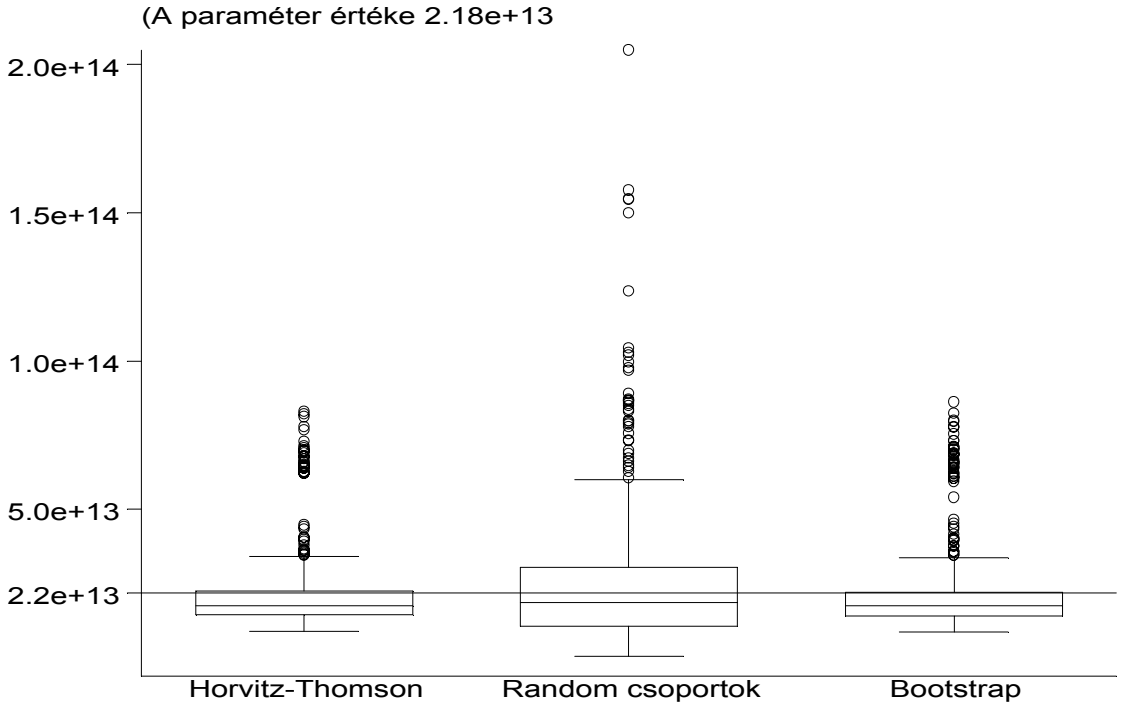
2. melléklet

Tartalomjegyzék

Szimulációs eredmények – grafikonok 1
1. ábra y_1 változó, (d_I, e_1) stratégia 1
2. ábra y_2 változó, (d_I, e_1) stratégia 2
3. ábra y_1 változó, (d_{II}, e_1) stratégia 2
4. ábra y_2 változó, (d_{II}, e_1) stratégia 3
Szimulációs programok, Stata-ban implementálva 3
Előkészítés 3
(d_I, e_1) stratégia 4
(d_I, e_2) stratégia 6
(d_{II}, e_1) stratégia 9
(d_{II}, e_2) stratégia 12
(d_{II}, e_3) stratégia 15

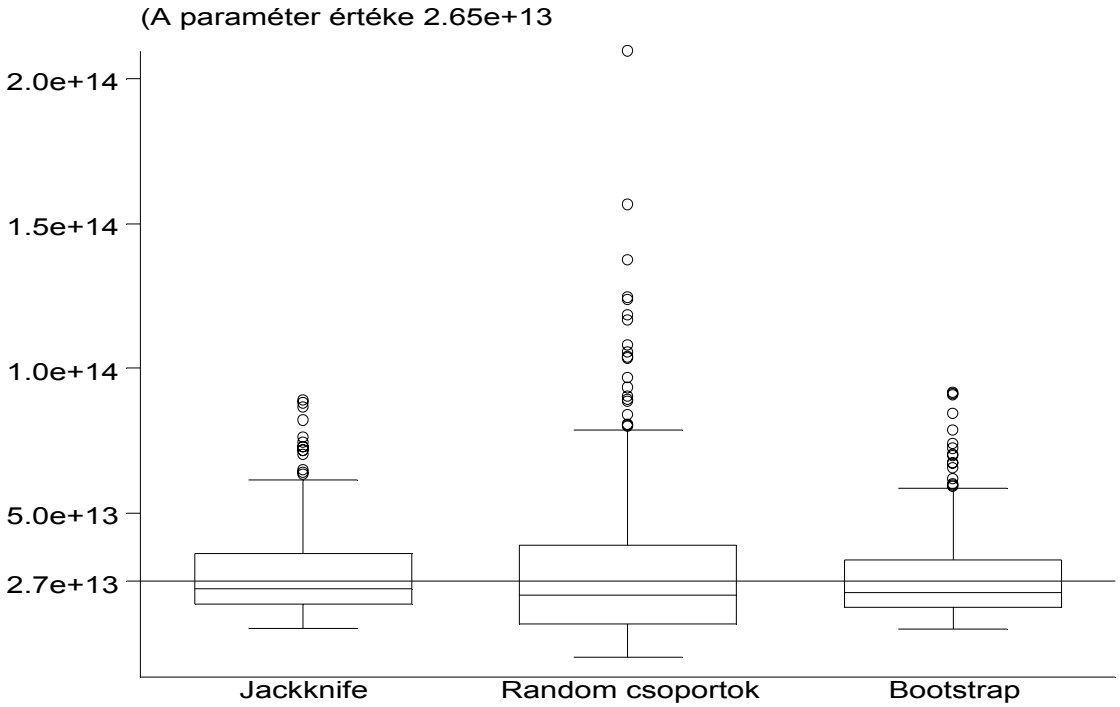
Szimulációs eredmények – grafikonok

1. ábra y_1 változó, (d_I, e_1) stratégia

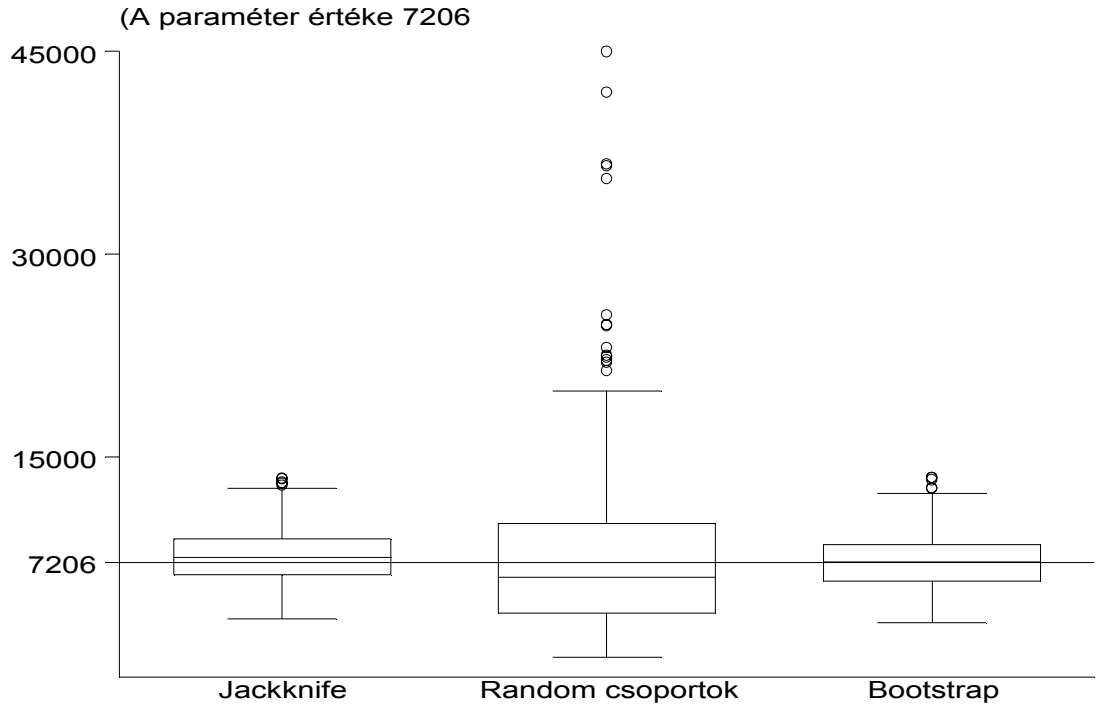


2. ábra y_2 változó, (d_I, e_1) stratégia

3. ábra y_1 változó, (d_{II}, e_1) stratégia



4. ábra y_2 változó, (d_{II}, e_1) stratégia



Szimulációs programok, Stata-ban implementálva

Előkészítés

```
clear
log using c:\nemeth\sim_1.log, replace
set memory 100m
set more off
set log linesize 120
use "C:\nemeth\d08_0.dta", clear

**********előkészítés
***változók előkészítése
   rename HAZON95 haztart
   lab var haztart "háztartás sorszáma"
   rename E4ANEME nem
   gen korcsop=.
   for num 1/3 \ num 55 35 -1: recode korcsop .=X if Y<E4ASZUL
   lab var korcsop "korcsoport"
   lab def korcsop 1 "16-39" 2 "40-59" 3 "60+"
   lab val korcsop korcsop
 ***1. paraméter: összesített havi nettó jövedelem országos szinten
   *a változó generálása
     gen jov=E4JOBECS
```

```
      recode jov 9=.
      lab var jov "havi nettó jövedelme"
 ***2. paraméter: a biztos MSZP-szavazók száma
   *a változó generálása
      gen mszp=1 if  E4XPAR94==4&E4XVALAM==1
      recode mszp .=0 if E4XPAR94~=.|E4XVALAM~=.
      lab def mszp 0 nem 1 igen
      lab val mszp mszp
      lab var mszp "biztos mszp szavazó"
***az adatbázis szűrése azokra, akikről minden információ rendelkezésre áll
   keep if mszp~=.&jov~=.
   save c:\nemeth\d08_1.dta, replace
**********
log close
clear
```

(d_1, e_1) stratégia

```
clear
log using c:\nemeth\sim_11a.log, replace
set memory 500m
set more off
set log linesize 120
use "C:\nemeth\d08_1.dta", clear

**1. paraméter
   *a paraméter valós értéke
     egen par=sum(jov)
     tab par

**********design I.

***a szükséges konstansok:
   *a háztartások száma a populációban
     quietly tab haztart
     scalar NI=r(r)
     di NI
   *a kiválasztott háztartások száma legyen 150.
     scalar nI=150
***a háztartásonkénti összeg számítása
   sort haztart
   egen ti=sum(jov), by(haztart)
***háztartások nagysága
   gen konst=1
   sort haztart
   egen haztnagy=sum(konst), by(haztart)
   lab var haztnagy "háztartás nagyság"


****1. becslés (pi-becslés)

****a populáció háztartásokra redukálása
   preserve
     sort haztart
     quietly by haztart: keep if _n==1
***a becslés varianciájának számítása
```

```
   **háztartások közötti populációs variancia számítása (S^2(tUI))
     quietly sum ti
     scalar S2U=r(Var)*NI/(NI-1)
   **a becslés varianciája:
    scalar V=NI^2*((1-nI/NI)/nI)*S2U
    di V
*szimuláció: 500 ismétlés
   program define sim11
     **a paraméter becslése
          gen unif=uniform()
          sort unif
                    gen minta=1 if _n<151
                    quietly sum ti if minta==1
                    scalar e`1'=NI*r(mean)
         **Yates-Grundy becslés nem adható, a jackknife-becslés megegyezik a HT-val, linearizációnak nincs
értelme a lineáris paraméterfüggvény miatt
          **a variancia Horvitz-Thomson-becslése
                    *háztartások közötti mintabeli variancia számítása (S^2(tsI))
               quietly sum ti if minta==1
               scalar S2s=r(Var)*NI/(NI-1)
        *a becslés varianciája:
               scalar h`1'=NI^2*((1-nI/NI)/nI)*S2s
          **a variancia random csoportok-becslése (A=5)
               gen unif2=uniform()
               sort minta unif2
               gen csoport=1 if _n<31
               for num 2/5 \ num 61 91 121 151: recode csoport .=X if _n<Y
               sort csoport
               egen ta=mean(ti), by(csoport)
               replace ta=. if csoport==.
               replace ta=NI*ta
               replace ta=(1/(5*(5-1)))*(ta-e`1')^2
               scalar r`1'=ta[1]+ta[31]+ta[61]+ta[91]+ta[121]
          **a variancia bootstrap-becslése (A=500 minta a virtuális populációból, visszatevéssel)
                    *a virtuális populáció konstruálása, [NI/nI]=13-szorosára duzzasztva a mintát, így 1950 elemet
kapunk:
                     sort minta
                     gen tivirt=.
                     quietly for num 1/150: replace tivirt=ti[X] if mod(_n,150)==mod(X,150)
                     replace tivirt=. if 1950<_n
               *500 minta visszatevéssel
                     quietly for num 1/500: gen uni=uniform()\sort uni\gen bsminta=1 if _n<151\quietly sum tivirt if
bsminta==1\scalar beX=NI*r(mean)\drop uni bsminta
                    *korrigált bootstrap-becslés a varianciára
                     scalar beatlag=be1
                     quietly for num 2/500:scalar beatlag=beatlag+beX
                     scalar beatlag=beatlag/500
                     quietly for num 1/500: scalar beX=(1/(500-1))*(beX-beatlag)^2
          scalar b`1'=0
                     quietly for num 1/500: scalar b`1'=b`1'+beX
                     *korrekció
                       scalar b`1'=b`1'*(NI-1)*nI/(NI*(nI-1))
         drop unif minta unif2 csoport ta tivirt
   end
  gen est11=.
  gen ht11=.
```

```
 gen rg11=.
 gen bs11=.
 for num 1/500: di X \ quietly sim11 X \ quietly sort CASE \ quietly replace est11=eX if _n==X \ quietly replace
ht11=hX if _n==X \ quietly replace rg11=rX if _n==X \ quietly replace bs11=bX if _n==X

 save c:\nemeth\d08_s11a.dta, replace

 restore

**********
**********
*a becslések eloszlásának jellemzői

        *design I, 1. becslés
        use c:\nemeth\d08_s11a.dta, clear
                sum est11 ht11 rg11 bs11, det
                *95%-os konfidencia-intervallum
                  for var ht11 rg11 bs11: gen Xci=cond(par<est11+1.96*sqrt(X)&est11-
1.96*sqrt(X)<par,1,0)\replace Xci=. if X==.|est11==.\sum Xci

log close
clear
```

(d_I, e_2) stratégia

```
clear
log using c:\nemeth\sim_12a.log, replace
set memory 500m
set more off
set log linesize 120
use "C:\nemeth\d08_1.dta", clear


**1. paraméter
   *a paraméter valós értéke
    egen par=sum(jov)
    tab par

**********design I.

***a szükséges konstansok:
   *a háztartások száma a populációban
    quietly tab haztart
    scalar NI=r(r)
    di NI
   *a kiválasztott háztartások száma legyen 150.
    scalar nI=150
***a háztartásonkénti összeg számítása
   sort haztart
   egen ti=sum(jov), by(haztart)
***háztartások nagysága
   gen konst=1
   sort haztart
   egen haztnagy=sum(konst), by(haztart)
   lab var haztnagy "háztartás nagyság"
```

```
****2. becslés (utólagos rétegezés korcsoportokra)
   *a populációs réteggyakoriság: Nh
    sort korcsop
    egen Nh=sum(konst), by(korcsop)
***a becslés nemlineáris, ezért variaciája és torzítása közvetlenül nem megadható
*szimuláció: 500 ismétlés
   program define sim12
     **a paraméter becslése
                set seed `1'
        gen unif=uniform()
                sort haztart
                quietly by haztart: replace unif=. if _n~=1
        sort unif
                gen mintas=1 if _n<151
                sort haztart
                egen minta=min(mintas), by(haztart)
                quietly sum minta
                scalar n=r(sum)
                *mintabeli réteggyakoriság: nh
                 sort minta korcsop
                 egen nh=sum(konst) if minta==1, by(korcsop)
                *súlyozott változó
                 gen sulyjov=jov*Nh/nh
                *a becslés
                quietly sum sulyjov if minta==1
                scalar e`1'=r(sum)
        **Horvitz-Thomson vagy Yates-Grundy becslés nem adható
        **a variancia linearizációs becslése
                sort korcsop
                egen tsh=sum(jov) if minta==1, by(korcsop)
                for num 1/3: gen iX=1 if korcsop==X&minta==1 \ recode iX .=0 if minta==1
                for num 1/3: gen yX=jov if korcsop==X&minta==1 \ recode yX .=0 if minta==1
                gen ch=Nh/nh
                gen dh=tsh*Nh/nh^2
                gen v=ch*(y1+y2+y3)-dh*(i1+i2+i3)
                sort haztart
                egen vhazt=sum(v), by(haztart)
                replace vhazt=. if minta~=1
                egen vossz =sum(v) if minta==1
                gen seged1=(1-(nI*(NI-1))/((nI-1)*NI))*v*(vossz-vhazt)
                egen seged2=sum(seged1) if minta==1
                gen seged3=(1-nI/NI)*v*vhazt
                egen seged4=sum(seged3) if minta==1
                sort minta
                scalar l`1'=seged2[1]+seged4[1]
        **a variancia jackknife-becslése
                quietly local i=1
                quietly sort minta haztart
                quietly while minta[`i']<. {
                 sort minta haztart
                 egen nha=sum(minta) if haztart~=haztart[`i'], by(korcsop)
                 replace nha=. if minta~=1
                 gen sulyjova=jov*Nh/nha
                 quietly sum sulyjova
                 scalar ta0`i'=r(sum)
```

```
  drop nha sulyjova
  local i=`i'+1
  sort minta haztart
 }
 quietly gen ta0=.
 sort minta haztart
 quietly sum minta
 local n=r(sum)
 quietly for num 1/`n': replace ta0=ta0X if _n==X
 *pszeudoértékek:
  gen ta=nI*e`1'*minta-(nI-1)*ta0
  replace ta=. if mintas~=1
  *a paraméter jackknife-becslése:
  egen tj=mean(ta)
 *a variancia 1. jackknife-becslése:
  gen seged5=(ta-tj)^2
  egen seged6=sum(seged5)
  sort mintas
  scalar j1`1'=seged6[1]/(nI*(nI-1))
 *a variancia 2. jackknife-becslése:
  gen seged7=(ta-e`1')^2
  egen seged8=sum(seged7)
  sort mintas
  scalar j2`1'=seged8[1]/(nI*(nI-1))
**a variancia random csoportok-becslése (A=5)
 gen unif2=uniform()
 sort mintas unif2
 gen csop=1 if _n<31
 for num 2/5 \ num 61 91 121 151: recode csop .=X if _n<Y
      egen csoport=min(csop), by(haztart)
 *csoportbeli réteggyakoriság: csh
       egen csh=sum(konst) if minta==1, by(korcsop csoport)
      *súlyozott változó
       gen sulycsop=jov*Nh/csh
      *a becslés
       for num 1/5: quietly sum sulycsop if csoport==X\ scalar taX=r(sum)
       scalar csest=(ta1+ta2+ta3+ta4+ta5)/5
      scalar r1`1'=0
      for num 1/5: scalar r1`1'=r1`1'+(taX-csest)^2
      scalar r2`1'=0
      for num 1/5: scalar r2`1'=r2`1'+(taX-e`1')^2
 scalar r1`1'=(1/(5*(5-1)))*r1`1'
      scalar r2`1'=(1/(5*(5-1)))*r2`1'
**a variancia bootstrap-becslése (A=500 minta a virtuális populációból, visszatevéssel)
      *a virtuális populáció konstruálása, [NI/nI]=13-szorosára duzzasztva a mintát:
      local tol=_N+1
      local ig=13*`n'
      if _N<`ig' {
            for num `tol'/`ig': set obs X
       }
       sort minta haztart
       gen jovvirt=.
       gen korcvirt=.
       gen haztbs=.
       gen htbs=.
       quietly for num 1/`n': replace jovvirt=jov[X] if mod(_n,`n')==mod(X,`n')
```

```
                quietly for num 1/`n': replace htbs=mintas[X] if mod(_n,`n')==mod(X,`n')
                quietly for num 1/`n': replace haztbs=haztart[X] if mod(_n,`n')==mod(X,`n')
                gen hanyadik=int(_n/(`n'+1))
                quietly for num 1/`n': replace korcvirt=korcsop[X] if mod(_n,`n')==mod(X,`n')
                replace jovvirt=. if `ig'<_n
                replace htbs=. if `ig'<_n
                replace haztbs=. if `ig'<_n
                replace hanyadik=. if `ig'<_n
                replace korcvirt=. if `ig'<_n
        *500 minta visszatevéssel
                quietly for num 1/500: set seed X\ gen uni=uniform()\sort htbs uni\gen bshminta=1 if
_n<151\egen bsminta=min(bshminta), by(haztbs hanyadik)\*mintabeli réteggyakoriság: nhbs\sort bsminta korcvirt\
egen nhbs=sum(konst) if bsminta==1, by(korcvirt)\ *súlyozott változó\ gen sulyjovb=jovvirt*Nh/nhbs \ *a becslés \
quietly sum sulyjovb if bsminta==1\scalar beX=r(sum)\drop uni nhbs sulyjovb bshminta bsminta
                scalar beatlag=be1
                quietly for num 2/500:scalar beatlag=beatlag+beX
                scalar beatlag=beatlag/500
                quietly for num 1/500: scalar beX=(1/(500-1))*(beX-beatlag)^2
        scalar b`1'=0
                quietly for num 1/500: scalar b`1'=b`1'+beX
                sort CASE
                drop if _n>3979
         drop unif-hanyadik
   end
  gen est12=.
  gen lin12=.
  gen jk112=.
  gen jk212=.
  gen rg112=.
  gen rg212=.
  gen bs12=.
  for num 1/500: di X \ quietly sim12 X \ quietly sort CASE \ quietly replace est12=eX if _n==X \ quietly replace
lin12=lX if _n==X \quietly replace jk112=j1X if _n==X \quietly replace jk212=j2X if _n==X \ quietly replace
rg112=r1X if _n==X \ quietly replace rg212=r2X if _n==X \ quietly replace bs12=bX if _n==X

 save c:\nemeth\d08_s12a.dta, replace

**********
*a becslések eloszlásának jellemzői

        *design I, 1. becslés
        use c:\nemeth\d08_s12a.dta, clear
                sum est12 lin12 jk112 jk212 rg112 rg212 bs12, det
                *95%-os konfidencia-intervallum
                 for var lin12 jk112 jk212 rg112 rg212 bs12: gen Xci=cond(par<est12+1.96*sqrt(X)&est12-
1.96*sqrt(X)<par,1,0)\replace Xci=. if X==.|est12==.\sum Xci

log close
clear
***************
```

(d_{II}, e_1) stratégia

```
clear
log using c:\nemeth\sim_21a.log, replace
set memory 500m
```

```
set more off
set log linesize 120
use "C:\nemeth\d08_1.dta", clear


**1. paraméter
   *a paraméter valós értéke
    egen par=sum(jov)
    tab par

**********design II.

***a szükséges konstansok:
   *a háztartások száma a populációban
    quietly tab haztart
    scalar NI=r(r)
    di NI
   *a kiválasztott háztartások száma legyen 225.
    scalar nI=225
***háztartások nagysága
   gen konst=1
   sort haztart
   egen haztnagy=sum(konst), by(haztart)
   lab var haztnagy "háztartás nagyság"

****1. becslés (pi-becslés)
*a becslés varianciája megadható:
          gen jovv=jov*haztnagy*NI/nI
          gen tag2=jovv^2*(nI/(haztnagy*NI)-(nI/(haztnagy*NI))^2)
          egen htossz=sum(jovv), by(haztart)
          gen tag1=-jovv*(nI/(haztnagy*NI))^2*(htossz-jovv)
          gen jovs2=jovv*(1/haztnagy)*((nI*(nI-1))/(NI*(NI-1))-(nI)^2/(NI)^2)
          egen jovs3=sum(jovs2)
          egen jovs4=sum(jovs2), by(haztart)
          gen tag3=(jovs3-jovs4)*jovv*(1/haztnagy)
          gen ossz=tag1+tag2+tag3
          egen var=sum(ossz)
          tab var
*szimuláció: 500 ismétlés
   program define sim21
        **a paraméter becslése
                set seed `1'
          gen unif=uniform()
                sort haztart unif
                gen minta=.
                quietly by haztart: replace minta=1 if _n==1
                gen unif2=uniform()
          sort minta unif2
                replace minta=. if 225<_n
                *a becslés
                quietly sum jovv if minta==1
                scalar e`1'=r(sum)
                ***a másodrendű kiválasztási valószínűségek nemnulla volta nem teljesül, ezért Horvitz-Thomson,
Yates-Grundy, ill. linearizált becslés nem adható
           **a variancia jackknife-becslése
                egen sj1=sum(jovv) if minta==1
```

```
                replace sj1=sj1*nI/(nI-1)
                gen ta0=sj1-jovv*nI/(nI-1)
                *pszeudoértékek:
                  gen ta=nI*e`1'*minta-(nI-1)*ta0
                *a variancia 1. jackknife-becslése (megegyezik 2.-kal):
                  gen seged5=(ta-e`1')^2
                  egen seged6=sum(seged5)
                  sort minta
                  scalar j`1'=seged6[1]/(nI*(nI-1))
          **a variancia random csoportok-becslése (A=5)
             gen unif3=uniform()
             sort minta unif3
             gen csop=1 if _n<46
             for num 2/5 \ num 91 136 181 226: recode csop .=X if _n<Y
                *a becslés
                  for num 1/5: quietly sum jovv if csop==X\ scalar taX=r(sum)*5
                scalar r`1'=0
                for num 1/5: scalar r`1'=r`1'+(taX-e`1')^2
             scalar r`1'=(1/(5*(5-1)))*r`1'
          **a variancia bootstrap-becslése (A=500 minta a virtuális populációból, visszatevéssel)
                *a virtuális populáció konstruálása:
                gen jovvir=jovv if minta==1
                  egen jovvirt=min(jovvir), by(haztart)
                local tol=_N+1
                quietly sum haztnagy if minta==1
                local n=r(sum)
                local ig=`n'*round(NI/nI,1)
                if _N<`ig' {
                        for num `tol'/`ig': set obs X
                 }
                  sort minta haztart
                  gen haztbs=.
                  gen htbs=.
                  quietly for num 1/`n': replace jovvirt=jovvirt[X] if mod(_n,`n')==mod(X,`n')
                  quietly for num 1/`n': replace htbs=minta[X] if mod(_n,`n')==mod(X,`n')
                  replace jovvirt=. if `ig'<_n
                  replace htbs=. if `ig'<_n
                  quietly sum htbs
                  scalar NIbs=r(N)
                  replace jovvirt=jovvirt*NIbs/NI
             *500 minta visszatevéssel
                  quietly for num 1/500: set seed X\ gen uni=uniform()\sort htbs uni\gen bshminta=1 if _n<226\*a
becslés \ quietly sum jovvirt if bshminta==1\scalar beX=r(sum)\drop uni bshminta
                  scalar beatlag=be1
                  quietly for num 2/500:scalar beatlag=beatlag+beX
                  scalar beatlag=beatlag/500
                  quietly for num 1/500: scalar beX=(1/(500-1))*(beX-beatlag)^2
          scalar b`1'=0
                  quietly for num 1/500: scalar b`1'=b`1'+beX
                  sort CASE
                  drop if _n>3979
           drop unif-htbs
   end
  gen est21=.
  gen rg21=.
  gen jk21=.
```

```
  gen bs21=.
  for num 1/500: di X \ quietly sim21 X \ quietly sort CASE \ quietly replace est21=eX if _n==X \ quietly replace
rg21=rX if _n==X \quietly replace jk21=jX if _n==X \ quietly replace bs21=bX if _n==X

 save c:\nemeth\d08_s21a.dta, replace

**********
*a becslések eloszlásának jellemzői

        *design I, 1. becslés
        use c:\nemeth\d08_s21a.dta, clear
                sum est21 jk21 rg21 bs21, det
                *95%-os konfidencia-intervallum
                  for var jk21 rg21 bs21: gen Xci=cond(par<est21+1.96*sqrt(X)&est21-
1.96*sqrt(X)<par,1,0)\replace Xci=. if X==.|est21==.\sum Xci

log close
clear
***************
```

(d_{II}, e_2) stratégia

```
clear
log using c:\nemeth\sim_22a.log, replace
set memory 500m
set more off
set log linesize 120
use "C:\nemeth\d08_1.dta", clear


**1. paraméter
   *a paraméter valós értéke
    egen par=sum(jov)
    tab par

**********design II.

***a szükséges konstansok:
   *a háztartások száma a populációban
    quietly tab haztart
    scalar NI=r(r)
    di NI
   *a kiválasztott háztartások száma legyen 225.
    scalar nI=225
***háztartások nagysága
   gen konst=1
   sort haztart
   egen haztnagy=sum(konst), by(haztart)
   lab var haztnagy "háztartás nagyság"

****2. becslés (utólagos rétegezés korcsoportokra)
   *a populációs réteggyakoriság: Nh
    sort korcsop
    egen Nh=sum(konst), by(korcsop)
*a becslés varianciája nem adható meg.
*szimuláció: 500 ismétlés
```

```
program define sim22
      **a paraméter becslése
                  set seed `1'
        gen unif=uniform()
                  sort haztart unif
                  gen minta=.
                  quietly by haztart: replace minta=1 if _n==1
                  gen unif2=uniform()
        sort minta unif2
                  replace minta=. if 225<_n
                  *mintabeli réteggyakoriság: nh
                   sort minta korcsop
                   egen nh=sum(konst) if minta==1, by(korcsop)
                  *súlyozott változó
                   gen sulyjov=jov*Nh/nh
                  *a becslés
                  quietly sum sulyjov if minta==1
                  scalar e`1'=r(sum)
                  ***a másodrendű kiválasztási valószínűségek nemnulla volta nem teljesül, ezért Horvitz-Thomson,
Yates-Grundy, ill. linearizált becslés nem adható
          **a variancia jackknife-becslése
                  quietly local i=1
                  quietly sort minta haztart
                  quietly while minta[`i']<. {
                   sort minta haztart
                   egen nha=sum(minta) if haztart~=haztart[`i'], by(korcsop)
                   sort minta haztart
                   replace nha=. if minta~=1|haztart==haztart[`i']
                   gen sulyjova=jov*Nh/nha
                   quietly sum sulyjova
                   scalar ta0`i'=r(sum)
                   drop nha sulyjova
                   local i=`i'+1
                   sort minta haztart
                  }
                  quietly gen ta0=.
                  sort minta haztart
                  quietly sum minta
                  local n=r(sum)
                  quietly for num 1/`n': replace ta0=ta0X if _n==X
                  *pszeudoértékek:
                   gen ta=nI*e`1'*minta-(nI-1)*ta0
                   *a paraméter jackknife-becslése:
                   egen tj=mean(ta)
                  *a variancia 1. jackknife-becslése:
                   gen seged5=(ta-tj)^2
                   egen seged6=sum(seged5)
                   sort minta
                   scalar j1`1'=seged6[1]/(nI*(nI-1))
                  *a variancia 2. jackknife-becslése:
                   gen seged7=(ta-e`1')^2
                   egen seged8=sum(seged7)
                   sort minta
                   scalar j2`1'=seged8[1]/(nI*(nI-1))
          **a variancia random csoportok-becslése (A=5)
             gen unif3=uniform()
```

```
sort minta unif3
gen csop=1 if _n<46
for num 2/5 \ num 91 136 181 226: recode csop .=X if _n<Y
*csoportbeli réteggyakoriság: csh
  egen csh=sum(konst) if minta==1, by(korcsop csop)
*súlyozott változó
  gen sulycsop=jov*Nh/csh
*a becslés
  for num 1/5: quietly sum sulycsop if csop==X\ scalar taX=r(sum)
  scalar csest=(ta1+ta2+ta3+ta4+ta5)/5
scalar r1`1'=0
for num 1/5: scalar r1`1'=r1`1'+(taX-csest)^2
scalar r2`1'=0
for num 1/5: scalar r2`1'=r2`1'+(taX-e`1')^2
scalar r1`1'=(1/(5*(5-1)))*r1`1'
  scalar r2`1'=(1/(5*(5-1)))*r2`1'
**a variancia bootstrap-becslése (A=500 minta a virtuális populációból, visszatevéssel)
  *a virtuális populáció konstruálása:
  gen jovvir=jov if minta==1
   egen jovvirt=min(jovvir), by(haztart)
  gen korcvir=korcsop if minta==1
   egen korcvirt=min(korcvir), by(haztart)
  local tol=_N+1
  quietly sum haztnagy if minta==1
  local n=r(sum)
  local ig=`n'*round(NI/nI,1)
  if _N<`ig' {
      for num `tol'/`ig': set obs X
   }
   sort minta haztart
   gen htbs=.
   quietly for num 1/`n': replace jovvirt=jovvirt[X] if mod(_n,`n')==mod(X,`n')
   quietly for num 1/`n': replace htbs=minta[X] if mod(_n,`n')==mod(X,`n')
   quietly for num 1/`n': replace korcvirt=korcsop[X] if mod(_n,`n')==mod(X,`n')
   replace jovvirt=. if `ig'<_n
   replace htbs=. if `ig'<_n
   replace korcvirt=. if `ig'<_n
   quietly sum htbs
   scalar NIbs=r(N)
*500 minta visszatevéssel
   quietly for num 1/500: set seed X\ gen uni=uniform()\sort htbs uni\gen bshminta=1 if
_n<226\*mintabeli réteggyakoriság: nhbs\sort bshminta korcvirt\ egen nhbs=sum(konst) if bshminta==1,
by(korcvirt)\ *súlyozott változó\ gen sulyjovb=jovvirt*Nh/nhbs \ *a becslés \ quietly sum sulyjovb if
bshminta==1\scalar beX=r(sum)\drop uni nhbs sulyjovb bshminta
   scalar beatlag=be1
   quietly for num 2/500:scalar beatlag=beatlag+beX
   scalar beatlag=beatlag/500
   quietly for num 1/500: scalar beX=(1/(500-1))*(beX-beatlag)^2
scalar b`1'=0
   quietly for num 1/500: scalar b`1'=b`1'+beX
   sort CASE
   drop if _n>3979
 drop unif-htbs
 end
gen est22=.
gen rg122=.
```

```
  gen rg222=.
  gen jk222=.
  gen jk122=.
  gen bs22=.
  for num 1/500: di X \ quietly sim22 X \ quietly sort CASE \ quietly replace est22=eX if _n==X \ quietly replace
rg122=r1X if _n==X\ quietly replace rg222=r2X if _n==X \quietly replace jk122=j1X if _n==X\quietly replace
jk222=j2X if _n==X \ quietly replace bs22=bX if _n==X

 save c:\nemeth\d08_s22a.dta, replace

**********
*a becslések eloszlásának jellemzői

        *design I, 1. becslés
        use c:\nemeth\d08_s22a.dta, clear
                sum est22 jk122 jk222 rg122 rg222 bs22, det
                *95%-os konfidencia-intervallum
                  for var jk122 jk222 rg122 rg222 bs22: gen Xci=cond(par<est22+1.96*sqrt(X)&est22-
1.96*sqrt(X)<par,1,0)\replace Xci=. if X==.|est22==.\sum Xci

log close
clear
****************
```

(d_{II}, e_3) stratégia

```
clear
log using c:\nemeth\sim_23a.log, replace
set memory 500m
set more off
set log linesize 120
use "C:\nemeth\d08_1.dta", clear


**1. paraméter
   *a paraméter valós értéke
    egen par=sum(jov)
    tab par

**********design II.

***a szükséges konstansok:
   *a háztartások száma a populációban
    quietly tab haztart
    scalar NI=r(r)
    di NI
   *a kiválasztott háztartások száma legyen 225.
    scalar nI=225
***háztartások nagysága
   gen konst=1
   sort haztart
   egen haztnagy=sum(konst), by(haztart)
   lab var haztnagy "háztartás nagyság"

****2. becslés (pi-becslés, rétegezéssel korcsoportokra)
   *a populációs réteggyakoriság: Nh
```

```
    sort korcsop
    egen Nh=sum(konst), by(korcsop)
*a becslés varianciája nem adható meg.
*szimuláció: 500 ismétlés
  program define sim23
      **a paraméter becslése
                set seed `1'
                gen unif=uniform()
                sort haztart unif
                gen minta=.
                quietly by haztart: replace minta=1 if _n==1
                gen unif2=uniform()
                sort minta unif2
                replace minta=. if 225<_n
                gen pik=(1/haztnagy)*(nI/NI) if minta==1
                gen pikrec=1/pik if minta==1
                *réteggyakoriság súlyozott becslése
                 sort minta korcsop
                 egen Nhest=sum(pikrec) if minta==1, by(korcsop)
                *súlyozott változó
                 gen sulyjov=jov*Nh/(pik*Nhest) if minta==1
                *a becslés
                quietly sum sulyjov if minta==1
                scalar e`1'=r(sum)
                ***a másodrendű kiválasztási valószínűségek nemnulla volta nem teljesül, ezért Horvitz-Thomson,
Yates-Grundy, ill. linearizált becslés nem adható
          **a variancia jackknife-becslése
                gen pikj=pik*(nI-1)/nI if minta==1
                gen pikjrec=1/pikj
                quietly local i=1
                quietly sort minta haztart
                quietly while minta[`i']<. {
                 sort minta haztart
                 egen nha=sum(pikjrec) if haztart~=haztart[`i'], by(korcsop)
                 sort minta haztart
                 replace nha=. if minta~=1|haztart==haztart[`i']
                 gen sulyjova=jov*Nh/(pikj*nha)
                 quietly sum sulyjova
                 scalar ta0`i'=r(sum)
                 drop nha sulyjova
                 local i=`i'+1
                 sort minta haztart
                }
                quietly gen ta0=.
                sort minta haztart
                quietly sum minta
                local n=r(sum)
                quietly for num 1/`n': replace ta0=ta0X if _n==X
                *pszeudoértékek:
                 gen ta=nI*e`1'*minta-(nI-1)*ta0
                 *a paraméter jackknife-becslése:
                 egen tj=mean(ta)
                *a variancia 1. jackknife-becslése:
                 gen seged5=(ta-tj)^2
                 egen seged6=sum(seged5)
                 sort minta
```

```
          scalar j1`1'=seged6[1]/(nI*(nI-1))
         *a variancia 2. jackknife-becslése:
          gen seged7=(ta-e`1')^2
          egen seged8=sum(seged7)
          sort minta
          scalar j2`1'=seged8[1]/(nI*(nI-1))
**a variancia random csoportok-becslése (A=5)
        gen pikrg=pik*45/nI
        gen pikrgrec=1/pikrg
   gen unif3=uniform()
   sort minta unif3
   gen csop=1 if _n<46
   for num 2/5 \ num 91 136 181 226: recode csop .=X if _n<Y
   *csoportbeli becsült réteggyakoriság: csh
         egen csh=sum(pikrgrec) if minta==1, by(korcsop csop)
        *súlyozott változó
         gen sulycsop=jov*Nh/(pikrg*csh)
        *a becslés
         for num 1/5: quietly sum sulycsop if csop==X\ scalar taX=r(sum)
         scalar csest=(ta1+ta2+ta3+ta4+ta5)/5
        scalar r1`1'=0
        for num 1/5: scalar r1`1'=r1`1'+(taX-csest)^2
        scalar r2`1'=0
        for num 1/5: scalar r2`1'=r2`1'+(taX-e`1')^2
   scalar r1`1'=(1/(5*(5-1)))*r1`1'
        scalar r2`1'=(1/(5*(5-1)))*r2`1'
**a variancia bootstrap-becslése (A=500 minta a virtuális populációból, visszatevéssel)
        *a virtuális populáció konstruálása:
        gen jovvir=jov if minta==1
         egen jovvirt=min(jovvir), by(haztart)
        gen korcvir=korcsop if minta==1
         egen korcvirt=min(korcvir), by(haztart)
        gen pikb=pik/((nI/NI)*round(NI/nI,1)) if minta==1
         egen pikvirt=min(pikb), by(haztart)
        gen pikrecb=1/pikb if minta==1
         egen pikrvirt=min(pikrecb), by(haztart)
        local tol=_N+1
        quietly sum haztnagy if minta==1
        local n=r(sum)
        local ig=`n'*round(NI/nI,1)
        if _N<`ig' {
                for num `tol'/`ig': set obs X
         }
         sort minta haztart
         gen htbs=.
         quietly for num 1/`n': replace jovvirt=jovvirt[X] if mod(_n,`n')==mod(X,`n')
         quietly for num 1/`n': replace htbs=minta[X] if mod(_n,`n')==mod(X,`n')
         quietly for num 1/`n': replace korcvirt=korcsop[X] if mod(_n,`n')==mod(X,`n')
         quietly for num 1/`n': replace pikvirt=pikvirt[X] if mod(_n,`n')==mod(X,`n')
         quietly for num 1/`n': replace pikrvirt=pikrvirt[X] if mod(_n,`n')==mod(X,`n')
         replace jovvirt=. if `ig'<_n
         replace htbs=. if `ig'<_n
         replace korcvirt=. if `ig'<_n
         replace pikvirt=. if `ig'<_n
         replace pikrvirt=. if `ig'<_n
   *500 minta visszatevéssel
```

```
                quietly for num 1/500: set seed X\ gen uni=uniform()\sort htbs uni\gen bshminta=1 if
_n<226\*réteggyakoriság súlyozott becslése\sort bshminta korcvirt\egen Nhestb=sum(pikrvirt) if bshminta==1,
by(korcvirt)\*súlyozott változó\gen sulyjovb=jovvirt*Nh/(pikvirt*Nhestb) if bshminta==1\*a becslés\quietly sum
sulyjovb if bshminta==1\scalar beX=r(sum)\drop uni Nhestb sulyjovb bshminta
                scalar beatlag=be1
                quietly for num 2/500:scalar beatlag=beatlag+beX
                scalar beatlag=beatlag/500
                quietly for num 1/500: scalar beX=(1/(500-1))*(beX-beatlag)^2
        scalar b`1'=0
                quietly for num 1/500: scalar b`1'=b`1'+beX
                sort CASE
                drop if _n>3979
        drop unif-htbs
   end
  gen est23=.
  gen rg123=.
  gen rg223=.
  gen jk223=.
  gen jk123=.
  gen bs23=.
  for num 1/500: di X \ quietly sim23 X \ quietly sort CASE \ quietly replace est23=eX if _n==X \ quietly replace
rg123=r1X if _n==X\ quietly replace rg223=r2X if _n==X \quietly replace jk123=j1X if _n==X\quietly replace
jk223=j2X if _n==X \ quietly replace bs23=bX if _n==X

 save c:\nemeth\d08_s23a.dta, replace

**********
*a becslések eloszlásának jellemzői

        *design I, 1. becslés
        use c:\nemeth\d08_s23a.dta, clear
                sum est23 jk123 jk223 rg123 rg223 bs23, det
                *95%-os konfidencia-intervallum
                 for var jk123 jk223 rg123 rg223 bs23: gen Xci=cond(par<est23+1.96*sqrt(X)&est23-
1.96*sqrt(X)<par,1,0)\replace Xci=. if X==.|est23==.\sum Xci

log close
clear
**********
```