TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
MATEMAATILISE STATISTIKA INSTITUUT

Kristjan Kokorev

Osakogumite kooskõlaline hindamine PPS hüpergeomeetrilise ja Poissoni valiku korral

Bakalaureusetöö

Juhendaja:
Kaur Lumiste

TARTU
2013

Sisukord

Sissejuhatus

Osakogumite hindamine on enamike uuringute korral tähtis eesmärk, eriti riiklike statistika agentuuride poolt läbiviidud suuremahuliste uuringute korral. Võimaluse korral kasutavad agentuurid hindamisel disainipõhist lähenemist. Selleks peab valimimaht olema piisavalt suur, et ka väiksemates osakogumites saadavad hinnangud oleksid rahuldava täpsusega (Estevao ja Särndal 2004). Antud töös käsitleme hindamismeetodeid, mille eelduseks on piisavalt suur valimimaht igas osakogumis. Simulatsioonülesande käigus uurime aga ka antud hinnangute käitumist väikestes osakogumites.

Kaasaegne infoajastu võimaldab meil kerge vaevaga kätte saada andmeid varajasematest uuringutest ja registritest. Sealt saadud lisainformatsiooni saab ära kasutada hinnangute täpsuse tõstmiseks. Valikuuringute korral kasutatakse laialdaselt Deville'i ja Särndali (1992) poolt välja töötatud kalibreerimismeetodit. Kalibreerimine eeldab, et me teame ühe või mitme abitunnuse kogusummat üldkogumis. Meetodi tööpõhimõte seisneb disainikaalude kalibreerimisel nii, et antud valimi korral kaalutud abitunnused summeeruksid teadaolevateks suurusteks.

Tänapäeval on sage olukord, kus viiakse samaaegselt või peaaegu samaaegselt läbi mitu uuringut, mis käsitlevad sama üldkogumit. Üsna tihti on nendes uuringutes mõned ühised tunnused. On loomulik nõuda, et hinnangud kahe erineva uuringu ühiste tunnuste parameetritele oleksid omavahel kooskõlas (Särndal ja Traat 2009). Hinnangute kooskõlalisus on eelkõige tähtis statistika tarbijatele, seega peab statistika tootja oma usaldusväärsuse säilitamiseks välja andma kooskõlalisi tulemusi.

Kooskõlaliste hinnangute saamiseks on Hollandi Statistikaametis välja töötatud korduvkaalumise meetod. Korduvkaalumine kasutab kalibreerimise tööpõhimõtet ja on oma olemuselt kaheetapiline kalibreerimine. Eeldame, et on tegu eelmises lõigus kirjeldatud olukorraga, kusjuures on teada lisainformatsiooni üldkogumi tasemel. Esimesel sammul leitakse uued kaalud, kalibreerides disainikaale lisainformatsiooniga. Teise sammul kalibreeritakse saadud kaale ühistelt tunnustelt saadud informatsiooniga nii, et saavutatakse kooskõla varasemast uuringust leitud hinnangutega. Särndal ja Traat (2009) pakkusid välja alternatiivse lähenemise kooskõlaliste hinnangute saamiseks – AC-kalibreerimise. Sarnaselt korduvkaalumisega kalibreeritakse disainikaale nii abiinformatsiooniga kui ka ühiste tunnustega, kuid erinevalt korduvakaalumisest tehakse see ühel sammul. AC-kalibreeritud kaaludega leitud hinnangud saavutavad kooskõla nii lisainformatsiooniga kui ka varasema uuringu hinnangutega, korduvkaalumise korral saadakse kooskõla vaid ühiste tunnuste teadaolevate hinnangutega.

Tulenevalt uuringu olemusest ja eesmärgist on mõnikord otstarbekas kasutada suurusega võrdelise tõenäosusega valikut. Antud disaini kasutatakse juhul, kui leidub taustatunnus, mis küllaltki hästi iseloomustab uuritava tunnuse muutumist – kui uuritav tunnus on oodatavalt ligikaudu võrdeline antud taustatunnusega. Antud bakalaureuse töö pühendubki AC-kalibreerimise ja korduvkaalumise meetodite rakendamisele suurusega võrdelise tõenäosusega disainide korral. Töös käsitleme kahte konkreetset disaini – Poissoni ja hüpergeomeetrilist (HG) valikut –, millest esimene on tagasipanekuta ja juhusliku valimimahuga ning teine tagasipanekuga ja fikseeritud valimimahuga disain.

Käesoleva töö eesmärk on anda ülevaade kalibreerimisest, korduvkaalumisest ja AC-kalibreeritud hinnangust. Peamine eesmärk on tuletada kooskõlaliste hinnangute valemid HG ja Poissoni valikudisaini jaoks ja hiljem simulatsioonülesandega neid kontrollida ning võrrelda kahe kooskõlaliste hinnangute saamise meetodi – AC-kalibreerimise ja korduvkaalumise – käitumist.

Töö esimeses peatükis toome sisse edasiseks vajalikud mõisted ja valemid, anname ülevaate suurusega võrdelise tõenäosusega valikust üldiselt ning kahest konkreetsest erijuhust – Poissoni ja hüpergeomeetrilisest valikudisainist. Teises peatükis tutvustame AC-kalibreerimise ja korduvkaalumise meetodeid ning tuletame valemid kahele vaatluse all olevale disainile. Töö viimases osas anname ülevaate läbiviidud simulatsioonülesandest ja saadud tulemustest.

1 Mõisted ja definitsioonid

1.1 Üldkogumi kogusumma hindamine

Olgu $U=(1,2,\ldots,N)$ N elemendist koosnev lõplik üldkogum ehk populatsioon. Juhuslikku vektorit $\mathbf{I}=(I_1,I_2,\ldots,I_N)$, kus komponent I_i näitab objekti i $(i\in U)$ võimalikke valikute arvu, nimetatakse valikuvektoriks. Tagasipanekuga (TGA) disainide korral $I_i\in\{0,1,2,\ldots\}$ $\forall i$ ja tagasipanekuta (TTA) disainide korral $I_i\in\{0,1\}$ $\forall i$. Vektori $\mathbf{I}$ realisatsiooniks on valim $\mathbf{k}=(k_1,k_2,\ldots,k_N)$, kus k_i näitab objekti i $(i\in U)$ valimisse sattumiste arvu. Olgu $s\subseteq U$ realiseerunud valim, st $s=\{i:k_i\geq 1, i\in U\}$. Valikuvektori $\mathbf{I}$ jaotus $p(\mathbf{k})=P(\mathbf{I}=\mathbf{k})$, $\sum_{\mathbf{k}} p(\mathbf{k})=1$ on valikudisain. Valikudisaini iseloomustavad selle karakteristikud, millest tähtsaimad on vektori $\mathbf{I}$ momendid $E(I_i)$, $V(I_i)$ ja $Cov(I_i,I_j)$. Eeldatakse, et $E(I_i)>0$ $\forall i$ ehk igal üldkogumi elemendil on positiivne valimisse kuulumise tõenäosus.

TTA disainide korral on I_i Bernoulli jaotusega, $I_i\sim Be(\pi_i)$, ja momendid on

$$E(I_i)=\pi_i, \qquad V(I_i)=\pi_i(1-\pi_i), \qquad Cov(I_i,I_j)=\pi_{ij}-\pi_i\pi_j,$$

kus $\pi_i=P(I_i=1)$ ja $\pi_{ij}=P(I_i=1,\ I_j=1)=E(I_iI_j)$ on vastavalt esimest ja teist järku kaasamistõenäosused. TGA disainide korral on I_i binoomjaotusega, $I_i\sim B(n,p_i)$, ja momendid on

$$E(I_i)=np_i, \qquad V(I_i)=np_i(1-p_i), \qquad Cov(I_i,I_j)=-np_ip_j,$$

kus p_i on tõenäosus, millega objekti i võidakse valida valimisse ühel sammul, $\sum_U p_i=1$.

Valikudisain võib olla kas fikseeritud või juhusliku valimimahuga. Üldjuhul sõltub valimimaht disainist ja on juhuslik suurus, mis avaldub valikuindikaatorite abil järgmiselt:

$$n_s=\sum_U I_i\,. \tag{1.1}$$

Valikudisaini nimetatakse fikseeritud mahuga disainiks, kui iga valimi realisatsiooni korral $\sum_U I_i\equiv n$.

Olgu $\mathbf{y}: m\times 1$ m uuritava tunnuse veeruvektor ja $\mathbf{y}_i$ väärtuste veeruvektor elemendil i. Vektor $\mathbf{y}$ võib sisaldada nii pidevaid kui ka nominaalseid tunnuseid. Nihketa hinnanguks üldkogumi U summadele $\mathbf{Y} = \sum_U \mathbf{y}_i$ on

$$\hat{\mathbf{Y}} = \sum_U a_i \mathbf{y}_i \,, \tag{1.2}$$

kus a_i on disainikaal, mis avaldub järgnevalt:

$$a_i = I_i / E(I_i) \,.$$

Kuna $a_i = 0$ mittevalitud objektide jaoks, siis on summa (1.2) valimilt mõõdetud suuruste $\mathbf{y}_i$ kaalutud summa ja konkreetse realiseerunud valimi s korral avaldub summa (1.2) järgmiselt:

$$\hat{\mathbf{Y}} = \sum_s a_i \mathbf{y}_i \,. \tag{1.3}$$

Objekti i kaal sõltub valikudisainist – TTA disainide korral $a_i = 1/E(I_i)$ ja TGA disainide korral $a_i = k_i / E(I_i)$. TTA disainide korral tunneme hinnangut (1.2) kui Horwitz-Thompsoni (HT) ja TGA disainide korral kui Hansen-Hurwitzi (HH) hinnangut.

1.2 Osakogumi kogusumma hindamine

Osakogumiteks nimetatakse kas uuritava tunnuse väärtuste või mõne taustatunnuse järgi klassifitseeritavaid üldkogumi alamhulki U_d, $U_d \subset U$. Sagedasti pakuvad huvi just erinevate parameetrite (nt kogusumma, keskmine, osakaal) hinnangud osakogumites. Osakogumiteks võivad olla näiteks haldusjaotusel põhinevad üksused, etnilised grupid antud riigis jms.

Olgu üldkogum U jagatud D osakogumiks ja tähistame osakogumeid $d = 1, 2, \dots, D$. Osakogumi U_d mahuks on N_d ja osakogumite mahud summeeruvad üldkogumi mahuks – $\sum_{d=1}^{D} N_d = N$. Objekt $i \in U_d$ kuulub valimisse, kui $I_i \geq 1$. Seega on osakogumisse U_d kuuluvate valimisse sattunud objektide $(i \in U_d)$ arv $n_d = \sum_{U_d} I_i$. Üldjuhul ei võeta valimit eraldi igast huvipakkuvast osakogumist, vaid see võetakse üldkogumist tervikuna, ning seetõttu on ka valimi suurus osakogumis juhuslik. Oodatavaks valimimahuks osakogumis U_d on $E(\mathrm{n}_d) = \sum_{U_d} E(I_i)$.

Osakogumite hindamiseks teeme osakogumitesse kuulumist määravad indikaatortunnused. Läbi nende defineerime uued tunnused, mille abil saame hinnata uuritavate tunnuste parameetreid osakogumites. Olgu δ_i^d indikaatortunnus, mis näitab kuuluvust osakogumisse U_d :

$$\delta_i^d = \begin{cases} 1, \text{ kui } i \in U_d; \\ 0, \text{ mujal.} \end{cases}$$

Osakogumi summa $\mathbf{Y}_d$ hindamiseks loome uue tunnuse:

$$\mathbf{y}_i^d = \delta_i^d \mathbf{y}_i = \begin{cases} \mathbf{y}_i, \text{ kui } i \in U_d; \\ 0, \text{ mujal.} \end{cases} \tag{1.4}$$

Kuna $\mathbf{y}_i^d = 0$, kui objekt $i \notin U_d$, saame osakogumi summa $\mathbf{Y}_d = \sum_{U_d} \mathbf{y}_i$ avaldada uue tunnuse summana üle üldkogumi:

$$\mathbf{Y}_d = \sum_U \mathbf{y}_i^d = \sum_U \delta_i^d \mathbf{y}_i \ .$$

Nihketa hinnang osakogumi U_d kogusummale $\mathbf{Y}_d$ järeldub avaldistest (1.2) ja (1.3) :

$$\hat{\mathbf{Y}}_d = \sum_U a_i \mathbf{y}_i^d = \sum_U a_i \delta_i^d \mathbf{y}_i = \sum_{U_d} a_i \mathbf{y}_i = \sum_{s_d} a_i \mathbf{y}_i \ , \tag{1.5}$$

kus $s_d = s \cap U_d$.

1.3 Suurusega võrdelise tõenäosusega valik

Suurusega võrdelise tõenäosusega valiku seletamisel võtame uuritava tunnuse ühedimensionaalseks. Vaadeldes HT või HH hinnangut $\hat{Y} = \sum_s a_i y_i = \sum_s y_i / E(I_i)$ näeme, et disainid, mille korral $E(I_i)$ on võrdelised y väärtustega, st $y_i / E(I_i) = \varphi,\ i = 1, 2, \ldots N$ ja φ on konstant, puudub hinnangul $\hat{Y}$ valimist valimisse varieeruvus. Kuna hinnang $\hat{Y}$ on nihketa, siis antud juhul oleksid need hinnangud täpsed. Niisuguse disaini konstrueerimine ei ole aga praktikas võimalik, kuna uuritav tunnus y ei ole eelnevalt teada. Praktikas kasutatakse tunnuse y asemel temaga tugevalt korreleeritud taustatunnust v, milleks sobib sageli objekti suuruse mingi näitaja. Sellest ka nimetus – suurusega võrdelise tõenäosusega valik (Traat, Inno, 1997, lk 121). Disaini nimetatakse alternatiivselt ka PPS (*Probabilities Proportional to Size*) valikuks.

Järgnevalt anname ülevaate disaini põhimõttest illustreeriva näite varal raamatust „Sampling: Design and Analysis“ (Lohr, 1999, lk 181-184).

Näide:

Oletame, et linnas on neli supermarketit, mille suurused on vahemikus 100 m^2–1000 m^2. Eesmärgiks on leida hinnang eelmise kuu kogumüügile linnas, võttes valimisse vaid ühe poodidest. Me võime eeldada, et suuremates supermarketites on läbimüük ja ka müügi dispersioon suurem. Olgu $\pi_i = P($pood i osutub valituks), $i = 1, 2, 3, 4$. Suurused π_i^{PPS} ja π_i^{LJV} tähistavad kaasamistõenäosusi vastavalt PPS ja LJV korral. Tähistame läbimüügi (tuhandetes) eelmises kuus y_i-ga. Kuna poe A põrandapind moodustab $1/16$ kõikide kaupluste pinnast, siis on tema kaasamistõenäosus $1/16$. Sarnaselt leitakse kaasamistõenäosused ka teistele supermarketitele.

Tabel 1.1: Kaasamistõenäosused PPS disaini korral

Pood	Suurus (m^2)	y_i	π_i^{PPS}
A	100	11	$1/16$
B	200	20	$2/16$
C	300	24	$3/16$
D	1 000	245	$10/16$
Kokku	1 600	300	1

Poe A kaaluks PPS disaini korral saame $1/\pi_i^{PPS} = 16$. Kui supermarketi suurus on ligikaudu võrdeline läbimüügiga antud poes, siis eeldatavasti moodustab poe A müük $1/16$ kogu läbimüügist linnas. Seega korrutades A müügi 16-ga saame hinnangu kogu linna eelmise kuu läbimüügile. Kasutades hinnangut (1.3) saame iga realiseerunud valimi s korral leida hinnangud müügile linnas (Tabel 1.2).

Tabel 1.2: Hinnangud PPS valiku ja LJV korral

Valim	y_i	PPS			LJV		
		π_i^{PPS}	$\hat{Y}^{PPS}$	$(\hat{Y}^{PPS}-Y)^2$	π_i^{LJV}	$\hat{Y}^{LJV}$	$(\hat{Y}^{LJV}-Y)^2$
$\{A\}$	11	1/16	176	15 375	1/4	44	65 536
$\{B\}$	20	2/16	160	19 600	1/4	80	48 400
$\{C\}$	24	3/16	128	29 584	1/4	96	41 616
$\{D\}$	245	10/16	392	8 464	1/4	980	462 400

Paneme tähele, et antud olukorras on valimi saamise tõenäosus võrdne objekti i kaasamistõenäosusega ja seega on ka hinnangud nihketa, st

$$E(\hat{Y}^{PPS}) = \sum_{i=1}^{4} P(s)\frac{y_i}{\pi_i^{PPS}} = \sum_{i=1}^{4} \pi_i^{PPS} \frac{y_i}{\pi_i^{PPS}} = Y,$$

kus $P(s)$ on valimi s saamise tõenäosus. Hinnangu kui juhusliku suuruse dispersioon on kujul

$$D(\hat{Y}) = E(\hat{Y}-Y)^2 = \sum_{i=1}^{4} P(s)(\hat{Y}-Y)^2 .$$

Antud näite korral saame $D(\hat{Y}^{PPS}) = 14\ 248$.

Võrdluseks leiame ka hinnangud ja dispersiooni lihtsa juhuvaliku korral (Tabelis 1.2). Ka antud juhul on hinnangud nihketa, aga dispersioon on suurem – $D(\hat{Y}^{LJV}) = 154\,488$. Seega annab lisainformatsiooni kasutamine PPS disaini puhul väiksema hajuvusega hinnangud.

Järgnevalt anname formaalse ülevaate suurusega võrdelise tõenäosusega valikust. Eeldame, et enne uuringu läbiviimist teame taustatunnuse v väärtusi, kusjuures me eeldame, et tunnus v on ligikaudu võrdeline uuritava tunnusega y. Disaini moodustamisel teeme eelduse, et oodatavad valikute arvud on võrdelised taustatunnuse väärtustega, st $E(I_i) \propto v_i$ ($E(I_i) = cv_i$, kus c on mingi positiivne konstant), ja tähistame $V = \sum_{i=1}^{N} v_i$. Oodatava valimimahu avaldisest

$$E(n_s) = \sum_{i=1}^{N} E(I_i) = \sum_{i=1}^{N} cv_i = c\sum_{i=1}^{N} v_i$$

saame, et c avaldub kujul $c = E(n_s)/V$. Elemendi i oodatav valikute arv PPS disaini korral on seega

$$E(I_i) = \begin{cases} E(n_s)v_i/V, \text{ juhusliku valimimahuga disainide korral;} \\ nv_i/V, \text{ fikseeritud mahuga disainide korral.} \end{cases}$$

Üldjuhul, tegemata kitsendust fikseeritud valimimahu kohta, saame kaasamistõenäosused TTA disainide korral $\pi_i = E(n_s)v_i/V$ ja valikutõenäosused TGA disainide korral $p_i = v_i/V$. Valimisse sattunud objekti i $(i \in s)$ kaal on $a_i = V/E(n_s)v_i$ TTA disainide korral ja $a_i = k_i V/E(n_s)v_i$ TGA disainide korral.

Kui tunnus v on suure hajuvusega, siis TTA disainide puhul ei pruugi olla võimalik võtta valimit nii, et kehtiks $\pi_i \propto v_i$. On selge, et tingimus $\pi_i \leq 1$ peab olema rahuldatud. Valimimahu $n=1$ korral on viimane nõue täidetud $\forall \pi_i$. Kui aga $n>1$ ja mõned väärtused v_i on suured, võib mõne elemendi i korral kehtida

$$\frac{nv_i}{V} \geq 1, \tag{1.6}$$

mis on vastuolus tingimusega $\pi_i \leq 1$. Üks võimalik meetod, kuidas sellest probleemist üle saada, on järgmine. Me määrame kõikide elementide jaoks, mille korral kehtib (1.6), $\pi_i = 1$ ja ülejäänud elemendid kaasatakse valimisse suurusega võrdelise tõenäosusega

$$\pi_i = (n - n_A)\frac{v_i}{\sum_{U-A} v_i},$$

kus A on n_A-elemendiline üldkogumi U alamhulk, mis sisaldab kõiki elemente i, mille korral kehtib (1.6). Vajadusel korratakse protseduuri, kuni $\forall i$ korral kehtib $\pi_i \leq 1$ (Särndal, Swensson, Wretman 2003, lk 89-90).

Järgnevas kahes peatükis anname ülevaate Poissoni ja hüpergeomeetrilisest valikust. Antud disainid on teatud eeldustel PPS disaini erijuhud, kusjuures esimene neist on juhusliku valimimahuga ja TTA ning teine fikseeritud valimimahuga ja TGA disain. Mõlemas peatükis eeldame, et uuritav tunnus ja taustatunnus on ühedimensionaalsed.

1.4 Poissoni valik

Selles peatükis defineerime esmalt Bernoulli valiku ja seejärel Poissoni valiku, mis on Bernoulli valiku üldistus, ja näitame kaasamistõenäosuste π_i mõne teadaoleva suurusega võrdeliselt valimise tähtsust Poissoni disaini korral.

Definitsioon 1.1 Bernoulli valikuks nimetatakse valikut, mille korral üldkogumi objektide kaasamisindikaatorid $I_1, I_2, \ldots, I_N$ on sõltumatud sama Bernoulli jaotusega juhuslikud suurused, st I_i jaotus avaldub iga objekti i $(i = 1, 2, \ldots, N)$ korral valemitega

$$P(I_i = 1) = \pi, P(I_i = 0) = 1 - \pi, \; 0 < \pi < 1.$$

Poissoni valiku korral loobume kaasamistõenäosuste konstantsuse nõudest, st elemendi i $(i \in U)$ kaasamistõenäosus võib olla suvaline π_i, $0 < \pi_i \le 1$.

Definitsioon 1.2 Poissoni valikuks nimetatakse valikut, mille korral üldkogumi objektide kaasamisindikaatorid $I_1, I_2, \ldots, I_N$ on sõltumatud juhuslikud suurused jaotusega $P(I_i = 1) = \pi_i$ ja $P(I_i = 0) = 1 - \pi_i$, kus suurused π_i on fikseeritud, $i = 1, 2, \ldots, N$.

Poissoni valik on TTA ja juhusliku valimimahuga disain, mille korral valimimaht on antud avaldisega (1.1). Valiku efektiivsuse tõstmiseks tuleks määrata elementide kaasamistõenäosused nii, et nad minimiseeriksid hinnangu dispersiooni. Eeldades, et $y_i > 0 \; \forall i$, saame dispersiooni minimiseerivaks tõenäosuseks iga $i = 1, 2, \ldots, N$ korral

$$\pi_i = n_0 y_i \Big/ \sum\nolimits_U y_i \, ,$$

kus $n_0 = E(n_s)$ on fikseeritud oodatav valimimaht. Kuna aga y_i väärtused pole terves üldkogumis teada, siis praktikas kasutatakse kaasamistõenäosuste π_i määramisel mõnda taustatunnust v, mis on teada iga $i \in U$ jaoks ja on ligikaudu võrdeline tunnusega y. Seega on Poissoni disain efektiivseim, kui kaasamistõenäosused on määratud mingi teadaoleva suurusega võrdeliselt. Sellel juhul $E(I_i) = \pi_i = n_0 v_i / V$ ja HT hinnang üldkogumi summale on

$$\hat{\mathbf{Y}} = \sum\nolimits_U a_i \mathbf{y}_i = \sum\nolimits_U \frac{I_i}{E(I_i)} \mathbf{y}_i = \sum\nolimits_s \frac{V}{n_0 v_i} \mathbf{y}_i \, .$$

Kuigi valimi võtmise faasis eeldasime, et uuritav tunnus on ühedimensionaalne ja seotud taustatunnusega v, siis hindamisfaasis võime hinnangud leida ka mitmele uuritavale tunnusele kasutades sama kaalukomplekti.

Grafström (2010) mainib Poissoni valiku puudusena juhuslikku valimimahtu, kuna sageli saadakse selle tulemusena vähem tõhusad hinnangud, eelisena toob ta aga välja valiku kerge teostamise.

1.5 Hüpergeomeetriline valik

Hüpergeomeetriline (HG) valik on laialdaselt kasutusel leibkonnauuringute läbiviimisel ja seetõttu anname järgnevalt ülevaate selle tekkemehhanismist just isik-leibkond kontekstis. Olgu meil M-isikuline üldkogum $U^{isik} = (1, 2, \ldots, M)$, mis on jaotunud N leibkonnaks nii, et m_i inimest kuulub leibkonda i, $i = 1, 2, \ldots, N$, $\sum_{i=1}^{N} m_i = M$. Valime nüüd isikute üldkogumist n isikut LJV TTA disainiga ja oleme saanud vektori $\mathbf{k} = (k_1, k_2, \ldots, k_N)$, kus element k_i näitab mitu i-nda leibkonna esindajat sattus valimisse. Seega on $\mathbf{k}$ leibkondade valim, mille tõenäosus on kujul:

$$p(\mathbf{k}) = \frac{\prod_{i=1}^{N} C_{m_i}^{k_i}}{C_M^n}, \text{ kui } |\mathbf{k}| = n,$$

kus $|\mathbf{k}| = \sum_{i=1}^{M} k_i$ on valimimaht. Valikuvektori $\mathbf{I} = (I_1, I_2, \ldots, I_N)$ jaotus $p(\mathbf{k}) = P(\mathbf{I} = \mathbf{k})$ on hüpergeomeetriline disain leibkondade üldkogumil:

$$\mathbf{I} \sim HG(M, n; m_1, m_2, \ldots, m_N).$$

HG disaini tekkemehhanism annab idee valimi võtmiseks suvalisest üldkogumist $U = (1, 2, \ldots, N)$ disainiga $\mathbf{I} \sim HG(M, n; m_1, m_2, \ldots, m_N)$. Selleks moodustame uue üldkogumi U' nii, et $\forall i \in U$ on uues üldkogumis esindatud m_i korda. Seejärel võtame üldkogumist U' n-elemendilise LJV TTA disainiga valemi. Saadud vektor $\mathbf{k} = (k_1, k_2, \ldots, k_N)$, milles on esialgse üldkogumi U objektide valikute arvud, ongi hüpergeomeetriline valim üldkogumist U.

Hüpergeomeetrilise jaotuse momendid on (Traat, Ilves, 2007):

$$E(I_i) = np_i, \qquad V(I_i) = cnp_i(1 - p_i), \qquad Cov(I_i, I_j) = -cnp_i p_j,$$

kus $c = (M - n)/(M - 1)$ ja $p_i = m_i/M$. Disaini kaalud saavad seega kuju $a_i = I_i/E(I_i) = I_i/np_i$ ja kogusumma hinnang on kujul:

$$\hat{\mathbf{Y}} = \sum\nolimits_U a_i \mathbf{y}_i = \sum\nolimits_U \frac{I_i}{np_i} \mathbf{y}_i = \sum\nolimits_s \frac{Mk_i}{nm_i} \mathbf{y}_i .$$

2 Kalibreerimine ja kooskõlaline hindamine

2.1 Kalibreerimine

Selles peatükis eeldame, et me tahame hinnanguid üldkogumi kogusummadele olukorras, kus meil on täiendavat informatsiooni kas registritest või varasematest uuringututest. Seda teavet saab kasutada hinnangute täpsuse suurendamiseks. Edasipidi kasutame lisainformatsiooni sisaldava tunnuse sünonüümina väljendit abitunnus. Üks meetod, mis võimaldab lisainformatsiooni hinnangutesse kaasata, on kalibreerimine. Ülevaade antud meetodist ja ka alternatiivsest lähenemisest – üldisest regressioonhinnangust (GREG) – on antud artiklis Särndal (2007). Järgnev kokkuvõtlik ülevaade põhinebki antud artiklil.

Eeldame, et üldkogumist $U = (1, 2, \ldots, N)$ on võetud tõenäosuslik valim s. Valikudisain määrab igale elemendile i $(i \in U)$ vastava oodatava valikute arvu $E(I_i)$. Olgu kaalud $a_i = I_i / E(I_i)$. Eeldame, et $\forall i \in s$ korral on teada uuritavate tunnuste väärtused $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{im})'$. Meie eesmärgiks on hinnata üldkogumi summasid $\mathbf{Y} = \sum_U \mathbf{y}_i$. Olgu $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})'$ p-dimensionaalne lisainformatsiooni vektor, mille kogusumma $\mathbf{X} = \sum_U \mathbf{x}_i$ on teada.

Kalibreerimise idee seisneb selles, et leitakse uued lisainformatsiooni arvesse võtvad kaalud, mis rahuldavad kalibreerimisvõrrandeid

$$\sum_s w_i \mathbf{x}_i = \mathbf{X}, \tag{2.1}$$

ja seejärel leitakse summale $\mathbf{Y}$ hinnang kujul $\hat{\mathbf{Y}}^{CAL} = \sum_s w_i \mathbf{y}_i$, kusjuures eesmärk on saada ligikaudselt nihketa hinnang. Tingimusest $\hat{\mathbf{Y}}^{CAL} = \hat{\mathbf{Y}} + \sum_s (w_i - a_i)\mathbf{y}_i$, kus $\hat{\mathbf{Y}}$ on (1.2), järeldub, et $E(\hat{\mathbf{Y}}^{CAL}) - \mathbf{Y} = E\left(\sum_s (w_i - a_i)\mathbf{y}_i\right)$, ja seega peab ligikaudse nihketuse korral kehtima $E\left(\sum_s (w_i - a_i)\mathbf{y}_i\right) \approx 0$. Järelikult peavad olema uued kaalud algsetele võimalikult lähedased.

Kaalude kalibreerimiseks kasutatakse peamiselt kahte lähenemist – kauguse minimiseerimist ja instrumentvektori meetodit. Antud töö raames kasutame instrumentvektori lähenemist. Selle meetodi korral saame uued kaalud kujul

$$w_i = a_i(1+\boldsymbol{\lambda}'\mathbf{z}_i)\,. \tag{2.2}$$

Vektor $\boldsymbol{\lambda}'$ avaldub tingimustest (2.1) ja (2.2) järgmiselt:

$$\boldsymbol{\lambda}' = \left(\sum\nolimits_U \mathbf{x}_i - \sum\nolimits_s a_i\mathbf{x}_i\right)' \left(\sum\nolimits_s a_i\mathbf{z}_i\mathbf{x}_i'\right)^{-1}. \tag{2.3}$$

Tähistades $\mathbf{M} := \sum\nolimits_s a_i\mathbf{z}_i\mathbf{x}_i'$ saame avaldise viia kujule

$$\boldsymbol{\lambda}' = \left(\mathbf{X} - \hat{\mathbf{X}}\right)' \mathbf{M}^{-1},$$

kus $\hat{\mathbf{X}}$ on HT hinnang. Vektorit $\mathbf{z}_i : p\times 1$ nimetatakse instrumentvektoriks ja ta on vabalt valitav, kusjuures sõltumata valikust rahuldavad kaalud w_i kalibreerimisvõrrandeid (2.1). Tavaliselt valitakse $\mathbf{z}_i = \mathbf{x}_i$, aga Särndal (2007) näitas, et ka küllaltki äärmusliku $\mathbf{z}_i$ tahtliku valiku korral saadakse väga häid tulemusi.

Nagu eelnevalt mainitud, on hinnang kogusummale $\mathbf{Y}$ leitav valimi kaalutud summana $\hat{\mathbf{Y}}^{CAL} = \sum\nolimits_s w_i\mathbf{y}_i$. Osakogumite hindamiseks kasutame vektori $\mathbf{y}_i$ asemel vektorit (1.4). Hinnanguks saame $\hat{\mathbf{Y}}_d^{CAL} = \sum\nolimits_s w_i\mathbf{y}_i^d = \sum\nolimits_s w_i\delta_i^d\mathbf{y}_i$.

Kalibreerimise kasutamise motiividena nimetasid Estevao ja Särndal (2000) kolme aspekti:

1. *Kooskõlalisus.* Kaalumissüsteem, mis rahuldab kalibreerimisvõrrandeid (2.1) on atraktiivne, sest see produtseerib iga abitunnuse jaoks täpse teadaoleva üldkogumi summa.
2. *Lähedus esialgsete disainipõhiste kaaludega.* Esialgsete kaaludega $a_i = I_i / E(I_i)$ hindamisel saadakse nihketa hinnangud. Seega peab igasugune erinevus neist kaaludest olema väike, et säilitada vähemalt asümptootiline nihketus.
3. *Kontroll abitunnuste summade üle.* Mida rohkem teadaolevaid summasid me lisainformatsioonina kasutame, seda parem on eeldatavalt kaalumissüsteem. Kalibreeritud hinnangute dispersioon näitab kahanevat tendentsi, kui kaasata kalibreerimisse rohkem abitunnuseid.

2.2 Kooskõlaline hindamine

Eeldame, et viiakse läbi uuring, mille üheks eesmärgiks on hinnata osakogumite kogusummasid. Üks võimalus hinnangu saamiseks on kasutada kas HT või HH hinnangut (1.5). Oletame nüüd aga, et meil on teada uuritava tunnuse kogusumma. Antud teadmine võib pärineda kas mõnest registrist või varasemast uuringust, mille korral uuriti seda sama meile huvipakkuvat tunnust, kuid meie uuringule vastavat osakogumiteks jagamist ei tehtud. Sellises olukorras oleks loomulik nõuda, et informatsioon kahest allikast oleks kooskõlas – käsitletavate osakogumite kogusummad peavad summeeruma varem teadaolevateks või teistest allikatest saadud suurusteks. Selleks peame aga tegema järgmised eeldused:

1. registrite ja uuringute andmed on võetud samal ajahetkel;
2. kõik andmeallikad käsitlevad sama populatsiooni;
3. kõikidel ühistel tunnustel on sama definitsioon.

Kui mõni nendest tingimustest ei kehti, siis kooskõla eri allikatest pärit andmete vahel ei peagi olema, sest andmed ei ole omavahel võrreldavad. Näiteks on selgesti tajutav, et 5 aastat tagasi läbiviidud rahvastiku etnilise koosseisu uuringu andmed ei pea olema kooskõlas tänapäevaste näitajatega. Esimene punkt on täidetud olukorras, kus me saame andmed reaalajas uuendatavast registrist, kaks uuringut viiakse läbi samal ajahetkel või aeg nende vahel on lubatavalt väike. Teiste tingimuste täitmist saab kindlustada uuringu planeerimisfaasis.

2.3 AC-kalibreerimine

Antud peatükis anname põgusa ülevaate professor Carl-Erik Särndali ja dotsent Imbi Traadi poolt arendatud kooskõlaliste hinnangute saamise meetodist – AC-kalibreerimisest. Oma nime on saanud meetod inglisekeelsetest sõnadest *auxiliary* (A, abitunnused) ja *common* (C, ühised tunnused). Edasises arutelus kasutame ka väljendeid A-informatsioon ja C-informatsioon. Nagu nimi vihjab on tegemist kalibreerimismeetodiga, mille puhul kasutatakse kalibreerimisel nii informatsiooni kahe erineva allika ühistest tunnustest kui ka täiendavat teavet (A-informatsioon). Kalibreerimine nii A- kui ka C-informatsiooniga tehakse ühel sammul.

Eeldame, et iga objekti i korral valimist $(i \in s)$ on teada vektorid $\mathbf{x}_i : p \times 1$ ja $\mathbf{y}_i : m \times 1$, kus $\mathbf{x}_i$ on lisainformatsiooni vektor, mille korral on teada üldkogumi summa $\mathbf{X} = \sum_U \mathbf{x}_i$, ja $\mathbf{y}_i$ on uuritavate tunnuste vektor. Me defineerime uued $(p+m)$ mõõtmelised veeruvektorid

$$\begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix}, i \in s, \qquad \begin{pmatrix} \mathbf{X} \\ \mathbf{Y}_o \end{pmatrix}, \qquad \begin{pmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \end{pmatrix},$$

kus $\mathbf{Y}_0$ on kas varasemast uuringust teadaolev kogusumma hinnangute vektor või registrist saadud suuruste vektor. Me eeldame, et esimesel juhul on $\mathbf{Y}_0$ ligikaudu nihketa ja suure täpsusega. Kui $\mathbf{Y}_0$ on saadud mõnest registrist, siis on ta üldkogumi tegelik summa $\mathbf{Y} = \sum_U \mathbf{y}_i$. Vektorid $\hat{\mathbf{X}} = \sum_U a_i \mathbf{x}_i$ ja $\hat{\mathbf{Y}} = \sum_U a_i \mathbf{y}_i$ on HT hinnangud praegusest uuringust. Instrumentvektori meetodil leitud kalibreeritud kaalud on kujul

$$w_{ACi} = a_i (1 + \boldsymbol{\lambda}'_{AC} \mathbf{z}_{ACi}), \tag{2.4}$$

kus $\mathbf{z}_{ACi} : (p+m) \times 1$ on instrumentvektor ja

$$\boldsymbol{\lambda}'_{AC} = \begin{pmatrix} \mathbf{X} - \hat{\mathbf{X}} \\ \mathbf{Y}_0 - \hat{\mathbf{Y}} \end{pmatrix}' \mathbf{M}^{-1} \text{ ning } \mathbf{M} = \sum_s a_i \mathbf{z}_{ACi} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix}'. \tag{2.5}$$

Maatriksil $\mathbf{M}$ leidub pöördmaatriks, kui vektorid $\mathbf{x}$ ja $\mathbf{y}$ on lineaarselt sõltumatud.

Kalibreerimisel saadud uued kaalud w_{ACi} on kooskõlas nii A-tunnustega kui ka C-tunnustega:

$$\sum_s w_{ACi} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y}_0 \end{pmatrix} \tag{2.6}$$

Kui me oleme kaalud w_{ACi} leidnud, siis saame nende abil hinnata kõiki huvipakkuvaid kogusummasid. Üldkogumi summa $\mathbf{Y} = \sum_U \mathbf{y}_i$ hinnanguks on

$$\hat{\mathbf{Y}}^{AC} = \sum_s w_{ACi} \mathbf{y}_i$$

ja osakogumi U_d summa hinnanguks on

$$\hat{\mathbf{Y}}_d^{AC} = \sum_{s_d} w_{ACi} \mathbf{y}_i = \sum_s w_{ACi} \mathbf{y}_i^d \ (d = 1, 2, \ldots, D). \tag{2.7}$$

Avaldistest (2.6) ja (2.7) järeldub, et $\sum_{d \in \mathfrak{D}} \hat{\mathbf{Y}}_d^{AC} = \mathbf{Y}_0$. Järelikult on osakogumite hinnangud kooskõlalised.

2.4 Korduvkaalumine

Korduvkaalumise (*repeated weighting* – RW) meetod on välja töötatud Hollandi riiklikus statistikaametis. Esialgne idee oli saada kooskõlalised sagedustabelid nii, et hinnatud marginaalid vastaksid varasemalt teadaolevatele suurustele. Antud peatüki eesmärk on anda põgus ja ülevaatlik tutvustus meetodi põhimõttest. Huvilistel on võimalik täpsemalt lugeda artiklitest Kroese ja Renssen (1999), Houbiers (2004) ning Knottnerus ja van Duin (2006).

Korduvkaalumine on sisuliselt kaheetapiline kalibreerimine. Esimesel etapil leiame kaalud w_i kujul (2.2) nii, et nad rahuldaksid kalibreerimisvõrrandeid (2.1). Teisel etapil kalibreerime kaale w_i kasutades C-informatsiooni. Uued kaalud saame kujul

$$w_{RWi} = w_i(1+\boldsymbol{\lambda}'_{RW}\mathbf{z}_{RWi}), \tag{2.8}$$

kus $\mathbf{z}_{RWi} : m\times 1$ on instrumentvektor ja

$$\boldsymbol{\lambda}'_{RW} = \left(\mathbf{Y}_0 - \hat{\mathbf{Y}}^{CAL}\right)' \mathbf{M}^{-1}, \qquad \mathbf{M} = \sum\nolimits_s w_i \mathbf{z}_{RWi}\mathbf{y}'_i, \qquad \hat{\mathbf{Y}}^{CAL} = \sum\nolimits_s w_i \mathbf{y}_i. \tag{2.9}$$

Vektor $\mathbf{Y}_0$ võib olla nagu eelmiseski peatükis kas üldkogumi tegelike summade või varajasemast uuringust leitud hinnangute vektor. Erinevalt AC-kalibreerimisest on uued kaalud w_{RWi} kooskõlalised ainult C-informatsiooniga, st

$$\sum\nolimits_s w_{RWi}\mathbf{y}_i = \mathbf{Y}_0. \tag{2.10}$$

Kaaludе w_{RWi} abil saame hinnangu üldkogumi kogusummadele kujul

$$\hat{\mathbf{Y}}^{RW} = \sum\nolimits_s w_{RWi}\mathbf{y}_i$$

ja osakogumi U_d summade hinnanguks on

$$\hat{\mathbf{Y}}_d^{RW} = \sum\nolimits_{s_d} w_{RWi}\mathbf{y}_i = \sum\nolimits_s w_{RWi}\mathbf{y}_i^d \ \ (d = 1, 2, \ldots, D) \tag{2.11}$$

Avaldistest (2.10) ja (2.11) järeldub, et $\sum_{d\in\mathfrak{D}} \hat{\mathbf{Y}}_d^{RW} = \mathbf{Y}_0$, ja seega on hinnangud (2.11) kooskõlalised.

2.5 AC-kalibreerimine ja korduvkaalumine Poissoni ja HG disaini korral

Kalibreeritud ja AC-kalibreeritud kaalud (2.2) ja (2.8) ning sellest tulenevalt ka kaalud (2.4) sõltuvad konkreetsest valikudisainist, sest nende arvutamisel võetakse aluseks esialgsed disainikaalud. Iga elemendi i korral realiseerunud valimist s $(i \in s)$ on Poissoni valiku korral kaalud $a_i^{Po} = V / n_0 v_i$ ja HG disaini korral $a_i^{HG} = Mk_i / nm_i$. Seega on Poissoni valiku korral kaalud (2.2), (2.4) ja (2.8) kujul

$$w_i = \frac{V}{n_0 v_i}(1+\boldsymbol{\lambda}'\mathbf{z}_i), \quad w_{ACi} = \frac{V}{n_0 v_i}(1+\boldsymbol{\lambda}'_{AC}\mathbf{z}_{ACi}), \quad w_{RWi} = \frac{V}{n_0 v_i}(1+\boldsymbol{\lambda}'\mathbf{z}_i)(1+\boldsymbol{\lambda}'_{RW}\mathbf{z}_{RWi})$$

ja HG valiku korral kujul

$$w_i = \frac{Mk_i}{nm_i}(1+\boldsymbol{\lambda}'\mathbf{z}_i), \quad w_{ACi} = \frac{Mk_i}{nm_i}(1+\boldsymbol{\lambda}'_{AC}\mathbf{z}_{ACi}), \quad w_{RWi} = \frac{Mk_i}{nm_i}(1+\boldsymbol{\lambda}'\mathbf{z}_i)(1+\boldsymbol{\lambda}'_{RW}\mathbf{z}_{RWi}) .$$

Nagu eelnevalt mainitud, on instrumentvektorid $\mathbf{z}_i$, $\mathbf{z}_{ACi}$ ja $\mathbf{z}_{RWi}$ vabalt valitavad. Estevao ja Särndal (2004) näitasid, et kalibreerimisel on optimaalne instrumentvektori valik fikseeritud valikudisaini ja vektori $\mathbf{x}$ korral

$$\mathbf{z}_i = a_i^{-1}\sum\nolimits_{j\in s}(a_i a_j - a_{ij})\mathbf{x}_j ,$$

kus $a_{ij} = \pi_{ij}^{-1}$. Poissoni valiku korral on valikuindikaatorid sõltumatud juhuslikud suurused ja seega $\pi_{ij} = \pi_i \pi_j$ ja $a_{ij} = 1/\pi_i \pi_j$. Optimaalne instrumentvektor saab kuju

$$\mathbf{z}_i = a_i^{-1}\sum\nolimits_{j\in s}(a_i a_j - a_{ij})\mathbf{x}_j = a_i^{-1}\sum\nolimits_{j\in s}(\frac{1}{\pi_i \pi_j} - \frac{1}{\pi_i \pi_j})\mathbf{x}_j$$

ehk on $p \times 1$ nullvektor. Maatriks $\mathbf{M} = \sum\nolimits_s a_i \mathbf{z}_i \mathbf{x}'_i$ on sellisel juhul nullmaatriks ja seetõttu ei leidu tema pöördmaatriksit. Võime sellisel juhul leida Moore-Penrose üldistatud pöördmaatriksi, mis antud juhul annaks tulemuseks nullmaatriksi (Kollo , von Rosen, 2005, lk 18) ja seega $w_i = a_i$ ehk Poissoni valiku puhul on algsed disainikaalud kalibreerimise mõttes asümptootiliselt optimaalsed. Probleem tekib siis, kui abitunnuste vektor on ühemõõtmeline, sest 0-i pöördväärtust ei eksisteeri. Seda juhtu me aga antud töös ei käsitle.

Kuigi meil on optimaalne $\mathbf{z}_i$ leitav, siis see hetkel meid palju ei aita, sest tahame saavutada kooskõla C-informatsiooniga ning algsete disainikaaludega hindamine seda ei garanteeri.

3 Simulatsioonid

Antud simulatsioonülesande eesmärk on testida peatükis 2.5 tuletatud valemeid ja võrrelda HT hinnanguid, AC-kalibreeritud ja RW hinnanguid nii Poissoni kui ka hüpergeomeetrilise valiku korral. Simulatsioonide jaoks kasutati reaalsetel andmetel põhinevat tehislikku üldkogumit. Esmalt määrati huvialused osakogumid ja leiti uuritavate tunnuste tegelikud kogusummad osakogumites. Seejärel võeti 1000 korda valimit nii Poissoni kui ka hüpergeomeetrilise valikuga ning leiti iga valimikorduse jaoks kogusummade hinnangud osakogumites. Tulemuste võrdlemiseks leiti hinnangute keskmised ning standardhälbed üle valimikorduste. Erinevate meetodite võrdlemiseks leiti täpsusnäitajad – suhteline nihe (RB – *relative bias*) ja suhteline ruutkeskmine viga (RRMSE – *relative root mean square error*).

Simulatsioonülesanne viidi läbi statistikapaketi R abil. Programmi kood on eraldi välja toodud töö lisas. Saadud tulemuste vormindamiseks ja kogumiseks ühtsetesse tabelitesse kasutati ka tarkvara Microsoft Excel.

3.1 Üldkogum

Tehislik üldkogum moodustati aastatel 2004-2007 läbiviidud Eesti Leibkonna Uuringu raames kogutud andmete põhjal. Üldkogumisse kuulub 17 540 leibkonda. Kuigi uuring ise hõlmas väga suurt hulka tunnuseid, valiti lihtsuse huvides tehislikku üldkogumisse vaid osa nendest. Iga leibkonna kohta oli teada järgmised tunnused.
Uuritavateks tunnusteks olid:

- netotulu (pidev tunnus) – leibkonna sissetulek koos maksudega uuringukuul (sisaldab sissetulekut palgatööst, tulu põllumajanduslikust tegevusest, omanditulu, siirdetulu ja finantsvahendite müügitulu);
- tarbimiskulu (pidev tunnus) – leibkonna tarbimiskulutused uuringukuul.

Abitunnusteks olid:

- siirdetulu (pidev tunnus) – riigilt või omavalitsuselt saadud rahaline toetus (nt pensionid, töötu abiraha, lastetoetus, toimetulekutoetus jne);
- alla 16 aastaste laste arv uuringu aasta 1. jaanuari seisuga (diskreetne tunnus).

Üldkogumi jagasime osakogumiteks järgnevate tunnuste abil:

- leibkonnapea majanduslik aktiivsus (binaarne tunnus) – väärtustega 1 – töötab, 0 – ei tööta;
- leibkonnapea sugu (binaarne tunnus) – väärtustega 1 – mees, 2 – naine;
- leibkonnapea haridustase (järjestustunnus) – väärtustega 1 – algtase, 2 – kesktase, 3 – kõrgtase.

Suurust iseloomustavaks taustatunnuseks oli „inimeste arv leibkonnas“ (diskreetne tunnus) ja hinnatavateks parameetriteks netotulu ja tarbimiskulude kogusummad osakogumites.

Üldkogum jagati tunnuste „leibkonnapea majanduslik aktiivsus“, „leibkonnapea sugu“ ja „leibkonnapea haridustase“ ristklassifitseerimisel 12 osakogumiks. Osakogumite suurused ja tunnuste „netotulu“ ja „tarbimiskulud“ karakteristikud on välja toodud Tabelis 3.1. Näeme, et

Tabel 3.1: Osakogumid ja uuritavate tunnuste karakteristikud

Leibkonnapea		Haridus-tase	d	N_d	%	Netosissetulek ($\times 10^3$)			Tarbimiskulud ($\times 10^3$)		
aktiivsus	sugu					Summa	Keskmine	Std	Summa	Keskmine	Std
Töötab	Mees	Algtase	1	712	4,1%	6343	8,9	7,8	5050	7,1	6,9
		Kesktase	2	4094	23,3%	41336	10,1	9,0	34535	8,4	8,2
		Kõrgtase	3	1840	10,5%	22510	12,2	12,7	19132	10,4	10,6
	Naine	Algtase	4	364	2,1%	2153	5,9	5,3	1832	5,0	6,0
		Kesktase	5	2393	13,6%	16961	7,1	7,1	14585	6,1	6,1
		Kõrgtase	6	2015	11,5%	19277	9,6	9,9	16345	8,1	8,4
Töötu	Mees	Algtase	7	923	5,3%	3964	4,3	3,5	3303	3,6	3,6
		Kesktase	8	1172	6,7%	5678	4,8	5,6	5409	4,6	7,1
		Kõrgtase	9	370	2,1%	2097	5,7	6,5	1820	4,9	4,9
	Naine	Algtase	10	1559	8,9%	5323	3,4	3,0	4467	2,9	2,9
		Kesktase	11	1478	8,4%	5545	3,8	3,6	5384	3,6	3,8
		Kõrgtase	12	620	3,5%	2655	4,3	4,6	2579	4,2	5,2
		Üldkogum		17540	100%	133841	7,6	8,5	114440	6,5	7,5

osakogumite suurused on väga erinevad – suurim osakogum moodustab 23,3% üldkogumist ja on ka kaks osakogumit, mis moodustavad vaid 2,1% üldkogumist. Suurim osakogum esindab leibkondasid, kus leibkonnapea on kesktaseme haridusega töötav mees. Väikseimate korral on leibkonnapeadeks algtaseme haridusega töötav naine ja kõrgtaseme haridusega töötu mees. Nagu sissejuhatuses öeldud, eeldavad AC-kalibreerimine ja korduvkaalumine piisavalt suurt valimit osakogumis, aga hetkel uurime simulatsioonülesandes ka hinnangute käitumist väikestes osakogumites.

Tabel 3.2: Uuritavate tunnuste keskmised sõltuvalt leibkonna suurusest

Leibkonna suurus	Keskmine netotulu	Keskmine tarbimiskulu
1	3 254	2 977
2	6 730	5 720
3	9 379	7 818
4	10 825	9 360
5	12 139	10 333
6	13 340	10 889
$\geq$ 7	13 013	11 389

Tabelis 3.2 on välja toodud keskmised netosissetulekud ja tarbimiskulud sõltuvalt leibkonna suurusest. Maksimaalne leibkonna suurus üldkogumis oli 14, aga kuna leibkondi, milles oli seitse või rohkem liiget, oli vähe, siis võtsime need kokku. Näeme, et keskmine netotulu ja tarbimiskulutused kasvavad leibkonna suuruse kasvades. Vaid leibkondades, kus on seitse või rohkem liiget, on keskmine netosissetulek väiksem kui kuueliikmeliste leibkondade korral. Paneme tähele, et uuritavad tunnused ei ole lineaarses sõltuvuses leibkonna suurusega – kasv uuritavate tunnuste keskmistes kahaneb leibkonna suuruse kasvades. Praktikas on aga raske leida olukorda, kus uuritav tunnus oleks taustatunnusega ligikaudselt lineaarselt seotud.

3.2 Simulatsiooni püstitus

Me tahame hinnata summaarset netotulu ja tarbimiskulu osakogumites. Suurust iseloomustavaks taustatunnuseks valime tunnuse „inimeste arv leibkonnas“. Me eeldame, et meil on teada summaarne netotulu ja tarbimiskulu või viimaste hinnangud ning abitunnuste summad üldkogumis. Seda informatsiooni kasutades leiame AC-kalibreeritud ja RW kaalud, mille abil hindame netosissetuleku ja tarbimiskulu kogusummasid osakogumites.

Antud töö raames viime läbi kaks simulatsiooni – Poissoni ja hüpergeomeetrilise valiku korral. Mõlema simulatsiooni käigus uurime kahte järgnevat juhtu:

- kogusummad $\mathbf{Y}_0$ on täpselt teada (nt registritest);
- kogusummad $\mathbf{Y}_0$ on hinnatud varajasemast uuringust.

Esimesel juhul arvutame $\mathbf{Y}_0$ tehislikust üldkogumist ja kasutame saadud vektorit kalibreerimisel. Teisel juhul võtame iga valimikorduse korral kaks valimit, millest esimene (U_{I} valim) täidab varajasema ja teine (U_{II} valim) käesoleva uuringu rolli. Mõlemad valimid võtame sama valikudisainiga. U_{I} valimist leiame hinnangud kogusummadele, mida seejärel kasutame U_{II} valimi pealt kalibreeritud kaalude leidmiseks. Tagamaks varajasemast uuringust leitud hinnangute täpsust, määrame U_{I} valimimahuks (Poissoni valiku korral oodatavaks valimimahuks) 2000 leibkonda. Valimimaht (Poissoni valiku korral oodatav valimimaht) U_{II} korral on 1000. Seega peame võtma 1000 sõltumatut U_{I} ja U_{II} valimit. Simulatsiooni viime läbi selliselt, et iga U_{II} valimi korral leiame hinnangud nii juhul, kus kogusummad on täpselt teada, kui ka juhul, kus kogusummad on U_{I} valimi pealt hinnatud.

Tabel 3.3: Valimimahud osakogumites üle valimikorduste Poissoni valiku korral

Osakogum	U_{I} valimimahud				U_{II} valimimahud			
	Keskmine	Osakaal	Min	Max	Keskmine	Osakaal	Min	Max
1	98	4,9%	73	129	49	4,9%	29	72
2	592	29,6%	531	659	296	29,6%	234	354
3	255	12,7%	207	301	127	12,7%	93	161
4	40	2,0%	22	61	20	2,0%	7	34
5	275	13,7%	224	323	137	13,7%	99	177
6	226	11,3%	180	270	113	11,3%	83	146
7	83	4,1%	55	117	41	4,1%	23	64
8	115	5,8%	88	151	58	5,8%	35	80
9	34	1,7%	17	53	17	1,7%	7	32
10	117	5,9%	82	149	58	5,8%	34	85
11	120	6,0%	89	156	60	6,0%	39	87
12	45	2,3%	24	66	23	2,3%	9	39
Kokku	2000	100%			999	100%		

Osakogumite valimimahud Poissoni ja hüpergeomeetrilise valiku korral on välja toodud Tabelites 3.3 ja 3.4 Näeme, et nii Poissoni kui ka HG disaini korral on mõnda valimisse sattunud väga vähe leibkondi osakogumitest 4 ja 9. Poissoni valiku korral on mõlemasse osakogumisse sattunud minimaalselt seitse ja HG valiku korral kuus inimest. Paneme tähele,

Tabel 3.4: Valimimahud osakogumites üle valimikorduste HG valiku korral

Osakogum	U_I valimimahud				U_{II} valimimahud			
	Keskmine	Osakaal	Min	Max	Keskmine	Osakaal	Min	Max
1	97	4,9%	70	127	48	4,8%	30	70
2	592	29,6%	530	655	297	29,7%	255	342
3	254	12,7%	215	305	127	12,7%	96	162
4	40	2,0%	24	58	20	2,0%	6	38
5	275	13,8%	224	319	137	13,7%	103	170
6	226	11,3%	183	272	114	11,4%	80	146
7	83	4,1%	52	117	41	4,1%	20	61
8	115	5,8%	86	146	58	5,8%	35	80
9	34	1,7%	19	53	17	1,7%	6	32
10	116	5,8%	78	155	58	5,8%	36	80
11	121	6,0%	89	164	60	6,0%	37	84
12	46	2,3%	25	71	23	2,3%	10	39
Kokku	2000	100%			1000	100%		

et valimites on leibkondi osakogumist 2 ligikaudu 30%, samas kui osakogum ise moodustab 23,3% üldkogumist. Ilmselt on selle põhjuseks võrreldes teiste osakogumitega keskmiselt suurem inimeste arv leibkonnas. Kuigi määrasime Poissoni valiku korral U_I ja U_{II} oodatavateks valimimahtudeks vastavalt 2000 ja 1000, siis keskmine U_{II} maht üle korduste oli 999.

Simulatsiooni käigus fikseerime instrumentvektorid järgmiselt:

$$\mathbf{z}_i = \mathbf{x}_i , \quad \mathbf{z}_{ACi} = \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix}, \quad \mathbf{z}_{RWi} = \mathbf{y}_i .$$

Vektor $\mathbf{x}_i : 2\times 1$ sisaldab informatsiooni saadud siirdetulu ja laste arvu kohta leibkonnas i, vektor $\mathbf{y}_i : 2\times 1$ netosissetuleku ja tarbimiskulutuste kohta.

3.3 Täpsusnäitajad

Hinnangute võrdlemiseks leiame nii HT hinnangute kui ka korduvkaalumisel ja AC-kalibreerimisel saadud hinnangute standardhälbed, suhtelised ruutkeskmised vead ja suhtelised nihked üle kõikide valimikorduste. Suhteline ruutkeskmine viga on

$$RRMSE(\hat{Y}_d) = \frac{\sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{Y}_d^{(t)} - Y_d)^2}}{Y_d}$$

ja suhteline nihe on

$$RB(\hat{Y}_d) = \frac{\frac{1}{T}\sum_{t=1}^{T}(\hat{Y}_d^{(t)} - Y_d)}{Y_d},$$

kus T on valimikorduste arv ja $\hat{Y}_d^{(t)}$ on uuritava tunnuse t-nda valimikorduse kogusumma hinnang osakogumis d ja Y_d vastava osakogumi tegelik kogusumma. Statistikute kujust näeme, et nad mõõdavad hinnangute erinevusi tegelikest kogusummadest, võttes arvesse ka vastavat kogusumma suurust. Hinnangu headuse korral on mõlemad suurused nullilähedased, kui kõik hinnangud üle valimikorduste on täpsed, on nii $RRMSE(\hat{Y}_d)$ kui ka $RB(\hat{Y}_d)$ väärtuseks 0. Kui suhteline ruutkeskmine viga on positiivsete väärtustega suurus, siis suhteline nihe võib olla ka negatiivne. Sellisel juhul hindame me kogusummat keskmiselt tegelikust väiksemaks, vastasel juhul aga suuremaks.

3.4 Simulatsioonide tulemused

Järgnevalt on välja toodud simulatsioonide tulemused. Tabelites 3.5 - 3.8 on toodud täpsusnäitajad üle valimi võtmiste korduste. Standardhälbed on toodud Lisas 1. Mõlemas tabelis on toodud standardhälbed kahe huvipakkuva disaini ja eelkirjeldatud kahe juhu jaoks ühe uuritava tunnuse kohta. Osakogumite eristamise ja parema ülevaate saamise eesmärgil on tabelites ära toodud varem defineeritud osakogumi number ja osakogumi maht.

Simulatsiooni käigus saime osakogumitele kooskõlalised hinnangud mõlema disaini ja nii täpselt teadaolevate kui ka hinnatud üldkogumi kogusummade korral. Üldkogumi kogusumma hinnangute standardhälbed täpse $\mathbf{Y}_0$ korral on mõlema meetodi ja disaini korral nullid, standardhälbed hinnatud $\mathbf{Y}_0$ korral on aga AC ja RW meetodi puhul täpselt sama suured.

Tabelis 3.5 on toodud netosissetuleku hinnangute täpsusnäitajad Poissoni valiku korral nii hinnatud kui ka täpse $\mathbf{Y}_0$ korral. Näeme, et enam kui poolte osakogumite korral on RW ja AC-kalibreeritud hinnangute absoluutne suhteline nihe suurem kui HT hinnangu korral, mis on aga oodatav, sest AC ja RW hinnangud on asümptootiliselt nihketa.. Kui võrrelda AC-kalibreeritud ja RW hinnangute suhtelisi nihkeid omavahel, siis näeme, et need on peaaegu sama suured. Osakogumites 8-12 on küll mõlemal juhul RW hinnangud väiksema absoluutse suhtelise nihkega, kuid vahed on küllaltki väikesed, kusjuures kõik nihked nendes osakogumites on negatiivsed ehk me hindame kogusummat tegelikust väiksemaks.

Tabel 3.5: Netosissetuleku hinnangute täpsusnäitajad Poissoni disaini korral

d	N_d	RRMSE					RB				
			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
		HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	,188	,186	,186	,188	,188	,010	,012	,012	,012	,012
2	4 094	,074	,063	,063	,071	,073	,004	,005	,006	,006	,006
3	1 840	,122	,104	,105	,114	,121	-,001	-,003	-,003	-,002	-,002
4	364	,276	,276	,276	,278	,278	-,004	-,003	-,002	-,002	-,002
5	2 393	,111	,103	,104	,106	,106	-,001	-,001	-,001	-,001	-,001
6	2 015	,133	,119	,118	,124	,125	-,001	-,003	-,003	-,003	-,002
7	923	,186	,180	,178	,180	,178	-,001	,002	,001	,003	,002
8	1 172	,186	,170	,168	,173	,171	-,006	-,011	-,013	-,011	-,013
9	370	,346	,330	,330	,334	,334	,000	-,006	-,007	-,005	-,006
10	1 559	,154	,147	,144	,147	,144	-,004	-,001	-,003	-,001	-,003
11	1 478	,157	,151	,150	,153	,151	-,015	-,013	-,014	-,012	-,014
12	620	,264	,254	,255	,256	,256	,001	-,001	-,002	-,001	-,002
ÜK	17 540	,043	,000	,000	,030	,030	,000	,000	,000	,000	,000

Kui nüüd aga vaadata RRMSE-sid, siis need on vastupidiselt suhtelistest nihetest viimastes osakogumites väiksemad just AC-kalibreeritud hinnangute korral. Sama muster on ka standardhälvete korral. Järelikult on AC-kalibreeritud hinnangud viimastes osakogumites küll väiksema hajuvusega, aga erinevad tegelikest osakogumi summadest rohkem kui RW hinnangud.

Netotulu hinnangute täpsusnäitajad HG disaini korral on Tabelis 3.6. Sarnaselt Poissoni disainiga on AC ja RW hinnangute absoluutsed suhtelised nihked mitme osakogumi korral suuremad vastavatest suurustest HT hinnangu korral. Võrreldes aga AC ja RW hinnangute

Tabel 3.6: Netosissetuleku hinnangute täpsusnäitajad HG disaini korral

d	N_d	RRMSE					RB				
			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
		HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	,186	,186	,186	,186	,187	-,007	-,006	-,006	-,007	-,007
2	4 094	,076	,067	,067	,071	,071	,003	,004	,004	,003	,003
3	1 840	,121	,110	,110	,115	,119	,003	-,001	-,001	-,002	-,002
4	364	,294	,293	,294	,292	,293	,004	,007	,007	,006	,006
5	2 393	,113	,112	,112	,114	,114	,000	,000	,001	,000	,001
6	2 015	,133	,120	,119	,122	,122	,000	-,002	-,002	-,003	-,004
7	923	,185	,180	,177	,180	,177	,001	,003	,003	,003	,003
8	1 172	,182	,174	,174	,176	,175	-,002	-,007	-,008	-,007	-,008
9	370	,342	,336	,337	,341	,341	-,015	-,016	-,016	-,016	-,016
10	1 559	,153	,149	,147	,149	,147	-,002	,000	,000	,000	,000
11	1 478	,161	,157	,156	,158	,156	-,004	-,002	-,003	-,003	-,003
12	620	,275	,267	,266	,269	,267	,002	,001	,000	,001	,001
ÜK	17 540	,032	,000	,000	,022	,022	,001	,000	,000	-,001	-,001

suhtelisi nihkeid omavahel, siis sellist mustrit nagu Poissoni disaini korral ei esine. RB-d on mõlema meetodi korral sama suured, esinevad vaid väga väikesed erinevused. Küll on aga sarnaselt Poissoni disainile viimastes osakogumites AC-kalibreeritud hinnangul väiksemad RRMSE-d ja standardhälbed. Seega võime väita, et AC-kalibreerimine HG disaini korral annab osakogumites 8-12 netosissetuleku kogusummale paremaid hinnanguid.

Tarbimiskulu hinnangute täpsusnäitajad Poissoni disaini korral on antud Tabelis 3.7. Näeme, et jällegi on selliseid osakogumeid, mille korral on AC-kalibreerimisel ja korduvkaalumisel leitud hinnangute RB suurem, ja samuti selliseid, mille korral on HT hinnangud suurema suhtelise nihkega. Ei saa väita, et kumbki meetod annaks võrreldes teisega väiksema suhtelise nihkega hinnanguid – kuigi RB väärtused ei ole küll mõlema juhu ja iga osakogumi korral täpselt samad, on väikeseid kõrvalekaldeid mõlema meetodi kasuks. Hinnatud kogusummade korral on RB-d peaaegu sama suured, kui teadaolevate üldkogumi summade korral. Näeme, et standardhälbed ja suhtelised ruutkeskmised vead jälgivad sama mustrit kui netopalga hinnangute (Poissoni valik) korral.

Tabel 3.7: Tarbimiskulu hinnangute täpsusnäitajad Poissoni disaini korral

d	N_d	RRMSE					RB				
			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
		HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	,195	,193	,194	,196	,196	,013	,015	,015	,015	,015
2	4 094	,079	,065	,065	,074	,075	,004	,005	,005	,005	,005
3	1 840	,120	,106	,106	,115	,121	-,003	-,003	-,003	-,003	-,003
4	364	,334	,334	,334	,341	,341	,012	,013	,013	,013	,013
5	2 393	,115	,107	,107	,110	,110	,001	,002	,003	,002	,002
6	2 015	,135	,121	,120	,126	,128	,000	-,003	-,003	-,003	-,002
7	923	,206	,201	,199	,201	,200	-,004	-,001	-,002	-,001	-,002
8	1 172	,248	,184	,183	,190	,189	,000	-,025	-,026	-,023	-,024
9	370	,316	,312	,312	,314	,314	,000	,001	-,001	,001	,000
10	1 559	,175	,168	,166	,169	,166	,000	,003	,001	,003	,001
11	1 478	,170	,163	,163	,165	,164	-,017	-,015	-,016	-,015	-,016
12	620	,283	,274	,275	,276	,275	,010	,008	,007	,006	,005
ÜK	17 540	,044	,000	,000	,030	,030	,001	,000	,000	,000	,000

Hüpergeomeetrilise valiku korral (Tabel 3.8) käituvad täpsusnäitajad väga sarnaselt eelmistele juhtudele. Paneme tähele, et AC ja RW hinnangute RB on hinnatud $\mathbf{Y}_0$ korral peaaegu samad, erinevus esineb vaid kahes osakogumis.

Tabel 3.8: Tarbimiskulu hinnangute täpsusnäitajad HG disaini korral

d	N_d	RRMSE					RB				
			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
		HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	,191	,191	,191	,191	,191	-,009	-,006	-,006	-,007	-,007
2	4 094	,079	,069	,069	,073	,074	,002	,003	,003	,002	,002
3	1 840	,118	,109	,109	,115	,118	,000	,000	,001	-,001	-,001
4	364	,337	,339	,340	,339	,340	,002	,005	,005	,005	,005
5	2 393	,114	,112	,112	,114	,114	-,001	,001	,002	,000	,001
6	2 015	,131	,121	,120	,125	,125	-,001	-,002	-,001	-,002	-,002
7	923	,209	,203	,201	,203	,200	-,002	,001	,000	,000	,000
8	1 172	,246	,189	,188	,202	,201	,008	-,016	-,018	-,015	-,016
9	370	,312	,311	,311	,311	,311	-,014	-,012	-,012	-,013	-,013
10	1 559	,172	,168	,166	,168	,166	,001	,004	,003	,003	,003
11	1 478	,178	,173	,171	,174	,172	-,006	-,004	-,005	-,005	-,005
12	620	,289	,284	,283	,289	,288	,010	,009	,008	,008	,008
ÜK	17 540	,034	,000	,000	,023	,023	,000	,000	,000	,001	,001

Kui võrrelda kahe disaini netosissetuleku ja tarbimiskulu hinnanguid omavahel, siis me ei saa väita, et kumbki disain oleks teisest parem. Mõlema uuritava tunnuse korral on muster väga sarnane. Suhtelised nihked on hüpergeomeetrilise valiku korral üldiselt väiksemad. Näeme, et kui $\mathbf{Y}_0$ on täpne, siis Poissoni valiku korral on korduvkaalumise ja AC-kalibreerimise hinnangud väiksemate standardhälvete ja suhteliste ruutkeskmiste vigadega. Märgime siinkohal veel ära, et juhul, kus $\mathbf{Y}_0$ on hinnatud varasemast uuringust, on mõlema disaini korral on nii RRMSE-d kui ka standardhälbed üldjuhul suuremad, kuid erinevused ei ole väga suured.

3.5 Järeldused

Simulatsioonülesande käigus testisime varem tuletatud HG ja Poissoni disaini valemeid juhtudel, millest esimese korral eeldasime, et uuritavate tunnuste kogusummad on teada ja teise korral hindasime eelnevalt kogusummad sõltumatu valimi pealt. Mõlemal juhul saavutasime AC ja RW meetoditega kooskõla osakogumite summade ja varem teadaoleva kogusummadega.

Üldiselt olid AC ja RW meetoditega leitud osakogumite hinnangud täpsemad kui HT hinnangud. HG disaini puhul olid hinnangud väiksema nihkega kui Poissoni disaini puhul.

Väiksemate valmimahtude juures (osakogumites 7-12) olid AC-kalibreeritud hinnangud täpsemad kui RW hinnangud, suuremate valimimahtude (osakogumites 1-6) puhul vastupidi. See võib olla tingitud asjaolust, et osakogumid 7-12 koosnevad leibkondadest, kus leibkonnapea on töötu ning siirdetulu moodustab keskmiselt suurema osa leibkonna kogu sissetulekust. Siirdetulu kasutati simulatsiooniülesandes A-informatsioonina ning AC-kalibreerimise puhul on kaalud kalibreeritud ka siirdetulu infot kasutades. RW meetodi puhul kasutati siirdetulu esimeses kaalude leidmise etapis ning lõplikud kaalud ei saavuta A-informatsiooniga kooskõla. Sama loogika kehtib ka leibkonna tarbimiskulude kohta.

Antud tulemuste valguses pakuks kindlasti huvi leitud hinnangute käitumine juhul, kui taustatunnus on uuritavate tunnustega tugevas lineaarses seoses, ja nende võrdlemine käesolevate tulemustega, kuid see väljub juba mahult praeguse töö raamidest.

Summary

Consistent estimation of domains for PPS hypergeometric and Poisson sampling designs.

Bachelor's thesis

Kristjan Kokorev

Domain estimation is an important area in survey sampling and domain estimates are widely used by statistics' users. Nowadays several studies are carried out simultaneously on the same population, and some variables are common in those studies. It is natural to demand that the estimations from different studies are consistent.

We want to estimate domain totals for some variables of interest. We assume that some of these variables were included in a survey done earlier or simultaneously, and we know the estimate of population total for these variables from this survey or that the exact totals are known (for example from registers). Our problem is that the estimates of our domains are most likely inconsistent with information from previous surveys (or registers). We can achieve consistency with specially developed calibration methods such as AC-calibration and repeated weighting (RW). Both of these methods are based on the theory of calibration developed by Deville and Särndal (1992).

In this thesis we give a short overview of the AC-calibration estimator and repeated weighting method. Also, we give a brief overview of probability proportional to size sampling design and of Poisson and hypergeometric (HG) sampling designs which under some assumptions are special cases of probabilities proportional to size (PPS) sampling. We derive formulas of AC-calibration and RW weights for both of these designs.

We tested our formulas in a simulation study. Simulations were done on an artificial population composed on real data from Estonian Household Survey. The results of our simulations show that consistency with known total estimates can be achieved with AC and RW estimators, even if the total was estimated from a separate and independent survey. The two estimators perform better than HT estimators. AC estimators performed better than RW

within domains with smaller sample size, but that might be due to the specifics of the used auxiliary information, and the fact that AC-calibration method gives us weights that are consistent with auxiliary information as well as the known totals, but RW weights are only consistent with the latter information.

Kasutatud kirjandus

1. Estevao, V.M., Särndal, C.-E., 2004. Borrowing Strength Is Not the Best Technique Within a Wide Class of Design-Consistent Domain Estimators. *Journal of Official Statistics*, 20, 645-669.
2. Deville, J.-C., Särndal, C.-E., 1992. Calibration Estimates in Survey Sampling. *Journal of the American Statistical Association*, 87, 376-382.
3. Särndal, C.-E., Traat, I., 2011. Domain estimators calibrated on information from another Survey. *Research report* No. 2009-1, vol. 15.
4. Traat, I., Inno, J., 1997. *Tõenäosuslik valikuuring*. Tartu: TÜ Kirjastus.
5. Lohr, S.L., 1999. *Sampling: Design and Analysis*. Pacific Grove, CA: Duxbury Press, Brooks/Cole Publishing.
6. Särndal, C.-E., Swensson, B., Wretman, J., 1992. *Model Assisted Survey Sampling.* New-York: Springer-Verlag
7. Grafström, A., 2010. *On Unequal Probability Sampling Designs.* (PhD dissertatsioon) Umeå.
8. Traat, I., Ilves, M., 2007. The Hypergeometric Sampling Design, Theory and Practice. *Acta Applicandae Mathematicae*, 97(1-3), 311-321.
9. Särndal, C.-E., 2007. The calibration approach in survey theory and practice. *Survey Methodology*, 33, 99-119.
10. Estevao, V.M., Särndal, C.-E., 2000. A Functional Form Approach to Calibration. *Journal of Official Statistics*, 16, 379-399.
11. Kroese, A.H., Renssen, R.H, 1999. Weighting and Imputation at Statistics Netherlands. *Proceedings if IASS Satellite Conference on Small Area Estimation*, Riga: Latvia, 109-120.
12. Houbiers, M., 2004. Towards a Social Statistical Database and Unified Estimates at Statistics Netherlands. *Journal of Official Statistics*, 20, 55-75.
13. Knottnerus, P., van Duin, C., 2006. Variances in Repeated Weighting with an Application to the Dutch Labour Force Survey. *Journal of Official Statistics*, 22, 565-584.
14. Kollo, T., von Rosen, D., 2005. Advanced Multivariate Statistics with Matrices, Dordrecht: Springer.

Lisa 1 Hinnangute standardhälbed osakogumites

Tabel 1: Netosissetuleku hinnangute standardhälbed osakogumites ja üldkogumis $(\times 10^3)$

d	N_d	Y_d	Poissoni valik					HG valik				
				Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
			HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	6 343	1 193	1 177	1 178	1 191	1 193	1 181	1 181	1 181	1 183	1 183
2	4 094	41 336	3 060	2 577	2 588	2 945	3 012	3 120	2 755	2 749	2 922	2 949
3	1 840	22 510	2 738	2 346	2 355	2 570	2 717	2 728	2 474	2 482	2 601	2 681
4	364	2 153	595	595	595	599	600	632	631	633	630	630
5	2 393	16 961	1 885	1 755	1 763	1 800	1 799	1 916	1 907	1 907	1 934	1 935
6	2 015	19 277	2 568	2 296	2 284	2 388	2 401	2 556	2 316	2 300	2 345	2 342
7	923	3 964	739	713	705	714	706	733	713	702	713	703
8	1 172	5 678	1 058	962	952	983	967	1 036	990	985	999	994
9	370	2 097	726	693	692	702	701	716	705	706	715	716
10	1 559	5 323	821	782	767	783	769	817	792	782	791	784
11	1 478	5 545	868	836	830	847	832	892	870	863	874	864
12	620	2 655	700	675	678	679	680	732	709	706	715	710
ÜK	17 540	133 841	5 800	0	0	3 986	3 986	4 230	0	0	2 901	2 901

Tabel 2: Tarbimiskulude hinnangute standardhälbed osakogumites ja üldkogumis $(\times 10^3)$

d	N_d	Y_d	Poissoni valik					HG valik				
				Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$			Täpne $\mathbf{Y}_0$		Hinnatud $\mathbf{Y}_0$	
d	maht	Summa	HT	RW	AC	RW	AC	HT	RW	AC	RW	AC
1	712	5 050	984	974	975	987	988	964	965	964	963	962
2	4 094	34 535	2 715	2 253	2 255	2 535	2 584	2 725	2 395	2 387	2 528	2 541
3	1 840	19 132	2 305	2 026	2 035	2 200	2 306	2 253	2 078	2 083	2 204	2 257
4	364	1 832	612	611	611	624	625	618	622	622	622	622
5	2 393	14 585	1 675	1 563	1 566	1 598	1 598	1 658	1 640	1 641	1 658	1 659
6	2 015	16 345	2 211	1 975	1 965	2 065	2 085	2 143	1 976	1 970	2 041	2 044
7	923	3 303	682	663	659	663	660	690	671	663	670	662
8	1 172	5 409	1 343	988	981	1 021	1 013	1 329	1 017	1 012	1 092	1 083
9	370	1 820	575	568	567	572	571	567	565	565	565	565
10	1 559	4 467	783	752	742	755	744	768	751	742	750	744
11	1 478	5 384	912	876	874	887	879	956	930	923	937	927
12	620	2 579	731	708	709	711	710	745	731	731	745	743
ÜK	17 540	114 440	5 030	0	0	3 450	3 450	3 882	0	0	2 634	2 634

Lisa 2 Programmi kood Poissoni disaini korral

```
#Paigaldame vajaliku paketi
#install.packages("sampling")
library(sampling)

#Loeme andmestiku sisse
andmed = read.csv("C:\\Users\\Kristjan\\Desktop\\yldkogum1.csv", header = T ,  sep = ",")
attach(andmed)
rep=1000    #Valimi võtmiste arv
N=nrow(andmed)    #Üldkogumi maht

#Lisame andmestikku tunnuse, mis on konstantselt 1
uhed=rep(1, N)
andmed$uhed=uhed

V=sum(lbk_suurus)    #Taustatunnuse leibkonna suurus üldkogumi summa
Y1=sum(netopalk)    #Uuritava tunnuse kogupalk tegelik (üldkogumi) summa
Y2=sum(Kulud2)    #Uuritava tunnuse tarbimiskulud tegelik  summa
X1=sum(siirdetulu)    #Abitunnuse siirdetulu tegelik summa
X2=sum(lapsi)     #Abitunnuse laste arv tegelik summa

#Leiame osakogumeid iseloomustavad suurused üldkogumis
Ud=data.frame(osakogum_nr=1:12)
Ud$netopalk=as.vector(by(netopalk, Osakogum, sum))
Ud$maht=as.vector(by(uhed, Osakogum, sum))
Ud$keskmine_np=Ud$netopalk/Ud$maht
Ud$minimaalne_np=as.vector(by(netopalk, Osakogum, min))
Ud$maksimaalne_np=as.vector(by(netopalk, Osakogum, max))
Ud$std_np=as.vector(by(netopalk, Osakogum, sd))
Ud$kulud=as.vector(by(Kulud2, Osakogum, sum))
Ud$keskmine_tk=Ud$kulud/Ud$maht
Ud$minimaalne_tk=as.vector(by(Kulud2, Osakogum, min))
Ud$maksimaalne_tk=as.vector(by(Kulud2, Osakogum, max))
Ud$std_tk=as.vector(by(Kulud2, Osakogum, sd))

#Defineerime Poissoni valiku, kus taustatunnuseks on leibkonna suurus ja parameetriks oodatav
#valimimaht
Poisson=function(n){
v_i=lbk_suurus
Pii=(n*v_i)/V     # Kaasamistõenäosused
s=UPpoisson(Pii)    # Leiame valimise sattunud elemendid
valim=(1:length(Pii))[s==1]    #Valim
return (valim)
}

#Defineerime kalibreerimise lisainformatsiooniga antud ülesande raames
kalibreerimine=function(valim, n){
abi=andmed[valim,]    #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused
ns=nrow(abi)    #Realiseerunud valimi maht
x1=abi$siirdetulu
x2=abi$lapsi

#Leiame abitunnuste hinnangud
X1_hinnang=sum((V/(n*abi$lbk_suurus))*(x1))
X2_hinnang=sum((V/(n*abi$lbk_suurus))*(x2))
```

```
#Teeme vektori (X-X_hinnang)
vec=c(X1-X1_hinnang, X2-X2_hinnang)

w=rep(0,ns) #Vektor, millese leiame kaalud
M=matrix(c(0,0,0,0),2)    #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:ns){
x_vec=c(x1[i],x2[i])
m_i=(V/(n*(abi$lbk_suurus)[i]))*(x_vec%*%t(x_vec))
M=M+m_i
}

M_poord=solve(M)    #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord    #Vektor lambda

#Leiame A-informatsiooniga kalibreeritud kaalud
for (i in 1:ns){
x_vec=c(x1[i],x2[i])
w[i]=(V/(n*abi$lbk_suurus[i]))*(1+lambda%*%x_vec)
}

#Tagastame saadud kaalud
return (w)
}

#Defineerime kalibreerimise C-informatsiooniga, kus parameetritena anname Poissoni valiku valimi,
#A-informatsiooniga kalibreeritud kaalud ja oodatava valimimahu ning kas uuritavate tunnuste
#tegelikud summad või hinnangud

korduvkaalumine=function(valim, w, n, Y1, Y2){
abi=andmed[valim,]    #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused
ns=nrow(abi)    #Realiseerunud valimi maht

#Eraldame C-informatsiooni vektorid
y1=abi$netopalk
y2=abi$Kulud2

#Leiame uuritavate tunnuste kalibreeritud hinnangud
Y1_CALhinnang=sum(w*y1)
Y2_CALhinnang=sum(w*y2)

#Teeme vektori (Y - Y-i kalibreeritud hinnang)
vec=c(Y1-Y1_CALhinnang, Y2-Y2_CALhinnang)

w_RW=rep(1,ns)    #Vektor, millese leiame kaalud
M=matrix(c(0,0,0,0),2)    #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:ns){
y_vec=c(y1[i],y2[i])
m_i= w[i]*(y_vec%*%t(y_vec))
M=M+m_i
}

M_poord=solve(M)    #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord    #Vektor lambda
```

```
#Leiame C-informatsiooniga kalibreeritud kaalud
for (i in 1:ns){
y_vec=c(y1[i],y2[i])
w_RW[i]=(w[i])*(1+lambda%*%y_vec)
}

#Tagstame leitud kaalud
return (w_RW)
}

#Defineerime funktsiooni, mis leiab ette antud Poissoni valiku valimile vastavad AC-kalibreeritud
#kaalud. Parameetriteks valim, oodatav valimimaht ja kas uuritavate tunnuste tegelikud summad või
#hinnangud

ACkalibreerimine=function(valim, n, Y1, Y2){
abi=andmed[valim,]    #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused
ns=nrow(abi)    #Realiseerunud valimi maht

#Eraldame uuritavate ja abitunnuste vektorid
x1=abi$siirdetulu
x2=abi$lapsi
y1=abi$netopalk
y2=abi$Kulud2

#Leiame abitunnuste hinnangud
X1_hinnang=sum((V/(n*abi$lbk_suurus))*(x1))
X2_hinnang=sum((V/(n*abi$lbk_suurus))*(x2))
Y1_hinnang=sum((V/(n*abi$lbk_suurus))*(y1))
Y2_hinnang=sum((V/(n*abi$lbk_suurus))*(y2))

#Teeme vektori ((X-X_hinnang)' (Y-Y_hinnang)')'
vec=c(X1-X1_hinnang, X2-X2_hinnang,Y1-Y1_hinnang,Y2-Y2_hinnang)

w_AC=rep(0,ns)    #Vektor, millese leiame kaalud
M=matrix(rep(0,16),4)    #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:ns){
xy_vec=c(x1[i],x2[i],y1[i],y2[i])
m_i=(V/(n*(abi$lbk_suurus)[i]))*(xy_vec%*%t(xy_vec))
M=M+m_i
}

M_poord=solve(M)    #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord    #Vektor lambda

#Leiame AC-kalibreeritud kaalud
for (i in 1:ns){
xy_vec=c(x1[i],x2[i],y1[i],y2[i])
w_AC[i]=(V/(n*abi$lbk_suurus[i]))*(1+lambda%*%xy_vec)
}
#Tagastame AC-kalibreeritud kaalud
return (w_AC)
}

#Defineerime funktsiooni, mis leiab etteantud valimi ja kaalude põhjal tunnuste hinnangud
#osakogumites
hindamine=function(valim, kaalud){
abi=andmed[valim, ]
```

```
#Leiame uuritavate tunnuste valimi vektori
y1=abi$netopalk
y2=abi$Kulud2

#Leiame vektorid (kaal*tunnus)
vec1=kaalud*y1
vec2=kaalud*y2

#Leiame vektori, mis määrab iga valimi elemendile vastava osakogumi
osak=abi$Osakogum

#Leiame kogusumma hinnangud  osakogumites
OK_hinnang1=as.vector(by(vec1,osak,sum))
OK_hinnang2=as.vector(by(vec2,osak,sum))
OK_hinnang3=as.vector(by(kaalud,osak,sum))

#Moodustame neist tabeli
OK_hinnangud=data.frame(osakogum=1:12)
OK_hinnangud$y1=OK_hinnang1
OK_hinnangud$y2=OK_hinnang2
OK_hinnangud$Nd=OK_hinnang3

#Tagastame viimase tabeli
return (OK_hinnangud)
}

#Teeme tühjad maatriksid, kuhu hakkame
#osakogumite hinnanguid koguma (juhul, kus üldkogumi summa on teada, J1 tähistab juhtu 1)
J1HT_osakY1=matrix(nrow=12, ncol=rep)
J1RW_osakY1=matrix(nrow=12, ncol=rep)
J1AC_osakY1=matrix(nrow=12, ncol=rep)
J1HT_osakY2=matrix(nrow=12, ncol=rep)
J1RW_osakY2=matrix(nrow=12, ncol=rep)
J1AC_osakY2=matrix(nrow=12, ncol=rep)
J1HT_osakN=matrix(nrow=12, ncol=rep)
J1RW_osakN=matrix(nrow=12, ncol=rep)
J1AC_osakN=matrix(nrow=12, ncol=rep)

#Abistavad suurused RRMSE-de ja RB-de leidmiseks (U summa teada)
J1RRMSEHT_abi1=rep(0,12)
J1RRMSEHT_abi2=rep(0,12)
J1RRMSEHT_abi3=rep(0,12)
J1RRMSERW_abi1=rep(0,12)
J1RRMSERW_abi2=rep(0,12)
J1RRMSERW_abi3=rep(0,12)
J1RRMSEAC_abi1=rep(0,12)
J1RRMSEAC_abi2=rep(0,12)
J1RRMSEAC_abi3=rep(0,12)

J1RBHT_abi1=rep(0,12)
J1RBHT_abi2=rep(0,12)
J1RBHT_abi3=rep(0,12)
J1RBRW_abi1=rep(0,12)
J1RBRW_abi2=rep(0,12)
J1RBRW_abi3=rep(0,12)
J1RBAC_abi1=rep(0,12)
J1RBAC_abi2=rep(0,12)
J1RBAC_abi3=rep(0,12)
```

```
#Teeme tühjad maatriksid, kuhu hakkame
#osakogumite hinnanguid koguma (juhul, kus üldkogumi summa on hinnatud, J2 tähistab juhtu 2)
J2HT_osakY1=matrix(nrow=12, ncol=rep)
J2RW_osakY1=matrix(nrow=12, ncol=rep)
J2AC_osakY1=matrix(nrow=12, ncol=rep)
J2HT_osakY2=matrix(nrow=12, ncol=rep)
J2RW_osakY2=matrix(nrow=12, ncol=rep)
J2AC_osakY2=matrix(nrow=12, ncol=rep)
J2HT_osakN=matrix(nrow=12, ncol=rep)
J2RW_osakN=matrix(nrow=12, ncol=rep)
J2AC_osakN=matrix(nrow=12, ncol=rep)

#Abistavad suurused RRMSE-de ja RB-de leidmiseks (U summa teada)
J2RRMSEHT_abi1=rep(0,12)
J2RRMSEHT_abi2=rep(0,12)
J2RRMSEHT_abi3=rep(0,12)
J2RRMSERW_abi1=rep(0,12)
J2RRMSERW_abi2=rep(0,12)
J2RRMSERW_abi3=rep(0,12)
J2RRMSEAC_abi1=rep(0,12)
J2RRMSEAC_abi2=rep(0,12)
J2RRMSEAC_abi3=rep(0,12)

J2RBHT_abi1=rep(0,12)
J2RBHT_abi2=rep(0,12)
J2RBHT_abi3=rep(0,12)
J2RBRW_abi1=rep(0,12)
J2RBRW_abi2=rep(0,12)
J2RBRW_abi3=rep(0,12)
J2RBAC_abi1=rep(0,12)
J2RBAC_abi2=rep(0,12)
J2RBAC_abi3=rep(0,12)

#Teeme vektorid, kuhu kogume IU kogusummade hinnangud
IU_hY1=rep(1, rep)
IU_hY2=rep(1, rep)
IU_hN=rep(1,rep)

#Teeme maatriksid kuhu kogume osakogumitesse sattunud vaatluste arve antud valimi korral
NdIU=matrix(nrow=12,ncol=rep)
NdIIU=matrix(nrow=12,ncol=rep)

for (i in 1:rep){
n=1000

#Võtame nii IU kui ka IIU valimid
valimIU=Poisson(2000)
valimIIU=Poisson(n)

abiIU=andmed[valimIU,]
abiIIU=andmed[valimIIU,]

#Leiame osakogumitesse sattunud  objektide arvud ja lisame maatriksitesse
vaatlusiIU=as.vector(by(abiIU$uhed,abiIU$Osak,sum))
vaatlusiIIU=as.vector(by(abiIIU$uhed,abiIIU$Osak,sum))
NdIU[,i]=vaatlusiIU
NdIIU[,i]=vaatlusiIIU
```

```
#Leiame HT kaalud ja IU hinnangud
w_IU=V/(2000*abiIU$lbk_suurus)
w_IIU=V/(n*abiIIU$lbk_suurus)

Y1_hinnangIU=sum(w_IU*abiIU$netopalk)
Y2_hinnangIU=sum(w_IU*abiIU$Kulud2)
IU_hY1[i]=Y1_hinnangIU
IU_hY2[i]=Y2_hinnangIU
IU_hN[i]=sum(w_IU)

w_C=kalibreerimine(valimIIU,n)

#Leiame AC ja RW kaalud nii teada  oleva üldkogumi kui
#ka hinnatud suuruse korral
J1w_RW=korduvkaalumine(valimIIU,w_C,n,Y1,Y2)
J1w_AC=ACkalibreerimine(valimIIU,n,Y1,Y2)
J2w_RW=korduvkaalumine(valimIIU,w_C,n,Y1_hinnangIU,Y2_hinnangIU)
J2w_AC=ACkalibreerimine(valimIIU,n,Y1_hinnangIU,Y2_hinnangIU)

#Leiame HT, RW ja AC hinnangud osakogumites esimesel juhul (teada  olevad summad)
HT_hinnangud1=hindamine(valimIIU,w_IIU)
RW_hinnangud1=hindamine(valimIIU,J1w_RW)
AC_hinnangud1=hindamine(valimIIU,J1w_AC)

#Leiame HT, RW ja AC hinnangud osakogumites teisel juhul (hinnatud summad)
HT_hinnangud2=hindamine(valimIIU,w_IIU)
RW_hinnangud2=hindamine(valimIIU,J2w_RW)
AC_hinnangud2=hindamine(valimIIU,J2w_AC)

#Täidame maatriksid vastavate hinnangutega
J1HT_osakY1[,i]=HT_hinnangud1[,2]
J1RW_osakY1[,i]=RW_hinnangud1[,2]
J1AC_osakY1[,i]=AC_hinnangud1[,2]
J1HT_osakY2[,i]=HT_hinnangud1[,3]
J1RW_osakY2[,i]=RW_hinnangud1[,3]
J1AC_osakY2[,i]=AC_hinnangud1[,3]
J1HT_osakN[,i]=HT_hinnangud1[,4]
J1RW_osakN[,i]=RW_hinnangud1[,4]
J1AC_osakN[,i]=AC_hinnangud1[,4]

J2HT_osakY1[,i]=HT_hinnangud2[,2]
J2RW_osakY1[,i]=RW_hinnangud2[,2]
J2AC_osakY1[,i]=AC_hinnangud2[,2]
J2HT_osakY2[,i]=HT_hinnangud2[,3]
J2RW_osakY2[,i]=RW_hinnangud2[,3]
J2AC_osakY2[,i]=AC_hinnangud2[,3]
J2HT_osakN[,i]=HT_hinnangud2[,4]
J2RW_osakN[,i]=RW_hinnangud2[,4]
J2AC_osakN[,i]=AC_hinnangud2[,4]

#Summerime
J1RRMSEHT_abi1=J1RRMSEHT_abi1+(HT_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSEHT_abi2=J1RRMSEHT_abi2+(HT_hinnangud1[,3]-Ud$kulud)**2
J1RRMSEHT_abi3=J1RRMSEHT_abi3+(HT_hinnangud1[,4]-Ud$maht)**2
J1RRMSERW_abi1=J1RRMSERW_abi1+(RW_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSERW_abi2=J1RRMSERW_abi2+(RW_hinnangud1[,3]-Ud$kulud)**2
J1RRMSERW_abi3=J1RRMSERW_abi3+(RW_hinnangud1[,4]-Ud$maht)**2
J1RRMSEAC_abi1=J1RRMSEAC_abi1+(AC_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSEAC_abi2=J1RRMSEAC_abi2+(AC_hinnangud1[,3]-Ud$kulud)**2
```

```
J1RRMSEAC_abi3=J1RRMSEAC_abi3+(AC_hinnangud1[,4]-Ud$maht)**2

J1RBHT_abi1=J1RBHT_abi1+(HT_hinnangud1[,2]-Ud$netopalk)
J1RBHT_abi2=J1RBHT_abi2+(HT_hinnangud1[,3]-Ud$kulud)
J1RBHT_abi3=J1RBHT_abi3+(HT_hinnangud1[,4]-Ud$maht)
J1RBRW_abi1=J1RBRW_abi1+(RW_hinnangud1[,2]-Ud$netopalk)
J1RBRW_abi2=J1RBRW_abi2+(RW_hinnangud1[,3]-Ud$kulud)
J1RBRW_abi3=J1RBRW_abi3+(RW_hinnangud1[,4]-Ud$maht)
J1RBAC_abi1=J1RBAC_abi1+(AC_hinnangud1[,2]-Ud$netopalk)
J1RBAC_abi2=J1RBAC_abi2+(AC_hinnangud1[,3]-Ud$kulud)
J1RBAC_abi3=J1RBAC_abi3+(AC_hinnangud1[,4]-Ud$maht)

J2RRMSEHT_abi1=J2RRMSEHT_abi1+(HT_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSEHT_abi2=J2RRMSEHT_abi2+(HT_hinnangud2[,3]-Ud$kulud)**2
J2RRMSEHT_abi3=J2RRMSEHT_abi3+(HT_hinnangud2[,4]-Ud$maht)**2
J2RRMSERW_abi1=J2RRMSERW_abi1+(RW_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSERW_abi2=J2RRMSERW_abi2+(RW_hinnangud2[,3]-Ud$kulud)**2
J2RRMSERW_abi3=J2RRMSERW_abi3+(RW_hinnangud2[,4]-Ud$maht)**2
J2RRMSEAC_abi1=J2RRMSEAC_abi1+(AC_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSEAC_abi2=J2RRMSEAC_abi2+(AC_hinnangud2[,3]-Ud$kulud)**2
J2RRMSEAC_abi3=J2RRMSEAC_abi3+(AC_hinnangud2[,4]-Ud$maht)**2

J2RBHT_abi1=J2RBHT_abi1+(HT_hinnangud2[,2]-Ud$netopalk)
J2RBHT_abi2=J2RBHT_abi2+(HT_hinnangud2[,3]-Ud$kulud)
J2RBHT_abi3=J2RBHT_abi3+(HT_hinnangud2[,4]-Ud$maht)
J2RBRW_abi1=J2RBRW_abi1+(RW_hinnangud2[,2]-Ud$netopalk)
J2RBRW_abi2=J2RBRW_abi2+(RW_hinnangud2[,3]-Ud$kulud)
J2RBRW_abi3=J2RBRW_abi3+(RW_hinnangud2[,4]-Ud$maht)
J2RBAC_abi1=J2RBAC_abi1+(AC_hinnangud2[,2]-Ud$netopalk)
J2RBAC_abi2=J2RBAC_abi2+(AC_hinnangud2[,3]-Ud$kulud)
J2RBAC_abi3=J2RBAC_abi3+(AC_hinnangud2[,4]-Ud$maht)
}

#Teeme tulemuste maatriksid iga uuritava tunnuse jaoks, et saada paremat ülevaadet
#Tunnused tabelites: keskmine HT hinnang, HT std, keskmine
#RW hinnang, RW std, keskmine AC hinnang, AC std
#Juht1
J1TULEMUS_Y1=matrix(nrow=12,ncol=6)
J1TULEMUS_Y2=matrix(nrow=12,ncol=6)
J1TULEMUS_N=matrix(nrow=12,ncol=6)

#Juht2
J2TULEMUS_Y1=matrix(nrow=12,ncol=6)
J2TULEMUS_Y2=matrix(nrow=12,ncol=6)
J2TULEMUS_N=matrix(nrow=12,ncol=6)

for(i in 1:12){
#Juht 1
J1TULEMUS_Y1[i,1]=mean(J1HT_osakY1[i,])
J1TULEMUS_Y1[i,2]=sd(J1HT_osakY1[i,])
J1TULEMUS_Y1[i,3]=mean(J1RW_osakY1[i,])
J1TULEMUS_Y1[i,4]=sd(J1RW_osakY1[i,])
J1TULEMUS_Y1[i,5]=mean(J1AC_osakY1[i,])
J1TULEMUS_Y1[i,6]=sd(J1AC_osakY1[i,])

J1TULEMUS_Y2[i,1]=mean(J1HT_osakY2[i,])
J1TULEMUS_Y2[i,2]=sd(J1HT_osakY2[i,])
J1TULEMUS_Y2[i,3]=mean(J1RW_osakY2[i,])
J1TULEMUS_Y2[i,4]=sd(J1RW_osakY2[i,])
```

```
J1TULEMUS_Y2[i,5]=mean(J1AC_osakY2[i,])
J1TULEMUS_Y2[i,6]=sd(J1AC_osakY2[i,])

J1TULEMUS_N[i,1]=mean(J1HT_osakN[i,])
J1TULEMUS_N[i,2]=sd(J1HT_osakN[i,])
J1TULEMUS_N[i,3]=mean(J1RW_osakN[i,])
J1TULEMUS_N[i,4]=sd(J1RW_osakN[i,])
J1TULEMUS_N[i,5]=mean(J1AC_osakN[i,])
J1TULEMUS_N[i,6]=sd(J1AC_osakN[i,])

#Juht 2
J2TULEMUS_Y1[i,1]=mean(J2HT_osakY1[i,])
J2TULEMUS_Y1[i,2]=sd(J2HT_osakY1[i,])
J2TULEMUS_Y1[i,3]=mean(J2RW_osakY1[i,])
J2TULEMUS_Y1[i,4]=sd(J2RW_osakY1[i,])
J2TULEMUS_Y1[i,5]=mean(J2AC_osakY1[i,])
J2TULEMUS_Y1[i,6]=sd(J2AC_osakY1[i,])

J2TULEMUS_Y2[i,1]=mean(J2HT_osakY2[i,])
J2TULEMUS_Y2[i,2]=sd(J2HT_osakY2[i,])
J2TULEMUS_Y2[i,3]=mean(J2RW_osakY2[i,])
J2TULEMUS_Y2[i,4]=sd(J2RW_osakY2[i,])
J2TULEMUS_Y2[i,5]=mean(J2AC_osakY2[i,])
J2TULEMUS_Y2[i,6]=sd(J2AC_osakY2[i,])

J2TULEMUS_N[i,1]=mean(J2HT_osakN[i,])
J2TULEMUS_N[i,2]=sd(J2HT_osakN[i,])
J2TULEMUS_N[i,3]=mean(J2RW_osakN[i,])
J2TULEMUS_N[i,4]=sd(J2RW_osakN[i,])
J2TULEMUS_N[i,5]=mean(J2AC_osakN[i,])
J2TULEMUS_N[i,6]=sd(J2AC_osakN[i,])
}

#Teeme maatriksid, kus on kõigi kolme tunnuse kõigi kolme hinnangu RRMSE ja RB
#Veergudes esmalt HT hinnagule vastavad suurused kolme tunnuse korral ja siis RW ja AC
J1RRMSE=matrix(nrow=12, ncol=9)
J1RB=matrix(nrow=12, ncol=9)
J2RRMSE=matrix(nrow=12, ncol=9)
J2RB=matrix(nrow=12, ncol=9)

J1RRMSE[,1]=sqrt(J1RRMSEHT_abi1/rep)/(Ud$netopalk)
J1RRMSE[,2]=sqrt(J1RRMSEHT_abi2/rep)/(Ud$kulud)
J1RRMSE[,3]=sqrt(J1RRMSEHT_abi3/rep)/(Ud$maht)
J1RRMSE[,4]=sqrt(J1RRMSERW_abi1/rep)/(Ud$netopalk)
J1RRMSE[,5]=sqrt(J1RRMSERW_abi2/rep)/(Ud$kulud)
J1RRMSE[,6]=sqrt(J1RRMSERW_abi3/rep)/(Ud$maht)
J1RRMSE[,7]=sqrt(J1RRMSEAC_abi1/rep)/(Ud$netopalk)
J1RRMSE[,8]=sqrt(J1RRMSEAC_abi2/rep)/(Ud$kulud)
J1RRMSE[,9]=sqrt(J1RRMSEAC_abi3/rep)/(Ud$maht)

J1RB[,1]=(J1RBHT_abi1/rep)/(Ud$netopalk)
J1RB[,2]=(J1RBHT_abi2/rep)/(Ud$kulud)
J1RB[,3]=(J1RBHT_abi3/rep)/(Ud$maht)
J1RB[,4]=(J1RBRW_abi1/rep)/(Ud$netopalk)
J1RB[,5]=(J1RBRW_abi2/rep)/(Ud$kulud)
J1RB[,6]=(J1RBRW_abi3/rep)/(Ud$maht)
J1RB[,7]=(J1RBAC_abi1/rep)/(Ud$netopalk)
J1RB[,8]=(J1RBAC_abi2/rep)/(Ud$kulud)
J1RB[,9]=(J1RBAC_abi3/rep)/(Ud$maht)
```

```
J2RRMSE[,1]=sqrt(J2RRMSEHT_abi1/rep)/(Ud$netopalk)
J2RRMSE[,2]=sqrt(J2RRMSEHT_abi2/rep)/(Ud$kulud)
J2RRMSE[,3]=sqrt(J2RRMSEHT_abi3/rep)/(Ud$maht)
J2RRMSE[,4]=sqrt(J2RRMSERW_abi1/rep)/(Ud$netopalk)
J2RRMSE[,5]=sqrt(J2RRMSERW_abi2/rep)/(Ud$kulud)
J2RRMSE[,6]=sqrt(J2RRMSERW_abi3/rep)/(Ud$maht)
J2RRMSE[,7]=sqrt(J2RRMSEAC_abi1/rep)/(Ud$netopalk)
J2RRMSE[,8]=sqrt(J2RRMSEAC_abi2/rep)/(Ud$kulud)
J2RRMSE[,9]=sqrt(J2RRMSEAC_abi3/rep)/(Ud$maht)

J2RB[,1]=(J2RBHT_abi1/rep)/(Ud$netopalk)
J2RB[,2]=(J2RBHT_abi2/rep)/(Ud$kulud)
J2RB[,3]=(J2RBHT_abi3/rep)/(Ud$maht)
J2RB[,4]=(J2RBRW_abi1/rep)/(Ud$netopalk)
J2RB[,5]=(J2RBRW_abi2/rep)/(Ud$kulud)
J2RB[,6]=(J2RBRW_abi3/rep)/(Ud$maht)
J2RB[,7]=(J2RBAC_abi1/rep)/(Ud$netopalk)
J2RB[,8]=(J2RBAC_abi2/rep)/(Ud$kulud)
J2RB[,9]=(J2RBAC_abi3/rep)/(Ud$maht)
```

Lisa 3 Programmi kood HG disaini korral

```
#Paigaldame vajaliku paketi
#install.packages("sampling")
library(sampling)

#Loeme sisse andmestiku esialgse üldkogumi kohta
U=read.csv("C:\\Users\\Kristjan\\Desktop\\yldkogum1.csv", header = T ,  sep = ",")
#Loeme sisse andmestiku uue üldkogumi, kus esialgse i. objekt on esindatud
#leibkonna liikmete arv korda.
andmed = read.csv("C:\\Users\\Kristjan\\Desktop\\HG_uldkogum.csv", header = T ,  s
attach(andmed)

N_uus=nrow(andmed)    #Uue üldkogumi maht
rep=1000    #Valimi võtmiste arv
n=1000
#Lisame vanasse andmestikku tunnuse, mis on konstantselt 1
uhed_v=rep(1, nrow(U))
U$uhed_v=uhed_v
#Lisame uude andmestikku tunnuse, mis on konstantselt 1
uhed=rep(1, N_uus)
andmed$uhed=uhed

Y1=sum(U$netopalk)    #Uuritava tunnuse kogupalk tegelik(üldkogumi) summa
Y2=sum(U$Kulud2)    # Uuritava tunnuse tarbimiskulud tegelik  summa
X1=sum(U$siirdetulu)    #Abitunnuse siirdetulu tegelik summa
X2=sum(U$lapsi)    #Abitunnuse laste arv tegelik summa

#Leiame osakogumeid iseloomustavad suurused Üldkogumis
Ud=data.frame(osakogum_nr=1:12)
Ud$netopalk=as.vector(by(U$netopalk, U$Osakogum, sum))
Ud$maht=as.vector(by(U$uhed_v, U$Osakogum, sum))
```

```
Ud$keskmine_np=Ud$netopalk/Ud$maht
Ud$minimaalne_np=as.vector(by(U$netopalk, U$Osakogum, min))
Ud$maksimaalne_np=as.vector(by(U$netopalk, U$Osakogum, max))
Ud$std_np=as.vector(by(U$netopalk, U$Osakogum, sd))
Ud$kulud=as.vector(by(U$Kulud2, U$Osakogum, sum))
Ud$keskmine_tk=Ud$kulud/Ud$maht
Ud$minimaalne_tk=as.vector(by(U$Kulud2, U$Osakogum, min))
Ud$maksimaalne_tk=as.vector(by(U$Kulud2, U$Osakogum, max))
Ud$std_tk=as.vector(by(U$Kulud2, U$Osakogum, sd))

#Defineerime kalibreerimise lisainformatsiooniga antud HG valimi korral
kalibreerimine=function(valim){
abi=andmed[valim,]    #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused

#Eraldame abitunnuste vektoris
x1=abi$siirdetulu
x2=abi$lapsi

#Leiame abitunnuste hinnangud
X1_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(x1))
X2_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(x2))

#Teeme vektori (X-X_hinnang)
vec=c(X1-X1_hinnang, X2-X2_hinnang)

w=rep(0,n)    #Vektor, millese leiame kaalud
M=matrix(c(0,0,0,0),2)    #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:n){
x_vec=c(x1[i],x2[i])
m_i=(N_uus/(n*(abi$lbk_suurus)[i]))*(x_vec%*%t(x_vec))
M=M+m_i
}
M_poord=solve(M)    #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord    #Vektor lambda

#Leiame A-informatsiooniga kalibreeritud kaalud
for (i in 1:n){
x_vec=c(x1[i],x2[i])
w[i]=(N_uus/(n*abi$lbk_suurus[i]))*(1+lambda%*%x_vec)
}

#Tagastame saadud kaalud
return (w)
}

#Defineerime kalibreerimise C-informatsiooniga, kus parameetritena anname HG valimi, A-
#informatsiooniga kalibreeritud kaalud ja kas uuritavate tunnuste tegelikud  summad või hinnangud

korduvkaalumine=function(valim, w, Y1, Y2){
abi=andmed[valim,]    #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused

#Eraldame C-informatsiooni vektorid
y1=abi$netopalk
y2=abi$Kulud2

#Leiame uuritavate tunnuste kalibreeritud hinnangud
Y1_CALhinnang=sum(w*y1)
```

```
Y2_CALhinnang=sum(w*y2)

#Teeme vektori (Y - Y-i kalibreeritud hinnang)
vec=c(Y1-Y1_CALhinnang, Y2-Y2_CALhinnang)

w_RW=rep(1,n)    #Vektor, millese leiame kaalud
M=matrix(c(0,0,0,0),2)    #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:n){
y_vec=c(y1[i],y2[i])
m_i= w[i]*(y_vec%*%t(y_vec))
M=M+m_i
}

M_poord=solve(M)    #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord    #Vektor lambda

#Leiame C-informatsiooniga kalibreeritud kaalud
for (i in 1:n){
y_vec=c(y1[i],y2[i])
w_RW[i]=(w[i])*(1+lambda%*%y_vec)
}


#Tagstame leitud kaalud
return (w_RW)
}

#Defineerime funktsiooni, mis leiab ette antud HG valiku valimile vastavad AC-kalibreeritud kaalud
#Parameetriteks valim ja kas uuritavate tunnuste tegelikud summad #või hinnangud

ACkalibreerimine=function(valim, Y1, Y2){
abi=andmed[valim,] #Teeme abistava andmestiku, kus on vaid valimisse kuuluvad vaatlused

#Eraldame uuritavate ja abitunnuste vektorid
x1=abi$siirdetulu
x2=abi$lapsi
y1=abi$netopalk
y2=abi$Kulud2

#Leiame abitunnuste hinnangud
X1_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(x1))
X2_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(x2))
Y1_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(y1))
Y2_hinnang=sum((N_uus/(n*abi$lbk_suurus))*(y2))

#Teeme vektori ((X-X_hinnang)' (Y-Y_hinnang)')'
vec=c(X1-X1_hinnang, X2-X2_hinnang,Y1-Y1_hinnang,Y2-Y2_hinnang)

w_AC=rep(0,n) #Vektor, millese leiame kaalud
M=matrix(rep(0,16),4) #Maatriks M, mis algul sisaldab nulle

#Leiame maatriksi M
for (i in 1:n){
xy_vec=c(x1[i],x2[i],y1[i],y2[i])
m_i=(N_uus/(n*(abi$lbk_suurus)[i]))*(xy_vec%*%t(xy_vec))
M=M+m_i
}
```

```
M_poord=solve(M) #Maatriksi M pöördmaatriks
lambda=t(vec)%*%M_poord #Vektor lambda

#Leiame AC-kalibreeritud kaalud
for (i in 1:n){
xy_vec=c(x1[i],x2[i],y1[i],y2[i])
w_AC[i]=(N_uus/(n*abi$lbk_suurus[i]))*(1+lambda%*%xy_vec)
}
#Tagastame AC-kalibreeritud kaalud
return (w_AC)
}


#Defineerime funktsiooni, mis leiab etteantud kaalude põhjal
#tunnuste hinnangud osakogumites
hindamine=function(valim, kaalud){
abi=andmed[valim, ]

#Leiame tunnuste valimi vektori
y1=abi$netopalk
y2=abi$Kulud2

#Leiame vektorid (kaal*tunnus)
vec1=kaalud*y1
vec2=kaalud*y2

#Leiame vektori, mis määrab iga valimi elemendile vastava osakogumi
osak=abi$Osakogum

#Leiame kogusumma hinnangud  osakogumites
OK_hinnang1=as.vector(by(vec1,osak,sum))
OK_hinnang2=as.vector(by(vec2,osak,sum))
OK_hinnang3=as.vector(by(kaalud,osak,sum))

#Moodustame neist tabeli
OK_hinnangud=data.frame(osakogum=1:12)
OK_hinnangud$y1=OK_hinnang1
OK_hinnangud$y2=OK_hinnang2
OK_hinnangud$Nd=OK_hinnang3

#Tagastame viimase tabeli
return (OK_hinnangud)
}

#Teeme tühjad maatriksid, kuhu hakkame
#osakogumite hinnanguid koguma (juhul, kus üldkogumi summa on teada, J1 tähistab juhtu 1)
J1HT_osakY1=matrix(nrow=12, ncol=rep)
J1RW_osakY1=matrix(nrow=12, ncol=rep)
J1AC_osakY1=matrix(nrow=12, ncol=rep)
J1HT_osakY2=matrix(nrow=12, ncol=rep)
J1RW_osakY2=matrix(nrow=12, ncol=rep)
J1AC_osakY2=matrix(nrow=12, ncol=rep)
J1HT_osakN=matrix(nrow=12, ncol=rep)
J1RW_osakN=matrix(nrow=12, ncol=rep)
J1AC_osakN=matrix(nrow=12, ncol=rep)

#Abistavad suurused RRMSE-de ja RB-de leidmiseks (U summa teada)
J1RRMSEHT_abi1=rep(0,12)
```

```
J1RRMSEHT_abi2=rep(0,12)
J1RRMSEHT_abi3=rep(0,12)
J1RRMSERW_abi1=rep(0,12)
J1RRMSERW_abi2=rep(0,12)
J1RRMSERW_abi3=rep(0,12)
J1RRMSEAC_abi1=rep(0,12)
J1RRMSEAC_abi2=rep(0,12)
J1RRMSEAC_abi3=rep(0,12)

J1RBHT_abi1=rep(0,12)
J1RBHT_abi2=rep(0,12)
J1RBHT_abi3=rep(0,12)
J1RBRW_abi1=rep(0,12)
J1RBRW_abi2=rep(0,12)
J1RBRW_abi3=rep(0,12)
J1RBAC_abi1=rep(0,12)
J1RBAC_abi2=rep(0,12)
J1RBAC_abi3=rep(0,12)

#Teeme tühjad maatriksid, kuhu hakkame osakogumite hinnanguid koguma (juhul, kus üldkogumi
#summa on hinnatud IU-st, J2 tähistab juhtu 2)
J2HT_osakY1=matrix(nrow=12, ncol=rep)
J2RW_osakY1=matrix(nrow=12, ncol=rep)
J2AC_osakY1=matrix(nrow=12, ncol=rep)
J2HT_osakY2=matrix(nrow=12, ncol=rep)
J2RW_osakY2=matrix(nrow=12, ncol=rep)
J2AC_osakY2=matrix(nrow=12, ncol=rep)
J2HT_osakN=matrix(nrow=12, ncol=rep)
J2RW_osakN=matrix(nrow=12, ncol=rep)
J2AC_osakN=matrix(nrow=12, ncol=rep)

#Abistavad suurused RRMSE-de ja RB-de leidmiseks (U summa teada)
J2RRMSEHT_abi1=rep(0,12)
J2RRMSEHT_abi2=rep(0,12)
J2RRMSEHT_abi3=rep(0,12)
J2RRMSERW_abi1=rep(0,12)
J2RRMSERW_abi2=rep(0,12)
J2RRMSERW_abi3=rep(0,12)
J2RRMSEAC_abi1=rep(0,12)
J2RRMSEAC_abi2=rep(0,12)
J2RRMSEAC_abi3=rep(0,12)

J2RBHT_abi1=rep(0,12)
J2RBHT_abi2=rep(0,12)
J2RBHT_abi3=rep(0,12)
J2RBRW_abi1=rep(0,12)
J2RBRW_abi2=rep(0,12)
J2RBRW_abi3=rep(0,12)
J2RBAC_abi1=rep(0,12)
J2RBAC_abi2=rep(0,12)
J2RBAC_abi3=rep(0,12)

#Teeme vektorid, kuhu kogume IU kogusummade hinnangud
IU_hY1=rep(1, rep)
IU_hY2=rep(1, rep)
IU_hN=rep(1,rep)

#Teeme maatriksid kuhu kogume osakogumitesse sattunud vaatluste arve
#antud valimi korral
```

```
NdIU=matrix(nrow=12,ncol=rep)
NdIIU=matrix(nrow=12,ncol=rep)

for (i in 1:rep){
n=1000

#Võtame nii IU kui ka IIU valimid
kIU=srswor(2000,N_uus
kIIU=srswor(1000,N_uus)
valimIU=(1:N_uus)[kIU==1]
valimIIU=(1:N_uus)[kIIU==1]

abiIU=andmed[valimIU,]
abiIIU=andmed[valimIIU,]

#Leiame osakogumitesse sattunud sattunud objektide arvud
#ja lisame maatriksitesse
vaatlusiIU=as.vector(by(abiIU$uhed,abiIU$Osak,sum))
vaatlusiIIU=as.vector(by(abiIIU$uhed,abiIIU$Osak,sum))
NdIU[,i]=vaatlusiIU
NdIIU[,i]=vaatlusiIIU

#Leiame HT kaalud ja IU hinnangud
w_IU=N_uus/(2000*abiIU$lbk_suurus)
w_IIU=N_uus/(n*abiIIU$lbk_suurus)

Y1_hinnangIU=sum(w_IU*abiIU$netopalk)
Y2_hinnangIU=sum(w_IU*abiIU$Kulud2)
IU_hY1[i]=Y1_hinnangIU
IU_hY2[i]=Y2_hinnangIU
IU_hN[i]=sum(w_IU)

w_C=kalibreerimine(valimIIU)

#Leiame AC ja RW kaalud nii teada  oleva üldkogumi kui
#ka hinnatud suuruse korral
J1w_RW=korduvkaalumine(valimIIU,w_C,n,Y1,Y2)
J1w_AC=ACkalibreerimine(valimIIU,n,Y1,Y2)
J2w_RW=korduvkaalumine(valimIIU,w_C,n,Y1_hinnangIU,Y2_hinnangIU)
J2w_AC=ACkalibreerimine(valimIIU,n,Y1_hinnangIU,Y2_hinnangIU)

#Leiame HT, RW ja AC hinnangud osakogumites esimesel juhul(teada  olevad summad)
HT_hinnangud1=hindamine(valimIIU,w_IIU)
RW_hinnangud1=hindamine(valimIIU,J1w_RW)
AC_hinnangud1=hindamine(valimIIU,J1w_AC)

#Leiame HT, RW ja AC hinnangud osakogumites teisel juhul(hinnatud summad)
HT_hinnangud2=hindamine(valimIIU,w_IIU)
RW_hinnangud2=hindamine(valimIIU,J2w_RW)
AC_hinnangud2=hindamine(valimIIU,J2w_AC)

#Täidame maatriksid vastavate hinnangutega
J1HT_osakY1[,i]=HT_hinnangud1[,2]
J1RW_osakY1[,i]=RW_hinnangud1[,2]
J1AC_osakY1[,i]=AC_hinnangud1[,2]
J1HT_osakY2[,i]=HT_hinnangud1[,3]
J1RW_osakY2[,i]=RW_hinnangud1[,3]
J1AC_osakY2[,i]=AC_hinnangud1[,3]
J1HT_osakN[,i]=HT_hinnangud1[,4]
```

```
J1RW_osakN[,i]=RW_hinnangud1[,4]
J1AC_osakN[,i]=AC_hinnangud1[,4]

J2HT_osakY1[,i]=HT_hinnangud2[,2]
J2RW_osakY1[,i]=RW_hinnangud2[,2]
J2AC_osakY1[,i]=AC_hinnangud2[,2]
J2HT_osakY2[,i]=HT_hinnangud2[,3]
J2RW_osakY2[,i]=RW_hinnangud2[,3]
J2AC_osakY2[,i]=AC_hinnangud2[,3]
J2HT_osakN[,i]=HT_hinnangud2[,4]
J2RW_osakN[,i]=RW_hinnangud2[,4]
J2AC_osakN[,i]=AC_hinnangud2[,4]

#Summerime
J1RRMSEHT_abi1=J1RRMSEHT_abi1+(HT_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSEHT_abi2=J1RRMSEHT_abi2+(HT_hinnangud1[,3]-Ud$kulud)**2
J1RRMSEHT_abi3=J1RRMSEHT_abi3+(HT_hinnangud1[,4]-Ud$maht)**2
J1RRMSERW_abi1=J1RRMSERW_abi1+(RW_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSERW_abi2=J1RRMSERW_abi2+(RW_hinnangud1[,3]-Ud$kulud)**2
J1RRMSERW_abi3=J1RRMSERW_abi3+(RW_hinnangud1[,4]-Ud$maht)**2
J1RRMSEAC_abi1=J1RRMSEAC_abi1+(AC_hinnangud1[,2]-Ud$netopalk)**2
J1RRMSEAC_abi2=J1RRMSEAC_abi2+(AC_hinnangud1[,3]-Ud$kulud)**2
J1RRMSEAC_abi3=J1RRMSEAC_abi3+(AC_hinnangud1[,4]-Ud$maht)**2

J1RBHT_abi1=J1RBHT_abi1+(HT_hinnangud1[,2]-Ud$netopalk)
J1RBHT_abi2=J1RBHT_abi2+(HT_hinnangud1[,3]-Ud$kulud)
J1RBHT_abi3=J1RBHT_abi3+(HT_hinnangud1[,4]-Ud$maht)
J1RBRW_abi1=J1RBRW_abi1+(RW_hinnangud1[,2]-Ud$netopalk)
J1RBRW_abi2=J1RBRW_abi2+(RW_hinnangud1[,3]-Ud$kulud)
J1RBRW_abi3=J1RBRW_abi3+(RW_hinnangud1[,4]-Ud$maht)
J1RBAC_abi1=J1RBAC_abi1+(AC_hinnangud1[,2]-Ud$netopalk)
J1RBAC_abi2=J1RBAC_abi2+(AC_hinnangud1[,3]-Ud$kulud)
J1RBAC_abi3=J1RBAC_abi3+(AC_hinnangud1[,4]-Ud$maht)

J2RRMSEHT_abi1=J2RRMSEHT_abi1+(HT_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSEHT_abi2=J2RRMSEHT_abi2+(HT_hinnangud2[,3]-Ud$kulud)**2
J2RRMSEHT_abi3=J2RRMSEHT_abi3+(HT_hinnangud2[,4]-Ud$maht)**2
J2RRMSERW_abi1=J2RRMSERW_abi1+(RW_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSERW_abi2=J2RRMSERW_abi2+(RW_hinnangud2[,3]-Ud$kulud)**2
J2RRMSERW_abi3=J2RRMSERW_abi3+(RW_hinnangud2[,4]-Ud$maht)**2
J2RRMSEAC_abi1=J2RRMSEAC_abi1+(AC_hinnangud2[,2]-Ud$netopalk)**2
J2RRMSEAC_abi2=J2RRMSEAC_abi2+(AC_hinnangud2[,3]-Ud$kulud)**2
J2RRMSEAC_abi3=J2RRMSEAC_abi3+(AC_hinnangud2[,4]-Ud$maht)**2

J2RBHT_abi1=J2RBHT_abi1+(HT_hinnangud2[,2]-Ud$netopalk)
J2RBHT_abi2=J2RBHT_abi2+(HT_hinnangud2[,3]-Ud$kulud)
J2RBHT_abi3=J2RBHT_abi3+(HT_hinnangud2[,4]-Ud$maht)
J2RBRW_abi1=J2RBRW_abi1+(RW_hinnangud2[,2]-Ud$netopalk)
J2RBRW_abi2=J2RBRW_abi2+(RW_hinnangud2[,3]-Ud$kulud)
J2RBRW_abi3=J2RBRW_abi3+(RW_hinnangud2[,4]-Ud$maht)
J2RBAC_abi1=J2RBAC_abi1+(AC_hinnangud2[,2]-Ud$netopalk)
J2RBAC_abi2=J2RBAC_abi2+(AC_hinnangud2[,3]-Ud$kulud)
J2RBAC_abi3=J2RBAC_abi3+(AC_hinnangud2[,4]-Ud$maht)
}

#Teeme tulemuste maatriksid iga uuritava tunnuse jaoks,
#et saada paremat ülevaadet
#Tunnused tabelites: keskmine HT hinnang, HT std, keskmine
#RW hinnang, RW std, keskmine AC hinnang, AC std
```

```
#Juht1
J1TULEMUS_Y1=matrix(nrow=12,ncol=6)
J1TULEMUS_Y2=matrix(nrow=12,ncol=6)
J1TULEMUS_N=matrix(nrow=12,ncol=6)

#Juht2
J2TULEMUS_Y1=matrix(nrow=12,ncol=6)
J2TULEMUS_Y2=matrix(nrow=12,ncol=6)
J2TULEMUS_N=matrix(nrow=12,ncol=6)

for(i in 1:12){
#Juht 1
J1TULEMUS_Y1[i,1]=mean(J1HT_osakY1[i,])
J1TULEMUS_Y1[i,2]=sd(J1HT_osakY1[i,])
J1TULEMUS_Y1[i,3]=mean(J1RW_osakY1[i,])
J1TULEMUS_Y1[i,4]=sd(J1RW_osakY1[i,])
J1TULEMUS_Y1[i,5]=mean(J1AC_osakY1[i,])
J1TULEMUS_Y1[i,6]=sd(J1AC_osakY1[i,])

J1TULEMUS_Y2[i,1]=mean(J1HT_osakY2[i,])
J1TULEMUS_Y2[i,2]=sd(J1HT_osakY2[i,])
J1TULEMUS_Y2[i,3]=mean(J1RW_osakY2[i,])
J1TULEMUS_Y2[i,4]=sd(J1RW_osakY2[i,])
J1TULEMUS_Y2[i,5]=mean(J1AC_osakY2[i,])
J1TULEMUS_Y2[i,6]=sd(J1AC_osakY2[i,])

J1TULEMUS_N[i,1]=mean(J1HT_osakN[i,])
J1TULEMUS_N[i,2]=sd(J1HT_osakN[i,])
J1TULEMUS_N[i,3]=mean(J1RW_osakN[i,])
J1TULEMUS_N[i,4]=sd(J1RW_osakN[i,])
J1TULEMUS_N[i,5]=mean(J1AC_osakN[i,])
J1TULEMUS_N[i,6]=sd(J1AC_osakN[i,])

#Juht 2
J2TULEMUS_Y1[i,1]=mean(J2HT_osakY1[i,])
J2TULEMUS_Y1[i,2]=sd(J2HT_osakY1[i,])
J2TULEMUS_Y1[i,3]=mean(J2RW_osakY1[i,])
J2TULEMUS_Y1[i,4]=sd(J2RW_osakY1[i,])
J2TULEMUS_Y1[i,5]=mean(J2AC_osakY1[i,])
J2TULEMUS_Y1[i,6]=sd(J2AC_osakY1[i,])

J2TULEMUS_Y2[i,1]=mean(J2HT_osakY2[i,])
J2TULEMUS_Y2[i,2]=sd(J2HT_osakY2[i,])
J2TULEMUS_Y2[i,3]=mean(J2RW_osakY2[i,])
J2TULEMUS_Y2[i,4]=sd(J2RW_osakY2[i,])
J2TULEMUS_Y2[i,5]=mean(J2AC_osakY2[i,])
J2TULEMUS_Y2[i,6]=sd(J2AC_osakY2[i,])

J2TULEMUS_N[i,1]=mean(J2HT_osakN[i,])
J2TULEMUS_N[i,2]=sd(J2HT_osakN[i,])
J2TULEMUS_N[i,3]=mean(J2RW_osakN[i,])
J2TULEMUS_N[i,4]=sd(J2RW_osakN[i,])
J2TULEMUS_N[i,5]=mean(J2AC_osakN[i,])
J2TULEMUS_N[i,6]=sd(J2AC_osakN[i,])
}


#Teeme maatriksid, kus on kõigi kolme tunnuse kõigi kolme hinnangu RRMSE ja RB
#Veergudes esmalt HT hinnagule vastavad suurused kolme tunnuse korral ja siis RW ja AC
```

```
J1RRMSE=matrix(nrow=12, ncol=9)
J1RB=matrix(nrow=12, ncol=9)
J2RRMSE=matrix(nrow=12, ncol=9)
J2RB=matrix(nrow=12, ncol=9)


J1RRMSE[,1]=sqrt(J1RRMSEHT_abi1/rep)/(Ud$netopalk)
J1RRMSE[,2]=sqrt(J1RRMSEHT_abi2/rep)/(Ud$kulud)
J1RRMSE[,3]=sqrt(J1RRMSEHT_abi3/rep)/(Ud$maht)
J1RRMSE[,4]=sqrt(J1RRMSERW_abi1/rep)/(Ud$netopalk)
J1RRMSE[,5]=sqrt(J1RRMSERW_abi2/rep)/(Ud$kulud)
J1RRMSE[,6]=sqrt(J1RRMSERW_abi3/rep)/(Ud$maht)
J1RRMSE[,7]=sqrt(J1RRMSEAC_abi1/rep)/(Ud$netopalk)
J1RRMSE[,8]=sqrt(J1RRMSEAC_abi2/rep)/(Ud$kulud)
J1RRMSE[,9]=sqrt(J1RRMSEAC_abi3/rep)/(Ud$maht)

J1RB[,1]=(J1RBHT_abi1/rep)/(Ud$netopalk)
J1RB[,2]=(J1RBHT_abi2/rep)/(Ud$kulud)
J1RB[,3]=(J1RBHT_abi3/rep)/(Ud$maht)
J1RB[,4]=(J1RBRW_abi1/rep)/(Ud$netopalk)
J1RB[,5]=(J1RBRW_abi2/rep)/(Ud$kulud)
J1RB[,6]=(J1RBRW_abi3/rep)/(Ud$maht)
J1RB[,7]=(J1RBAC_abi1/rep)/(Ud$netopalk)
J1RB[,8]=(J1RBAC_abi2/rep)/(Ud$kulud)
J1RB[,9]=(J1RBAC_abi3/rep)/(Ud$maht)

J2RRMSE[,1]=sqrt(J2RRMSEHT_abi1/rep)/(Ud$netopalk)
J2RRMSE[,2]=sqrt(J2RRMSEHT_abi2/rep)/(Ud$kulud)
J2RRMSE[,3]=sqrt(J2RRMSEHT_abi3/rep)/(Ud$maht)
J2RRMSE[,4]=sqrt(J2RRMSERW_abi1/rep)/(Ud$netopalk)
J2RRMSE[,5]=sqrt(J2RRMSERW_abi2/rep)/(Ud$kulud)
J2RRMSE[,6]=sqrt(J2RRMSERW_abi3/rep)/(Ud$maht)
J2RRMSE[,7]=sqrt(J2RRMSEAC_abi1/rep)/(Ud$netopalk)
J2RRMSE[,8]=sqrt(J2RRMSEAC_abi2/rep)/(Ud$kulud)
J2RRMSE[,9]=sqrt(J2RRMSEAC_abi3/rep)/(Ud$maht)

J2RB[,1]=(J2RBHT_abi1/rep)/(Ud$netopalk)
J2RB[,2]=(J2RBHT_abi2/rep)/(Ud$kulud)
J2RB[,3]=(J2RBHT_abi3/rep)/(Ud$maht)
J2RB[,4]=(J2RBRW_abi1/rep)/(Ud$netopalk)
J2RB[,5]=(J2RBRW_abi2/rep)/(Ud$kulud)
J2RB[,6]=(J2RBRW_abi3/rep)/(Ud$maht)
J2RB[,7]=(J2RBAC_abi1/rep)/(Ud$netopalk)
J2RB[,8]=(J2RBAC_abi2/rep)/(Ud$kulud)
J2RB[,9]=(J2RBAC_abi3/rep)/(Ud$maht)
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Kristjan Kokorev,
(*autori nimi*)
(sünnikuupäev: 30.09.1988)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Osakogumite kooskõlaline hindamine PPS hüpergeomeetrilise ja Poissoni valiku korral,
(*lõputöö pealkiri*)

mille juhendaja on Kaur Lumiste,
(*juhendaja nimi*)

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **06.05.2013**