New functionalities in R2BEAT: prepareInputToAllocation_beat.1st, beat.1cv and beat.1st

Ilaria Bombelli

2024-04-08

Introduction

This short report is devoted to detailly explain the new functionalities introduced by the functions prepareInputToAllocation_beat.1st, beat.1cv and beat.1st.

Load R2BEAT package and the new functions:

```
# load packages----
library(R2BEAT)
```

```
## Caricamento del pacchetto richiesto: sampling

## Caricamento del pacchetto richiesto: glue

## Caricamento del pacchetto richiesto: parallel

## Caricamento del pacchetto richiesto: foreach

## Caricamento del pacchetto richiesto: doParallel

## Caricamento del pacchetto richiesto: iterators

##

## Please install ReGenesees --> devtools::install_github('DiegoZardetto/ReGenesees')
## Report issues at https://github.com/barcaroli/R2BEAT/issues
## Get a complete documentation on https://barcaroli.github.io/R2BEAT/
```

```
# Definiamo la working directory-----
setwd("S:/Cartelle personali/Funzioni_R_R2BEAT/Aggiornati_Aprile2024")
source("beat.1CV.R")
source("beat.1st.R")
source("prepareInputToAllocation_beat.1st.R")
```

prepareInputToAllocation_beat.1st

The function returns a dataframe, starting from the sampling frame (either universe or sample of a previous survey) with strata information. This output can be used as input dataframe stratif for R2BEAT one-stage sample design (beat.1st).

beat.1cv

- The function returns a dataframe with planned and expected coefficients of variation (CV) in a multivariate multi-domain allocation problem, separately for each domain and for each domain's category.
- It takes as input: a) dataset with strata information; b) dataset with constraints on CV considered during the allocation step; c) vector with a given allocation.

The computation of the expected coefficients of variation is consistent with the column Actual CV of the dataframe sensitivity, one of the outputs of the function beat.1st.

beat.1st

The new functionality concerns uniform and proportional allocations. Whereas in the previous version of the function (currently included in the R2BEAT package), census strata information was not taken into account to obtain the above allocations, in this new version of the function the uniform and proportional allocations are consistent with census strata information. More in details, given the optimal sample size determined by the optimal allocation, the function allocates N_h (population size) units into census strata, while the remaining units (n^{opt}-sum(N_h), ∀ stratum h to be census) are distributed in the nocensus strata according to the uniform and proportional allocations.

In practice

In the following we will import a dataframe and set variables "active","inactive","unemployed","income_hh" as target variables.

```
load("S:/Cartelle personali/Funzioni_R_R2BEAT/sample.RData")
target_vars <- c("active","inactive","unemployed","income_hh")
head(samp)
```

```
##   municipality id_ind region province id_hh stratum stratum_label sex  cl_age
## 1            1     11  north  north_1  H101   12000     north_1_6   2 (24,34]
## 2            1     19  north  north_1  H104   12000     north_1_6   2 (44,54]
## 3            1     34  north  north_1  H109   12000     north_1_6   2 (44,54]
## 4            1     67  north  north_1  H118   12000     north_1_6   1 (44,54]
## 5            1    141  north  north_1  H141   12000     north_1_6   1  (0,14]
## 6            1    166  north  north_1  H147   12000     north_1_6   1  (0,14]
##   active income_hh unemployed inactive  Prob_1st   Prob_2st    Prob_tot weight
## 1      1 17043.123          0        0 0.1867375 0.03234153 0.006039377 165.58
## 2      1 28143.500          0        0 0.1867375 0.03234153 0.006039377 165.58
## 3      0  4791.146          1        0 0.1867375 0.03234153 0.006039377 165.58
## 4      1 28042.687          0        0 0.1867375 0.03234153 0.006039377 165.58
## 5      0 27184.523          0        1 0.1867375 0.03234153 0.006039377 165.58
## 6      0 13731.880          0        1 0.1867375 0.03234153 0.006039377 165.58
##   SR nSR stratum_2
## 1  0   1   12000-2
## 2  0   1   12000-2
## 3  0   1   12000-2
## 4  0   1   12000-2
## 5  0   1   12000-2
## 6  0   1   12000-2
```

Now we will use prepareInputToAllocation_beat.1st function in order to obtain a dataset with strata information from the universe.

```
strata <- prepareInputToAllocation_beat.1st(samp_frame = samp,
                                             ID = "id_hh",
                                             stratum = "stratum_label",
                                             dom = "region",
                                             target = target_vars)
```

```
##
## Calculating strata...
## Computations are being done on population data
##
## DOM1
##   |                                                                              |
## DOM1 category: north
##   |                                                                              |===================
## DOM1 category: center
##   |                                                                              |===================
## DOM1 category: south
##   |                                                                              |===================
## Number of strata:  24
## ... of which with only one unit:  0
```

```
head(strata)
```

```
##             STRATUM stratum_label DOM1  DOM2    N        M1        M2
## north_1_3 north_1_3     north_1_3    1 north  557 0.7540395 0.2226212
## north_1_4 north_1_4     north_1_4    1 north  587 0.7938671 0.1890971
## north_1_5 north_1_5     north_1_5    1 north 1300 0.7753846 0.2046154
## north_1_6 north_1_6     north_1_6    1 north  400 0.7650000 0.2100000
## north_2_3 north_2_3     north_2_3    1 north  703 0.7681366 0.1977240
## north_2_4 north_2_4     north_2_4    1 north  577 0.7400347 0.2305026
##                   M3       M4        S1        S2        S3       S4 CENS COST
## north_1_3 0.02333932 27867.90 0.4306552 0.4160060 0.1509788 22118.92    0    1
## north_1_4 0.01703578 29380.88 0.4045270 0.3915857 0.1294046 26891.68    0    1
## north_1_5 0.02000000 28161.63 0.4173288 0.4034203 0.1400000 41217.60    0    1
## north_1_6 0.02500000 24152.16 0.4239988 0.4073082 0.1561249 16278.73    0    1
## north_2_3 0.03413940 28488.83 0.4220223 0.3982829 0.1815872 22492.04    0    1
## north_2_4 0.02946274 24342.36 0.4386153 0.4211545 0.1690996 15365.51    0    1
```

Now, we will specify the constraints on the coefficients of variation.

```
cv <- as.data.frame(list(DOM = c("DOM1","DOM2"),
                         CV1 = c(0.05,0.10),
                         CV2 = c(0.05,0.10),
                         CV3 = c(0.05,0.10),
                         CV4 = c(0.05,0.10)))
cv
```

```
##    DOM  CV1  CV2  CV3  CV4
## 1 DOM1 0.05 0.05 0.05 0.05
## 2 DOM2 0.10 0.10 0.10 0.10
```

Now that the inputs to run beat.1st are ready, we can run beat.1st (currently implemented in the R2BEAT package):

```
allocation <- R2BEAT::beat.1st(stratif=strata,errors=cv)
```

Among the available outputs, we can use the optimal allocation in order to compute the expected CV with the function beat.1cv.

```
out=beat.1cv(strata,cv,allocation$n)
```

It is possible to notice that the expected CV are consistent with the ones returned by beat.1st in the sensitivity output.

```
df= cbind(out, allocation$sensitivity$`Actual CV`)
colnames(df)=c(colnames(out), "Actual CV (sensititivy output)")
df
```

```
##    TYPE DOMAIN/VAR PLANNED_CV  ACTUAL_CV Actual CV (sensititivy output)
## 1  DOM1       1/V1       0.05     0.0097                         0.0097
## 2  DOM1       1/V2       0.05     0.0290                         0.0290
## 3  DOM1       1/V3       0.05     0.0497                         0.0497
## 4  DOM1       1/V4       0.05     0.0145                         0.0145
## 5  DOM2  center/V1       0.10     0.0095                         0.0095
## 6  DOM2  center/V2       0.10     0.0336                         0.0336
## 7  DOM2  center/V3       0.10     0.0999                         0.0999
## 8  DOM2  center/V4       0.10     0.0177                         0.0177
## 9  DOM2   north/V1       0.10     0.0265                         0.0265
## 10 DOM2   north/V2       0.10     0.0645                         0.0645
## 11 DOM2   north/V3       0.10     0.0995                         0.0995
## 12 DOM2   north/V4       0.10     0.0295                         0.0295
## 13 DOM2   south/V1       0.10     0.0522                         0.0522
## 14 DOM2   south/V2       0.10     0.0919                         0.0919
## 15 DOM2   south/V3       0.10     0.0688                         0.0688
## 16 DOM2   south/V4       0.10     0.0377                         0.0377
```

Now, we can introduce some census strata.

```
strata2=strata
strata2$CENS[c(2, 7, 10)] <- 1 #stratum 2, 7, 10 to be census
```

and we can apply the function beat.1st (currently implemented in the R2BEAT package) in order to determine the optimal, proportional and uniform allocations.

```
alloc2 <- R2BEAT::beat.1st(strata2, cv)
alloc2$alloc
```

```
##                STRATUM ALLOC       PROP    EQUAL
## north_1_3    north_1_3   146  254.11167 179.0833
## north_1_4    north_1_4   587  267.79811 179.0833
## north_1_5    north_1_5   316  593.07929 179.0833
## north_1_6    north_1_6   109  182.48594 179.0833
## north_2_3    north_2_3   222  320.71903 179.0833
```

```
## north_2_4   north_2_4   170  263.23596  179.0833
## north_2_5   north_2_5  1361  620.90840  179.0833
## north_2_6   north_2_6   315  410.59336  179.0833
## center_1_3 center_1_3    68  130.47744  179.0833
## center_1_4 center_1_4   452  206.20911  179.0833
## center_1_5 center_1_5    48   91.24297  179.0833
## center_1_6 center_1_6    20   45.62148  179.0833
## center_2_3 center_2_3    65   99.91105  179.0833
## center_2_4 center_2_4    28   45.62148  179.0833
## center_2_5 center_2_5    32   45.62148  179.0833
## center_2_6 center_2_6    29   45.62148  179.0833
## south_1_3   south_1_3    29   70.25709  179.0833
## south_1_4   south_1_4    45   91.24297  179.0833
## south_1_5   south_1_5    76  159.67519  179.0833
## south_1_6   south_1_6    47   91.24297  179.0833
## south_2_3   south_2_3    26   57.02685  179.0833
## south_2_4   south_2_4    36   68.43223  179.0833
## south_2_5   south_2_5    47   91.24297  179.0833
## south_2_6   south_2_6    24   45.62148  179.0833
##                 Total  4298 4298.00000 4298.0000
```

We can notice that while the optimal allocation takes into account the census strata information, the proportional and the uniform allocations do not. Indeed:

```
df=cbind(alloc2$file_strata$STRATUM,  alloc2$file_strata$N,
         alloc2$file_strata$CENS,alloc2$alloc[-nrow(alloc2$alloc),-1])
colnames(df)=c("STRATUM", "N", "CENS",
               "ALLOC","PROP","EQUAL")
df[order(df$CENS, decreasing=TRUE),]
```

```
##               STRATUM    N CENS ALLOC      PROP    EQUAL
## north_1_4   north_1_4  587    1   587 267.79811 179.0833
## north_2_5   north_2_5 1361    1  1361 620.90840 179.0833
## center_1_4 center_1_4  452    1   452 206.20911 179.0833
## north_1_3   north_1_3  557    0   146 254.11167 179.0833
## north_1_5   north_1_5 1300    0   316 593.07929 179.0833
## north_1_6   north_1_6  400    0   109 182.48594 179.0833
## north_2_3   north_2_3  703    0   222 320.71903 179.0833
## north_2_4   north_2_4  577    0   170 263.23596 179.0833
## north_2_6   north_2_6  900    0   315 410.59336 179.0833
## center_1_3 center_1_3  286    0    68 130.47744 179.0833
## center_1_5 center_1_5  200    0    48  91.24297 179.0833
## center_1_6 center_1_6  100    0    20  45.62148 179.0833
## center_2_3 center_2_3  219    0    65  99.91105 179.0833
## center_2_4 center_2_4  100    0    28  45.62148 179.0833
## center_2_5 center_2_5  100    0    32  45.62148 179.0833
## center_2_6 center_2_6  100    0    29  45.62148 179.0833
## south_1_3   south_1_3  154    0    29  70.25709 179.0833
## south_1_4   south_1_4  200    0    45  91.24297 179.0833
## south_1_5   south_1_5  350    0    76 159.67519 179.0833
## south_1_6   south_1_6  200    0    47  91.24297 179.0833
## south_2_3   south_2_3  125    0    26  57.02685 179.0833
## south_2_4   south_2_4  150    0    36  68.43223 179.0833
```

```
## south_2_5   south_2_5  200    0    47   91.24297 179.0833
## south_2_6   south_2_6  100    0    24   45.62148 179.0833
```

In order to fix this issue, the new function beat.1st has been implemented By applying it and by comparing its output (suffix ".new") with the previous output (suffix ".old"), we can notice that all the allocations are consistent with census strata information.

```r
base::detach("package:R2BEAT", unload = TRUE)
#NEW FUNCTION
alloc3 <- beat.1st(strata2, cv)
alloc3$alloc
```

```
##                STRATUM ALLOC       PROP      EQUAL
## north_1_3   north_1_3   146  150.57485   90.38095
## north_1_4   north_1_4   587  587.00000  587.00000
## north_1_5   north_1_5   316  351.43142   90.38095
## north_1_6   north_1_6   109  108.13274   90.38095
## north_2_3   north_2_3   222  190.04330   90.38095
## north_2_4   north_2_4   170  155.98148   90.38095
## north_2_5   north_2_5  1361 1361.00000 1361.00000
## north_2_6   north_2_6   315  243.29868   90.38095
## center_1_3 center_1_3    68   77.31491   90.38095
## center_1_4 center_1_4   452  452.00000  452.00000
## center_1_5 center_1_5    48   54.06637   90.38095
## center_1_6 center_1_6    20   27.03319   90.38095
## center_2_3 center_2_3    65   59.20268   90.38095
## center_2_4 center_2_4    28   27.03319   90.38095
## center_2_5 center_2_5    32   27.03319   90.38095
## center_2_6 center_2_6    29   27.03319   90.38095
## south_1_3   south_1_3    29   41.63111   90.38095
## south_1_4   south_1_4    45   54.06637   90.38095
## south_1_5   south_1_5    76   94.61615   90.38095
## south_1_6   south_1_6    47   54.06637   90.38095
## south_2_3   south_2_3    26   33.79148   90.38095
## south_2_4   south_2_4    36   40.54978   90.38095
## south_2_5   south_2_5    47   54.06637   90.38095
## south_2_6   south_2_6    24   27.03319   90.38095
##                  Total  4298 4298.00000 4298.00000
```

```r
# comparison
df=cbind(alloc2$file_strata$STRATUM,  alloc2$file_strata$N,
         alloc2$file_strata$CENS,
         alloc2$alloc[-nrow(alloc2$alloc),-1], alloc3$alloc[-nrow(alloc3$alloc), -1])
colnames(df)=c("STRATUM", "N", "CENS",
               "ALLOC.old","PROP.old","EQUAL.old",
               "ALLOC.new","PROP.new","EQUAL.new")
df[order(df$CENS, decreasing=TRUE),]
```

```
##                STRATUM    N CENS ALLOC.old  PROP.old EQUAL.old ALLOC.new
## north_1_4   north_1_4  587    1       587 267.79811  179.0833       587
## north_2_5   north_2_5 1361    1      1361 620.90840  179.0833      1361
## center_1_4 center_1_4  452    1       452 206.20911  179.0833       452
## north_1_3   north_1_3  557    0       146 254.11167  179.0833       146
```

```
## north_1_5   north_1_5 1300    0       316 593.07929  179.0833        316
## north_1_6   north_1_6  400    0       109 182.48594  179.0833        109
## north_2_3   north_2_3  703    0       222 320.71903  179.0833        222
## north_2_4   north_2_4  577    0       170 263.23596  179.0833        170
## north_2_6   north_2_6  900    0       315 410.59336  179.0833        315
## center_1_3 center_1_3  286    0        68 130.47744  179.0833         68
## center_1_5 center_1_5  200    0        48  91.24297  179.0833         48
## center_1_6 center_1_6  100    0        20  45.62148  179.0833         20
## center_2_3 center_2_3  219    0        65  99.91105  179.0833         65
## center_2_4 center_2_4  100    0        28  45.62148  179.0833         28
## center_2_5 center_2_5  100    0        32  45.62148  179.0833         32
## center_2_6 center_2_6  100    0        29  45.62148  179.0833         29
## south_1_3   south_1_3  154    0        29  70.25709  179.0833         29
## south_1_4   south_1_4  200    0        45  91.24297  179.0833         45
## south_1_5   south_1_5  350    0        76 159.67519  179.0833         76
## south_1_6   south_1_6  200    0        47  91.24297  179.0833         47
## south_2_3   south_2_3  125    0        26  57.02685  179.0833         26
## south_2_4   south_2_4  150    0        36  68.43223  179.0833         36
## south_2_5   south_2_5  200    0        47  91.24297  179.0833         47
## south_2_6   south_2_6  100    0        24  45.62148  179.0833         24
##              PROP.new  EQUAL.new
## north_1_4   587.00000  587.00000
## north_2_5  1361.00000 1361.00000
## center_1_4  452.00000  452.00000
## north_1_3   150.57485   90.38095
## north_1_5   351.43142   90.38095
## north_1_6   108.13274   90.38095
## north_2_3   190.04330   90.38095
## north_2_4   155.98148   90.38095
## north_2_6   243.29868   90.38095
## center_1_3   77.31491   90.38095
## center_1_5   54.06637   90.38095
## center_1_6   27.03319   90.38095
## center_2_3   59.20268   90.38095
## center_2_4   27.03319   90.38095
## center_2_5   27.03319   90.38095
## center_2_6   27.03319   90.38095
## south_1_3    41.63111   90.38095
## south_1_4    54.06637   90.38095
## south_1_5    94.61615   90.38095
## south_1_6    54.06637   90.38095
## south_2_3    33.79148   90.38095
## south_2_4    40.54978   90.38095
## south_2_5    54.06637   90.38095
## south_2_6    27.03319   90.38095
```