Survey Samplings with R

Camelia Goga
LMB - Univ. of Bourgogne Franche-Comté, Besançon
en collaboration with
Anne Ruiz-Gazen (TSE - Univ. Toulouse 1 Capitole)

uRos Bucarest
6-7 November 2017
Survey Samplings with R

- Large expansion of R packages dedicated to survey sampling over the last 10 years: from few packages to more than eighty;

- Comprehensive list of all packages dedicated to survey sampling techniques and official statistics at https://cran.r-project.org/web/views/OfficialStatistics.html maintained by Matthias Templ.
Outline of my talk

1. sample selection
2. estimation of totals and means: the Horvitz-Thompson estimator and variance estimation
3. improving the Horvitz-Thompson estimator: the calibration estimator
4. more advanced issues

Book (in work) in collaboration with H. Juillard and A. Ruiz-Gazen (Univ. Toulouse 1 Capitole)
Sample selection and estimation of totals

Multi-stages sampling designs

Improving the efficiency of the Horvitz-Thompson estimator

More advanced issues
Sample selection and estimation of totals

- Sample selection with package `sampling` of Alina Matei and Yves Tillé (Univ. of Neuchâtel) is concerned with the selection of samples according to many designs:
  - simple random sampling without replacement
  - unequal probability sampling designs
  - stratified sampling design
  - multistage samplings

- Estimation and variance estimation with package `survey` of Thomas Lumley (Univ. of Auckland), for which methods are called from package `srvyr` using the `dplyr` syntax;
  *Complex surveys, a guide to analysis using R* (2010).
Estimation of totals or means

- Consider a finite population $U = \{1, \ldots, N\}$ and a sample $s$ is selected from $U$ according to a sampling design.
- Let $Y$ be the interest variable and we want to estimate the finite population total of $Y$ on $U$:
  \[
  t_Y = \sum_{U} y_k
  \]
or the population mean $\overline{Y}_U = t_Y / N$.
- Weighted estimators are built to estimate $t_Y$:
  \[
  \hat{t}_w = \sum_{s} w_i y_i.
  \]

The Horvitz-Thompson estimator $\hat{t}_\pi$ is obtained for $w_i = d_i = 1/\pi_i$ with $\pi_i = P(k \in s) > 0$ for all $i \in U$.

- Weights are crucial in building survey estimators.
Variance estimation

- Horvitz-Thompson variance estimator (supposing that all $\pi_{kl} > 0$):

$$\hat{V}_{HT}(\hat{t}_d) = \sum_{k \in U} \sum_{l \in U} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_{kl}} \frac{y_k}{\pi_k} \frac{y_l}{\pi_l}$$

or Yates-Grundy-Sen variance estimator:

$$\hat{V}_{YGS}(\hat{t}_d) = -\frac{1}{2} \sum_{k \in U} \sum_{l \in U} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_{kl}} \left( \frac{y_k}{\pi_k} - \frac{y_l}{\pi_l} \right)^2$$

- Variance estimators based on replicates weights in package `survey`;

Camelia Goga  Survey Samplings with R
More complex parameters of interest

- The parameter $\theta$ is the solution (explicite or implicite) of
  
  $S(\theta) = \sum_{U} S_k(\theta) = 0$

  examples: coefficients of linear and logistic regression, ratio and
calibration estimators, ...; for implicit parameters, the solution is
approached by numerical algorithms;

- The estimator of $\theta$ is $\hat{\theta}_d$, the solution of
  
  $\hat{S}(\hat{\theta}_d) = \sum_{s} \frac{S_k(\hat{\theta}_d)}{\pi_k} = 0$

- By Taylor linearization, the variance estimator is given by (Binder,
1983):

  $\hat{V}(\hat{\theta}_d) = A^{-1} \cdot \hat{V}_{HT}(\hat{S}(\theta)) \cdot (A^t)^{-1}$

  where $A = \hat{S}(\hat{\theta}_d)$. 
Data description

- We consider data `rec99` from the last French Census performed in 1999;
- Data is collected on $N = 554$ French towns from the south of France (the region Haute-Garonne) about:
  - `POPSDC99`: the number of habitants in 1999
  - `LOG`: the number of dwellings
  - `LOGVAC`: the number of empty dwellings
  - `BVQ` ("bassin de vie quotidien") : a cluster variable
  - `stratlog`: a stratification variable based on the town population size ("1" small, ..., "4" very large);
- The goal is the estimation of the total of `LOGVAC`.

```r
> rec99=read.csv('rec99.csv')
> rec99[1:3,] ###the first 3 individuals from rec99
   CODE_N COMMUNE BVQ_N  POPSDC99  LOG LOGVAC stratlog
1   31014  ARGUENOS  31020      57   94     1       1
2   31131  CAZAUNOUS 31020      47   56     4       1
3   31348  MONCAUP  31020      26   57     2       1
```
Simple random sampling without replacement

- Selection of the simple random sampling without replacement of size \( n = 70 \):
  
  ```r
  si.rec <- srswor(70, 554)
  ```

  Return a vector of size \( N = 554 \) containing \( n = 70 \) ones and \( N - n = 484 \) zeros;

- The inclusion probability vector and extraction of the sample information:
  
  ```r
  pik.si <- rep(70/554, 70)
  si.Logvac <- rec$LOGVAC[which(si.rec == 1)]
  ```

- The Horvitz-Thompson estimator and its variance:
  
  ```r
  Logvac_si <- HTestimator(si.Logvac, pik.si)
  > 15379.04
  > Logvac_var <- varHT(si.Logvac, pikl.si, method = 1)
  ```
Estimation of totals with survey (1)

The creation of the design object:

```
ech.si <- svydesign(id=~CODE_N, weights=rep(554/70,70),
fpc=rep(70/554,70),data=rec[which(si.rec==1),])
```

- **id**: the label unity, always asked; \((id = 1)\) means no cluster
- **weights**: formula or vector of inclusion probabilities;
- **fpc**: formula or vector with the finite population correction (same size as weights); if fpc not specified, then the sample is with replacement;
- **data**: the population within the sample is selected containing the variable values;
- **weights, fpc, data** are optional.

```
#the estimation of the total and standard-deviation
> svytotal(~LOGVAC, ech.si)
      total     SE
LOGVAC 13613 2488.4
```
Estimation of totals with survey (2)

```r
### getting data from the object ech.si
> attributes(ech.si)

$names
[1] "cluster" "strata" "has.strata" "prob"
 "allprob" "call" "variables"
[8] "fpc" "pps"

$class
[1] "survey.design2" "survey.design"
### inclusion probabilities and the variable values
> ech.si$allprob
> ech.si$variables
```
Estimation of totals with survey (3)

```r
> summary(ech.si)

Independent Sampling design
svydesign(id = ~CODE_N, weights = rep(554/70, 70), fpc = rep(70/554, 70), data = rec[which(si.rec == 1), ])

Probabilities:
       Min. 1st Qu. Median Mean 3rd Qu. Max.  
  0.1264  0.1264  0.1264  0.1264  0.1264  0.1264
Population size (PSUs): 554
Data variables:
[1] "CODE_N"  "COMMUNE"  "BVQ_N"  "POPSDC99"  "LOG"  "LOGVAC"  "stratlog"
```
Unequal with replacement sampling designs and Hansen-Hurvitz estimator

- We select (with replacement) a first individual $k$ with probability of selection $p_k$, we repeat $m$ times; a unit may be selected several times;

- The total $t_Y$ is estimated by the Hansen-Hurvitz estimator:

$$\hat{t}_{HH} = \frac{1}{m} \sum_{i=1}^{m} \frac{y_{ki}}{p_{ki}} = \sum_{k=1}^{N} S_k \frac{y_k}{mp_k}$$

where $S_k = \text{no. of times } k \text{ has been selected}$.

- Selection of a sample with the function `UPmultinomial`

```r
> pik=c(0.2,0.7,0.8,0.5,0.4,0.4)
> sum(pik)###the number $m$ of selections is implicitly contained in pik [1] 3
> (s.pps=UPmultinomial(pik))
[1] 0 1 1 0 0 1
```
Selection of a with replacement sample proportional to \( \text{LOG} \):

\[
> \text{tLOG} = \text{sum(\text{LOG})} \\
> \text{tLOG} \\
### 197314 \\
> \text{pk} = \text{LOG/tLOG} ### \\
> \text{pik} = 70 * \text{pk} \\
> \text{s.pps} = \text{UPmultinomial(pik)} \\
> \text{weights.pps} = (1/\text{pik}) * \text{s.pps}
\]

Creation of the design object, no fpc option for the with-replacement sample:

\[
> \text{ech.pps} = \text{svydesign(id=~CODE_N, weights=weights.pps[weights.pps>0], data=rec99[which(s.pps!=0),])}
\]

Total estimation:

\[
> \text{svytotal(~LOGVAC, ech.pps)} \\
\text{total SE} \\
\text{LOGVAC 12609 1024.9}
\]
Unequal without replacement sampling designs

Package sampling provides the following unequal without replacement sampling designs:

- **UPpoisson** gives Poisson sampling with inclusion probabilities $\pi_k$; for constant $\pi_k = \pi$ for all $k \in U$, we obtain the Bernoulli sampling;

  ```r
  # selection of a Bernoulli sample with pi=70/554
  pik = rep(70/554, 554)
  s.be = UPpoisson(pik)
  ```

- **UPsystematic** gives systematic sampling with inclusion probabilities $\pi_k = \frac{nx_k}{\sum_U x_k}$ where $x_k$ is an auxiliary variable; for $\pi_k = \frac{n}{N}$, we have the usual systematic sampling;

  ```r
  # selection of a proportional to LOG systematic sample of size n=70
  pk = LOG/tLOG
  pik = 70*pk
  sys.rec = UPsystematic(pik)
  ```

- the balanced sampling, the Brewer, Sampford, Tillé, ... samplings.
The balanced sampling : the cube method

- Consider the auxiliary variables $X_1, \ldots, X_p$ and let
  \[ x'_k = (X_{1k}, \ldots, X_{pk}) \quad \text{pour} \quad k = 1, \ldots, N \]

- Let $d_k = 1/\pi_k$, $k \in U$ be the sampling weights;

- A balanced sample $s$ is selected by the cube method (Deville and Tillé, 2004)
  \[ \sum_{k \in s} d_k x_k = \sum_{k \in U} x_k \]
The balanced sampling with R

Packages `sampling` and `balancedsampling` of Anton Grafstrom (Swedish Univ. of Agriculture Science); fonction `cube` very fast thanks to C++ and Rcpp;

\[
\text{cube}(\text{prob}, \text{Xbal})
\]

- `prob` : vector of inclusion probabilities, size N ;
- `Xbal` : matrix of balancing auxiliary variables of N rows and r columns.

Fonctions `lpm1`, `lpm2` implements the local pivot in order to select a spatially balanced sample; used for the national seashore inventory in Sweden, 2015;

Insee plans to use spatially balanced samples for the master sampling and several methods are actually analyzed.
Example of application for spatial sampling

Example from the A. Grafstrom’s webpage:

```r
N = 1000; # population size
> n = 100; # sample size
> p = rep(n/N,N); # inclusion probabilities
> X = cbind(runif(N),runif(N)); # matrix of auxiliary variables
> s = lpm1(p,X); # select sample

> plot(X[,1],X[,2],xlab="x",ylab="y",pch=19,col="gray");
> points(X[s,1],X[s,2],pch=19,col="red");
```
Stratified sampling design with strata fonction

#### the LOG-optimal allocation
```r
> sdlog = tapply(LOG, stratlog, sd)
> n = 50; N = 554
> nech = round(n*N*sdlog/sum(N*sdlog))
### 1 2 3 4
### 3 6 13 28
#### the strata fonction
> stsi.rec = strata(rec, "stratlog", size = nech, method = "srswor")
> stsi.data = getdata(rec, stsi.rec)
#### the estimation and standard deviation
> N1 = 221; N2 = 169; N3 = 110; N4 = 54
> n1 = nech[1], n2 = nech[2], n3 = nech[3], n4 = nech[4]
> poids_stsi = c(rep(N1/n1, n1), rep(N2/n2, n2), rep(N3/n3, n3), rep(N4/n4, n4))
> fpcst = c(rep(n1/N1, n1), rep(n2/N2, n2), rep(n3/N3, n3), rep(n4/N4, n4))
> ech.stsi = svydesign(id = ~CODE_N, strata = ~stratlog, weights = poids_stsi, fpc ~ fpcst, data = stsi.data)
> svytotal(~LOGVAC, ech.stsi)
```
More on stratified sampling

- package `stratification` construct strata according to the Lavallée-Hidiroglou method;

- package `SamplingStrata` determines the optimal stratification of sampling frames for multipurpose sampling surveys;
Sample selection and estimation of totals

Multi-stages sampling designs

Improving the efficiency of the Horvitz-Thompson estimator

More advanced issues
Cluster sampling

Fonction `mstage` from package `sampling` allows selecting a multi-stage sample with equal/unequal probabilities and `cluster` a cluster sample;

```r
###the function  cluster to select a sample
grap.rec=cluster(rec,clusternames="BVQ_N",size=4,method="srswor")
grap.rec[1:5,]
   BVQ_N ID_unit Prob
 1  31239  276 0.125
 2  31239  273 0.125
 3  31239  274 0.125
 4  31239  275 0.125
 5  31239  262 0.125
###getting the variables values
grap.data=getdata(rec,grap.rec)
###the estimation and standard deviation
nech=nrow(grap.data)
ech.grap=svydesign(id=~BVQ_N,weights=rep(32/4,nech),
data=grap.data,fpc=rep(4/32,nech)
svyytotal(~LOGVAC,ech.grap)
   total    SE
LOGVAC 10512 3058.1
```
Sample selection and estimation of totals

Multi-stages sampling designs

Improving the efficiency of the Horvitz-Thompson estimator

More advanced issues
The calibration estimator (Deville and Sarndal, 1992)

- Consider the auxiliary variables $x_1, \ldots, x_p$ and let

$$x_k' = (X_{1k}, \ldots, X_{pk}) \quad \text{pour} \quad k = 1, \ldots, N$$

- Find weights $w_{s}^{\text{cal}} = (w_{ks}^{\text{cal}})_{k \in s}$ such that

$$w_{s}^{\text{cal}} = \arg\min_{w} \Upsilon_{s}(w)$$

subject to the calibration constraints

$$\sum_{k \in s} w_{ks}^{\text{cal}} x_k = t_x.$$ 

- The calibrated estimator is

$$\hat{t}_y^{\text{cal}} = \sum_{k \in s} w_{ks}^{\text{cal}} y_k.$$
Properties of the calibration estimator

- Several distance functions $\Upsilon_s(\mathbf{w})$ have been considered in Deville and Sarndal (1992): the chi-squared, raking, logit distances;

- It was proved that the calibration estimator is asymptotically equivalent to the one obtained for the chi-squared distance:

$$\hat{t}_{GREG} = \sum_{k \in s} w_k s y_k = \sum_{k \in s} d_k y_k - \left( \sum_{k \in s} d_k x_k - \sum_{k \in U} x_k \right)' \hat{\beta}_X$$

where $\hat{\beta}_X = (\sum_{k \in s} d_k x_k x_k')^{-1} \sum_{k \in s} d_k x_k y_k$.

- The asymptotic variance of $\hat{t}_{y}^{cal}$ is

$$AV(\hat{t}_{y}^{cal}) = \sum_{k \in U} \sum_{k \in U} (\pi_k l - \pi_k \pi_l) \frac{y_k - x_k' \tilde{\beta}_X}{\pi_k} \frac{y_l - x_l' \tilde{\beta}_X}{\pi_l}$$

where $\tilde{\beta}_X = (\sum_U x_k x_k')^{-1} \sum_U x_k y_k$. 
With R

- functions `calib`, `gencalib` in package `sampling` compute the calibration weights and the $g$-weights (quite slow);

- the `calibWeights` function in package `laeken` and `calibSample` function in package `simPop` faster (depending on the example) implementation of parts of `calib`;

- functions `svyratio`, `poststratified`, `calibrate` in package `survey` compute the calibration estimators and variance estimators obtained by Taylor linearization;

- very recent package `icarus` gives the calibration weights.
The ratio estimator: the svyratio function

- Determines first \( \hat{R} = \frac{\sum_s d_k y_k}{\sum_s d_k x_k} \) and its variance estimator by using the \texttt{svyratio} function;

\[
tLOG = \text{sum}(LOG) \\
[1] 197314 \\
(est.ratio <- \text{svyratio}(~LOGVAC, ~LOG, ech.si))
\]

Ratio estimator: \texttt{svyratio.survey.design2}(~LOGVAC, ~LOG, ech.si)

\texttt{Ratios=}

\begin{verbatim}
   LOG \\
LOGVAC  0.04795361
SEs=

   LOG \\
LOGVAC  0.004471045
\end{verbatim}

- The ratio estimator and its variance are obtained by using \texttt{predict}:

\begin{verbatim}
> \text{predict}(est.ratio, total=197314)
$\text{total}$

   LOG \\
LOGVAC  9461.918
$\text{se}$

   LOG \\
LOGVAC  882.1999
\end{verbatim}
The calibration estimator: the `calibrate` function

```
calibrate(design, formula, population, variance=NULL, 
           bounds=c(-Inf,Inf), calfun=c("linear","raking","logit"), 
           maxit=50, epsilon=1e-7, trim=NULL, ...)
```

- **design**: the survey design objet;
- **formula**: model formula for calibration model
- **population**: vectors of population totals
- **calfun**: distance functions used in calibration
- **bounds**: bounds for the calibration weights
- **variance**: if not NULL, then the calibration variance is proportional to the linear combination of those columns of the model matrix
Example of application

- The calibration estimator on rec99 data
  ```
  > ech.cal<-calibrate(ech.si,~LOG,c(554,197314))
  > svytotal(~LOGVAC, ech.cal)
  total       SE
  LOGVAC    9853.5  858.84
  ```

- The ratio estimator can be also obtained as follows:
  ```
  > ech.cal2<-calibrate(ech.si,~LOG-1,197314,variance=1)
  > svytotal(~LOGVAC,ech.cal2)
  total       SE
  LOGVAC    9461.9  882.2
  ```
The calibration estimator: package Icarus

- The package **Icarus** of Antoine Rebecq (ex Insee, now Ubisoft, Montreal) has been suggested for the users from the INSEE (Institut National de la Statistique et des Etudes Economiques);
- It is inspired from the very popular macro CALMAR ("calage sur marges") for SAS and it has been already used at INSEE;
- Nonresponse can be handled (idem to CALMAR 2);
- New: **penalized calibration** (ridge type) is also achieved by this package;
- Calibration estimators of totals and means may be obtained, but the variance estimation is not provided, just like CALMAR.
The calibration estimator with calibration

calibration(data, marginMatrix, colWeights, method = "linear", bounds = NULL, q = NULL, maxIter = 2500, ...)

- **data**: survey data;
- **marginMatrix**: population totals of auxiliary variables;
- **colWeights**: the sampling weights;
- **method**: the method used to calibrate ("linear", "raking", "logit", "truncated");
- **bounds**: vector of lower and upper bounds for the calibration weights;
- **q**: vector of $q_k$;
Example of application

tLOG=sum(LOG)
# [1] 197314
si.rec=srswor(70,554)
##### population totals
m_1=c("taillepop",0,554)
m_LOG=c("LOG",0,197314)
marges_rec=rbind(m_1,m_LOG)

poids=rep(554/70,70)
taillepop=rep(1,70)

essai=cbind(rec[which(si.rec==1),],taillepop,poids)

poids_cal=calibration(data=essai,marginMatrix=marges_rec,colWeights="poids",
bounds=c(0.4,2.1),description=TRUE)

logvac_cal=weightedTotal(essai$LOGVAC,poids_cal)
# [1] 9853.461
Surveys with a very large number of auxiliary variables

Nowadays, we may have many auxiliary variables.

- Surveys of national statistical institutes (INSEE, Statistics Canada, ...)

- Big data and surveys

- Surveys with functional data (electricity load curves) at EDF, the French Company of Electricity or Médiamétrie (the French company of measuring audience)
Estimation with over-calibration

Using a very large set of auxiliary variables can entail:

1. negative, very large and unstable weights $w_{ks}$; difficulty of satisfying the predefined benchmarks on the weight ratio $w_{ks}/d_k$;

2. impossibility of computing the calibration weights when the matrix $\sum_{k \in S} d_k x_k x_k^T$ is non-invertible;

3. increase of the variance of $\hat{t}_{yw}$ (Silva and Skinner, 1997; Chauvet and Goga, 2017):

$$\frac{1}{N}(\hat{t}_{GREG} - t_y) = \frac{1}{N}(\hat{t}_{y,diff} - t_y) + O_p \left( \frac{p \sqrt{p}}{n} \right)$$

where $\hat{t}_{y,diff} = \hat{t}_{yd} - (\hat{t}_{xd} - t_x)'\tilde{\beta}_X$. 
Suggested solutions

1. choice of the most important variables; with multipurpose surveys, this choice may be difficult to put into practice.

2. use of a generalized inverse

3. relaxing the calibration equations:
   - using penalized calibration via the *ridge regression* (Bardsley and Chambers, 1984; Chambers, 1996; Rao and Singh, 2009; Guggemos and Tillé, 2010; Beaumont and Bocci, 2008).
   - using the *principal component regression* (Cardot, Goga & Shehzad, 2017) : $r$ new calibration variables with $r << p$ which keep the maximum of ”information” from the initial variables.
The penalized calibration estimator

- The calibration equations are "released" : they are only approximatively satisfied and we control the error between $\hat{t}_{xw}$ and $t_x$ by means of a penalty;

- We search for the weights $w_{ks}^{\text{pen}}, k \in s$ satisfying

$$w_{ks}^{\text{pen}}(\lambda) = \arg\min_w \sum_{k \in s} \left( \frac{w_k - d_k}{d_k} \right)^2 + \lambda \sum_{j=1}^{p} C_j (\hat{t}_w, x_j - t_{x_j})^2$$

where $C_j$ is a user-specified cost associated with not satisfying the $j$th calibration equation;

  - if $\lambda C_j = 0$, then the constrain upon $t_{x_j}$ is not considered;
  - if $\lambda C_j$ very large, then the constraint upon $t_{x_j}$ is exactly satisfied;

- Already used at Statistics Canada and INSEE (French Statistical Institute).
The penalized estimator and its asymptotique variance

- The penalized weights are given by:
  \[ w_{k,s}^{\text{pen}} = d_k - d_k x_k' \left( \sum_{k \in s} d_k x_k x_k + \lambda^{-1} C^{-1} \right)^{-1} (\hat{t}_x d - t_x), \quad k \in s \]

- The penalized calibration estimator is a GREG-type estimator with a ridge coefficient of regression:
  \[ \hat{t}_{yw}^{\text{pen}}(\lambda) = \sum_{k \in s} d_k y_k - \left( \sum_{k \in s} d_k x_k - \sum_{k \in U} x_k \right)' \hat{\beta}_X(\lambda) \]

where \( \hat{\beta}_X(\lambda) = \left( \sum_{k \in s} d_k x_k x_k + \lambda^{-1} C^{-1} \right)^{-1} \sum_{k \in s} d_k x_k y_k \).

- The asymptotic variance of \( \hat{t}_{yw}^{\text{cal}} \) is
  \[ AV(\hat{t}_{yw}^{\text{pen}}(\lambda)) = \sum_{k \in U} \sum_{k \in U} (\pi_{kl} - \pi_k \pi_l) \frac{y_k - x_k' \hat{\beta}_X(\lambda)}{\pi_k} \frac{y_l - x_l' \hat{\beta}_X(\lambda)}{\pi_l} \]
Penalized calibration with calibration

calibration(data, marginMatrix, colWeights, method = "linear", bounds = NULL, q = NULL, costs = NULL, gap = NULL, maxIter = 2500, lambda = NULL, ...)

- **costs**: vector of $C_j$ costs, they must be given;
- **gap**: $\max(w_k/d_k) - \min(w_k/d_k)$
- **lambda**: the initial $\lambda$ used in penalized calibration; by default, chosen automatically by the algorithm.

No need to consider bounded distances this is why only the chi-square and the raking distance are considered.
Calibration on principal components

- The alternative to penalized calibration is to “compress” the information contained in the $X$-matrix by considering principal components analysis;

- Consider the principal components $Z_1, \ldots, Z_p$

  \[ Z_j = Xv_j, \quad j = 1, \ldots, p. \]

  where $v_j$ is the $j$th eigenvector associated to the $j$th eigenvalue $\lambda_j$ of $N^{-1} X^\top X$.

- We search for weights $w_{k,s}^{pc}, k \in s$ satisfying :

  \[ w_{k,s}^{pc} = \arg\min_w \sum_{k \in s} \frac{(w_k - d_k)^2}{d_k} \]

  subject to

  \[ \hat{t}_w, Z_j = tZ_j \quad \text{for} \quad j = 1, \ldots, r \quad \text{with} \quad r << p. \]
The PC calibration estimator and its asymptotic variance

- The penalized calibration estimator is a GREG-type estimator with a principal component coefficient of regression:

$$
\hat{t}_{yw}^{pc}(r) = \sum_{k \in s} d_k y_k - \left( \sum_{k \in s} d_k x_k - \sum_{k \in U} x_k \right)' \hat{\beta}_X^{pc}(r)
$$

where

$$
\hat{\beta}_X^{pc}(r) = (v_1, \ldots, v_r)' \left( \sum_{k \in s} d_k x_k x_k \right)^{-1} \sum_{k \in s} d_k x_k y_k.
$$

- The asymptotic variance of $\hat{t}_{yw}^{cal}$ is

$$
AV(\hat{t}_{yw}^{pen}(\lambda)) = \sum_{k \in U} \sum_{k \in U} (\pi_{kl} - \pi_k \pi_l) \frac{y_k - x_k' \hat{\beta}_X^{pc}(r)}{\pi_k} \frac{y_l - x_l' \hat{\beta}_X^{pc}(r)}{\pi_l}
$$

- If no complete auxiliary information is available, then the PC variables $Z_j$ may be estimated by $\hat{Z}_j = X \hat{v}_j$ and consider calibration on the zero-totals of $\hat{Z}_j$ (Cardot, Goga, Shehzad 2017).
PC estimation with `svyprcomp`

```
svyprcomp(formula, design, center = TRUE, scale. = FALSE, scores = FALSE, ...)
```

- **formula**: model formula describing variables to be used
- **design**: the survey design object
- **center = TRUE**: data is centered by default but not scaled

The value is an object similar to `prcomp` but with survey design information.
Data description

- Commission for Energy Regulation (Ireland)
  http://www.cer.ie/

- A (sous) population of size $N = 6291$ individuals (households and companies).

- The electricity consumption is recorded every 30 minutes during several months.

- We consider a study period of 14 consecutive days. The study parameter is the total electricity consumption during the second week ($t_y = \sum_{k \in U} \sum_{d=1}^{336} Y_k(t_d)$).
  The auxiliary information is the load electricity curve during the first week ($X_k(t_d), d = 1, \ldots, p = 336$).
A sample of 5 load curves during the 1st week
Distribution of weights for calibration on 336 variables, 
n=1000 SRS
Small simulation

- We select $I = 1000$ SRS samples of size $n = 600$;
- We compute the Greg estimator on 336 variables and PC calibration estimator;
- Our benchmarks is the estimator $\hat{t}_{\ell w}$ calibrated on all the $p = 336$ auxiliary variables. For each day $\ell$, the performances of an estimator $\hat{\theta}$ of the total $t_\ell$ were measured by considering the relative mean squared error,

$$R_\ell(\hat{\theta}) = \frac{\sum_{i=1}^{I} (\hat{\theta}(i) - t_\ell)^2}{\sum_{i=1}^{I} (\hat{t}_{\ell w}(i) - t_\ell)^2}$$
Coefficient of variation of weights for n=600
Distribution of errors $\| \sum_{k \in S} w_k x_k - t_x \|^2$

Sample principal components

Camelia Goga
Survey Samplings with R
Sample selection and estimation of totals

Multi-stages sampling designs

Improving the efficiency of the Horvitz-Thompson estimator

More advanced issues
More advanced issues

- **survey** provides univariate quantile estimation (but multivariate quantile with **Gmedian**), two-sample tests, rank tests, generalised linear models, cumulative link models, Cox models, loglinear models, ...

- **laeken** and **convey** for inequality indicators (Gini, Theil, ...)

- many packages on missing data but most of them are model-based

Still, work must be done on nonparametric regression with survey data, robust estimation for totals, indirect sampling, ...
10ᵉ COLLOQUE FRANCOPHONE SUR LES SONDAGES
24 au 26 octobre 2018
UNIVERSITÉ DE LYON - FRANCE
Some references

- Chauvet, G. and Goga, C. (2017), Selecting the calibration variables by a bootstrap method, in work.