TEACHING SURVEY SAMPLING WITH THE 'SAMPLING' R PACKAGE

Yves Tillé and Alina Matei
Institute of Statistics, University of Neuchâtel, Switzerland
yves.tille@unine.ch

The R language is a free software environment for statistical computing and graphics. This software is complemented by almost 2000 packages developed by researchers. R is thus the most complete statistical software environment. At the University of Neuchâtel, Alina Matei and Yves Tillé have developed the 'sampling' R package that contains a set of tools for selecting and calibrating samples. Modern procedures, like the cube algorithm for selecting balanced samples or calibration with several distances and bounds, are implemented. This package is very easy to use and is thus an interesting tool for teaching. Simulations can be done very quickly. Different methods of sampling and calibration can be run and compared. We will present several exercises that can be done quickly and that allow a practical application of the sampling theory.

INTRODUCTION

R is a free software environment for statistical computing and graphics. The basis is completed by a set of almost two thousands complementary packages in all the fields of statistics. R can be downloaded freely from the address: http://www.r-project.org/. The 'sampling' R package (Tillé and Matei, 2009) is set of sampling and estimation functions developed at the Institute of Statistics of the University of Neuchâtel. It allows us to select statistical units from a population by means of complex sampling designs such as: stratification, cluster sampling, balanced sampling, and two-stage sampling. The samples can be selected with equal or unequal inclusion probabilities. For unequal probability sampling, several algorithms are implemented. The package also allows us to compute the weights for a calibrated estimator and its variance estimator, and to handle nonresponse. Three data basis are included in the package: the MU284 population from the book of Särndal, Swensson and Wretman (1992), and data on the Belgian and Swiss municipalities coming from the Belgian and Swiss national institutes of statistics. Several examples illustrate each function. The 'sampling' R package fulfils two main objectives:

- It is a free software for survey processing;
- It offers functions similar to commercial software.

The package was created in 2005 in order to be used as a pedagogical tool for advanced courses on survey methodology organised by the Swiss Federal Statistical Office under the aegis of the European Free Trade Association (EFTA). The package is continually improved and the current version is 2.2.

In order to install the package (version 2.2) during an R session, one can use the command: *install.packages("sampling")*. In order to load the package, one uses: *library(sampling)* The list of all the available functions can be obtained using: *help(package=sampling)*. The function names are given below in italic. For instance, *UPpoisson* is the function for selecting a sample using a Poisson sampling design. Help on this function can be obtained by using: *help(UPpoisson)*.

FUNCTIONS OF THE R PACKAGE SAMPLING

The sampling package contains functions to select samples with or without replacement. A simple random sampling (with equal probabilities) can be selected by means of two functions (*srswor, srswor1)* for sampling without replacement, and *srswr* for sampling with replacement. The following sampling designs with unequal inclusion probabilities are implemented in the package: the Brewer design (1963), the maximum entropy design (Hájek, 1964), the Midzuno design (1952), the multinomial design (Hansen and Hurwitz, 1943), the pivotal method (Deville and Tillé, 1998), the ordered design (Rosén, 1997), the Poisson design (Hájek, 1958), the minimal support design (Deville and Tillé, 1998), the systematic design (Madow, 1949), the Sampford design (1967) and the Tillé design (1996). The function names for selecting samples with unequal inclusion probabilities start by the prefix UP ("unequal probability"). The list of the UP-functions is:

In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia*. Voorburg, The Netherlands: International Statistical Institute. www.stat.auckland.ac.nz/~iase/publications.php [© 2010 ISI/IASE]

UPbrewer, UPmaxentropy, UPmidzuno, UPmultinomial, UPpivotal, UPopips, Prandompivotal, UPpoisson, UPminimalsupport, UPsampford, UPsystematic, UPrandomsystematic, UPtille. Additional functions for computing the joint inclusion probabilities are available for the following sampling designs: maximum entropy, Midzuno, systematic and Tillé.

Complex designs are also implemented in the package. It is possible to select a sample by means of:

- a stratified design (*strata*), with equal or unequal inclusion probabilities;
- a cluster sampling design (*cluster*), with equal or unequal inclusion probabilities;
- a multi-stage sampling design (*mstage*), using a stratified design, a cluster sampling or a simple random sampling at each stage, with equal or unequal inclusion probabilities.

A sample is said to be balanced on a set of balancing variables if the Horvitz-Thompson estimators of totals of these variables are equal to the population totals for all these variables. The cube method (Deville and Tillé, 2004; Tillé, 2006) is a general algorithm that allows selecting balanced sample with equal or unequal inclusion probabilities (*samplecube*). The algorithm is made of two phases: the flight phase and the landing phase that can be run separately (*fastflightcube, landingcube*). Additional functions are also available for balanced stratified sampling, cluster sampling, and two-stage sampling (*balancedstratification, balancedcluster, balancedtwostage*). In survey sampling, the basic Horvitz-Thompson estimator can be used to estimate a total without bias. It can be computed using the functions *HTestimator* and *HTstrata*.

Calibration (Deville and Särndal, 1992) is a technique that uses auxiliary information to weight the statistical units. This technique improves the Horvitz-Thompson estimator. The weights of the calibrated estimator are close to the weights of the Horvitz-Thompson estimator, which ensures that the calibrated estimator is almost unbiased. Moreover the calibrated estimator is equal to the population total for all the benchmarking variables. Calibration allows us to reduce the length of the confidence interval by taking advantage of the auxiliary information given by the auxiliary variables whose the totals are known in the population. The *calib* function allows us to compute the "g-weights" for calibrated estimators and to choose the calibration function and eventually bounds on the weights. Four methods are implemented: "linear", "raking", "truncated", "logit". The *checkcalibration* function checks the calibration result and indicates if the calibration is possible or not. The calibrated estimator and its variance are calculated by means of the *calest* function.

Like the calibration, poststratification also allows us to improve the accuracy of the estimates. Suppose that we know the distribution by sex/age in the population, but that it is not possible to use this information at the sampling stage (by stratification), because the auxiliary variables are not available at the unit level. This information can however be use after the sample selection by modifying the sampling weights, in order to calibrate on the population frequencies. Poststratification is implemented by means of the *poststrata* function, and the poststratified estimator is computed using *postest.*

Other estimators using the auxiliary information are available:

- the ratio estimator (*ratioest, ratioest_strata*);
- the generalized regression estimator (*regest, regest_strata*), with the computation of regression coefficients, the standard error, the p-value of the Student test, the variance-covariance matrix. This information is not available with the *calib* function.

Whatever is the quality of the sampling design and the data collection, the sample is always distorted because of nonresponse. Unit nonresponse means all the responses are missing for a statistical unit. Item nonresponse means that only some questions of the questionnaire are missing.

In case of the unit nonresponse, the units are generally reweighted using the inverse of a response probability estimator. Two approaches are implemented:

- creation of response homogeneous groups (when the respondents have an homogeneous response into a group; see *rhg, rhg_strata*)

- reweighting based on statistical models (one predicts the response probabilities by means of a logistic regression; see *rmodel*).

Both approaches can be used in the calibration function *calib*. Additional functions are available in order to:

- compute the inclusion probabilities for a sampling design with probability proportional to size (*inclusionprobabilities*);
- compute the inclusion probabilities for a stratified sampling design (*inclusionprobastrata*);
- extract information from a data base for a selected sample (*getdata*);
- list all the possible samples with fixed sample size (*writesample*);
- delete the empty strata (*cleanstrata*);
- create a disjunctive codification from a categorical variable (*disjunctive*).

EXAMPLES 1: CALIBRATION AND ADJUSTMENT FOR NONRESPONSE

- one generates artificial data, a set of data with 235 observations and 4 variables: “state” (with two categories A/B), “region” (with three categories), “income” (a continuous variable), “sex” (with two categories, male- 1/female - 2); this data set is our finite population.
- one computes the inclusion probabilities by using the variable ‘income’ and a random noise following the Uniform (0, 1) distribution.
- one selects a sample of size 75 from the generated population using the function *UPmaxentropy* (maximum entropy sampling).
- one adds the “status” variable that indicates if a selected units has answered or not.
- one creates response homogeneous groups according to the “region” and “state” variables (*rhg*);
- one applies the calibration technique using the known population totals of the “sex” variable.
- one computes the calibrated estimator for the “income” variable and its variance estimator; finally a 95% confidence interval for the population income is computed.

The R code is given below. One uses '#' to include comments.

```
# Generation of artificial data
data = rbind(matrix(rep("A", 165), 165, 1, byrow = TRUE),matrix(rep("B", 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")
# Computation of the inclusion probabilities
pik=inclusionprobabilities(data$income+runif(1),75)
# The inclusion probabilities are added to the data base
data=cbind.data.frame(data,Prob=pik)
# Selection of a sample by maximum entropy sampling
s=s1=UPmaxentropy(pik)
# Extraction of the information from the data base for the selected sample
s=getdata(data, s)
# Variable 'status' is randomly generated using a Uniform(0,1) random variable
# 1 means that a unit answers, 0 otherwise
# this variable is added to the data base
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)
# Computation of the response homogeneous groups
# using variables 'region' and 'state'
s = rhg(s, selection = c("region","state"))
# Selection of the sample of respondents
```

```
sr = s[s$status == 1,]
# Creation of the auxiliary matrix X containing the variables 'sex' and 'age'
# at the population level;
# the sample will be calibrated on the 'sex' variable
X = matrix(0, nrow = nrow(data), ncol = 2)
for (i in 1:nrow(data)) {
   if (data$sex[i] == 1)
   X[i, 1] = 1
   if (data$sex[i] == 2)
   X[i, 2] = 1
}
# One computes the population totals for the categories of the 'sex' variable
total = c(t(rep(1, nrow(data))) %*% X)
# Creation of the matrix Xs containing the information about the 'sex' variable,
# on the selected sample
Xs = matrix(0, nrow = nrow(sr), ncol = 2)
for (i in 1:nrow(sr)) {
   if (sr$sex[i] == 1)
   Xs[i, 1] = 1
   if (sr$sex[i] == 2)
   Xs[i, 2] = 1
}
# The population totals are
total
# The sample totals are
colSums(Xs)
# Computation of the initial weights by using the inclusion probabilities
# and the probabilities of response
d = 1/(sr$Prob * sr$prob_resp)
# Computation of the g-weights for calibration using the 'raking' method
g=calib(Xs, d, total, method = "raking", description=TRUE)
# Checking of the calibration validity
checkcalibration(Xs, d, total, g)
# Computation of the joint inclusion probabilities
pikl=UPmaxentropypi2(pik)
# Extraction of the joint inclusion probabilities of the respondents
pikls=pikl[sr$ID_unit,sr$ID_unit]
# The variable of interest is the income
Ys=sr$income
# Computation of the calibrated estimator
est=calibev(Ys,Xs,total,pikls,d,g,with=FALSE,EPS=1e-6)
print(est)
# Computation of a confidence interval at 95%
cat("[",est$calest-1.96*sqrt(est$evar),",",est$calest+1.96*sqrt(est$evar),"]\n")
```

EXAMPLE 2: UNEQUAL PROBABILITY SAMPLING

This is an example of unequal probability (UP) sampling functions: selection of samples using the Belgian municipalities' data set, with equal or unequal probabilities, and comparison of the Horvitz-Thompson estimator accuracy using boxplots. The following sampling schemes are used: Poisson, random systematic, random pivotal, Tillé, Midzuno, systematic, pivotal, and simple random sampling without replacement. Monte Carlo simulations are used to study the accuracy of the Horvitz-Thompson estimator of a population total. The sample size is 200 in each simulation. The aim of this example is to demonstrate the effect of an auxiliary information incorporation in the sampling design. We use:

- some πps sampling designs with Horvitz-Thompson estimation, using in the sampling design the information on size measures of population elements;
- simple random sampling without replacement with Horvitz-Thompson estimation. The result of the program is given in Figure 1.

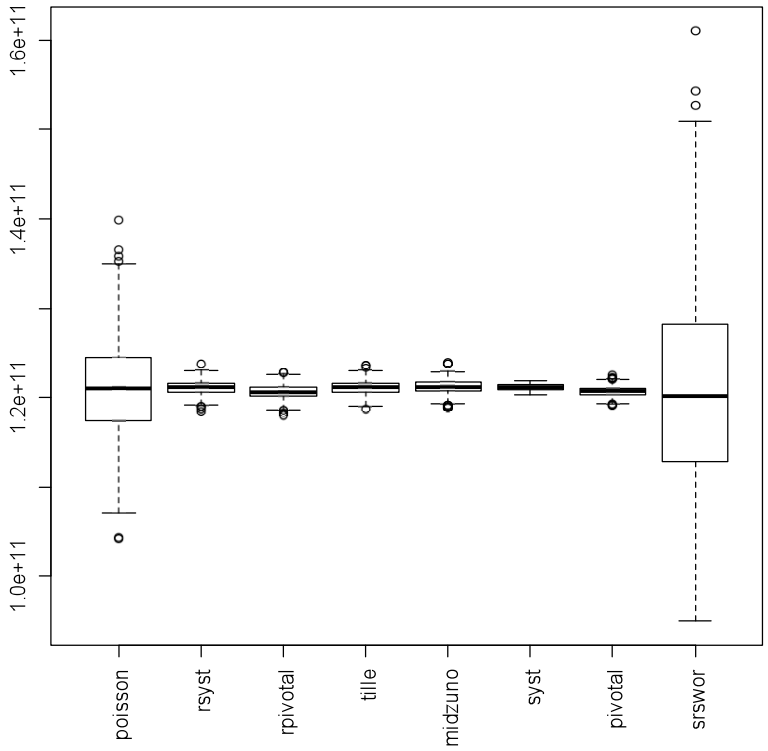

Figure 1. comparison of the accuracy if the Horvitz-Thompson estimator under 8 sampling designs

```
# Load the data set
b = data(belgianmunicipalities)
# Computes the inclusion probabilities using the 'Tot04' variable and a sample
size of 200
pik = inclusionprobabilities(belgianmunicipalities$Tot04,200)
# Defines the population size
N = length(pik)
# Defines the sample size
n = sum(pik)
# Defines the number of simulations (for an accurate result, increase this value)
sim = 1000
ss = array(0, c(sim, 8))
# Defines the variable of interest ('Taxable Income')
y = belgianmunicipalities$TaxableIncome
# Simulation and computation of the Horvitz-Thompson estimator
ht = numeric(8)
for (i in 1:sim) {
   cat("Step ", i, "\n")
   s = UPpoisson(pik)
   ht[1] = HTestimator(y[s == 1], pik[s == 1])
   s = UPrandomsystematic(pik)
   ht[2] = HTestimator(y[s == 1], pik[s == 1])
   s = UPrandompivotal(pik)
   ht[3] = HTestimator(y[s == 1], pik[s == 1])
   s = UPtille(pik)
   ht[4] = HTestimator(y[s == 1], pik[s == 1])
   s = UPmidzuno(pik)
   ht[5] = HTestimator(y[s == 1], pik[s == 1])
   s = UPsystematic(pik)
   ht[6] = HTestimator(y[s == 1], pik[s == 1])
   s = UPpivotal(pik)
   ht[7] = HTestimator(y[s == 1], pik[s == 1])
   s = srswor(n, N)
   ht[8] = HTestimator(y[s == 1], rep(n/N, n))
   ss[i, ] = ss[i, ] + ht
```

```
}
# boxplots of the estimators
colnames(ss)=c("poisson", "rsyst", "rpivotal", "tille", "midzuno", "syst",
"pivotal","srswor")
boxplot(data.frame(ss), las = 3)
```

USING THE 'SAMPLING' R PACKAGE FOR TEACHING

The 'sampling' R package allows us to quickly run examples with complex sampling designs, calibration and nonresponse treatment. With several lines of programme, students can simulate different strategies and compare for instance the accuracy obtained by calibration or balancing. It is thus a very interesting tool for practical works during a course of survey sampling. During our talk, we will propose several exercises with a full solution implemented in R.

REFERENCES

Brewer, K. R. W. (1963). A model of systematic sampling with unequal probabilities. *Australian Journal of Statistics, 5*, 5-13.

Deville, J.-C., & Tillé, Y. (1998). Unequal probability sampling without replacement through a splitting method. *Biometrika*, *85*, 89-101.

Deville, J.-C., & Tillé, Y. (2004). Efficient balanced sampling: the cube method. *Biometrika*, *91*, 893-912.

Deville, J.-C., & Särndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, *87*, 376–382.

Hájek, J. (1958). Some contributions to the theory of probability sampling. *Bulletin of the International Statistical Institute: Proceedings of the 30th session (Stockholm), ISI, The Hague*, *36*(3), 127-134.

Hájek, J. (1964). Asymptotic theory of rejective sampling with varying probabilities from a finite population. *Annals of Mathematical Statistics, 35*, 1491-1523.

Hansen, M. H., & Hurwitz, W. N. (1943). On the theory of sampling from finite populations. *Annals of Mathematical Statistics, 14*, 333-362.

Madow, W. G. (1949). On the theory of systematic sampling, II. *Annals of Mathematical Statistics, 20*, 333-354.

Midzuno, H. (1952). On the sampling system with probability proportional to sum of size. *Annals of the Institute of Statistical Mathematics, 3*, 99-107.

Rosén, B. (1997). Asymptotic theory for order sampling. *Journal of Statistical Planning and Inference*, 62, 135-158.

Sampford, M. R. (1967). On sampling without replacement with unequal probabilities of selection. *Biometrika*, *54*, 499-513.

Särndal, C.-E., Swensson, B., & Wretman, J. H. (1992). *Model Assisted Survey Sampling*. New York: Springer,

Tillé, Y. (1996). An elimination procedure of unequal probability sampling without replacement. *Biometrika*, *83*, 238-241.

Tillé, Y. (2006). *Sampling Algorithms*, Springer Series in Statistics. New York: Springer.

Tillé, Y., & Matei, A. (2009). *Sampling: Survey Sampling.* R package version 2.2. Online: http://cran.r-project.org/src/contrib/Descriptions/sampling.html.