavinių sprendimui naudojant virtualiojo eksperimento metodiką. Ji remiasi matematinių modelių sudarymu ir eksperimentinių rezultatų apdorojimu. Matematinis modeliavimas leidžia be didelių išlaidų ir pakankamai greitai atlikti tiriamo objekto savybių tyrimą įvairiausiose situacijose.

Šiame darbe matematinis modeliavimas atliekamas naudojant kompiuterinę programą MATLAB. Šiuo matematiniu paketu parašytos programos (darbo pabaigoje pateikiamos kaip priedas) apskaičiuoja naujai sukonstruotų Horvico ir Tompsono įvertinio dispersijos įvertinių tikslumo matus bei gautų įvertinių Teiloro ištiesintų įvertinių empirinę dispersiją. Gauti eksperimentų rezultatai pateikiami toliau.

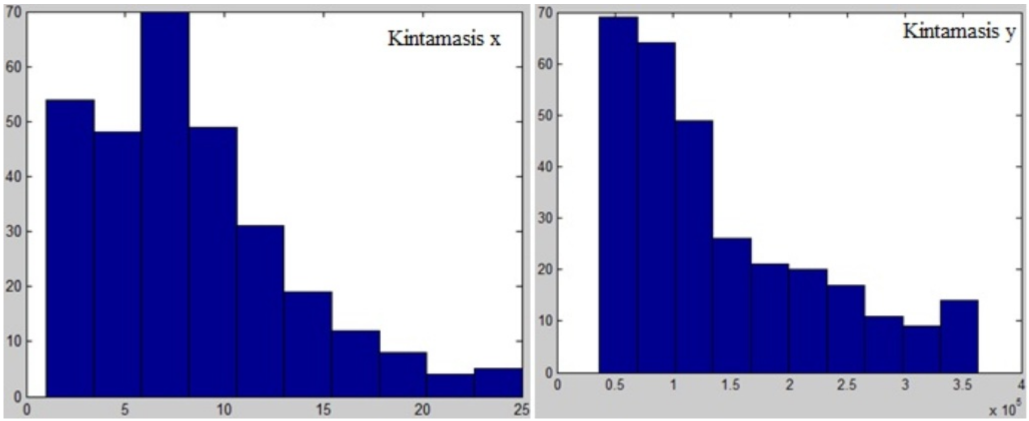
3.1 Tiriamos populiacijos

Eksperimentiniame tyrime buvo naudojamos keturios realios baigtinės populiacijos, kurias trumpai pristatysime.

Pažymėkime $\rho(y,x)$ - kintamųjų y, x koreliacijos koeficientą; μ - atitinkamo kintamojo vidurkį; s^2 - atitinkamo kintamojo dispersiją.

1 populiacija. Baigtinę populiaciją sudaro $N = 300$ elementų. Šioje populiacijoje apibrėžtas vienas tyrimo kintamasis y ir papildomas kintamasis x tokie, kad
$\rho(y,x) = 0.603$, o $\mu_y = 609.552$, $s_y^2 = 1196163$, $\mu_x = 8.427$, $s_x^2 = 26.628$.

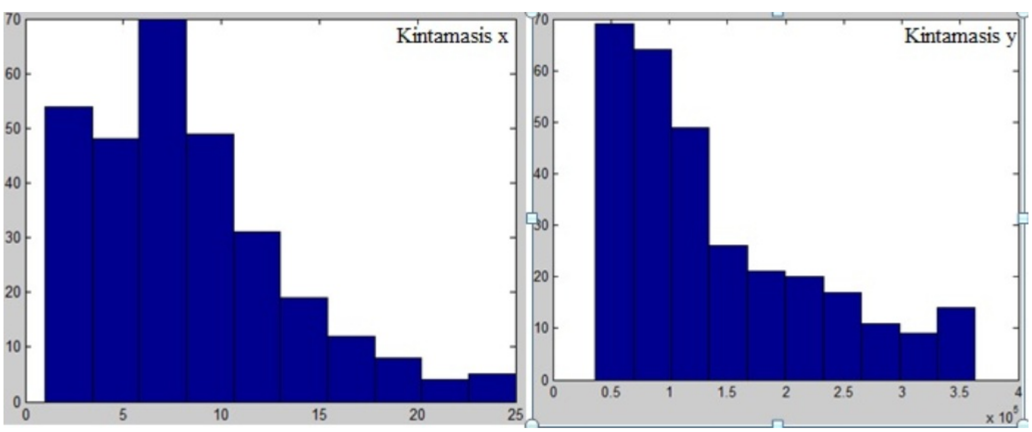
1 paveikslėlyje pateikiamos populiacijos kintamųjų histogramos, parodančios kintamųjų skirstinių formą.



1 pav.: Pirmos populiacijos kintamųjų histogramos

2 populiacija. Baigtinę populiaciją sudaro $N = 300$ elementų. Šioje populiacijoje apibrėžtas vienas tyrimo kintamasis y ir papildomas kintamasis x tokie, kad $\rho(y,x) = 0.807$, o $\mu_y = 139383.4$, $s_y^2 = 7439506720$, $\mu_x = 8.247$, $s_x^2 = 26.628$.

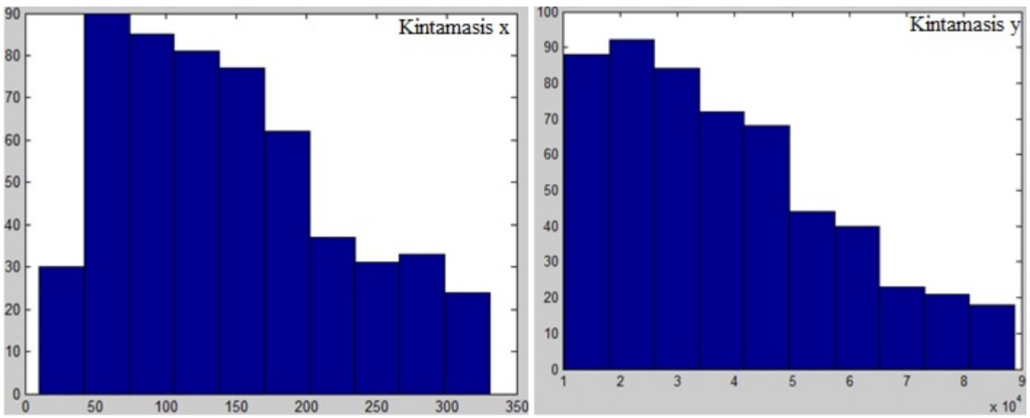
2 paveikslėlyje pateikiamos populiacijos kintamųjų histogramos, parodančios kintamųjų skirstinių formą.



2 pav.: Antros populiacijos kintamųjų histogramos

3 populiacija. Baigtinę populiaciją sudaro $N = 550$ elementų. Šioje populiacijoje apibrėžtas vienas tyrimo kintamasis y ir papildomas kintamasis x tokie, kad $\rho(y,x) = 0.804$, o $\mu_y = 38408.27$, $s_y^2 = 380242978$, $\mu_x = 144.673$, $s_x^2 = 6125.543$.

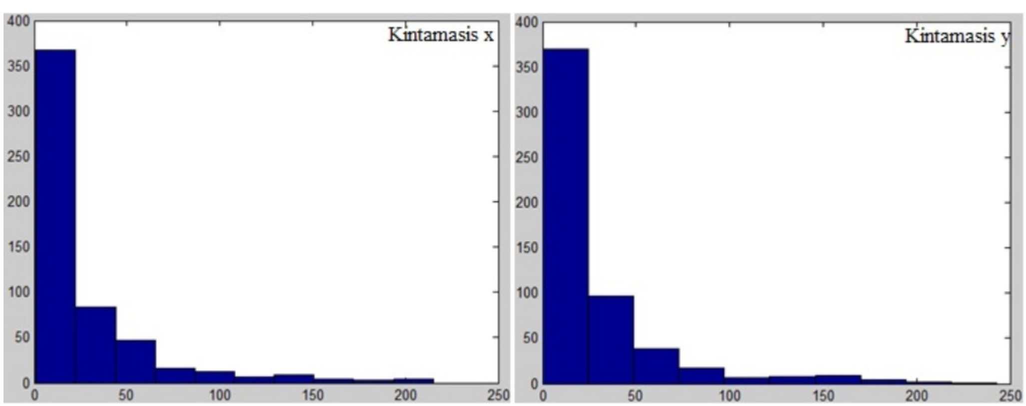
3 paveikslėlyje pateikiamos populiacijos kintamųjų histogramos, parodančios kintamųjų skirstinių formą.



3 pav.: Trečios populiacijos kintamųjų histogramos

4 populiacija. Baigtinę populiaciją sudaro $N = 554$ elementų. Šioje populiacijoje apibrėžtas vienas tyrimo kintamasis y ir papildomas kintamasis x tokie, kad
$\rho(y,x) = 0.971$, o $\mu_y = 27.022$, $\quad s_y^2 = 1326.423$, $\mu_x = 27.417$, $\quad s_x^2 = 1298.555$.

4 paveikslėlyje pateikiamos populiacijos kintamųjų histogramos, parodančios kintamųjų skirstinių formą.



4 pav.: Ketvirtos populiacijos kintamųjų histogramos

3.2 Įvertinių tikslumo matų palyginimas

Modeliuojant sukonstruoti nauji įvertiniai palyginami tarpusavyje ir su standartiniu Horvico ir Tompsono įvertinio dispersijos įvertiniu

$$\widehat{\mathbf{D}}(\widehat{Y}_{HT}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \quad d_{ij} = \frac{\pi_i\pi_j - \pi_{ij}}{\pi_{ij}}.$$

Buvo įvertinti šie įvertinių tikslumo matai, apskaičiuojami pagal formules:
įvertinio vidurkis

$$\widehat{\mathbf{E}}\widehat{\theta} = \frac{1}{\mathbf{s}}\sum_{k=1}^{\mathbf{s}} \widehat{\theta}^{(\mathbf{i}_k)},$$

poslinkis

$$\widehat{\mathbf{POSL}}(\widehat{\theta}) = \widehat{\mathbf{E}}\widehat{\theta} - \theta,$$

dispersija

$$\widehat{\mathbf{D}}\widehat{\theta} = \frac{1}{\mathbf{s}-1}\sum_{k=1}^{\mathbf{s}} \left(\widehat{\theta}^{(\mathbf{i}_k)} - \widehat{\mathbf{E}}\widehat{\theta}\right)^2,$$

vidutinė kvadratinė paklaida

$$\widehat{\mathbf{VKP}}(\widehat{\theta}) = \widehat{\mathbf{D}}\widehat{\theta} + (\widehat{\mathbf{POSL}}(\widehat{\theta}))^2,$$

ir variacijos koeficientas

$$\widehat{\mathbf{cv}}(\widehat{\theta}) = \frac{\sqrt{\widehat{\mathbf{D}}\widehat{\theta}}}{\widehat{\mathbf{E}}\widehat{\theta}}.$$

Formulėse esantis dydis $\mathbf{s}$ - tai viso renkamų imčių skaičius; $\widehat{\theta}$ - parametro θ įvertinys.

Pagrindiniai įvertinio tikslumą (kokybę) apibūdinantys parametrai yra įvertinio poslinkis ir dispersija. **VKP** yra abu šiuos parametrus vienu metu apibūdinanti charakteristika, todėl lygindami įvertinių tikslumą būtent jai skirsime didžiausią dėmesį. Taip pat atsižvelgsime ir į **cv**, nes kuo šis koeficientas mažesnis, tuo tikslesnis įvertinys.

I eksperimentas. Kiekvienam įvertiniui renkama po 1000 paprastųjų atsitiktinių negrąžintinių imčių, kai imties dydis $n = 100$. Kiekvienai imčiai apskaičiuojami Horvico ir Tompsono įvertinio dispersijos įverčiai. Naudojantis gautais rezultatais, įvertinami pagrindiniai įvertinių tikslumo matai (žr. 1 lentelė).

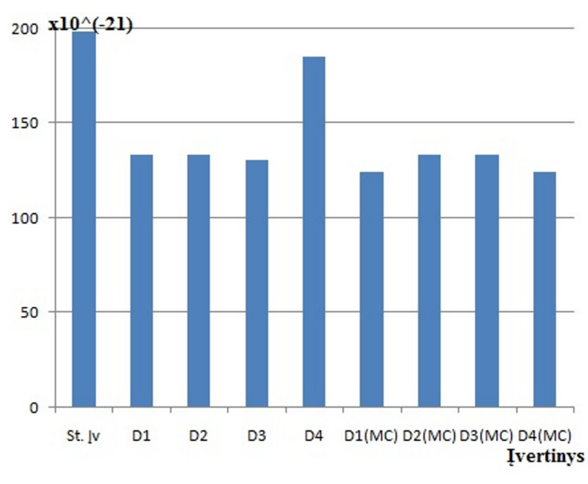
1 lentelė: ***Pagrindiniai įvertinių tikslumo matai***

Įvertinys	**E** $\times 10^{-10}$	**POSL** $\times 10^{-7}$	**D** $\times 10^{-21}$	**VKP** $\times 10^{-21}$	**cv**
1 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	445.59	-776.02	391.77	391.83	0.1405
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	448.34	1971.90	267.01	267.40	0.1152
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	448.33	1961.20	266.82	267.20	0.1152
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	446.97	597.96	260.39	260.42	0.1142
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	451.93	5563.20	369.35	372.45	0.1345
$\widehat{\mathbf{D}}_{MC}^{(1)}$	446.80	424.77	248.48	248.50	0.1116
$\widehat{\mathbf{D}}_{MC}^{(2)}$	448.34	1971.90	267.01	267.40	0.1152
$\widehat{\mathbf{D}}_{MC}^{(3)}$	448.34	1971.90	267.01	267.40	0.1152
$\widehat{\mathbf{D}}_{MC}^{(4)}$	446.80	424.77	248.48	248.50	0.1116
2 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	445.46	-914.64	386.97	387.05	0.1396
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	447.02	645.41	256.88	256.93	0.1134
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	447.01	636.79	256.70	256.74	0.1133
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	446.24	-134.61	252.76	252.76	0.1127
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	449.97	3603.40	355.28	356.57	0.1325
$\widehat{\mathbf{D}}_{MC}^{(1)}$	445.79	-577.64	238.96	238.99	0.1096
$\widehat{\mathbf{D}}_{MC}^{(2)}$	447.02	645.41	256.88	256.93	0.1134
$\widehat{\mathbf{D}}_{MC}^{(3)}$	447.02	645.41	256.88	256.93	0.1134
$\widehat{\mathbf{D}}_{MC}^{(4)}$	445.79	-577.64	238.96	238.99	0.1096
3 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	94.39	276.64	13.74	13.75	0.1242
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	94.37	258.49	9.62	9.63	0.1039
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	94.37	258.20	9.62	9.62	0.1039
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	94.38	267.56	9.79	9.80	0.1048
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	94.52	413.32	11.31	11.33	0.1125
$\widehat{\mathbf{D}}_{MC}^{(1)}$	94.35	239.72	9.26	9.26	0.1020
$\widehat{\mathbf{D}}_{MC}^{(2)}$	94.37	258.49	9.62	9.63	0.1039
$\widehat{\mathbf{D}}_{MC}^{(3)}$	94.37	258.49	9.62	9.63	0.1039
$\widehat{\mathbf{D}}_{MC}^{(4)}$	94.35	239.72	9.26	9.26	0.1020

4 populiacija					
Įvertinys	**E** $\times 10^{-4}$	**POSL** $\times 10^{-2}$	**D** $\times 10^{-10}$	**VKP** $\times 10^{-10}$	**cv**
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	333.84	21.95	105.53	1055.30	0.3077
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	337.55	393.58	13.04	131.95	0.1070
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	337.55	393.83	13.04	131.96	0.1070
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	336.42	280.25	12.85	129.28	0.1065
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	335.69	207.77	34.24	342.79	0.1743
$\widehat{\mathbf{D}}_{MC}^{(1)}$	337.85	423.68	13.12	132.99	0.1072
$\widehat{\mathbf{D}}_{MC}^{(2)}$	337.55	393.58	13.04	131.95	0.1070
$\widehat{\mathbf{D}}_{MC}^{(3)}$	337.55	393.58	13.04	131.95	0.1070
$\widehat{\mathbf{D}}_{MC}^{(4)}$	337.85	423.68	13.12	132.99	0.1072

Pagal charakteristikas **VKP** ir **cv** matome, kad esant I - ojo eksperimento sąlygoms, naujieji Horvico ir Tompsono įvertinio dispersijos įvertiniai yra tikslesni nei standartinis Horvico ir Tompsono įvertinio dispersijos įvertinys. Lyginant naujuosius įvertinius tarpusavyje, turime, kad visų įvertinių, išskyrus $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$, tikslumas panašus. Taip pat matome, jog $\widehat{\mathbf{D}}_{MC}^{(2)}$ įvertinio tikslumo matai sutampa su $\widehat{\mathbf{D}}_{MC}^{(3)}$ įvertinio charakteristikomis, o $\widehat{\mathbf{D}}_{MC}^{(1)}$ su $\widehat{\mathbf{D}}_{MC}^{(4)}$. Šių įvertinių sutapimą lemia tai, kad renkamos paprastosios atsitiktinės negrąžintinės imtys, kurių atveju įvertiniuose svoriai $d_k = \frac{N}{n}$, todėl, nepaisant atstumo funkcijos, kalibruotų įvertinių, sukonstruotų naudojantis ta pačia kalibravimo lygtimi, įverčiai sutampa.

Pažvelgus į 5 pav. matome, kad mažiausią **VKP** turi $\widehat{\mathbf{D}}_{MC}^{(1)}$ ir $\widehat{\mathbf{D}}_{MC}^{(4)}$ įvertiniai.



5 pav.: I eksperimento įvertinių VKP vidurkiai

Taigi, norėdami įvertinti Horvico ir Tompsono įvertinio dispersiją, esant paprastajam atsitiktiniam negrąžintiniam ėmimui, galima naudoti bet kurį vieną iš naujų kalibruotųjų įvertinių, tačiau tiksliausi yra $\widehat{\mathbf{D}}_{MC}^{(1)}$ ir $\widehat{\mathbf{D}}_{MC}^{(4)}$ įvertiniai.

II eksperimentas. Atliekant šį eksperimentą buvo naudojamas sluoksninis ėmimas, kai populiacija suskaidoma į keturis sluoksnius, o sluoknių ribos nustatomos taikant šaknies iš f metodą. Kiekvienam įvertiniui renkama po 1000 sluoksninių paprastųjų atsitiktinių negrąžintinių imčių, kai imties dydis $n = 100$. Kiekvienai imčiai apskaičiuojami Horvico ir Tompsono įvertinio dispersijos įverčiai. Naudojantis gautais rezultatais, įvertinami pagrindiniai įvertinių tikslumo matai (žr. 2 lentelė).

2 lentelė: ***Pagrindiniai įvertinių tikslumo matai***

Įvertinys	**E** $\times 10^{-9}$	**POSL** $\times 10^{-7}$	**D** $\times 10^{-18}$	**VKP** $\times 10^{-18}$	**cv**
1 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	265.82	-368.98	650.18	663.79	0.0959
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	268.82	-68.61	822.07	822.54	0.1067
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	268.74	-77.06	799.77	800.36	0.1052
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	267.32	-219.17	664.53	669.34	0.0964
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	274.13	461.88	2137.00	2158.30	0.1686
$\widehat{\mathbf{D}}_{MC}^{(1)}$	268.13	-137.60	761.63	763.53	0.1029
$\widehat{\mathbf{D}}_{MC}^{(2)}$	268.52	-99.39	808.84	809.83	0.1059
$\widehat{\mathbf{D}}_{MC}^{(3)}$	268.82	-69.37	807.70	808.18	0.1057
$\widehat{\mathbf{D}}_{MC}^{(4)}$	268.20	-131.41	748.15	749.88	0.1020
2 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	266.00	-350.80	595.54	607.84	0.0917
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	269.73	22.56	842.67	842.72	0.1076
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	269.71	19.90	823.11	823.15	0.1064
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	267.90	-161.05	647.73	650.33	0.0950
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	275.97	646.12	2232.20	2273.90	0.1712
$\widehat{\mathbf{D}}_{MC}^{(1)}$	268.89	-62.03	767.87	768.25	0.1031
$\widehat{\mathbf{D}}_{MC}^{(2)}$	269.36	-15.39	822.66	822.68	0.1065
$\widehat{\mathbf{D}}_{MC}^{(3)}$	269.80	28.70	832.52	832.61	0.1070
$\widehat{\mathbf{D}}_{MC}^{(4)}$	269.08	-43.24	760.27	760.46	0.1025

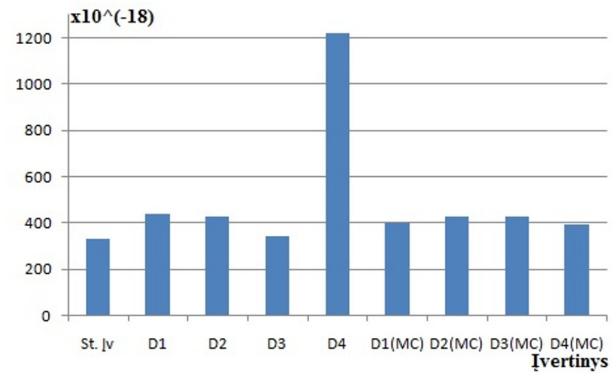
3 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	71.10	-51.31	48.09	48.35	0.0975
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	73.66	205.31	78.11	82.32	0.1200
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	73.57	196.27	74.99	78.85	0.1177
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	72.36	75.26	52.43	52.99	0.1000
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	77.37	575.85	411.15	444.31	0.2621
$\widehat{\mathbf{D}}_{MC}^{(1)}$	73.28	167.09	70.43	73.22	0.1145
$\widehat{\mathbf{D}}_{MC}^{(2)}$	73.60	199.21	79.36	83.33	0.1210
$\widehat{\mathbf{D}}_{MC}^{(3)}$	73.63	201.82	76.29	80.37	0.1186
$\widehat{\mathbf{D}}_{MC}^{(4)}$	73.24	163.20	67.76	70.43	0.1124
4 populiacija					
Įvertinys	**E** $\times 10^{-3}$	**POSL** $\times 10^{-2}$	**D** $\times 10^{-6}$	**VKP** $\times 10^{-7}$	**cv**
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	152.15	16.39	339.71	34.24	0.1211
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	174.96	211.67	683.19	113.13	0.1494
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	171.17	173.75	726.64	102.85	0.1575
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	161.66	78.705	354.36	41.63	0.1164
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	168.52	147.28	1683.90	190.08	0.2435
$\widehat{\mathbf{D}}_{MC}^{(1)}$	178.09	242.94	759.47	134.97	0.1548
$\widehat{\mathbf{D}}_{MC}^{(2)}$	178.03	242.38	764.65	135.22	0.1553
$\widehat{\mathbf{D}}_{MC}^{(3)}$	171.17	173.80	731.76	103.38	0.1580
$\widehat{\mathbf{D}}_{MC}^{(4)}$	170.99	172.02	694.16	99.01	0.1541

Nagrinėjant charakteristiką **cv** matome, kad esant II - ojo eksperimento sąlygoms, naujieji Horvico ir Tompsono įvertinio dispersijos įvertiniai, beveik visais atvejais, nėra tikslesni už standartinį Horvico ir Tompsono įvertinio dispersijos įvertinį. Įvertinys $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ savo tikslumu, pirmose trijose populiacijose, yra artimiausias standartiniam įvertiniui $\widehat{\mathbf{D}}(\widehat{Y}_{HT})$, o ketvirtoje populiacijoje - tikslesnis. Likusieji įvertiniai, išskyrus $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$, savo tikslumu panašūs.

Pagal 2 lentelės duomenis matome, kad nėra sutampančių įvertinių, nes sluoksninio ėmimo atveju $d_k \neq \frac{N}{n}$.

Taip pat pastebime, kad naudojant ketvirtos populiacijos duomenis įvertiniai duoda didžiausią netikslumą. Šį netikslumą įtakoja populiacijos duomenų pasiskirstymas.

Nagrinėjant įvertinių **VKP** (žr. 6 pav.) matome, kad šiuo atveju mažiausia **VKP** yra standartinio įvertinio, o didžiausia - $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$.



6 pav.: II eksperimento įvertinių VKP vidurkiai

Taigi, naudojant šaknies iš f sluoksniavimo metodą ir norėdami įvertinti Horvico ir Tompsono įvertinio dispersiją, geriau naudoti standartinį šios dispersijos įvertinį, o iš naujų sukonstruotų kalibruotųjų įvertinių - $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$.

III eksperimentas. Atliekant šį eksperimentą buvo naudojamas sluoksninis ėmimas, kai populiacija suskaidoma į keturis sluoksnius, o sluoknių ribos nustatomos taikant geometrinį metodą. Kiekvienam įvertiniui renkama po 1000 sluoksninių paprastųjų atsitiktinių negrąžintinių imčių, kai imties dydis $n = 100$. Kiekvienai imčiai apskaičiuojami Horvico ir Tompsono įvertinio dispersijos įverčiai. Naudojantis gautais rezultatais, įvertinami pagrindiniai įvertinių tikslumo matai (žr. 3 lentelė).

3 lentelė: ***Pagrindiniai įvertinių tikslumo matai***

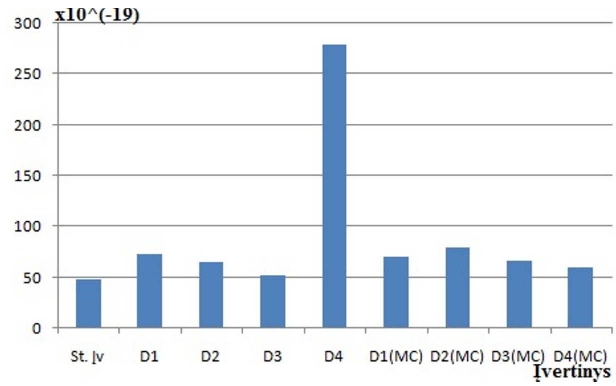
Įvertinys	**E** $\times 10^{-9}$	**POSL** $\times 10^{-7}$	**D** $\times 10^{-18}$	**VKP** $\times 10^{-19}$	**cv**
1 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	265.66	-252.45	979.10	98.55	0.1178
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	275.15	696.92	1345.80	139.44	0.1333
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	273.17	498.5	1261.10	128.60	0.1300
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	269.54	135.76	1035.20	103.71	0.1194
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	284.53	1634.30	4885.70	515.28	0.2457
$\widehat{\mathbf{D}}_{MC}^{(1)}$	275.06	687.22	1313.60	136.08	0.1318
$\widehat{\mathbf{D}}_{MC}^{(2)}$	276.28	809.40	1449.70	151.52	0.1378
$\widehat{\mathbf{D}}_{MC}^{(3)}$	273.42	523.97	1280.30	130.78	0.1309
$\widehat{\mathbf{D}}_{MC}^{(4)}$	271.85	366.63	1169.70	118.32	0.1258
2 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	265.09	-309.22	862.82	87.24	0.1108
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	274.14	595.63	1352.60	138.81	0.1342
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	272.45	426.74	1207.10	122.53	0.1275
$\widehat{V}^{(3)}(\widehat{Y}_{HT})$	268.9	71.52	953.46	95.40	0.1148
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	281.55	1336.70	5263.40	544.21	0.2577
$\widehat{\mathbf{D}}_{MC}^{(1)}$	273.95	576.87	1320.00	135.33	0.1326
$\widehat{\mathbf{D}}_{MC}^{(2)}$	275.04	685.24	1482.20	152.91	0.1400
$\widehat{\mathbf{D}}_{MC}^{(3)}$	272.71	452.26	1228.90	124.94	0.1285
$\widehat{\mathbf{D}}_{MC}^{(4)}$	271.11	292.94	1103.10	111.16	0.1225
3 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	82.84	-55.15	70.52	7.08	0.1014
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	85.32	192.87	120.50	12.42	0.1287
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	85.34	194.95	113.67	11.75	0.1249
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	84.12	72.75	83.99	8.45	0.1090
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	88.04	465.21	528.35	55.00	0.2611
$\widehat{\mathbf{D}}_{MC}^{(1)}$	85.05	165.44	111.52	11.43	0.1242
$\widehat{\mathbf{D}}_{MC}^{(2)}$	85.34	194.79	125.71	12.95	0.1314
$\widehat{\mathbf{D}}_{MC}^{(3)}$	85.40	200.65	115.66	11.97	0.1259
$\widehat{\mathbf{D}}_{MC}^{(4)}$	85.04	164.64	103.72	10.64	0.1198

4 populiacija					
Įvertinys	**E** $\times 10^{-3}$	**POSL** $\times 10^{-1}$	**D** $\times 10^{-6}$	**VKP** $\times 10^{-6}$	**cv**
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	231.50	-464.27	1280.20	1301.80	0.1545
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	234.29	-185.55	765.05	768.49	0.1181
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	234.54	-160.33	793.62	796.19	0.1201
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	233.02	-312.65	882.99	892.76	0.1275
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	233.90	-224.42	797.38	802.41	0.1207
$\widehat{\mathbf{D}}_{MC}^{(1)}$	234.70	-144.88	771.49	773.59	0.1183
$\widehat{\mathbf{D}}_{MC}^{(2)}$	234.22	-192.89	756.92	760.64	0.1175
$\widehat{\mathbf{D}}_{MC}^{(3)}$	234.53	-161.02	794.28	796.87	0.1202
$\widehat{\mathbf{D}}_{MC}^{(4)}$	234.66	-148.43	793.85	796.05	0.1201

Gavome, kad esant III - ojo eksperimento sąlygoms, lyginant charakteristiką **cv**, naujieji Horvico ir Tompsono įvertinio dispersijos įvertiniai, pirmose trijose populiacijose, nėra tikslesni už standartinį Horvico ir Tompsono įvertinio dispersijos įvertinį, o ketvirtoje populiacijoje - tikslesni. Įvertinys $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ savo tikslumu, pirmose trijose populiacijose, yra artimiausias standartiniam įvertiniui $\widehat{\mathbf{D}}(\widehat{Y}_{HT})$. Likusieji įvertiniai, išskyrus $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$, savo tikslumu panašūs.

Kaip jau minėjau, ketvirtoje populiacijoje naujieji įvertiniai tikslesnis už standartinį Horvico ir Tompsono įvertinį. Todėl galime teigti, kad taikant geometrinį sluoksniavimo metodą, populiacijos duomenų pasiskirstymas įtakoja įvertinių tikslumą.

Pagal charakteristiką **VKP** (žr. 7 pav.) turime, kad tiksliausią įvertį duoda standartinis Horvico ir Tompsono įvertinio dispersijos įvertinys, o netiklsiausias - $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$.



7 pav.: III eksperimento įvertinių VKP vidurkiai

Taigi, naudojant geometrinį sluoksniavimo metodą ir norėdami įvertinti Horvico ir Tompsono įvertinio dispersiją, turėtume atsižvelgti į populiacijos duomenų pasiskirstymą. Daugeliu atvejų geriau naudoti standartinį šios dispersijos įvertinį, tačiau nereikėtų atmesti ir naujų sukonstruotų kalibruotųjų įvertinių.

IV eksperimentas. Atliekant šį eksperimentą buvo naudojamas sluoksninis ėmimas, kai populiacija suskaidoma į keturis sluoksnius, o sluoknių ribos nustatomos taikant laipsnių metodą ($\alpha = 0.6$). Kiekvienam įvertiniui renkama po 1000 sluoksninių paprastųjų atsitiktinių negrąžintinių imčių, kai imties dydis $n = 100$. Kiekvienai imčiai apskaičiuojami Horvico ir Tompsono įvertinio dispersijos įverčiai. Naudojantis gautais rezultatais, įvertinami pagrindiniai įvertinių tikslumo matai (žr. 4 lentelė).

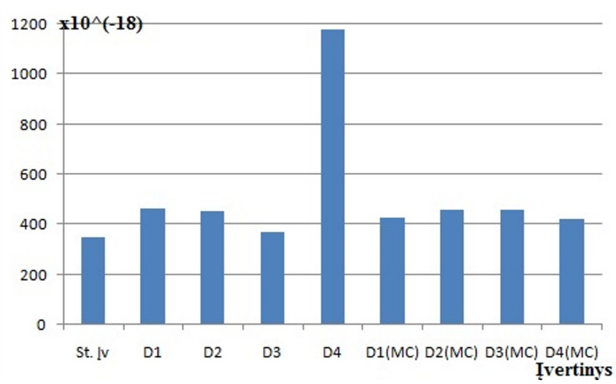
4 lentelė: ***Pagrindiniai įvertinių tikslumo matai***

Įvertinys	**E** $\times 10^{-9}$	**POSL** $\times 10^{-7}$	**D** $\times 10^{-18}$	**VKP** $\times 10^{-18}$	**cv**
1 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	262.27	-421.45	657.00	674.76	0.0977
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	265.02	-146.56	883.74	885.89	0.1122
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	265.1	-138.53	867.83	869.75	0.1111
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	263.72	-276.62	703.97	711.62	0.1006
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	269.49	300.31	2171.30	2180.30	0.1729
$\widehat{\mathbf{D}}_{MC}^{(1)}$	264.45	-203.16	821.03	825.16	0.1084
$\widehat{\mathbf{D}}_{MC}^{(2)}$	264.79	-169.92	875.15	878.04	0.1117
$\widehat{\mathbf{D}}_{MC}^{(3)}$	265.17	-131.80	877.10	878.84	0.1117
$\widehat{\mathbf{D}}_{MC}^{(4)}$	264.62	-186.43	806.67	810.15	0.1073
2 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	262.60	-388.28	654.37	669.44	0.0974
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	266.73	24.15	893.58	893.64	0.1121
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	266.73	24.40	878.28	878.34	0.1111
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	264.71	-177.26	709.07	712.21	0.1006
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	273.05	656.46	2173.30	2216.40	0.1707
$\widehat{\mathbf{D}}_{MC}^{(1)}$	265.97	-51.73	826.82	827.08	0.1081
$\widehat{\mathbf{D}}_{MC}^{(2)}$	266.47	-1.91	882.09	882.09	0.1115
$\widehat{\mathbf{D}}_{MC}^{(3)}$	266.82	33.76	887.77	887.89	0.1117
$\widehat{\mathbf{D}}_{MC}^{(4)}$	266.06	-42.25	815.14	815.31	0.1073
3 populiacija					
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	71.14	-10.24	44.79	44.80	0.0941
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	72.47	123.59	69.35	70.87	0.1149
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	72.32	108.60	64.51	65.69	0.1110
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	71.74	50.26	48.76	49.02	0.0973
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	74.34	310.27	311.61	321.23	0.2374
$\widehat{\mathbf{D}}_{MC}^{(1)}$	72.22	98.08	62.44	63.40	0.1094
$\widehat{\mathbf{D}}_{MC}^{(2)}$	72.35	111.42	69.07	70.33	0.1149
$\widehat{\mathbf{D}}_{MC}^{(3)}$	72.35	110.76	65.43	66.66	0.1118
$\widehat{\mathbf{D}}_{MC}^{(4)}$	72.20	96.09	59.44	60.36	0.1068

4 populiacija					
Įvertinys	**E** $\times 10^{-3}$	**POSL** $\times 10^{-2}$	**D** $\times 10^{-7}$	**VKP** $\times 10^{-7}$	**cv**
$\widehat{\mathbf{D}}(\widehat{Y}_{HT})$	126.74	12.36	24.50	24.66	0.1235
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	181.25	532.77	156.41	440.25	0.2182
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	179.19	512.18	149.90	412.23	0.2161
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	153.07	250.91	48.47	111.43	0.1438
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	180.60	526.21	468.99	745.88	0.3792
$\widehat{\mathbf{D}}^{(1)}_{MC}$	183.04	550.60	165.46	468.62	0.2222
$\widehat{\mathbf{D}}^{(2)}_{MC}$	183.65	556.72	167.12	477.06	0.2226
$\widehat{\mathbf{D}}^{(3)}_{MC}$	179.39	514.18	150.92	415.30	0.2166
$\widehat{\mathbf{D}}^{(4)}_{MC}$	177.50	495.27	144.25	389.54	0.2140

Gavome, kad esant IV - ojo eksperimento sąlygoms, naujieji Horvico ir Tompsono įvertinio dispersijos įvertiniai nėra tikslesni už standartinį Horvico ir Tompsono įvertinio dispersijos įvertinį. Įvertiniai $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ ir $\widehat{\mathbf{D}}^{(1)}_{MC}$ savo tikslumu, pirmose trijose populiacijose, yra artimiausi standartiniam įvertiniui $\widehat{\mathbf{D}}(\widehat{Y}_{HT})$. Likusieji įvertiniai, išskyrus $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$, savo tikslumu panašūs. Iš 4 lentelės matome, kad ketvirtoje populiacijoje naujieji įvertiniai, pagal **cv**, yra patys netiksliausi. Todėl galime teigti, kad taikant laipsninį sluoksniavimo metodą, populiacijos duomenų pasiskirstymas įtakoja įvertinių tikslumą.

Iš sukonstruotų įvertinių, pagal charakteristiką **VKP** (žr. 8 pav.), pats tiksliausias yra $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$, o pats netiksliausias įvertinys $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$.



8 pav.: IV eksperimento įvertinių VKP vidurkiai

Taigi, naudojant laipsninį sluoksniavimo metodą ir norėdami įvertinti Horvico ir Tompsono įvertinio dispersiją geriausia naudoti standartinį šios dispersijos įvertinį.

Apžvelgus visų eksperimentų rezultatus, ypač charakteristiką **VKP**, rodančią įvertinio kokybę, galime teigti, kad kalibruotieji Horvico ir Tompsono įvertinio dispersijos įvertiniai tiksliausi tada, kai taikomas sluoksninis ėmimas, o sluoksnių ribos nustatomos pagal šaknies iš f taisyklę.

Iš darbe sukonstruotų dviejų pavidalų kalibruotųjų Horvico ir Tompsono įvertinio dispersijos įvertinių, tiksliausi yra $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ ir $\widehat{\mathbf{D}}^{(4)}_{MC}$.

3.3 Įvertinių dispersijų palyginimas

Eksperimentui buvo naudojama baigtinė $N = 300$ dydžio populiacija. Viso buvo renkama 1000 paprastųjų atsitiktinių negrąžintinių $n = 100$ dydžio imčių. Apskaičiuotos sukonstruotų Horvico ir Tompsono įvertinio dispersijos įvertinių ir Teiloro ištiesintų įvertinų empirinės dispersijos. Gauti rezultatai pateikiami 5 lentelėje.

5 lentelė: ***Įvertinių empirinės dispersijos***

Įvertinys	**Empirinė dispersija** $\times 10^{-23}$	**Teiloro ištiesintojo įvertinio empirinė dispersija** $\times 10^{-23}$
$\widehat{\mathbf{D}}^{(1)}(\widehat{Y}_{HT})$	2.6449	2.4775
$\widehat{\mathbf{D}}^{(2)}(\widehat{Y}_{HT})$	2.6428	1.464
$\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$	2.4775	2.4535
$\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$	3.7069	3.3518
$\widehat{\mathbf{D}}^{(1)}_{MC}$	1.0137	0.71408
$\widehat{\mathbf{D}}^{(2)}_{MC}$	1.1161	1.0065
$\widehat{\mathbf{D}}^{(3)}_{MC}$	1.1161	1.0065
$\widehat{\mathbf{D}}^{(4)}_{MC}$	1.0137	0.71408

Matome, kad Teiloro ištiesintų įvertinių empirinė dispersija artima pradinių įvertinių empirinei dispersijai. Mažiausias skirtumas pastebimas tarp $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ įvertinio empirinių dispersijų, didžiausias - tarp $\widehat{\mathbf{D}}^{(4)}(\widehat{Y}_{HT})$, $\widehat{\mathbf{D}}_{MC}^{(1)}$ ir $\widehat{\mathbf{D}}_{MC}^{(4)}$ įvertinių empirinių dispersijų. Skirtumą tarp dispersijų įtakoja tai, kad imama tik tiesinė Teiloro eilutės dalis. Kadangi šis skirtumas yra nedidelis, tai galime teigti, kad atsisakydami netiesinės Teiloro eilutės dalies nedaug ką prarandame.

Apžvelgę gautus rezultatus, galime teigti, kad vietoj pradinio įvertinio dispersijos galime imti Teiloro ištiesintojo įvertinio dispersiją, kurios skaičiavimas yra paprastesnis ir reikalauja mažiau laiko.

IŠVADOS IR REZULTATAI

- Naudojant Lagranžo neapibrėžtųjų daugiklių metodą ir taikant skirtingas kalibravimo lygtis bei atstumo funkcijas, išvesti dviejų tipų nauji Horvico ir Tompsono įvertinio dispersijos įvertiniai.

- Naudojant Teiloro ištiesinimo metodą, apskaičiuotos sukonstruotų įvertinių apytikslės dispersijos ir pasiūlyti šių dispersijų įvertiniai.

- Sukonstruotų įvertinių ir Teiloro ištiesinimo metodu gautų įvertinių empirinės dispersijos palygintos tarpusavyje. Atsisakius netiesinės Teiloro eilutės dalies, prarandama nedaug, nes skirtumai tarp lyginamų empirinių dispersijų yra nedideli. Mažiausias skirtumas pastebimas tarp $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$ įvertinio empirinių dispersijų.

- Taikant matematinį modeliavimą, sukonstruoti įvertiniai palyginti tarpusavyje ir su standartiniu įvertiniu. Remiantis charakteristika **VKP**, gavome, kad nauji kalibruotieji Horvico ir Tompsono įvertinio dispersijos įvertiniai tiksliausi tada, kai taikomas sluoksninis ėmimas, o sluoksnių ribos nustatomos pagal šaknies iš f taisyklę. Iš įvertinių, turinčių (8) pavidalą, tiksliausias yra $\widehat{\mathbf{D}}^{(3)}(\widehat{Y}_{HT})$, o iš įvertinių, kurie yra (13) pavidalo, tiksliausias $\widehat{\mathbf{D}}^{(4)}_{MC}$.

LITERATŪROS SĄRAŠAS

1. Čekanavičius V., Murauskas G. Statistika ir jos taikymai. I dalis. TEV, Vilnius, 2003.
2. Gunning P., Horgan J.M., Yancey W. Geometric stratification of accounting data. Revista Contaduría y Administración No. 214, septiembre - diciembre, 2004.
3. Horvitz D., Thompson D. A generalization of sampling without replacement from a finite universe. Journal of the American Statistical Association, 74, No. 260, 1952, p. 663 - 685.
4. Krapavickaitė D., Plikusas A. Estimation of a ratio in the finite population. Informatica, 16, No. 3, 2005, p. 347 - 364.
5. Krapavickaitė D., Plikusas A. Imčių teorijos pagrindai. Vilnius: Technika, 2005.
6. Lipeikienė J. Matematika su kompiuteriu. Kompiuterinės matematinės sistemos DERIVE, MAPLE, MATLAB. Mokslo aidai, Vilnius, 2002.
7. Plikusas A. Calibrated estimators of the ratio. Lietuvos matematikos rinkinys, 41, spec. numeris, 2001, p. 457 - 462.
8. Plikusas A., Pumputis D. Estimation of the finite population covariance using calibration. Nonlinear Analysis: Modelling and Control, 2010, Vol. 15, No. 3, p. 325 - 340.
9. Pumputis D. Baigtinės populiacijos parametrų statistiniai įvertinimai, gauti naudojant papildomą informaciją. Daktaro disertacija. Technika, Vilnius, 2008.
10. Pumputis D. Stratification of populations with skewed distribution. Lietuvos matematikos rinkinys, 2007, Vol. 47, special issue, p. 369 - 374.
11. Pumputis D., Čiginas A. Estimation of quadratic finite population function using calibration. STATISTICS IN TRANSITION - new series, Vol. 12, No. 2, October 2011, p. 309 - 330.

12. Scheaffer R. L., Mendenhall W., Lyman Ott R., Gerow K. Elementary survey sampling. Brooks/Cole, Cengage Learning, 2006.

13. Singh S., Farrell P. Model - assisted higher - order calibration of estimators of variance. Austral & New Zealand J. Statist., 2005, Vol. 47(3), p. 375 - 383.

14. Singh S., Horn S., Chowdhury S., Yu F. Calibration of the estimators of variance. Austral & New Zealand J. Statist., 1999, Vol. 41(2), p. 199 - 212.

15. Sitter R., Wu C. Efficient estimation of quadratic finite population functions in the presence of auxiliary information. Journal of the American Statistical Association, 97, No. 458, 2002, p. 535 - 543.

SANTRAUKA

Šiame magistro diplominiame darbe, naudojant skirtingas atstumo funkcijas ir kalibravimo lygtis, išvedami Horvico ir Tompsono įvertinio dispersijos įvertinių svoriai. Tokiu būdu, sukonstruojami aštuoni nauji Horvico ir Tompsono įvertinio dispersijos įvertiniai. Naudojant Teiloro ištiesinimo metodą pateikiamos sukonstruotų įvertinių apytikslės dispersijos ir pasiūlyti šių dispersijų įvertiniai. Be to, darbe atliekamas matematinis modeliavimas, kurio eksperimentai atlikti naudojant darbo autorės sukurtas MATLAB programas. Matematinio modeliavimo tikslas - naujus įvertinius palyginti tarpusavyje ir su standartiniu įvertiniu. Tiriama, kaip įvertinių tikslumas priklauso nuo pasirinkto imties plano.

Raktiniai žodžiai: apytikslė dispersija, baigtinė populiacija, Horvico ir Tompsono įvertinys, įvertinys, įvertis, kalibruotasis įvertinys, sluoksninis ėmimas.

SUMMARY

In this master's graduation work, the weights of estimators of Horvitz & Thompson estimator of variance are defined by using some different distance function and calibration equations. In such a way, the new eight estimators of Horvitz & Thompson estimator of variance were constructed. Using the Taylor linearization method the approximate variances of the constructed estimators were derived. The estimators of the variances of these estimators are proposed as well. Also we perform here a mathematical modeling using MATLAB program. The aim of this mathematical modeling is to compare the new estimators with each other and with a standard one. We analyze also how the accuracy of estimators depends of selected sampling design.

Key words: approximate variance, finite population, Horvitz & Thompson estimator, estimator, estimate, calibrated estimator, stratified sampling design.

PRIEDAS

1. Programa pradinių įvertinių tikslumo matams ir Teiloro ištiesintųjų įvertinių empirinei dispersijai apskaičiuoti

```
fid=fopen('realus08-09.txt','r');
A=fscanf(fid,'%f',[5,300]);
fclose(fid);
disp('kintamasis x:');
x=A(1,1:end);
disp('papildomas kintamasis y:');
y=A(4,1:end);
N=300;
n=input('Iveskite imties dydi n = ');
s=input('Iveskite kiek imciu rinksime s = ');
Pik=n/N; Pikl=((n*(n-1))/(N*(N-1)));
qk=1; qkl=1; dk=1/Pik;
dij=(Pik^2-Pik)/Pik;
dkl=(Pik^2-Pikl)/Pikl;
gk=1/Pik; gkl=1/Pikl;
N1=N*(N-1)/2; n1=n*(n-1)/2;
N2=N*N; n2=n*n;
tk=n2/N2;
tkl=(n2*(n2-1))/(N2*(N2-1));
suma=0;
% skaiciuojame suma Vyg(Xht)
for i=1:N
for j=1:N
if i==j
summ=(Pik^2-Pik)*(x(i)/Pik-x(j)/Pik)^2;
else
summ=(Pik^2-Pikl)*(x(i)/Pik-x(j)/Pik)^2;
end
suma=suma+summ;
end
end
Vx=1/2*suma;
% skaiciuojame suma Vyg(Yht)
suma=0;
for i=1:N
for j=1:N
if i==j
summ=(Pik^2-Pik)*(y(i)/Pik-y(j)/Pik)^2;
else
summ=(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
end
suma=suma+summ;
end
end
Vy=1/2*suma;
%1,4 Teiloro istiesintam ivertiniui
suma11=0; suma12=0; suma13=0; suma14=0;
```

```
suma15=0; suma16=0; suma17=0; suma18=0;
for i=1:N
for j=1:N
if i==j
summ11=(Pik^2-Pik)*qkl*(x(i)/Pik-x(j)/Pik)^4;
summ12=(Pik^2-Pik)*qkl*(x(i)/Pik-x(j)/Pik)^2*(y(i)/Pik-y(j)/Pik)^2;
summ13=(Pik^2-Pik)*(y(i)/Pik-y(j)/Pik)^2;
summ14=(Pik^2-Pik)*qkl*(x(i)/Pik-x(j)/Pik)^2;
summ15=(Pik^2-Pik)*qkl; summ16=(Pik^2-Pik);
summ17=(Pik^2-Pik)*(x(i)/Pik-x(j)/Pik)^2;
summ18=(Pik^2-Pik)*qkl*(y(i)/Pik-y(j)/Pik)^2;
else
summ11=(Pik^2-Pikl)*qkl*(x(i)/Pik-x(j)/Pik)^4;
summ12=(Pik^2-Pikl)*qkl*(x(i)/Pik-x(j)/Pik)^2*(y(i)/Pik-y(j)/Pik)^2;
summ13=(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
summ14=(Pik^2-Pikl)*qkl*(x(i)/Pik-x(j)/Pik)^2;
summ15=(Pik^2-Pikl)*qkl; summ16=(Pik^2-Pikl);
summ17=(Pik^2-Pikl)*(x(i)/Pik-x(j)/Pik)^2;
summ18=(Pik^2-Pikl)*qkl*(y(i)/Pik-y(j)/Pik)^2;
end
suma11=suma11+summ11; suma12=suma12+summ12; suma13=suma13+summ13;
suma14=suma14+summ14; suma15=suma15+summ15; suma16=suma16+summ16;
suma17=suma17+summ17; suma18=suma18+summ18;
end
end
h1=suma11;  h2=suma12;  h3=suma13;  h4=suma14;
h5=suma15;  h6=suma16;  h7=suma17;  h8=suma18;
% ai - atitinkamos isvestines, pagal ti ivertinius, taske(t1,t2,...,t8)
laisn1=1/(2*(h1)^2)*(2*h1*h2*Vx); a11=1/(2*(h1)^2)*(h2*h7-2*h2*Vx);
a12=1/(2*(h1)^2)*(2*h1*Vx-h1*h7); a13=1/(2*(h1)^2)*((h1)^2);
a14=1/(2*(h1)^2)*(-h1*h2);
laisn4=1/(2*(h5*h1-(h4)^2)^2)*(-2*h2*(h4)^2*h5*Vx+2*h2*(h5)^2*h1*Vx+
2*h8*(h4)^3*Vx-2*h8*h5*h1*h4*Vx);
a41=1/(2*(h5*h1-(h4)^2)^2)*((h5)^2*h2*h7-2*(h5)^2*h2*Vx-h5*h2*h4*h6+
h8*h6*(h4)^2+2*h8*h5*h4*Vx-h8*h5*h7*h4);
a42=1/(2*(h5*h1-(h4)^2)^2)*(2*(h5)^2*h1*Vx-2*h5*(h4)^2*Vx+h5*h7*(h4)^2-
(h5)^2*h7*h1+h4*h6*h5*h1-(h4)^3*h6);
a43=1/(2*(h5*h1-(h4)^2)^2)*((h1)^2*(h5)^2-2*h1*(h4)^2*h5+(h4)^4);
a44=1/(2*(h5*h1-(h4)^2)^2)*(h2*(h4)^2*h6+h2*h5*h1*h6+4*h2*h4*h5*Vx-
2*h2*h4*h5*h7-2*h8*(h4)^2*Vx+h8*(h4)^2*h7-2*h8*h1*h6*h4-2*h8*h5*h1*Vx+h8*h5*h1*h7);
a45=1/(2*(h5*h1-(h4)^2)^2)*(h7*(h4)^2*h2-2*Vx*(h4)^2*h2-h6*h1*h2*h4+
(h1)^2*h6*h8+2*h4*h8*h1*Vx-h7*h4*h8*h1);
a46=1/(2*(h5*h1-(h4)^2)^2)*(h2*h4*h5*h1-h8*(h1)^2*h5+h8*h1*(h4)^2-h2*(h4)^3);
a47=1/(2*(h5*h1-(h4)^2)^2)*(h5*h2*(h4)^2-(h5)^2*h2*h1-(h4)^3*h8+ h4*h8*h5*h1);
a48=1/(2*(h5*h1-(h4)^2)^2)*(2*(h4)^3*Vx-h7*(h4)^3-2*h4*h5*h1*Vx+
h1*h6*(h4)^2-(h1)^2*h6*h5+h7*h4*h5*h1);
%2,3 Teiloro istiesintam ivertiniui
suma1=0;  suma2=0;  suma3=0;  suma4=0;
suma5=0;  suma6=0;  suma7=0;  suma8=0;
for i=1:N
for j=1:N
if i==j
```

```
summ1=(Pik^2-Pik)*dij*qkl; summ2=(Pik^2-Pik)*(x(i)/Pik-x(j)/Pik)^2;
summ3=(Pik^2-Pik)*dij*qkl*(x(i)/Pik-x(j)/Pik)^4;
summ4=(Pik^2-Pik)*dij*qkl*(x(i)/Pik-x(j)/Pik)^2; summ5=(Pik^2-Pik);
summ6=(Pik^2-Pik)*dij*qkl*(x(i)/Pik-x(j)/Pik)^2*(y(i)/Pik-y(j)/Pik)^2;
summ7=(Pik^2-Pik)*(y(i)/Pik-y(j)/Pik)^2;
summ8=(Pik^2-Pik)*dij*qkl*(y(i)/Pik-y(j)/Pik)^2;
else
summ1=(Pik^2-Pikl)*dkl*qkl; summ2=(Pik^2-Pikl)*(x(i)/Pik-x(j)/Pik)^2;
summ3=(Pik^2-Pikl)*dkl*qkl*(x(i)/Pik-x(j)/Pik)^4;
summ4=(Pik^2-Pikl)*dkl*qkl*(x(i)/Pik-x(j)/Pik)^2; summ5=(Pik^2-Pikl);
summ6=(Pik^2-Pikl)*dkl*qkl*(x(i)/Pik-x(j)/Pik)^2*(y(i)/Pik-y(j)/Pik)^2;
summ7=(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
summ8=(Pik^2-Pikl)*dkl*qkl*(y(i)/Pik-y(j)/Pik)^2;
end
suma1=suma1+summ1; suma2=suma2+summ2; suma3=suma3+summ3; suma4=suma4+summ4;
suma5=suma5+summ5; suma6=suma6+summ6; suma7=suma7+summ7; suma8=suma8+summ8;
end
end
t1=suma1;  t2=suma2;  t3=suma3;  t4=suma4;
t5=suma5;  t6=suma6;  t7=suma7;  t8=suma8;
laisn=-1/(2*(t1*t3-t4^2)^2)*(t6*t1*t4^2*Vx-t6*t1^2*t3*Vx-t8*t4^3*Vx+t8*t1*t3*Vx);
a1=-1/(2*(t1*t3-t4^2)^2)*(Vx*t4^2*t6-t2*t4^2*t6-t5*t3^2*t8+t2*t4*t3*t8-
Vx*t4*t3*t8+t5*t3*t4*t6);
a2=-1/(2*(t1*t3-t4^2)^2)*(t1^2*t6*t3-t1*t6*t4^2+t4^3*t8-t4*t8*t1*t3);
a3=-1/(2*(t1*t3-t4^2)^2)*(t1^2*Vx*t6-t1^2*t2*t6-t5*t4^2*t8+t1*t2*t4*t8-
t1*Vx*t4*t8+t5*t4*t1*t6);
a4=-1/(2*(t1*t3-t4^2)^2)*(2*t6*t1*t2*t4-2*t6*t1*Vx*t4+2*t8*t5*t3*t4-
t8*t1*t3*t2+t8*t1*t3*Vx-t6*t1*t5*t3+t8*Vx*t4^2-t6*t5*t4^2-t8*t4^2*t2);
a5=-1/(2*(t1*t3-t4^2)^2)*(t3^2*t8*t1-t3*t8*t4^2+t4^3*t6-t4*t6*t1*t3);
a6=-1/(2*(t1*t3-t4^2)^2)*(t1^2*t2*t3+t1*Vx*t4^2-t1^2*Vx*t3-t1*t2*t4^2
-t5*t4*t1*t3+t5*t4^3);
a7=-1/(2*(t1*t3-t4^2)^2)*(2*t4^2*t1*t3-t4^4-t1^2*t3^2);
a8=-1/(2*(t1*t3-t4^2)^2)*(t5*t3^2*t1-t5*t3*t4^2+t2*t4^3-Vx*t4^3-
t2*t4*t1*t3+Vx*t4*t1*t3);
laisn3=(-1/(2*(t3)^2))*(-2*t6*t3*Vx); a31=(-1/(2*(t3)^2))*(t6*t3);
a32=(-1/(2*(t3)^2))*(2*Vx*t6-t2*t6); a33=(-1/(2*(t3)^2))*(t2*t3-2*Vx*t3);
a34=(-1/(2*(t3)^2))*(-(t3)^2);
% 5 - 8 Teiloro istiesintiems ivertiniams
suma1=0;  suma2=0;
for i=1:N
A=[1]; vx=[A;x(i)];
summ1=gk*vx*vx'; suma1=suma1+summ1;
summ2=gk*vx*y(i); suma2=suma2+summ2;
end
betta1=suma1^(-1)*suma2;
suma=0;
for i=1:N-1
for j=i+1:N
vxi=[1;x(i)]; vxj=[1;x(j)];
summ=(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2;
suma=summ+suma;
end
```

```
end
SN=suma;
suma51=0;  suma52=0;  suma53=0;  suma54=0; suma55=0;  suma56=0;  suma57=0;
suma58=0; suma71=0;  suma72=0;  suma73=0;  suma74=0;  suma75=0;
for i=1:N-1
for j=i+1:N
vxi=[1;x(i)]; vxj=[1;x(j)];
summ51=(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
summ52=qkl*(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
summ53=qkl*(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2*
(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2; summ54=qkl;
summ55=qkl*(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2;
summ56=1;
summ57=qkl*((Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2)^2;
summ58=(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2;
suma71=qkl*qkl*(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2*
(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
suma72=qkl*qkl*((Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2)^2;
suma73=qkl*qkl*(Pik^2-Pikl)*(y(i)/Pik-y(j)/Pik)^2;
suma74=qkl*qkl;
suma75=qkl*qkl*(Pik^2-Pikl)*((vxi'*betta1)/Pik-(vxj'*betta1)/Pik)^2;
suma51=suma51+summ51; suma52=suma52+summ52; suma53=suma53+summ53;
suma54=suma54+summ54; suma55=suma55+summ55; suma56=suma56+summ56;
suma57=suma57+summ57; suma58=suma58+summ58; suma71=suma71+suma71;
suma72=suma72+suma72; suma73=suma73+suma73; suma74=suma74+suma74;
suma75=suma75+suma75;
end
end
g1=suma51;  g2=suma52;  g3=suma53;  g4=suma54;
g5=suma55;  g6=suma56;  g7=suma57;  g8=suma58;
g9=suma71;  g10=suma72;  g11=suma73;  g12=suma74;
g13=suma75;
laisn5=-1/((g4)^2*(g7*g4-g5^2)^2)*(2*g2*g4^3*g5^4*g6-g2*g4^3*g7^2*N1-
g2*g4*g5^2*N1-g2*g4^3*g5^4*N1-g2*g4^2*g5^3*SN-2*g4^4*g3*g5^3*g6+g4^4*
g3*g5^3*N1-g4^4*g3*g7*SN+g4^3*g3*g5^2*SN+2*g7*g2*g4^2*g5^2*N1-2*g2*
g4^4*g5^2*g6*g7+g2*g4^4*g5^2*g7*N1+g2*g4^3*g5*g7*SN+2*g4^5*g3*g5*
g6*g7-g4^5*g3*g5*g7*N1);
a51=-1/((g4)^2*(g7*g4-g5^2)^2)*(2*g4^3*g7*g5^2-g7^2*g4^4-g4^2*g5^2);
a52=-1/((g4)^2*(g7*g4-g5^2)^2)*(g4^3*g5^4*N1-g4^3*g5^4*g6-
g7^2*g4^3*N1-g4^2*g5^3*SN+ g4^2*g5^3*g8+g6*g7^2*g4^3-g4*g5^4*N1+
g4*g6*g5^4+2*g7*g4^2*g5^2*N1+g4^4*g5^2*g6*g7- g4^4*g5^2*g7*N1+
g4^3*g5*g7*SN-g4^3*g5*g8*g7-2*g6*g7*g4^2*g5^2);
a53=-1/((g4)^2*(g7*g4-g5^2)^2)*(g4^4*g6*g5^3-g4^4*g5^3*N1-
g4^4*g7*SN+g4^3*g5^2*SN+g4^4*g8*g7-g4^3*g8*g5^2-g4^5*g6*g5*g7+g4^5*g5*g7*N1);
a54=-1/((g4)^2*(g7*g4-g5^2)^2)*(g2*g5^4*N1-g2*g6*g5^4-
g5*g3*g6*g7*g4^4+2*g3*g6*g5^3*g4^3+ g5*g3*g7*g4^4*N1-
2*g5^3*g4^3*g3*N1+g5^2*g4^2*g3*SN-g5^2*g4^2*g3*g8-g2*g6*g4^2*g5^4+
g2*g4^2*g5^4*N1+g2*g7^2*g4^2*N1-2*g2*g7*g4*g5^2*N1-g2*g4^2*g5*g7*SN+
g2*g5*g8*g7*g4^2-g2*g6*g7^2*g4^2+2*g2*g6*g7*g4*g5^2);
a55=-1/((g4)^2*(g7*g4-g5^2)^2)*(g4^5*g3*g7*N1-g4^5*g3*g6*g7-
g4^4*g3*g6*g5^2+g4^4*g3*g5^2*N1-2*g4^3*g3*g5*SN+2*g4^3*g3*g5*g8+
2*g4^4*g2*g5*g6*g7-2*g4^4*g2*g5*g7*N1+g4^3*g2*g7*SN-g4^3*g2*g7*g8+
```

```
g4^2*g2*g5^2*SN-g4^2*g2*g8*g5^2);
a56=-1/((g4)^2*(g7*g4-g5^2)^2)*(g5^3*g3*g4^4+g2*g7^2*g4^3+
g4*g2*g5^4-g2*g4^3*g5^4-g5*g3*g4^5*g7-2*g2*g7*g4^2*g5^2+g2*g4^4*g5^2*g7);
a57=-1/((g4)^2*(g7*g4-g5^2)^2)*(g4^4*g3*SN-g4^4*g8*g3+
g4^5*g6*g5*g3-g4^4*g6*g5^2*g2-g4^5*g5*g3*N1+g4^4*g5^2*g2*N1-
g4^3*g5*g2*SN+g4^3*g8*g5*g2);
a58=-1/((g4)^2*(g7*g4-g5^2)^2)*(g4^4*g3*g7-g4^3*g3*g5^2+
g4^2*g5^3*g2-g4^3*g5*g2*g7);
laisn6=-1/(g7^2)*g3*g7*SN; a61=-1/(g7^2)*(-g7^2);
a62=-1/(g7^2)*(g7*g8-g7*SN); a63=-1/(g7^2)*(g7*SN-g3*g8);
a64=-1/(g7^2)*g3*g7;
laisn7=1/(g10^2)*g9*g10*SN;
a71=1/(g10^2)*g10^2; a72=1/(g10^2)*(-g9*g10);
a73=1/(g10^2)*(g10*SN-g10*g8); a74=1/(g10^2)*(g9*g8-g9*SN);
laisn8=-1/(g12^2*g10^2*(g10-g12)^2)*(2*g11*g10^2*g12^3*N1-
g11*g10^2*g12^3*N1-g11*g10^4*g12*N1-g12^4*g9*g10*SN+
g12^3*g9*g10^2*SN+g11*g10*g12^3*g13*SN-g11*g10^2*g12^2*g13*SN-
g11*g10*g12^2*g13^2*N1+g11*g10^2*g12*g13^2*N1+g12^3*g9*g10*g13*N1-
g12^2*g9*g10^2*g13*N1);
a81=-1/(g12^2*g10^2*(g10-g12)^2)*(2*g12^3*g10^3-g12^4*g10^2-g10^4*g12^2);
a82=-1/(g12^2*g10^2*(g10-g12)^2)*(g10^2*g12^3*g11-2*g10^3*g12^2*g11+
g10^4*g12*g11+g10*g12^2*g11*g13^2-g10^2*g12*g11*g13^2-g10*g12^3*g9*g13);
a83=-1/(g12^2*g10^2*(g10-g12)^2)*(g12^4*g10*g9*g13-g12^3*g10*g11*g13+
g12^2*g10^2*g11*g13-g12^3*g10^2*g9);
a84=-1/(g12^2*g10^2*(g10-g12)^2)*(g12^3*g10^2*SN+g12^4*g10*g8-
g12^4*g10*SN-g12^3*g10^2*g8+g12^3*g10*g13*N1-g12^2*g10^2*g13*N1-
g12^3*g10*g13*g6+g12^2*g10^2*g13*g6);
a85=-1/(g12^2*g10^2*(g10-g12)^2)*(g12^4*g9*SN-g8*g12^4*g9-
g11*g13*g12^3*SN+g11*g13*g8*g12^3+g11*g13^2*g12^2*N1-g9*g13*g12^3*N1-
g11*g13^2*g12^2*g6+g12^3*g13*g6*g9+2*g11*g13*g10*g12^2*SN-
2*g12^3*g9*g10*SN-2*g11*g13*g10*g8*g12^2+2*g8*g9*g10*g12^3-
2*g10*g11*g12*g13^2*N1+
2*g12^2*g9*g10*g13*N1+2*g12*g11*g10*g6*g13^2-2*g12^2*g9*g10*g13*g6);
a86=-1/(g12^2*g10^2*(g10-g12)^2)*(2*g10^3*g12^2*N1-g10^2*g12^3*N1+
g10^2*g12^3*g6-2*g10^3*g12^2*g6-g10^4*g12*N1+g10^4*g12*g6+
g10*g12^3*g13*SN-g10*g12^3*g13*g8-g10*g12^2*g13^2*N1+g10^2*g12*g13^2*N1+
g10*g6*g12^2*g13^2-g10^2*g12*g13^2*g6-g10^2*g12^2*g13*SN+
g10^2*g12^2*g13*g8);
a87=-1/(g12^2*g10^2*(g10-g12)^2)*(g11*g10^4*N1-g11*g6*g10^4+
g12^2*g9*g10^2*SN-g12^2*g9*g10^2*g8-g10*g9*g13*g12^2*N1+
g10*g9*g13*g12^2*g6+2*g10*g11*g13^2*g12*N1-2*g10*g11*g13^2*g12*g6+
g11*g10^2*g12^2*N1-2*g11*g10^3*g12*N1-g10*g11*g13*g12^2*SN+
g10*g11*g13*g8*g12^2-g11*g13^2*g10^2*N1+g11*g13^2*g6*g10^2-
g11*g6*g10^2*g12^2+2*g11*g6*g10^3*g12);
a88=-1/(g12^2*g10^2*(g10-g12)^2)*(g10*g12^3*g9*N1-g10^2*g12^2*g9*N1-
g10*g12^3*g9*g6+g10^2*g12^2*g9*g6-2*g10*g12^2*g11*g13*N1+
2*g10^2*g12*g11*g13*N1+2*g10*g12^2*g11*g13*g6+g10*g12^3*g11*SN-
2*g10^2*g12*g11*g13*g6-g10^2*g12^2*g11*SN-g10*g12^3*g11*g8+g10^2*g12^2*g11*g8);
for l=1:s
indx = sort(randi(N,[n,1]));
imtisx=x(indx); imtisy=y(indx);
% 1-asis ivertinys
```

```
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
V1=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
end
suma=suma+summ;
end
end
V2=suma;
lamda=(Vx-1/2*V1)/(1/2*V2);
suma=0;
for i=1:n
for j=1:n
if i==j
summ=(dij+lamda*dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=(dkl+lamda*dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
W1(l)=1/2*suma;
% Teiloro istiesintas pradinis 1 ivertinys
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
end
suma=suma+summ;
end
end
VV11=suma;
suma=0;
for i=1:n
```

```
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV12=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV13=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV17=suma;
WW1(l)=laisn1+a11*VV11+a12*VV12+a13*VV13+a14*VV17;
%2-asis ivertinys
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
V3=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
```

```
summm=dij^2*qkl;
else
summm=dkl^2*qkl;
end
suma=suma+summ;
end
end
V4=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
else
summ=dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
end
suma=suma+summ;
end
end
V5=suma;
V6=V4*V5/V3;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij;
else
summ=dkl;
end
suma=suma+summ;
end
end
V7=suma;
lamda2=1/(V3-V6)*(2*Vx+V7/V3*V5-V1);
lamda1=1/V3*((-1)*lamda2*V4-V7);
suma=0;
for i=1:n
for j=1:n
if i==j
summ=(lamda1*dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2+lamda2*dij^2*qkl+dij)*
(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=(lamda1*dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2+
lamda2*dkl^2*qkl+dkl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
W2(l)=1/2*suma;
% Teiloro istiesintas pradinis 2 ivertinys
suma=0;
for i=1:n
```

```
for j=1:n
if i==j
summ=dij^2*qkl;
else
summ=dkl^2*qkl;
end
suma=suma+summ;
end
end
VV1=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV2=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
else
summ=dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
end
suma=suma+summ;
end
end
VV3=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV4=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij;
else
```

```
summm=dkl;
end
suma=suma+summ;
end
end
VV5=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl^2*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV6=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV7=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij^2*qkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl^2*qkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV8=suma;
WW2(l)=laisn+a1*VV1+a2*VV2+a3*VV3+a4*VV4+a5*VV5+a6*VV6+a7*VV7+a8*VV8;
% 3-asis ivertinys
lamda=(Vx-1/2*V1)/(1/2*V5);
suma=0;
for i=1:n
for j=1:n
if i==j
summ=(lamda*(1/2*dij^2*qkl)*(imtisx(i)/Pik-imtisx(j)/Pik)^2+dij)*
(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
```

```
summm=(lamda*(1/2*dkl^2*qkl)*(imtisx(i)/Pik-imtisx(j)/Pik)^2+dkl)*
(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
W3(l)=1/2*suma;
% Teiloro istiesintas pradinis 3 ivertinys
WW3(l)=laisn3+a31*VV2+a32*VV3+a33*VV6+a34*VV7;
% 4-asis ivertinys
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
V8=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl;
else
summ=dkl*qkl;
end
suma=suma+summ;
end
end
V9=suma;
lamda2=1/(V8-((V9*V2)/V8))*(2*Vx+1/V8*V7*V2-V1);
lamda1=1/V8*((-1)*lamda2*V9-V7);
suma=0;
for i=1:n
for j=1:n
if i==j
summ=(lamda1*dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2+lamda2*dij*qkl+dij)*
(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=(lamda1*dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2+lamda2*dkl*qkl+dkl)*
(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
W4(l)=1/2*suma;
% Teiloro istiesintas pradinis 4 ivertinys
suma=0;
```

```
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^4;
end
suma=suma+summ;
end
end
VV41=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV42=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV43=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*qkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV44=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl;
```

```
else
summm=dkl*qkl;
end
suma=suma+summ;
end
nd
VV45=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij;
else
summm=dkl;
end
suma=suma+summ;
end
end
VV46=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
else
summ=dkl*(imtisx(i)/Pik-imtisx(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV47=suma;
suma=0;
for i=1:n
for j=1:n
if i==j
summ=dij*qkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
else
summ=dkl*qkl*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
end
suma=suma+summ;
end
end
VV48=suma;
WW4(l)=laisn4+a41*VV41+a42*VV42+a43*VV43+a44*VV44+a45*VV45+a46*VV46+a47*VV47+a48*VV48;
% 5-asis ivertinys
% skaiciuojame regresijos keof betta
suma1=0;
suma2=0;
for i=1:n
A=[1]; vx=[A;imtisx(i)];
summ1=gk*vx*vx'; suma1=suma1+summ1;
summ2=gk*vx*imtisy(i); suma2=suma2+summ2;
```

```
end
betta=suma1^(-1)*suma2;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)^2*gkl*qkl;
uma=suma+summ;
end
end
V10=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl*qkl;
suma=suma+summ;
end
end
V11=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)*gkl*qkl;
suma=suma+summ;
end
end
V12=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl;
suma=suma+summ;
end
end
V13=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)*gkl;
suma=suma+summ;
end
end
V14=suma;
lamda2=1/(V10-1/V11*V12^2)*(1/V11*(V13*V12-N1*V12)-V14+SN);
lamda1=1/V11*(N1-lamda2*V12-V13);
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda1*gkl*qkl+lamda2*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)*gkl*qkl+gkl)*
```

```
(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
W5(l)=suma;
% Teiloro istiesintas pradinis 5 ivertinys
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl*(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
VV51=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl*qkl*(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
VV52=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl*qkl*(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2*
(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
VV53=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl*qkl;
suma=suma+summ;
end
end
VV54=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl*qkl*(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2;
suma=suma+summ;
end
end
VV55=suma;
suma=0;
for i=1:n-1
for j=i+1:n
```

```
summm=gkl;
suma=suma+summ;
end
end
VV56=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl*qkl*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)^2;
suma=suma+summ;
end
end
VV57=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl*(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2;
suma=suma+summ;
end
end
VV58=suma;
WW5(l)=laisn5+a51*VV51+a52*VV52+a53*VV53+a54*VV54+a55*VV55+a56*VV56+
a57*VV57+a58*VV58;
% 6-asis ivertinys
lamda=1/V10*(SN-V14);
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)
*gkl*qkl+gkl)*(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
W6(l)=suma;
% Teiloro istiesintas pradinis 6 ivertinys
WW6(l)=laisn6+a61*VV51+a62*VV53+a63*VV57+a64*VV58;
% 7-asis ivertinys
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)^2*gkl^2*qkl;
suma=suma+summ;
end
end
V15=suma;
lamda=1/V15*(SN-V14);
suma=0;
for i=1:n-1
```

```
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda*gkl^2*qkl*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)+gkl)*
(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
W7(l)=suma;
% Teiloro istiesintas pradinis 7 ivertinys
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl^2*qkl*(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2*
(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
VV79=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl^2*qkl*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)^2;
suma=suma+summ;
end
end
VV710=suma;
WW7(l)=laisn7+a71*VV51+a72*VV58+a73*VV79+a74*VV710;
% 8-asis ivertinys
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2*gkl^2*qkl;
suma=suma+summ;
end
end
V16=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl^2*qkl;
suma=suma+summ;
end
end
V17=suma;
lamda2=1/(V15-V16^2/V17)*(SN-V14-V16/V17*(N1-V13));
lamda1=1/V17*(N1-lamda2*V16-V13);
suma=0;
for i=1:n-1
for j=i+1:n
```

```
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda1*gkl^2*qkl+lamda2*gkl^2*qkl*((Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2)
+gkl)*(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
W8(l)=suma;
% Teiloro istiesintas pradinis 8 ivertinys
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl^2*qkl*(Pik^2-Pikl)*(imtisy(i)/Pik-imtisy(j)/Pik)^2;
suma=suma+summ;
end
end
VV81=suma;
suma=0;
for i=1:n-1
for j=i+1:n
summ=gkl^2*qkl;
suma=suma+summ;
end
end
VV82=suma;
suma=0;
for i=1:n-1
for j=i+1:n
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=gkl^2*qkl*(Pik^2-Pikl)*((vxim'*betta)/Pik-(vxjm'*betta)/Pik)^2;
suma=suma+summ;
end
end
VV83=suma;
WW8(l)=laisn8+a81*VV51+a82*VV56+a83*VV58+a84*VV79+a85*VV710+a86*VV81+a87*VV82+a88*VV83;
end
disp('1-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W1(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W1(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid;
disp('Teiloro istiesintojo 1 ivertinio');
```

```
suma=0;
for k=1:s
    summ=WW1(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW1(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('2-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W2(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W2(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid;
disp('Teiloro istiesinto 2 ivertinio');
suma=0;
for k=1:s
    summ=WW2(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW2(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('3-iojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W3(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W3(k)-Vid)^2;
```

```
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid;
disp('Teiloro istiesinto 3 ivertinio');
suma=0;
for k=1:s
    summ=WW3(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW3(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('4-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W4(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W4(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid;
disp('Teiloro istiesinto 4 ivertinio');
suma=0;
for k=1:s
    summ=WW4(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW4(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('5-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W5(k);
    suma=summ+suma;
```

```
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W5(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid
disp('Teiloro istiesinto 5 ivertinio');
suma=0;
for k=1:s
    summ=WW5(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW5(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('6-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W6(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W6(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid
disp('Teiloro istiesinto 6 ivertinio');
suma=0;
for k=1:s
    summ=WW6(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW6(k)-Vid)^2;
    suma=suma+summ;
end
```

```
Dis=1/(s-1)*suma
disp('7-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W7(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W7(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid
disp('Teiloro istiesinto 7 ivertinio');
suma=0;
for k=1:s
    summ=WW7(k);
    suma=summ+suma;
end
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW7(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
disp('8-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W8(k);
    suma=summ+suma;
end
Vid=1/s*suma;
Posl=Vid-Vy;
suma=0;
for k=1:s
    summ=(W8(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2;
cv=sqrt(Dis)/Vid
disp('Teiloro istiesinto 8 ivertinio');
suma=0;
for k=1:s
    summ=WW8(k);
    suma=summ+suma;
end
```

```
Vid=1/s*suma;
suma=0;
for k=1:s
    summ=(WW8(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
```

2. Sluoksniavimo šaknies iš f metodu programa

```
fid=fopen('pop3_N554x.txt','r');
a=fscanf(fid,'%f',[4,554]);
fclose(fid);
disp('kintamasis x:');
x=a(2,1:end);
disp('papildomas kintamasis y:');
y=a(4,1:end);
N=554; qkl=1; N1=N*(N-1)/2;
H=input('iveskite sluoksniu skaiciu H= ');
n=input('iveskite imties dydi n = ');
s=input('iveskite kiek imciu rinksime s = ');
% Sluoksniu ribu nustatymas ir optimalus imties dydzio paskirstymas i sluoksnius pagal Neymana
disp('Saknies is f metodas');
disp(' ');
r=round(1.72*N^(1/3));
h=round((max(y)-min(y))/r);
I(1)=min(y);
for i=1:r
if i==r
I(i+1)=max(y);
else
I(i+1)=I(i)+h;
end
disp(['Intervalas ',num2str(i),' : ', num2str(I(i)),'-',num2str(I(i+1))]);
end
suma=0;
for i=1:r
daz=0;
for j=1:N
if (i==1 && y(j)<=I(i+1) && y(j)>=I(i))
daz=daz+1; d(i)=daz;
elseif (i>1 && y(j)<=I(i+1) && y(j)>I(i))
daz=daz+1; d(i)=daz;
end
end
sak(i)=sqrt(daz);
suma=suma+sqrt(daz);
S(i)=suma;
disp([num2str(i),'-ojo intervalo daznis: ', num2str(d(i)) ,
', saknis is daznio = ', num2str(sak(i)),'. Suma = ',num2str(S(i))]);
end
Q=1/H*S(r);
```

```
sh(1)=1;
disp([' sh(',num2str(1),') : ', num2str(sh(1))]);
for i=1:H
suma=0;
for j=1:r
suma=suma+sak(j);
skirt(j)=abs(i*Q-suma);
end;
[t,v]=min(skirt);
suma=0;
for k=1:v
suma=suma+d(k);
sh(i+1)=suma;
end
if i==1
Nh(i)=length(y(1,sh(i):sh(i+1)));
else
Nh(i)=length(y(1,(sh(i)+1):sh(i+1)));
end
disp([' Nh(',num2str(i),') : ', num2str(Nh(i))]);
disp(' ');
disp([' sh(',num2str(i+1),') : ', num2str(sh(i+1))]);
end
disp(' ' );
Nh=[];
for i=1:H
if i==1
hy=y(sh(i):sh(i+1));
else
hy=y(sh(i)+1:sh(i+1));
end
Nh(i)=length(hy);
stn(i)=std(hy);
end
vard=0;
for i=1:H
vard=vard+(Nh(i)*stn(i));
end;
for i=1:H
skait(i)=Nh(i)*stn(i);
end;
suma=0;
nh=[];
for i=1:H
nh(i)=round(n*skait(i)/vard);
end
nn=sum(nh);
for i=1:H
disp([' nh(',num2str(i),') : ', num2str(nh(i))]);
end;
% skaiciuosime elementu tikimybes pagal visa populiacija
P=zeros(N,N);
```

```
for i=1:N
for j=1:N
for k=1:H
if (k==1 && i<=sh(k+1) && i>=sh(k))
inr=k;
elseif (k>1 && i<=sh(k+1) && i>=sh(k)+1)
inr=k;
end
if (k==1 && j<=sh(k+1)&& j>=sh(k))
jnr=k;
elseif (k>1 && j<=sh(k+1) && j>=sh(k)+1)
jnr=k;
end
end
if inr==jnr
P(i,j)=(nh(inr)*(nh(inr)-1))/(Nh(inr)*(Nh(inr)-1));
else
P(i,j)=nh(inr)/Nh(inr)*nh(jnr)/Nh(jnr);
end
end
tiN(i)=nh(inr)/Nh(inr);
end
piN=tiN;
PN=P;
% skaiciuojame suma Vyg(Xht)
suma=0;
for i=1:N
for j=1:N
if i==j
summ=(piN(i)^2-piN(i))*(x(i)/piN(i)-x(j)/piN(j))^2;
else
summ=(piN(i)*piN(j)-PN(i,j))*(x(i)/piN(i)-x(j)/piN(j))^2;
end
suma=suma+summ;
end
end
Vx=1/2*suma;
% skaiciuojame suma Vyg(Yht)
suma=0;
for i=1:N
for j=1:N
if i==j
summ=(piN(i)^2-piN(i))*(y(i)/piN(i)-y(j)/piN(j))^2;
else
summ=(piN(i)*piN(j)-PN(i,j))*(y(i)/piN(i)-y(j)/piN(j))^2;
end
suma=suma+summ;
end
end
Vy=1/2*suma;
for l=1:s
imtisx=[];
```

```
imtisy=[];
indx=[];
for i=1:H
for j=1:nh(i)
if i==1
ind=randi([sh(i),sh(i+1)]);
else
ind=randi([sh(i)+1,sh(i+1)]);
end
indx=[indx ind];
x2=x(ind);
imtisx=[imtisx x2];
y2=y(ind);
imtisy=[imtisy y2];
end
end
indx; imtisx; imtisy;
% skaiciuosime elementu tikimybes pagal imti
V=zeros(nn,nn);
for i=1:nn
for j=1:nn
for k=1:H
if (k==1 && indx(i)<=sh(k+1) && indx(i)>=sh(k))
inr=k;
elseif (k>1 && indx(i)<=sh(k+1) && indx(i)>=sh(k)+1)
inr=k;
end
if (k==1 && indx(j)<=sh(k+1) && indx(j)>=sh(k))
jnr=k;
elseif (k>1 && indx(j)<=sh(k+1) && indx(j)>=sh(k)+1)
jnr=k;
end
end
inr; jnr;
if inr==jnr
V(i,j)=(nh(inr)*(nh(inr)-1))/(Nh(inr)*(Nh(inr)-1));
else
V(i,j)=nh(inr)/Nh(inr)*nh(jnr)/Nh(jnr);
end
end
tin(i)=nh(inr)/Nh(inr);
end
pin=tin;
Pn=V;
% standartinis ivertinys
suma=0;
for i= 1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
```

```
end
suma=suma+summ;
end
end
W(l)=1/2*suma;
% 1-asis ivertinys
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
end
suma=suma+summ;
end
end
V1=suma;
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^4;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^4;
end
suma=suma+summ;
end
end
V2=suma;
lamda=(Vx-1/2*V1)/(1/2*V2);
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=(((pin(i)^2-pin(i))/pin(i))+lamda*((pin(i)^2-pin(i))/pin(i))*qkl*
(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2)*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
else
summ=(((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))+lamda*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl*
(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2)*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
end
suma=suma+summ;
end
end
W1(l)=1/2*suma;
%2-asis ivertinys
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
else
```

```
summm=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
end
suma=suma+summ;
end
end
V3=suma;
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))^2*qkl;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl;
end
suma=suma+summ;
end
end
V4=suma;
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^4;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^4;
end
suma=suma+summ;
end
end
V5=suma;
V6=V4*V5/V3;
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=(pin(i)^2-pin(i))/pin(i);
else
summ=(pin(i)*pin(j)-Pn(i,j))/Pn(i,j);
end
suma=suma+summ;
end
end
V7=suma;
lamda2=1/(V3-V6)*(2*Vx+V7/V3*V5-V1);
lamda1=1/V3*((-1)*lamda2*V4-V7);
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=(lamda1*((pin(i)^2-pin(i))/pin(i))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2+lamda2*
((pin(i)^2-pin(i))/pin(i))^2*qkl+((pin(i)^2-pin(i))/pin(i)))*
(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
```

```
else
summ=(lamda1*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2+
lamda2*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl+((pin(i)*pin(j)-Pn(i,j))/Pn(i,j)))*
(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
end
suma=suma+summ;
end
end
W2(l)=1/2*suma;
% 3-asis ivertinys
lamda=(Vx-1/2*V1)/(1/2*V5);
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=(lamda*(1/2*((pin(i)^2-pin(i))/pin(i))^2*qkl)*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2+
((pin(i)^2-pin(i))/pin(i)))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
else
summ=(lamda*(1/2*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))^2*qkl)*(imtisx(i)/pin(i)-
imtisx(j)/pin(j))^2+((pin(i)*pin(j)-Pn(i,j))/Pn(i,j)))*
(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
end
suma=suma+summ;
end
end
W3(l)=1/2*suma;
% 4-asis ivertinys
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2;
end
suma=suma+summ;
end
end
V8=suma;
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=((pin(i)^2-pin(i))/pin(i))*qkl;
else
summ=((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl;
end
suma=suma+summ;
end
end
V9=suma;
lamda2=1/(V8-V9*V2/V8)*(2*Vx+1/V8*V7*V2-V1);
```

```
lamda1=1/V8*((-1)*lamda2*V9-V7);
suma=0;
for i=1:nn
for j=1:nn
if i==j
summ=(lamda1*((pin(i)^2-pin(i))/pin(i))*qkl*(imtisx(i)/pin(i)-imtisx(j)/pin(j))^2+
lamda2*((pin(i)^2-pin(i))/pin(i))*qkl+((pin(i)^2-pin(i))/pin(i)))*
(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
else
summ=(lamda1*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl*(imtisx(i)/pin(i)-
imtisx(j)/pin(j))^2+lamda2*((pin(i)*pin(j)-Pn(i,j))/Pn(i,j))*qkl+((pin(i)*pin(j)-
Pn(i,j))/Pn(i,j)))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
end
suma=suma+summ;
end
end
W4(l)=1/2*suma;
% 5-asis ivertinys
% skaiciuojame regresijos keof betta
suma1=0;
suma2=0;
for i=1:nn
A=[1];
vx=[A;imtisx(i)];
summ1=(1/pin(i))*vx*vx';
suma1=suma1+summ1;
summ2=(1/pin(i))*vx*imtisy(i);
suma2=suma2+summ2;
end
betta=suma1^(-1)*suma2;
suma=0;
for i=1:(N-1)
for j=(i+1):N
vxi=[1;x(i)];
vxj=[1;x(j)];
summ=(piN(i)*piN(j)-PN(i,j))*((vxi'*betta)/piN(i)-(vxj'*betta)/piN(j))^2;
suma=summ+suma;
end
end
SN=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)^2*(1/Pn(i,j))*qkl;
suma=suma+summ;
end
end
V10=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
```

```
summm=(1/Pn(i,j))*qkl;
suma=suma+summ;
end
end
V11=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)*(1/Pn(i,j))*qkl;
suma=suma+summ;
end
end
V12=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
summ=1/Pn(i,j);
suma=suma+summ;
end
end
V13=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)*(1/Pn(i,j));
suma=suma+summ;
end
end
V14=suma;
lamda2=1/(V10-1/V11*V12^2)*(1/V11*(V13*V12-N1*V12)-V14+SN);
lamda1=1/V11*(N1-lamda2*V12-V13);
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda1*(1/Pn(i,j))*qkl+lamda2*((pin(i)*pin(j)-Pn(i,j))*
((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)*(1/Pn(i,j))*qkl+(1/Pn(i,j)))*
pin(i)*pin(j)-Pn(i,j))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
suma=suma+summ;
end
end
W5(l)=suma;
% 6-asis ivertinys
lamda=1/V10*(SN-V14);
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda*((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)*
(1/Pn(i,j))*qkl+(1/Pn(i,j)))*(pin(i)*pin(j)-Pn(i,j))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
```

```
suma=suma+summ;
end
end
W6(l)=suma;
% 7-asis ivertinys
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)^2*(1/Pn(i,j))^2*qkl;
suma=suma+summ;
end
end
V15=suma;
lamda=1/V15*(SN-V14);
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda*(1/Pn(i,j))^2*qkl*((pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-
(vxjm'*betta)/pin(j))^2)+(1/Pn(i,j)))*(pin(i)*pin(j)-Pn(i,j))*
(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
suma=suma+summ;
end
end
W7(l)=suma;
% 8-asis ivertinys
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(pin(i)*pin(j)-Pn(i,j))*((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2*(1/Pn(i,j))^2*qkl;
suma=suma+summ;
end
end
V16=suma;
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
summ=(1/Pn(i,j))^2*qkl;
suma=suma+summ;
end
end
V17=suma;
lamda2=1/(V15-V16^2/V17)*(SN-V14-V16/V17*(N1-V13));
lamda1=1/V17*(N1-lamda2*V16-V13);
suma=0;
for i=1:(nn-1)
for j=(i+1):nn
vxim=[1;imtisx(i)]; vxjm=[1;imtisx(j)];
summ=(lamda1*(1/Pn(i,j))^2*qkl+lamda2*(1/Pn(i,j))^2*qkl*((pin(i)*pin(j)-Pn(i,j))*
((vxim'*betta)/pin(i)-(vxjm'*betta)/pin(j))^2)+(1/Pn(i,j)))*(pin(i)*pin(j)-
```

```
Pn(i,j))*(imtisy(i)/pin(i)-imtisy(j)/pin(j))^2;
suma=suma+summ;
end
end
W8(l)=suma;
end
disp('standartinio ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W(k)-Vid)^2;
    suma=suma+summ;
end
Disp=(1/(s-1))*suma
VKP=Disp+Posl^2
cv=sqrt(Disp)/Vid
disp('1-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W1(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W1(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('2-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W2(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W2(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
```

```
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('3-iojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W3(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W3(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('4-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W4(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W4(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('5-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W5(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W5(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('6-ojo ivertinio tikslumo matai');
suma=0;
```

```
for k=1:s
    summ=W6(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W6(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('7-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W7(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W7(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
disp('8-ojo ivertinio tikslumo matai');
suma=0;
for k=1:s
    summ=W8(k);
    suma=summ+suma;
end
Vid=1/s*suma
Posl=Vid-Vy
suma=0;
for k=1:s
    summ=(W8(k)-Vid)^2;
    suma=suma+summ;
end
Dis=1/(s-1)*suma
VKP=Dis+Posl^2
cv=sqrt(Dis)/Vid
```

3. Sluoksniavimo geometriniu metodu programa

Ši programa analogiška programai, sluoksniuojančiai populiaciją pagal šaknies iš f metodą. Skiriasi tik programos dalis, nustatanti sluoksnių ribas ir optimalų imties dydį sluoksniuose. Atitinkama dalis pakeičiama taip:

```
<...>
% Sluoksniu ribu nustatymas ir optimalus imties dydzio paskirstymas i sluoksnius pagal Neymana
disp('Geometrinis metodas');
disp(' ');
kr(1)=min(x);
kr(H+1)=max(x);
r=(kr(H+1)/kr(1))^(1/H);
for i=1:H-1
    kr(i+1)=kr(1)*r^i;
end
for i=1:H+1
for j=1:N
skirt(j)=abs(x(j)-kr(i));
end;
[t,v]=min(skirt);
sh(i)=v;
disp([' sh(',num2str(i),') : ', num2str(sh(i))]);
end
disp(' ');
Nh=[];
for i=1:H
    if i==1
      hx=x(sh(i):sh(i+1));
    else
      hx=x(sh(i)+1:sh(i+1));
    end
    Nh(i)=length(hx);
    stn(i)=std(hx);
end
vard=0;
for i=1:H
vard=vard+(Nh(i)*stn(i));
end;
for i=1:H
skait(i)=Nh(i)*stn(i);
end;
suma=0;
nh=[];
for i=1:H
nh(i)=round(n*skait(i)/vard);
end
nn=sum(nh);
for i=1:H
disp([' nh(',num2str(i),') : ', num2str(nh(i))]);
end;
<...>
```

4. Sluoksniavimo laipsnių metodu programa

Ši programa analogiška antrai programai. Skiriasi tik programos dalis, nustatanti sluoksnių ribas ir optimalų imties dydį sluoksniuose. Atitinkama dalis pakeičiama taip:

```
<...>
```

```
% Sluoksniu ribu nustatymas ir optimalus imties dydzio paskirstymas i sluoksnius pagal Neymana
disp('Laipsniu metodas');
disp(' ');
suma=0;
for i=1:N
suma=suma+(x(i))^(0.6);
end;
const=suma/H;
sh(1)=1;
disp([' sh(',num2str(1),') : ', num2str(sh(1))]);
for i=1:H
suma=0;
for j=1:N
suma=suma+(x(j))^(0.6);
skirt(j)=abs(i*const-suma);
end;
[t,v]=min(skirt);
sh(i+1)=v;
disp([' sh(',num2str(i+1),') : ', num2str(sh(i+1))]);
end
disp(' ');
Nh=[];
for i=1:H
    if i==1
      hx=x(sh(i):sh(i+1));
    else
      hx=x(sh(i)+1:sh(i+1));
    end
    Nh(i)=length(hx);
    stn(i)=std(hx);
end
vard=0;
for i=1:H
vard=vard+(Nh(i)*stn(i));
end;
for i=1:H
skait(i)=Nh(i)*stn(i);
end;
suma=0;
nh=[];
for i=1:H
nh(i)=round(n*skait(i)/vard);
end
nn=sum(nh);
for i=1:H
disp([' nh(',num2str(i),') : ', num2str(nh(i))]);
end;
<...>
```