WP.22
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Ljubljana, Slovenia, 9-11 May 2011)

Topic (iv): Micro editing – methods and software

Statistical Matching and Imputation of Survey Data with the Package "StatMatch" for the R Environment

Invited Paper

Prepared by Marcello D'Orazio, Italian National Institute of Statistics (Istat), Italy

I. Introduction

1. *Statistical matching* (hereafter denoted as SM) aims at integrating two data sources (usually data from sample surveys) referred to the same target population. In the usual SM framework, the variables **X** and *Y* are observed the survey *A*, while **X** and *Z* are observed in *B*; while the **X** variables are common to both the surveys, the variables *Y* and *Z* are not jointly observed. The SM techniques integrate *A* and *B* in order to investigate the relationship between *Y* and *Z*. This objective can be achieved through a *micro* or a *macro* approach (cf. D'Orazio *at al.*, 2006a). In the micro approach the SM aims at creating a "synthetic" data source in which all the variables, **X**, *Y* and *Z*, are available (usually *A* filled in with the values of *Z*). In the *macro* approach the data sources are used to derive an estimate of the interest parameter, e.g. the correlation coefficient between *Y* and *Z* or the contingency table $Y \times Z$. The SM can be performed in a *parametric* or in a *nonparametric* framework. The *parametric approach* requires the explicit adoption of a model, obviously if the model is wrong the results will not be reliable. The *nonparametric approach* does not require the explicit usage of a model and is more flexible in handling complex situations (a lot of variables of mixed type, categorical and continuous).

2. *Nonparametric micro* approach is very popular in SM. In fact, most of the applications of SM consist in creating the synthetic data set by filling *A* with the values of *Z* by means of a nonparametric imputation technique such as hot deck methods (*random hot deck*, *nearest neighbour hot deck*, etc.). When the objective of the SM is micro, it is possible to mix parametric and nonparametric methods. The *mixed methods* consists in fitting a model (all the parameters of the model are estimated) and then a nonparametric approach is used to create the synthetic data set. This approach permits to maintain the advantages of both the approaches. In the case of continuous variables several SM methods based on *predictive mean matching* are available (cf. Section 2.5 and 3.6 in D'Orazio *et al.*, 2006a). The following Table provides a summary of the objectives and approaches to SM (D'Orazio *et al.*, 2009):

Table 1 – Objectives and approaches of statistical matching

Objectives of Stat. matching	Approaches to statistical matching		
	Parametric	Nonparametric	Mixed
Macro	✓	✓	
Micro	✓	✓	✓

3. In the traditional SM framework when only *A* and *B* are available, all the SM methods (parametric, nonparametric and mixed) that use common variables **X** to match *A* and *B* implicitly assume the *conditional independence* (CI) of *Y* and *Z* given the **X** variables:

$$f(\mathbf{x}, y, z) = f(y \mid \mathbf{x}) f(z \mid \mathbf{x}) f(\mathbf{x})$$

This assumption is particularly strong and difficulty holds in practice. In order to avoid the CI assumption the SM should use some auxiliary information concerning the relationship between *Y* and *Z* (see Chapter 3 in D'Orazio *et al.*, 2006a). The auxiliary information can be at micro level (a new data source in which *Y* and *Z* or **X**, *Y* and *Z* are jointly observed) or at macro level (e.g. an estimate of the correlation coefficient ρ_{YX} or an estimate of the contingency table $Y \times Z$) or simply consist of some logic constraints about the relationship between *Y* and *Z* (structural zeros, etc.; for further details see D'Orazio *et al.* 2006b).

4. A different approach to SM consists in exploring uncertainty due to the lack of knowledge in the typical SM framework (only *A* and *B* are available). When the objective is macro this approach leads to conclude with an interval of plausible values for the interest parameter of the model chosen for (**X**,*Y*,*Z*). For instance, when (*X*,*Y*,*Z*) follow a multivariate normal distribution, it is possible to estimate all the elements of the correlation matrix with the exception of ρ_{YZ} ; given that the correlation matrix must be positive semidefinite, it comes out:

$$\hat{\rho}_{XY}\hat{\rho}_{XZ} - \left[\left(1-\hat{\rho}^2_{XY}\right)\left(1-\hat{\rho}^2_{XZ}\right)\right]^{1/2} \leq \rho_{YZ} \leq \hat{\rho}_{XY}\hat{\rho}_{XZ} + \left[\left(1-\hat{\rho}^2_{XY}\right)\left(1-\hat{\rho}^2_{XZ}\right)\right]^{1/2}$$

Small intervals denote low uncertainty and, in this case, the usage of methods based on the CI assumption can provide results not far from the true. In fact the estimate of the unknown parameter under the CI assumption is always included in the uncertainty interval (in the previous example it is the midpoint $\hat{\rho}^{CIA}_{YZ} = \hat{\rho}_{XY}\hat{\rho}_{XZ}$). When dealing with categorical variables, the uncertainty bounds for the cell probabilities in the contingency table $Y \times Z$ can be derived by considering the Fréchet classes (see Section II.E).

II. The package "StatMatch" for the R environment

A. Brief history

5. The package "StatMatch" for the R environment (R Development core team, 2011) is the result of a generalization and optimization of the code provided with the monograph about SM by D'Orazio *et al.* (2006a). The choice of disseminating code written in the R language responded to the need of having software that could be freely used by all the researchers interested in SM. The first version of StatMatch (version 0.4), made available to the R community through the CRAN (Comprehensive R Archive Network), was released in 2008. In the beginning of 2011 the version 1.0.1 has been released; this version presented a significant improvement of the functionalities of the previous version (0.8 released in 2009). It is worth noting that the functions in StatMatch are based uniquely on R code, there no calls to other external software or compiled C or Fortran codes. This choice favours the full portability of the package, that can be used in R under all the various operating systems (including 64 bit versions of MS windows)

6. The significant improvement to StatMatch from version 0.8 to 1.0.1 is essentially due to a series of activities about SM carried out in the context of the ESSnet on "Data Integration" funded by Eurostat[1]. The functions made available in the latest version of StatMatch can be divided in five main groups:

(a) functions to perform nonparametric SM at micro level by means of hot deck imputation (`NND.hotdeck`, `RANDwNND.hotdeck`, `rankNND.hotdeck`);
(b) a function to perform mixed SM at micro level for continuous variables (`mixed.mtc`);
(c) functions to integrate data from complex sample surveys through weights calibration as proposed by Renssen (1998) (`harmonize.x` and `comb.samples`);
(d) functions to explore uncertainty on the contingency table $Y \times Z$ (`Frechet.bounds.cat` and `Fbwidhts.by.x`);
(e) other functions to compute distances (`gower.dist` and `maximum.dist`), to create the synthetic data set (`create.fused`), etc.

[1] http://epp.eurostat.ec.europa.eu/portal/page/portal/essnet/data_integration

B. Nonparametric micro techniques

7. The *nearest neighbour distance hot deck* is implemented in the function `NND.hotdeck()`. This function searches in B (argument `data.don`) the nearest neighbour of each unit in A (`data.rec`); the distance is computed on the matching variables $\mathbf{X}_M$ (`match.vars`) which usually consist in a suitable subset of all the available common variables ($\mathbf{X}_M \subseteq \mathbf{X}$). By default the Manhattan (city block) distance is considered (`dist.fun="Manhattan"`). Many other distance functions can be used by resorting to the package “proxy” (Meyer and Buchta, 2010). For some particular distances it was decided to write specific R functions: `gower.dist()` and `maximum.dist()`. The first one permits to compute the Gowers’s dissimilarity (Gower, 1971) which can handle mixed type variables: it is an average of the distances computed on the single variables according to different rules, depending on the type of the variable. All the distances are scaled to range from 0 to 1, hence the overall distance cat take a value in [0,1]. The function `maximum.dist()` implements the maximum distance (L^{∞} norm); this function works on the true observed values (continuous variables) or on transformed values based on the ranks, as suggested in Kovar *et al.* (1988); the transformation (ranks divided by the number of units) removes the effect of different scales and the new values are uniformly distributed in the interval [0,1].
The function `NND.hotdeck()` allows to define some donation classes (`don.class`): for a record in given imputation class it will be selected a donor in the same class. Usually, the donation classes are defined according to one or more categorical common variables (geographic area, etc.) and permit to reduce the computational effort (the distances are computed only among units belonging to the same class).
In the following, a simple example of usage of `NND.hotdeck()` is reported. The example uses artificial data which resemble EU-SILC survey data and are generated by means of the R package “simPopulation” (Alfons and Kraft, 2010):

```
> install.packages("simPopulation") # install simPopulation
> library(simPopulation) # loads package simPopulation
> data(eusilcS) # artificial sample data based on EUSILC

> silc.16 <- subset(eusilcS, age>15) # select obs. with age>16
> nrow(silc.16) # no. of obs. with age>15
[1] 9522
> N <- round(sum(silc.16$rb050)) # estimate the pop (age>16) size
> N
[1] 67803

# simulates a SM framework
> X.vars <- c("hsize","db040","age","rb090","pb220a","rb050") #common vars.
> y.var <- "pl030"  # person’s economic status (7 categories)
> z.var <- "netIncome"  # personal net income (continuous var)
> n <- nrow(silc.16)
> set.seed(123456)
> obs.A <- sample(n, 4000, replace=F)
> rec.A <- silc.16[obs.A, c(X.vars, y.var)]
> rec.A$wwA <- rec.A$rb050/sum(rec.A$rb050)*N  # new weights
> don.B <- silc.16[-obs.A, c(X.vars, z.var)]
> don.B$wwB <- don.B$rb050/sum(don.B$rb050)*N  # new weights

> library(StatMatch) # loads StatMatch

# Nearest neighbour with Gower's distance
> group.v <- c("db040","rb090") # variables that identify donation classes
> X.mtc <- c("hsize","age","pb220a") # matching variables
> out.nnd <- NND.hotdeck(data.rec=rec.A, data.don=don.B, match.vars=X.mtc,
+                         don.class=group.v, dist.fun="Gower")
# creates the synthetic data set
> fill.A.nnd <- create.fused(data.rec=rec.A, data.don=don.B,
+                     mtc.ids=out.nnd$mtc.ids, z.vars="netIncome")

> head(fill.A.nnd, 2) # first 2 obs.
    hsize      db040 age rb090 pb220a    rb050 pl030       ww netIncome
401     5 Burgenland  45  male     AT 4.545916     1 10.85782  47159.21
71      2 Burgenland  65  male     AT 6.151409     5 14.69250  20561.23
```

By default `NND.hotdeck()` does not pose constraints on the “usage” of donors: a record in the donor data set can be selected many times as a donor. The multiple usage of a donor can be avoided by resorting to a *constrained hot deck* (`constrained=TRUE`) in which a donor can be used just once and all the donors are selected in order to minimize the overall matching distance. In practice, the donors are identified by solving a travelling salesperson problem; two alternative algorithms are available the classic one (`constr.alg="lpSolve"`) and the RELAX–IV algorithm (Bertsekas and Tseng, 1994) (`constr.alg="relax"`). This latter one is much faster but there are some restrictions on its licence. The constrained matching requires a higher computational effort but preserves better the marginal distribution of the variable imputed in the synthetic data set. Obviously the overall matching distance tends to be greater than the one in the unconstrained case.

8. The function `RANDwNND.hotdeck()` carries out the random selection of each donor from a suitable subset of all the available donors. This subset can be formed in different ways, e.g. by considering all the donors sharing the same characteristics of the recipient (gender, region, etc.) or simply the closest donors according to a particular rule. The traditional *random hot deck* (cf. Singh *et al.*, 1993) within imputation classes it is performed when no matching variables are specified (`match.vars=NULL`). The donor is picked up completely at random or with probability proportional to a weight (specified with the argument `weight.don`); in this latter case the *weighted random hot deck* is applied (cf. Andridge and Little, 2010).
`RANDwNND.hotdeck()` implements others alternative methods to restrict the set of the potential donors. These methods are based essentially on a distance measure computed on the matching variables (`match.vars`). In practice, when `cut.don="rot"` only the subset of the $\lfloor \sqrt{n_D} \rfloor + 1$ closest donors is considered (n_D is the number of available donors). With `cut.don="span"` a proportion `k` of the closest available donors ($[n_D \times k]$ is considered ($0 < k \leq 1$). By setting `cut.don="exact"` the `k` closest donors are retained ($1 \leq k \leq n_D$). When `cut.don="min"` only the donors at the minimum distance from the recipient are retained. Finally, when `cut.don="k.dist"` only the donors whose distance from the recipient is less or equal to `k` are considered.
In all the cases the selection of a donor within the subset of the closest donors can be with equal probability or with probability proportional to a weight (`weight.don`).
The following R code provides some examples of usage of `RANDwNND.hotdeck()`.

```
> # traditional random hot deck within classes
> group.v <- c("db040","rb090") # variables that identify donation classes
> rnd.1 <- RANDwNND.hotdeck(data.rec=rec.A, data.don=don.B, match.vars=NULL,
+                          don.class=group.v)
> # creates the synthetic data set
> fillA.rnd <- create.fused(data.rec=rec.A, data.don=don.B,
+                              mtc.ids=rnd.1$mtc.ids, z.vars="netIncome")

> # weighted random hot deck within classes
> rnd.2 <- RANDwNND.hotdeck(data.rec=rec.A, data.don=don.B, match.vars=NULL,
+                          don.class=group.v, weight.don="wwB")
> fillA.wrnd <- create.fused(data.rec=rec.A, data.don=don.B,
+                          mtc.ids=rnd.2$mtc.ids, z.vars="netIncome")

> # random choiches of a donor among the closest k=10
> X.mtc <- c("hsize","age","pb220a") # matching variables
> rnd.3 <- RANDwNND.hotdeck(data.rec=rec.A, data.don=don.B, match.vars=X.mtc,
+                     don.class=group.v, dist.fun="gower", cut.don="exact", k=10)
> fillA.knnd <- create.fused(data.rec=rec.A, data.don=don.B,
+                           mtc.ids=rnd.3$mtc.ids, z.vars="netIncome")
```

9. The function `rankNND.hotdeck()` implements the *rank hot deck distance* method introduced by Singh *et al.* (1993). It searches for the donor at a minimum distance from the given recipient record but, in this case, the distance is computed on the percentage points of the empirical cumulative distribution function of the unique (continuous) common variable X being considered. In estimating the empirical cumulative distribution it is possible to consider the weighs of the observations (arguments `weight.rec` and `weight.don`). This transformation of the origin values produces values uniformly distributed in the interval [0,1]; moreover, it can be useful when the values of X can not be directly

compared because of measurement errors which however do not affect the “position” of a unit in the whole distribution (cf. D’Orazio *et al.*, 2006a, pp. 199-200). This function permits to defining some donation classes. In this case the empirical cumulative distribution is estimated separately class by class. The following R code provides some examples of rank hot deck.

```
> # unweighted rank hot deck
> rnk.1 <- rankNND.hotdeck(data.rec=rec.A, data.don=don.B, var.rec="age",
+                          var.don="age", don.class="db040")
> fillA.rnk <- create.fused(data.rec=rec.A, data.don=don.B,
+                          mtc.ids=rnk.1$mtc.ids, z.vars="netIncome")

> # weighted rank hot deck
> rnk.2 <- rankNND.hotdeck(data.rec=rec.A, data.don=don.B, var.rec="age",
+         var.don="age", don.class="db040", weight.rec="wwA", weight.don="wwB")
> fillA.wrnk <- create.fused(data.rec=rec.A, data.don=don.B,
+                          mtc.ids=rnk.2$mtc.ids, z.vars="netIncome")
```

10. It is worth noting that all the functions in StatMatch that implement the hot deck techniques can be used to impute missing values in a data set. In this case it is necessary to separate the observations in two data sets: the file *A* will contain the units with missing values while the file *B* will contain the available donors.

C. Mixed methods

11. The mixed methods consist of two steps: (1) a model is fitted and all its parameters are estimated, then (2) a nonparametric approach is used to create the synthetic data set. The model is more parsimonious while nonparametric approach offers “protection” against model misspecification. The proposed mixed approaches for SM are based essentially on *predictive mean matching* imputation methods (cf. Section 2.5 and 3.6 in D’Orazio *et al.*, 2006a). The function `mixed.mtc()` in StatMatch implements two similar mixed methods that deals with continuous variables ($\mathbf{X}_M$, Y, Z) whose joint distribution is the multivariate normal. The main difference consists in the estimation of the parameters of the two regressions Y vs. $\mathbf{X}_M$ and Z vs. $\mathbf{X}_M$. By default the parameters are estimated thought maximum likelihood (argument `method="ML"`); in alternative it is available a method proposed by Moriarity and Scheuren (2001 and 2003) (`method="MS"`). D'Orazio *et al.* (2005) compared these methods in an extensive simulation study: in general ML tends to perform better, moreover it permits to avoid some incoherencies in the estimation of the parameters that can happen with the Moriarity and Scheuren approach.
After the estimation of the parameters of the two regression models, the “intermediate” values of Y in B ($\tilde{y}_b$) and of Z in A ($\tilde{z}_a$) are computed; these values are obtained by adding a random residual to the predicated value. Finally, in the step (2) each record in A is filled in with the value of Z observed on the donor found in B according to a constrained distance hot deck; the Mahalanobis distance is computed by considering the intermediate and live values: $(y_a, \tilde{z}_a)$ in A and $(\tilde{y}_b, z_b)$ in B.
In the following example the iris data set is used to show how `mixed.mtc()` works.

```
> # uses iris data
> iris.A <- iris[101:150, 1:3]     # recipient
> iris.B <- iris[1:100, c(1:2,4)]  # donor
> X.mtc <- c("Sepal.Length","Sepal.Width") # matching variables

> # parameters estimated using ML under the CI assumption
> mix.1 <- mixed.mtc(data.rec=iris.A, data.don=iris.B, match.vars=X.mtc,
+                      y.rec="Petal.Length", z.don="Petal.Width", method="ML",
+                      rho.yz=0, micro=TRUE, constr.alg="lpSolve")
> fillA.MLmix <- create.fused(data.rec=iris.A, data.don=iris.B,
+                           mtc.ids=mix.1$mtc.ids, z.vars="Petal.Width")
> # parameters estimated usign Moriarity & Scheuren method under the CI
> mix.2 <- mixed.mtc(data.rec=iris.A, data.don=iris.B, match.vars=X.mtc,
+                      y.rec="Petal.Length", z.don="Petal.Width", method="MS",
+                      rho.yz=0, micro=TRUE, constr.alg="lpSolve")
input value for rho.yz is 0
low(rho.yz)= -0.7404069
```

```
up(rho.yz)= 0.8621375
The input value for rho.yz is admissible
> fillA.MSmix <- create.fused(data.rec=iris.A, data.don=iris.B,
+                       mtc.ids=mix.2$mtc.ids, z.vars="Petal.Width")
```

12. The function `mixed.mtc()` by default performs mixed SM under the CI assumption ($\rho_{YZ|X} = 0$; argument `rho.yz=0`). When some additional auxiliary information about the correlation between Y and Z it is available (estimates from previous surveys or form external sources) then it can be exploited in SM by specifying a guess for $\rho_{YZ|X} = 0$ when using the ML estimation or for $\rho_{YZ} = 0$ when estimating the parameters by using the Moriarity and Scheuren method. The following R code provides some examples.

```
> # parameters estimated using ML and rho_YZ|X=0.85
> X.mtc <- c("Sepal.Length","Sepal.Width") # matching variables
> mix.3 <- mixed.mtc(data.rec=iris.A, data.don=iris.B, match.vars=X.mtc,
+                     y.rec="Petal.Length", z.don="Petal.Width", method="ML",
+                     rho.yz=0.85, micro=TRUE, constr.alg="lpSolve")
> fillA.MLmix1 <- create.fused(data.rec=iris.A, data.don=iris.B,
+                        mtc.ids=mix.3$mtc.ids, z.vars="Petal.Width")

> # parameters estimated using MS and rho_YZ=0.75
> mix.4 <- mmixed.mtc(data.rec=iris.A, data.don=iris.B, match.vars=X.mtc,
+                     y.rec="Petal.Length", z.don="Petal.Width", method="MS",
+                     rho.yz=0.75, micro=TRUE, constr.alg="lpSolve")
input value for rho.yz is 0.75
low(rho.yz)= -0.7404069
up(rho.yz)= 0.8621375
The input value for rho.yz is admissible
> fillA.MSmix1 <- create.fused(data.rec=iris.A, data.don=iris.B,
+                   mtc.ids=mix.4$mtc.ids, z.vars="Petal.Width")
```

D. Statistical matching with data from complex sample surveys

13. In the first step of the mixed methods the parameters of the regression models are estimated by assuming that the observed values in *A* and *B* are i.i.d. Unfortunately, when dealing with samples selected from a finite population by means of complex sampling designs (with stratification, clustering, etc.) it is difficult to maintain the i.i.d. assumption (it would mean that the sampling design can be ignored); in most of the cases the sampling design and the weights assigned to the units (usually design weights corrected for unit nonresponse, frame errors, etc.) can not be ignored when making inference.
Some SM nonparametric micro methods (`RANDwNND.hotdeck` and `rankNND.hotdeck`) allow the usage of the weights when searching for the donors. In general, with micro SM methods, the synthetic data set (i.e. *A* filled in with the values of *Z*) is the base of inference; when *A* is the result of a complex sample survey carried out on a finite population, the common practice consists in considering the sampling design and the weights attached to the units of *A* to make inference from the synthetic file too.

14. In literature there are few SM methods that explicitly take into account the sampling design and the corresponding sampling weights: Renssen's *calibrations based approach* (Renssen, 1998); Rubin's *file concatenation* (Rubin, 1986) and Wu's approach based on *empirical likelihood* methods (2004). These approaches have been compared in a simulation study by D'Orazio *et al.* (2010). Among them only the first one has been implemented in StatMatch by developing two functions: `harmonize.x()` and `comb.samples()`.

15. The Renssen's approach consists in a series of calibration steps of the survey weights of *A* and *B* in order to achieve consistency between estimates (mainly totals) computed separately from them. Calibration is a technique for deriving new survey weights, as close as possible to the starting ones, which fulfil a series of constraints (usually concerning totals). The first step in the Renssen's procedure consists in calibrating weights in *A* and weights in *B* such that the new weights when applied to the set of the matching variables, $\mathbf{X}_M$, allow to reproduce some known population totals. The calibrated weights

can be used to derive estimates from *A* and/or *B*. For instance, in case of categorical variables, the joint distribution $P(Y,Z)$ under the CI assumption is estimated by:

$$\hat{P}^{(CIA)}(Y,Z) = \hat{P}^{(A)}(Y|\mathbf{X}_M) \times \hat{P}^{(B)}(Z|\mathbf{X}_M) \times \hat{P}(\mathbf{X}_M)$$

where $\hat{P}(\mathbf{X}_M)$ can be derived indifferently from *A* or *B*.

In StatMatch this harmonization step can be performed by using `harmonize.x()`. This function performs weights calibration (or poststratification) by resorting to some functions made available by the R package "survey" (Lumley, 2010). The following example shows how to harmonize the joint distribution of the matching variables.

```
> # preliminary data manipulations
> # categorization of age
> rec.A$c.age <- cut(rec.A$age, breaks=c(16,24,49,64,100), include.lowest=T)
> don.B$c.age <- cut(don.B$age, breaks=c(16,24,49,64,100), include.lowest=T)
> # recode person economic status
> rec.A$c.pl030 <- cut(as.integer(rec.A$pl030), breaks=c(1,2,7),
+                         include.lowest=T, labels=c("work","don't work"))
> # categorize person net Income
> don.B$c.netI <- cut(don.B$netIncome/1000,
+                  breaks=c(-6,0,5, 10, 15, 20, 25, 30, 40, 50, 200))
>
> library(survey) # loads survey
> # creates svydesign objects
> svy.rec.A <- svydesign(~1, weights=~wwA, data=rec.A)
> svy.don.B <- svydesign(~1, weights=~wwB, data=don.B)
>
> # harmonizes wrt to joint distr. of gender vs. c.age
> # note: pop. totals are unknown
> out.hz <- harmonize.x(svy.A=svy.rec.A, svy.B=svy.don.B,
+                 form.x=~c.age:rb090-1)

> summary(out.hz$weights.A) # summaries of new calibrated weights for A
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  8.647  14.390  16.570  16.950  19.030  31.470
> summary(out.hz$weights.B) # summaries of new calibrated weights for B
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.279  10.540  11.840  12.280  13.910  22.400
```

16. The Renssen's approach permits to exploit some auxiliary information represented by third data source *C*, containing all the variables ($\mathbf{X}_M$,*Y*,*Z*) or just *Y* and *Z*. Two alternative methods to estimate the contingency table $Y \times Z$ are available: a) *incomplete two way stratification*; and b) *synthetic two way stratification*. Both the methods estimate $Y \times Z$ from *C* after some further calibration steps (for details see Renssen 1998). Both the methods are implemented in the function `comb.samples()` of the package StatMatch. When *C* is not available `comb.samples()` provides an estimate of $Y \times Z$ under the CI assumption, as shown in the following example.

```
> # estimating c.pl030 vs. c.netI under the CI assumption
> out <- comb.samples(svy.A=out.hz$cal.A, svy.B=out.hz$cal.B,
+             svy.C=NULL, y.lab="c.pl030", z.lab="c.netI",
+             form.x=~c.age:rb090-1)
>
> addmargins(t(out$yz.CIA)) # transposed table estimated under the CI
         c.pl030work c.pl030don't work        Sum
c.netI1    4203.9273         3929.4698  8133.3971
c.netI2    3212.7539         2941.5722  6154.3261
c.netI3    4436.4472         5108.0075  9544.4547
c.netI4    5648.5383         6199.2373 11847.7756
c.netI5    7129.6193         5716.1572 12845.7765
c.netI6    5391.3879         3802.7509  9194.1388
c.netI7    2877.6585         1696.1470  4573.8055
c.netI8    2249.5066         1256.9719  3506.4786
c.netI9     555.7829          345.2169   900.9998
c.netI10    688.8992          412.9481  1101.8473
Sum       36394.5210        31408.4790 67803.0000
```

E. Exploring uncertainty due to the statistical matching framework

17. A different approach to SM consists in exploring the uncertainty on the model chosen for ($\mathbf{X}_M$,*Y*,*Z*) due to the lack of knowledge typical of the basic SM framework (no auxiliary information is available). This approach is of help when the objective of SM is macro, but it does not produce a unique estimate of the unknown parameter characterizing the joint p.d.f. for ($\mathbf{X}_M$,*Y*,*Z*), rather it permits to identify an interval of plausible values for it. In particular, the function `Frechet.bounds.cat()` available in StatMatch permits to derive the uncertainty bounds for the probabilities in the contingency table $Y \times Z$, starting from the marginal tables $\mathbf{X}_M \times Y$, $\mathbf{X}_M \times Z$ and the joint distribution of the $\mathbf{X}_M$ variables (only categorical variables are handled). The bounds are derived by considering the following formulas:

$$P^{(low)}(y=j,z=k)=\sum_i P(x=i)\ \max\left(0;\ P(y=j|x=i)+P(z=k|x=i)-1\right)$$

$$P^{(up)}(y=j,z=k)=\sum_i P(x=i)\ \min\left(P(y=j|x=i);\ P(z=k|x=i)\right)$$

The table $P(\mathbf{X}_M ,Y)$ is estimated from *A*; $P(\mathbf{X}_M ,Z)$ is estimated from *B* while $P(\mathbf{X}_M)$ can be estimated indifferently on *A* or on *B*. This procedure implicitly assume that the joint distribution $P(\mathbf{X}_M)$ is the same on *A* and *B* (from the practical viewpoint, before computing the uncertainty bounds it would be preferable to harmonize $P(\mathbf{X}_M)$ in *A* and *B* it by using `harmonize.x()` function). It is worth mentioning that `Frechet.bounds.cat()` permits to estimate the bounds of the cells in $Y \times Z$ when no matching variables are considered and it provides also an estimate of $Y \times Z$ under the CI assumption. The following example shows how it works (the data with an harmonized distribution of $\mathbf{X}_M$).

```
> # estimate the needed contingency tables
> xx <- xtabs(out.hz$weights.A~db040+c.age+rb090+pb220a, data=rec.A)
> xy <- xtabs(out.hz$weights.A~db040+c.age+rb090+pb220a+c.pl030, data=rec.A)
> xz <- xtabs(out.hz$weights.B~db040+c.age+rb090+pb220a+c.netI, data=don.B)

> # estimates of the uncertainty bounds for Y vs. Z
> out.fb <- Frechet.bounds.cat(tab.x=xx, tab.xy=xy, tab.xz=xz,
print.f="data.frame")
> out.fb
      c.pl030   c.netI low.u       low.cx         CIA      up.cx       up.u
1        work   (-6,0]     0 0.0045584565 0.059694829 0.10073921 0.11995630
2  don't work   (-6,0]     0 0.0192630514 0.060307428 0.11544380 0.11995630
...
19       work (50,200]     0 0.0000000000 0.010004560 0.01454030 0.01625072
20 don't work (50,200]     0 0.0009444021 0.005480145 0.01548471 0.01625072
```

III. Open issues and further development of StatMatch

A. The choice of the matching variables

18. In the statistical matching applications based on the CI assumption the available data sources *A* and *B* may share a very high number of variables in common (**X**). In such cases it necessary to discard some of them and use only the most relevant ones, $\mathbf{X}_M$ ($\mathbf{X}_M \subseteq \mathbf{X}$), in explaining both *Y* and *Z*. In fact, even when using nonparametric hot deck methods, the usage of too many matching variables ($\mathbf{X}_M$) can affect negatively the matching procedure due to the *matching noise* (cf. Marella *et al*., 2008). The problem of choosing $\mathbf{X}_M$ still exists when fitting the regression models in the mixed SM procedures or when using the calibration of weights with the Renssen's procedures. In all the cases, it is necessary to resort to further analyses to identify $\mathbf{X}_M$. This is not a simple task because in the basic SM framework the variables **X**, *Y* and *Z* are not jointly observed; the relationship between **X** and *Y* can be investigated in *A* while the relationship between *Z* and **X** can be investigated in file *B*. Then the results of the two separate analyses have to be "combined" (two extreme cases: $\mathbf{X}_M = \mathbf{X}_M^{(A)} \cup \mathbf{X}_M^{(B)}$ or $\mathbf{X}_M = \mathbf{X}_M^{(A)} \cap \mathbf{X}_M^{(B)}$). Clearly, this is not the optimal procedure as argued by Cohen (1991).
Reasoning in terms of uncertainty offers the possibility of solving the problem in a better way by searching for the subset $\mathbf{X}_M$ that provide a noticeable reduction of the whole uncertainty while keeping

low the number of matching variables. In the case of categorical variables the function `Fbwidths.by.x()` is very helpful in fulfilling this task because it computes the bounds for cell probabilities in the contingency table $Y \times Z$ by considering all the possible subsets of the $\mathbf{X}$ variables that are provided in input. At the moment, the reduction of the uncertainty is measured in terms of the average of the widths of the bounds:

$$\overline{w} = \frac{1}{J \times K} \sum_{j=1}^{J} \sum_{k=1}^{K} \left[p^{(up)}(y = j, z = k) - p^{(low)}(y = j, z = k) \right]$$

For instance, with the artificial data resembling EUSILC survey it comes out:

```
> out.fbw <-  Fbwidths.by.x(tab.x=xx, tab.xy=xy, tab.xz=xz)
> out.fbw$av.widths # average widhts of uncertainty bounds
                          n.vars   av.width
|db040                         1 0.10000000
|rb090                         1 0.10000000
|pb220a                        1 0.10000000
|c.age                         1 0.08327851
|rb090+pb220a                  2 0.10000000
|db040+rb090                   2 0.10000000
|db040+pb220a                  2 0.09965135
|c.age+pb220a                  2 0.08318128
|db040+c.age                   2 0.08253795
|c.age+rb090                   2 0.07676515
|db040+rb090+pb220a            3 0.09873921
|db040+c.age+pb220a            3 0.08040068
|c.age+rb090+pb220a            3 0.07578177
|db040+c.age+rb090             3 0.07525343
|db040+c.age+rb090+pb220a      4 0.07118402
```

B. Computational efficiency

19. All the functions made available in StatMatch are based uniquely on R code without calls to other external compiled C or Fortran codes. This choice is not optimal from the computational viewpoint but offers the advantage of the full portability of the package among the various operating systems (including 64 bit versions of MS Windows).
As far as computational efficiency is concerned, the following Table reports the results of some experiments for the hot deck methods carried out with artificial data (simulated using the package simPoplation); in particular the data set *A* contains 14,000 observations while about 54,000 potential donors are available in *B*.

Table 2 – Computational efficiency of the R functions implementing hot deck imputation techniques.

Hot deck methods	StatMatch Function	No. of matching variables	No. of donation classes	Processing time (seconds)	Notes
UNconstrained NND	`NND.hotdeck()`	4	36	1282	`dist.fun="Gower"`
Constrained NND	`NND.hotdeck()`	4	36	1446	`dist.fun="Gower", constr.alg="relax"`
Random hot deck	`RANDwNND.hotdeck()`	4	36	1936	`dist.fun="Gower", cut.don="exact", k=10`

Note: PC with CPU Pentium IV 3GHz, 3GB RAM, MS Windows XP Professional (SP 3; 32bit)

C. Further development

21. The further development of StatMatch will follow three lines: (1) improve the functions for exploring uncertainty, (2) provide new functions for applying mixed methods to more general situations and, (c) extend the documentation about the package.
As far as uncertainty is concerned, it is planned to improve `Fbwidths.by.x()` by introducing more accurate criteria to identify which is the "better" $\mathbf{X}_M$ in reducing the uncertainty and, at the same time, keeping the number of matching variables as small as possible. Moreover, it will be evaluated whether it is possible to handle categorical and continuous variables.

The mixed SM methods seem promising. Here again the idea it that of improving `mixed.mtc()` in order to accept mixed type predictors. Another issue under investigation is the possibility of using nonparametric regression in the first step, leading to a completely nonparametric two step SM procedure. Finally, as far as documentation is concerned, it is planned to release a package "vignette" that explains in detail how to use the StatMatch to perform SM or missing data imputation. This vignette is planned to be released by July 2011.

References

Alfons A., Kraft S. (2010). simPopulation: Simulation of synthetic populations for surveys based on sample data. R package version 0.2.1. http://CRAN.R-project.org/package=simPopulation.

Andridge R.R., Little R.J.A. (2010) "A Review of Hot Deck Imputation for Survey Nonresponse". *International Statistical Review*, **78**, 40–64.

Bertsekas D.P., Tseng P. (1994). "RELAX–IV: A Faster Version of the RELAX Code for Solving Minimum Cost Flow Problems". *Technical Report*, LIDS-P-2276, Massachusetts Institute of Technology, Cambridge. http://web.mit.edu/dimitrib/www/RELAX4_doc.pdf

Cohen M. L. (1991) "Statistical matching and microsimulation models", in Citro and Hanushek (eds) *Improving Information for Social Policy Decisions: The Uses of Microsimulation Modeling. Vol II Technical papers*. Washington D.C.

D'Orazio M. (2011). StatMatch: Statistical Matching. R package version 1.0.1. http://CRAN.R-project.org/package=StatMatch

D'Orazio M., Di Zio M., Scanu, M. (2005) "A comparison among different estimators of regression parameters on statistically matched files trough an extensive simulation study". *Technical Report Contributi 2005/10*, Istat, Roma.

D'Orazio M., Di Zio M., Scanu M. (2006a) *Statistical Matching: Theory and Practice*. Wiley, New York.

D'Orazio M., Di Zio M., Scanu M. (2006b) "Statistical Matching for Categorical Data: Displaying Uncertainty and Using Logical Constraints", *Journal of Official Statistics*, **22**, 137-157.

D'Orazio M., Di Zio M., Scanu M. (2009) "The statistical matching workflow", in: *State of the art on statistical methodologies for integration of surveys and administrative data*, Report of WP1 of the "ESSnet Statistical Methodology Project on Integration of Survey and Administrative Data"

D'Orazio M., Di Zio M., Scanu M. (2010) "Old and new approaches in statistical matching when samples are drawn with complex survey designs". *Proceedings of the 45th "Riunione Scientifica della Società Italiana di Statistica"*, Padova 16-18 June 2010.

Gower J. C. (1971) "A general coefficient of similarity and some of its properties". *Biometrics*, **27**, 623–637.

Kovar J.G., MacMillan J., Whitridge P. (1988) "Overview and strategy for the Generalized Edit and Imputation System". Statistics Canada, *Methodology Working Paper*, No. BSMD 88-007 E/F.

Lumley T. (2010) "survey: analysis of complex survey samples". R package version 3.23-2. http://CRAN.R-project.org/package=survey

Marella D., Conti P.L., Scanu M. (2008) "On the matching noise of some nonparametric imputation procedures", *Statistics and Probability Letters*

Meyer D., Buchta C. (2010) "proxy: Distance and Similarity Measures". R package version 0.4-6. http://CRAN.R-project.org/package=proxy

Moriarity C., Scheuren F. (2001) "Statistical matching: a paradigm for assessing the uncertainty in the procedure". *Journal of Official Statistics*, **17**, 407–422.

Moriarity C., Scheuren F. (2003). "A note on Rubin's statistical matching using file concatenation with adjusted weights and multiple imputation", *Jour. of Business and Economic Statistics*, **21**, 65–73.

R Development Core Team (2011) *R: A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria. http://www.R-project.org/

Renssen R.H. (1998) "Use of statistical matching techniques in calibration estimation". *Survey Methodology*, **24**, 171-183

Rubin D.B. (1986) "Statistical matching using file concatenation with adjusted weights and multiple imputations". *Journal of Business and Economic Statistics*, **4**, 87-94

Singh A.C., Mantel H., Kinack M., Rowe G. (1993). "Statistical matching: use of auxiliary information as an alternative to the conditional independence assumption". *Survey Methodology*, **19**, 59–79.

Wu, C. (2004) "Combining information from multiple surveys through the empirical likelihood method". *Canadian Journal of Statistics*, **32**, 1-12