Calibration

December 18, 2009

1 Example 1

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

Creates the population data frame (4 variables, 'state', 'region', 'income' and 'sex'; 'state' has 2 categories 'nc' and 'sc'; 'region' has 3 categories 1,2,3; 'income' and 'sex' are randomnly generated):

```
> data = rbind(matrix(rep("nc", 165), 165, 1, byrow = TRUE),
+     matrix(rep("sc", 70), 70, 1, byrow = TRUE))
> data = cbind.data.frame(data, c(rep(1, 100), rep(2,
+     50), rep(3, 15), rep(1, 30), rep(2, 40)),
+     1000 * runif(235))
> 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")
```

Computes the population stratum sizes:

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

Not run:

nc sc

165 70

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")
```

Obtains the observed data:

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

The 'status' variable is used in the 'rhg_strata' function. Adds the 'status' column to s (1 - sample respondent, 0 otherwise); it is randomnly generated:

```
> 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)
```

Computes the response homeogeneity groups using the 'region' variable:

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

Selects only the sample respondents:

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

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
+ }
```

Computes the population totals for each sex and region:

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

Creates the sample data frame of sex and region indicators:

```
> Xs = matrix(0, nrow = nrow(sr), ncol = 5)
> for (i in 1:nrow(sr)) {
+     if (sr$sex[i] == 1)
+         Xs[i, 1] = 1
```

```
+     if (sr$sex[i] == 2)
+         Xs[i, 2] = 1
+     if (sr$region[i] == 1)
+         Xs[i, 3] = 1
+     if (sr$region[i] == 2)
+         Xs[i, 4] = 1
+     if (sr$region[i] == 3)
+         Xs[i, 5] = 1
+ }
```

Computes the initial weights using the inclusion and response probabilities:

```
> d = 1/(sr$Prob * sr$prob_resp)
```

Computes the g-weights:

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

Checks the calibration:

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

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 the 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 $Y_j = 5 * Z_j * (\varepsilon_j + \sum_{i=1}^{100} X[i,j]), \varepsilon_j \sim N(0, 1/3), j = 1, \ldots, 100$. The calibration estimator uses the linear method. Simulations are used to compare the two variance estimators of the calibration estimator under the criterion of mean square error. For the Horvitz-Thompson estimator, the variance can be computed 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 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, 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")
> cat("the mean under simulations of the calibration estimator:",
+     mean(c1), "\n")
> cat("for the calibration estimator:\n")
> cat("the relative bias of the variance estimator using calibev function:",
+     (mean(c2) - sum(Y))/sd(c2), "\n")
> cat("MSE of the previous estimator:", var(c2) +
+     (mean(c2) - sum(Y))^2, "\n")
> cat("the relative bias of the variance estimator using varest function:",
+     (mean(c3) - sum(Y))/sd(c3), "\n")
> cat("MSE of the previous estimator:", var(c3) +
+     (mean(c3) - sum(Y))^2, "\n")
> cat("the mean under simulations of the Horvitz-Thompson estimator:",
+     mean(c5), "\n")
> cat("the mean under simulations of the variance estimator of the H-T estimator:",
+     mean(c4), "\n")
> N = length(Y)
> delta = matrix(0, N, N)
> for (k in 1:N) for (l in 1:N) if (k != l) delta[k,
+     l] = pikl[k, l] - pik[k] * pik[l]
> diag(delta) = pik * (1 - pik)
> varHT = 0
> for (k in 1:N) for (l in 1:N) varHT = varHT +
+     Y[k] * Y[l] * delta[k, l]/(pik[k] * pik[l])
> cat("the variance of the Horvitz-Thompson estimator:",
+     varHT, "\n")
```