

LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS ANALĪZES KATEDRA

**DAUDZDIMENSIJU KLĀSTERU IZLAŠU DIZAINU SUMMĀRO UN
ATTIECĪBAS TIPĀ UN TO IZMAIŅU STATISTIKU KVALITĀTES
RĀDĪTĀJU NOVĒRTĒJUMU FUNKCIJU IZSTRĀDE PROGRAMMĀ R
UN TO PIELIETOJUMS EU-SILC APSEKOJUMĀ**

BAKALaura DARBS

Autors: Santa Ivanova

Studenta apliecības Nr.: si10007

Darba vadītājs: Mg. math. Juris Breidaks

RĪGA 2015

Anotācija

Katrā Eiropas Savienības dalībvalstī katru gadu tiek īstenots EU-SILC apsekojums jeb "Eiropas Savienības statistika par ienākumiem un dzīves apstākļiem". EU-SILC nodrošina šķērsriezuma datus.

Darba mērķis ir novērtēt šķērsriezuma rādītāju un to izmaiņu kvalitātes rādītājus, kad tie ir summāri un attiecību tipa. Kvalitātes rādītāju novērtēšanai šķērsriezuma rādītājiem un to izmaiņām tika izstrādātas funkcijas programmā *R vardcros* un *vardchanges*. Funkcijas balstās uz izstrādāto teoriju, kas tiek īstenota pēc Osiera un Bergera metodoloģijas, kur tiek konstruēts daudzfaktoru lineārās regresijas modelis un tiek izmantoti novērtētie atlikumi.

Atslēgas vārdi: šķērsriezums, summārie rādītāji, attiecību tipa rādītāji, šķērsriezuma rādītāju izmaiņas, EU-SILC apsekojums, R, vardcros, vardchanges.

Abstract

EU-SILC survey or „European Union Statistics on Income and Living Conditions” is an annual survey in each country of European Union. EU-SILC provides cross-sectional data.

The aim of this diploma paper is to evaluate cross-sectional indicators and its qualities of change when indicators are total and ratio. The functions, *vardcros* and *vardchanges* in programme *R*, were worked out to evaluate the indicator of quality, cross-sectional indicators and its changes. The functions are found on the developed theory, which is based on methodology by Osier and by Berger, where multivariate linear regression model is constructed and estimated residuals are used.

Key words: Cross-sectional, total, ratio, measures, changes, EU-SILC survey R, *vardcros*, *vardchanges*.

SATURS

IEVADS	6
1. EIROPAS SAVIENĪBAS STATISTIKA PAR IENĀKUMIEM UN DZĪVES APSTĀKĻIEM.....	7
1.1. Rotējošā paneļa apsekojums	7
1.2. EU-SILC dati	8
2. ŠĶĒRSGRIEZUMA RĀDĪTĀJU DISPERSIJAS NOVĒRTĒJUMS	9
2.1. Galīgā klāsteru metode	9
2.2. Summārā tipa statistika.....	9
2.3. Attiecību tipa statistika EU-SILC šķērsgriezuma rādītājam.....	11
2.4. Attiecību tipa statistika	12
3. ŠĶĒRSGRIEZUMA RĀDĪTĀJU IZMAIŅU DISPERSIJAS NOVĒRTĒJUMS.....	16
3.1. Summārā tipa statistika.....	16
3.2. Attiecību tipa statistika	19
4. ŠĶĒRSGRIEZUMA RĀDĪTĀJU KVALITĀTES RĀDĪTĀJU NOVĒRTĒJUMI PROGRAMMĀ <i>R</i> UN TO PIELIETOJUMS EU-SILC DATIEM.....	23
4.1. Funkcijas <i>vardcros</i> apraksts programmā <i>R</i>	23
4.2. Funkcijas <i>vardcros</i> programmā <i>R</i> testēšanas rezultāti uz simulatīviem datiem	24
4.3. Funkcijas <i>vardcros</i> programmā <i>R</i> testēšanas rezultāti uz reāliem EU-SILC Latvijas datiem.....	25
5. ŠĶĒRSGRIEZUMA RĀDĪTĀJU IZMAIŅU KVALITĀTES RĀDĪTĀJU NOVĒRTĒJUMI PROGRAMMĀ <i>R</i> UN TO PIELIETOJUMS EU-SILC DATIEM.....	28
5.1. Funkcijas <i>vardchanges</i> apraksts programmā <i>R</i>	28
5.2. Funkcijas <i>vardchanges</i> programmā <i>R</i> testēšanas rezultāti uz simulatīviem datiem....	28
5.3. Funkcijas <i>vardchanges</i> programmā <i>R</i> testēšanas rezultāti uz reāliem EU-SILC Latvijas datiem.....	30
SECINĀJUMI	32
IZMANTOTĀ LITERATŪRA UN AVOTI	33
PIELIKUMI.....	35
1. pielikums. Funkcijas <i>vardcros</i> mainīgo apraksts programmā <i>R</i>	35
2. pielikums. Funkcijas <i>vardcros</i> testēšanai sagatavotie simulatīvie <i>eusilc</i> dati šķērsgriezuma summārā un attiecību tipa rādītājiem.....	37
3. pielikums. Funkcijas <i>vardchanges</i> mainīgo apraksts programmā <i>R</i>	41
4. pielikums. Funkcijas <i>vardchanges</i> testēšanai sagatavotie simulatīvie <i>eusilc</i> dati šķērsgriezuma summārā un attiecību tipa rādītāju izmaiņām.....	43

5. pielikums. Funkcijas <i>vardcros</i> R programmas kods.....	47
6. pielikums. Funkcijas <i>vardchanges</i> R programmas kods.....	64

IEVADS

EU-SILC jeb "Eiropas Savienības statistika par ienākumiem un dzīves apstākļiem" ir ikgadējs apsekojums, kas tiek veikts visās Eiropas Savienības valstīs, kā arī Šveicē, Norvēģijā, Islandē un Turcijā. EU-SILC apsekojuma metodoloģiju un veikšanu koordinē EUROSTAT. EU-SILC apsekojums izmanto rotējošo paneli, kas sniedz precīzu parametru novērtējumu laikā un nodrošina šķērsriezuma datus. Šajā darbā ir raksturoti šķērsriezuma datu rādītāji, tādi kā nabadzības vai sociālās atstumtības risks, kuram ir trīs apakšrādītāji tādi kā nabadzības risks, dziļa materiālā nenodrošinātība un personas, kuru vecums ir mazāks par 60 gadiem, nodarbinātas darbā ar zemu intensitāti.

Darba mērķis ir novērtēt šķērsriezuma rādītāju un to izmaiņu kvalitātes rādītājus, kad rādītāji ir summāri un attiecību tipa.

Mērķa sasniegšanai tiek izvirzīti uzdevumi tādi kā, apgūt teoriju par šķērsriezuma rādītājiem un šķērsriezuma rādītāju izmaiņām, kad rādītāji ir summāri un attiecību tipa, saskaņā ar teoriju izstrādāt funkcijas programmā *R*, kā arī testēt izstrādātās funkcijas uz simulatīviem un reāliem Latvijas datiem.

Darbā ir iekļautas piecas nodaļas. Pirmajā nodaļā ir raksturots EU-SILC apsekojums, tā būtība, datu veidi un rotējošā paneļa modelis. Otrajā un trešajā nodaļā ir definēta galīgā klāsteru metode, šķērsriezuma summārie un attiecību tipa rādītāji, un rādītāju izmaiņas. Šķērsriezuma un šķērsriezuma rādītāju izmaiņu kvalitātes rādītāju novērtēšanai tiek definētas divas metodoloģijas, Osiera un Bergera. Metodoloģijas balstās uz novērtētiem atlikumiem, kas tiek iegūti no daudzfaktoru lineārās regresijas. Ceturtajā nodaļā ir aprakstīti simulatīvie *eusilc* dati un izstrādātās funkcijas *vardcros* mainīgie. Ir veikta rezultātu analīze simulatīviem un reāliem Latvijas datiem 2011., 2012. un 2013. gadam personu un mājsaimniecību līmenī, kā arī ir rezultātu apskats funkcijai *vardom*. Piektajā nodaļā ir izstrādātās funkcijas *vardchanges* mainīgo apraksts un rezultātu analīze simulatīviem *eusilc* datiem un reāliem Latvijas EU-SILC datiem personu un mājsaimniecību līmenī 2011 un 2012 gadam, un 2012 un 2013 gadam.

1. EIROPAS SAVIENĪBAS STATISTIKA PAR IENĀKUMIEM UN DZĪVES APSTĀKĻIEM

„Eiropas Savienības statistika par ienākumiem un dzīves apstākļiem” jeb EU-SILC ir ikgadējs apsekojums, kas tiek veikts visās ES dalībvalstīs, kā arī Šveicē, Norvēģijā, Islandē un Turcijā.

EU-SILC galvenais mērķis ir iegūt datu salīdzināmību starp dalībvalstīm, kas tiek sasniegts, veicot metodoloģiskos pētījumus EU-SILC datu apkopošanas sākumā. Tiek iegūti salīdzināmi un savlaicīgi šķērsriezuma dati par ienākumiem, nabadzības līmeni, un sociālo atstumtību. Katra EU-SILC dalībvalsts izmanto citu izlases dizainu. Apsekojuma veikšanu un metodoloģiju koordinē EUROSTAT. EU-SILC apsekojuma pamatvienība ir personas, kuras dzīvo privātās mājsaimniecībās. Latvijas EU-SILC apsekojuma izlasē netiek iekļautas kolektīvās mājsaimniecības (vecu ļaužu pansionāti, bērnu invalīdu internāti, studentu kopmītnes, viesnīcas, kazarmas, slimnīcas, sanatorijas, brīvības atņemšanas iestādes u.tml.). Apsekojumā tiek iegūta informācija gan par visu mājsaimniecību kopā, gan arī par katru mājsaimniecības locekli, kam ir 16 gadi un vairāk. Latvijā tiek izmantots četru gadu rotācijas panelis, un izlases personas ir visi mājsaimniecības locekļi, kas pirmajā apsekojuma gadā ir vismaz 14 gadus veci. [8], [13]

1.1. Rotējošā paneļa apsekojums

EU-SILC gadījumā, rotējošā paneļa apsekojums ir ikgadējs paneļa apsekojums, kur katrā gadā mājsaimniecības no izlases tiek atlasītas un aizstātas ar mājsaimniecībām, kuras bija apsekojumā četrus gadus pēc kārtas. Pirmajā vilnī ir četras apakšizlases. Katrā nākamajā vilnī, kāda viena apakšizlase tiek izņemta un tiek aizvietota ar citu.

Izlase vilnī t tiek apzīmēta kā s_t . Rotējošā paneļa apsekojums tiek izmantots šķērsriezuma rādītāju aprēķināšanā. Rotējošā paneļa galvenā priekšrocība ir precīzi parametru izmaiņu novērtējumi laikā, bet galvenie trūkumi ir augsta respondentu noslodze un pārklājumu kļūdu pieaugums. [8], [13]

2. ŠĶĒRSGRIEZUMA RĀDĪTĀJU DISPERSIJAS NOVĒRTĒJUMS

Ir 2 veidu šķērsgriezuma rādītāji, tādi kā summāra un attiecību tipa. Šķērsgriezuma dispersijas novērtēšanai tiek izmantota galīgā klāsteru metode un daudzfaktoru lineārās regresijas modelis.

2.1. Galīgā klāsteru metode

Galīgā klāsteru metode ir universāla metode, kas tiek pielietota dažādu dizainu dispersijas novērtēšanai.

Tiek pieņemts, ka $n_h \geq 2$, visām stratām, t.i., divas vai vairākas primārās izlases vienības (PIV) tiek atlasītas no katras stratas, gadījumā, ja $n_h < 2$, tad ir jāapvieno strata ar līdzīgu stratu, tā, lai tiktu izpildīts nosacījums $n_h \geq 2$, tad var tikt izmantots dispersijas novērtējums:

$$\hat{V}(\hat{\theta}) = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{i=1}^{n_h} (\theta_{hi\cdot} - \bar{\theta}_{h\cdot\cdot})^2, \quad (2.1.1)$$

kur $\theta_{hi\cdot} = \sum_{j=1}^{m_{hi}} w_{hij} \cdot \theta_{hij}$ un $\bar{\theta}_{h\cdot\cdot} = \frac{(\sum_{i=1}^{n_h} \theta_{hi\cdot})}{n_h}$.

- h ir strata, bet H ir stratu skaits. Ja nav stratifikācijas, tad visa mērķa populācija U tiek uzskatīta par vienu stratu.
- i ir primārā izlases vienība (PIV) h – tajā stratā. n_h norāda primārās izlases vienību skaitu h – tajā stratā.
- j ir mājsaimniecības indekss h – tajā stratā i – tajā PIV. m_{hi} norāda mājsaimniecību skaitu. Vienpakāpju izlases gadījumā, katra mājsaimniecība ir uztverama kā PIV.
- w_{hij} ir j – tās mājsaimniecības svars h – tajā stratā i – tajā PIV.
- θ_{hij} ir j – tās mājsaimniecības pētāmais mainīgais h – tajā stratā i – tajā PIV. [2], [9], [10], [13], [14]

2.2. Summārā tipa statistika

Šķērsgriezuma summārā tipa statistika tiek definēta:

$$Y_t = \sum_{i \in U_t} y_{t;i}, \quad (2.2.1)$$

kur U_t – populācija viltņi t ,

$y_{t;i}$ – i – tā elementa indikatora vērtība vilnī t .

Šajā nodaļā tiks aplūkots šķērsgriezums no viļņa t , tad šķērsgriezuma summārā tipa statistika tiek definēta:

$$Y = \sum_{i \in U} y_i, \quad (2.2.2)$$

kur U – populācija vilnī t ,

y_i – i – tā elementa indikatora vērtība vilnī t .

Šķērsgriezuma summārā tipa statistikas novērtējums tiek definēts:

$$\hat{Y} = \sum_{i \in s} w_i \cdot y_i, \quad (2.2.3)$$

kur s – izlase vilnī t ,

w_i – i – tā elementa svari vilnī t .

Šķērsgriezuma summārā tipa statistikas dispersijas novērtēšanai tiek izmantota galīgā klāsteru metode un daudzfaktoru lineārās regresijas modelis. Ja izlases dizains ir vienpakāpju klāsteris, tad par PIV tiek uzskatītas mājsaimniecības vai personas, ja ir divpakāpju klāsteris, tad tautas skaitīšanas iecirknis.

Novērtējuma \hat{Y} agregācija katrai PIV vilnī t :

$$\check{y}_j = \sum_{i \in s_{PIV_j}} w_i \cdot y_i, \quad (2.2.4)$$

kur $j = \overline{1, m}$,

m – apzīmē PIV skaitu vilnī t ,

$PIV_j - j$ – tā PIV vilnī t .

Šķērsgriezuma summārā tipa statistikas dispersijas novērtējums pēc galīgās klāsteru metodes:

$$\widehat{var}_{cross;t}(\hat{Y}) = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{i=1}^{n_h} (y_{hi\cdot} - \bar{y}_{h\cdot\cdot})^2, \quad (2.2.5)$$

kur $y_{hi\cdot} = \sum_{j=1}^{m_{hi}} w_{hij} \cdot y_{hij}$ un $\bar{y}_{h\cdot\cdot} = \frac{(\sum_{i=1}^{n_h} y_{hi\cdot})}{n_h}$. Šķērsgriezuma summārā tipa statistikas dispersijas novērtēšanai $(y_{hi\cdot} - \bar{y}_{h\cdot\cdot})^2$ vietā tiks izmantoti novērtētie atlikumi no regresijas modeļa.

Regresijas modelis:

$$\check{y}_j = \beta' \cdot v_j^\circ + e_j^\circ, \quad (2.2.6)$$

kur kreisā puse regresijas modelim satur mainīgo \check{y}_j , kas tiek definēts, izmantojot izteiksmi (2.2.4), bet labā puse regresijas modelim satur β' , v_j° , e_j° , kur v_j° ir vektors, kas sastāv no H (stratu skaita) fiktīvajām komponentēm un tiek definēts:

$$v_j^\circ = (v_{1;j}, \dots, v_{h;j}, \dots, v_{H;j})', \quad (2.2.7)$$

kur katra fiktīvā komponente var pieņemt divas vērtības 0 vai 1, atkarībā no tā vai j – tā PIV pieder h – tajā stratā vilnī t vai nepieder. Fiktīvā komponente tiek definēta:

$$v_{h;j} = \begin{cases} 1, & \text{ja } PIV_j \in U_h \\ 0, & \text{ja } PIV_j \notin U_h \end{cases} \quad (2.2.8)$$

kur U_h ir h – tās stratas PIV vienību rāmis vilnim t . β ir regresijas modeļa parametri, kas tiek definēti vektora formā:

$$\beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_H \end{pmatrix}, \quad (2.2.9)$$

kur e_j° ir regresijas atlikumi (katram elementam savs atkarībā no tā, pa kuru PIV tika veikta agregācija jeb summēšana), atlikumi būs m skaitā. No regresijas modeļa tiek iegūti atlikumi \hat{e}_j° , kas ir e_j° novērtējums.

Iegūtiem novērtētiem atlikumiem tiek izmantota izteiksmes (2.2.5), un summārā tipa statistikas dispersijas novērtējums pēc Osiera metodoloģijas [13] tiek definēts:

$$\widehat{var}_{cross;t}(\hat{Y}) = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{j=1}^{n_h} (\hat{e}_j^\circ)^2, \quad (2.2.10)$$

kur h – strata vilnī t ,

H – stratu skaits.

n_h – norāda PIV skaitu h – tā stratā vilnī t .

2.3. Attiecību tipa statistika EU-SILC šķērsriezuma rādītājam

EU-SILC šķērsriezuma rādītājs, attiecību tipa statistikas gadījumā, ir nabadzības vai sociālās atstumtības risks jeb AROPE pret populācijas apjomu.

Nabadzības vai sociālās atstumtības risks – rādītājs, kurš norāda, vai personas ienākumi ir mazāki par nabadzības riska sliekšni, vai personas ir dziļi materiāli nenodrošinātas, vai personas, kuru vecums mazāks par 60 gadiem, nodarbinātas darbā ar zemu intensitāti. [8], [11]

Nabadzības riska sliekšnis – 60% no rīcībā esošo ienākumu mediānas pārrēķinātas uz ekvivalento patērētāju skaitu mājsaimniecībā. [1], [8], [11]

Nabadzības vai sociālās atstumtības riskam jeb AROPE ir trīs apakšrādītāji – nabadzības risks jeb POV, dziļas materiālas nenodrošinātība jeb DEP un rādītājs, kurš norāda, vai personas, kuru vecums mazāks par 60 gadiem, ir nodarbinātas darbā ar zemu intensitāti jeb LWI. [1], [8], [11]



2.3.1. att. Nabadzības vai sociālās atstumtības risks (ARPE) ar trīs apakšrādītājiem

2.4. Attiecību tipa statistika

Šķērsgriezuma attiecību tipa statistika tiek definēta:

$$R = \frac{Y}{Z} = \frac{\sum_{i \in U} y_i}{\sum_{i \in U} z_i}, \quad (2.4.1)$$

kur U – populācija vīlnī t ,

$z_i - i$ – tā elementa mainīgā z vērtība vīlnī t ,

$y_i - i$ – tā elementa indikatora vērtība vīlnī t .

Novērtējums attiecību tipa šķērsgriezuma statistikai:

$$\hat{R} = \frac{\hat{Y}}{\hat{Z}} = \frac{\sum_{i \in s} w_i \cdot y_i}{\sum_{i \in s} w_i \cdot z_i}, \quad (2.4.2)$$

kur s – izlase vīlnī t ,

$w_i - i$ – tā elementa bāzes svāri, kas tiek rēķināti vīlnī t .

Izteiksme (2.4.2) izsaka novērtējumu starp divām summārām funkcijām. Dispersijas novērtēšanai tiks izmantota delta metode, kas izsaka šķērsgriezuma attiecību tipa statistikas tuvinājumu:

$$\hat{R} - R \approx \text{grad}(\tau)'(\hat{t} - \tau), \quad (2.4.3)$$

kur $\tau = (Y, Z)'$, $\hat{t} = (\hat{Y}, \hat{Z})'$, un $\text{grad}(\tau)$ ir gradients, kas ir 2×1 vektors, kas sastāv no divām komponentēm, kuras tiek iegūtas atvasinot R pēc τ . $\text{grad}(\tau)$ tiek definēts:

$$\text{grad}(\tau) = \left(\frac{1}{Z}; -\frac{Y}{Z^2} \right)' \quad (2.4.4)$$

Dispersijas tuvinājums, izmantojot delta metodi, šķērsgriezuma attiecību tipa statistikai tiek definēts:

$$\text{var}_{\text{cross};t}(\hat{R}) \approx \text{grad}(\tau)' \cdot \text{cov}_{\text{cross};t}(\hat{t}) \cdot \text{grad}(\tau), \quad (2.4.5)$$

kur $\text{cov}_{\text{cross};t}(\hat{t})$ ir 2×2 simetriska kovariāciju matrica starp novērtējumiem $\hat{t} = (\hat{Y}, \hat{Z})'$ vilnī t un matrica tiek definēta:

$$\text{cov}_{\text{cross};t}(\hat{t}) = \begin{pmatrix} \text{var}(\hat{Y}) & \text{cov}(\hat{Y}, \hat{Z}) \\ \text{cov}(\hat{Y}, \hat{Z}) & \text{var}(\hat{Z}) \end{pmatrix} \quad (2.4.6)$$

Uz diagonāles $\text{cov}_{\text{cross};t}(\cdot)$ atrodas dispersija, novērtētam \hat{Y} un \hat{Z} , bet pārējie elementi ir $\text{cov}(\cdot, \cdot)$, kas ir kovariācija starp \hat{Y} un \hat{Z} .

Kovariāciju matricas $\text{cov}_{\text{cross};t}(\hat{t})$ novērtēšana tiek iegūta pēc Osiera un Bergera metodoloģijas. Abas metodoloģijas izmantoto lineāro daudzfaktoru regresijas modeli. Ja izlases dizains ir vienpakāpju klāsteris, tad par primārās izlases vienībām (PIV) tiek uzskatītas mājsaimniecības vai personas, ja ir divpakāpju klāsteris, tad tautas skaitīšanas iecirknis.

Regresijas modeļa konstruēšanai tiek veikta novērtējumu \hat{Y} un \hat{Z} agregācija katrai PIV:

$$\check{y}_j = \sum_{i \in S \cap PIV_j} w_i \cdot y_i, \quad (2.4.7)$$

$$\check{z}_j = \sum_{i \in S \cap PIV_j} w_i \cdot z_i, \quad (2.4.8)$$

kur $j = \overline{1, m}$,

m – apzīmē PIV skaitu vilnī t ,

PIV_j – j – tā PIV vilnī t .

Daudzfaktoru regresijas modelis:

$$\begin{pmatrix} \check{y}_j \\ \check{z}_j \end{pmatrix} = \beta' \cdot v_j^\circ + e_j^\circ, \quad (2.4.9)$$

kur kreisā puse regresijas modelim satur mainīgo \check{y}_j , kas tiek definēts izmantojot izteiksmi (2.4.7) un mainīgais \check{z}_j tiek definēts izmantojot izteiksmi (2.4.8), bet labā puse regresijas modelim satur $\beta', v_j^\circ, e_j^\circ$, kur v_j° ir vektors, kas sastāv no H (stratu skaita) fiktīvām komponentēm un tiek definēts:

$$v_j^\circ = (v_{1,j}, \dots, v_{h,j}, \dots, v_{H,j})', \quad (2.4.10)$$

$$v_{h,j} = \begin{cases} 1, & \text{ja } PIV_j \in U_h \\ 0, & \text{ja } PIV_j \notin U_h \end{cases} \quad (2.4.11)$$

kur U_h ir h – tās stratas PIV vienību rāmis vilnī t .

β ir regresijas modeļa parametri, kas tiek definēti kā matrica:

$$\beta = \begin{pmatrix} \beta_{11} & \beta_{21} \\ \vdots & \vdots \\ \beta_{1H} & \beta_{2H} \end{pmatrix}, \quad (2.4.12)$$

Atlikumi e_j° ir vektors, kas sastāv no divām komponentēm:

$$e_j^\circ = \begin{pmatrix} e_{y,j} \\ e_{z,j} \end{pmatrix}, \quad (2.4.13)$$

kur katra komponente, $e_{y,j}$ un $e_{z,j}$, ir skaitā m ,

$e_{y,j}$ ir atlikumi mainīgam \check{y}_j vilnī t ,

$e_{z,j}$ ir atlikumi mainīgam \check{z}_j vilnī t .

Atlikumiem e_j° nav nepieciešam noteikt sadalījumu, jo parastā MKM metode novērtē regresijas modeļa parametrus. No regresijas modeļa tiek iegūti atlikumi $\hat{e}_{y,j}$, kas ir $e_{y,j}$ novērtējums, un attiecīgi $\hat{e}_{z,j}$, kas ir $e_{z,j}$ novērtējums.

Pēc Bergera metodoloģijas kovariāciju matricas novērtējums tiek definēts:

$$\widehat{cov}_{cross;t}(\hat{t}) = (n - H) \cdot \widehat{\Sigma}, \quad (2.4.14)$$

$\widehat{\Sigma}$ ir simetriska novērtēto atlikumu kovariāciju matrica, kas tiek iegūta no regresijas modeļa (2.4.9) un tiek definēta:

$$\widehat{\Sigma} = \begin{pmatrix} \widehat{\Sigma}_{11} & \widehat{\Sigma}_{12} \\ \widehat{\Sigma}_{21} & \widehat{\Sigma}_{22} \end{pmatrix} = \frac{\epsilon^T \epsilon}{n - H}, \quad (2.4.15)$$

kur $\epsilon = \begin{pmatrix} \hat{e}_{y;j} \\ \hat{e}_{z;j} \end{pmatrix}, j = \overline{1, m}$,

kur n – kopējais PIV skaits,

H – stratu skaits.

Pēc Osiera metodoloģijas kovariāciju matricas novērtējums [13] tiek definēts:

$$\widehat{cov}_{cross;t}(\hat{t}) = \begin{pmatrix} \widehat{\Sigma}_{11} & \widehat{\Sigma}_{12} \\ \widehat{\Sigma}_{21} & \widehat{\Sigma}_{22} \end{pmatrix}, \quad (2.4.16)$$

$$\widehat{\Sigma}_{11} = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{j=1}^{n_h} (\hat{e}_{y;j})^2, \quad (2.4.17)$$

$$\widehat{\Sigma}_{22} = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{j=1}^{n_h} (\hat{e}_{z;j})^2, \quad (2.4.18)$$

Tā kā matrica $\widehat{\Sigma}$ ir simetriska matrica, tad komponente $\widehat{\Sigma}_{12}$ ir vienāda ar $\widehat{\Sigma}_{21}$, un tiek definēta:

$$\widehat{\Sigma}_{12} = \sum_{h=1}^H \frac{n_h}{n_h - 1} \sum_{j=1}^{n_h} \hat{e}_{y;j} \cdot \hat{e}_{z;j}, \quad (2.4.19)$$

kur H – stratu skaits,

n_h – PIV skaits vilnī t h – tajā stratā.

Gan Bergera, gan Osiera metodoloģija šķērsgriezuma attiecību tipa statistikas dispersijas novērtēšanai izmanto delta metodi, kur atšķirās vienīgi matricas $\widehat{cov}_{cross;t}(\hat{t})$ novērtēšana. Tā kā ir iegūta matrica $\widehat{cov}_{cross;t}(\hat{t})$, tad šķērsgriezuma attiecību tipa statistikas dispersijas novērtējums tiek definēts:

$$\widehat{var}_{cross;t}(\hat{R}) = grad(\hat{t})' \cdot \widehat{cov}_{cross;t}(\hat{t}) \cdot grad(\hat{t}), \quad (2.4.20)$$

kur $grad(\hat{t})$ ir tāds pats vektors kā izteiksmē (2.4.4), tikai τ vietā tiek izmantots tā novērtējums \hat{t} . [2], [5], [9], [13]

3. ŠĶĒRSGRIEZUMA RĀDĪTĀJU IZMAIŅU DISPERSIJAS NOVĒRTĒJUMS

Izmaiņas laikā ir svarīgs jautājums daudzās sociālās, ekonomikas un demogrāfijas jomās. Ļoti nozīmīgi ir salīdzināt divus šķērsgriezumu novērtējumus diviem pētāmajiem rādītājiem, kas tiek ņemti no diviem dažādiem viļņiem.

Izmaiņas šķērsgriezuma rādītājiem, gan attiecību tipa, gan summārā tipa, tiek definētas, kā starpība starp diviem rādītājiem no diviem dažādiem viļņiem. [4], [5]

3.1. Summārā tipa statistika

Izmaiņas šķērsgriezuma summārā tipa rādītājiem tiek definētas:

$$\Delta = Y_2 - Y_1, \quad (3.1.1)$$

$Y_1 = \sum_{i \in U_1} y_{1;i}$ – 1. viļņa populācijas summārais rādītājs,

$Y_2 = \sum_{i \in U_2} y_{2;i}$ – 2. viļņa populācijas summārais rādītājs,

$y_{1;i} - i$ – tā elementa mainīgā y vērtība 1. vilnī,

$y_{2;i} - i$ – tā elementa mainīgā y vērtība 2. vilnī.

Izmaiņu Δ novērtējums:

$$\hat{\Delta} = \hat{Y}_2 - \hat{Y}_1, \quad (3.1.2)$$

kur \hat{Y}_1 un \hat{Y}_2 tiek definēti:

$$\hat{Y}_1 = \sum_{i \in s_1} w_{1;i} \cdot y_{1;i}, \quad (3.1.3)$$

$$\hat{Y}_2 = \sum_{i \in s_2} w_{2;i} \cdot y_{2;i}, \quad (3.1.4)$$

kur s_1 – 1. viļņa izlase,

s_2 – 2. viļņa izlase,

$w_{1;i} - i$ – tā elementa svāri 1. vilnī,

$w_{2;i} - i$ – tā elementa svāri 2. vilnī.

Šķērsgriezuma rādītāju izmaiņu novērtējums ir divu summāro funkciju starpība, tāpēc dispersijas novērtēšanai tiek izmantota delta metode:

$$\hat{\Delta} - \Delta \approx \text{grad}(\tau)'(\hat{\tau} - \tau), \quad (3.1.5)$$

kur $\tau = (Y_1, Y_2)'$, $\hat{\tau} = (\hat{Y}_1, \hat{Y}_2)'$, un $\text{grad}(\tau)$ ir gradients, kas ir 2×1 vektors, kas sastāv no divām komponentēm, kuras tiek iegūtas atvasinot Δ pēc τ .

$grad(\tau)$ tiek definēts:

$$grad(\tau) = (-1; 1)' \quad (3.1.6)$$

Dispersijas tuvinājums šķērsgriezuma summārā tipa statistikas izmaiņām tiek definēts:

$$var_{chans}(\hat{\Delta}) \approx grad(\tau)' \cdot cov_{chans}(\hat{t}) \cdot grad(\tau), \quad (3.1.7)$$

kur $cov_{chans}(\hat{t})$ ir 2×2 simetriska kovariāciju matrica starp novērtējumiem $\hat{t} = (\hat{Y}_1, \hat{Y}_2)'$, un tā tiek definēta:

$$cov_{chans}(\hat{t}) = \begin{pmatrix} var(\hat{Y}_1) & cov(\hat{Y}_1, \hat{Y}_2) \\ cov(\hat{Y}_1, \hat{Y}_2) & var(\hat{Y}_2) \end{pmatrix} \quad (3.1.8)$$

Uz diagonāles $cov_{chans}(\cdot)$ atrodas dispersija novērtētiem \hat{Y}_1 un \hat{Y}_2 , bet $cov(\cdot, \cdot)$ ir kovariācija starp \hat{Y}_1 un \hat{Y}_2 , kas tiek definēta izmantojot korelāciju:

$$cov(\hat{Y}_1, \hat{Y}_2) = corr(\hat{Y}_1, \hat{Y}_2) \cdot \sqrt{var_{cross;1}(\hat{Y}_1) \cdot var_{cross;2}(\hat{Y}_2)}, \quad (3.1.9)$$

kur $var_{cross;1}(\hat{Y}_1)$ tiek iegūts no summārā tipa statistikas šķērsgriezumam 1. vilnī un $var_{cross;2}(\hat{Y}_2)$ attiecīgi 2. vilnī, bet $corr(\hat{Y}_1, \hat{Y}_2)$ novērtēšana tiek iegūtā gan pēc Bergera, gan pēc Osiera metodoloģijas. Abas metodoloģijas izmanto lineāro daudzfaktoru regresijas modeli. Ja izlases dizains ir vienpakāpju klāsteris, tad par primārās izlases vienībām (PIV) tiek uzskatītas mājsaimniecības vai personas, ja ir divpakāpju klāsteris, tad tautas skaitīšanas iecirknis. Regresijas modeļa konstruēšanai tiek veikta novērtējumu \hat{Y}_1 un \hat{Y}_2 agregācija katrai PIV, kas tiek definēta:

$$\check{y}_{1;j} = \sum_{i \in s_1 \cap PIV_j} w_{1;i} \cdot y_{1;i}, \quad (3.1.10)$$

$$\check{y}_{2;j} = \sum_{i \in s_2 \cap PIV_j} w_{2;i} \cdot y_{2;i}, \quad (3.1.11)$$

Katra viļņa šķērsgriezuma summārā tipa statistikas dispersijas novērtējums tiek iegūts no regresijas modeļa atlikumiem un galīgās klāsteru metodes. [1], [2], [4]

Daudzfaktoru regresijas modelis diviem šķērsgriezuma summārā tipa rādītājiem no diviem dažādiem viļņiem:

$$\begin{pmatrix} \check{y}_{1;j}^\circ \\ \check{y}_{2;j}^\circ \end{pmatrix} = \begin{pmatrix} \beta_1^{(1)} \cdot u_{1;j} + \beta_2^{(1)} \cdot u_{2;j} + \beta_{12}^{(1)} \cdot u_{1;j} \cdot u_{2;j} \\ \beta_1^{(2)} \cdot u_{1;j} + \beta_2^{(2)} \cdot u_{2;j} + \beta_{12}^{(2)} \cdot u_{1;j} \cdot u_{2;j} \end{pmatrix} + e_j^\circ, \quad (3.1.12)$$

kur $\check{y}_{1;j}^\circ$ un $\check{y}_{2;j}^\circ$ ir atbildētības mainīgie, kas tiek definēti:

$$\check{y}_{1;j}^\circ = \begin{cases} \check{y}_{1;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases} \quad (3.1.13)$$

$$\check{y}_{2;j}^{\circ} = \begin{cases} \check{y}_{2;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases} \quad (3.1.14)$$

$$u_{1;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_1 \\ 0, & \text{ja } PIV_j \notin s_1 \end{cases} \quad (3.1.15)$$

$$u_{2;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_2 \\ 0, & \text{ja } PIV_j \notin s_2 \end{cases} \quad (3.1.16)$$

$$u_{1;j} \cdot u_{2;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_1 \cap s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cap s_2 \end{cases} \quad (3.1.17)$$

Atlikumi e_j° ir vektors, kas sastāv no divām komponentēm:

$$e_{t;j}^{\circ} = \begin{pmatrix} e_{y;1;j} \\ e_{y;2;j} \end{pmatrix}, \quad (3.1.18)$$

kur komponente $e_{y;1;j}$ ir skaitā m_1 un $e_{y;2;j}$ ir skaitā m_2 .

No regresijas modeļa tiek iegūti atlikumi $\hat{e}_{y;1;j}$, kas ir $e_{y;1;j}$ novērtējums, un attiecīgi $\hat{e}_{y;2;j}$, kas ir $e_{y;2;j}$ novērtējums.

Pēc Bergera metodoloģijas korelācijas novērtējums tiek definēts:

$$\widehat{corr}(\hat{Y}_1, \hat{Y}_2) = \frac{V_{12}}{\sqrt{V_{11} \cdot V_{22}}}, \quad (3.1.19)$$

$$\hat{V} = \frac{\epsilon^T \epsilon}{n - H} \quad (3.1.20)$$

kur $\epsilon = \begin{pmatrix} \hat{e}_{y;1;j} \\ \hat{e}_{y;2;j} \end{pmatrix}$.

Pēc Osiera metodoloģijas korelācijas novērtējums [13] tiek definēts:

$$\widehat{corr}(\hat{Y}_1, \hat{Y}_2) = \frac{\sum_{j \in s_1 \cup s_2} \hat{e}_{y;1;j} \cdot \hat{e}_{y;2;j}}{\sqrt{\sum_{j \in s_1 \cup s_2} (\hat{e}_{y;1;j})^2} \cdot \sqrt{\sum_{j \in s_1 \cup s_2} (\hat{e}_{y;2;j})^2}}, \quad (3.1.21)$$

Gan Bergera, gan Osiera metodoloģija šķērsgriezuma summārā tipa statistikas izmaiņu dispersijas novērtēšanai izmanto delta metodi, kur atšķirās vienīgi matricas $\widehat{cov}_{chans}(\hat{\tau})$ novērtēšana, kas tiek īstenota izmantojot korelāciju. Tā kā ir iegūts korelācijas novērtējums $\widehat{corr}(\hat{Y}_1, \hat{Y}_2)$, tad šķērsgriezuma summārā tipa statistikas izmaiņu dispersijas novērtējums tiek definēts:

$$\widehat{var}_{chans}(\hat{\Delta}) = grad(\hat{\tau})' \cdot \widehat{cov}_{chans}(\hat{\tau}) \cdot grad(\hat{\tau}), \quad (3.1.22)$$

kur $grad(\hat{\tau})$ ir tāds pats vektors kā izteiksmē (3.1.6), tikai τ vietā tiek izmantots tā novērtējums $\hat{\tau}$. [2], [5], [9], [13]

3.2. Attiecību tipa statistika

Absolūtās izmaiņas šķērsgriezuma attiecību tipa rādītājiem tiek definētas:

$$\Delta = R_2 - R_1, \quad (3.2.1)$$

$$R_1 = \frac{Y_1}{Z_1} = \frac{\sum_{i \in U_1} y_{1;i}}{\sum_{i \in U_1} z_{1;i}}, \quad (3.2.2)$$

$$R_2 = \frac{Y_2}{Z_2} = \frac{\sum_{i \in U_2} y_{2;i}}{\sum_{i \in U_2} z_{2;i}}, \quad (3.2.3)$$

kur U_1 – 1. viļņa populācija,

U_2 – 2. viļņa populācija,

$y_{1;i}$ – i – tā elementa indikatora y vērtība 1. vilnī,

$y_{2;i}$ – i – tā elementa indikatora y vērtība 2. vilnī,

$z_{1;i}$ – i – tā elementa mainīgā z vērtība 1. vilnī,

$z_{2;i}$ – i – tā elementa mainīgā z vērtība 2. vilnī.

Izmaiņu $\hat{\Delta}$ novērtējums:

$$\hat{\Delta} = \hat{R}_2 - \hat{R}_1 = \frac{\hat{Y}_2}{\hat{Z}_2} - \frac{\hat{Y}_1}{\hat{Z}_1} = \frac{\sum_{i \in S_2} w_{2;i} y_{2;i}}{\sum_{i \in S_2} w_{2;i} z_{2;i}} - \frac{\sum_{i \in S_1} w_{1;i} y_{1;i}}{\sum_{i \in S_1} w_{1;i} z_{1;i}}, \quad (3.2.4)$$

kur $w_{1;i}$ ir i – tā elementa svāri, kas tiek rēķināti 1. vilnī,

$w_{2;i}$ ir i – tā elementa svāri, kas tiek rēķināti 2. vilnī.

Izmaiņu novērtējums sastāv no četrām summārām funkcijām, tāpēc dispersijas novērtēšanai tiek izmantota delta metode, kas tiek definēta:

$$\hat{\Delta} - \Delta \approx \text{grad}(\tau)'(\hat{\tau} - \tau), \quad (3.2.5)$$

kur $\tau = (Y_1, Z_1, Y_2, Z_2)'$, $\hat{\tau} = (\hat{Y}_1, \hat{Z}_1, \hat{Y}_2, \hat{Z}_2)'$, un $\text{grad}(\tau)$ ir gradients, kas ir 4×1 vektors, kas sastāv četrām komponentēm, kuras tiek iegūtas atvasinot Δ pēc τ .

$\text{grad}(\tau)$ tiek definēts:

$$\text{grad}(\tau) = \left(-\frac{1}{Z_1}; \frac{Y_1}{Z_1^2}; \frac{1}{Z_2}; -\frac{Y_2}{Z_2^2} \right)' \quad (3.2.6)$$

Dispersijas tuvinājums šķērsgriezuma attiecību tipa statistikas izmaiņām tiek definēts:

$$\text{var}_{chans}(\hat{\Delta}) \approx \text{grad}(\tau)' \cdot \text{cov}_{chans}(\hat{\tau}) \cdot \text{grad}(\tau), \quad (3.2.7)$$

kur $\text{cov}_{chans}(\hat{\tau})$ ir 4×4 simetriska kovariāciju matrica starp novērtējumiem. $\hat{\tau} = (\hat{Y}_1, \hat{Z}_1, \hat{Y}_2, \hat{Z}_2)'$, un tā tiek definēta:

$$cov_{chans}(\hat{\tau}) = \begin{pmatrix} var(\hat{Y}_1) & cov(\hat{Y}_1, \hat{Z}_1) & cov(\hat{Y}_1, \hat{Y}_2) & cov(\hat{Y}_1, \hat{Z}_2) \\ cov(\hat{Y}_1, \hat{Z}_1) & var(\hat{Z}_1) & cov(\hat{Z}_1, \hat{Y}_2) & cov(\hat{Z}_1, \hat{Z}_2) \\ cov(\hat{Y}_1, \hat{Y}_2) & cov(\hat{Z}_1, \hat{Y}_2) & var(\hat{Y}_2) & cov(\hat{Y}_2, \hat{Z}_2) \\ cov(\hat{Y}_1, \hat{Z}_2) & cov(\hat{Z}_1, \hat{Z}_2) & cov(\hat{Y}_2, \hat{Z}_2) & var(\hat{Z}_2) \end{pmatrix} \quad (3.2.8)$$

Uz diagonāles $cov_{chans}(\cdot)$ atrodas dispersija, novērtētiem \hat{Y}_1 , \hat{Z}_1 , \hat{Y}_2 un \hat{Z}_2 , bet $cov(\cdot, \cdot)$, kas ir kovariācija starp \hat{Y}_1 , \hat{Z}_1 , \hat{Y}_2 un \hat{Z}_2 , kas tiek definēta izmantojot korelāciju, tā pat kā (3.1.9).

Korelācijas novērtēšana tiek iegūtā gan pēc Bergera, gan pēc Osiera metodoloģijas. Abas metodoloģijas izmanto daudzfaktoru regresijas modelis. Ja izlases dizains ir vienpakāpju klāsteris, tad par primārās izlases vienībām (PIV) tiek uzskatītas mājsaimniecības vai personas, ja ir divpakāpju klāsteris, tad tautas skaitīšanas iecirknis. Regresijas modeļa konstruēšanai tiek veikta novērtējumu \hat{Y}_1 , \hat{Z}_1 , \hat{Y}_2 un \hat{Z}_2 agregācija katrai PIV:

$$\check{y}_{1;j} = \sum_{i \in s_1 \cap PIV_j} w_{1;i} \cdot y_{1;i}, \quad (3.2.9)$$

$$\check{y}_{2;j} = \sum_{i \in s_2 \cap PIV_j} w_{2;i} \cdot y_{2;i}, \quad (3.2.10)$$

$$\check{z}_{1;j} = \sum_{i \in s_1 \cap PIV_j} w_{1;i} \cdot z_{1;i}, \quad (3.2.11)$$

$$\check{z}_{2;j} = \sum_{i \in s_2 \cap PIV_j} w_{2;i} \cdot z_{2;i}, \quad (3.2.12)$$

Daudzfaktoru regresijas modelis attiecība tipa šķērsgriezuma rādītāju izmaiņu kovariācijas matricu novērtēšanai:

$$\begin{pmatrix} \check{y}_{1;j}^\circ \\ \check{z}_{1;j}^\circ \\ \check{y}_{2;j}^\circ \\ \check{z}_{2;j}^\circ \end{pmatrix} = \begin{pmatrix} \beta_1^{(1)} \cdot u_{1;j} + \beta_2^{(1)} \cdot u_{2;j} + \beta_{12}^{(1)} \cdot u_{1;j} \cdot u_{2;j} \\ \beta_1^{(2)} \cdot u_{1;j} + \beta_2^{(2)} \cdot u_{2;j} + \beta_{12}^{(1)} \cdot u_{1;j} \cdot u_{2;j} \\ \beta_1^{(3)} \cdot u_{1;j} + \beta_2^{(3)} \cdot u_{2;j} + \beta_{12}^{(3)} \cdot u_{1;j} \cdot u_{2;j} \\ \beta_1^{(4)} \cdot u_{1;j} + \beta_2^{(4)} \cdot u_{2;j} + \beta_{12}^{(4)} \cdot u_{1;j} \cdot u_{2;j} \end{pmatrix} + e_j^\circ, \quad (3.2.13)$$

kur $\check{y}_{1;j}^\circ$, $\check{z}_{1;j}^\circ$, $\check{y}_{2;j}^\circ$ un $\check{z}_{2;j}^\circ$ ir mainīgie, kas tiek definēti:

$$\check{y}_{1;j}^\circ = \begin{cases} \check{y}_{1;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases}, \quad (3.2.14)$$

$$\check{z}_{1;j}^\circ = \begin{cases} \check{z}_{1;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases}, \quad (3.2.15)$$

$$\check{y}_{2;j}^\circ = \begin{cases} \check{y}_{2;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases}, \quad (3.2.16)$$

$$\check{z}_{2;j}^\circ = \begin{cases} \check{z}_{2;j}, & \text{ja } PIV_j \in s_1 \cup s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cup s_2 \end{cases}, \quad (3.2.17)$$

$$u_{1;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_1 \\ 0, & \text{ja } PIV_j \notin s_1' \end{cases} \quad (3.2.18)$$

$$u_{2;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_2 \\ 0, & \text{ja } PIV_j \notin s_2' \end{cases} \quad (3.2.19)$$

$$u_{1;j} \cdot u_{2;j} = \begin{cases} 1, & \text{ja } PIV_j \in s_1 \cap s_2 \\ 0, & \text{ja } PIV_j \notin s_1 \cap s_2' \end{cases} \quad (3.2.20)$$

Atlikumi e_j° ir vektors, kas sastāv no četrām komponentēm:

$$e_{t;j}^\circ = \begin{pmatrix} e_{y;1;j} \\ e_{z;1;j} \\ e_{y;2;j} \\ e_{z;2;j} \end{pmatrix}, \quad (3.2.21)$$

kur komponentes $e_{y;1;j}$, $e_{z;1;j}$, $e_{y;2;j}$ un $e_{z;2;j}$, un katra komponente ir skaitā m_t , jo $j = \overline{1, m_t}$
 m_t – apzīmē PIV skaitu vilnī t , $t = \overline{1, 2}$. [1], [2], [4], [5]

No daudzfaktoru regresijas modeļa tiek iegūti novērtētie atlikumi $\hat{e}_{y;1;j}$, $\hat{e}_{z;1;j}$, $\hat{e}_{y;2;j}$ un $\hat{e}_{z;2;j}$, kas veido novērtēto atlikumu kovariāciju matricu $\widehat{cov}_{chans}(\hat{t})$, kas tiek definēta [3]:

$$\widehat{cov}_{chans}(\hat{t}) = \widehat{D}' \cdot \widehat{\Sigma} \cdot \widehat{D}, \quad (3.2.22)$$

pēc Bergera metodoloģijas [13]:

$$\widehat{\Sigma} = \frac{\epsilon^T \epsilon}{n - H}, \quad (3.2.23)$$

kur $\epsilon^T \epsilon = \begin{pmatrix} \hat{e}_{y;1;j} \\ \hat{e}_{z;1;j} \\ \hat{e}_{y;2;j} \\ \hat{e}_{z;2;j} \end{pmatrix}$, bet \widehat{D} ir diagonālmatrix, kas tiek definēta:

$$\widehat{D} = \text{diag} \left\{ \sqrt{\widehat{var}_{cross;1}(\hat{Y}_1) \widehat{\Sigma}_{11}^{-1}}, \dots, \sqrt{\widehat{var}_{cross;2}(\hat{Z}_2) \widehat{\Sigma}_{44}^{-1}} \right\}, \quad (3.2.24)$$

Pēc Osiera metodoloģijas uz diagonāles $\widehat{cov}_{chans}(\hat{t})$ atrodas dispersijas novērtējumi šķērsgriezumam $\widehat{var}_{cross;t}(\cdot)$, $t = \overline{1, 2}$, kas tiek definēti kā (2.4.16), bet kovariācijas tiek definētas izmantojot korelāciju, kā (3.1.20), gan $\widehat{var}_{cross;t}(\cdot)$, gan $\widehat{cov}_{chans}(\cdot, \cdot)$ vietā tiek ņemtas attiecīgās komponentes \hat{Y}_t vai \hat{Z}_t .

Bergera un Osiera metodoloģija šķērsgriezuma attiecību tipa statistikas izmaiņu dispersijas novērtēšanai izmanto delta metodi, kur atšķirās vienīgi matricas $\widehat{cov}_{chans}(\hat{t})$.

Tā kā ir iegūts korelācijas novērtējums $\widehat{corr}(\cdot, \cdot)$, tad šķērsriezuma attiecību tipa statistikas izmaiņu dispersijas novērtējums tiek definēts:

$$\widehat{var}_{chans}(\hat{\Delta}) = grad(\hat{\tau})' \cdot \widehat{cov}_{chans}(\hat{\tau}) \cdot grad(\hat{\tau}), \quad (3.2.25)$$

kur $grad(\hat{\tau})$ ir tāds pats vektors kā izteiksmē (3.2.6), tikai τ vietā tiek izmantots tā novērtējums $\hat{\tau}$. [2], [5], [9], [13]

4. ŠĶĒRSGRIEZUMA RĀDĪTĀJU KVALITĀTES RĀDĪTĀJU NOVĒRTĒJUMI PROGRAMMĀ R UN TO PIELIETOJUMS EU-SILC DATIEM

Šajā nodaļā tiks aprakstīta funkcija *vardcros* [7], kas ir izstrādāta programmā R. *vardcros* novērtē kvalitātes rādītājus šķērsgriezuma rādītājiem, vienpakāpju un daudzpakāpju klāsteru dizainiem. Funkcija ir universāla un sniedz plašas iespējas lietotājam, šajā darbā tiks nodrošināti vairāki piemēri *eusilc* simulatīviem un reāliem EU-SILC Latvijas datiem.

4.1. Funkcijas *vardcros* apraksts programmā R

Funkcija *vardcros* [7] novērtē kvalitātes rādītājus šķērsgriezuma rādītājiem, vienpakāpju un daudzpakāpju klāsteru dizainiem. Šķērsgriezuma rādītāji var būt gan summāri, gan attiecību tipa. Funkcija sastāv no mainīgajiem – *Y*, *H*, *PSU*, *w_final*, *id*, *Dom*, *Z*, *country*, *period*, *dataset*, *linratio*, *use.estVar*, *household_level_max*, *withoutperiod*, *netchanges*, *confidence*. Summārā tipa gadījumā *Y* ir šķērsgriezuma rādītājs, kuram jānovērtē kvalitātes rādītāji. Attiecību tipa gadījumā *Y* būs mainīgais, kas atradīsies skaitītājā. *H* ir stratu mainīgais. *PSU* norāda primārās izlases vienības. Mainīgais *w_final* ir izlases svars, bet *id* ir mājsaimniecību vai personu ID. *Dom* ir domēna mainīgais un *Z* ir saucēja mainīgais, šie mainīgie pēc lietotāja izvēles var tikt uzdoti vai arī būt tukši. Ja mainīgais *Z* tiek uzdots, tad funkcija kvalitātes rādītājus izvada attiecību tipa šķērsgriezuma rādītājam, bet ja *Z* pieņem vērtību *NULL*, kā tas ir pēc noklusējuma, tad šķērsgriezuma rādītājs ir summāra tipa. Mainīgie *country* un *period* norāda valsti un periodu. Datu kopa programmā tiek apzīmēta kā *dataset*. *linratio* ir loģiskā vērtība, no kuras tiek noteikts, vai kvalitātes rādītāju novērtēšanā tiks izmantoti linearizētie mainīgie vai gradienti dispersijas novērtēšanā. *use.estVar* ir loģiskā vērtība, kas nosaka, vai funkcijā tiks izmantota funkcija *estVar* atlikumu kovariāciju matricas novērtēšanai. Pēc noklusējuma *use.estVar* pieņem vērtību *FALSE*, kas nozīmē, ka kvalitātes rādītāji tiek novērtēti izmantojot Osiera metodoloģiju, bet gadījumā, ja loģiskā vērtība būtu *TRUE*, tad tiktu īstenota Bergera metodoloģija. *household_level_max* ir loģiskā vērtība, ja loģiskā vērtība ir *TRUE*, tad rādītāju vērtība vienkāršā gadījuma izlases dispersijas novērtēšanai mājsaimniecību līmenī tiek ņemta kā rādītāju maksimālā vērtība, bet ja vērtība ir *FALSE*, tad tiek ņemta kā rādītāju summa. *withperiod* ir loģiskā vērtība, kas nosaka vai periods ir vai nav jāņem vērā. Pēc noklusējuma periods tiek ņemts vērā. *netchanges* ir loģiskā vērtība, kas var pieņemt divas vērtības *TRUE* vai *FALSE*, pēc noklusējuma loģiskā vērtība ir *TRUE*, tas nozīmē, ka tiek veidoti divi objekti, viens no tiem ir agregācija svērtiem datiem pēc perioda, stratas, valsts un primārās izlases vienībām, bet otrs objekts ir paredzēts rādītāja *Y* novērtēšanai, gradients un dispersijas novērtējums pēc

valsts un perioda. *confidence* ir pozitīva vērtība ticamības intervālam, kas pēc noklusējuma ir 0.95. Funkcijas *vardcros* mainīgo apraksts ir apskatāms 1. pielikumā un programmas kods 5. pielikumā.

4.2. Funkcijas *vardcros* programmā *R* testēšanas rezultāti uz simulatīviem datiem

Izstrādātā funkcija *vardcros* [7] programmā *R* sākotnēji tiek testēta uz simulatīviem datiem. Simulatīvie *eusilc* dati ir sintētiski ģenerēti no reāliem Austrijas EU-SILC datiem, kas ir iebūvēti programmā *R*. Tikai daži mainīgie tiek izmantoti no simulatīvās *eusilc* datu kopas funkcijas *vardcros* testēšanai, bet pārējie nepieciešamie tiek simulatīvi ģenerēti. Funkcijas *vardcros* testēšanā no simulatīviem *eusilc* datiem tiek izmantoti iebūvētie mainīgie: *rb050* – personas izlases svāri, *db030* – mājsaimniecības ID, *age* – personas vecums. Simulatīvi ģenerētie mainīgie, izmantojot *eusilc* datu kopu ir gads, strata, primārā izlases vienība, un šķērsgriezuma rādītāji tādi kā POV, DEP, LWI un AROPE.

Izstrādātā piemērā no simulatīviem *eusilc* datiem funkcijas *vardcros* testēšanai sākotnēji tiek izveidots 2011 gads un visiem izvadītiem rezultātiem tiek piekārtota valsts „AT”, t.i., Austrija, jo tiek izmantoti sintētiski ģenerēti reāli Austrijas EU-SILC dati. Tiek izveidotas primārās izlases vienības, kas tiek apzīmētas ar *PSU*, un kas tiek izveidotas izmantojot gadījuma skaitļu ģenerātoru. Simulatīvi, izmantojot gadījuma skaitļu ģenerātoru, tiek izveidoti rādītāji – POV, DEP, LWI un AROPE. Funkcijas *vardcros* [7] iegūtie rezultāti tiks salīdzināti ar funkcijas *vardom* [6], [7] iegūtajiem rezultātiem.

Funkcijas *vardcros* [7] un *vardom* [6], [7] testēšana šķērsgriezuma summārā tipa rādītāju rezultāti simulatīviem *eusilc* datiem programmā *R*:

4.2.1. tabula šķērsgriezuma summārā tipa rādītāju izvades rezultāti programmā *R*

Rādītājs	\hat{Y}	\hat{S}_{Osier}	CV_{Osier}	\hat{S}_{Berger}	CV_{Berger}	\hat{S}_{vardom}	CV_{vardom}
Personu skaits	8182222	120110	1.467941	119508	1.460583	120110	1.467932
Personu skaits, kuru vecums ir zem 60 gadiem	6410431	101283	1.579972	100775	1.572052	101282	1.579962

Summārie rādītāji tabulā 4.2.1 ir kopējais novērtētais personu skaits un kopējais novērtētais personu skaits, kuru vecums ir mazāks par 60 gadiem. Iegūtie rezultāti parāda, ka funkcija *vardcros* pēc Osiera metodoloģijas dod ļoti tuvus novērtētos kvalitātes rādītājus salīdzinot ar funkciju *vardom* [6], [7], savukārt *vardcros* [7] funkcija pēc Bergera metodoloģijas izvada nedaudz atšķirīgus rezultātus. Tabula 4.2.1 parāda, ka novērtētie kvalitātes rādītāji visos

gadījumos ir kvalitatīvi, jo variācijas koeficients ir ļoti mazs. Simulatīvā piemēra programmas kods ar funkcijas *vardcros* pielietojumiem ir aplūkojams 2. pielikumā.

Funkcijas *vardcros* [7] un *vardom* [6], [7] testēšana šķērsriezuma attiecību tipa rādītāju rezultāti simulatīviem *eusilc* datiem programmā *R*:

4.2.2. tabula šķērsriezuma attiecību tipa rādītāju izvades rezultāti programmā *R*

Rādītājs	\hat{R}	\hat{s}_{Osier}	\hat{CV}_{Osier}	\hat{s}_{Berger}	\hat{CV}_{Berger}	\hat{s}_{vardom}	\hat{CV}_{vardom}
AROPE	0.84729	0.00281057	0.331714	0.00279648	0.330051	0.00281055	0.331712
POV	0.49964	0.00468630	0.937937	0.00466281	0.933235	0.00468627	0.937931
DEP	0.49864	0.00376527	0.755103	0.00374640	0.751318	0.00376525	0.755099
LWI	0.50371	0.00471795	0.936639	0.00469430	0.931944	0.00471792	0.936633

Tabula 4.2.2 parāda funkcijas *vardcros* un *vardom* programmā *R* izvades rezultātus, ja rādītāji ir attiecību tipa, tas ir spēkā, ja mainīgais *Z* tiek uzdots. Šajā gadījumā AROPE, POV un DEP tiek dalīti ar kopējo novērtēto personu skaitu, bet rādītājs LWI tiek dalīts ar kopējo novērtēto personu skaitu, kuras vecums ir mazāks par 60 gadiem. Attiecību tipa rādītāju kvalitātes rādītāji tiek rēķināti pēc funkcijas *vardcros* [7], kad tiek izmantotas divas metodoloģijas Osiera un Bergera, kā arī izmantojot funkciju *vardom*. Iegūtie rezultāti parāda, ka novērtētie kvalitātes rādītāji ir kvalitatīvi, jo variācijas koeficients ir ļoti mazs. Simulatīvā piemēra programmas kods ar funkcijas *vardcros* pielietojumiem ir aplūkojams 2. pielikumā.

4.3. Funkcijas *vardcros* programmā *R* testēšanas rezultāti uz reāliem EU-SILC Latvijas datiem

Funkcijas *vardcros* testēšana uz reāliem datiem ir ekvivalenta ar testēšanu uz simulatīviem datiem, atšķirība ir tāda, ka tiek izmantoti reāli EU-SILC dati un novērtētie kvalitātes rādītāji tiek izvadīti personu un mājsaimniecību līmenī. Dati ir sagatavoti ar SAS programmas palīdzību [12]. Funkcija *vardcros* izvadīs rezultātus reāliem EU-SILC Latvijas datiem 2011, 2012 un 2013 gadam personu un mājsaimniecību līmenī, šķērsriezuma summārā un attiecību tipa rādītājiem. Rezultāti funkcijai *vardcros* [7] tiks izvadīti pēc Osiera un Bergera metodoloģijas, kā arī tiks parādīti rezultāti kvalitātes rādītājiem, izmantojot funkciju *vardom* [6], [7].

Funkcijas *vardcros* [7] un *vardom* [6], [7] testēšana šķērsriezuma summārā tipa rādītāju rezultāti reāliem Latvijas EU-SILC datiem programmā *R*:

4.3.1. tabula šķērsgriezuma summārā tipa rādītāju izvades rezultāti programmā R personu līmenī 2011., 2012. un 2013. gadam

Rādītājs	Gads	Ŷ	šē _{Osier}	CV _{Osier}	šē _{Berger}	CV _{Berger}	šē _{vardom}	CV _{vardom}
Skaitis	2011	2047474	38618	1.886147	38556	1.883113	38606	1.885586
Skaitis	2012	2018109	41728	2.067679	41655	2.064085	41716	2.670896
Skaitis	2013	1997382	42008	2.103151	41931	2.099281	41996	2.102587

4.3.2. tabula šķērsgriezuma summārā tipa rādītāju izvades rezultāti programmā R mājsaimniecību līmenī 2011., 2012. un 2013. gadam

Rādītājs	Gads	Ŷ	šē _{Osier}	CV _{Osier}	šē _{Berger}	CV _{Berger}	šē _{vardom}	CV _{vardom}
Skaitis	2011	840519	13468	1.602361	13446	1.599834	13458	1.601174
Skaitis	2012	835803	14385	1.721115	14361	1.718178	14375	1.719919
Skaitis	2013	831086	14587	1.755211	14561	1.752050	14577	1.754057

Funkcijas *vardcros* [7] un *vardom* [6], [7] testēšana šķērsgriezuma attiecību tipa rādītāju rezultāti reāliem Latvijas EU-SILC datiem programmā R:

4.3.3. tabula šķērsgriezuma attiecību tipa rādītāju izvades rezultāti programmā R personu līmenī 2011., 2012. un 2013. gadam

Rādītājs	Gads	Ŕ	šē _{Osier}	CV _{Osier}	šē _{Berger}	CV _{Berger}	šē _{vardom}	CV _{vardom}
AROPE	2011	0.40116	0.008567	2.135945	0.008555	2.132492	0.008566	2.135314
DEP	2011	0.30988	0.008476	2.735235	0.008462	2.730726	0.008473	2.734427
LWI	2011	0.12534	0.005780	4.611084	0.005770	4.603643	0.005778	4.609743
POV	2011	0.18970	0.006710	3.537340	0.006699	3.531511	0.006708	3.536344
AROPE	2012	0.36228	0.008277	2.284834	0.008263	2.280785	0.008275	2.284182
DEP	2012	0.25647	0.007468	2.911801	0.007454	2.906654	0.007466	2.910954
LWI	2012	0.11652	0.005382	4.619081	0.005373	4.611024	0.005381	4.617816
POV	2012	0.19205	0.006828	3.555487	0.006816	3.549135	0.006827	3.554532
AROPE	2013	0.35139	0.008333	2.371438	0.008317	2.366992	0.008331	2.370818
DEP	2013	0.24007	0.007583	3.158584	0.007568	3.152543	0.007581	3.157753
LWI	2013	0.09916	0.005136	5.179375	0.005127	5.169837	0.005135	5.178051
POV	2013	0.19382	0.007229	3.729740	0.007216	3.722639	0.007228	3.728796

4.3.4. tabula šķērsgriezuma attiecību tipa rādītāju izvades rezultāti programmā R mājsaimniecību līmenī 2011., 2012. un 2013. gadam

Rādītājs	Gads	Ŕ	šē _{Osier}	CV _{Osier}	šē _{Berger}	CV _{Berger}	šē _{vardom}	CV _{vardom}
AROPE	2011	0.42613	0.007608	1.785376	0.007596	1.782488	0.007602	1.784059
DEP	2011	0.32666	0.007722	2.363805	0.007709	2.359897	0.007716	2.362063
LWI	2011	0.17085	0.006265	3.666968	0.006255	3.661090	0.006261	3.664273
POV	2011	0.18982	0.005600	2.950315	0.005591	2.945475	0.005596	2.948181
AROPE	2012	0.39910	0.007296	1.828021	0.007283	1.824853	0.007291	1.826740
DEP	2012	0.27811	0.006916	2.486648	0.006904	2.482317	0.006911	2.484907
LWI	2012	0.15990	0.005921	3.703209	0.005911	3.696985	0.005917	3.700634
POV	2012	0.20753	0.005963	2.873332	0.005953	2.868292	0.005959	2.871368
AROPE	2013	0.39125	0.007337	1.875426	0.007324	1.871994	0.007333	1.874197
DEP	2013	0.26745	0.006999	2.617091	0.006986	2.612183	0.006995	2.615379
LWI	2013	0.13851	0.005654	4.082014	0.005644	4.074729	0.005650	4.793447
POV	2013	0.21631	0.006269	2.898030	0.006257	2.892693	0.006264	2.896160

Tabula 4.3.1 un tabula 4.3.2 parāda rezultātus novērtētiem šķērsriezuma summāriem rādītājiem, bet tabula 4.3.3 un tabula 4.3.4 šķērsriezuma attiecību tipa rādītājiem. Novērtētie kvalitātes rādītāji reāliem Latvijas EU-SILC datiem tiek parādīti personu un mājsaimniecību līmenī. Katrā tabulā tiek parādīti funkcijas *vardcros* rezultāti pēc Osiera un Bergera metodoloģijas un pēc funkcijas *vardom*. Rezultāti visos gadījumos ir tuvi, bet nedaudz atšķirīgi, tad var secināt, ka izstrādātā funkcija *vardcros* darbojas saskaņā ar pamatoto teoriju pēc abām metodoloģijām.

5. ŠĶĒRSGRIEZUMA RĀDĪTĀJU IZMAIŅU KVALITĀTES RĀDĪTĀJU NOVĒRTĒJUMI PROGRAMMĀ R UN TO PIELIETOJUMS EU-SILC DATIEM

Šajā nodaļā tiks aprakstīta funkcija *vardchanges* [7], kas ir izstrādāta programmā *R*. *vardchanges* novērtē kvalitātes rādītājus šķērsgriezuma rādītāju izmaiņām, vienpakāpju un daudzpakāpju klāsteru dizainiem. Funkcija balstās uz *vardcros*, kas tiek izmantota kā apakšfunkcijas, no kuras tiek iegūti nepieciešamie dati un mainīgie šķērsgriezuma rādītāju izmaiņu kvalitātes rādītāju novērtēšanai. Funkcija ir universāla un sniedz plašas iespējas lietotājam, šajā darbā uzskatāms pielietojums tiks nodrošināts *eusilc* simulatīviem un reāliem EU-SILC Latvijas datiem.

5.1. Funkcijas *vardchanges* apraksts programmā *R*

Funkcija *vardchanges* [7] novērtē kvalitātes rādītājus šķērsgriezuma rādītāju izmaiņām, vienpakāpju un daudzpakāpju klāsteru dizainiem. Šķērsgriezuma rādītāju izmaiņas var būt gan summāras, gan attiecību tipa. Funkcija sastāv no mainīgajiem – *Y*, *H*, *PSU*, *w_final*, *id*, *Dom*, *Z*, *country*, *periods*, *dataset*, *period1*, *period2* *linratio*, *use.estVar*, *confidence*. Summārā tipa gadījumā *Y* ir šķērsgriezuma rādītāju izmaiņas, kurām jānovērtē kvalitātes rādītāji. Attiecību tipa gadījumā *Y* būs skaitītāja mainīgais. Mainīgie *H*, *PSU*, *w_final*, *id*, *Dom*, *Z*, *country*, *periods*, *dataset*, *linratio*, *use.estVar* un *confidence* atbilst tādām pašām raksturojumam, kā nodaļā 4.1, kur tiek sniegts funkcijas *vardcros* apraksts programmā *R*. Mainīgais *period1* raksturo mainīgā *periods* pirmā perioda vienu rindu un attiecīgi *period2* mainīgā *periods* otrā perioda vienu rindu. Funkcijas *vardchanges* mainīgo apraksts ir apskatāms 3. pielikumā un programmas kods 6. pielikumā.

5.2. Funkcijas *vardchanges* programmā *R* testēšanas rezultāti uz simulatīviem datiem

Izstrādātā funkcija *vardchanges* [7] programmā *R* sākotnēji tiek testēta uz simulatīviem datiem. Simulatīvie *eusilc* dati un mainīgo saraksts, kas tiek izmantots no iebūvētiem datiem *eusilc* datu kopas un mainīgie, kas tiek simulatīvi ģenerēti ir raksturoti nodaļā 4.2.

Izstrādātā piemērā no simulatīviem *eusilc* datiem funkcijas *vardchanges* testēšanai tiek uzdots mainīgais *period1*, kas atbilst 2010 gadam un *period2*, kas atbilst 2011 gadam. Visiem izvadītiem rezultātiem tiek piekārtota valsts „AT”, t.i., Austrija, jo tiek izmantoti sintētiski ģenerēti reāli Austrijas EU-SILC dati. Tieši tā pat kā funkcijā *vardcros* tiek izveidotas primārās

izlases vienības, kas tiek apzīmētas ar PSU, un rādītāji – POV, DEP, LWI un AROPE, izmantojot gadījuma skaitļu ģenerātoru. Tiek izveidoti mainīgie personu skaitam (pielikumā 3 mainīgais „exp”) un personu skaitam, kuru vecums ir zem 60 gadiem (pielikumā 3 mainīgais „exp2”). Funkcijas *vardchanges* izvadīs rezultātus balstoties uz Osiera un Bergera metodoloģiju.

Funkcijas *vardchanges* [7] testēšana šķērsgriezuma summārā tipa rādītāju izmaiņu rezultāti simulatīviem *eusilc* datiem programmā *R* pēc Osiera un Bergera metodoloģijas:

5.2.1. tabula šķērsgriezuma summārā tipa rādītāju izmaiņu izvades rezultāti programmā *R*

Rādītājs	$\hat{\Delta}$	$\hat{s}e_{Osier}$	Ticamības int. _{Osier}	$\hat{s}e_{Berger}$	Ticamības int. _{Berger}
Personu skaits	0	0.000122	[-0.000239; 0.000239]	0	0
Personu skaits, kuru vecums ir zem 60 gadiem	0	0	0	0	0

Rezultāti tabulā 5.2.1 parāda, ka summārais rādītājs, personu skaits pēc Osiera metodoloģijas ir atšķirīgs no nulles, tas ir saistīts ar to, ka simulatīviem datiem tiek izmantoti tikai pirmie 50 ieraksti. Gadījumā, kad tiek izmantota visa simulatīvā *eusilc* datu kopa, tad rezultāti pēc abām metodoloģijām abiem summāriem rādītājiem būs nulles, tas ir saistīts ar to, ka tie ir simulatīvie dati un identiski sakrīt abos viļņos. Simulatīvā piemēra, kas atbilst tabulai 5.2.1 programmas kods ir aplūkojams 4. pielikumā.

Funkcijas *vardchanges* [7] testēšana šķērsgriezuma attiecību tipa rādītāju izmaiņu rezultāti simulatīviem *eusilc* datiem programmā *R* pēc Osiera un Bergera metodoloģijas:

5.2.2. tabula šķērsgriezuma attiecību tipa rādītāju izmaiņu izvades rezultāti programmā *R*

Rādītājs	$\hat{\Delta}$	$\hat{s}e_{Osier}$	Ticamības int. _{Osier}	$\hat{s}e_{Berger}$	Ticamības int. _{Berger}
AROPE	-0.053250	0.019073	[-0.090634; -0.015867]	0.017059	[-0.086686; -0.019814]
DEP	0.106969	0.133272	[-0.154239; 0.368178]	0.119202	[-0.126663; 0.340613]
LWI	-0.160265	0.131744	[-0.418479; 0.097949]	0.117836	[-0.391218; 0.070688]
POV	-0.010627	0.069737	[-0.147310; 0.126055]	0.062375	[-0.132880; 0.111625]

Tabula 5.2.2 parāda funkcijas *vardchanges* [7] rezultātus pēc abām metodoloģijām, kas tika iepriekš aprakstītas. Iegūtie rezultātiem ir attiecību tipa rādītāju izmaiņām, kur ir parādīti kvalitātes rādītāju novērtējumi pēc Osiera un Bergera metodoloģijas. Tiek parādīta novērtētā standartklūda un ticamības intervāls, kas parāda kādā intervālā atrodas patiesā novērtējuma vērtība. Simulatīvā piemēra programmas kods ar funkcijas *vardchanges* pielietojumiem ir aplūkojams 4. pielikumā.

5.3. Funkcijas *vardchanges* programmā *R* testēšanas rezultāti uz reāliem EU-SILC Latvijas datiem

Funkcijas *vardchanges* testēšana uz reāliem datiem ir ekvivalenta ar testēšanu uz simulatīviem datiem, atšķirība ir tāda, ka tiek izmantoti reāli EU-SILC dati un novērtētie kvalitātes rādītāji tiek izvadīti personu un mājsaimniecību līmenī. Dati ir sagatavoti ar SAS programmas palīdzību [12]. Funkcija *vardchanges* izvadīs rezultātus summārā tipa rādītāja skaita izmaiņām gan personu, gan mājsaimniecību līmenī, kā arī attiecību tipa rādītāju AROPE, POV, DEP un LWI izmaiņām EU-SILC Latvijas datos starp 2011 un 2012, kā arī starp 2012 un 2013 gadu. Rezultāti funkcijai *vardchanges* [7] tiks izvadīti pēc Osiera un Bergera metodoloģijas.

Funkcijas *vardchanges* [7] šķērsgriezuma summārā tipa rādītāju izmaiņu rezultāti reāliem Latvijas EU-SILC datiem programmā *R*:

5.3.1. tabula šķērsgriezuma summārā tipa rādītāju izmaiņu izvades rezultāti programmā *R* personu līmenī

Rādītājs	Periods	$\hat{\Delta}$	\hat{se}_{Osier}	Ticamības int. _{Osier}	\hat{se}_{Berger}	Ticamības int. _{Berger}
Skaitis	2011-2012	29365	38868	[-46815; 105545]	38803	[-46687; 105417]
Skaitis	2012-2013	20727	39689	[-57062; 98516]	39618	[-56923; 98377]

5.3.2. tabula šķērsgriezuma summārā tipa rādītāju izmaiņu izvades rezultāti programmā *R* mājsaimniecību līmenī

Rādītājs	Periods	$\hat{\Delta}$	\hat{se}_{Osier}	Ticamības int. _{Osier}	\hat{se}_{Berger}	Ticamības int. _{Berger}
Skaitis	2011-2012	4716	13212	[-21179; 30610]	13190	[-21136; 30567]
Skaitis	2012-2013	4718	13464	[-21671; 31106]	13440	[-21624; 31060]

Funkcijas *vardchanges* [7] šķērsgriezuma attiecību tipa rādītāju izmaiņu rezultāti reāliem Latvijas EU-SILC datiem programmā *R*:

5.3.3. tabula šķērsgriezuma attiecību tipa rādītāju izmaiņu izvades rezultāti programmā *R* personu līmenī

Rādītājs	Periods	$\hat{\Delta}$	\hat{se}_{Osier}	Ticamības int. _{Osier}	\hat{se}_{Berger}	Ticamības int. _{Berger}
AROPE	2011-2012	0.038883	0.013404	[0.012613; 0.065154]	0.013382	[0.012655; 0.065112]
DEP	2011-2012	0.053414	0.014646	[0.024708; 0.082120]	0.014623	[0.024754; 0.082074]
LWI	2011-2012	0.008829	0.019741	[-0.029863; 0.047521]	0.019709	[-0.029801; 0.047459]
POV	2011-2012	-0.002356	0.015749	[-0.033223; 0.028511]	0.015724	[-0.033174; 0.028462]
AROPE	2012-2013	0.010887	0.015166	[-0.018837; 0.040611]	0.015139	[-0.018786; 0.040559]
DEP	2012-2013	0.016399	0.016417	[-0.015779; 0.048576]	0.016389	[-0.015723; 0.048521]
LWI	2012-2013	0.017348	0.021618	[-0.025022; 0.059718]	0.021580	[-0.024948; 0.059644]
POV	2012-2013	-0.001776	0.017646	[-0.036361; 0.032810]	0.017615	[-0.036301; 0.032750]

5.3.4. tabula šķērsgriezuma attiecību tipa rādītāju izmaiņu izvades rezultāti programmā R mājsaimniecību līmenī

Rādītājs	Periods	$\hat{\Delta}$	\hat{s}_{Osier}	Ticamības int. _{Osier}	\hat{s}_{Berger}	Ticamības int. _{Berger}
AROPE	2011-2012	0.027031	0.010710	[0.006040; 0.048021]	0.010693	[0.006074; 0.047988]
DEP	2011-2012	0.048545	0.012011	[0.025005; 0.072086]	0.011992	[0.025042; 0.072049]
LWI	2011-2012	0.010956	0.016174	[-0.020745; 0.042656]	0.016148	[-0.020694; 0.042606]
POV	2011-2012	-0.017712	0.012869	[-0.042935; 0.007510]	0.012849	[-0.042895; 0.007470]
AROPE	2012-2013	0.007848	0.011619	[-0.014925; 0.030621]	0.011599	[-0.014887; 0.030583]
DEP	2012-2013	0.010668	0.013094	[-0.014996; 0.036331]	0.013072	[-0.014952; 0.036287]
LWI	2012-2013	0.021384	0.017649	[-0.013207; 0.055975]	0.017619	[-0.013148; 0.055917]
POV	2012-2013	-0.008779	0.013896	[-0.036014; 0.018456]	0.013872	[-0.035968; 0.018410]

Tabula 5.3.1 un 5.3.2 parāda šķērsgriezuma summārā tipa rādītāju izmaiņas 2011 un 2012 gadam, un 2012 un 2013 gadam EU-SILC Latvijas datiem, bet tabula 5.3.3 un 5.3.4 parāda attiecību tipa šķērsgriezuma rādītāju izmaiņas tiem pašiem datiem. Iegūtie rezultāti ir sadalīti pa tabulām, kas parāda rezultātus personu un mājsaimniecību līmenī. Ir iegūti izmaiņu novērtējumi, un standartklūdas novērtējumi pēc Osiera un Bergera metodoloģijas, kā arī ticamības intervāli, kas parāda kādā intervālā atrodas patiesā vērtība novērtējumam $\hat{\Delta}$.

SECINĀJUMI

Šajā darbā tika teorētiski pamatots šķērsriezuma summārā un attiecību tipa rādītāji un šķērsriezuma rādītāju izmaiņu kvalitātes rādītāju novērtēšana, balstoties uz Osiera un Bergera metodoloģiju. Izmantojot teorētiski pamatoto teoriju tika izstrādātas divas funkcijas programmā *R*, *vardcros*, kas novērtē šķērsriezuma summārā un attiecību tipa rādītāju kvalitātes rādītājus, un funkcija *vardchanges*, kas novērtē šķērsriezuma summārā un attiecību tipa rādītāju izmaiņu kvalitātes rādītājus. Izstrādātās funkcijas tika testētas uz simulatīviem un reāliem datiem. Simulācijām tika izmantoti iebūvēti programmā *R* *eusilc* dati pieģenerējot trūkstošos datus, bet reāliem datiem tika izmantoti EU-SILC Latvijas dati 2011, 2012 un 2013 gadam. Funkciju testēšana uz simulatīviem un reāliem datiem ir ekvivalenta, bet atšķirība ir tikai datos. Rezultāti reāliem datiem tiek nodrošināti personu un mājsaimniecību līmenī. Šķērsriezuma summārie rādītāji EU-SILC Latvijas datiem ir personu skaits un mājsaimniecību skaits, bet attiecību tipa gadījumā AROPE, DEP, LWI un POV.

Rezultātu salīdzināšanai 4. nodaļā tika parādīti kvalitātes rādītāju novērtējumi, izmantojot funkciju *vardom*, kas dod ļoti tuvus rezultātus ar funkciju *vardcros* pēc Osiera metodoloģijas. Ir ļoti svarīgi, lai starp funkcijām *vardcros* un *vardchanges* tiek izmantota vienota metodoloģija, jo funkcija *vardcros* kalpo kā apakšfunkcija funkcijai *vardchanges*, kas sagatavo attiecīgiem periodiem datus un nepieciešamos mainīgos. 5. nodaļā tika parādīti šķērsriezuma rādītāju izmaiņas, izmantojot funkciju *vardchanges* pēc Osiera un Bergera metodoloģijas.

Darbā tika sasniegts darba mērķis, jo visi uzdevumi tika veiksmīgi izpildīti. Darbu ir iespējams turpināt un attīstīt tālāk citos sociālos apsekojumos, kur problēmu risināšanā var tikt izmantotas izstrādātās metodoloģijas un funkcijas, ņemot vērā citu sociālo apsekojumu specifiskāciju. Uz esošā bakalaura darba bāzes ir izstrādāts raksts APLIMAT konferencei un tas ir iesniegts recenzēšanai.

IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] ALPER, M. O., BERGER, Y. G., Variance estimation of change of poverty based upon the Turkish EU-SILC survey, *Journal of Official Statistics, (In Press), 1-14*, 2014, <http://eprints.soton.ac.uk/id/eprint/351065>.
- [2] BERGER, Y. G., OSIER, G., GOEDEMÉ, T., Handbook on Standard error estimation and other related sampling issues in EU-SILC, 2013. <http://www.cros-portal.eu/content/handbook-standard-error-estimation-and-other-related-sampling-issues-ver-29072013>.
- [3] BERGER, Y.G., PRIAM, R., “Estimation of Correlations between Cross-Sectional Estimates from Repeated Surveys - an Application to the Variance of Change”. Proceedings of the 2010 Statistic Canada Symposium, 2010.
- [4] BERGER, Y.G., and PRIAM, R., A simple variance estimator of change for rotating repeated surveys: an application to the EU-SILC household surveys, S3RI Working paper, 2013. <http://eprints.soton.ac.uk/347142/>.
- [5] BERGER, Y. G., and PRIAM, R., Variance estimation of change in rotating repeated surveys, University of Southampton, Southampton Statistical Sciences Research Institute, Southampton, 2014, <https://www.esrc.ac.uk/my-esrc/grants/RES-000-22-3045/outputs/Download/c28fb90e-e8ba-4214-9ffe-439d059921c3>.
- [6] BREIDAKS, J., Precision estimation using ultimate cluster method in software R, Workshop of BNU Network in survey statistics, August 25-28, 2014, Tallinn, Estonia, ISBN 978-9985-74-563-2, <http://www.stat.ee/74630>
- [7] BREIDAKS, J., LIBERTS, M., IVANOVA, S., (2015). vardpoor: Variance Estimation for Sample Surveys by the Ultimate Cluster. R package version 0.2.4.
- [8] Monetārās nabadzības, ienākumu nevienlīdzības indikatori (EU-SILC apsekojums), Centrālās Statistikas pārvalde, 2014, <http://www.csb.gov.lv/statistikas-temas/metodologija/nabadzibas-ienakumu-nevienlidziba-36833.html>.
- [9] DI MEGLIO, E., OSIER, G., GOEDEMÉ, T., BERGER, Y. G., and DI FALCO, E., Standard Error Estimation in EU-SILC – First Results of the Net-SILC2 Project, NTTS (New Techniques and Technologies for Statistics), 2013, http://www.cros-portal.eu/sites/default/files/NTTS2013fullPaper_144.pdf.

- [10] GRAF. M., WENGER, A., and NEDYALKOVA, D., Quality of EU-SILC data, 2011, https://www.unitrier.de/fileadmin/fb4/projekte/SurveyStatisticsNet/Ameli_Delivrables/AMEL_I-WP5-D5.1-20110331.pdf.
- [11] EIROPAS PARLAMENTS UN EIROPAS SAVIENĪBAS PADOME, EIROPAS PARLAMENTA UN PADOMES REGULA (EK) Nr. 1177/2003 par Kopienas statistiku attiecībā uz ienākumiem un dzīves apstākļiem (EU-SILC), 2003. gada 16. jūnijs, Eiropas Savienības Oficiālais Vēstnesis, 16/1. sēj., 32003R1177, L165/1
<http://eur-lex.europa.eu/legal-content/LV/TXT/PDF/?uri=CELEX:32003R1177&from=LV>
- [12] EUROPEAN COMMISSION, ESTAT production channel for EU-SILC datasets - adapted for local SAS Base software, EUROSTAT, Luxembourg, 9 September 2014
- [13] OSIER, G. BERGER, Y., GOEDEME, T., Standard error estimation for the EU-SILC indicators of poverty and social exclusion, Eurostat Methodologies and Working papers, 2013, 7-19, ISBN: 978-92-79-30374-62013.
- [14] OSIER, G., 2009. Variance estimation for complex indicators of poverty and inequality using linearization techniques. *Survey Research Methods*, Vol. 3, No. 3, pp. 167-195.
<http://w4.ub.uni-konstanz.de/srm/article/view/369>.

PIELIKUMI

1. pielikums.

Funkcijas *vardcros* mainīgo apraksts programmā *R*

Y – pētāmais mainīgais, kuram tiks novērtēta dispersija. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā.

H – stratu mainīgais. Ja mainīgais ir viendimensionāls, tad tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā .

PSU – primārās izlases vienības mainīgais. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits datu kopā.

w_final – svaru mainīgais. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits datu kopā.

id – mainīgā id kods. Mājsaimniecību apsekojumā tas ir otrās pakāpes vienību id. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits datu kopā.

Dom – izvēles mainīgais, kas tiek lietots, lai tiktu definēti domēni populācijā. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā.

Z – izvēles mainīgais, kas tiek lietots, ja tiek rēķināta attiecību tipa statistika. Pēc noklusējuma šis mainīgais pieņem nulles vērtību. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā.

country – mainīgais apsekojuma valstīm. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits datu kopā.

period – mainīgais apsekojuma periodam. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits datu kopā.

dataset – izvēles apsekojuma datu kopa.

linratio – loģiskā vērtībā, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad attiecību tipa rādītāju gadījumā mainīgie tiek linearizēti, un tiek rēķināts dispersijas novērtējums, bet ja loģiskā vērtība ir FALSE, tad tiek izmantots gradients dispersija novērtējuma aprēķināšanā.

use.estVar – loģiskā vērtībā, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad *R* funkcija *estVar* tiek izmantota atlikumu kovariāciju matricas novērtēšanai, bet ja loģiskā vērtība ir FALSE, tad *R* funkcija *estVar* netiek izmantota atlikumu kovariāciju matricas novērtēšanai.

household_level_max – loģiskā vērtība, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad vienkāršās gadījuma izlases dispersijas novērtēšanai mājtsaimniecību līmenī tiek ņemta rādītāju maksimālā vērtība, bet ja FALSE, tad tiek ņemta rādītāju summa.

withoutperiod – loģiskā vērtība, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad rezultāti ir ar periodu, bet ja loģiskā vērtība ir FALSE, tad rezultāti tiek izvadīti bez perioda.

netchanges – loģiskā vērtība, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad tiek veidoti divi objekti, pirmais objekts ir agregācija svērtiem datiem pēc perioda, valsts, stratas un primārās izlases vienībām, bet otrs objekts ir novērtējums mainīgam *Y*, dispersija, gradients skaitītājam un saucējam pēc valsts un perioda. Ja loģiskā vērtība ir FALSE, tad abi objekti pieņem nulles vērtību.

confidence – pozitīva vērtība ticamības intervālam. Šī vērtība pēc noklusējuma ir 0.95.

2. pielikums.

**Funkcijas *vardcros* testēšanai sagatavotie simulatīvie *eusilc* dati šķērsriezuma
summārā un attiecību tipa rādītājiem**

```
library(vardpoor)
library(gdata)
library(reshape2)

data(eusilc)
set.seed(1)
data <- data.table(eusilc,
                   year=rep(2010, nrow(eusilc)),
                   country=rep("AT", nrow(eusilc)))
data[, country:=as.character(country)]
data[age<0, age:=0]
PSU <- data[,.N, keyby="db030"]
PSU[, N:=NULL]
PSU[, PSU:=trunc(runif(nrow(PSU), 0, 100))]
data <- merge(data, PSU, by="db030", all=TRUE)
PSU <- eusilc <- 0

data[, strata:="XXXX"]
data[, t_pov:=trunc(runif(nrow(data), 0, 2))]
data[, t_dep:=trunc(runif(nrow(data), 0, 2))]
data[, t_lwi:=trunc(runif(nrow(data), 0, 2))]
data[, exp:= 1]
data[, exp2:= 1 * (age < 60)]

# At-risk-of-poverty (AROP)
data[, pov:= ifelse (t_pov == 1, 1, 0)]
```

```

# Severe material deprivation (DEP)
data[, dep:= ifelse (t_dep == 1, 1, 0)]

# Low work intensity (LWI)
data[, lwi:= ifelse (t_lwi == 1 & exp2 == 1, 1, 0)]

# At-risk-of-poverty or social exclusion (AROPE)
data[, arope:= ifelse (pov == 1 | dep == 1 | lwi == 1, 1, 0)]

result11 <- vardcros(Y=c("exp", "exp2"),
                    H="strata", PSU="PSU", w_final="rb050",
                    id="db030", Dom=NULL, Z=NULL,
                    country="country", periods="year",
                    dataset=data,
                    linratio=FALSE,
                    use.estVar=FALSE,
                    withperiod=TRUE,
                    netchanges=TRUE,
                    confidence = .95)

result11 <- result11$results
result11 <- result11[, c("year", "country",
                        "namesY", "estim", "se", "cv"), with=FALSE]
setnames(result11, c("se", "cv"), c("se1", "cv1"))

result12 <- vardcros(Y= c("exp", "exp2"),
                    H="strata", PSU="PSU", w_final="rb050",
                    id="db030", Dom=NULL, Z=NULL,
                    country="country", periods="year",
                    dataset=data,
                    linratio=FALSE,
                    use.estVar=TRUE,
                    withperiod=TRUE,
                    netchanges=TRUE,
                    confidence = .95)

```

```

result12 <- result12$results
result12 <- result12[, c("year", "country",
                        "namesY", "se", "cv"), with=FALSE]
setnames(result12, c("se", "cv"), c("se2", "cv2"))

result13 <- vardom(Y=c("exp", "exp2"), H="strata",
                  PSU="PSU", w_final="rb050",
                  Dom=NULL, Z=NULL,
                  period=c("country", "year"),
                  dataset=data)

result13 <- result13$all_result
result13 <- result13[, c("year", "country",
                        "variable", "se", "cv"), with=FALSE]
setnames(result13, c("se", "cv"), c("se3", "cv3"))

setkeyv(result11, c("year", "country", "namesY"))
setkeyv(result12, c("year", "country", "namesY"))
setkeyv(result13, c("year", "country", "variable"))

result11 <- merge(result11, result12, all=TRUE)
result11[, variable:=paste("R", get("namesY"), sep="__")]
setkeyv(result11, c("year", "country", "variable"))
result11_sum <- merge(result11, result13, all=TRUE)
result11 <- result12 <- result13 <- NULL

result11 <- vardcros(Y=c("pov", "dep", "lwi", "arope"),
                    H="strata", PSU="PSU", w_final="rb050",
                    id="db030", Dom=NULL,
                    Z=c("exp", "exp", "exp2", "exp"),
                    country="country", periods="year",
                    dataset=data,
                    linratio=FALSE,
                    use.estVar=FALSE,
                    withperiod=TRUE,
                    netchanges=TRUE,

```

```

        confidence = .95)
result11 <- result11$results
result11 <- result11[, c("year", "country",
        "namesY", "namesZ", "estim", "se", "cv"), with=FALSE]
setnames(result11, c("se", "cv"), c("se1", "cv1"))

result12 <- vardcros(Y=c("pov", "dep", "lwi", "arope"),
        H="strata", PSU="PSU", w_final="rb050",
        id="db030", Dom=NULL,
        Z= c("exp", "exp", "exp2", "exp"),
        country="country", periods="year",
        dataset=data,
        linratio=FALSE,
        use.estVar=TRUE,
        withperiod=TRUE,
        netchanges=TRUE,
        confidence = .95)
result12 <- result12$results
result12 <- result12[, c("year", "country",
        "namesY", "namesZ", "se", "cv"), with=FALSE]
setnames(result12, c("se", "cv"), c("se2", "cv2"))

result13 <- vardom(Y=c("pov", "dep", "lwi", "arope"), H="strata",
        PSU="PSU", w_final="rb050",
        Dom=NULL, Z=c("exp", "exp", "exp2", "exp"),
        period=c("country", "year"),
        dataset=data)
result13 <- result13$all_result
result13 <- result13[, c("year", "country",
        "variable", "se", "cv"), with=FALSE]
setnames(result13, c("se", "cv"), c("se3", "cv3"))

setkeyv(result11, c("year", "country", "namesY", "namesZ"))
setkeyv(result12, c("year", "country", "namesY", "namesZ"))
setkeyv(result13, c("year", "country", "variable"))

```

```

result11 <- merge(result11, result12, all=TRUE)
result11[, variable:=paste("R", get("namesY"), get("namesZ"), sep="__")]
setkeyv(result11, c("year", "country", "variable"))
result11_attieciba <- merge(result11, result13, all=TRUE)
result12 <- result13 <- NULL
result11_sum
result11_attieciba

```

3. pielikums.

Funkcijas *vardchanges* mainīgo apraksts programmā R

Y – pētāmais mainīgais, kuram tiks novērtēta dispersija. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

H – stratu mainīgais. Ja mainīgais ir viendimensionāls, tad tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

PSU – primārās izlases vienības mainīgais. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

w_final – svaru mainīgais. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

id – mainīgā id kods. Mājsaimniecību apsekojumā tas ir otrās pakāpes vienību id. Ja mainīgais ir viendimensionāls tas tiek pārveidots par kolonnu *data.frame* tipa, mainīgais var būt arī kā

kolonnu numuri vai loģisks vektors ar vienu patiesu vērtību. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

Dom – izvēles mainīgais, kas tiek lietots, lai tiktu definēti domēni populācijā. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

Z – izvēles mainīgais, kas tiek lietots, ja tiek rēķināta attiecību tipa statistika. Pēc noklusējuma šis mainīgais pieņem nulles vērtību. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

country – mainīgais apsekojuma valstīm. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

periods – mainīgais apsekojuma periodiem. Pārveidojams uz *data.frame* tipu, var būt arī kā kolonnu numuri vai loģisks vektors. Vektora garums šim mainīgam ir tāds pats kā kolonnu skaits attiecīgajā datu kopā (*dataset*).

dataset – izvēles apsekojuma datu kopa.

period1 – raksturo mainīgā periods pirmā perioda vienu rindu.

period2 – raksturo mainīgā periods otrā perioda vienu rindu.

linratio – loģiskā vērtībā, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad attiecību tipa rādītāju gadījumā mainīgie tiek linearizēti, un tiek rēķināts dispersijas novērtējums, bet ja loģiskā vērtība ir FALSE, tad tiek izmantots gradients dispersija novērtējuma aprēķināšanā.

use.estVar – loģiskā vērtībā, kas pieņem divas vērtības TRUE vai FALSE. Ja loģiskā vērtība ir TRUE, tad *R* funkcija *estVar* tiek izmantota atlikumu kovariāciju matricas novērtēšanai, bet ja loģiskā vērtība ir FALSE, tad *R* funkcija *estVar* netiek izmantota atlikumu kovariāciju matricas novērtēšanai.

confidence – pozitīva vērtība ticamības intervālam. Šī vērtība pēc noklusējuma ir 0.95.

4. pielikums.

**Funkcijas *vardchanges* testēšanai sagatavotie simulatīvie *eusilc* dati šķērsriezuma
summārā un attiecību tipa rādītāju izmaiņām**

```
library(vardpoor)

data("eusilc")

set.seed(1)

eusilc1 <- eusilc[1:50,]

set.seed(1)

data <- data.table(rbind(eusilc1, eusilc1),
                   year=c(rep(2010, nrow(eusilc1)),
                          rep(2011, nrow(eusilc1))),
                   country=c(rep("AT", nrow(eusilc1)),
                              rep("AT", nrow(eusilc1))))

data[age<0, age:=0]

PSU <- data[,.N, keyby="db030"]

PSU[, N:=NULL]

PSU[, PSU:=trunc(runif(nrow(PSU), 0, 5))]

setkeyv(PSU, "db030")

setkeyv(data, "db030")

data <- merge(data, PSU, all=TRUE)

PSU <- eusilc <- NULL

data[, strata:=c("XXXX")]
```

```

data[, strata:=as.character(strata)]

data[, t_pov:=trunc(runif(nrow(data), 0, 2))]
data[, t_dep:=trunc(runif(nrow(data), 0, 2))]
data[, t_lwi:=trunc(runif(nrow(data), 0, 2))]

data[, exp:= 1]
data[, exp2:= 1 * (age < 60)]

# At-risk-of-poverty (AROP)
data[, pov:= ifelse (t_pov == 1, 1, 0)]

# Severe material deprivation (DEP)
data[, dep:= ifelse (t_dep == 1, 1, 0)]

# Low work intensity (LWI)
data[, lwi:= ifelse (t_lwi == 1 & exp2 == 1, 1, 0)]

# At-risk-of-poverty or social exclusion (AROPE)
data[, arope:= ifelse (pov == 1 | dep == 1 | lwi == 1, 1, 0)]

result <- vardchanges(Y=c("exp", "exp2"),
  H="strata", PSU="PSU", w_final="rb050",
  id="db030", Dom=NULL, Z=NULL,
  country="country", periods="year",
  dataset=data,
  period1=2010, period2=2011,
  use.estVar=FALSE)

```

```

result1 <- result$changes_results

result1 <- result1[, c("country", "namesY", "estim", "se", "CI_lower",
                      "CI_upper"),
                  with=FALSE]

setnames(result1, c("se", "CI_lower", "CI_upper"),
         c("se_Osier", "Conf_l_Osier", "Conf_u_Osier"))

result <- vardchanges(Y=c("exp", "exp2"),
                    H="strata", PSU="PSU", w_final="rb050",
                    id="db030", Dom=NULL, Z=NULL,
                    country="country", periods="year",
                    dataset=data,
                    period1=2010, period2=2011,
                    use.estVar=TRUE)

result2 <- result$changes_results

result2 <- result2[, c("country", "namesY", "se", "CI_lower",
                      "CI_upper"),
                  with=FALSE]

setnames(result2, c("se", "CI_lower", "CI_upper"),
         c("se_Berger", "Conf_l_Berger",
           "Conf_u_Berger"))

setkeyv(result1, c("country", "namesY"))
setkeyv(result2, c("country", "namesY"))

result_sum <- merge(result1, result2, all=TRUE)

result1 <- result2 <- NULL

result <- vardchanges(Y=c("pov", "dep", "lwi", "arope"),

```

```

H="strata", PSU="PSU",
w_final="rb050",
id="db030", Dom=NULL,
Z=c("exp", "exp", "exp2", "exp"),
country="country", periods="year",
dataset=data,
period1=2010, period2=2011,
use.estVar=FALSE)

```

```
result1 <- result$changes_results
```

```
result1 <- result1[, c("country", "namesY", "namesZ",
"estim", "se", "CI_lower",
```

```
"CI_upper"),
```

```
with=FALSE]
```

```
setnames(result1, c("se", "CI_lower", "CI_upper"),
```

```
c("se_Osier", "Conf_l_Osier", "Conf_u_Osier"))
```

```
result <- vardchanges(Y=c("pov", "dep", "lwi", "arope"),
```

```
H="strata", PSU="PSU",
```

```
w_final="rb050",
```

```
id="db030", Dom=NULL,
```

```
Z=c("exp", "exp", "exp2", "exp"),
```

```
country="country", periods="year",
```

```
dataset=data,
```

```
period1=2010, period2=2011,
```

```
use.estVar=TRUE)
```

```

result2 <- result$changes_results

result2 <- result2[, c("country", "namesY", "namesZ",
                     "se", "CI_lower", "CI_upper"), with=FALSE]

setnames(result2, c("se", "CI_lower", "CI_upper"),
         c("se_Berger", "Conf_l_Berger", "Conf_u_Berger"))

setkeyv(result1, c("country", "namesY", "namesZ"))
setkeyv(result2, c("country", "namesY", "namesZ"))

result_attieciba <- merge(result1, result2, all=TRUE)

result <- result1 <- result2 <- NULL

result_sum
result_attieciba

```

5. pielikums.

Funkcijas *vardcros* R programmas kods

```

vardcros <- function(Y, H, PSU, w_final, id,
                    Dom = NULL,
                    Z = NULL,
                    country, periods,
                    dataset = NULL,
                    linratio = FALSE,
                    use.estVar = FALSE,
                    household_level_max = TRUE,
                    withperiod = TRUE,
                    netchanges = TRUE,
                    confidence = .95) {

  ### Checking
  if (!is.logical(linratio)) stop("'linratio' must be logical")

```

```

if (!is.logical(household_level_max)) stop("'household_level_max' must be logical")
if (!is.logical(withperiod)) stop("'withperiod' must be logical")
if (!is.logical(netchanges)) stop("'netchanges' must be logical")
if (is.null(Z)==linratio & linratio==TRUE) stop("'linratio' must be FALSE")

if(!is.numeric(confidence) || length(confidence) != 1 || confidence[1] < 0 || confidence[1] > 1)
{
  stop("'confidence' must be a numeric value in [0,1]") }

if(!is.null(dataset)) {
  dataset <- data.frame(dataset)
  aY <- Y
  if (min(Y %in% names(dataset))!=1) stop("'Y' does not exist in 'dataset'!")
  if (min(Y %in% names(dataset))==1) {
    Y <- data.frame(dataset[, Y], check.names=FALSE)
    names(Y) <- aY }
  if(!is.null(H)) {
    aH <- H
    if (min(H %in% names(dataset))!=1) stop("'H' does not exist in 'dataset'!")
    if (min(H %in% names(dataset))==1) {
      H <- as.data.frame(dataset[, aH], stringsAsFactors=FALSE)
      names(H) <- aH }}
  if(!is.null(id)) {
    aid <- id
    if (min(id %in% names(dataset))!=1) stop("'id' does not exist in 'dataset'!")
    if (min(id %in% names(dataset))==1) {
      id <- as.data.frame(dataset[, aid], stringsAsFactors=FALSE)
      names(id) <- aid }}
  if(!is.null(PSU)) {
    aPSU <- PSU
    if (min(PSU %in% names(dataset))!=1) stop("'PSU' does not exist in 'dataset'!")
    if (min(PSU %in% names(dataset))==1) {
      PSU <- as.data.frame(dataset[, aPSU], stringsAsFactors=FALSE)
      names(PSU) <- aPSU }}
  if(!is.null(w_final)) {

```

```

aw_final <- w_final
if (min(w_final %in% names(dataset))!=1) stop("'w_final' does not exist in 'dataset'!")
if (min(w_final %in% names(dataset))==1) {
  w_final <- data.frame(dataset[, aw_final])
  names(w_final) <- aw_final }}
if(!is.null(Z)) {
  aZ <- Z
  if (min(Z %in% names(dataset))!=1) stop("'Z' does not exist in 'dataset'!")
  if (min(Z %in% names(dataset))==1) {
    Z <- data.frame(dataset[, aZ], check.names=FALSE,
stringsAsFactors=FALSE)
    names(Z) <- aZ }}
if(!is.null(country)) {
  country2 <- country
  if (min(country %in% names(dataset))!=1) stop("'country' does not exist in 'dataset'!")
  if (min(country %in% names(dataset))==1) country <- as.data.frame(dataset[, country],
stringsAsFactors=FALSE)
  names(country) <- country2 }

if(!is.null( periods)) {
  periods2 <- periods
  if (min( periods %in% names(dataset))!=1) stop("'periods' does not exist in 'dataset'!")
  if (min( periods %in% names(dataset))==1) periods <- data.frame(dataset[, periods],
stringsAsFactors=FALSE)
  names( periods) <- periods2 }

if (!is.null(Dom)) {
  Dom2 <- Dom
  if (min(Dom %in% names(dataset))!=1) stop("'Dom' does not exist in 'data'!")
  if (min(Dom %in% names(dataset))==1) {
    Dom <- as.data.frame(dataset[, Dom2], stringsAsFactors=FALSE)
    names(Dom) <- Dom2 } }
}

# Y

```

```

Y <- data.table(Y, check.names=TRUE)
n <- nrow(Y)
m <- ncol(Y)
if (!all(sapply(Y, is.numeric))) stop("'Y' must be numeric values")
if (any(is.na(Y))) stop("'Y' has unknown values")
if (is.null(names(Y))) stop("'Y' must be colnames")

# H
H <- data.table(H)
if (nrow(H) != n) stop("'H' length must be equal with 'Y' row count")
if (ncol(H) != 1) stop("'H' must be 1 column data.frame, matrix, data.table")
if (any(is.na(H))) stop("'H' has unknown values")
if (is.null(names(H))) stop("'H' must be colnames")

# id
id <- data.table(id)
if (any(is.na(id))) stop("'id' has unknown values")
if (nrow(id) != n) stop("'id' length must be equal with 'Y' row count")
if (ncol(id) != 1) stop("'id' must be 1 column data.frame, matrix, data.table")
if (is.null(names(id)) || (names(id) == "id")) setnames(id, names(id), "h_ID")

# PSU
PSU <- data.table(PSU)
if (any(is.na(PSU))) stop("'PSU' has unknown values")
if (nrow(PSU) != n) stop("'PSU' length must be equal with 'Y' row count")
if (ncol(PSU) != 1) stop("'PSU' has more than 1 column")

# w_final
w_final <- data.frame(w_final)
if (nrow(w_final) != n) stop("'w_final' must be equal with 'Y' row count")
if (ncol(w_final) != 1) stop("'w_final' must be vector or 1 column data.frame, matrix,
data.table")
w_final <- w_final[,1]
if (!is.numeric(w_final)) stop("'w_final' must be numerical")
if (any(is.na(w_final))) stop("'w_final' has unknown values")

```

```

# country
country <- data.table(country)
if (any(is.na(country))) stop("'country' has unknown values")
if (nrow(country) != n) stop("'country' length must be equal with 'Y' row count")
if (ncol(country) != 1) stop("'country' has more than 1 column")
if (!is.character(country[[names(country)]])) stop("'country' must be character")

# periods
if (withperiod) {
  period <- data.table(periods)
  if (any(is.na(periods))) stop("'periods' has unknown values")
  if (nrow(periods) != n) stop("'periods' length must be equal with 'Y' row count")
} else if (!is.null(periods)) stop("'periods' must be NULL for those data")

# Dom
namesDom <- NULL
if (!is.null(Dom)) {
  Dom <- data.table(Dom)
  if (any(duplicated(names(Dom))))
    stop("'Dom' are duplicate column names: ",
         paste(names(Dom)[duplicated(names(Dom))], collapse = ", "))
  if (nrow(Dom) != n) stop("'Dom' and 'Y' must be equal row count")
  if (any(is.na(Dom))) stop("'Dom' has unknown values")
  if (is.null(names(Dom))) stop("'Dom' must be colnames")
  namesDom <- names(Dom)
  Dom <- Dom[, lapply(.SD, as.character), .SDcols = namesDom]
  Dom_agg <- Dom[, .N, keyby=namesDom][, N:=NULL]
  Dom_agg1 <- Dom_agg[, lapply(namesDom, function(x) make.names(paste0(x, ".",
get(x))))]
  Dom_agg1[, Dom := Reduce(function(x, y) paste(x, y, sep="___"), .SD)]
  Dom_agg <- data.table(Dom_agg, Dom_agg1[, "Dom", with=FALSE])
}

```

```

# Z
if (!is.null(Z)) {
  Z <- data.table(Z, check.names=TRUE)
  if (!all(sapply(Z, is.numeric))) stop("'Z' must be numeric values")
  if (nrow(Z) != n) stop("'Z' and 'Y' must be equal row count")
  if (ncol(Z) != m) stop("'Z' and 'Y' must be equal column count")
  if (any(is.na(Z))) stop("'Z' has unknown values")
  if (is.null(names(Z))) stop("'Z' must be colnames")
}

# Calculation

# Domains
n_h <- stratASF <- name1 <- name2 <- nhcor <- n_h <- num1 <- num <- NULL
den1 <- den <- num_den1 <- dom <- name2 <- variable <- name1 <- NULL
names <- num1 <- grad1 <- grad2 <- den1 <- num_den1 <- estim <- NULL
total <- pop_size <- N <- stderr_nw <- sample_size <- sd_nw <- NULL
stderr_w <- sd_w <- se <- rse <- cv <- absolute_margin_of_error <- NULL
relative_margin_of_error <- CI_lower <- CI_upper <- nams <- NULL
totalZ <- Z1 <- namesY <- namesZ <- namesZ1 <- H_sk <- NULL

if (!is.null(Dom)) Y1 <- domain(Y, Dom) else Y1 <- Y

size <- data.table(size=rep(1, nrow(Y1)))
if (!is.null(Dom)) size1 <- domain(size, Dom) else size1 <- copy(size)

namesDom <- names(Dom)
DTp <- copy(country)
if (withperiod) DTp <- data.table(periods, country)
namesperc <- names(DTp)
namesperc2 <- c("period_country", names(DTp))
period_country <- do.call("paste", c(as.list(DTp), sep="_"))
if (!is.null(Y1)) namesY1 <- names(Y1) else namesY1 <- NULL
namesYZ <- namesY1

```

```

nams <- data.table(name1=namesY1)

if (!is.null(Z)) {
  if (!is.null(Dom)) Z1 <- domain(Z, Dom) else Z1 <- Z
  if (!is.null(Z1)) namesZ1 <- names(Z1) else namesZ1 <- NULL
  namesYZ <- c(namesYZ, namesZ1)
  nams <- data.table(nams, nameZ=namesZ1)
  if (linratio){
    sorts <- unlist(split(Y1[, .I], period_country))
    lin1 <- lapply(split(Y1[, .I], period_country),
                  function(i) lin.ratio(Y1[i], Z1[i], w_final[i], Dom=NULL))
    Y2 <- rbindlist(lin1)[sorts]
    if (any(is.na(Y2))) print("Results are calculated, but there are cases where Z = 0")
  } else Y2 <- copy(Y1)

Y2 <- data.table(Y2, size1, check.names=TRUE)
names_size1 <- names(Y2)[-ncol(size1):-1+1+ncol(Y2)]

Y2w <- Y2 * w_final
namesY2 <- names(Y2)

namesY2w <- paste0(namesY2, "w")
setnames(Y2w, namesY2, namesY2w)

DT <- data.table(period_country, DTp, H, PSU, id, w_final, Y2, Y2w)

DTp <- data.table(period_country, w_final*Y1)
if (!is.null(Z)) DTp <- data.table(DTp, w_final*Z1)

DTp <- DTp[, lapply(.SD, sum, na.rm=TRUE), keyby="period_country",
.SDcols=namesYZ]
setnames(DTp, namesYZ, paste0(namesYZ, "_sum"))

```

```

DT1 <- copy(DT)
names_id <- names(id)
names_H <- names(H)
names_PSU <- names(PSU)
names_country <- names(country)
period_country <- ech <- ech1 <- NULL
id <- Dom <- Z1 <- country <- NULL
H <- PSU <- Y1 <- Y2 <- NULL
Y2W <- nh <- nh_cor <- NULL

DT2 <- DT1[, lapply(.SD, sum, na.rm=TRUE),
              keyby=namesperc2,
              .SDcols = c(names_size1, namesY2w)]
setnames(DT2, names(DT2)[-c(1:length(namesperc2))],
         paste0(names(DT2)[-c(1:length(namesperc2))], "s"))

#-----*
# AGGREGATION AT PSU LEVEL ("ULTIMATE CLUSTER" APPROACH) |
#-----*

DT3 <- DT1[, lapply(.SD, sum, na.rm=TRUE), keyby=c(namesperc2,
          names_H, names_PSU), .SDcols = namesY2w]
setnames(DT3, namesY2w, namesY2)
DT1 <- copy(DT3)
DT1[, (c("period_country", paste0(names_size1, ""))):=NULL]
if (!netchanges) DT1 <- NULL

setkeyv(DT2, namesperc2)
DT3 <- merge(DT2, DT3, all=TRUE)

# VECTOR OF THE PARTIAL DERIVATIVES (GRADIENT FUNCTION)

if (!is.null(namesZ1) & !linratio) {
  Z1ws <- paste0(namesZ1, "ws")

```

```

Y1ws <- paste0(namesY1, "ws")

DT4 <- DT3[, lapply(paste0(namesZ1, "ws"), function(z) 1/get(z))]
setnames(DT4, names(DT4), paste0("g1__", namesY1, "___", namesZ1))

DT3 <- cbind(DT3, DT4)

DT4 <- data.table(DT3[, mapply(function(y, z) -get(y)/(get(z))^2, Y1ws, Z1ws)])
setnames(DT4, names(DT4), paste0("g2__", namesY1, "___", namesZ1))
DT3 <- cbind(DT3, DT4)
}

# NUMBER OF PSUs PER STRATUM
setkeyv(DT3, c(namesperc2, names_H))
DT3[, nh:=.N, by=c(namesperc2, names_H)]

#-----*
# MULTIVARIATE REGRESSION APPROACH USING STRATUM DUMMIES AS
REGRESSORS AND |
# STANDARD ERROR ESTIMATION
|
#-----*

DT3H <- DT3[[names_H]]
DT3H <- factor(DT3H)
if (length(levels(DT3H))==1) { DT3[, stratasf:=1]
  DT3H <- "stratasf"
} else { DT3H <- data.table(model.matrix( ~ DT3H-1))
  DT3 <- cbind(DT3, DT3H)
  DT3H <- names(DT3H) }

fits <-lapply(1:length(namesY1), function(i) {
  fitss <- lapply(split(DT3, DT3$period_country), function(DT3c) {
    y <- namesY1[i]

```

```

if (!(is.null(namesZ1)) & !linratio) z <- paste0(",", toString(namesZ1[i])) else
z <- ""

funkc <- as.formula(paste("cbind(", trim(toString(y)), z, "~",
paste(c(-1, DT3H), collapse= "+")))
res1 <- lm(funkc, data=DT3c)

if (use.estVar==TRUE) {res1 <- data.table(crossprod(res1$res))
} else res1 <- data.table(res1$res)
setnames(res1, names(res1)[1], "num")
res1[, namesY1:=y]
if (!(is.null(namesZ1) & !linratio) {
setnames(res1, names(res1)[2], "den")
res1[, namesZ1:=namesZ1[i]]}

if (use.estVar==TRUE) {
setnames(res1, "num", "num1")
if (!(is.null(namesZ1) & !linratio) {
res1[, num_den1:=res1[["den"]][1]]
res1[, den1:=res1[["den"]][2]] }
res1 <- data.table(res1[1], DT3c[1])
} else {
res1 <- data.table(res1, DT3c)
res1[, nhcor:=ifelse(nh==1,1, nh/(nh-1))]
res1[, num1:=nhcor * num * num]
if (!(is.null(namesZ1) & !linratio) {
res1[, num_den1:=nhcor * num * den]
res1[, den1:=nhcor * den * den]
}}
namep <- c("namesY1", "namesZ1")
namep <- namep[namep %in% names(res1)]
varsp <- c("num1", "den1", "num_den1")
varsp <- varsp[varsp %in% names(res1)]
fits <- res1[, lapply(.SD, sum),
keyby=c("period_country",
namesperc, namep),

```

```

        .SDcols=varsp]
      return(fits)
    })
  return(rbindlist(fitss))
})
res <- rbindlist(fits)
a0 <- unlist(lapply(res$namesY1, function(d) regexpr("__", d)[1]))+2
if (!is.null(namesDom)) res[, Dom:=substr(res$namesY1, a0, nchar(res$namesY1))]
a0 <- fits <- DT3H <- NULL

sd1 <- paste0(c(names_size1, namesY2w), "s")
if (!(is.null(namesZ1))&(!linratio)) {
  sd1 <- c(sd1, paste0("g1__", namesY1, "___", namesZ1),
           paste0("g2__", namesY1, "___", namesZ1))}

DT2 <- DT3[, lapply(.SD, mean, na.rm=TRUE),
             keyby="period_country",
             .SDcols = sd1]

main <- melt(DT2[, c(paste0(names_size1, "s"), "period_country"), with=FALSE],
            id="period_country")
main[, variable:=trim(as.character(variable))]
main[nchar(variable)>6, Dom:=substr(variable, 7, nchar(variable)-1)]
if (is.null(namesDom)) npcs <- "period_country" else npcs <- c("period_country", "Dom")
main[, variable:=NULL]
setnames(main, "value", "sample_size")
setkeyv(main, npcs)
setkeyv(res, npcs)
res <- merge(res, main, all.y=TRUE)

main <- melt(DT2[, c(paste0(names_size1, "ws"), "period_country"), with=FALSE],
            id="period_country")
main[, variable:=trim(as.character(variable))]
main[nchar(variable)>6, Dom:=substr(variable, 7, nchar(variable)-2)]
main[, variable:=NULL]

```

```

setnames(main, "value", "pop_size")
setkeyv(main, npcs)
res <- merge(res, main, all=TRUE)

total <- totalY <- totalZ <- NULL
main <- melt(DT2[, c(paste0(namesY1, "ws"), "period_country"), with=FALSE],
id="period_country")
main[, variable:=trim(as.character(variable))]
main[, namesY1:=substr(variable, 1, nchar(variable)-2)]
main[, variable:=NULL]
setnames(main, "value", "total")
setkeyv(main, c("period_country", "namesY1"))
setkeyv(res, c("period_country", "namesY1"))
res <- merge(res, main, all=TRUE)

main <- melt(DTp[, c(paste0(namesY1, "_sum"), "period_country"), with=FALSE],
id="period_country")
main[, variable:=trim(as.character(variable))]
main[, namesY1:=substr(variable, 1, nchar(trim(as.character(variable)))-4)]
main[, variable:=NULL]
setnames(main, "value", "totalY")
setkeyv(main, c("period_country", "namesY1"))
setkeyv(res, c("period_country", "namesY1"))
res <- merge(res, main, all=TRUE)

if (!is.null(res$namesZ1)) {
  main <- melt(DTp[, c(paste0(namesZ1, "_sum"), "period_country"),
with=FALSE], id="period_country")
  main[, variable:=trim(as.character(variable))]
  main[, namesZ1:=substr(variable, 1, nchar(variable)-4)]
  main[, variable:=NULL]
  setnames(main, "value", "totalZ")
  setkeyv(main, c("period_country", "namesZ1"))
  setkeyv(res, c("period_country", "namesZ1"))
  res <- merge(res, main, all=TRUE)
}

```

```

}

res[, names:=namesY1]
if (!is.null(res$namesZ1)) res[, names:=paste0(namesY1,"___",namesZ1)]

if (!is.null(res$namesZ1) & !linratio) {
  main <- melt(DT2[,c(paste0("g1___", namesY1,"___", namesZ1), "period_country"),
with=FALSE], id="period_country")
  main[, variable:=trim(as.character(variable))]
  main[, names:=substr(variable, 5, nchar(variable))]
  main[, variable:=NULL]
  setnames(main, "value", "grad1")
  setkeyv(main, c("period_country", "names"))
  setkeyv(res, c("period_country", "names"))
  res <- merge(res, main, all=TRUE)

  main <- melt(DT2[, c(paste0("g2___", namesY1,"___", namesZ1),
"period_country"), with=FALSE], id="period_country")
  main[, variable:=trim(as.character(variable))]
  main[, names:=substr(variable, 5, nchar(variable))]
  main[, variable:=NULL]
  setnames(main, "value", "grad2")
  setkeyv(main, c("period_country", "names"))
  res <- merge(res, main, all=TRUE)
}

rm(DT2, DT3, DTp)
res[, var:=num1]
if (!is.null(res$totalZ) & !linratio) res[, var:=(grad1*grad1*num1)+
      (grad2*grad2*den1)+
      2*(grad1*grad2*num_den1)]
res[, estim:=totalY]
if (!is.null(res$totalZ)) {res[, estim:=totalY/totalZ]
  res[, namesZ:=namesZ1] }
res[, namesY:=paste0(namesY1, "__")]
res[, namesY:=substr(namesY, 1, regexpr("__", namesY)-1)]

```

```

if (!is.null(namesZ1)) { res[, namesZ:=paste0(namesZ1, "__")]
      res[, namesZ:=substr(namesZ, 1, regexpr("__", namesZ)-1)] }

if (!is.null(namesDom)) {
  setkeyv(res, "Dom")
  setkeyv(Dom_agg, "Dom")
  res <- merge(res, Dom_agg, all.x=TRUE)
}
Dom_agg <- NULL

main <- c(namesperc, namesDom)
if (!is.null(namesDom)) main <- c(main, "Dom")
main <- c(main, "namesY", "namesY1")
if (!is.null(res$namesZ)) main <- c(main, "namesZ", "namesZ1")
main2 <- c(main, "estim", "num1")
if (!is.null(namesZ1) & !linratio) main2 <- c(main2, "den1", "grad1", "grad2")

if (netchanges) { res1 <- res[, main2[!(main2 %in% c("Dom", "namesY1", "namesZ1"))],
with=FALSE]
  } else res1 <- NULL

main <- c(main, "sample_size", "pop_size", "estim", "var")
res <- res[, .N, keyby=main]

#-----*
# DESIGN EFFECT (DEFF) ESTIMATION - VARIANCE UNDER SIMPLE RANDOM
# SAMPLING |
#-----*

# We aggregate the target variables at household level

DTs <- DT[, lapply(.SD, sum, na.rm=TRUE), keyby=c(namesperc2, names_id, "w_final"),
.SDcols = namesY2]
if (household_level_max) {

```

```

DTm <- DT[, lapply(.SD, max, na.rm=TRUE), keyby=c(namesperc2, names_id),
.SDcols = names_size1]
} else {
DTm <- DT[, lapply(.SD, sum, na.rm=TRUE), keyby=c(namesperc2, names_id),
.SDcols = names_size1]
}

```

```

setnames(DTm, names_size1, paste0(names_size1, "m"))

```

```

DTs <- merge(DTs, DTm)

```

```

# Linearised variables

```

```

if ((!is.null(namesZ1))&(!linratio)) {
lin1 <- lapply(split(DTs[, .I], DTs$period_country), function(i)
lin.ratio(Y=DTs[i, namesY1, with=FALSE],
Z=DTs[i, namesZ1, with=FALSE],
weight=DTs[["w_final"]][i], Dom=NULL))
Y2a <- rbindlist(lin1)
setnames(Y2a, names(Y2a), paste0("lin___", namesY1))
DTs <- data.table(DTs, Y2a)
Y2a <- paste0("lin___", namesY1)
} else Y2a <- namesY1

```

```

w_final <- DTs[["w_final"]]

```

```

DTsd <- DTs[, lapply(.SD[, Y2a, with = FALSE], function(x)
sum(w_final*((x-sum(w_final*x)/sum(w_final))^2))/(sum(w_final)-1)),
keyby="period_country"]

```

```

setnames(DTsd, Y2a, paste0("sd_w___", namesY1))

```

```

setkeyv(DTs, "period_country")

```

```

DTs <- merge(DTs, DTsd)

```

```

DTm <- DTs[, lapply(.SD[, paste0(names_size1, "m"), with = FALSE], function(x)
sum(w_final*x, na.rm=TRUE)),
keyby="period_country"]

```

```

setnames(DTm, paste0(names_size1, "m"), paste0("pop_", names_size1))
DTs <- merge(DTs, DTm)

DTsd <- DTs[, lapply(.SD, sd, na.rm=TRUE), keyby="period_country", .SDcols = Y2a]
setnames(DTsd, Y2a, paste0("sd_nw__", namesY1))
DTs <- merge(DTs, DTsd)

DTm <- DTs[, lapply(.SD, sum, na.rm=TRUE), keyby="period_country", .SDcols =
names_size1]
setnames(DTm, names_size1, paste0("samp_", names_size1))
DTs <- merge(DTs, DTm)

DTx <- DTs[, .N, keyby=c(namesperc, paste0("sd_w__", namesY1),
paste0("sd_nw__", namesY1),
paste0("pop_", names_size1),
paste0("samp_", names_size1))]
DTx[, N:=NULL]

main <- melt(DTx[, c(namesperc, paste0("sd_w__", namesY1)), with=FALSE],
id=namesperc)
main[, namesY1:=substr(variable, 7, nchar(trim(as.character(variable))))]
main[, variable:=NULL]
setnames(main, "value", "sd_w")
setkeyv(main, c(namesperc, "namesY1"))
setkeyv(res, c(namesperc, "namesY1"))
res <- merge(res, main, all=TRUE)

main <- melt(DTx[, c(namesperc, paste0("sd_nw__", namesY1)), with=FALSE],
id=namesperc)
main[, namesY1:=substr(variable, 8, nchar(trim(as.character(variable))))]
main[, variable:=NULL]
setnames(main, "value", "sd_nw")
setkeyv(main, c(namesperc, "namesY1"))
setkeyv(res, c(namesperc, "namesY1"))
res <- merge(res, main, all=TRUE)

```

```

main <- melt(DTx[, c(namesperc, paste0("pop_", names_size1)), with=FALSE],
id=namesperc)
if (!is.null(namesDom)) main[, Dom:=substr(variable, 11,
nchar(trim(as.character(variable))))]
main[, variable:=NULL]
setnames(main, "value", "pop")
if (is.null(namesDom)) nds <- namesperc else nds <- c(namesperc, "Dom")
setkeyv(main, nds)
setkeyv(res, nds)
res <- merge(res, main, all.x=TRUE)

main <- melt(DTx[, c(namesperc, paste0("samp_", names_size1)), with=FALSE],
id=namesperc)
if (!is.null(namesDom)) main[, Dom:=substr(variable, 12,
nchar(trim(as.character(variable))))]
main[, variable:=NULL]
setnames(main, "value", "sampl_siz")
setkeyv(main, nds)
setkeyv(res, nds)
res <- merge(res, main, all=TRUE)

res[, stderr_nw:=100*sqrt((1-(sample_size/pop_size))/pop_size * sd_nw *
sd_nw/sample_size )]
res[, stderr_w:=100*sqrt((1-(sample_size/pop_size))/pop_size * sd_w * sd_w/sample_size )]

DT <- DTw <- DTx <- DTs <- DTsd <- sd1 <- nds <-NULL

res[, namesY1:=NULL]
if (!is.null(namesZ1)) res[, namesZ1:=NULL]
res[, N:=NULL]
res[, se:=sqrt(var)]
res[, rse:=se/estim]
res[, cv:=rse*100]

```

```

tsad <- qnorm(0.5*(1+confidence))
res[, absolute_margin_of_error:=tsad*se]
res[, relative_margin_of_error:=tsad*cv]
res[, CI_lower:=estim - tsad*se]
res[, CI_upper:=estim + tsad*se]

main <- namesperc
if (!is.null(namesDom)) main <- c(main, namesDom)
main <- c(main, "namesY")
if (!is.null(res$namesZ)) main <- c(main, "namesZ")
main <- c(main, "sample_size", "pop_size", "estim", "se",
          "var", "rse", "cv", "absolute_margin_of_error",
          "relative_margin_of_error", "CI_lower", "CI_upper",
          "sd_w", "sd_nw", "pop", "sampl_siz", "stderr_nw",
          "stderr_w")
res <- res[, main, with=FALSE]
list(data_net_changes=DT1, var_grad=res1, results=res)
}

```

6. pielikums.

Funkcijas *vardchanges* R programmas kods

```

vardchanges <- function(Y, H, PSU, w_final, id,
                        Dom = NULL, Z = NULL,
                        country, periods,
                        dataset = NULL,
                        period1, period2,
                        linratio = FALSE,
                        use.estVar = FALSE,
                        confidence=0.95) {

### Checking

if (!is.logical(linratio)) stop("'linratio' must be the logical value")
if (!is.logical(use.estVar)) stop("'use.estVar' must be the logical value")

```

```

if(!is.numeric(confidence) || length(confidence) != 1 || confidence[1] < 0 || confidence[1] > 1)
{
  stop("'confidence' must be a numeric value in [0,1]") }

if(!is.null(dataset)) {
  dataset <- data.frame(dataset)
  aY <- Y
  if (min(Y %in% names(dataset))!=1) stop("'Y' does not exist in 'dataset'!")
  if (min(Y %in% names(dataset))==1) {
    Y <- data.frame(dataset[, Y], check.names=FALSE)
    names(Y) <- aY }

  if(!is.null(H)) {
    aH <- H
    if (min(H %in% names(dataset))!=1) stop("'H' does not exist in 'dataset'!")
    if (min(H %in% names(dataset))==1) {
      H <- as.data.frame(dataset[, aH], stringsAsFactors=FALSE)
      names(H) <- aH }}

  if(!is.null(id)) {
    aid <- id
    if (min(id %in% names(dataset))!=1) stop("'id' does not exist in 'dataset'!")
    if (min(id %in% names(dataset))==1) {
      id <- as.data.frame(dataset[, aid], stringsAsFactors=FALSE)
      names(id) <- aid }}

  if(!is.null(PSU)) {
    aPSU <- PSU
    if (min(PSU %in% names(dataset))!=1) stop("'PSU' does not exist in 'dataset'!")
    if (min(PSU %in% names(dataset))==1) {
      PSU <- as.data.frame(dataset[, aPSU], stringsAsFactors=FALSE)
      names(PSU) <- aPSU }}

  if(!is.null(w_final)) {
    aw_final <- w_final
    if (min(w_final %in% names(dataset))!=1) stop("'w_final' does not exist in 'dataset'!")
    if (min(w_final %in% names(dataset))==1) {
      w_final <- data.frame(dataset[, aw_final])

```

```

        names(w_final) <- aw_final } }
if(!is.null(Z)) {
  aZ <- Z
  if (min(Z %in% names(dataset))!=1) stop("'Z' does not exist in 'dataset'!")
  if (min(Z %in% names(dataset))==1) {
    Z <-data.frame(dataset[, aZ], check.names=FALSE,
stringsAsFactors=FALSE)
    names(Z) <- aZ } }
if(!is.null(country)) {
  acountry <- country
  if (min(country %in% names(dataset))!=1) stop("'country' does not exist in 'dataset'!")
  if (min(country %in% names(dataset))==1) country <- data.frame(dataset[, acountry],
stringsAsFactors=FALSE)
  names(country) <- acountry }

if(!is.null( periods)) {
  aperiods <- periods
  if (min( periods %in% names(dataset))!=1) stop("'periods' does not exist in 'dataset'!")
  if (min( periods %in% names(dataset))==1) periods <- data.frame(dataset[, aperiods])
  names(periods) <- aperiods }

if (!is.null(Dom)) {
  aDom <- Dom
  if (min(Dom %in% names(dataset))!=1) stop("'Dom1' does not exist in 'dataset'!")
  if (min(Dom %in% names(dataset))==1) {
    Dom <- as.data.frame(dataset[, aDom], stringsAsFactors=FALSE)
    names(Dom) <- aDom } }
}

# Y
Y <- data.table(Y, check.names=TRUE)
n <- nrow(Y)
m <- ncol(Y)
if (!all(sapply(Y, is.numeric))) stop("'Y' must be numerical")
if (any(is.na(Y))) stop("'Y' has unknown values")

```

```

if (is.null(names(Y))) stop("'Y' must be colnames")

# H
H <- data.table(H)
if (nrow(H) != n) stop("'H' length must be equal with 'Y' row count")
if (ncol(H) != 1) stop("'H' must be 1 column data.frame, matrix, data.table")
if (any(is.na(H))) stop("'H' has unknown values")
if (is.null(names(H))) stop("'H' must be colnames")

# id
id <- data.table(id)
if (any(is.na(id))) stop("'id' has unknown values")
if (nrow(id) != n) stop("'id' length must be equal with 'Y' row count")
if (ncol(id) != 1) stop("'id' must be 1 column data.frame, matrix, data.table")
if (is.null(names(id)) || (names(id) == "id")) setnames(id, names(id), "ID")

# PSU
PSU <- data.table(PSU)
if (any(is.na(PSU))) stop("'PSU' has unknown values")
if (nrow(PSU) != n) stop("'PSU' length must be equal with 'Y' row count")
if (ncol(PSU) != 1) stop("'PSU' has more than 1 column")

# w_final
w_final <- data.frame(w_final)
if (nrow(w_final) != n) stop("'w_final' must be equal with 'Y' row count")
if (ncol(w_final) != 1) stop("'w_final' must be vector or 1 column data.frame, matrix,
data.table")
w_final <- w_final[,1]
if (!is.numeric(w_final)) stop("'w_final' must be numerical")
if (any(is.na(w_final))) stop("'w_final' has unknown values")

# country
country <- data.table(country)
if (any(is.na(country))) stop("'country' has unknown values")
if (nrow(country) != n) stop("'country' length must be equal with 'Y' row count")

```

```

if (ncol(country) != 1) stop("'country' has more than 1 column")
if (ncol(country) != 1) stop("'country' has more than 1 column")
if (!is.character(country[[names(country)]])) stop("'country' must be character")

# periods
periods <- data.table(periods, check.names=TRUE)
if (any(is.na(periods))) stop("'periods' has unknown values")
if (nrow(periods) != n) stop("'periods' length must be equal with 'Y' row count")
if (nrow(periods[,N, by=names(periods)])!=2) stop("'periods' must be two periods")

# Dom
if (!is.null(Dom)) {
  Dom <- data.table(Dom)
  if (any(duplicated(names(Dom))))
    stop("'Dom' are duplicate column names: ",
         paste(names(Dom)[duplicated(names(Dom))], collapse = ","))
  if (nrow(Dom) != n) stop("'Dom' and 'Y' must be equal row count")
  if (any(is.na(Dom))) stop("'Dom' has unknown values")
  if (is.null(names(Dom))) stop("'Dom' must be colnames")
}

namesZ <- NULL
if (!is.null(Z)) {
  Z <- data.table(Z, check.names=TRUE)
  if (nrow(Z) != n) stop("'Z' and 'Y' must be equal row count")
  if (ncol(Z) != m) stop("'Z' and 'Y' must be equal column count")
  if (any(is.na(Z))) stop("'Z' has unknown values")
  if (is.null(names(Z))) stop("'Z' must be colnames")
  namesZ <- names(Z)
}

# period1
period1 <- data.table(t(period1), check.names=TRUE)
if (nrow(period1) != 1) stop("'period1' must be 1 row")

```

```

if (ncol(period1) != ncol(periods)) stop("'period1' column and 'periods' row count must be
equal")
setnames(period1, names(period1), names(periods))
if (any(is.na(period1))) stop("'period1' has unknown values")
setkeyv(period1, names(periods))
periodss <- copy(periods)
periodss[,periodss:=1]
setkeyv(periodss, names(periods))
if (any(is.na(merge(period1, periodss, all.x=TRUE)))) stop("'period1' row must be exist in
'periods'")

```

```

# period2
period2 <- data.table(t(period2), check.names=TRUE)
if (nrow(period2) != 1) stop("'period2' must be 1 row")
if (ncol(period2) != ncol(periods)) stop("'period2' column and 'periods' row count must be
equal")
setnames(period2, names(period2), names(periods))
if (any(is.na(period2))) stop("'period2' has unknown values")
setkeyv(period2, names(periods))
if (any(is.na(merge(period2, periodss, all.x=TRUE)))) stop("'period2' row must be exist in
'periods'")

```

```

data <- vardcros(Y=Y, H=H, PSU=PSU, w_final=w_final,
               id=id, Dom=Dom, Z=Z, country=country,
               periods=periods, dataset=NULL,
               linratio=linratio, use.estVar=use.estVar,
               household_level_max=TRUE,
               withperiod=TRUE, netchanges=TRUE,
               confidence=confidence)

```

```

crosssectional_results <- data$results
Dom <- names(Dom)
H <- names(H)
Y <- names(Y)

```

```

country <- names(country)
np <- ifelse(!is.null(Dom), length(Dom), 0)
np <- 2 + np + as.integer(!is.null(namesZ))
N <- namesY <- PSU <- w_final <- id <- NULL

dataset <- namesYs <- namesZs <- grad1 <- grad2 <- NULL
Z <- rot01 <- rot02 <- stratascf <- name1 <- NULL
num1 <- num1num1 <- den1den1 <- den1 <- num2num2 <- NULL
den2den2 <- den2 <- num1den1 <- num1num2 <- NULL
num2 <- num1den2 <- den1num2 <- den1den2 <- num2den2 <- NULL
C22 <- C12 <- num1_1 <- den1_1 <- num1den1 <- NULL
C11 <- den1den1 <- C13 <- num1_2 <- C14 <- den1_2 <- NULL
C23 <- C24 <- C33 <- C34 <- C44 <- estim <- estim_1 <- NULL
estim_2 <- grad1_1 <- grad1_2 <- CI_upper <- grad2_1 <- NULL
grad2_2 <- se <- rse <- cv <- CI_lower <- absolute_margin_of_error <- NULL
relative_margin_of_error <- NULL

var_grad <- data$var_grad
setkeyv(var_grad, names( periods ))
var_grad1 <- merge( period1, var_grad, all.x=TRUE )
var_grad2 <- merge( period2, var_grad, all.x=TRUE )
var_grad1[, (names( periods )):=NULL]
var_grad2[, (names( periods )):=NULL]
if (!is.null(var_grad1$grad1)){var_grad1[, grad1:=grad1]
      var_grad1[, grad2:=grad2] }

setnames(var_grad1, names(var_grad1)[-c(1:np)], paste0(names(var_grad1)[-c(1:np)], "_1"))
setnames(var_grad2, names(var_grad2)[-c(1:np)], paste0(names(var_grad2)[-c(1:np)], "_2"))
setkeyv(var_grad1, names(var_grad1)[1:np])
setkeyv(var_grad2, names(var_grad2)[1:np])
var_grad <- merge(var_grad1, var_grad2, all=TRUE)
var_grad1 <- var_grad2 <- NULL

data <- data$data_net_changes
setkeyv(data, names( periods ))

```

```

data1 <- merge(period1, data, all.x=TRUE)
data2 <- merge(period2, data, all.x=TRUE)
data1[, (names(periods)):=NULL]
data2[, (names(periods)):=NULL]
nrowv <- nrow(var_grad)

period <- names(data1)[4:ncol(data1)]
setnames(data1, names(data1)[-c(1:3)], paste0(names(data1)[-c(1:3)], "_1"))
setnames(data2, names(data2)[-c(1:3)], paste0(names(data2)[-c(1:3)], "_2"))
period2 <- names(data2)[4:ncol(data2)]
data1[, rot01:=1]
data2[, rot02:=1]
setkeyv(data1, names(data1)[1:3])
setkeyv(data2, names(data2)[1:3])
data <- merge(data1, data2, all=TRUE)
data1 <- data2 <- NULL

recode.NA <- function(DT, cols = seq_len(ncol(DT))) {
  for (j in cols) if (is.numeric(DT[[j]]))
    set(DT, which(is.na(DT[[j]])), j, ifelse(is.integer(DT[[j]]), 0L, 0))
}
recode.NA(data, c(paste0(period, "_1"), paste0(period, "_2"),
  "rot01", "rot02"))

dataH <- data[[H]]
dataH <- factor(dataH)
if (length(levels(dataH))==1) { data[, stratasf:= 1]
  dataH <- "stratasf"
} else { dataH <- data.table(model.matrix(~ dataH-1))
  data <- cbind(data, dataH)
  dataH <- names(dataH) }

den1 <- den2 <- NULL

fit <- lapply(1:nrowv, function(i) {
  fits <- lapply(split(data, data[[country]]), function(DT3c) {

```

```

y1 <- paste0(period[i], "_1")
y2 <- paste0(period[i], "_2")
if (!is.null(namesZ)) {
  z1 <- paste0(",", toString(period[i + nrowv]), "_1")
  z2 <- paste0(",", toString(period[i + nrowv]), "_2")
} else z1 <- z2 <- ""
vect <- c("rot01*", "rot02*", "rot01*rot02*")

funkt <- as.formula(paste0("cbind(", trim(toString(y1)), z1, ", ",
  trim(toString(y2)), z2, ")~-1+",
  paste(t(unlist(lapply(dataH, function(x)
    paste0("rot01*", toString(x), "+",
      "rot02*", toString(x), "+",
      "rot01*rot02*", toString(x))))),
    collapse= "+")))

res <- lm(funkt, data=DT3c)

if (use.estVar) { res <- data.table(estVar(res))
  } else res <- data.table(lm(funkt, data=DT3c)$res)
if (!is.null(namesZ)) {
  setnames(res, names(res), c("num1", "den1", "num2", "den2"))
  res[, namesZ:=period[i + nrowv]]
} else setnames(res, names(res), c("num1", "num2"))

Zvn <- as.integer(!is.null(namesZ))
res [, namesY:=period[i]]

if (use.estVar) {
  res[, num1num1:=res[["num1"]][1]]
  res[, num2num2:=res[["num2"]][2+Zvn]]
  res[, num1num2:=res[["num1"]][2+Zvn]]
  if (!is.null(namesZ)) {
    res[, den1den1:=res[["den1"]][1+Zvn]]
    res[, den2den2:=res[["den2"]][3+Zvn]]
    res[, num1den1:=res[["num1"]][1+Zvn]]

```

```

    res[, num1den2:=res[["num1"]][3+Zvn]]

    res[, den1num2:=res[["den1"]][2+Zvn]]
    res[, den1den2:=res[["den1"]][3+Zvn]]
    res[, num2den2:=res[["num2"]][3+Zvn]] }
  res <- data.table(res[1], DT3c[1])
} else {
  res[, num1num1:=num1 * num1]
  res[, num2num2:=num2 * num2]
  res[, num1num2:=num1 * num2]
  if (!is.null(namesZ)) {
    res[, den1den1:=den1 * den1]
    res[, den2den2:=den2 * den2]
    res[, num1den1:=num1 * den1]
    res[, num1den2:=num1 * den2]
    res[, den1num2:=den1 * num2]
    res[, den1den2:=den1 * den2]
    res[, num2den2:=num2 * den2] }
  res <- data.table(res, DT3c)}

varsp <- c("num1num1", "den1den1",
          "num2num2", "den2den2",
          "num1den1", "num1num2",
          "num1den2", "den1num2",
          "den1den2", "num2den2")
varsp <- varsp[varsp %in% names(res)]
keynames <- c(country, "namesY", "namesZ")
keynames <- keynames[keynames %in% names(res)]
fits <- res[, lapply(.SD, sum), keyby=keynames, .SDcols=varsp]
return(fits)
})
rbindlist(fits)
})
res <- rbindlist(fit)

```

```

set(res, j=country, value=as.character(res[[country]]))
namesYZ <- c("namesY", "namesZ")
namesYZ <- namesYZ[namesYZ %in% names(res)]
setnames(res, namesYZ, paste0(namesYZ, "s"))

if (!is.null(Dom)) {
  var_grad[, paste0(Dom, "_ss"):=lapply(Dom, function(x) paste0(x, ".", get(x)))]
  var_grad[, namesYs:=Reduce(function(x, y)
    paste(x, y, sep = "__"), .SD),
    .SDcols=c("namesY", paste0(Dom, "_ss"))]
  if (!is.null(namesZ)) { var_grad[, namesZs:=Reduce(function(x, y)
    paste(x, y, sep = "__"), .SD),
    .SDcols=c("namesZ", paste0(Dom, "_ss"))]
  }
} else { var_grad[, namesYs:=namesY]
  if (!is.null(namesZ)) var_grad[, namesZs:=namesZ]}

setkeyv(res, c(country, paste0(namesYZ, "s")))
setkeyv(var_grad, c(country, paste0(namesYZ, "s")))
data <- merge(res, var_grad, all=TRUE)
res <- fit <- var_gr <- NULL
data[, namesYs:=NULL]

data[, C11:=num1_1]
data[, C33:=num1_2]
data[, C13:=sqrt(num1_1*num1_2/(num1num1*num2num2))*num1num2]

if (!is.null(namesZ)) {
  data[, namesZs:=NULL]
  data[, C22:=den1_1]
  data[, C44:=den1_2]
  data[, C12:=sqrt(num1_1*den1_1/(num1num1*den1den1))*num1den1]
  data[, C14:=sqrt(num1_1*den1_2/(num1num1*den2den2))*num1den2]
  data[, C23:=sqrt(den1_1*num1_2/(den1den1*num2num2))*den1num2]
  data[, C24:=sqrt(den1_1*den1_2/(den1den1*den2den2))*den1den2]
}

```

```

    data[, C34:=sqrt(num1_2*den1_2/(num2num2*den2den2))*num2den2]
  }

data[, estim:=estim_1 - estim_2]
if (!is.null(namesZ)) {
  data[, var:= (grad1_1 * grad1_1 * C11) +
    (grad1_2 * grad1_2 * C22) +
    (grad2_1 * grad2_1 * C33) +
    (grad2_2 * grad2_2 * C44) +
    2*((grad1_1 * grad1_2 * C12) +
    (grad1_1 * grad2_1 * C13) +
    (grad1_1 * grad2_2 * C14) +
    (grad1_2 * grad2_1 * C23) +
    (grad1_2 * grad2_2 * C24) +
    (grad2_1 * grad2_2 * C34))]
  } else data[, var:=num1_1 + num1_2 - 2 * C13]

data[, se:=sqrt(var)]
data[, rse:=se/estim]
data[, cv:=rse*100]
tsad <- qnorm(0.5*(1+confidence))
data[, absolute_margin_of_error:= tsad * se]
data[, relative_margin_of_error:= tsad * cv]
data[, CI_lower:=estim - tsad*se]
data[, CI_upper:=estim + tsad*se]
changes_results <- data[, c(country, Dom, namesYZ, "estim",
  "var", "se", "rse", "cv",
  "absolute_margin_of_error",
  "relative_margin_of_error",
  "CI_lower", "CI_upper"), with=FALSE]

list(crossectional_results=crossectional_results, changes_results=changes_results)
}

```

Bakalaura darbs „Daudzdimensiju klāsteru dizainu summāro un attiecības tipa un to izmaiņu statistiku kvalitātes rādītāju novērtējumu funkciju izstrāde programmā R un to pielietojums EU-SILC apsekojumā” izstrādāts LU Fizikas un matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Santa Ivanova

Rekomendēju darbu aizstāvēšanai

Vadītājs: Mg. math. Juris Breidaks _____ 07.01.2014.

Recenzents: Dr. math. Natalja Budkina

Darbs iesniegts Matemātikas nodaļā ____01.2015.

Dekāna pilnvarotā persona: vecākā metodiķe Dzintra Holsta

Darbs aizstāvēts Valsts pārbaudījuma komisijas sēdē

____ 01.2015. prot. Nr. _____

Komisijas sekretāre: docente Ingrīda Uljane