Describing PPS designs to R

Thomas Lumley

March 14, 2011

The survey package has always supported PPS (ie, arbitrary unequal probability) sampling with replacement, or using the with-replacement single-stage approximation to a multistage design. No special notation is required: just specify the correct sampling weights.

Version 3.11 added an another approximation for PPS sampling without replacement, and version 3.16 added more support. There are two broad classes of estimators for PPS sampling without replacement: approximations to the Horvitz–Thompson and Yates–Grundy estimators based on approximating the pairwise sampling probabilities, and estimators of Hájek type that attempt to recover the extra precision of a without-replacement design by conditioning on the estimated population size.

Hájek-type estimators

Using the standard recursive algorithm for stratified multistage sampling when one or more stages are actually PPS gives an approximation due to Brewer. This is simple to compute, always non-negative, and appears to be fairly efficient.

Approximating π_{ij}

Given the pairwise sampling probabilities π_{ij} we can define the weighted covariance of sampling indicators

$$\check{\Delta}_{ij} = 1 - \frac{\pi_i \pi_j}{\pi_{ij}}$$

and the weighted observations

$$\check{x}_i = \frac{1}{\pi_i} x_i.$$

Two unbiased estimators of the variance of the total of x are the Horvitz–Thompson estimator

$$\hat{V}_{HT} = \sum_{i,j=1}^{n} \check{\Delta} \check{x}_i \check{x}_j$$

and the Yates–Grundy(–Sen) estimator

$$\hat{V}_{YG} = \frac{1}{2} \sum_{i,j=1}^{n} \check{\Delta} (\check{x}_i - \check{x}_j)^2$$

The Yates–Grundy estimator appears to be preferred in most comparisons. It is always non-negative (up to rounding error, at least).

In principle, π_{ij} might not be available and various approximations have been proposed. The (truncated) Hartley–Rao approximation is

$$\check{\Delta}_{ij} = 1 - \frac{n - \pi_i - \pi_j + \sum_{k=1}^{N} \pi_k^2/n}{n-1}$$

which requires knowing π_i for all units in the population. The population sum can be estimated from the sample, giving a further approximation

$$\check{\Delta}_{ij} = 1 - \frac{n - \pi_i - \pi_j + \sum_{k=1}^{n} \pi_k/n}{n-1}.$$

that requires only the sample π_i. Overton's approximation is

$$\check{\Delta}_{ij} = 1 - \frac{n - (\pi_i + \pi_j)/2}{n-1}$$

which also requires only the sample π_i.

In practice, given modern computing power, π_{ij} should be available either explicitly or by simulation, so the Hartley–Rao and Overton approximations are not particularly useful.

0.1 Using the PPS estimators

At the moment, only Brewer's approximation can be used as a component of multistage sampling, though for any sampling design it is possible to work out the joint sampling probabilities and use the other approaches. The other approaches can be used for cluster sampling or for sampling of individual units. This is likely to change in the future.

To specify a PPS design, the sampling probabilities must be given in the `prob` argument of `svydesign`, or in the `fpc` argument, with `prob` and `weight` unspecified. In addition, it is necessary to specify which PPS computation should be used, with the `pps` argument. The optional `variance` argument specifies the Horvitz–Thompson (`variance="HT"`) or Yates–Grundy (`variance="YG"`) estimator, with the default being `"HT"`.

Some estimators require information in addition to the sampling probabilities for units in the sample. This information is supplied to the `pps=` argument of `svydesign` using wrapper functions that create objects with appropriate classes. To specify the population sum $\sum pi_i^2/n$ needed for the Hartley–Rao approximation, use `HR()`, and to specify a matrix of pairwise sampling probabilities use `ppsmat()`. The function `HR()` without an argument will use the Hartley–Rao approximation and estimate the population sum from the sample.

The data set `election` contains county-level voting data from the 2004 US presidential elections, with a PPS sample of size 40 taken using Tillé's splitting method, from the `sampling` package. The sampling probabilities vary widely, with Los Angeles County having a probability of 0.9 and many small counties having probabilities less than 0.0005.

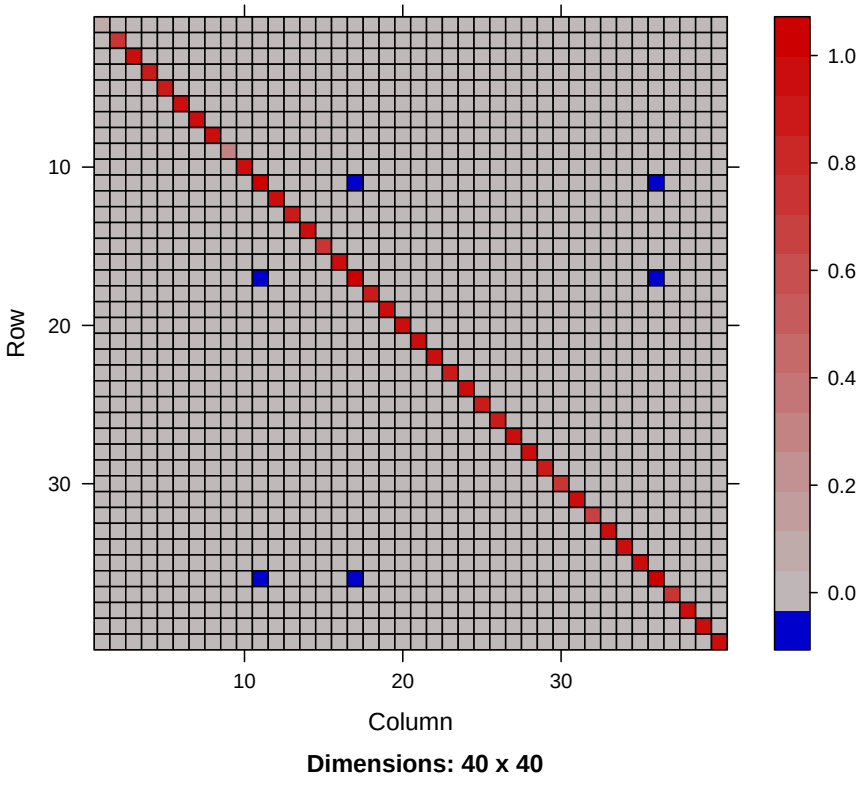
```
> library(survey)
> data(election)
> summary(election$p)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
1.377e-06 7.260e-04 2.250e-03 8.696e-03 5.729e-03 9.037e-01

> summary(election_pps$p)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0001429 0.0153800 0.0398100 0.1107000 0.1103000 0.9037000
```

Some possible survey design specifications for these data are:

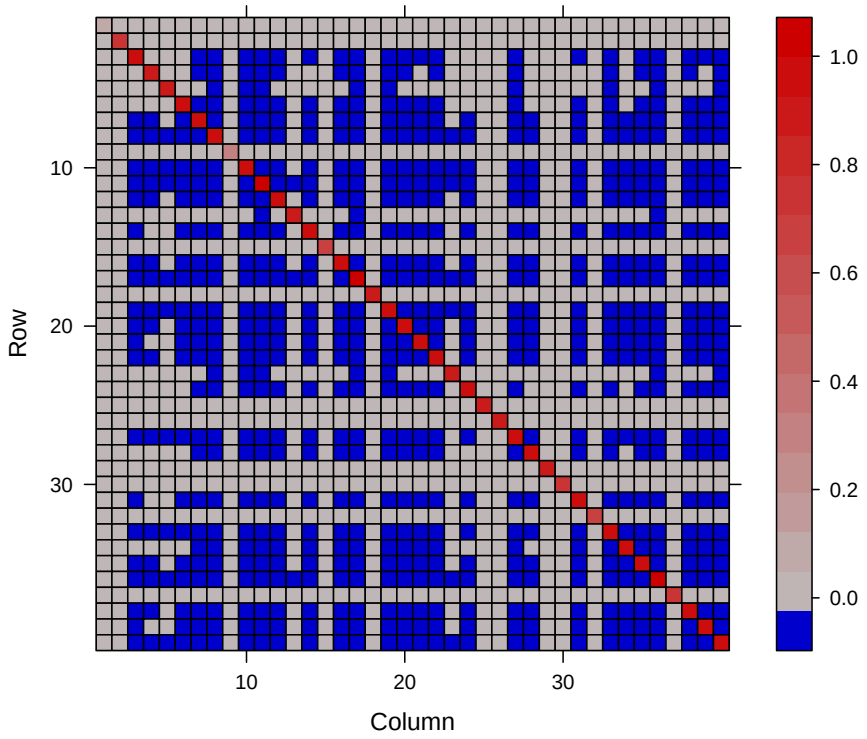
```
> dpps_br <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = "brewer")
> dpps_ov <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = "overton")
> dpps_hr <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = HR(sum(election$p^2)/40))
> dpps_hr1 <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = HR())
> dpps_ht <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = ppsmat(election_jointprob))
> dpps_yg <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = ppsmat(election_jointprob), variance = "YG")
> dpps_hryg <- svydesign(id = ~1, fpc = ~p, data = election_pps,
+     pps = HR(sum(election$p^2)/40), variance = "YG")
> dppswr <- svydesign(id = ~1, probs = ~p, data = election_pps)
```

All the without-replacement design objects except for Brewer's method include a matrix $\check{\Delta}$. These can be visualized with the `image()` method. These plots use the `lattice` package and so need `show()` to display them inside a program:

```
> show(image(dpps_ht))
```



```
> show(image(dpps_ov))
```



Dimensions: 40 x 40

In this example there are more negative entries in $\check{\Delta}$ with the approximate methods than when the full pairwise sampling matrix is supplied.

The estimated totals are the same with all the methods, but the standard errors are not.

```
> svytotal(~Bush + Kerry + Nader, dpps_ht)

         total      SE
Bush  64518472 2604404
Kerry 51202102 2523712
Nader   478530  102326
```

```
> svytotal(~Bush + Kerry + Nader, dpps_yg)

         total      SE
Bush  64518472 2406526
Kerry 51202102 2408091
Nader   478530  101664
```

```
> svytotal(~Bush + Kerry + Nader, dpps_hr)

         total      SE
Bush  64518472 2624662
Kerry 51202102 2525222
Nader   478530  102793
```

```
> svytotal(~Bush + Kerry + Nader, dpps_hryg)

         total      SE
Bush  64518472 2436738
Kerry 51202102 2439845
Nader   478530  102016

> svytotal(~Bush + Kerry + Nader, dpps_hr1)

         total      SE
Bush  64518472 2472753
Kerry 51202102 2426842
Nader   478530  102595

> svytotal(~Bush + Kerry + Nader, dpps_br)

         total      SE
Bush  64518472 2447629
Kerry 51202102 2450787
Nader   478530  102420

> svytotal(~Bush + Kerry + Nader, dpps_ov)

         total      SE
Bush  64518472 2939608
Kerry 51202102 1964632
Nader   478530  104373

> svytotal(~Bush + Kerry + Nader, dppswr)

         total      SE
Bush  64518472 2671455
Kerry 51202102 2679433
Nader   478530  105303
```