Basic direct and indirect estimators in sae package

Isabel Molina*, Yolanda Marhuenda

March, 2015

This document presents design unbiased direct estimators and simple indirect estimators of domain means $\bar{Y}_d$, $d = 1, \ldots, D$. For a general random sampling without replacement within each domain U_d. We denote by π_{dj} the inclusion probability of j-th unit from d-th domain in the corresponding domain sample s_d and $w_{dj} = \pi_{dj}^{-1}$ is the corresponding sampling weight. A design-unbiased direct estimator of $\bar{Y}_d$ is the Horvitz-Thompson (HT) estimator, given by

$$\hat{\bar{Y}}_d^{DIR} = N_d^{-1} \sum_{j \in s_d} w_{dj} Y_{dj}. \tag{1}$$

Unbiased estimation of the sampling variance of the HT estimator requires availability of the second order inclusion probabilities $\pi_{d,jk}$ of each pair of units j and k in s_d. A simple approximation that avoids the use of second order inclusion probabilities is obtained by considering $\pi_{d,jk} \approx \pi_{dj}\pi_{dk}$ and is given by

$$\hat{V}_\pi(\hat{\bar{Y}}_d^{DIR}) = \frac{1}{N_d^2} \sum_{j \in s_d} w_{dj}(w_{dj} - 1) Y_{dj}^2. \tag{2}$$

Under Poisson sampling, $\pi_{d,jk} = \pi_{dj}\pi_{dk}$ and in that case the estimator in (2) is exactly unbiased. Under simple random sampling (SRS) without replacement within each area U_d, $d = 1, \ldots, D$, the HT estimator of the mean $\bar{Y}_d$ is the usual sample mean $\hat{\bar{Y}}_d = \bar{y}_d = n_d^{-1} \sum_{j \in s_d} Y_{dj}$, and the (exactly) unbiased estimator of the sampling variance is $\hat{V}_\pi(\hat{\bar{Y}}_d^{DIR}) = (1 - f_d) S_d^2 / n_d$, for $S_d^2 = \sum_{j \in s_d} (Y_{dj} - \bar{y}_d)^2 / (n_d - 1)$.

When the sampling is with replacement within each domain U_d, and units are selected with probabilities P_{dj}, $j = 1, \ldots, N_d$, proportional to some size measure, if we define new weights $w_{dj} = (n_d P_{dj})^{-1}$, the estimator defined in (1) remains unbiased and the unbiased estimator of the sampling variance is given by

$$\hat{V}_\pi(\hat{\bar{Y}}_d^{DIR}) = \frac{1}{n_d} \sum_{j \in s_d} \left(f_d w_{dj} Y_{dj} - \hat{\bar{Y}}_d \right)^2,$$

*Department of Statistics, Universidad Carlos III de Madrid. Address: C/Madrid 126, 28903 Getafe (Madrid), Spain, Tf: +34 916249887, Fax: +34 916249849, E-mail: isabel.molina@uc3m.es

which becomes S_d^2/n_d under SRS with replacement.

The post-stratified synthetic estimator assumes that data are distributed into K (large) groups called post-strata that cut across the domains, and such that the within group mean is constant across domains, that is, if $\bar{Y}_{dk}$ denotes the mean in the crossing of post-stratum k and domain d and $\bar{Y}_{+k}$ is the mean of post-stratum k, it holds that $\bar{Y}_{dk} = \bar{Y}_{+k}$, $k = 1, \ldots, K$. The groups are assumed to have large enough sample sizes to allow direct estimation with high efficiency. Since the mean of domain d is given by $\bar{Y}_d = N_d^{-1} \sum_{k=1}^{K} N_{dk} \bar{Y}_{dk}$, replacing $\bar{Y}_{dk} = \bar{Y}_{+k}$ by the ratio HT estimator $\hat{\bar{Y}}_{+k}^{R} = \hat{Y}_{+k}^{DIR}/\hat{N}_{+k}^{DIR}$, where $\hat{Y}_{+k}^{DIR}$ is the direct estimator of the total in post-stratum k and $\hat{N}_{+k}$ is the direct estimator of the population size N_{+k} in the same post-stratum, we obtain the post-stratified synthetic estimator

$$\hat{\bar{Y}}_d^{SYN} = \frac{1}{N_d} \sum_{k=1}^{K} N_{dk} \hat{\bar{Y}}_{+k}^{R}.$$

Note that this estimator requires the population sizes of the crossings between each post-stratum k and domain d, N_{dk} for all k and d.

The direct estimator is inefficient for a domain with small sample size. On the other hand, the post-stratified synthetic estimator is biased when the assumption of constant means across domains within a stratum does not hold. To balance the bias of a synthetic estimator and the instability of the direct estimator, [1] proposed the sample-size dependent (SSD) estimator defined as a composition of the two mentioned estimators, that is,

$$\hat{\bar{Y}}_d^{SSD} = \phi_d \hat{\bar{Y}}_d^{DIR} + (1 - \phi_d) \hat{\bar{Y}}_d^{SYN},$$

where the composition weight ϕ_d depends on the sample size of the domain as

$$\phi_d = \begin{cases} 1, & \hat{N}_d^{DIR} \geq \delta N_d; \\ \hat{N}_d^{DIR}/(\delta N_d), & \hat{N}_d^{DIR} < \delta N_d, \end{cases}$$

for a given constant $\delta > 0$ that controls how much weight is attached to the synthetic estimator, with larger value of δ meaning that more strength is borrowed from other domains. However, if the expected sample size is small, then the SSD estimator is not borrowing strength in domains d with $\hat{N}_d^{DIR} \geq \delta N_d$ even if they have small sample sizes.

Functions `direct()`, `pssynt()` and `ssd()` give respectively direct, post-stratified synthetic and sample size dependent estimates. The calls to these functions are:

```
direct(y, dom, sweight, domsize, data, replace = FALSE)
pssynt(y, sweight, ps, domsizebyps, data)
ssd(dom, sweight, domsize, direct, synthetic, delta = 1, data)
```

Function `direct()` returns unbiased direct estimates of the area means, where the result depends on the sampling design specified through the sampling weight vector `sweight` and the argument `replace` for with or without replacement sampling. We must provide the area population sizes in the data frame `domsize`, whose first column must contain the area codes.

In `pssynt()`, we must specify our selected post-stratifying variable in argument `ps`. The population sizes of each crossing between domain and post-strata must be specified in the data frame `domsizebyps`, whose first column must be again the area codes.

Function `ssd()` gives SSD estimators obtained by composition of direct and synthetic estimators. We need to introduce the direct estimators (`direct`) and the synthetic estimators (`synthetic`) to compose, together with the constant δ (`delta`) involved in the SSD estimator. Domain codes (`dom`) and domain population sizes (`domsize`) are also required arguments.

The vector of sampling weights (`sweight`) must be included in the three functions. The variables specified in `y`, `dom`, `sweight` and `ps` can be selected from the data set specified in argument `data`.

Example. Poverty mapping

In this example, we calculate several simple estimates of poverty incidences in Spanish provinces, namely direct estimates, post-stratified synthetic estimates with education levels as post-strata and SSD estimates obtained from the composition of direct and post-stratified synthetic estimates.

The poverty incidence for a province is the province mean of a binary variable taking value 1 when person's income is below a given poverty line and 0 otherwise. Direct estimates can be obtained easily applying the usual theory for means to this binary variable. First, we load the data set `incomedata` containing the input data for each individual and the data sets `sizeprov` and `sizeprovedu` containing the population sizes and the population sizes by education level, respectively.

```
> library("sae")
> data("incomedata")
> data("sizeprov")
> data("sizeprovedu")
```

Next, we define the poverty line `z`, calculate the binary variable `poor`, with value `1` if the corresponding income value is below the poverty line and `0` otherwise, and calculate province poverty incidences as province means of this variable.

```
> z <- 6557.143
> poor <- as.integer(incomedata$income < z)
```

We use the province name `provlab` as the domain code (`dom`) and calculate direct estimates `DIR`.

```
> Popn <- sizeprov[, c("provlab", "Nd")]
> DIR <- direct(y = poor, dom = incomedata$provlab,
+               sweight = incomedata$weight, domsize = Popn)
```

Next, we calculate post-stratified synthetic estimates with education levels as post-strata. For the function `pssynt()`, we construct the data frame `domsizebyps`, containing the domain codes `provlab` in the first column and, in the remaining columns, the province sizes by education level. The names of the columns (except for the first one) in this data frame must be the education levels, namely `0` (age<16), `1` (primary education), `2` (secondary education) and `3` (post-secondary education):

```
> Popn.educ <- sizeprovedu[,-2]
> colnames(Popn.educ) <- c("provlab","0", "1", "2", "3")
> PSYN.educ <- pssynt(y = poor, sweight = incomedata$weight,
+                     ps = incomedata$educ,
+                     domsizebyps = Popn.educ)
```

We calculate SSD estimates by composition of the previous direct and post-stratified estimates, and taking the default value `delta=1` in function `ssd()`. Again, the first columns of `domsize`, `direct` and `synthetic` must be the province names.

```
> SSD <- ssd(dom = provlab, sweight = weight, domsize = Popn,
+            direct = DIR[, c("Domain", "Direct")],
+            synthetic = PSYN.educ, data = incomedata)
```

We collect the province names, sample sizes and the three sets of percent poverty incidence estimates in the data frame `results`:

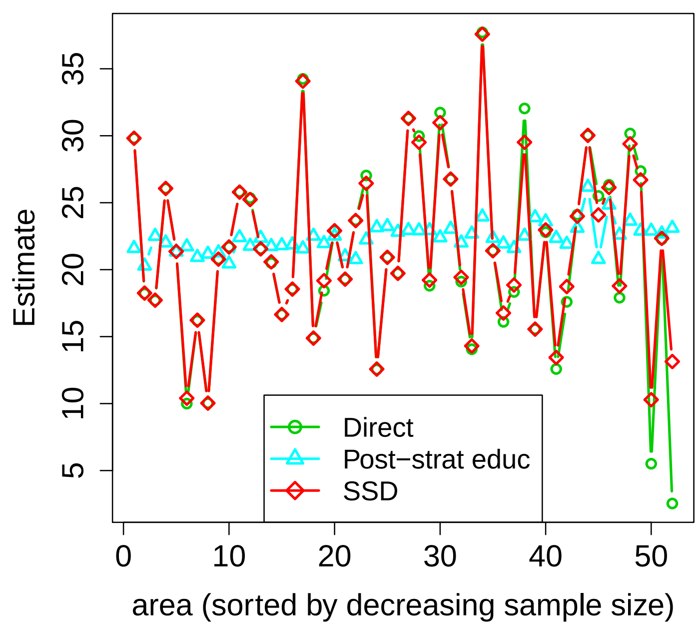
```
> results <- data.frame(Province = DIR$Domain,
+                       SampleSize = DIR$SampSize,
+                       DIR = DIR$Direct * 100,
+                       PSYN.educ = PSYN.educ$PsSynthetic * 100,
+                       SSD = SSD$ssd * 100)
> print(results, row.names = FALSE)
```

```
   Province SampleSize       DIR PSYN.educ      SSD
      Alava         96 25.503732  20.77880 24.08931
   Albacete        173 14.059242  22.67562 14.30411
   Alicante        539 20.785096  21.26954 20.78510
    Almeria        198 26.763976  23.02936 26.76398
      Avila         58  5.512200  22.89330 10.28835
    Badajoz        494 21.553890  22.35924 21.55389
   Baleares        634  9.999792  21.71882 10.40240
  Barcelona       1420 29.812535  21.59556 29.81253
     Burgos        168 21.413150  22.35331 21.41315
```

Caceres	282	27.031324	22.23249	26.44514
Cadiz	398	14.887351	22.51448	14.88735
Castellon	118	17.598199	21.91192	18.73778
Ceuta	235	19.724796	22.81006	19.72480
CiudadReal	250	20.921534	23.23302	20.92153
Cordoba	224	29.975708	22.91798	29.51045
CorunaLa	495	25.347550	21.76006	25.23624
Cuenca	92	26.334059	24.83639	26.13496
Gerona	142	18.337421	21.59600	18.85399
Granada	208	31.727340	22.39243	30.97619
Guadalajara	89	17.908182	22.59389	18.78456
Guipuzcoa	285	23.690549	20.76857	23.66709
Huelva	122	12.583449	22.35069	13.44200
Huesca	115	24.107606	23.10616	23.98812
Jaen	232	31.294198	22.93972	31.29420
Leon	218	18.801572	22.93115	19.22223
Lerida	130	15.559590	23.89632	15.55959
Lugo	173	37.718722	23.94922	37.58235
Madrid	944	18.218209	20.28249	18.25089
Malaga	379	22.918462	22.51928	22.90551
Melilla	180	19.109119	22.00697	19.43014
Murcia	885	17.703167	22.50054	17.72239
Navarra	564	16.190765	20.92992	16.22866
Orense	129	22.799612	23.58691	22.96765
Oviedo	803	26.064010	22.00916	26.06401
Palencia	72	30.166074	23.63212	29.39216
PalmasLas	472	16.651843	21.80900	16.65184
Pontevedra	448	18.549072	21.86237	18.54907
RiojaLa	510	25.811811	22.40296	25.78924
Salamanca	164	16.104513	21.93240	16.76284
Santander	434	34.244429	21.56598	34.07708
Segovia	58	22.262002	22.67927	22.33761
Sevilla	482	20.503036	21.74189	20.58245
Soria	20	2.541207	23.10395	13.14019
Tarragona	134	32.035438	22.51761	29.51279
Tenerife	381	18.429619	21.96155	19.17768
Teruel	72	27.364239	22.89205	26.70145
Toledo	275	12.553377	23.14442	12.57643
Valencia	714	21.360678	21.32963	21.36054
Valladolid	299	19.292332	20.98068	19.29233
Vizcaya	524	21.694466	20.44194	21.69447
Zamora	104	30.027442	26.17055	30.02744
Zaragoza	564	10.034577	21.17064	10.03458

These estimates are plotted in the Figure for each province (area), with provinces sorted by decreasing sample size. This Figure shows that direct estimates and SSD estimates are very similar, with direct estimates slightly more unstable. However, the post-stratified synthetic estimates appear to be too stable, giving practically the same values for all provinces. This estimator is based on the unrealistic assumption of constant poverty incidence for all the population with the same education level and therefore might be seriously biased.

```
> # Sorted results by decreasing sample size
> results <- results[order(results$SampleSize,
+                           decreasing = TRUE), ]
> plot(results$DIR, type = "n",
+      xlab = "area (sorted by decreasing sample size)",
+      ylab = "Estimate", cex.axis = 1.5, cex.lab = 1.5)
> points(results$DIR, type = "b", col = 3, lwd = 2, pch = 1)
> points(results$PSYN.educ, type= "b", col = 5, lwd = 2, pch = 2)
> points(results$SSD, type = "b", col = 2, lwd = 2, pch = 5)
> legend("bottom", legend = c("Direct", "Post-strat educ", "SSD"),
+        ncol = 1, col = c(3, 5, 2), lwd = rep(2, 3),
+        pch = c(1, 2, 5), cex = 1.3)
```



Comparing direct estimates with the EB estimates of poverty incidences obtained in the data frame `results.EB` of Example 5 in [2], we can see that estimates differ significantly for the 5 selected provinces and the CVs show great gains in efficiency of EB estimates as compared with direct estimates.

```
> DIR[c("42","5","34","44","40"), -4]

     Domain SampSize     Direct       CV
42    Soria       20 0.02541207 99.97815
5     Avila       58 0.05512200 46.35946
34 Palencia       72 0.30166074 23.80085
44   Teruel       72 0.27364239 24.57017
40  Segovia       58 0.22262002 25.33449
```

References

[1] DREW, D., SINGH, M.P. & CHOUDHRY, G.H. (1982). Evaluation of small area estimation techniques for the Canadian Labour Force Survey. *Survey Methodology* **8**, 17–47.

[2] MOLINA, I. & MARHUENDA, Y. (1982). sae: An R package for Small Area Estimation. *R Journal*, Under revision.