Calibration and generalized calibration

January 12, 2021

1 Example 1

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

We create the following 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
 Length:250         Min.   :1.00   Min.   :  3.015
 Class :character   1st Qu.:1.00   1st Qu.:227.056
 Mode  :character   Median :2.00   Median :468.045
                    Mean   :1.84   Mean   :489.860
                    3rd Qu.:2.00   3rd Qu.:759.193
                    Max.   :3.00   Max.   :989.742
      sex
 Min.   :1.000
 1st Qu.:1.000
 Median :2.000
 Mean   :1.656
```

```
 3rd Qu.:2.000
 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 20, respectively. The method is 'srswor' (equal probability, without replacement).

```
> s=strata(data,c("state"),size=c(25,20), 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 U(0,1). The response probability for all units is 0.3.

```
> 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 create the population data frame of sex and region indicators:

```
> X=cbind(disjunctive(data$sex),disjunctive(data$region))
```

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)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.000   5.000   7.200   8.065  12.000  13.500
```

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.
 0.8331  0.9029  0.9029  0.9880  1.1018  1.1459
```

The final weights are:

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

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.166   4.515   7.640   8.065  11.247  13.751
```

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)
```

```
$message
[1] "the calibration is done"

$result
[1] TRUE

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

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` and `varHT` functions).

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 $Y_j = 5 * Z_j * (\varepsilon_j + \sum_{i=1}^{100} X_{ij}), \varepsilon_j \sim N(0, 1/3), iid, j = 1, \ldots, 100$. The calibration estimator uses the linear method. Simulations are conducted to estimate the MSE of the two variance estimators of the calibration estimator. Since the linear method is used in calibration, the calibration estimator corresponds to the generalized regression estimator. For the latter 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. Use 10000 simulation runs to obtain accurate results (for time consuming reason, in the following program, the number of runs is only 10).

```
> X=cbind(c(rep(1,50),rep(0,50)),c(rep(0,50),rep(1,50)),1:100)
> # vector of population totals
> total=apply(X,2,"sum")
> Z=150:249
> # the variable of interest
> Y=5*Z*(rnorm(100,0,sqrt(1/3))+apply(X,1,"sum"))
> # inclusion probabilities
> pik=inclusionprobabilities(Z,20)
> # joint inclusion probabilities
> pikl=UPtillepi2(pik)
> # number of runs; let nsim=10000 for an accurate result
> nsim=10
> c1=c2=c3=c4=c5=c6=numeric(nsim)
> for(i in 1:nsim)
+ {
+ # draws a sample
+ s=UPtille(pik)
+ # computes the inclusion prob. for the sample
+ piks=pik[s==1]
+ # the sample matrix of auxiliary information
+ Xs=X[s==1,]
+ # computes the g-weights
+ g=calib(Xs,d=1/piks,total,method="linear")
+ # computes the variable of interest in the sample
+ Ys=Y[s==1]
+ # computes the joint inclusion prob. for the sample
+ pikls=pikl[s==1,s==1]
+ # computes the calibration estimator and its variance estimation
+ cc=calibev(Ys,Xs,total,pikls,d=1/piks,g,with=FALSE,EPS=1e-6)
+ c1[i]=cc$calest
```

```
+ c2[i]=cc$evar
+ # computes the variance estimator of the calibration estimator (second method)
+ c3[i]=varest(Ys,Xs,pik=piks,w=g/piks)
+ # computes the variance estimator of the HT estimator using varest()
+ c4[i]=varest(Ys,pik=piks)
+ # computes the variance estimator of the HT estimator using varHT()
+ c5[i]=varHT(Ys,pikls,2)
+ # computes the Horvitz-Thompson estimator
+ c6[i]=HTestimator(Ys,piks)
+ }
> cat("the population total:",sum(Y),"\n")

the population total: 5561364

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

the calibration estimator under simulations: 5510423

> 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)
> var_HT=0
> var_asym=0
> e=lm(Y~X)$resid
> for(k in 1:N)
+   for(l in 1:N) {var_HT=var_HT+Y[k]*Y[l]*delta[k,l]/(pik[k]*pik[l])
+                  var_asym=var_asym+e[k]*e[l]*delta[k,l]/(pik[k]*pik[l])}
> cat("the approximate variance of the calibration estimator:",var_asym,"\n")

the approximate variance of the calibration estimator: 5564861682

> 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)-var_asym)^2,"\n")

MSE of the first variance estimator: 4.847537e+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)-var_asym)^2,"\n")

MSE of the second variance estimator: 3.225956e+18

> cat("the Horvitz-Thompson estimator under simulations:", mean(c6),"\n")

the Horvitz-Thompson estimator under simulations: 5733728

> cat("the variance of the HT estimator:", var_HT, "\n")

the variance of the HT estimator: 317320803552

> cat("the variance estimator of the HT estimator under simulations:", mean(c4),"\n")

the variance estimator of the HT estimator under simulations: 287142787029

> cat("MSE of the variance estimator 1 of HT estimator:", var(c4)+(mean(c4)-var_HT)^2,"\n")

MSE of the variance estimator 1 of HT estimator: 3.755112e+21

> cat("MSE of the variance estimator 2 of HT estimator:", var(c5)+(mean(c5)-var_HT)^2,"\n")

MSE of the variance estimator 2 of HT estimator: 4.206846e+21
```

3 Example 3

This is an example of generalized calibration used to handle unit nonresponse with different forms of response probabilities.

Consider the population U, the sample s and the set of respondents r with $r \subseteq s \subseteq U$.

The response mechanism is given by the distribution $q(r|s)$ such that for every fixed s we have

$$q(r|s) \geq 0, \text{ for all } r \in \mathcal{R}_s \text{ and } \sum_{s \in \mathcal{R}_s} q(r|s) = 1,$$

where $\mathcal{R}_s = \{r|r \subseteq s\}$. The variable of interest y_k is known only for $k \in r$. Under unit nonresponse we define the response indicator $R_k = 1$ if unit $k \in r$ and 0 otherwise and the response probabilities $p_k = Pr(R_k = 1|k \in s)$. It is assumed that R_k are independent Bernoulli variables with expected value equal to p_k. We assume that the units respond independently of each other and of s and so

$$q(r|s) = \prod_{k \in r} p_k \prod_{k \in \bar{r}} (1 - p_k).$$

The nonresponse model can be rewritten as

$$q(r|s,\boldsymbol{\gamma}) = \prod_{k\in r} F_k^{-1}(\boldsymbol{\gamma}) \prod_{k\in \bar{r}} (1 - F_k^{-1}(\boldsymbol{\gamma})).$$

In calibration method it is assumed that

$$\sum_{k\in r} \mathbf{x}_k d_k F_k(\boldsymbol{\gamma}) = \sum_{k\in r} \mathbf{x}_k d_k F(\boldsymbol{\gamma}^T \mathbf{x}_k) = \sum_{k\in U} \mathbf{x}_k,$$

where $F_k(\boldsymbol{\gamma}) = F(\boldsymbol{\gamma}^T \mathbf{x}_k), p_k = F_k(\boldsymbol{\gamma})^{-1}$, and d_k are the initial weigths.

In generalized calibration a different equation is used

$$\sum_{k\in r} \mathbf{x}_k d_k F(\boldsymbol{\gamma}^T \mathbf{z}_k) = \sum_{k\in U} \mathbf{x}_k,$$

where $\mathbf{z}_k$ is not necessary equal to $\mathbf{x}_k$, but $\mathbf{z}_k$ and $\mathbf{x}_k$ have to be highly correlated. $\mathbf{z}_k$ should be known only for $k \in r$. The components of $\mathbf{z}_k$ that are not also components of $\mathbf{x}_k$ are often known as *instrumental variables*. Let w_k be the final weights (obtained after applying generalized calibration).

It is possible to assume different forms of response probabilities:

- Linear weight adjustment (it can be implemented by using the argument `method="linear"` in gencalib() function or `method="truncated"` if bounds are allowed): $p_k = 1/(1 + \boldsymbol{\gamma}^T \mathbf{z}_k)$ and $w_k = d_k(1 + \mathbf{h}^T \mathbf{z}_k)$, where $\mathbf{h}$ is a consistent estimate of $\boldsymbol{\gamma}$.
- Raking weight adjustment (it can be implemented by using the argument `method="raking"` in gencalib()): $p_k = 1/\exp(\boldsymbol{\gamma}^T \mathbf{z}_k)$ and $w_k = d_k \exp(\mathbf{h}^T \mathbf{z}_k)$.
- Logistic weight adjustment (it can be implemented by using the argument `method="raking"` in gencalib()): $p_k = 1/(1 + \exp(\boldsymbol{\gamma}^T \mathbf{z}_k)), w_k = d_k(1 + \exp(\mathbf{h}^T \mathbf{z}_k))$, but we calibrate on $\sum_{k\in U} \mathbf{x}_k - \sum_{k\in r} \mathbf{x}_k d_k$ instead of $\sum_{k\in U} \mathbf{x}_k$.
- Generalized exponential weight adjustment (Folsom and Singh, 2000; it can be implemented by using the argument `method="logit"` in gencalib()):

  $$p_k = 1/F(\boldsymbol{\gamma}^T \mathbf{z}_k), w_k = d_k F(\mathbf{h}^T \mathbf{z}_k),$$

  $$F(\mathbf{h}^T \mathbf{z}_k) = \frac{L(U-C) + U(C-L)\exp(A\mathbf{h}^T \mathbf{z}_k)}{(U-C) + (C-L)\exp(A\mathbf{h}^T \mathbf{z}_k)} \in (L, U),$$

  where $A = (U-L)/((C-L)(U-C))$ and $L \geq 0, 1 < U \leq \infty, U > C > L$, ($C = 1$ in the paper of Deville and Sarndal, 1992). The g-weights are centered around of C. For $L = 1, C = 2$ and $U = \infty, F(\mathbf{h}^T \mathbf{z}_k)$ approaches $1 + \exp(\mathbf{h}^T \mathbf{z}_k)$ and for $C = 1, L = 0, U = \infty$, $\exp(\mathbf{h}^T \mathbf{z}_k)$.

We exemplify the last form of response probabilities (generalized exponential weight adjustment) using artificial data. We generate a population of size $N = 400$ and consider the auxiliary information X following a Gamma distribution with parameters 3 and 4. The instrumental variable Z is generated using the model $Z = 2X + \varepsilon$, where $\varepsilon \sim U(0, 1)$. The variable of interest is Y generated using the model $Y = 3X + \varepsilon_1$, where $\varepsilon_1 \sim N(0, 1)$. We consider here that the nonresponse is not missing at random and the response probabilities

p depend on the variable of interest y which may be missing. We draw a simple random sampling without replecement of size $n = 100$ and generate the set of respondents r using Poisson sampling with the probabilties p. The bounds are fixed to 1 and 5, and the constant $C = 1.5$. Three estimators are computed:

- the generalized calibration estimator using Z as instrumental variable,
- the generalized calibration estimator using Y as instrumental variable,
- the generalized calibration estimator using X as instrumental variable, which is the same with the calibration estimator, but the g-weights are centered around C.

The convergence of the method is not guaranteed due to the bounds. Thus $g1, g2, g3$ can be null. If it the case, repeat the code (considering another s and r).

```
> N=400
> n=100
> X=rgamma(N,3,4)
> total=sum(X)
> Z=2*X+runif(N)
> Y=3*X+rnorm(N)
> print(cor(X,Y))

[1] 0.8078879

> print(cor(X,Z))

[1] 0.9525512

> L=1
> U=5
> C=1.5
> A=(U-L)/((C-L)*(U-C))
> p=((U-C)+(C-L)*exp(A*Y*0.3))/(L*(U-C)+U*(C-L)*exp(A*Y*0.3))
> summary(p)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2040  0.2980  0.3945  0.4167  0.5097  0.8416

> bounds=c(L,U)
> s=srswor(n,N)
> r=numeric(n)
> for(j in 1:n) if(runif(1)<p[s==1][j]) r[j]=1
> print("Size of r is:")

[1] "Size of r is:"
```

```
> nr=sum(r)
> print(nr)

[1] 40

> Xr=X[s==1][r==1]
> Yr=Y[s==1][r==1]
> Zr=Z[s==1][r==1]
> pikr=rep(n/N,times=nr)
> d=1/(pikr)
> g1=gencalib(Xr,Zr,d,total,method="logit",bounds=bounds,C=C)
> g2=gencalib(Xr,Yr,d,total,method="logit",bounds=bounds,C=C)
> g3=gencalib(Xr,Xr,d,total,method="logit",bounds=bounds,C=C)
> if(is.null(g1))
+ print("g1 is null") else
+ if(checkcalibration(Xr,d,total,g1)$result)
+ {print("the gen.calibration estimator using Zs as instrumental variable")
+ print(sum(Yr*g1*d))
+ }

[1] "the gen.calibration estimator using Zs as instrumental variable"
[1] 859.5932

> if(is.null(g2))
+ print("g2 is null") else
+ if(checkcalibration(Xr,d,total,g2)$result)
+ {
+ print("the gen.calibration estimator using Ys as instrumental variable")
+ print(sum(Yr*g2*d))
+ }

[1] "the gen.calibration estimator using Ys as instrumental variable"
[1] 917.0216

> if(is.null(g3))
+ print("g3 is null") else
+ if(checkcalibration(Xr,d,total,g3)$result)
+ {
+ print("the calibration estimator")
+ print(sum(Yr*g3*d))
+ }

[1] "the calibration estimator"
[1] 865.4158

> cat("The population total is:", sum(Y),"\n")

The population total is: 921.8406
```