Calibration

January 21, 2011

1 Example 1

This is an example of using the `calib` function for calibration and nonresponse adjustment (with response homogeneity groups).

We create the population data frame (the population size is $N = 250$):

- there are four variables: `state`, `region`, `income` and `sex`;
- the `state` variable has 2 categories: 'A' and 'B'; the `region` variable has 3 categories: 1, 2, 3 (regions within states);
- the `income` and `sex` variables are randomly generated using the uniform distribution.

```
> data = rbind(matrix(rep("A", 150), 150, 1, byrow = TRUE),
+     matrix(rep("B", 100), 100, 1, byrow = TRUE))
> data = cbind.data.frame(data, c(rep(1, 60), rep(2,
+     50), rep(3, 60), rep(1, 40), rep(2, 40)),
+     1000 * runif(250))
> sex = runif(nrow(data))
> for (i in 1:length(sex)) if (sex[i] < 0.3) sex[i] = 1 else sex[i] = 2
> data = cbind.data.frame(data, sex)
> names(data) = c("state", "region", "income", "sex")
> summary(data)
```

```
 state       region          income              sex       
 A:150   Min.   :1.00   Min.   :  2.933   Min.   :1.000  
 B:100   1st Qu.:1.00   1st Qu.:288.040   1st Qu.:1.000  
         Median :2.00   Median :506.748   Median :2.000  
         Mean   :1.84   Mean   :504.124   Mean   :1.684  
         3rd Qu.:2.00   3rd Qu.:749.557   3rd Qu.:2.000  
         Max.   :3.00   Max.   :986.274   Max.   :2.000  
```

We compute the population stratum sizes:

```
> table(data$state)
```

```
  A   B
150 100
```

We select a stratified sample. The `state` variable is used as a stratification variable. The sample stratum sizes are 25 and 10, respectively. The method is 'srswor' (equal probability, without replacement).

```
> s = strata(data, c("state"), size = c(25, 10),
+     method = "srswor")
```

We obtain the observed data:

```
> s = getdata(data, s)
```

The `status` variable is used in the `rhg_strata` function. The `status` column is added to s (1 - sample respondent, 0 otherwise); it is randomly generated using the uniform distribution:

```
> status = runif(nrow(s))
> for (i in 1:length(status)) if (status[i] < 0.3) status[i] = 0 else status[i] = 1
> s = cbind.data.frame(s, status)
```

We compute the response homeogeneity groups using the `region` variable:

```
> s = rhg_strata(s, selection = "region")
```

We select only the sample respondents:

```
> sr = s[s$status == 1, ]
```

We creates the population data frame of sex and region indicators:

```
> X = matrix(0, nrow = nrow(data), ncol = 5)
> for (i in 1:nrow(data)) {
+     if (data$sex[i] == 1)
+         X[i, 1] = 1
+     if (data$sex[i] == 2)
+         X[i, 2] = 1
+     if (data$region[i] == 1)
+         X[i, 3] = 1
+     if (data$region[i] == 2)
+         X[i, 4] = 1
+     if (data$region[i] == 3)
+         X[i, 5] = 1
+ }
```

We compute the population totals for each sex and region:

```
> total = c(t(rep(1, nrow(data))) %*% X)
```

The first method consists in calibrating with all strata. The respondent data frame of `sex` and `region` indicators is created. The initial weights using the inclusion prob. and the response probabilities are computed.

```
> Xs = X[sr$ID_unit, ]
> d = 1/(sr$Prob * sr$prob_resp)
> summary(d)
```

We compute the g-weights using the linear method:

```
> g = calib(Xs, d, total, method = "linear")
> summary(g)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5056  0.5056  0.5669  1.0720  0.5669  3.7130
```

The final weights are:

```
> w = d * g
> summary(w)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.413   4.535   4.535  10.000  11.340  30.000
```

We check the calibration:

```
> checkcalibration(Xs, d, total, g)

$message
[1] "the calibration is done"

$result
[1] TRUE

$value
[1] 1e-06
```

The second method consists in calibrating in each stratum. The respondent data frame of `sex` and `region` indicators is created in each stratum. The initial weights using the inclusion prob. and response probabilities are computed in each stratum.

```
> cat("stratum 1\n")

stratum 1
```

```
> data1 = data[data$state == "A", ]
> X1 = X[data$state == "A", ]
> total1 = c(t(rep(1, nrow(data1))) %*% X1)
> sr1 = sr[sr$Stratum == 1, ]
> Xs1 = X[sr1$ID_unit, ]
> d1 = 1/(sr1$Prob * sr1$prob_resp)
> g1 = calib(Xs1, d1, total1, method = "linear")
> checkcalibration(Xs1, d1, total1, g1)

$message
[1] "the calibration is done"

$result
[1] TRUE

$value
[1] 1e-06

> cat("stratum 2\n")

stratum 2

> data2 = data[data$state == "B", ]
> X2 = X[data$state == "B", ]
> total2 = c(t(rep(1, nrow(data2))) %*% X2)
> sr2 = sr[sr$Stratum == 2, ]
> Xs2 = X[sr2$ID_unit, ]
> d2 = 1/(sr2$Prob * sr2$prob_resp)
> g2 = calib(Xs2, d2, total2, method = "linear")
> checkcalibration(Xs2, d2, total2, g2)

the calibration cannot be done. The max EPS value is given by 'value'.
$message
NULL

$result
[1] FALSE

$value
[1] 1
```

2 Example 2

This is an example of:

- variance estimation of the calibration estimator (using the `calibev` and `varest` functions),

- variance estimator of the Horvitz-Thompson estimator (using the `varest` function).

We generate an artificial population and use Tillé sampling. The population size is 100, and the sample size is 20. There are three auxiliary variables (two categorical and one continuous; the matrix X). The vector $Z = (150, 151, \ldots, 249)'$ is used to compute the first-order inclusion probabilities. The variable of interest Y is computed using the model $Y_j = 5 * Z_j * (\varepsilon_j + \sum_{i=1}^{100} X_{ij}), \varepsilon_j \sim N(0, 1/3), j = 1, \ldots, 100$. The calibration estimator uses the linear method. Simulations are conducted to compute the MSE of the two variance estimators of the calibration estimator. Since the linear method is used in calibration, the calibration estimator is the generalized regression estimator. Thus an approximate variance can be computed on the population level and used in the bias estimation of the variance estimators. For the Horvitz-Thompson estimator, the variance can be computed on the population level and compared with the simulations' result. Run 10000 simulations to obtain accurate results (for time consuming reason, in the following program, the number of simulations is only 10).

```
> X = cbind(c(rep(1, 50), rep(0, 50)), c(rep(0,
+     50), rep(1, 50)), 1:100)
> total = apply(X, 2, "sum")
> Z = 150:249
> Y = 5 * Z * (rnorm(100, 0, sqrt(1/3)) + apply(X,
+     1, "sum"))
> pik = inclusionprobabilities(Z, 20)
> pikl = UPtillepi2(pik)
> nsim = 10
> c1 = c2 = c3 = c4 = c5 = numeric(nsim)
> for (i in 1:nsim) {
+     s = UPtille(pik)
+     piks = pik[s == 1]
+     Xs = X[s == 1, ]
+     g = calib(Xs, d = 1/piks, total, method = "linear")
+     Ys = Y[s == 1]
+     pikls = pikl[s == 1, s == 1]
+     cc = calibev(Ys, Xs, total, pikls, d = 1/piks,
+         g, with = FALSE, EPS = 1e-06)
+     c1[i] = cc$calest
+     c2[i] = cc$evar
+     c3[i] = varest(Ys, Xs, pik = piks, w = g/piks)
+     c4[i] = varest(Ys = Ys, pik = piks)
+     c5[i] = HTestimator(Ys, piks)
+ }
> cat("the population total:", sum(Y), "\n")

the population total: 5552560

> cat("the calibration estimator under simulations:",
+     mean(c1), "\n")

the calibration estimator under simulations: 5541103
```

```
> N = length(Y)
> delta = matrix(0, N, N)
> for (k in 1:(N - 1)) for (l in (k + 1):N) delta[k,
+     l] = delta[l, k] = pikl[k, l] - pik[k] * pik[l]
> diag(delta) = pik * (1 - pik)
> varHT = 0
> varasym = 0
> e = lm(Y ~ X)$resid
> for (k in 1:N) for (l in 1:N) {
+     varHT = varHT + Y[k] * Y[l] * delta[k, l]/(pik[k] *
+         pik[l])
+     varasym = varasym + e[k] * e[l] * delta[k,
+         l]/(pik[k] * pik[l])
+ }
> cat("the approximate variance of the calibration estimator:",
+     varasym, "\n")

the approximate variance of the calibration estimator: 5863267486

> cat("first variance estimator of the calibration est. using calibev function:\n")

first variance estimator of the calibration est. using calibev function:

> cat("MSE of the first variance estimator:", var(c2) +
+     (mean(c2) - varasym)^2, "\n")

MSE of the first variance estimator: 3.535871e+18

> cat("second variance estimator of the calibration est. using varest function:\n")

second variance estimator of the calibration est. using varest function:

> cat("MSE of the second variance estimator:", var(c3) +
+     (mean(c3) - varasym)^2, "\n")

MSE of the second variance estimator: 2.96538e+18

> cat("the Horvitz-Thompson estimator under simulations:",
+     mean(c5), "\n")

the Horvitz-Thompson estimator under simulations: 5674414

> cat("the variance of the Horvitz-Thompson estimator:",
+     varHT, "\n")
```

```
the variance of the Horvitz-Thompson estimator: 317349937125

> cat("the variance estimator of the H-T estimator under simulations:",
+     mean(c4), "\n")

the variance estimator of the H-T estimator under simulations: 330676060497

> cat("MSE of the variance estimator:", var(c4) +
+     (mean(c4) - varHT)^2, "\n")

MSE of the variance estimator: 3.860782e+21
```