Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Matemaatika ja statistika instituut

Sören Mirski

Tõenäosuslik valikuuring tarkvara R pakettide „sampling“ ja „survey“ abil

Matemaatilise statistika erialal

Bakalaureusetöö (9 EAP)

Juhendaja lektor Natalja Lepik

Tartu 2017

Tõenäosuslik valikuuring tarkvara R pakettide „sampling“ ja „survey“ abil

Bakalaureusetöö

Sören Mirski

Lühikokkuvõte. Käesolev bakalaureusetöö annab ülevaate statistikatarkvara R lisapakettide „sampling“ ja „survey“ funktsioonidest, millega saab teostada levinumaid tõenäosuslikke valikuid ja hinnata üldkogumi kogusummat ning selle dispersiooni. Iga töös kirjeldatud funktsiooni kasutamise kohta tuuakse vähemalt üks praktiline näide. Võimalusel lahendatakse sama näide mõlema paketi funktsioonidega, et tuvastada võimalikke erinevusi kahe paketi vahel. Eelistatumaks osutus pakett „sampling“, kuna selle funktsioonidega saab teostada rohkem tõenäosuslikke valikuid ning hinnangute arvutamiseks kasutatakse kursuses „Valikuuringute teooria I“ käsitletud valemeid.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika

Märksõnad: valikuuringud, R (programmeerimiskeel), tõenäosuslikud valikud, hindamine

Survey Sampling Using R Packages 'sampling' and 'survey'

Bachelor's thesis

Sören Mirski

Abstract. The purpose of this bachelor's thesis is to give an overview of the R add-on packages 'sampling' and 'survey'. Functions that can be used to carry out common probability sampling methods and estimate the population total and its variance are described. At least one practical example is given for every described function. Wherever possible, every practical example is solved using functions from both packages to discover any differences between the two. The preferred package ended up being 'sampling' because it contains more functions that can be used for probability sampling. Additionally, the functions used for estimation use formulas covered in the course 'Survey Sampling Theory I'.

CERCS research specialisation: P160 Statistics, operations research, programming, actuarial mathematics

Keywords: sample surveys, R (programming language), probability sampling, estimation

Sisukord

Sissejuhatus

Käesoleva bakalaureusetöö eesmärk on tutvustada statistikatarkvara R lisapakettide „sampling“ ja „survey“ võimalusi tõenäosusliku valiku teostamiseks ning üldkogumi kogusumma ja selle dispersiooni hindamiseks. Töös on keskendutud neljale tõenäosuslikule valikumeetodile, mida käsitletakse kursuses „Valikuuringute teooria I“. Nendeks on lihtne juhuslik valik, süstemaatiline valik, Poissoni valik ja kihtvalik. Üldkogumi kogusumma hindamisel on vaadeldud Horvitz-Thompsoni hinnangu, suhte- ja kalibreerimishinnangu leidmist.

Seni on kursuse „Valikuuringute teooria I“ praktikumides kasutatud valikuuringute jaoks spetsiaalselt välja töötatud statistikatarkvara SAS funktsioone. Antud töös esitatud materjal on uus ning seda on plaanis kursuse praktikumides alternatiivina kasutusele võtta. Enamikes töös esitatud praktilistes näiteprogrammides on sarnaselt mainitud kursuse praktikumidega kasutatud üldkogumina hüpoteetilise küla *StatVillage* versiooni *Mini village* või selle osakogumit. Seetõttu on töös toodud näidete abil lihtsam koostada juhendit, mis tutvustaks tarkvara R võimalusi valikuuringute valdkonnas ja mida saaks kasutada kursuses „Valikuuringute teooria I“.

Töö koosneb kahest suuremast peatükist. Esimese peatüki alguses defineeritakse töös kasutatavad valikuteooria mõisted ja antakse ülevaade hüpoteetilisest külast *StatVillage*. Seejärel kirjeldatakse lisapakettide „sampling“ ja „survey“ funktsioone, millega saab levinumaid tõenäosuslikke valikuid teostada. Teises peatükis kasutatakse esimeses peatükis võetud valimeid, loodud andmestikke ning pakettide „sampling“ ja „survey“ funktsioone, et hinnata üldkogumi kogusummat ja selle dispersiooni või standardhälvet.

Töö vormistamiseks on kasutatud tarkvara R kasutajaliidest RStudio ja paketti „Sweave“, mis võimaldab koostada LaTeX formaadis dokumente koos tarkvara R programmide ja väljunditega. Töös esitatud näiteprogrammide kirjutamiseks on kasutatud tarkvara R versiooni 3.2.2 ning lisapakette „sampling“ (versioon 2.8) ja „survey“ (versioon 3.31-5).

Autor tänab antud bakalaureusetöö juhendajat Natalja Lepikut asjakohaste paranduste, märkuste ja soovituste eest.

1 Tõenäosusliku valiku teostamine

Käesolevas peatükis on välja toodud tarkvara R lisapakettide „sampling“ ja „survey“ funktsioonid, millega saab lõplikust üldkogumist tõenäosuslikke valimeid võtta. Keskendutud on neljale levinud tõenäosuslikule valikumeetodile, milleks on lihtne juhuslik valik, süstemaatiline valik, Poissoni valik ja kihtvalik.

1.1 Vajalikud mõisted valikuteooriast

Antud alapeatükis defineeritakse peamised töös kasutatavad valikuteooria mõisted. Allikana on kasutatud õpikut (Traat ja Inno, 1997).

Definitsioon 1. Freimiks nimetatakse loetelu, mille abil pääseb üldkogumi objekti juurde ja mis sisaldab igat objekti identifitseerivaid ja selle ülesleidmist võimaldavaid parameetreid.

Definitsioon 2. Tõenäosuslikuks valikuks nimetatakse niisugust valikut lõplikust üldkogumist, mille korral

- saab defineerida kõigi võimalike valimite hulga S;
- iga valimi $s \in S$ jaoks on teada tema valikutõenäosus $p(s)$;
- iga üldkogumi objekti valimisse sattumise tõenäosus on teada ja on positiivne;
- valimi võtmiseks kasutatav juhuslik mehhanism tagab, et valimi s valikutõenäosus on $p(s)$.

Tõenäosuslikuks valimiks nimetatakse hulka $s \in S$, mis on realiseerunud kõiki ülaltoodud tingimusi arvestades. Edaspidi mõistetakse valimi all üksnes tõenäosuslikke valimeid. Tõenäosuslikud valikumeetodid jagunevad kaheks: eristatakse tagasipanekuta (TTA) ja tagasipanekuga (TGA) valikut. Tagasipanekuta valiku korral saab iga üldkogumi objekt valimisse sattuda maksimaalselt ühe korra. Tagasipanekuga valiku korral seda piirangut ei ole, suvaline üldkogumi objekt võib valimisse sattuda korduvalt.

Definitsoon 3. Valikudisainiks nimetatakse tõenäosusjaotust $p(s)$ kõigi antud valiku jaoks võimalike valimite hulgal S.

Definitsioon 4. Üldkogumi objekti i ($i = 1, 2, \ldots, N$) kaasamistõenäosuseks π_i nimetatakse tõenäosust, millega see objekt kaasatakse valimisse s antud disaini

korral:

$$\pi_i = \Pr(i \in s).$$

Analoogiliselt defineeritakse kahe üldkogumi objekti i ja j üheaegne kaasamistõenäosus ehk teist järku kaasamistõenäosus antud disaini korral:

$$\pi_{ij} = \Pr(i, j \in s).$$

Tõenäosusliku valimi põhjal saab konstrueerida nn pseudoüldkogumi, mis koosneb valimisse s sattunud objektidest ja nende koopiatest. Valimi elementi $i \in s$ on pseudoüldkogumis d_i korda. See tähendab, et valimi element $i \in s$ esindab d_i üldkogumi objekti. Suurust d_i nimetatakse elemendi $i \in s$ valikukaaluks. Tagasipanekuta valikumeetodite korral kehtib võrdus $d_i = 1/\pi_i$ $(i \in s)$.

1.2 Hüpoteetiline küla *StatVillage*

Antud bakalaureusetöös on enamikes näidetes üldkogum Kanadas asuv hüpoteetiline küla *StatVillage*, mida kasutatakse kursuse „Valikuuringute teooria I“ praktikumides. Järgnev informatsioon küla *StatVillage* kohta pärineb allikast (Schwarz, 1997). Küla *StatVillage* majad moodustavad ristkülikukujulise plokkide võrgustiku, kus iga plokk sisaldab kaheksat maja. Igale majale vastab ploki number ja plokisisene majanumber. Järgnevalt on näitena toodud ploki 35 kuju.

1	2	3
4	**35**	5
6	7	8

Küla *StatVillage* iga majapidamise kohta on mõõdetud 48 tunnuse väärtused, mille hulgas on demograafilisi tunnuseid, sissetulekut ja hõivatust puudutavaid näitajaid jpm. Kõigi näitajate väärtused pärinevad 1991. aastal Kanadas korraldatud rahvaloendusest. Iga mõõdetud tunnuse täpne kirjeldus ja väärtuste kodeeringud on toodud küla *StatVillage* kodulehel.

Külast *StatVillage* on kolm erineva suurusega versiooni:

- *Micro village* – 36 plokki;
- *Mini village* – 60 plokki;
- *Maximal village* – 128 plokki.

Käesolevas töös kasutatakse versiooni *Mini village*. Seetõttu on küla *StatVillage*

kasutavates näidetes üldkogumimaht $N = 480$ majapidamist.

Tõenäosusliku valiku teostamiseks tuleb kõigepealt luua üldkogumi objektidele (antud juhul majapidamistele) vastav freim. Selleks on kasutatud lisas 1 toodud programmi. Pärast freimist valimi võtmist tuleb valimisse sattunud majad küla *StatVillage* kodulehel märgistada ning saadavad andmed salvestada tekstifailina. Kodulehel on olemas statistikatarkvara SAS programm andmete sisselugemiseks loodud failist, kuid vastavat tarkvara R koodi pole. Valimisse sattunud majapidamiste andmete hankimiseks ning tarkvarasse R sisselugemiseks saab kasutada lisas 2 toodud õpetust ja programmi.

1.3 Pakett „sampling“

Käesolevas alapeatükis esitatud informatsioon paketi „sampling“ ja selle funktsioonide kohta pärineb allikast (Tillé ja Matei, 2016).

Tarkvara R lisapaketi „sampling“ autorid on Yves Tillé ja Alina Matei (Šveits, Neuchâteli Ülikool) ning uusim versioon (2.8), mida on antud töös kasutatud, avaldati 22. detsembril 2016. Pakett sisaldab funktsioone, millega saab teostada erinevaid tõenäosuslikke valikuid, hinnata huvipakkuva tunnuse üldkogumi kogusummat ja selle dispersiooni erinevate meetodite abil.

Lisapaketis leiduvate funktsioonide kasutamiseks tuleb pakett kõigepealt käsuga `install.packages("paketinimi")` installeerida. Kui pakett on arvutisse paigaldatud, siis seda uuesti paigaldama ei pea. Samas tuleb iga kord tarkvara R käivitades soovitud pakett sisse laadida käsuga `library(paketinimi)`. Järgmine programm paigaldab lisapaketi „sampling“.

```
install.packages("sampling") # vaja ainult esmakordsel paigaldamisel
library(sampling) # paketi sisselaadimine
```

Järgnevas neljas alapeatükis toodud tõenäosuslike valikumeetodite kirjeldused pärinevad õpikust (Traat ja Inno, 1997), kui pole viidatud teisiti.

1.3.1 Lihtne juhuslik valik

Üks levinuim tõenäosuslik valikumeetod on lihtne juhuslik valik, mille korral on kõigil üldkogumi objektidel võrdne valimisse sattumise tõenäosus. Iga üldkogumi objekti i $(i = 1, \ldots, N)$ korral kehtib võrdus $\pi_i = n/N$, kus n on valimimaht ja N on üldkogumimaht.

Käesolevas alapeatükis on tutvustatud kahte paketi „sampling“ funktsiooni, mis teostavad lihtsat juhuslikku valikut (nii TTA kui ka TGA). Nendeks funktsioonideks on `srswor()` ja `srswr()`. Funktsiooniga `srswor()` saab teostada lihtsat juhuslikku valikut TTA, milleks tuleb funktsioonile anda järgmised argumendid:

- `n` – planeeritav valimimaht;
- `N` – üldkogumimaht.

Funktsiooni rakendamisel tagastatakse nullidest ja ühtedest koosnev N-mõõtmeline vektor $\mathbf{k} = (k_1, k_2, \ldots, k_N)$. Üldkogumi objekt i sattus valimisse, kui vektori $\mathbf{k}$ element k_i võrdub ühega. Analoogiliselt saab kasutada funktsiooni `srswr()` lihtsa juhusliku valiku TGA teostamiseks. Funktsioon `srswr()` tagastab sarnase vektori $\mathbf{k} = (k_1, k_2, \ldots, k_N)$, kus iga $i = 1, \ldots, N$ korral $k_i \in \{0, 1, 2, \ldots, n\}$. Elemendi k_i väärtus näitab, mitu korda objekt i sattus valimisse.

Enamikes antud töös esitatud näidetes on programmi esimesel real fikseeritud juhuslik seeme, et lugejal oleks võimalik saada samu tulemusi (vt näide 1). Kui mõnes näites pole kasutatud käsku `set.seed(1234)`, siis vastava näite tulemused ei sõltu juhuslikust seemnest. Kõigis näiteprogrammides tähistab teatud ridade alguses olev sümbol `##`, et tegu on tarkvara R väljundiga. Mainitud tähisele järgnev nurksulgudes paiknev arv näitab selle järel oleva elemendi järjekorranumbrit väljastatud vektoris.

Järgnevalt on toodud kaks näidet funktsioonide `srswor()` ja `srswr()` rakendamise kohta.

Näide 1. Lihtsa juhusliku valiku teostamine paketi „sampling“ funktsioonidega

Käesolevas näites võetakse üldkogumist mahuga $N = 10$ viieelemendiline valim ($n = 5$) lihtsa juhusliku valiku abil. Teostatakse nii TTA kui ka TGA valik.

```
set.seed(1234) # kõigis näidetes on kasutatavaks juhuslikuks seemneks 1234

LJV_TTA=srswor(n=5, N=10) # lihtsa juhusliku valiku TTA teostamine
LJV_TTA # saadud valimile vastav nullidest ja ühtedest koosnev vektor

##  [1] 0 1 0 0 1 1 0 1 1 0

LJV_TGA=srswr(n=5, N=10) # lihtsa juhusliku valiku TGA teostamine
LJV_TGA # saadud valimile vastav vektor (eelviimane objekt on valimis kaks korda)

##  [1] 1 0 0 1 0 1 0 0 2 0
```

Näide 2. Lihtne juhuslik valik küla *StatVillage* versioonist *Mini village*

Antud näites on kasutatud üldkogumi freimi (andmestik `freim`), mis loodi lisas 1 toodud programmiga. Kõigepealt valitakse freimist 50 majapidamist lihtsa juhusliku valiku TTA abil. Siis eraldatakse freimist valimisse sattunud majad koos neile vastavate ploki ja maja numbritega.

```
set.seed(1234)

# versiooni Mini village korral on üldkogumimaht N=480
s1=srswor(n=50, N=480) # LJV TTA valim mahuga n=50
(1:480)[s1==1] # valimisse sattunud majade järjekorranumbrid freimis

##  [1]   5  19  21  55  73  81  87  90 100 106 108 110 115 120 124
## [16] 133 137 139 144 146 206 211 219 227 239 243 244 256 272 284
## [31] 292 294 298 299 305 315 326 330 339 357 369 376 390 410 415
## [46] 432 439 448 466 468

# andmestik valimisse sattunud majade ja neile vastavate plokkide numbritega
valim_LJVTTA=freim[s1==1, ]
```

Analoogiliselt saab valimi võtmiseks kasutada lihtsat juhuslikku valikut TGA.

```
set.seed(1234)

s2=srswr(n=50, N=480) # LJV TGA valim (sama valimi- ja üldkogumimaht)
(1:480)[s2!=0] # valimisse sattus 48 erinevat majapidamist

##  [1]  14  28  39  72  81  86  90  92 113 116 117 122 123 124 131
## [16] 135 137 142 149 156 158 185 192 199 210 216 240 245 249 283
## [31] 293 302 305 308 330 333 339 341 344 355 356 366 371 389 413
## [46] 438 445 453

s2[s2!=0] # mitu korda iga majapidamine valimisse sattus

##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [31] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1

# andmestiku loomine, mis sisaldab ka iga majapidamise korduse arvu valimis
# funktsioon cbind() lisab andmestikku freim[s2!=0, ] veeru kordus
valim_LJVTGA=cbind(freim[s2!=0, ], kordus=s2[s2!=0])
valim_LJVTGA[20:25, ] # andmestiku kuju (ridade 20 kuni 25 näitel)

##     number plokk maja kordus
## 156    156    20    4      1
## 158    158    20    6      2
## 185    185    24    1      1
## 192    192    24    8      1
## 199    199    25    7      1
## 210    210    27    2      1
```

Näites 2 lihtsa juhusliku valikuga TTA võetud valimi (andmestik `valim_LJVTTA`) korral läbiti lisas 2 toodud sammud ja loodi andmestik nimega `Yldkogum`. Seda

andmestikku kasutatakse edaspidi mõne funktsiooni tutvustamisel, sest lisas 1 oleva programmiga loodud freim ei sisalda piisavalt informatsiooni, et teostada näiteks kihtvalikut.

Lihtsa juhusliku valiku teostamiseks saab kasutada ka funktsiooni `sample()`, mis on üks tarkvara R põhifunktsioonidest. Selle funktsiooni võimaluste kohta saab lugeda tarkvara R dokumentatsioonist või jooksutades käsku `?sample`. Keerulisemaid valikuid on aga võimalik teostada vaid lisapakettidest leitavate funktsioonidega või neid ise programmeerides.

1.3.2 Süstemaatiline valik

Süstemaatilise valiku korral valitakse üldkogumi freimi esimese m elemendi hulgast juhuslikult võrdse tõenäosusega valimi esimene element. Valimi moodustavad valitud element ja iga järgnev element, mis on freimis fikseeritud sammu m kaugusel valimi eelmisest elemendist. Süstemaatiline valik on TTA valikumeetod.

Kui üldkogumimaht N on fikseeritud, siis samm m määrab valimi suuruse. Kehtib võrdus $N = nm + c$, kus $n = [N/m]$ on valimimaht (nurksulud tähistavad täisosa võtmist) ja täisarv c on valikujääk, $0 \leq c < m$. Realiseeruv valimimaht n_s sõltub esimese m elemendi hulgast juhuslikult valitud alguspunktist r:

$$n_s = \begin{cases} n, & \text{kui } r > c; \\ n+1, & \text{kui } r \leq c. \end{cases}$$

Kui näiteks üldkogumimahu $N = 480$ korral on fikseeritud samm $m = 9$, siis $n = 53$ ja $c = 3$.

Süstemaatilist valikut teostab paketi „sampling“ funktsioon `UPsystematic(pik)`, mille ainus argument on N-mõõtmeline kaasamistõenäosuste vektor `pik`. Süstemaatilise valiku korral on kõigil üldkogumi objektidel sama (konstantne) kaasamistõenäosus $\pi_i = 1/m$ $(i = 1, \ldots, N)$. Funktsioon tagastab eelmises alapeatükis vaadeldud funktsiooniga `srswor()` analoogilise vektori.

Näide 3. Konstantsete kaasamistõenäosustega süstemaatilise valiku teostamine funktsiooniga `UPsystematic(pik)`

Antud näites on üldkogum küla *StatVillage* versioon *Mini village*. Järgmises programmis on fikseeritud samm $m = 9$ ja teostatakse süstemaatiline valik. Eelnevalt on teada, et sel juhul $\pi_i = 1/9$ $(i = 1, \ldots, 480)$ ja $n_s \in \{53, 54\}$.

```r
set.seed(1234)

pik=rep(x=1/9, times=480) # kaasamistõenäosust 1/9 korratakse 480 korda
SYS=UPsystematic(pik) # süstemaatilise valiku teostamine
SYS[1:30] # saadud valim (vektori esimesed 30 elementi)

##  [1] 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0

sum(SYS) # realiseerunud valimimaht on n=54, sest r=2 < c=3

## [1] 54
```

Funktsiooniga `UPsystematic(pik)` saab teostada ka mittevõrdsete kaasamistõenäosustega süstemaatilist valikut. Järgmine lõik, kus kirjeldatakse selle valikumeetodi algoritmi, ja näide põhinevad allikal (Tillé, 2010).

Tähistagu $U = \{1, \ldots, N\}$ lõplikku üldkogumit. Olgu iga üldkogumi objekti $i \in U$ kaasamistõenäosus π_i teada. Kehtigu $\sum_{i=1}^{N} \pi_i = n$, kus n on planeeritav valimimaht. Iga $i \in U$ korral leitakse summa $S_i = \sum_{k=1}^{i} \pi_k$, kusjuures $S_0 = 0$. Seejärel genereeritakse pseudojuhuslik arv u ühtlasest jaotusest $U(0, 1)$ ja algab valimi moodustamine. Valimi esimene element $j_1 \in U$ valitakse selline, et kehtivad võrratused $S_{j_1-1} \leq u < S_{j_1}$. Teine element j_2 valitakse nii, et kehtivad $S_{j_2-1} \leq u + 1 < S_{j_2}$ jne. Üldjuhul peab valimi element j_k rahuldama võrratusi $S_{j_k-1} \leq u + (k-1) < S_{j_k}$. Nii jätkatakse üldkogumist objektide võtmist, kuni saadakse valim mahuga n.

Järgmine näide selgitab algoritmi tööd väikese üldkogumi korral. Näites olevad kaasamistõenäosused on ette antud.

Näide 4. Mittevõrdsete kaasamistõenäosustega süstemaatilise valiku teostamine funktsiooniga `UPsystematic(pik)`

Olgu üldkogumimaht $N = 7$ ja planeeritav valimimaht $n = 3$. Üldkogumi objektide kaasamistõenäosused ja arvutatud summad S_i $(i = 0, \ldots, 7)$ on toodud tabelis 1.

Tabel 1. Üldkogumi objektide kaasamistõenäosused

i	0	1	2	3	4	5	6	7
π_i	0	0,3	0,4	0,5	0,6	0,5	0,4	0,3
S_i	0	0,3	0,7	1,2	1,8	2,3	2,7	3

Olgu $u = 0{,}114$. Siis moodustavad valimi üldkogumi objektid järjekorranumbritega 1, 3 ja 5, sest vastavalt $0 \leq u < 0{,}3$, $0{,}7 \leq u + 1 < 1{,}2$ ja $1{,}8 \leq u + 2 < 2{,}3$. Järgmine programm teostab sama valiku funktsiooniga `UPsystematic(pik)`.

```r
set.seed(1234)

yldkogum=c(1:7) # üldkogumi elementide freim (üldkogumimaht N=7)
pik=c(0.3, 0.4, 0.5, 0.6, 0.5, 0.4, 0.3) # esimest järku kaasamistõenäosused
sum(pik) # planeeritav valimimaht n=3

## [1] 3

SYS2=UPsystematic(pik) # süstemaatilise valiku teostamine
yldkogum[SYS2==1] # valimisse sattunud elementide järjekorranumbrid freimis

## [1] 1 3 5
```

Valimid tulid samad, kuna juhusliku seemne 1234 korral on jaotusest $U(0,1)$ genereeritav esimene pseudojuhuslik arv 0,1137034 $\approx$ 0,114.

1.3.3 Poissoni valik

Poissoni valiku korral otsustab iga üldkogumi objekti valimisse sattumise Bernoulli jaotusega sõltumatu juhusliku katse tulemus. Seejuures võib igal objektil olla erinev kaasamistõenäosus π_i $(i = 1, \ldots, N)$. Poissoni valik on TTA valikumeetod ja selle algoritm on järgmine: iga $i = 1, \ldots, N$ korral genereeritakse pseudojuhuslik arv $u_i \sim U(0,1)$ ja objekt i kaasatakse valimisse, kui $u_i < \pi_i$. Tulemuseks on juhusliku mahuga valim.

Paketis „sampling“ on Poissoni valiku teostamiseks funktsioon `UPpoisson(pik)`, mis sarnaneb eelmises alapeatükis vaadeldud funktsiooniga `UPsystematic(pik)`.

Näide 5. Poissoni valiku teostamine funktsiooniga `UPpoisson(pik)`
Järgmine programm võtab üldkogumist mahuga $N = 15$ valimi Poissoni valiku abil. Esimese viie üldkogumi objekti kaasamistõenäosus on 0,2, järgmise viie objekti korral 0,4 ja viimase viie korral 0,8. Realiseeruv valimimaht n_s ei ole enne valiku teostamist teada, kuid planeeritav valimimaht on hetkel $n = 5{\cdot}0{,}2{+}5{\cdot}0{,}4{+}5{\cdot}0{,}8 = 7$.

```r
set.seed(1234)

# esimest järku kaasamistõenäosused
pik=c(rep(x=0.2, times=5), rep(x=0.4, times=5), rep(x=0.8, times=5))
valim_Po=UPpoisson(pik) # Poissoni valiku teostamine
sum(valim_Po) # realiseerunud valimimaht

## [1] 7

valim_Po  # saadud valimile vastav nullidest ja ühtedest koosnev vektor

##  [1] 1 0 0 0 0 0 1 1 0 0 1 1 1 0 1
```

Juhul kui kõik kaasamistõenäosused π_i $(i = 1, \ldots, N)$ on võrdsed, nimetatakse Poissoni valikut Bernoulli valikuks. Kui n on planeeritav valimimaht, siis sel juhul iga $i = 1, \ldots, N$ korral $\pi_i = n/N$.

1.3.4 Kihtvalik

Kihtvalik on praktikas enim kasutatav valikumeetod. Kihtvaliku korral jaotatakse üldkogumi objektid mõne teadaoleva tausttunnuse (kihistava tunnuse) väärtuste järgi lõikumatuteks osadeks ehk kihtideks. Kihte käsitletakse üksteisest sõltumatute kogumitena, mistõttu erinevates kihtides võib rakendada erinevaid valikumeetodeid.

Kihtvalikut saab teostada paketi „sampling“ funktsiooniga `strata()`. Enne funktsiooni kasutamist tuleb üldkogumi objektide andmestik sorteerida kihistava(te) tunnus(t)e järgi, mille väärtused peavad olema teada kõigi objektide korral enne valimi võtmist. Argumendid, mida funktsioonile saab anda, on järgmised:

- `data` – sorteeritud üldkogumi objektide andmestik (ridade arv on N);
- `stratanames` – kihistav(ad) tunnus(ed);
- `size` – kihtide planeeritavad valimimahud (järjekorra määrab sorteeritud andmestik);
- `method` – valikumeetod, mida rakendatakse igas kihis (`srswor`, `srswr`, `poisson` või `systematic`);
- `pik` – kaasamistõenäosused või lisainformatsioon, millega nad arvutatakse;
- `description` – väärtuse `TRUE` korral väljastatakse iga kihi üldkogumimaht, igast kihist valimisse kaasatud objektide arv, kihtide arv ja realiseerunud valimimaht (vaikimisi `FALSE` ja mainitud arve ei väljastata).

Kohustuslikeks argumentideks on `data`, `stratanames`, `size`, `method` ning Poissoni ja süstemaatilise valiku korral ka `pik`. Kui argumendile `pik` anda kaasamistõenäosusi sisaldava vektori asemel näiteks mõni andmestikus olev abitunnus x, siis kasutatakse funktsiooni `inclusionprobabilities()`, et arvutada selle tunnuse väärtustega võrdelised kaasamistõenäosused. Abitunnuse x kõik väärtused peavad olema positiivsed ja kaasamistõenäosused arvutatakse valemiga

$$\pi_i = \frac{n \cdot x_i}{\sum_{j=1}^{N} x_j},$$

kus $i = 1, \ldots, N$ ja n on planeeritav valimimaht (Traat ja Inno, 1997, 122).

Funktsioon `strata()` tagastab andmestiku, mis sisaldab iga valimisse sattunud objekti kohta järgmist informatsiooni:

- `ID_unit` – järjekorranumber sorteeritud üldkogumi andmestikus;
- `Prob` – esimest järku kaasamistõenäosus;
- `Stratum` – kihistava tunnuse väärtus.

Kui tarkvaras R jooksutada käsku `?strata`, siis selgub, et ka lisapakett „survival“ sisaldab funktsiooni nimega `strata()` (pakett „survival“ laetakse paketiga „survey“ samaaegselt alla). Tagamaks, et järgmises kahes näites kasutab R paketi „sampling“ funktsiooni `strata()`, on funktsiooni ette lisatud paketi nimi ja kolm koolonit.

Näide 6. Lihtsa juhusliku kihtvaliku teostamine funktsiooniga `strata()`

Käesolevas näites on üldkogum andmestik `Yldkogum` ($N = 50$), mis loodi alapeatükis 1.3.1 (vt näide 2 lk 10–11). Kihistavaks tunnuseks on valitud leibkonna perepea sugu ehk tunnus `hmsex` (Schwarz, 1997). Selle tunnuse kaks väärtust on kodeeritud järgmiselt: 1, kui perepea on meessoost, ja 2, kui perepea on naissoost. Enne kihtvaliku teostamist uuritakse mõlema kihi üldkogumimahtu, et välja selgitada, mitu majapidamist igast kihist valimisse võetakse.

```
set.seed(1234)

kihid=table(Yldkogum$hmsex) # kihtide mahud üldkogumis
names(kihid)=c("meessoost", "naissoost")
kihid

## meessoost naissoost
##        41         9
```

Kuna kihtide üldkogumimahud on väga erinevad, siis on mõistlik planeeritavad valimimahud arvutada proportsionaalselt. Näiteks võtta mõlemast kihist valimisse ligikaudu 40% majapidamistest ehk antud juhul esimesest kihist 16 ja teisest kihist 4 majapidamist. Enne lihtsa juhusliku kihtvaliku teostamist sorteeritakse andmestik `Yldkogum` tunnuse `hmsex` järgi.

```
# üldkogumi andmestiku sorteerimine kihistava tunnuse järgi
Yldkogum=Yldkogum[order(Yldkogum$hmsex), ]
# lihtsa juhuslik kihtvaliku teostamine funktsiooniga strata()
kiht=sampling:::strata(data=Yldkogum, stratanames="hmsex", size=c(16,4),
                       method="srswor", description=FALSE)
```

```
kiht$ID_unit # valimisse sattunud majapidamiste järjekorranumbrid andmestikus

##  [1]  1  5  8  9 17 22 24 25 26 31 32 33 34 37 38 39 43 44 48 50

tail(kiht) # funktsiooni strata() poolt tagastatud andmestiku viimased read

##    hmsex ID_unit      Prob Stratum
## 38     1      38 0,3902439       1
## 39     1      39 0,3902439       1
## 43     2      43 0,4444444       2
## 44     2      44 0,4444444       2
## 48     2      48 0,4444444       2
## 50     2      50 0,4444444       2

# valimisse sattunud majapidamiste andmete eraldamine andmestikust Yldkogum
valimKiht=getdata(data=Yldkogum, m=kiht)
```

Viimases programmis kasutati üldkogumist valimisse sattunud 20 majapidamise andmete eraldamiseks ja andmestiku `valimKiht` loomiseks paketi „sampling“ funktsiooni `getdata()` (vt Tillé ja Matei, 2016, 19–20).

Järgmises näites on vaadeldud olukorda, kus kihistavaid tunnuseid on kaks. Parema ülevaate saamiseks on kasutatud väikest üldkogumit, millele vastav andmestik kõigepealt luuakse.

Näide 7. Kihtvaliku teostamine Poissoni valikuga igas kihis

Moodustagu antud näites üldkogumi 15 tallinlast ja 15 tartlast, kes elavad pindalalt kolmes suurimas Tallinna linnaosas ja kahes suurimas Tartu linnaosas. Lisaks on teada nende inimeste viimase kuu sissetulek eurodes. Kirjeldatud üldkogumile vastava andmestiku loomiseks kasutatakse funktsioone `matrix()`, `rbind()` ja `cbind.data.frame()`. Sissetulekute leidmiseks kasutatakse pseudojuhuslikke arve jaotusest $U(0,1)$, mis genereeritakse funktsiooniga `runif()`.

```
set.seed(1234)

# kirjeldatud üldkogumile vastava andmestiku loomine
# esimene tunnus (linn): funktsiooniga matrix() luuakse kaks veeruvektorit ja
# funktsiooniga rbind() paigutatakse need üksteise alla
andmed=rbind(matrix(data=rep("Tallinn"), nrow=15, ncol=1, byrow=TRUE),
             matrix(data=rep("Tartu"), nrow=15, ncol=1, byrow=TRUE))

# teine tunnus (linnaosa): funktsiooniga cbind.data.frame() lisatakse
# andmestikku linnaosa näitav tunnus
# kolmas tunnus (sissetulek): arvutatakse funktsiooniga runif() genereeritud
# pseodujuhuslikke arve kasutades
andmed=cbind.data.frame(andmed, c(rep(1,5),rep(2,5),rep(3,5),rep(1,10),rep(2,5)),
                                500+round(1000*runif(30), 0))
colnames(andmed)=c("linn", "linnaosa", "sissetulek") # tunnustele nimede lisamine
```

```
table("Linnaosa"=andmed$linnaosa, andmed$linn) # kihtide üldkogumimahud

##
## Linnaosa Tallinn Tartu
##        1       5    10
##        2       5     5
##        3       5     0
```

Antud näites on kihistavateks tunnusteks `linn` ja `linnaosa` ning igas kihis rakendatakse Poissoni valikut. Esimest järku kaasamistõenäosuste vektori asemel antakse funktsiooni `strata()` argumendile `pik` tunnuse `sissetulek` väärtuste vektor. Igast kihist võetakse valimisse 60% inimestest (neljandast kihist kuus ja teistest kihtidest kolm inimest).

```
# üldkogumile vastava andmestiku sorteerimine (igas linnas linnaosade järgi)
andmed=andmed[order(andmed$linn, andmed$linnaosa), ]

# kihtvaliku teostamine
kiht2=sampling:::strata(data=andmed, stratanames=c("linn", "linnaosa"),
                 size=c(3,3,3,6,3), method="poisson", pik=andmed$sissetulek)

# realiseerunud valimimahud kihtides (Poissoni valik annab juhusliku valimimahu)
valimikihid=table(kiht2$Stratum)
names(valimikihid)=c("Tallinn1", "Tallinn2", "Tallinn3", "Tartu1", "Tartu2")
valimikihid

## Tallinn1 Tallinn2 Tallinn3   Tartu1   Tartu2
##        4        2        4        6        4

valimKiht2=getdata(data=andmed, m=kiht2) # lõpliku andmestiku loomine
```

Kuna Poissoni valiku korral on saadav valimimaht juhuslik, siis ei realiseerunud planeeritud valimimaht $n = 18$. Lõplik valimimaht tuli $n = 20$.

1.4 Pakett „survey“

Käesolevas alapeatükis esitatud informatsioon paketi „survey“ ja selle funktsioonide kohta on pärit allikast (Lumley, 2016).

Lisapaketi „survey“ autor on Thomas Lumley (Uus-Meremaa, Aucklandi Ülikool) ning antud töös on kasutatud selle paketi versiooni 3.31-5, mis avaldati 1. detsembril 2016. Pakett sisaldab funktsioone, millega saab analüüsida tõenäosuslike valikumeetoditega võetud valimeid. Nende funktsioonide kasutamiseks tuleb pakett kõigepealt arvutisse paigaldada ning seejärel sisse laadida nagu tehti paketiga „sampling“ alapeatükis 1.3.

1.4.1 Lihtne juhuslik kihtvalik

Kuigi paketi „survey“ funktsioonide põhirõhk on üldkogumi parameetrite hinnangute arvutamisel, sisaldab pakett funktsiooni `stratsample()` lihtsa juhusliku kihtvaliku teostamiseks. Selle funktsiooni kaks kohustuslikku argumenti on

- `strata` – kihistav tunnus;
- `counts` – kihtide planeeritavad valimimahud koos sõnedena antud eelmise argumendi vastavate väärtustega (vt näide 8).

Funktsioon `stratsample()` ei nõua, et argumendile `strata` antav vektor oleks sorteeritud. Funktsioon tagastab vektori, mis sisaldab valimisse sattunud objektide järjekorranumbreid argumendile `strata` antud vektoris.

Näide 8. Lihtsa juhusliku kihtvaliku teostamine funktsiooniga `stratsample()`

Käesolevas näites korratakse näites 6 (lk 15–16) tehtut. Seega on üldkogum andmestik `Yldkogum` ja kihistavaks tunnuseks `hmsex`. Esimesest kihist kaasatakse valimisse 16 ja teisest kihist 4 majapidamist. Järgmises programmis funktsiooniga `stratsample()` võetud valim ja näites 6 saadud valim koosnevad samadest majapidamistest, kuna enne valiku teostamist andmestik sorteeritakse kihistava tunnuse järgi ning kasutatakse sama juhuslikku seemet.

```
set.seed(1234)

Yldkogum=Yldkogum[order(Yldkogum$hmsex), ] # kihistava tunnuse järgi sorteerimine

# lihtsa juhusliku kihtvaliku teostamine
# kihistaval tunnusel on kaks võimalikku väärtust: 1=mees ja 2=naine
kiht3=stratsample(strata=Yldkogum$hmsex, counts=c("1"=16, "2"=4))
sort(kiht3) # sorteeritud järjekorranumbrid

##  [1]  1  5  8  9 17 22 24 25 26 31 32 33 34 37 38 39 43 44 48 50

valimKiht3=Yldkogum[kiht3, ] # lõpliku andmestiku loomine

# kas kihtide mahud tulid õiged?
valimikihid=table(valimKiht3$hmsex)
names(valimikihid)=c("meessoost", "naissoost")
valimikihid

## meessoost naissoost
##        16         4
```

Kuna funktsioon `stratsample()` võimaldab teostada vaid lihtsat juhuslikku kihtvalikut, siis tuleks kihtvaliku teostamiseks eelistada paketi „sampling“ funktsiooni `strata()`, mida kirjeldati alapeatükis 1.3.4. Pakett „survey“ ei sisalda

rohkem funktsioone, millega saaks antud bakalaureusetöös vaadeldavate tõenäosuslike valikumeetoditega valimeid võtta.

1.4.2 Valikudisaini objekt

Paketi „survey“ funktsiooniga `svydesign()` saab omavahel siduda olemasoleva valimi ja selle võtmiseks kasutatud valikumeetodi kohta teadaoleva informatsiooni. Selle funktsiooni olulisemad argumendid on

- `id` – klastrid määrav(ad) tunnus(ed), väärtuseks tuleb panna ∼1, kui tegu pole klastervalikuga[1];
- `strata` – kihistav tunnus (vaikimisi on väärtus `NULL` ehk tegu pole kihtvalikuga);
- `fpc` – nn lõpliku üldkogumi parandus;
- `weights` – valimi elementide valikukaalud (pole vaja lihtsa juhusliku valiku TTA korral, kui argument `fpc` on antud);
- `data` – valimile vastav andmestik.

Argumentidele `strata`, `fpc` ja `weights` antavad n-mõõtmelised vektorid ei tohi sisaldada puuduvaid väärtusi (n on valimimaht). Kihtvaliku korral tuleb argumendile `fpc` anda kihtide üldkogumimahte sisaldav n-mõõtmeline vektor. Kui kihte ei ole, siis tuleb anda ainult üldkogumimahtu sisaldav n-mõõtmeline vektor. Funktsioon tagastab valikudisaini objekti (*survey design object*), mis sisaldab kogu informatsiooni olemasoleva valimi kohta ja mida saab kasutada kõigi soovitud hinnangute arvutamisel. Paketis „survey“ kasutatakse sümbolit ∼, et viidata andmestikus `data` olevatele tunnustele.

Näide 9. Lihtne juhuslik valik TTA ja funktsioon `svydesign()`

Järgmises programmis defineeritakse näites 2 (lk 10–11) võetud lihtsale juhuslikule valimile (andmestik `Yldkogum`) vastav valikudisaini objekt. Argumenti `weights` pole hetkel vaja anda, kuna selle puudumise korral eeldab funktsioon, et kasutatud on lihtsat juhuslikku valikut TTA. Sel juhul peab aga olema antud argument `fpc`, sest muidu ei ole üldkogumimaht $N = 480$ teada.

```
# andmestikku üldkogumimahtu sisaldava veeru lisamine (argumendi fpc jaoks)
valimLJV=cbind(Yldkogum, maht=rep(x=480, times=50))

# lihtsale juhuslikule valimile vastava disaini defineerimine
LJV_disain=svydesign(id=~1, fpc=~maht, data=valimLJV)
```

[1] Klastervaliku kohta saab täpsemalt lugeda õpikust (Traat ja Inno, 1997, 137–141).

Defineeritud valikudisaini objektist ülevaate saamiseks saab kasutada käsku `summary(LJV_disain)`.

Näide 10. Lihtne juhuslik kihtvalik ja funktsioon `svydesign()`

Antud näites defineeritakse näites 6 (lk 15–16) loodud andmestikule `valimKiht` vastav valikudisaini objekt. Kuna igas kihis rakendati lihtsat juhuslikku valikut TTA, siis pole ka seekord vaja anda argumenti `weights`, kui argument `fpc` on antud.

```
# funktsiooniga ifelse() lisatakse andmestikku kihtide üldkogumimahtude veerg
# kui hmsex=1, siis maht=41, vastasel juhul maht=9 (teada näitest 6)
valimKiht$maht=ifelse(test=valimKiht$hmsex==1, yes=41, no=9)

# kihistav tunnus Stratum on andmestikus, sest valimi võtmiseks
# kasutati funktsiooni strata()
kiht_disain=svydesign(id=~1, strata=~Stratum, fpc=~maht, data=valimKiht)
```

Viimases kahes näites defineeritud valikudisaini objekte kasutatakse järgmises peatükis, kui paketi „survey“ funktsioonidega hinnatakse üldkogumi kogusummat ja selle standardhälvet.

2 Üldkogumi kogusumma hindamine

Käesolevas peatükis esitatud informatsioon paketi „sampling“ funktsioonide kohta pärineb allikast (Till*é* ja Matei, 2016) ja paketi „survey“ funktsioonide kirjeldused allikast (Lumley, 2016). Hinnangute arvutamisel kasutatavate tunnuste kirjeldused on saadud küla *StatVillage* kodulehelt (Schwarz, 1997).

Antud peatükis on tutvustatud tarkvara R lisapakettide „sampling“ ja „survey“ võimalusi olemasoleva valimi põhjal üldkogumi kogusumma hinnangu ja selle dispersiooni või standardhälbe hinnangu arvutamiseks. Keskendutud on ainult uuritava tunnuse kogusumma hindamisele, kuna teisi parameetreid on võimalik kogusummade kaudu esitada. Näiteks on üldkogumi keskmine, osakaal, suhe ja isegi uuritava tunnuse dispersioon esitatavad kogusummade kaudu (vt Traat ja Inno, 1997, 53–55). Edaspidi eeldatakse, et kõik vaadeldavad valikudisainid on tagasipanekuta.

Selles peatükis on $U = \{1, 2, \ldots, N\}$ lõplik üldkogum ja y uuritav tunnus väärtustega $y_1, y_2, \ldots, y_N$. Üldkogumist on võetud valim s mahuga n. Hinnatav parameeter on uuritava tunnuse üldkogumi kogusumma, mis avaldub kujul

$$t_y = \sum_{i=1}^{N} y_i = \sum\nolimits_U y_i,$$

kus tähis $\sum_U$ tähendab, et summeeritakse üle kõigi üldkogumi objektide ehk $i \in U$.

2.1 Kogusumma Horvitz-Thompsoni hinnang ja selle täpsus

Järgnev alapeatükk põhineb õpikul (Traat ja Inno, 1997, 68–71).

Uuritava tunnuse kogusumma π-hinnanguks ehk Horvitz-Thompsoni hinnanguks nimetatakse π-laiendatud valimiväärtuste summat

$$\hat{t}_{y\pi} = \sum\nolimits_s \frac{y_i}{\pi_i}. \tag{1}$$

Tähis $\sum_s$ tähendab, et summeerimine toimub üle valimi ehk $i \in s$. Toodud hinnang on valikuteoorias väga tähtsal kohal, kuna tegu on kogusumma t_y nihketa hinnanguga mistahes tagasipanekuta valikudisaini korral.

Kui iga $i, j \in U$ korral $\pi_{ij} > 0$, siis kogusumma hinnangu (1) dispersiooni nihketa

hinnang avaldub kujul

$$\hat{V}(\hat{t}_{y\pi}) = \sum\sum_s \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \frac{y_i}{\pi_i} \frac{y_j}{\pi_j}. \tag{2}$$

Fikseeritud valimimahuga n valikudisaini ja tingimuse $\pi_{ij} > 0$ iga $i, j \in U$ korral eelistatakse dispersiooni $V(\hat{t}_{y\pi})$ nihketa hinnangu arvutamiseks kasutada valemit

$$\hat{V}(\hat{t}_{y\pi}) = -\frac{1}{2}\sum\sum_s \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \left(\frac{y_i}{\pi_i} - \frac{y_j}{\pi_j}\right)^2, \tag{3}$$

kuna see hinnang on stabiilsem. Viimast dispersiooni hinnangut nimetatakse Sen-Yates-Grundy hinnanguks. Hinnangud (2) ja (3) ei ole üldjuhul võrdsed.

Järgnevas kahes alapeatükis on vaadeldud pakettide „sampling“ ja „survey“ funktsioone, millega saab arvutada uuritava tunnuse kogusumma hinnangut (1) ja hinnata selle dispersiooni.

2.1.1 Horvitz-Thompsoni hinnang paketis „sampling“

Paketis „sampling“ on kogusumma Horvitz-Thompsoni hinnangu (1) arvutamiseks funktsioon `HTestimator()`. See funktsioon nõuab kahte argumenti ja nendeks on

- `y` – uuritava tunnuse väärtused valimis;
- `pik` – valimi elementide esimest järku kaasamistõenäosused.

Mõlemad argumendid peavad olema n-mõõtmelised vektorid (n on valimimaht). Hinnangu (1) dispersiooni hindamiseks saab kasutada funktsiooni `varHT()`. Argumendid, mida sellele funktsioonile anda tuleb, on järgmised:

- `y` – uuritava tunnuse väärtused valimis;
- `pikl` – valimi elementide teist järku kaasamistõenäosuste maatriks;
- `method` – dispersiooni hinnangu arvutamiseks kasutatav valem.

Vaikimisi on `method=1` ja dispersiooni hindamiseks kasutatakse valemit (2). Kui `method=2`, siis funktsioon arvutab dispersiooni Sen-Yates-Grundy hinnangu (3), mis eeldab fikseeritud valimimahuga disaini.

Funktsioon `varHT()` ja veel kaks hiljem vaadeldavat paketi „sampling“ funktsiooni nõuavad ühe argumendina valimisse kaasatud objektide teist järku kaasamistõenäosuste maatriksit. Tagasipanekuta disaini korral on selle maatriksi

peadiagonaalil elemendid $\pi_{ii} = \pi_i$ $(i \in s)$ ning väljaspool peadiagonaali π_{ij} $(i, j \in s)$. Maatriksi dimensioonideks on $n \times n$.

Kuna ka mõned antud bakalaureusetöös mittevaadeldavad paketi „sampling“ funktsioonid kasutavad argumendina maatriksit `pikl`, siis sisaldab see pakett funktsioone, mis leiavad vajaliku maatriksi keerulisemate valikudisainide korral. Nendeks funktsioonideks on `UPsystematicpi2(pik)`, `UPsampfordpi2(pik)`, `UPtillepi2(pik)` ja `UPmidzunopi2(pik)`. Kuna kõigis järgnevates näidetes pole maatriksi `pikl` defineerimine keeruline, siis vajalikud maatriksid konstrueeritakse „käsitsi“.

Näide 11. Horvitz-Thompsoni hinnangu arvutamine ja selle dispersiooni hindamine paketi „sampling“ funktsioonidega

Käesolevas näites on uuritavaks tunnuseks majapidamise kogusissetulek ühes aastas ehk tunnus `totinch`. Selle tunnuse kogusumma hinnangu (1) arvutamiseks kasutatakse lihtsat juhuslikku valimit TTA, mis võeti esimeses peatükis ja salvestati andmestikku `Yldkogum` (vt näide 2 lk 10–11). Hinnang leitakse funktsiooniga `HTestimator()` ja tulemuse õigsuse kontrollimiseks ka valemiga (1).

Õpiku (Traat ja Inno, 1997, 91) kohaselt avalduvad lihtsa juhusliku valiku TTA esimest ja teist järku kaasamistõenäosused antud näite korral kujul

$$\pi_i = \frac{n}{N} = \frac{50}{480}, \quad \pi_{ij} = \frac{n(n-1)}{N(N-1)} = \frac{50 \cdot 49}{480 \cdot 479},$$

kus $i, j = 1, \ldots, 480$, $i \neq j$.

```
# valimi elementide esimest järku kaasamistõenäosused
pik=rep(x=50/480, times=50) # valimimaht n=50

# uuritava tunnuse (totinch) kogusumma Horvitz-Thompsoni hinnangu arvutamine
HTestimator(y=Yldkogum$totinch, pik=pik)

##          [,1]
## [1,] 41254013

# võrdluseks valemiga (1) "käsitsi" leitud hinnang (tulemus tuleb sama)
sum(Yldkogum$totinch/pik)

## [1] 41254013
```

Järgnevalt defineeritakse teist järku kaasamistõenäosuste maatriks ja funktsiooniga `varHT()` arvutatakse dispersiooni hinnang (2). Kõigis paketi „sampling“ funktsioone kasutavates näidetes väljastatakse dispersiooni hinnangu asemel

standardhälbe hinnang ehk standardviga, et tulemust oleks mugavam võrrelda paketi „survey“ funktsioonidega leitud vastava hinnanguga.

```
# valimi elementide teist järku kaasamistõenäosuste maatriksi defineerimine
abivektor=rep(x=50*49/(480*479), times=50^2) # 250-elemendiline abivektor
pikl=matrix(data=abivektor, nrow=50, ncol=50) # 50 rea ja veeruga maatriks
diag(pikl)=50/480 # loodud maatriksi peadiagonaali asendamine

# dispersiooni hinnangu (2) arvutamine funktsiooniga varHT()
dispersioon=varHT(y=Yldkogum$totinch, pikl=pikl, method=1)
sqrt(dispersioon) # standardhälbe hinnang

## [1] 2138987
```

Eelnevas näites tagastaks funktsioon `varHT()` sama dispersiooni hinnangu ka `method=2` korral, kuna kasutatav valim on saadud lihtsa juhusliku valiku TTA abil. Lihtsa juhusliku valiku TTA korral saab näidata, et dispersiooni hinnangute valemid (2) ja (3) langevad kokku. Selle kontrollimine ei ole raske ülesanne. Valemi (3) lihtsustatud kuju lihtsa juhusliku valiku TTA korral koos tuletuskäiguga on toodud õpikus (Traat ja Inno, 1997, 93) ja valem (2) lihtsustub analoogiliselt.

Antud alapeatükis kirjeldatud kahte funktsiooni saab kasutada iga tagasipanekuta valikumeetodiga võetud valimi korral. Samas ei ole funktsiooni `varHT()` kasutamiseks vajaliku teist järku kaasamistõenäosuste maatriksi leidmine alati lihtne. Seetõttu kasutatakse dispersiooni hinnangu arvutamisel sageli ligikaudseid valemeid, mis ei nõua teist järku kaasamistõenäosuste teadmist.

Fikseeritud valimimahuga disaini korral saab Horvitz-Thomposoni hinnangu (1) dispersiooni ligikaudseks hindamiseks kasutada nn Deville'i meetodit (Deville, 1993). Sel juhul arvutatakse dispersiooni hinnang valemiga

$$\hat{V}_{\text{Deville}}(\hat{t}_{y\pi}) = \frac{1}{1-\sum_s a_i^2} \sum_s (1-\pi_i) \left(\frac{y_i}{\pi_i} - \frac{\sum_s (1-\pi_j) y_j/\pi_j}{\sum_s (1-\pi_j)} \right), \tag{4}$$

kus $a_i = (1-\pi_i)/\sum_s (1-\pi_j)$ $(i \in s)$.

Valemiga (4) hindab dispersiooni paketi „sampling“ funktsioon `varest()`. Hinnangu leidmiseks piisab funktsioonile anda argumendid `Ys` ja `pik`, mis langevad kokku funktsiooni `HTestimator()` argumentidega.

Näide 12. Dispersiooni hindamine funktsiooniga `varest()`

Järgmine programm hindab näites 11 arvutatud tunnuse `totinch` kogusumma Horvitz-Thompsoni hinnangu dispersiooni Deville'i meetodil.

```
# argument pik defineeriti eelmise näite alguses (vt näide 11 lk 23)
dispersioon=varest(Ys=Yldkogum$totinch, pik=pik) # dispersiooni hinnangu (4)
sqrt(dispersioon) # standardhälbe hinnang

## [1] 2138987
```

Väljastatud standardhälbe hinnang on võrdne näites 11 funktsiooniga `varHT()` arvutatud hinnanguga.

2.1.2 Horvitz-Thompsoni hinnang paketis „survey“

Paketi „survey“ korral saab kogusumma Horvitz-Thompsoni hinnangu (1) ja selle standardvea arvutamiseks kasutada funktsiooni `svytotal()`. Selle funktsiooni põhilisteks argumentideks on

- `x` – uuritava tunnuse väärtused valimis;
- `design` – valimile vastav valikudisaini objekt (vt alapeatükk 1.4.2 lk 19–20);
- `na.rm` – väärtuse `TRUE` (vaikimisi `FALSE`) korral jäetakse puuduvad väärtused hindamisel välja.

Kui uuritava tunnuse valimiväärtuste vektor sisaldab puuduvaid väärtusi ja argumendi `na.rm` väärtus on `FALSE`, siis hinnangute asemel tagastatakse puuduvad väärtused `NA`. Argumenti `na.rm` aktsepteerivad kõik antud töös vaadeldavad paketi „survey“ funktsioonid, millega hinnatakse üldkogumi kogusummat. Edaspidi funktsioonide kirjeldamisel argumenti `na.rm` eraldi välja ei tuua. Paketi „sampling“ funktsioonidel analoogilist argumenti pole ja puuduvate väärtuste korral tuleb vastavad valimi elemendid andmestikust eemaldada. Selleks saab kasutada näiteks tarkvara R põhifunktsiooni `subset()`.

Järgnevas näites korratakse näites 11 tehtut paketi „survey“ funktsioonidega.

Näide 13. Horvitz-Thompsoni hinnangu arvutamine ja selle standardhälbe hindamine paketi „survey“ funktsioonidega

Antud näites kasutatav andmestikule `Yldkogum` vastav valikudisaini objekt `LJV_disain` defineeriti esimese peatüki lõpus (vt näide 9 lk 19).

```
# vaikimisi na.rm=FALSE ja hinnangud väljastatakse, sest puuduvaid väärtusi pole
svytotal(x=~totinch, design=LJV_disain)

##            total      SE
## totinch 41254013 2138987
```

Saadud hinnangud on võrdsed paketi „sampling“ funktsioonidega `HTestimator()` ja `varHT()` leitud hinnangutega.

2.2 Horvitz-Thompsoni hinnang kihtvaliku korral

Kihtvaliku korral avaldub uuritava tunnuse kogusumma Horvitz-Thompsoni hinnang summana

$$\hat{t}_{y\pi} = \sum_{h=1}^{H} \hat{t}_{y\pi_h}, \tag{5}$$

kus $\hat{t}_{y\pi_h}$ on uuritava tunnuse kogusumma hinnang (1) h-ndas kihis ($h = 1, \ldots, H$). Hinnangu (5) dispersiooni hinnang arvutatakse valemiga

$$\hat{V}(\hat{t}_{y\pi}) = \sum_{h=1}^{H} \hat{V}(\hat{t}_{y\pi_h}), \tag{6}$$

kus iga $h = 1, \ldots, H$ korral $\hat{V}(\hat{t}_{y\pi_h})$ on hinnangu $\hat{t}_{y\pi_h}$ dispersiooni hinnang (2) või (3). Kihtvalikust ja toodud hinnangutest on põhjalikumalt kirjutatud õpikus (Traat ja Inno, 1997, 124–128). Järgnevas kahes alapeatükis on kirjeldatud funktsioone, millega saab hinnanguid (5) ja (6) leida.

2.2.1 Horvitz-Thompsoni hinnang kihtvaliku korral paketis „sampling“

Kogusumma hinnangu (5) arvutamiseks on paketis „sampling“ funktsioon `HTstrata()`. Selle funktsiooni esimesed kaks argumenti (`y` ja `pik`) langevad kokku alapeatükis 2.1.1 kirjeldatud funktsiooni `HTestimator()` argumentidega. Funktsiooni `HTstrata()` ülejäänud kaks argumenti on kihistav tunnus `strata` ja argument `description`. Kui viimase väärtus on `TRUE` (vaikimisi `FALSE`), siis funktsioon väljastab ka summa (5) liidetavad ehk uuritava tunnuse kogusumma hinnangu (1) igas kihis. Argumendid `y`, `pik` ja `strata` peavad olema n-mõõtmelised vektorid. Dispersiooni hinnangu (6) valemis olevate liidetavate leidmiseks saab kasutada funktsiooni `varHT()` või `varest()`, mida on tehtud järgmises näites.

Näide 14. Horvitz-Thompsoni hinnangu arvutamine ja selle dispersiooni hindamine kihtvaliku korral paketi „sampling“ funktsioonidega

Antud näites on uuritavaks tunnuseks majapidamise kogusissetulek ühes aastas ehk tunnus `totinch`. Hinnangute (5) ja (6) arvutamiseks kasutatakse andmestikku `valimKiht`, mis loodi esimeses peatükis pärast lihtsa juhusliku kihtvaliku teostamist (vt näide 6 lk 15–16). Valimi võtmiseks kasutatud funktsiooni `strata()` rakendamise tulemusena on andmestikus `valimKiht` valimi elementide kaasamistõenäosuste tunnus `Prob` ja kihistav tunnus `Stratum`.

```
# näites 6 oli kihistavaks tunnuseks perepea sugu ehk tunnus hmsex
# kihid on kodeeritud järgmiselt: kiht1=mehed, kiht2=naised

# uuritava tunnuse (totinch) kogusumma hinnangu (5) arvutamine
HTstrata(y=valimKiht$totinch, pik=valimKiht$Prob,
         strata=valimKiht$Stratum, description=TRUE)

## For stratum 1 ,the Horvitz-Thompson estimator is: 3549109
## For stratum 2 ,the Horvitz-Thompson estimator is: 706990,5
## The Horvitz-Thompson estimator is:
##          [,1]
## [1,] 4256099
```

Dispersiooni hinnangu (6) liidetavate arvutamiseks funktsiooniga `varHT()` tuleb iga kihi korral defineerida vastav teist järku kaasamistõenäosuste maatriks ja luua sellesse kihti kuuluvate valimi elementide andmestik. Kui igale kihile vastav andmestik on olemas, siis saab liidetavate leidmiseks kasutada ka funktsiooni `varest()`. Järgmine programm arvutab mõlema funktsiooniga dispersiooni hinnangu esimeses kihis ehk summa (6) esimese liidetava.

```
# esimesele kihile vastava andmestiku loomine
valimKiht1=valimKiht[valimKiht$Stratum==1, ]

# esimese kihi valimimaht on n_1=16 ja üldkogumimaht N_1=41 (vt näide 6 lk 15-16)
# teist järku kaasamistõenäosused (analoogiline näites 11 tehtuga, vt lk 23-24)
abivektor=rep(x=16*15/(41*40), times=16^2)
pikl=matrix(data=abivektor, nrow=16, ncol=16)
diag(pikl)=16/41

# dispersiooni hinnangu leidmine esimeses kihis funktsiooniga varHT()
disp.kiht1=varHT(y=valimKiht1$totinch, pikl=pikl, method=1)
sqrt(disp.kiht1) # standardhälbe hinnang esimeses kihis

## [1] 344784,1

# dispersiooni hinnangu leidmine esimeses kihis funktsiooniga varest()
dev.kiht1=varest(Ys=valimKiht1$totinch, pik=valimKiht1$Prob)
sqrt(dev.kiht1) # standardhälbe hinnang esimeses kihis

## [1] 344784,1
```

Kui eelmises programmis luua teise kihti kuuluvate valimi elementide andmestik ja kaasamistõenäosused ära muuta, siis selgub, et funktsioonidega `varHT()` ja `varest()` arvutatud hinnangud on võrdsed ka teise kihi korral.

2.2.2 Horvitz-Thompsoni hinnang kihtvaliku korral paketis „survey“

Paketi „survey“ korral saab kogusumma hinnangu (5) ja selle standardvea arvutamiseks kasutada funktsiooni `svyby()`. Selle funktsiooni olulisemad argumendid on järgmised:

- `formula` – uuritava tunnuse väärtused valimis;
- `by` – kihistav tunnus;
- `design` – valimile vastav valikudisaini objekt;
- `FUN` – funktsioon, mida rakendatakse uuritavale tunnusele igas kihis.

Järgmises näites lahendatakse näide 14 paketi „survey“ funktsioonidega.

Näide 15. Horvitz-Thompsoni hinnangu arvutamine ja selle standardhälbe hindamine kihtvaliku korral paketi „survey“ funktsioonidega

Hinnangu (5) ja selle standardhälbe hinnangu arvutamiseks rakendatakse uuritavale tunnusele igas kihis funktsiooni `svytotal()`. Andmestikule `valimKiht` vastav valikudisaini objekt `kiht_disain`, mida järgmine programm kasutab, defineeriti esimese peatüki lõpus (vt näide 10 lk 20).

```
# kogusumma hinnangu (5) arvutamine ja selle standardhälbe hindamine
svyby(formula=~totinch, by=~Stratum, design=kiht_disain, FUN=svytotal)

##   Stratum   totinch       se
## 1       1 3549108,6 344784,1
## 2       2  706990,5  91741,7
```

Viimased hinnangud langevad kokku näites 14 arvutatud hinnangutega.

2.3 Suhtehinnang ja selle täpsus

Suhtehinnangu kirjeldamisel on kasutatud õpikut (Traat ja Inno, 1997, 163–166).

Kogusumma t_y suhtehinnang on praktikas levinud hinnangufunktsioon. Suhtehinnang põhineb ühel abitunnusel, mis on positiivselt korreleeritud uuritava tunnusega. Olgu selleks abitunnuseks x, mille väärtus x_i peab olema teada iga valimisse kaasatud objekti $i \in s$ korral. Samuti peab olema teada üldkogumi kogusumma $t_x = \sum_U x_i$. Kogusumma t_y suhtehinnang avaldub kujul

$$\hat{t}_r = t_x \frac{\hat{t}_{y\pi}}{\hat{t}_{x\pi}}, \tag{7}$$

kus $\hat{t}_{y\pi}, \hat{t}_{x\pi}$ on vastavalt kogusummade t_y, t_x Horvitz-Thompsoni hinnangud. Hinnang (7) on ligikaudu nihketa ja selle dispersiooni hinnang arvutatakse valemiga

$$\hat{V}(\hat{t}_r) = \left(\frac{t_x}{\hat{t}_{x\pi}}\right)^2 \sum\sum_s \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \frac{y_i - \hat{R}x_i}{\pi_i} \frac{y_j - \hat{R}x_j}{\pi_j}, \tag{8}$$

kus

$$\hat{R} = \frac{\hat{t}_{y\pi}}{\hat{t}_{x\pi}} \tag{9}$$

on jagatise $R = t_y/t_x$ hinnang.

2.3.1 Suhtehinnang paketis „sampling“

Paketis „sampling“ on kogusumma t_y suhtehinnangu (7) arvutamiseks funktsioon `ratioest()`, mis nõuab järgmisi argumente:

- `y` – uuritava tunnuse väärtused valimis;
- `x` – abitunnuse väärtused valimis;
- `Tx` – abitunnuse üldkogumi kogusumma t_x;
- `pik` – valimi elementide esimest järku kaasamistõenäosused.

Argumendid `y`, `x` ja `pik` peavad olema n-mõõtmelised vektorid. Pakett sisaldab ka funktsiooni `ratioest_strata()`, millega saab arvutada suhtehinnangut (7) kihtvaliku korral. Selleks tuleb funktsioonile anda kõik eespool kirjeldatud argumendid, kuid `Tx` tuleb asendada argumendiga `TX_strata`, mis sisaldab abitunnuse x kogusummat kihtide kaupa. Samuti tuleb anda kihistav tunnus `strata` ning kui lisada `description=TRUE`, siis väljastatakse ka suhtehinnang igas kihis.

Dispersiooni hinnangu (8) leidmiseks saab kasutada funktsiooni `vartaylor_ratio()`, mis kasutab järgmist kolme argumenti:

- `Ys` – uuritava tunnuse väärtused valimis;
- `Xs` – abitunnuse väärtused valimis;
- `pikls` – valimi elementide teist järku kaasamistõenäosuste maatriks.

Funktsioon tagastab hinnangu (9) ja selle dispersiooni hinnangu (Traat ja Inno, 1997, 80)

$$\hat{V}(\hat{R}) = \frac{1}{\hat{t}_{x\pi}^2} \sum\sum_s \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \frac{y_i - \hat{R}x_i}{\pi_i} \frac{y_j - \hat{R}x_j}{\pi_j}. \tag{10}$$

Dispersiooni hinnangu (8) saamiseks tuleb hinnangut (10) korrutada suurusega t_x^2. Kuna väljastatakse ka hinnang $\hat{R}$, mille korrutamine kogusummaga t_x annab suhtehinnangu (7), siis ei ole vaja funktsiooni `ratioest()` eraldi rakendada.

Näide 16. Suhtehinnangu arvutamine ja selle dispersiooni hindamine paketi „sampling“ funktsioonidega

Käesolevas näites on üldkogum andmestik `Yldkogum`. Uuritava tunnuse `totinch` suhtehinnangu (7) ja selle dispersiooni hinnangu (8) arvutamiseks võetakse üldkogumist Bernoulli valim planeeritava mahuga $n = 20$. Abitunnuseks x on valitud tunnus `hhsize` (majapidamises elavate inimeste arv).

```
set.seed(1234) # juhusliku seemne fikseerimine

# üldkogumist Bernoulli valimi võtmine
pik=rep(x=20/50, times=50) # esimest järku kaasamistõenäosused
s=UPpoisson(pik) # Bernoulli valik = võrdsete kaasamistõenäosuste Poissoni valik
valim=Yldkogum[s==1, ] # saadud valimile vastava andmestiku loomine

piks=pik[s==1] # valimi elementide kaasamistõenäosused
summa=sum(Yldkogum$hhsize) # abitunnuse üldkogumi kogusumma

# suhtehinnangu (7) arvutamine funktsiooniga ratioest()
ratioest(y=valim$totinch, x=valim$hhsize, Tx=summa, pik=piks)

## [1] 4434402
```

Tänu elementide sõltumatule valikule avalduvad teist järku kaasamistõenäosused Poissoni valiku korral seosega $\pi_{ij} = \pi_i \pi_j$ $(i, j \in U, i \neq j)$ (Traat ja Inno, 1997, 118). Kuna Poissoni valik on tagasipanekuta valikumeetod, siis iga $i \in U$ korral $\pi_{ii} = \pi_i$. Järgnevalt defineeritakse valimi elementide teist järku kaasamistõenäosuste maatriks ja funktsiooni `vartaylor_ratio()` abil leitakse hinnangud (7) ja (8).

```
# teist järku kaasamistõenäosuste maatriksi defineerimine
# funktsioon outer() leiab kahe vektori (hetkel mõlemad pik) otsekorrutise
pikl=outer(X=pik, Y=pik, FUN="*")
diag(pikl)=pik # peadiagonaali asendamine
pikls=pikl[s==1, s==1] # valimi elementide kaasamistõenäosused

# suhtehinnangu (7) ja selle dispersiooni hinnangu (8) arvutamine
(tulem=vartaylor_ratio(Ys=valim$totinch, Xs=valim$hhsize, pikls))

## $ratio
## [1] 26239,07
##
## $estvar
## [1] 7030664

# tulem$ratio annab suhte R hinnangu (9)
(hinnang=summa*tulem$ratio) # suhtehinnang (7)

## [1] 4434402

# tulem$estvar annab suhte R dispersiooni hinnangu (10)
dispersioon=(summa)^2*tulem$estvar # dispersiooni hinnang (8)
sqrt(dispersioon) # standardhälbe hinnang

## [1] 448110,3
```

Funktsiooniga `ratioest()` arvutatud suhtehinnang (7) on võrdne funktsiooni `vartaylor_ratio()` abil leitud hinnanguga. Seega ei olnud funktsiooni `ratioest()` eraldi kasutamine tõepoolest vajalik.

2.3.2 Suhtehinnang paketis „survey“

Paketiga „survey“ saab suhtehinnangu (7) ja selle standardvea leidmiseks kasutada funktsiooni `svyratio()`. Selle funktsiooni olulisemad argumendid on

- `numerator` – uuritava tunnuse väärtused valimis;
- `denominator` – abitunnuse väärtused valimis;
- `design` – valimile vastav valikudisaini objekt;
- `total` – abitunnuse üldkogumi kogusumma t_x.

Kuna funktsioon tagastab hinnangu $\hat{R}$ ja selle standardvea $\sqrt{\hat{V}(\hat{R})}$, tuleb suhtehinnangu (7) ja selle standardhälbe hinnangu $\sqrt{\hat{V}(\hat{t}_r)}$ saamiseks väljastatud suurusi korrutada kogusummaga t_x.

Näide 17. Suhtehinnangu arvutamine ja selle standardhälbe hindamine paketi „survey“ funktsioonidega

Järgmine programm leiab näites 16 arvutatud hinnangud paketi „survey“ funktsioonidega. Enne hinnangute arvutamist defineeritakse vajalik valikudisaini objekt `Be_disain`.

```
# valimile vastavasse andmestikku üldkogumimahu ja valikukaalude veeru lisamine
# andmestik valim loodi näites 16
valim=cbind(valim, maht=rep(x=50, times=nrow(valim)), kaal=1/piks)

# Bernoulli valimile vastava valikudisaini objekti defineerimine
Be_disain=svydesign(id=~1, fpc=~maht, weights=~kaal, data=valim)

# suhtehinnangu (7) ja selle standardhälbe hinnangu arvutamine
# argument summa defineeriti näites 16
suhe=svyratio(numerator=~totinch, denominator=~hhsize, design=Be_disain,
              total=summa)

(hinnang=summa*suhe$ratio[1,1]) # suhtehinnang (7)

## [1] 4434402

(standardviga=summa*SE(suhe)) # suhtehinnangu (7) standardviga

## totinch/hhsize
##       434669,1
```

Funktsiooni `svyratio()` abil leitud standardhälbe hinnang on väiksem kui vastav funktsiooniga `vartaylor_ratio()` arvutatud hinnang. Hinnangud on erinevad, kuna funktsioon `svyratio()` ei kasuta täpseid teist järku kaasamistõenäosusi, vaid lähendab neid esimest järku kaasamistõenäosustega.

2.4 Kalibreerimishinnang ja selle täpsus

Järgnevate lõikude koostamisel on kasutatud artiklit (Deville ja Särndal, 1992), kui pole viidatud teisiti.

Veel üks hinnangufunktsioon kogusumma t_y hindamiseks, mis kasutab lisainformatsiooni, on kalibreerimishinnang. Kalibreerimismeetod on üheks objektide kaalumismeetodiks, mis kasutab teadaolevat lisainformatsiooni, et leida uued kaalud nii, et kaalutud abitunnuste valimisummad oleksid võrdsed teadaolevate üldkogumi kogusummadega. Neid kaale proovitakse leida nii, et erinevus uute ja esialgsete kaalude vahel oleks võimalikult väike, kuna viimastega arvutatud kogusumma hinnang on nihketa. Seega on kalibreerimishinnang peaaegu nihketa hinnang kogusummale t_y.

Olgu iga valimi elemendi $i \in s$ korral teada abitunnuste väärtuste vektor $\mathbf{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{iJ})^T$, kus J on abitunnuste arv. Samuti eeldatakse, et abitunnuste üldkogumi kogusummade vektor $\mathbf{t}_x = \sum_U \mathbf{x}_i$ on teada. Kogusumma t_y kalibreerimishinnang avaldub kujul

$$\hat{t}_{cal} = \sum\nolimits_s w_i y_i, \tag{11}$$

kus w_i on valimi i-nda elemendi kalibreerimiskaal. Need kaalud leitakse selliselt, et kehtiksid kalibreerimisvõrrandid

$$\sum\nolimits_s w_i \mathbf{x}_i = \mathbf{t}_x \tag{12}$$

ja kaalud ise oleksid mingi kaugusmõõdu suhtes võimalikult lähedased esialgsetele valikukaaludele. Väga levinud on hii-ruut tüüpi kaugusmõõt

$$\frac{(w_i - d_i)^2}{d_i},$$

kus $i \in s$ ja d_i on valimi i-nda elemendi esialgne kaal. Selle kaugusmõõdu korral leitakse kalibreerimiskaalud valemiga $w_i = g_i \cdot d_i$ $(i \in s)$, kus valimi i-nda elemendi

nn g-kaal avaldub kujul (Särndal *et al.*, 1992, 232)

$$g_i = 1 + \left(\mathbf{t}_x - \sum\nolimits_s d_i \mathbf{x}_i\right)^T \left[\sum\nolimits_s d_i \mathbf{x}_i \mathbf{x}_i^T\right]^{-1} \mathbf{x}_i. \tag{13}$$

Kalibreerimishinnangu (11) dispersiooni hinnangu arvutamiseks tagasipanekuta disainide korral saab kasutada järgmist kahte valemit:

$$\hat{V}(\hat{t}_{cal}) = \sum\sum\nolimits_s \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} (d_i e_i)(d_j e_j) \tag{14}$$

ja

$$\hat{V}(\hat{t}_{cal}) = \sum\sum\nolimits_s \frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} (w_i e_i)(w_j e_j), \tag{15}$$

kus e_i ja e_j on vastavalt valimi i-nda ja j-nda elemendi jääk. Siiski on näidatud, et hinnangu (15) omadused on paremad ning seetõttu kasutatakse seda praktikas sagedamini kui hinnangut (14).

2.4.1 Kalibreerimishinnang paketis „sampling“

Paketi „sampling“ korral saab kalibreerimishinnangu (11) ja selle dispersiooni hinnangu arvutamiseks kasutada funktsiooni `calibev()`. Argumendid, mida sellele funktsioonile anda tuleb, on järgmised:

- `Ys` – uuritava tunnuse väärtused valimis;
- `Xs` – abitunnuste väärtused valimis;
- `total` – abitunnuste üldkogumi kogusummade vektor;
- `pikl` – valimi elementide teist järku kaasamistõenäosuste maatriks;
- `d` – valimi elementide esialgsed valikukaalud;
- `g` – valimi elementide g-kaalud;
- `with` – väärtuse `TRUE` korral kasutatakse dispersiooni hinnangu arvutamiseks esialgseid kaale ehk valemit (14), vastasel juhul (vaikimisi) arvutatakse kalibreerimiskaalud valemiga $w_i = g_i \cdot d_i$ $(i \in s)$ ja leitakse hinnang (15).

Vektorid `Ys`, `d` ja `g` peavad olema n-mõõtmelised ning maatriksi `Xs` dimensioonideks on $n \times J$. Argumendi `g` ehk g-kaalude leidmiseks on paketis „sampling“ funktsioon `calib()`, mille põhilisteks argumentideks on

- `Xs`, `d`, `total` – langevad kokku vastavate funktsiooni `calibev()` argumentidega;

- `method` – g-kaalude leidmiseks kasutatav meetod (`linear`, `raking`, `logit` või `truncated`);
- `bounds` – g-kaalude alumine ja ülemine tõke (kohustuslik ainult meetodite `logit` ja `truncated` korral);
- `description` – väärtuse `TRUE` korral väljastakse kõigi kaalude (d, g ja w) karpdiagrammid ja histogrammid (vaikimisi `FALSE` ja jooniseid ei väljastata).

Antud töö viimases kahes näites kasutatakse g-kaalude leidmiseks praktikas enim levinud meetodit `linear`, mis vastab valemile (13).

Näide 18. Kalibreerimishinnangu arvutamine ja selle dispersiooni hindamine paketi „sampling“ funktsioonidega

Antud näites on üldkogum andmestik `Yldkogum` ja uuritavaks tunnuseks `totinch`. Abitunnusteks on valitud `hhsize` (majapidamises elavate inimeste arv) ja `roomh` (tubade arv majapidamises). Hinnangute (11) ja (15) arvutamiseks kasutatakse üldkogumist võetud Poissoni valimit planeeritava mahuga $n = 20$. Vajalike esimest järku kaasamistõenäosuste arvutamiseks kasutatakse tunnuse `nuirh` (sissetuleku saajate arv majapidamises) väärtusi.

```
set.seed(1234) # juhusliku seemne fikseerimine

# Poissoni valiku teostamine
pik=inclusionprobabilities(Yldkogum$nuirh, 20) # esimest järku kaasamistõenäosused
s=UPpoisson(pik) # valimi võtmine
piks=pik[s==1] # valimi elementide kaasamistõenäosused

# teist järku kaasamistõenäosuste maatriksi defineerimine
pikl=outer(X=pik, Y=pik, FUN="*")
diag(pikl)=pik
pikls=pikl[s==1, s==1]

X=cbind(Yldkogum$hhsize, Yldkogum$roomh) # abitunnuste maatriksi defineerimine
summad=colSums(X) # abitunnuste üldkogumi kogusummad
Ys=Yldkogum$totinch[s==1] # uuritava tunnuse väärtused valimis
Xs=X[s==1, ] # abitunnuste väärtused valimis

# valimi elementide g-kaalude leidmine funktsiooniga calib()
g=calib(Xs, d=1/piks, total=summad, method="linear", description=FALSE)

# kalibreerimishinnangu (11) ja selle dispersiooni hinnangu (15) arvutamine
(kalib=calibev(Ys, Xs, total=summad, pikl=pikls, d=1/piks, g, with=FALSE))

## $calest
## [1] 4233801
##
## $evar
## [1] 45989047659
```

```
sqrt(kalib$evar) # standardhälbe hinnang

## [1] 214450,6
```

Antud alapeatükis vaadeldud funktsioon `calibev()` kasutab dispersiooni hinnangu leidmiseks teist järku kaasamistõenäosuste maatriksit. Keerulisemate disainide korral võib selle maatriksi defineerimine olla raskendatud. Järgnevalt on kirjeldatud paketi „survey“ funktsioone, millega saab kalibreerimishinnangut ja selle standardviga leida tõenäosusi π_{ij} $(i, j \in s)$ kasutamata.

2.4.2 Kalibreerimishinnang paketis „survey“

Kalibreerimishinnangu (11) ja selle standardvea arvutamiseks paketi „survey“ funktsioonidega tuleb kõigepealt esialgne valikudisaini objekt ümber kaaluda ja täiendada seda üldkogumi kohta teadaoleva lisainformatsiooniga. Seda teeb funktsioon `calibrate()`, mille olulisemad argumendid on

- `design` – valimile vastav valikudisaini objekt;
- `formula` – kasutatavad abitunnused (vt näide 19);
- `population` – abitunnuste üldkogumi kogusummade vektor;
- `calfun` – kalibreerimiskaalude leidmiseks kasutatav meetod (`linear`, `raking` või `logit`);
- `bounds` – kalibreerimiskaalude alumine ja ülemine tõke (kohustuslik ainult `logit` meetodi korral).

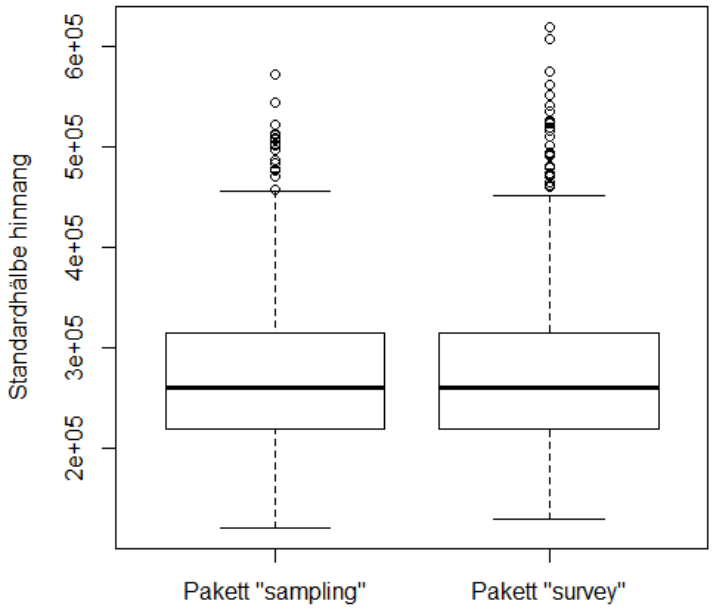
Funktsioon `calibrate()` tagastab uue valikudisaini objekti, millele saab rakendada alapeatükis 2.1.2 kirjeldatud funktsiooni `svytotal()`, et leida kogusumma t_y kalibreerimishinnang (11) ja selle standardviga.

Näide 19. Kalibreerimishinnangu arvutamine ja selle standardhälbe hindamine paketi „survey“ funktsioonidega

Käesolevas näites on uuritav tunnus ja lisainformatsioonina kasutatavad tunnused samad, mis olid näites 18. Järgmises programmis luuakse kõigepealt näites 18 võetud Poissoni valimile vastav andmestik, millele lisatakse üldkogumimahtu ja valikukaale sisaldavad veerud. Seejärel defineeritakse esialgne valikudisaini objekt, millele rakendatakse funktsiooni `calibrate()`. Argumendi `formula` väärtuse lõpus on arv -1, sest vastasel juhul lisatakse abitunnuste maatriksile ühtedest koosnev abitunnus. Soovitud hinnangud arvutatakse funktsiooniga `svytotal()`.

```r
# valimile vastava andmestiku loomine (vektor s on näitest 18)
valim=Yldkogum[s==1, ]

# üldkogumimahtu ja valikukaale sisaldavate veergude lisamine
valim=cbind(valim, maht=rep(50, nrow(valim)), kaal=1/piks)

# Poissoni valimile vastava valikudisaini objekti defineerimine
Po_disain=svydesign(id=~1, fpc=~maht, weights=~kaal, data=valim)

# esialgse valikudisaini objekti ümberkaalumine (vektor summad on näitest 18)
kalib_disain=calibrate(design=Po_disain, formula=~hhsize+roomh-1,
                       population=summad, calfun="linear")

# kalibreerimishinnangu (11) ja selle standardhälbe hinnangu arvutamine
svytotal(x=~totinch, design=kalib_disain)

##           total     SE
## totinch 4233802 206237
```

On näha, et funktsioon `svytotal()` ei kasuta standardhälbe hindamisel täpseid teist järku kaasamistõenäosusi, vaid lähendab neid esimest järku kaasamistõenäosustega. Seetõttu ei ole pakettide „sampling“ ja „survey“ funktsioonidega arvutatud standardhälbe hinnangud alati võrdsed, mis kajastus viimases neljas näites.



Joonis 1. Pakettide „sampling“ ja „survey“ funktsioonidega arvutatud standardhälbe hinnangute karpdiagrammid

Uurimaks, kui palju erinevad näidetes 18 ja 19 kasutatud funktsioonidega arvutatud standardhälbe hinnangud, on läbi viidud simulatsioon. Andmestikust `Yldkogum` võeti 1000 Poissoni valimit planeeritava mahuga $n = 20$. Iga valimi korral arvutati mõlema paketi funktsioonidega tunnuse `totinch` kogusumma

kalibreerimishinnangu (11) standardhälbe hinnang analoogiliselt näidetega 18 ja 19. Arvutatud hinnangute karpdiagrammid on toodud joonisel 1 (lk 36). Simulatsiooni läbiviimiseks ja joonise tegemiseks kasutatud tarkvara R programm on toodud lisas 3.

Jooniselt 1 (lk 36) on näha, et pakettide „sampling“ ja „survey“ funktsioonidega arvutatud kalibreerimishinnangu standardhälbe hinnangute karpdiagrammid on väga sarnased. See tähendab, et keskmiselt lähendab funktsioon `svytotal()` Poissoni valiku korral teist järku kaasamistõenäosusi hästi. Täpsemate järelduste tegemiseks tuleks pakettide „sampling“ ja „survey“ funktsioonide erinevusi uurida ka teiste valikudisainide ja valimimahtude korral.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli anda ülevaade statistikatarkvara R lisapakettide „sampling“ ja „survey“ funktsioonidest, millega saab teostada levinumaid tõenäosuslikke valikuid ja hinnata üldkogumi kogusummat ning selle dispersiooni või standardhälvet. Vaadeldud valikumeetoditeks olid lihtne juhuslik valik (TTA ja TGA), süstemaatiline valik (klassikaline ja mittevõrdsete kaasamistõenäosustega), Poissoni valik ja kihtvalik. Iga töös kirjeldatud funktsiooni kasutamise kohta toodi vähemalt üks praktiline näide, kus vastava funktsiooni rakendamist selgitati. Seejuures lahendati igal võimalusel sama näide nii paketi „sampling“ kui ka paketi „survey“ funktsioonidega, et tuvastada võimalikke erinevusi kahe paketi vahel. Enamikes näidetes oli üldkogum Kanadas asuv hüpoteetiline küla *StatVillage* või selle osakogum.

Töö käigus selgus, et erinevate tõenäosuslike valikute teostamiseks on paketis „sampling“ rohkem võimalusi. Paketi „survey“ põhirõhk on olemasolevate valimite analüüsimisel ja selles paketis on ainult üks funktsioon, millega saab tõenäosuslikke valimeid võtta. Selleks on lihtsat juhuslikku kihtvalikut teostav funktsioon `stratsample()`. Siiski on kihtvaliku teostamiseks mugavam kasutada paketi „sampling“ funktsiooni `strata()`, kuna sellega saab kihtides rakendada ka teisi valikumeetodeid. Samuti lihtsustab funktsiooni `strata()` kasutamine kogusumma hindamist, kuna valimi võtmisel tagastatav andmestik sisaldab valimi elementide kaasamistõenäosusi ja nende kihte määravat tunnust.

Üldkogumi kogusumma hinnangu ja selle dispersiooni hinnangu arvutamiseks kasutatud paketi „sampling“ funktsioonide dokumentatsioon on põhjalikum kui vastavate paketi „survey“ funktsioonide kirjeldused, kus ei ole välja toodud kõiki kasutatavaid valemeid. Paketi „sampling“ funktsioonid kasutavad hinnangute arvutamiseks valemeid, mida käsitletakse kursuses „Valikuuringute teooria I“.

Paketi „survey“ funktsioonid ei kasuta kogusumma hinnangu standardvea arvutamisel teist järku kaasamistõenäosusi, vaid lähendavad neid esimest järku kaasamistõenäosustega. Seetõttu erinesid neljas näites kahe paketi funktsioonidega leitud standardvead. Kalibreerimishinnangu standardvigade võrdlemiseks Poissoni valiku korral viidi läbi simulatsioon. Mõlema paketi funktsioonidega arvutati 1000 standardhälbe hinnangut ja neid võrreldes selgus, et paketi „survey“ funktsioon `svytotal()` lähendab Poissoni valiku korral teist järku kaasamistõenäosusi hästi. Vaatamata sellele ei ole täpselt teada, kui hästi lähendavad paketi „survey“

funktsioonid valimi elementide teist järku kaasamistõenäosusi muude valikudisainide ja valimimahtude korral. Selle uurimiseks tuleks läbi viia rohkem simulatsioone.

Autor loodab, et töös esitatud näiteprogrammide abil on lihtsam koostada juhendit, mis tutvustaks tarkvara R võimalusi valikuuringute valdkonnas ja mida saaks kasutada kursuse „Valikuuringute teooria I“ praktikumides.

Kasutatud kirjandus

Deville, J.-C. (1993). *Estimation de la variance pour les enquetes en deux phases.* Paris: INSEE.

Deville, J.-C., Särndal, C.-E. (1992). Calibration Estimators in Survey Sampling. *Journal of the American Statistical Association*, **87** (418), 376–382.

Lumley, T. (2016). Package 'survey'. *Analysis of Complex Survey Samples.* Kasutatud 23.04.2017 https://cran.r-project.org/web/packages/survey/survey.pdf

Schwarz, C. J. (1997). StatVillage: An On-Line, WWW-Accessible, Hypothetical City Based on Real Data for Use in an Introductory Class in Survey Sampling. *Journal of Statistics Education*, **5** (2). Kasutatud 21.04.2017 http://www.lenato.eu/StatVillage/index.html

Särndal, C.-E., Swensson, B., Wretman, J. (1992). *Model Assisted Survey Sampling.* New York: Springer-Verlag.

Tillé, Y. (2010). Algorithms of sampling with equal or unequal probabilities. *International Statistical Seminar XXIII.* Basque Statistical Institute (EUSTAT). November 2010. Kasutatud 07.05.2017 http://www.eustat.eus/productosServicios/52.1_Unequal_prob_sampling.pdf

Tillé, Y., Matei, A. (2016). Package 'sampling'. *Survey Sampling.* Kasutatud 23.04.2017 https://cran.r-project.org/web/packages/sampling/sampling.pdf

Traat, I., Inno, J. (1997). *Tõenäosuslik valikuuring.* Tartu: Tartu Ülikooli kirjastus.

Lisad

Lisa 1. Küla *StatVillage* freimi loomine

Järgmine programm loob küla *StatVillage* versioonile *Mini village* vastava üldkogumi freimi.

```
# küla StatVillage versiooni Mini village freimi loomine
# kokku on 60 plokki ja igas plokis on 8 maja (üldkogumimaht on 480)
freim=data.frame(number=1:480, plokk=rep(1:60, each=8), maja=rep(1:8, times=60))
```

Loodud andmestiku kuju on järgmine.

```
freim[1:10, ] # üldkogumi freimi esimesed 10 rida

##    number plokk maja
## 1       1     1    1
## 2       2     1    2
## 3       3     1    3
## 4       4     1    4
## 5       5     1    5
## 6       6     1    6
## 7       7     1    7
## 8       8     1    8
## 9       9     2    1
## 10     10     2    2
```

Lisa 2. Küla *StatVillage* andmete sisselugemine

Kui oled küla *StatVillage* versiooni *Mini village* üldkogumi freimist võtnud valimi, siis valimisse sattunud majapidamiste andmete hankimiseks teosta järgmised sammud.

1. Vali küla *StatVillage* kodulehel versioon *Mini village*.
2. Avanevas ruudustikus märgista valimisse sattunud majadele vastavad ruudud.
3. Vajuta nupule „Get the sample units“, kopeeri kõik saadavad andmeread tekstifaili ning salvesta.
4. Kasuta alltoodud programmi, et tekitada võetud valimile vastav tarkvara R andmestik.

```
# Sammul 3 loodud tekstifailist on näha, et puuduvate väärtuste tähistamiseks
# on kasutatud punkti ja tunnuste väärtused on üksteisest eraldatud tühikutega.
andmed=read.table(file="<faili_asukoht>", header=FALSE, sep="",
                  na.strings=".", colClasses="character")

# Samuti on näha, et 13 tunnuse väärtused on antud ühe arvuna (veerg 4).
# Tagamaks, et R arvu alguses olevaid nulle ära ei kaotaks, loetakse väärtused
# sisse sõnedena. Seejärel eraldatakse väärtused ning moodustatakse uus
# andmestik, kus igale veerule vastab üks mõõdetud tunnus.
abiks=t(as.data.frame(strsplit(andmed$V4, ""))) # väärtuste eraldamine
rownames(abiks)=NULL # korralike ridade nimede (1 kuni n) lisamine
andmestik=cbind(andmed[, 1:3], abiks, andmed[, 5:36]) # uue andmestiku loomine

# tunnustele ingliskeelsete nimede lisamine ja väärtuste teisendamine arvudeks
colnames(andmestik)=c("block", "unit", "hhsize", "hhpera",
"hhperb1", "hhperb2", "hhperd1", "hhperd2", "hhpere1", "hhpere2",
"hhperf1", "hhperf2", "hhperg1", "hhperg2", "hhperh1", "hhperh2",
"nuempinh", "nuirh", "empinch", "invsth", "govinch", "otinch",
"totinch", "dtypeh", "builth", "tenurh", "morgh", "roomh", "broomh",
"valueh", "grosrth", "omph", "hmage", "hmsex", "hmmtn", "hmhlos",
"hmocc91", "hmlfact", "hmwkswk", "hmempin", "shmage", "shmsex",
"shmmtn", "shmhlos", "shmocc91", "shmlfact", "shmwkswk", "shmempin")

for (i in 1:48) { # läbi tuleb käia kõik 48 veergu
  andmestik[, i]=as.numeric(as.character(andmestik[, i]))
}
str(andmestik) # kontroll (kas kõik tunnused arvulised)
```

Lisa 3. Kalibreerimishinnangu standardhälbe hinnangute võrdlemise simulatsioon

Järgmine programm kordab näidetes 18 ja 19 tehtut 1000 korda.

```
set.seed(1234) # juhusliku seemne fikseerimine

# tühjad vektorid, kuhu salvestatakse arvutatud standardhälbe hinnangud
sampling=rep(NA, times=1000)
survey=rep(NA, times=1000)

pik=inclusionprobabilities(Yldkogum$nuirh, 20) # kaasamistõenäosused
X=cbind(Yldkogum$hhsize, Yldkogum$roomh) # abitunnuste maatriks
summad=colSums(X) # abitunnuste üldkogumi kogusummad

for (i in 1:1000) {
  s=UPpoisson(pik) # Poissoni valiku teostamine
  pikl=outer(X=pik, Y=pik, FUN="*") # teist järku kaasamistõenäosuste maatriks
  diag(pikl)=pik

  piks=pik[s==1] # valimi elementide kaasamistõenäosused
  pikls=pikl[s==1, s==1] # valimi elementide teist järku kaasamistõenäosused
  Xs=X[s==1, ] # abitunnuste väärtused valimis
  Ys=Yldkogum$totinch[s==1] # uuritava tunnuse väärtused valimis

  # standardhälbe hinnangu arvutamine paketi "sampling" funktsioonidega
  g=calib(Xs, d=1/piks, total=summad, method="linear", description=FALSE)
  kalib_samp=calibev(Ys, Xs, total=summad, pikl=pikls, d=1/piks, g, with=FALSE)

  # standardhälbe hinnangu arvutamine paketi "survey" funktsioonidega
  valim=Yldkogum[s==1, ]
  valim=cbind(valim, maht=rep(50, nrow(valim)), kaal=1/piks)
  Po_disain=svydesign(id=~1, fpc=~maht, weights=~kaal, data=valim)
  kalib_disain=calibrate(design=Po_disain, formula=~hhsize+roomh-1,
                         population=summad, calfun="linear")
  hinnangud=svytotal(x=~totinch, design=kalib_disain)

  # standardhälvete salvestamine loodud vektoritesse
  sampling[i]=sqrt(kalib_samp$evar)
  survey[i]=SE(hinnangud)
}

# karpdiagrammide joonistamine
boxplot(sampling, survey, ylab="Standardhälbe hinnang",
        names=c('Pakett "sampling"', 'Pakett "survey"'))
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Sören Mirski,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Tõenäosuslik valikuuring tarkvara R pakettide „sampling“ ja „survey“ abil“, mille juhendaja on Natalja Lepik,

    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, **09.05.2017**