SANAMETRIX

Westat®

SDCNway USER'S GUIDE

JULY 8, 2020

Sanametrix
1530 Wilson Blvd., Suite 670
Arlington, VA 22209

Westat
An Employee-Owned Research Corporation®
1600 Research Boulevard
Rockville, Maryland 20850-3129

TABLE OF CONTENTS

1 Overview of Method and Functions

A risk assessment tool, an R program called *SDCNway*, is supplied through the Data Protection Toolkit to help agencies quantify the reidentification risk. The Data Protection Toolkit is created in support of the Federal Data Strategy Initiative, (https://strategy.data.gov/action-plan/), the Federal Committee on Statistical Methodology, the Interagency Council on Statistical Policy, and the Department of Education. The Data Protection Toolkit is a website containing existing resources used by Federal government agencies for data protection.

Unless very clearly defined, risk metrics, such as in *SDCNway*, do not measure absolute risk, however, they can be very useful in determining the relative risk in testing various scenarios (e.g., adding variables to the file, recodes). The *SDCNway* program offers two main functions that are explained below: one for individual risk, and one for file risk. The individual risk is addressed through exhaustive tabulations with a useful set of reports that inform data coarsening, variable suppression and random perturbation. There are three sets of metrics to measure the file level re-identification risk, based on log-linear modeling by Skinner and Shlomo (2008), Mu-Argus, and those proposed in El Emam (2011). Given a set of key variables whose cross-tabulation forms *K* cells, all three approaches can be used to derive the file-level re-identification risk by aggregating the individual-level risk estimated from models. We briefly describe each approach, and then discuss practical issues when employing these procedures.

The *SDCNway* package consists of two functions: `sdc_extabs` and `sdc_loglinear`.

- `sdc_extabs` mainly uses an exhaustive tabulation method to quantify the disclosure risk. It tabulates cell counts for different combinations of variables provided by the user. Using these counts, this function identifies variable categories and records which are considered high risk for disclosure.
- `sdc_loglinear` uses the log-linear model approach (Skinner and Shlomo, 2008) to estimate file-level re-identification risk.

1.1 sdc_extabs function

The basic idea of `sdc_extabs` is to identify the attributes or combinations of attributes that make a record different from the others in survey data. This is achieved through the exhaustive tabulation method. The unique or rare cases are associated with high disclosure risk. If a sample case can be uniquely identified by a small number of less detailed attributes, it is even more risky because it is

highly likely to be a population unique. Disclosure risk may arise if an intruder intends to identify individuals and disclose their identities or attributes through the matching of known information to external sources.

For forming the tabulations, it is recommended to use factual identifiers such as demographic and geographical variables. Variables with subjective responses, such as cognitive items, are not visible or identifiable by data intruders and would typically be excluded from the extensive multi-way analysis.

Within `sdc_extabs`, a limited recoding utility is offered, if requested by the user. The user can recode some non-missing values of a variable to be missing so that these values will not be involved in the subsequent risk analysis. That is, the values will not be included in any combination of attributes to determine sample uniqueness or rareness. After initial recodes are implemented, the next phase of the risk analysis is to process all possible tables of certain dimensions for a specified number of variables. Violations are flagged when table cell counts (and/or weighted cell counts) are less than a given threshold rule – three for example (i.e., Rule of 3). For each category of each variable, the proportion of table cells with violations is computed among all cells in which a variable is involved. The algorithm counts the number of violations in which a record is involved for the set of tables generated.

This `sdc_extabs` function provides a summary that informs the decisions about the need for data perturbation, further recoding or suppression. Recoding (or collapsing) non-missing values to other non-missing values needs to be done outside of `sdc_extabs`. Besides the method described above, two other sets of risk measures are also calculated within `sdc_extabs`, including the Mu-Argus (Polettini 2003) score and the risk measures proposed by El Emam (2011). The details can be found in Section 4.

1.2 sdc_loglinear function

The `sdc_loglinear` function estimates the file-level re-identification risk using the log-linear model approach. This approach assumes the intruder knows everything in a set of factual identifiers about a person. Cells are formed by cross-tabulating the entire set of factual identifiers. Sample uniques in the cells are determined. Population counts in the cells are estimated by fitting log-linear models using the iterative proportional fitting method. Based on the fitted population counts, the re-identification risk for the sample uniques are calculated and summarized across all sample uniques in the file.

The `sdc_loglinear` function does not offer a recoding utility except for blank values. The input data should not contain any missing values among factual identifiers and the weight variable. Otherwise, the function will give an error message.

This function provides a summary of the file-level re-identification risk that informs the decisions about the need for data perturbation, further recoding or suppression to reduce risk.

Sections 2 and 3 describe the parameters and syntax for `sdc_extabs` and `sdc_loglinear`, respectively. Section 4 presents a technical description of the algorithm of the two functions. Section 5 describes the test dataset that accompanies the package and Section 6 includes examples using these data.

2 sdc_extabs Arguments and Output

This section discusses the usage of `sdc_extabs`. Section 2.1 provides details of the arguments in `sdc_extabs` and Section 2.2 describes the output created by `sdc_extabs`.

The following shows the general usage of sdc_extabs.

```
sdc_extabs(data, ID = NULL, weight = NULL, varpool = names(data), forcelist =
character(0), forcenum = 1, missingdef = list(), mindim = 1, maxdim = 2,
threshold = NULL, wgtthreshold = NULL, condition = NULL, output_filename =
NULL, tau1 = 0.2, tau2 = 0.2, include_mu_argus = TRUE)

## S3 method for class 'sdc_extabs'
print(x, cutoff = 50, summary_outfile = NULL, ...)

## S3 method for class 'sdc_extabs'
plot(x, plotpath = NULL, plotvar1 = character(0),plotvar2 = character(0),
...)
```

2.1 Arguments

The `sdc_extabs` function uses the following arguments.

2.1.1 Input Controllers

`data` = *Data frame* containing the data for which we are to measure disclosure risk. Unexpected behavior may result if any column name begins with a period. This parameter is required.

`ID` = *Name* of column which identifies record. If NULL (default), an ID column named ROW_NUMBER is created and used in reports.

`weight` = Column *name* for the weight variable to be used to calculate the weighted counts. NULL or empty if none. This parameter is only required if `wgtthreshold` is specified. Note that if a weight variable is not provided, the number of statistics and plots that are produced is significantly reduced.

2.1.2 Algorithm Controllers

`varpool` = Vector of column *names* over which to form tables. This parameter is required.

`forcelist` = Vector of variable *names*. A specified number (see `forcenum` below) of the variables in `forcelist` are included in all tabulations. That is, all tabulations will have a number of variables from `forcelist` exactly equal to `forcenum`. This parameter is optional. The default value is 1. The variables in `forcelist` must also be included in `varpool`.

`forcenum` = *integer* specifies the number of variables in `forcelist` that are mandatory for all tabulations. The default is set to 1 if `forcelist` is not blank but `forcenum` is not specified.

`missingdef` = A named *list* specifying missing values. The names correspond to column names in data. It indicates values of `varpool` to be recoded as missing values. This parameter is optional. If `missingdef` is left blank, then no recoding will occur. The recoding is temporary and only lasts until the run is finished. In the output file all the variables will keep their original values as in the input file. Note if character and factor variables have blank values, blank needs to be specified in the `missingdef` in order to be treated as missing. On the other hand, NA values in data are treated as missing regardless of `missingdef`. If you do not want NA values to be treated as missing, please recode them before passing the data to this function.

`mindim` = *integer* specifies the minimum number of `varpool` variables (including `forcelist` variables) that can be used to form tables. `mindim` should not be smaller than `forcenum`. This parameter is optional. The default is set equal to 1.

`maxdim` = *integer* specifies the maximum number of `varpool` variables (including `forcelist` variables) that can be used to form tables. This parameter is optional. The default is set equal to 2.

`threshold` = *integer* specifies the threshold rule that is used to identify violation cells in terms of unweighted counts. If the number of records in a cell is less than `threshold`, the records in the cell are flagged as violations, and their violation counts increase by 1. This parameter is optional. If both `threshold` and `wgtthreshold` are NULL, `threshold` will be set to 3 and the weighted threshold will not be used. The threshold can be used alone or together with `wgtthreshold` (see below).

`wgtthreshold` = *constant* specifies the threshold rule that is used to identify violation cells in terms of weighted counts. If the sum of weights in a cell is less than `wgtthreshold`, the

records in the cell are flagged as violations, and their violation counts increase by 1. This parameter is optional. `wgtthreshold` can be used alone or together with `threshold`.

`condition` = *"and"* (or *"AND"*) or *"or"* (or *"OR"*) specifies the condition to connect the unweighted threshold rule (`threshold`) and the weighted threshold rule (`wgtthreshold`) when both threshold rules are applied. If both `threshold` and `wgtthreshold` are specified, condition should not be blank. This parameter is ignored if weight is NULL.

`tau1` = *value* specifies the threshold to compute the risk measure pRa. The default value is 0.2. This parameter is ignored if weight is NULL.

`tau2` = *value* specifies the threshold to compute the risk measure jRa. The default value is 0.2.

2.1.3 Output Controllers

`output_filename` = *name* of csv file to save the data set with violation counts and mu-argus scores attached. NULL if none. The file has the same content as the file specified in DATA parameter, except that it contains two additional variables mentioned earlier. If no path is specified, it will be put in the current working directory, but you can specify a full path if you prefer.

`include_mu_argus` = *TRUE* or *FALSE*. This is a flag indicating whether Mu-Argus and El-Emam statistics should be calculated. The default value is TRUE.

`cutoff` = *integer* specifies that the output report will display, for each table dimension, the top cutoff variable categories with the highest violation rates. The violation rate for a variable/category is computed as the percent of violation cells in all possible cells involving this variable/category at a given table dimension. This parameter is optional. The default is set equal to 50.

`summary_outfile` = *name* of permanent summary output file in .txt format. If not NULL, console output is copied to the file. Default is NULL (no logging of output). Errors and warnings are not diverted (consider running in batch mode if logging is needed).
`x` = an *object* of class `sdc_extabs`, as returned by the `sdc_extabs` function.

`plotpath` = *directory* to save plots. If the directory does not exist, the function will create it if the parent directory exists. Plots are saved as jpegs (quality = 100%). If `plotpath` is NULL (default), plots are not saved.

`plotvar1` = a vector of *names* of discrete variables for boxplots. Each of these variables is used as the variable for the x-axis, and Mu-Argus score and violation counts are used as variables for the y-axis. If none, boxplots are not produced.

`plotvar2` = a vector of *names* of continuous variables for scatterplots. Each of these variables is used as the variable for the x-axis, and Mu-Argus score and violation counts are used as variables for the y-axis. If none, scatterplots are not produced.

2.2 Output

The output of `sdc_extabs` contains the following sections.

- A printout shows crosstabs of the original variable and its recoded version for each and every one of the variables specified in `varpool`.
- A printout shows the total number of records in the data file, and statistics of violation counts such as minimum, maximum, and so on. A printout shows percentage of violations by table dimension, variable, and category of variable for each table dimension. The printout is sorted by ascending table dimension and descending percentage of violations. The percentage of violations for a category of a variable indicates the percentage of violation cells among all possible table cells which are of a certain dimension and involve this specific variable/category. This output summarizes the risk at the variable level, which helps the user to make decisions on confidentiality treatments such as suppression or recoding. Categories with 0 violations are not printed.
- A printout shows the percent of records with violations for each category of each `varpool` variable. It identifies the variable/category that contributes most to the disclosure risk and may be used to facilitate the decision on collapsing categories or suppressing variables.
- If `include_mu_argus` = TRUE, a printout shows the total and mean Mu-Argus scores by cell sizes of the tables created by cross-tabulating all `varpool` variables.
- If `include_mu_argus` = TRUE, a printout shows the risk measures proposed by El Emam (2011). See Section 4.1.5 for details.
- A printout shows top 10 records with most violations.

Additionally, boxplots and scatterplots can be produced if requested by the user. The plots show the distribution of the Mu-Argus risk score and violation counts, respectively, by the auxiliary variables specified by the user.

3 sdc_loglinear Arguments and Output

This section discusses the usage of `sdc_loglinear` function. Section 3.1 provides details of the arguments in `sdc_loglinear` and Section 3.2 describes the output created by `sdc_loglinear.`

The following shows the general usage of sdc_loglinear.

`sdc_loglinear(data, weight, varpool, degree = 2, numiter = 40,` epsilon = 0.001, blanks_as_missing = TRUE, output_filename = NULL)

```
## S3 method for class 'sdc_loglinear'
print(x, summary_outfile = NULL, ...)
```

```
## S3 method for class 'sdc_loglinear'
plot(x, plotpath = NULL, plotvar1 = character(0), plotvar2 =
character(0),n...)
```

3.1 Arguments

Each argument of the `sdc_loglinear` function is discussed in detail below.

3.1.1 Input Controllers

`data` = *data frame* containing the data for which we are to measure file-level disclosure risk. Unexpected behavior may result if any column name begins with a period. This parameter is required.

`weight` = column *name* for the weight variable. Note the data should not contain any missing values for the weight variable. This parameter is required.

3.1.2 Algorithm Controllers

`varpool` = vector of column *names* to be used in the model. Note the data should not contain any missing values among `varpool` variables. NA values in `data` are treated as missing. If you

do not want NA values to be treated as missing, please recode them before passing the data to this function. This parameter is required.

`degree` = *integer* for the degree of highest interaction terms to be used in the log-linear model. The default value is 2.

`numiter` = *integer* for themaximum number of iterations to run iterative proportional fitting for log-linear model. The default value is 40.

`blanks_as_missing` = *TRUE* or *FALSE*. If TRUE, character and factor variables that are blank or pure whitespace are treated as missing values. The default value is TRUE. Note the data should not contain any missing values among `varpool` variables. If any of the `varpool` variables have blank values, you need to either set `blanks_as_missing=FALSE`, or recode blanks to nonmissing values before running the `sdc_loglinear` function. Otherwise an error message will appear. If `blanks_as_missing=FALSE` then blank will be treated as a valid value.

`epsilon` = *value* for the maximum deviation allowed between observed and fitted margins. It should be a number much smaller than 1 to make sure the log-linear model is fitted well. The default is set equal to 0.001. The program will stop iteration once either the `epsilon` criterion or the `numiter` criterion is met.

3.1.3 Output Controllers

`output_filename` = *name* of the csv file to save the data set with record-level risk measures, .tau1_rec_cellmean, tau2_rec_cellmean, tau1_rec_overallmean, and .tau2_rec_overallmean, attached. The record-level risk measures are based on the model with the degree of interactions specified by the `degree` parameter. NULL if no output file is to be saved.

`x` = an *object* of class `sdc_loglinear`, as returned by the `sdc_loglinear` function. It is the input for the plot and print methods.

`summary_outfile` = *name* of permanent summary output file in .txt format. If not NULL, console output is copied to the file. Default is NULL (no logging of output). Errors and

warnings are not diverted (if logging of these is needed, consider running your program in batch mode).

`plotpath` = *directory* to save plots. If the directory does not exist, the function will create it if the parent directory exists. Plots are saved as jpegs (quality = 100%). If `plotpath` is NULL (default), plots are not saved.

`plotvar1` = a vector of *names* of discrete variables for boxplots. Each of these variables is used as the variable for the x-axis, and the record-level tau1and tau2 are used as variables for the y-axis. If none, boxplots are not produced.

`plotvar2` = a vector of *names* of continuous variables for scatterplots. Each of these variables is used as the variable for the x-axis, and the record-level tau1and tau2 are used as variables for the y-axis. If none, scatterplots are not produced.

3.2 Output

The output of `sdc_loglinear` contains the following sections.

- A printout shows the risk estimates and goodness of fit statistics using the inverse of overall average weight to estimate cell sampling rate. The printout has two rows. The first row shows the results for main-effect model, and the second row shows the results for the model containing all #-way interactions, where # is specified by the `degree` parameter. There are eleven columns in the printout as described below.
  - The first two columns show the number of records (sampsize) in the file and average number of records per cell (avg_cell_size), respectively;
  - The third column (interaction) shows the number/degree of interaction terms included in the log-linear model;
  - The fourth and fifth columns (tau1Risk and tau2Risk) show two types of risk estimates[1] averaged over all records in the file;
  - The next two columns (tau1 and tau2) show two types of overall risk estimates;

[1] See Section 4.2 for details about the risk estimates and goodness of fit statistics.

- The next two columns (B_tau1_type1 and B_tau1_type2) show two types of goodness of fit statistics for the type 1 risk (tau1); and
- The last two columns (B_tau2_type1 and B_tau2_type2) show two types of goodness of fit statistics for the type 2 risk (tau2).

- A printout similar to the bullet above except using the inverse of cell average weight to estimate cell sampling rate.
- Boxplots and scatterplots can be produced if requested by the user. The plots show the distribution of record-level risk measures by the auxiliary variables specified by the user.

4 Technical Description

This section provides a technical description for the SDCNway package.

- For the `sdc_extabs` function: definition of missing values (Section 4.1.1), formation of tabulations (Section 4.1.2), identification of violations (Section 4.1.3), interpretation of the risk measures (Section 4.1.4), and formulae for Mu-Argus and El-Emam risk metrics.
- For the `sdc_loglinear` function, descriptions of risk measures and goodness of fit statistics.

4.1 sdc_extabs

4.1.1 Definition of Missing Values

In the first phase of `sdc_extabs`, the user may recode some values of `varpool` variables from non-missing to missing through the option `missingdef`. `sdc_extabs` only examines the tables based on complete cases. In other words, cases with missing values in any of the variables in a table will be excluded. For example, if a record has variable A equal to missing, then this record will not be used in any tabulations involving variable A. However, this record will still be used in the tabulations that do not involve variable A if it contains no missing values in any of the table variables. Sometimes answers such as "Don't Know", "Refused", "Inapplicable", etc, are coded into special non-missing values such as 7, 8, 9, -1, 999, … in the survey data. If the user determines that these values do not carry useful information to the intruders and should not be involved in the risk analysis, he may choose to recode these values to be missing in the `sdc_extabs` function. These recodes are temporary and only last until the end of the run. The output file still contains the original variables and values.

4.1.2 Formation of Tabulations

The function exhaustively forms *m*-way tables using the variables in `varpool` or the temporarily recoded variables if `missingdef` is specified. The lower and upper bounds for the dimension parameter *m* are specified in `mindim` and `maxdim`. The tables contain complete cases only. In total, the function forms and scans *n* tables, where

$$n = \sum_{m=MINDIM}^{MAXDIM} \binom{p}{m} = \sum_{m=MINDIM}^{MAXDIM} \frac{p(p-1)\cdots(p-m+1)}{m(m-1)\cdots 1},$$

and p is the number of variables in `varpool`.

4.1.3 Identification of Violations

If the number of cases in a cell is less than `threshold` or the weighted cell count is less than `wgtthreshold`, this cell is flagged as a violation cell, and the variables/categories (e.g. sex = 2 and region = 3) used to define this cell are identified as "contributing to cell violations." Meanwhile, violation counts keep track of the number of times a case has been in any violation cells. In this case, the violation count for the cases in this cell increase by one.

4.1.4 Interpretation of the Risk Measures

Two risk measures are summarized in the output report: (1) violation counts; (2) percent of cell violations by variable/category. The first measure indicates the risk level of each data record. The output report shows some summary statistics, such as minimum, median, maximum, mean, and sum of the violation counts. The results can be helpful for applying data perturbation techniques such as data swapping.

The second measure indicates which variables/categories contribute to cell violations more than others. The percent of cell violations for a variable/category is computed as the number of violation cells involving this variable/category divided by the total number of cells formed by this variable/category. In the report, the percent of cell violations is displayed by table dimension. The cutoff variables/categories with the highest percentages are shown in descending order of percentages for each table dimension. The results can be useful for determining variable suppression or recoding.

4.1.5 Mu-Argus and Other Risk Measures

Mu-Argus is a measure of individual re-identification risk. Consider the contingency table built by cross-tabulating the key variables. A combination k is defined as the k-th cell in the contingency table. The set of combinations $\{1, \ldots, k, \ldots, K\}$ defines a partition of both the population and the sample into cells. Let f_k and F_k denote, respectively, the number of records in the sample and the number of units in the population in cell k; F_k is unknown for each k. The following assumptions are made to determine the probability mass function of $F_k|f_k$ (Bethlehem et al. 1990; Rinott 2003; Polettini 2003) based on a super-population approach:

$$F_k|p_k \sim Poisson(Np_k)$$
$$f_k|F_k, \pi_k, p_k \sim Binom(F_k, \pi_k)$$

The parameter p_k follows improper prior distribution, proportional to $1/p_k$, and Np_k is the mean of the Poisson distribution, where N is the population size. The parameter π_k is the probability that a case in the population cell k falls into the sample. Under this model, the posterior distribution of $F_k|f_k$ is negative binomial. The individual risk is therefore measured as the posterior mean of $1/F_k$ with respect to the distribution of $F_k|f_k$ by

$$r_i = E\left(\frac{1}{F_k}|f_k\right) = \frac{\hat{\pi}_k^{\,f_k}}{f_k}\,{}_2F_1(f_k, f_k : f_k + 1; 1 - \hat{\pi}_k)$$

where ${}_2F_1$ is the hypergeometric function, $\hat{\pi}_k$ is estimated by $f_k / \sum_{i \in k} w_i$, and w_i is the sample weight that takes into account differential sampling rates, nonresponse, and weight calibration adjustment. Polettini (2003) derived approximated risk measures to reduce the complexity of numerically evaluating the hypergeometric function. The approximations are as follows:

If the cell count is 1, or $f_k = 1$,

$$r_i = -log(\hat{\pi}_k)\frac{\hat{\pi}_k}{1 - \hat{\pi}_k}$$

If the cell count is 2, or $f_k = 2$,

$$r_i = \frac{\hat{\pi}_k}{(1 - \hat{\pi}_k)^2}[\hat{\pi}_k log(\hat{\pi}_k) + 1 - \hat{\pi}_k]$$

If the cell count is 3, or $f_k = 3$,

$$r_i = \frac{\hat{\pi}_k}{2(1 - \hat{\pi}_k)^3}\{(1 - \hat{\pi}_k)[3(1 - \hat{\pi}_k) - 2] - 2\hat{\pi}_k{}^2 log(\hat{\pi}_k)\}$$

If the cell count is greater than 3, or $f_k > 3$,

$$r_i = \frac{\hat{\pi}_k}{f_k}\left\{1 + \frac{1 - \hat{\pi}_k}{f_k + 1} + \frac{2(1 - \hat{\pi}_k)^2}{(f_k + 1)(f_k + 2)} + \cdots + \frac{7!\,(1 - \hat{\pi}_k)^7}{(f_k + 1)(f_k + 2)\ldots(f_k + 7)}\right\}$$

If $\hat{\pi}_k$ is approximately 1, then $r_i = 1/f_k$.

This method can be applied to the sampled data only. If the sampling weights are not available, or the data are from census, it is not meaningful to use this strategy to estimate the individual risk. In the case of simple random sampling or other equal selection probability designs, this estimation approach is also not useful unless the calibration procedures conducted by the statistical agencies introduce variation to the sampling weights.

The individual Mu-Argus risk measure can be summarized at the file level. For example, the sum of Mu-Argus score over the sample or a subset of sample (e.g., sample uniques, or $f_k = 1$) estimates the expected number of sampled cases that are re-identified in the population. The mean of Mu-Argus score over the sample or a subset of sample gives the average re-identification risk. The sum of Mu-Argus score divided by the sum of weights estimates the proportion of population that is re-identified.

Two sets of re-identification metrics were introduced in the Appendix of El Emam (2011). They were classified based on prosecutor risk and journalist risk. The former assumes that the intruder knows the target is in the disclosed dataset and the latter assumes the opposite. The metrics included in `sdc_extabs` are listed as follows:

- pR_a is the proportion of cases in the cells with cell count less than $\frac{1}{\tau_1}$, where τ_1 is a threshold value.

$$pR_a = \frac{1}{n}\sum_{k \in K} f_k \times I\left(\frac{1}{f_k} > \tau_1\right)$$

- pR_b is the inverse of the minimum cell count.

$$pR_b = \frac{1}{min(f_k)}$$

- pR_c is a ratio of the number of cells to the number of cases in the data.

$$pR_c = \frac{K}{n}$$

- jR_a is the proportion of cases in the cells with $\frac{1}{F_k}$ greater than τ_2.

$$jR_a = \frac{1}{n}\sum_{k \in K} f_k \times I\left(\frac{1}{F_k} > \tau_2\right)$$

- jR_b is the inverse of minimum F_k.

$$jR_b = \frac{1}{min(F_k)}$$

- jR_c is the average of re-identification risk $\frac{1}{F_k}$.

$$jR_c = \frac{1}{n}\sum_{k \in K} \frac{f_k}{F_k}$$

In the journalist risk scenario, F_k needs to be known or estimated. In `sdc_extabs`, $\frac{1}{F_k}$ is estimated by the Mu-Argus risk score. If jR_c >0.5, the Mu-Argus score may underestimate the re-identification risk. In this case, please consider running a log-linear model (the `sdc_loglinear` function) to compute the risk.

4.2 sdc_loglinear

Skinner and Shlomo (2008) estimated the following two measures for file-level re-identification risk using log-linear models.

(1) Expected number of sample uniques that are population uniques:

$$\tau_1^* = \sum_{SU} P(F_k = 1 | f_k = 1),$$ and

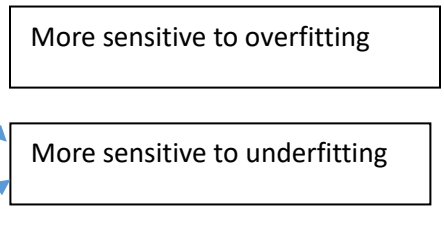
(2) Expected number of correct matches for sample uniques:

$$\tau_2^* = \sum_{SU} E(1/F_k | f_k = 1)$$,

where SU denotes sample unique cells, f_k is the sample frequency in cell k, and F_k is the population frequency in cell k.

Corresponding to the two measures above, the `sdc_loglinear` function computes tau1 (τ_1^*) and tau2 (τ_2^*) respectively. The function also computes the following goodness of fit statistics to help pick an appropriate model since over/underfitting of the models could lead to under/over-estimation of the re-identification risk.

- b_tau1_type1 ($\hat{B}_1/\sqrt{v_R}$ in Skinner and Shlomo (2008))
- b_tau1_type2 ($\hat{B}_1/\sqrt{v}$ in Skinner and Shlomo (2008))
- b_tau2_type1 ($\hat{B}_2/\sqrt{v_R}$ in Skinner and Shlomo (2008))
- b_tau2_type2 ($\hat{B}_2/\sqrt{v}$ in Skinner and Shlomo (2008))



If the statistics above are much greater than 0 (e.g., b_tau1_type1=4), then the risk estimate tau1 is conservative and overestimates the risk. The closeness to 0 of the statistics above is evidence of an absence of underfitting. Since the criteria were derived primarily to detect underfitting and numerical work showed they were more effective to detect underfitting than overfitting, a forward search algorithm was suggested to fit log-linear models. As a practical approach, we suggest first computing the main effects model and the all 2-way interaction model (which can be achieved using the parameter `degree`=2). If the latter model shows sign of underfitting, then include higher degree of interactions by specifying `degree`=3, 4, etc. All 2-way interactions are often sufficient.

The function also computes the following two file-level risk measures:

- Tau1Risk: the percent of sample uniques that are population uniques, and
- Tau2Risk: the percent of correct matches for sample uniques.

The risk measures and goodness of fit statistics are created using the following two approaches separately:

- estimating cell sampling rate by the overall sampling rate (i.e., $\hat{\pi}$); and
- estimating cell sampling rate using the cell sample size and cell sum of weights (i.e., $\hat{\pi}_k$)

The function prints two sets of the final summary (one for estimates using $\hat{\pi}$, another for estimates using $\hat{\pi}_k$) showing for each model, sampsize (number of records included in the model), avg_cell_size (average cell size), interaction (the degree of interaction terms included in the model if any), the values of Tau1Risk, Tau2Risk, tau1 (Expected number of sample uniques that are population unique), tau2 (Expected number of correct matches for sample uniques), b_tau1_type1, b_tau1_type2, b_tau2_type1, and b_tau2_type2. If there is no sample unique in a model, no risk measure will be computed and a message will be printed about it.

5 Test Data

A subset of the 1992 National Adult Literacy Study (NALS) prison study public-use microdata file is used in Section 6 as a test dataset. The values of the variance unit and variance stratum are not those created for that release. The dataset is called EXAMPLEDATA, and it has 20 variables and 182 records. The name, description and values, for each of the 20 variables are given below.

Table 5-1 Variable information, by description and possible values

Variables	Description	Possible values
BIB1201	Ever been in program to improve basic skills?	1 (Yes), 2 (No)
BIC0501	Ever been placed on probation?	1 (Yes), 2 (No)
BID0101	Do you have work assignments inside or outside?	1 (Yes), 2 (No)
BIE0601	How often write letters/memos in English?	1 (Never), 2 (Less than once a month), 3 (Less than once a week but at least once a month), 4 (At least once a week but not every day), 5 (Everyday)
BORNUS	Born in USA?	1 (Yes), 2 (No)
CASEID	Identification No.	ID from “90110104” to “93210309”
CENREG	Census region	1 (Northeast), 2 (Midwest), 3 (South), 4 (West)
DAGE	Age derived from date of birth	values ranging from 17 to 63
DAGE3	Derived age with three categories	1 (DAGE<30), 2 (30<=DAGE<50), 3 (DAGE>=50)
DIC0401	Derived years since admission	values ranging from 0.08 to 17.6
DRACE3	Derived Race/ethnicity with three categories	1 (Hispanic); 2 (NH Black); 3 (Other)

EDUC3	Recoded highest education level with three categories	1:less than high school, 2: high school , 3: >high school
EDUC_DET	Detailed highest level of education received	values ranging from 2 to 11
GENDER	Gender	1 (male), 2 (female)
RATE	Sampling rate	0.05; 0.02
RiskStra	Risk stratum	0-4
SCORE	Average literacy score	values ranging from 13 to 400
VARSTR	Variance stratum	values ranging from 1 to 91
VARUNIT	Variance unit	1 or 2
WEIGHT	Full sample weight	continuous with values ranging from 136 to 1788

SOURCE: U.S. Department of Education, National Center for Education Statistics, National Adult Literacy Study, 1992 Public-use File.

6 Examples

Two examples are presented here to illustrate how to use the two functions in the *SDCNway* package to analyze disclosure risks in microdata.

6.1 Example for sdc_extabs

The first example illustrates the basic features for `sdc_extabs` including specifying variables, table dimensions, and threshold rules involved in the analysis, recoding nonmissing values to missing, and displaying some of the standard output. The input dataset is the test data (EXAMPLEDATA), which is described in the previous Section 5. The R code and output from the example are provided in appendix A.

The example is set up to scan through all possible two-way (`mindim = 2`) and three-way (`maxdim = 3)` tables formed by a list of 10 variables (`varpool= BIB1201 BIC0501 BID0101 BIE0601 BORNUSA CENREG DAGE3 DRACE3 EDUC3 GENDER`). Table cells with less than three units (`threshold = 3`) or weighted count less than 3,000 (`wgtthreshold = 3000`) are identified as sparse cells or violation cells. Value 5 of the fourth variable in the `varpool`, BIE0601, is recoded to missing so that this value is not involved in the risk analysis. Or in other words, units with BIE0601 = 5 are excluded from the tables formed by BIE0601 and other variables; therefore, BIE0601 = 5 does not contribute to any violations of threshold rules.

In the output for `sdc_extabs`, 15 variables/categories (`cutoff = 15`) with the highest percentages of violations are displayed for each table dimension on pages A-5 and A-6. On page A-9, one box-plot shows Mu-Argus risk score by BORNUSA and another boxplot shows violation counts by BORNUSA. On pages A-10, one scatterplot shows Mu-Argus risk score versus sample weight and another scatterplot shows violation counts versus sample weight. In the scatterplot, the dots at the very bottom are the records which are in cells of large sizes and therefore associated with low disclosure risk. The plot shows a clear negative relationship between sample weight and disclosure risk for the records in small cells. Sample weight represents the number of cases in the population with similar characteristics to a sampled case. As a result, the larger the sample weight, the smaller the disclosure risk.

However, we note that pR_c >0.5 (too many cells per case), and therefore the Mu-Argus score may underestimate the re-identification risk. In this case, we run the log-linear model to compute the risk, as discussed below.

6.2 Example for sdc_loglinear

The second example illustrates the basic features for `sdc_loglinear`. The input dataset is the same as the input for `sdc_extabs`. In this example, we assess the disclosure risk if a data intruder knows all of the following six indirect identifiers: BORNUSA CENREG DAGE3 DRACE3 EDUC3 GENDER, which are listed in the `varpool` parameter. The R code and output for the example are provided in appendix B. The output prints two sets of summaries, which have similar layouts but use different approaches to estimate cell sampling rates, as indicated by the titles "RESULTS - Uses overall average weight" and "RESULTS - Uses cell average weights". Each summary shows for the main effect model (interaction=none) and all 2-way-interaction model (interaction=2-way) respectively, the sample size (sampsize), average cell size (avg_cell_size), percent of sample uniques that are also population uniques (Tau1risk), percent of correct matches for sample uniques (Tau2risk), number of samples uniques that are also population uniques (Tau1), number of correct matches for sample uniques (Tau2) and the four model diagnostic statistics (B_tau1_type1, B_tau1_type2, B_tau2_type1, B_tau2_type2). In the example, the risk estimates and model diagnostics in the two summaries are similar, indicating the robustness of the approach in this application. Also, the model diagnostics are approximately 0 for the main effects model indicating that the interaction terms are not needed. Although close to zero, the value of B_tau1_type1 is negative, which indicates slight overfitting. In this case it may be considered to drop a variable or two from the risk assessment and re-process. Tau1risk and Tau2risk are both close to 0 indicating very low disclosure risk.

On page B-4, two box-plots show the record-level tau1 risk and tau2 risk by BORNUSA, respectively. The two plots are for the approach that uses overall average weight to estimate cell sampling rate. As can be seen, the value of tau1 risk for all records clusters around 0 for both categories of BORNUSA, except that BORNUSA=2 has one outlier which is still very small (around 0.0012). For tau2 risk, the majority of the records have values close to 0. Although tau2 risk has more outliers, the outliers have small values with a maximum of about 0.05 for BORNUSA=1 and 0.15 for BORNUSA=2. Page B-5 has two similar box-plots but using the cell average weight to estimate cell sampling rate. The boxplots using the two approaches are very similar, re-affirming what was seen in the summary printout.

Pages B-6 and B-7 show scatterplots of record-level tau1 risk and tau2 risk versus sample weight using the two approaches to estimate cell sampling rate, respectively. Again the scatter plots are very similar between the two approaches. In the scatterplots, almost all of the dots are at the very bottom (around 0) with a few outliers which correspond to medium-size weight (around 500-600). All of the records that have large sample weights also have disclosure risk near 0.

Appendix A Example R code and Output for sdc_extabs

Example R code

```
library(SDCNway)

data(exampledata)

plotdir <- "../../Testing/output/sdcnway_example/plots"outfile <-
"../../Testing/output/sdcnway_example/output.csv"
logfile <- "../../Testing/output/sdcnway_example/logfile.txt"

vars <- c("BIB1201", "BIC0501", "BID0101", "BIE0601", "BORNUSA", "CENREG",
          "DAGE3", "DRACE3", "EDUC3", "GENDER")
results <- sdc_extabs(data,
                      "CASEID",
                      weight="WEIGHT",
                      varpool=vars,
                      mindim=2,
                      maxdim=3,
                      missingdef=list(BIE0601=5),
                      threshold=3,
                      wgtthreshold=3000,
                      condition="or",
                      output_filename=outfile,
                      tau1=0.5,
                      tau2=0.001)


print(results, cutoff=15, summary_outfile=logfile)
plot(results, plotpath=plotdir, plotvar1="BORNUSA", plotvar2="WEIGHT")
```

Example R Output

```
Cross-tabulation of original vs. recoded variables.

 BIB12O1 :

 Original Recoded Frequency

        1        1         29

        2        2        153


 BIC05O1 :

 Original Recoded Frequency

        1        1        108

        2        2         74


 BID01O1 :

 Original Recoded Frequency

        1        1        128

        2        2         54


 BIE06O1 :

 Original Recoded Frequency

        1        1         53

        2        2         66

        3        3         28

        4        4         20

        5     <NA>         15


 BORNUSA :

 Original Recoded Frequency

        1        1        162

        2        2         20


 CENREG :
```

```
 Original Recoded Frequency
        1       1        29
        2       2        44
        3       3        68
        4       4        41

 DAGE3 :
 Original Recoded Frequency
        1       1        87
        2       2        88
        3       3         7

 DRACE3 :
 Original Recoded Frequency
        1       1        32
        2       2        79
        3       3        71

 EDUC3 :
 Original Recoded Frequency
        1       1        87
        2       2        53
        3       3        42

 GENDER :
 Original Recoded Frequency
        1       1       168
        2       2        14

Number of records:  182

      Summary of violation counts
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sum
0.00	1.00	6.00	11.29	16.00	74.00	2055.00

Top 15 violations by variable and variable categories for each table dimension

Unweighted count < 3 or Weighted count < 3000

Number of variables involved in tables	Variable	Category of variable	Percent of Cells with violations based on threshold rules
2	DAGE3	3	77.27%
2	GENDER	2	47.83%
2	BORNUSA	2	37.50%
2	BIE0601	4	36.36%
2	BIB1201	1	24.00%
2	CENREG	1	21.74%
2	CENREG	2	21.74%
2	EDUC3	3	20.83%
2	BIE0601	3	17.39%
2	DRACE3	1	16.67%
2	CENREG	3	13.04%
2	CENREG	4	13.04%
2	BID0101	2	12.00%
2	BIE0601	1	8.70%
2	DRACE3	2	8.33%
3	DAGE3	3	92.70%
3	GENDER	2	84.32%
3	BORNUSA	2	76.33%
3	BIE0601	4	65.32%
3	BIB1201	1	64.38%
3	BIE0601	3	60.00%
3	DRACE3	1	58.72%
3	EDUC3	3	57.73%

3	CENREG	1	56.44%
3	CENREG	4	48.13%
3	BID0101	2	42.67%
3	CENREG	2	42.19%
3	BIE0601	1	38.07%
3	DRACE3	3	38.03%
3	EDUC3	2	37.56%

Percent records with violations by variable and category

Variable	Category of variable	Percent
BIB1201	1	100.0%
BIB1201	2	73.20%
BIC0501	1	74.07%
BIC0501	2	82.43%
BID0101	1	75.78%
BID0101	2	81.48%
BIE0601	1	79.25%
BIE0601	2	66.67%
BIE0601	3	100.0%
BIE0601	4	85.00%
BIE0601	5	66.67%
BORNUSA	1	74.69%
BORNUSA	2	100.0%
CENREG	1	100.0%
CENREG	2	81.82%
CENREG	3	55.88%
CENREG	4	92.68%
DAGE3	1	74.71%
DAGE3	2	78.41%
DAGE3	3	100.0%
DRACE3	1	100.0%
DRACE3	2	70.89%

DRACE3	3	74.65%
EDUC3	1	74.71%
EDUC3	2	73.58%
EDUC3	3	88.10%
GENDER	1	75.60%
GENDER	2	100.0%

MU-ARGUS summaries

Cell count	Number of cases	Total Argus score	Mean Argus score	Total Argus score/total number of observations	Total Argus score/overall sum of weights
=1	156	1.6187	0.0104	0.008894196	0.000013149
<=2	176	1.6492	0.0094	0.009061788	0.000013397
<=3	182	1.6529	0.0091	0.009081850	0.000013427

Re-identification Risk Metrics (El-Emam)

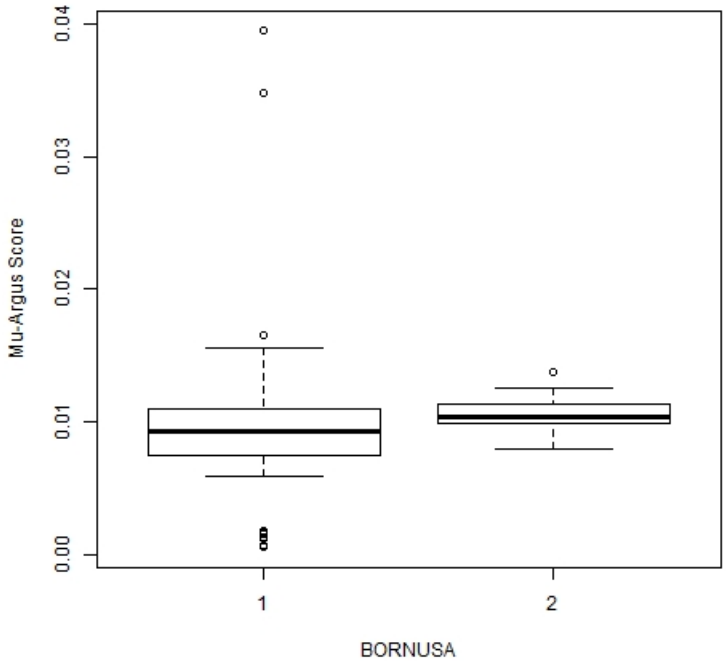
pRa	pRb	pRc	jRa	jRb	jRc
0.8571	1.000	0.9231	0.9670	0.03945	0.009082

Top 10 Records with Most Violations

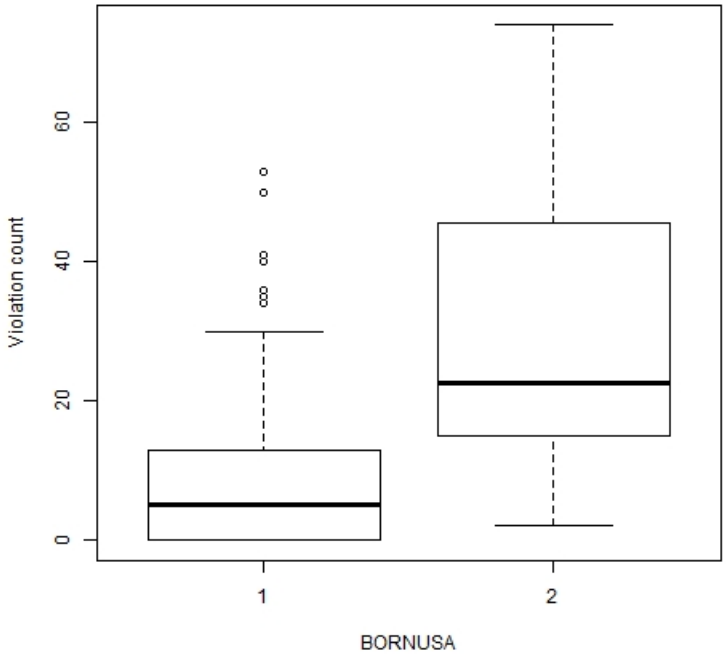
CASEID	BIB1201	BIC0501	BID0101	BIE0601	BORNUSA	CENREG	DAGE3	DRACE3	EDUC3
91510304	1	2	1	1	2	3	2	1	2
90520116	1	2	1	4	2	1	2	1	3
90320206	2	1	2	3	2	4	2	1	1

```
90320112        1       1       1       3       2       4       2       1       3

92820224        1       1       1       3       1       3       3       1       1

92210206        1       1       1       3       2       1       1       3       1

92410205        2       1       1       4       1       2       3       3       3

90410201        2       1       2       1       1       4       3       3       2

90410304        1       2       1       2       1       4       2       3       1

91110105        1       1       1       1       2       2       2       2       3
```

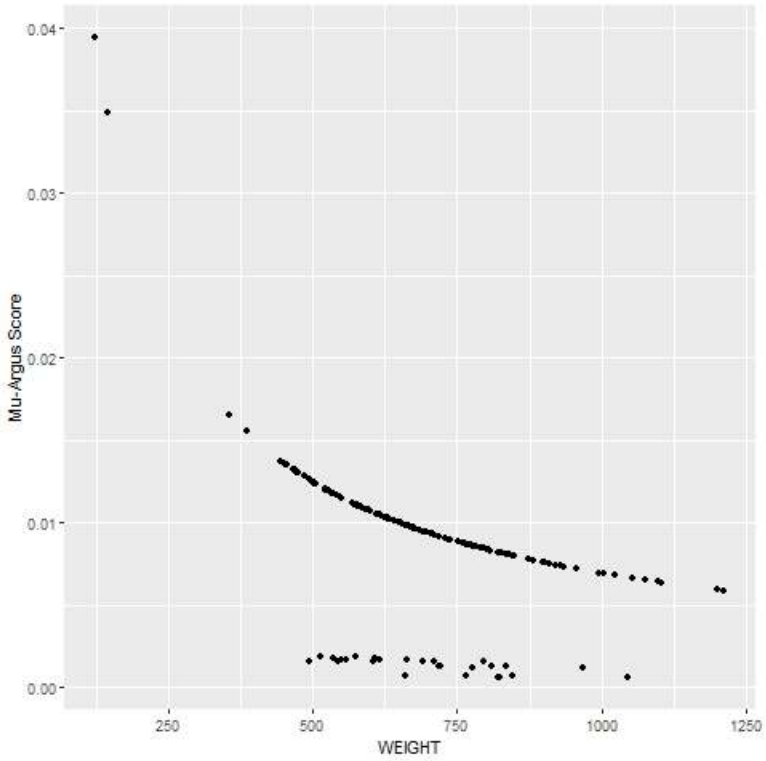
GENDER	WEIGHT	Mu-Argus Score	Violation Count
2	528.7428	0.011881739	74
1	444.4994	0.013747366	66
2	546.5149	0.011555251	64
1	540.9243	0.011655855	55
1	629.2628	0.010257728	53
1	648.2234	0.010003092	50
1	525.7367	0.011938941	50
1	772.5935	0.008618208	41
2	623.9615	0.010331439	41
1	626.4238	0.010297061	41



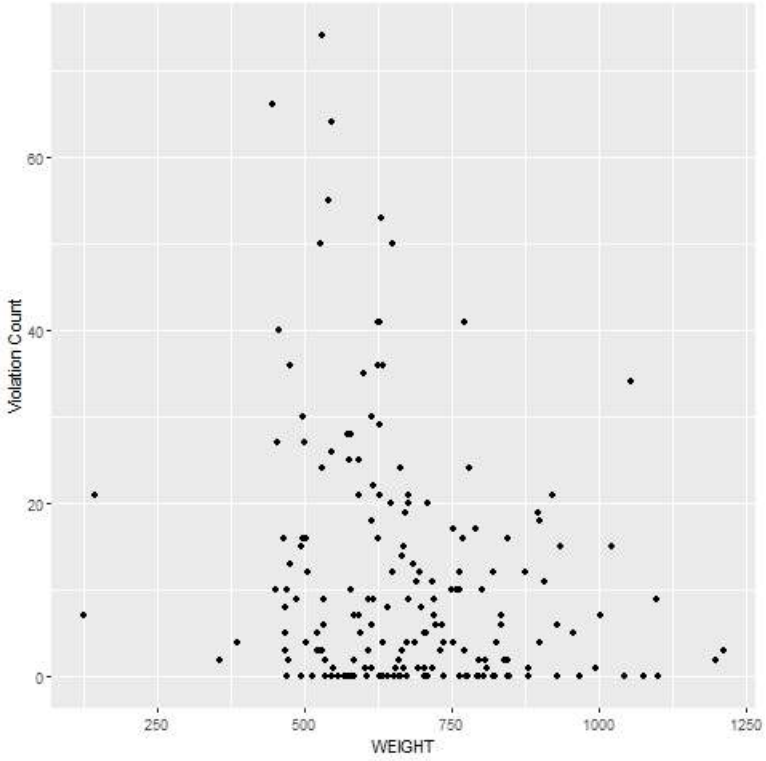
box--mu_argus--BORNUSA.jpeg



box--violation_count--BORNUSA.jpeg



scatter--mu_argus--WEIGHT.jpeg



scatter--violation_count--WEIGHT.jpeg

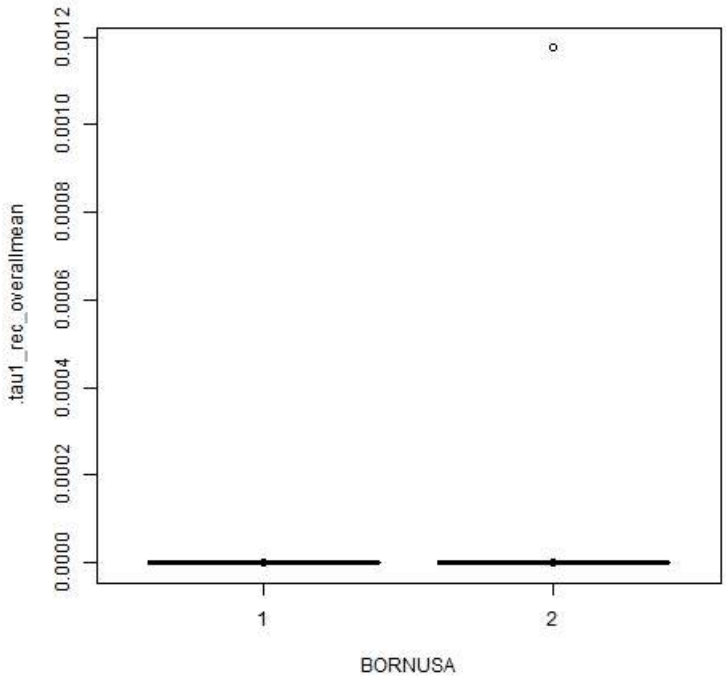
Appendix B Example R code and Output for sdc_loglinear

Example R code

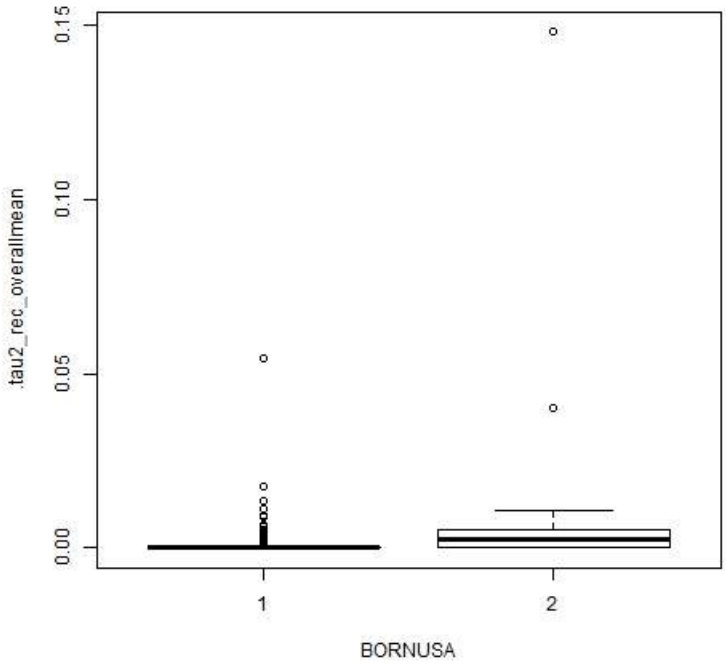
```
library(data.table)
library(MASS)
library(SDCNway)

data(exampledata)

vars <- c("BORNUSA", "CENREG", "DAGE3", "DRACE3", "EDUC3", "GENDER")
wgt <- "WEIGHT"

results <- sdc_loglinear(exampledata, wgt, vars, degree=2)
print(results, summary_outfile="ll_output.txt")
plot(results, plotvar1="BORNUSA", plotvar2="WEIGHT")
```

Example R output

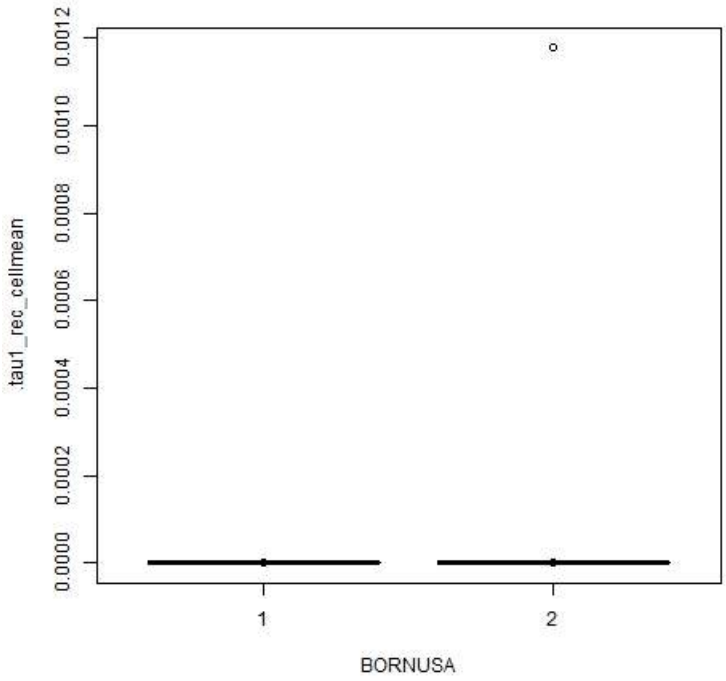
```
RESULTS - Uses overall average weight

 sampsize avg_cell_size interaction     tau1Risk  tau2Risk

      182        0.4213        none 0.0000015722 0.0028091

      182        0.4213       2-way 0.0000064554 0.0023804


       tau1    tau2 B_tau1_type1 B_tau1_type2 B_tau2_type1 B_tau2_type2

 0.00028614 0.51126     -0.53246   -0.0036307      0.86652      0.32756

 0.00117488 0.43323     -0.83796   -0.0175682     -2.28694     -0.18463


RESULTS - Uses cell average weight

 sampsize avg_cell_size interaction     tau1Risk  tau2Risk

      182        0.4213        none 0.0000015775 0.0028099

      182        0.4213       2-way 0.0000064734 0.0023812


       tau1    tau2 B_tau1_type1 B_tau1_type2 B_tau2_type1 B_tau2_type2

 0.00028711 0.51140     -0.53046   -0.0036091       0.8085      0.27794

 0.00117817 0.43339     -0.83762   -0.0175252      -2.2924     -0.18394
```



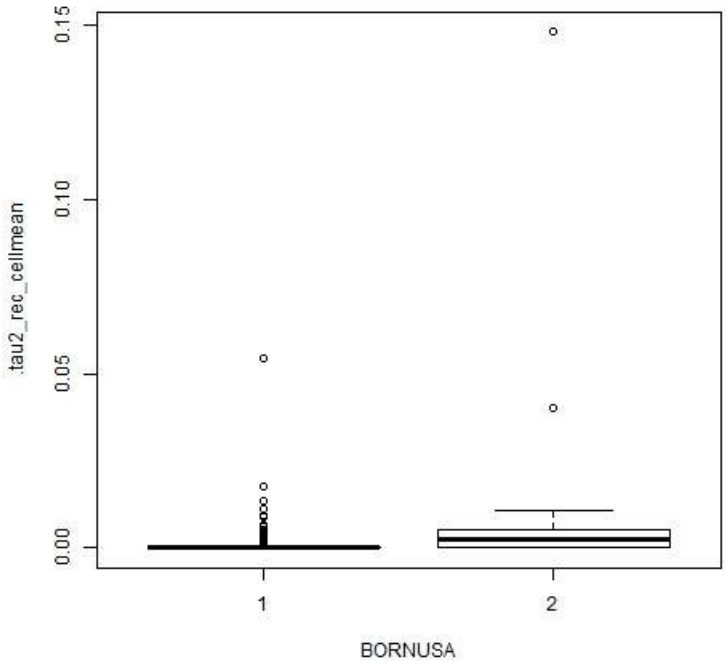
box–tau1_overallmean–BORNUSA.jpeg



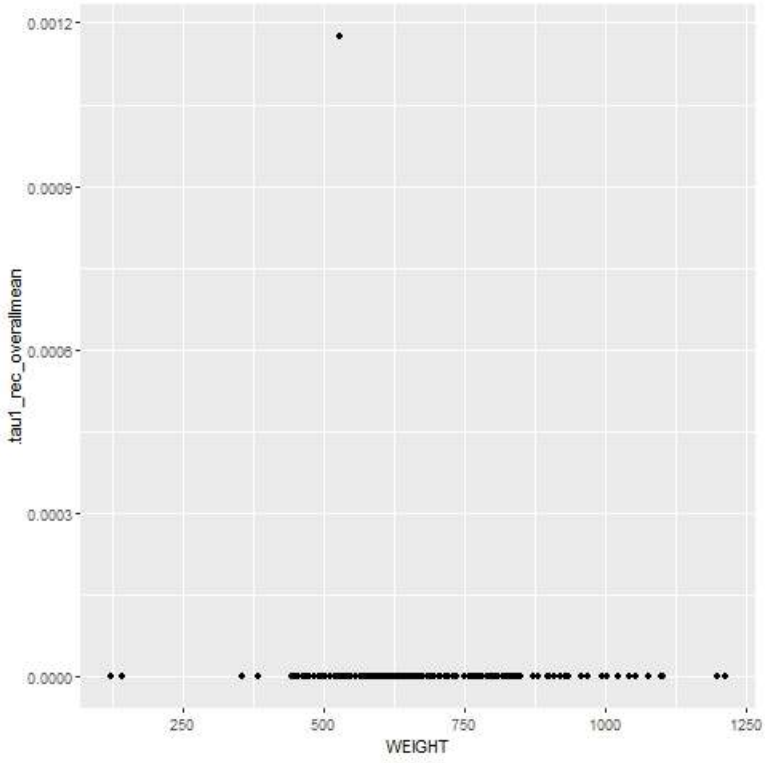
box–tau2_overallmean–BORNUSA.jpeg



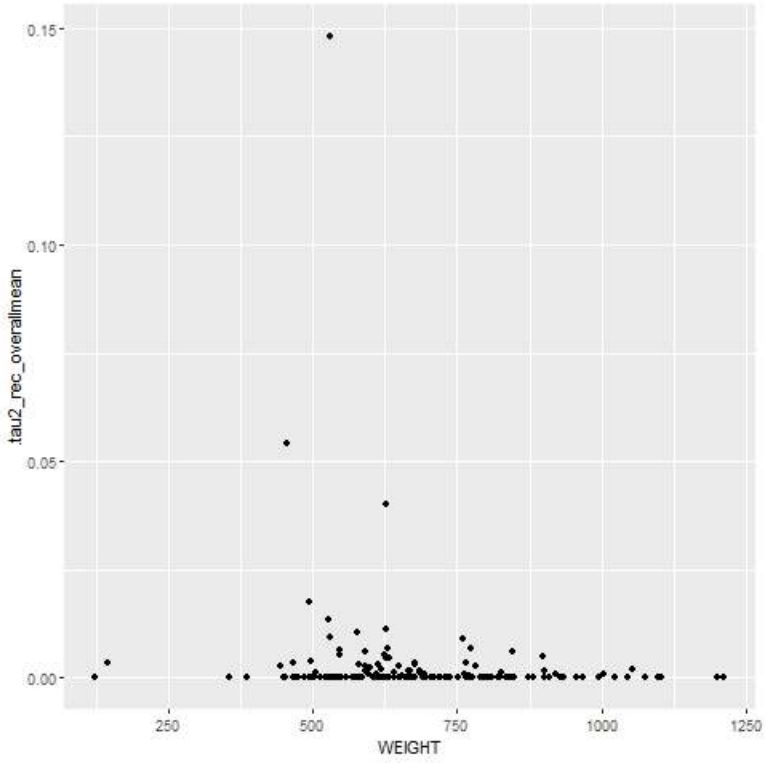
box–tau1_cellmean–BORNUSA.jpeg



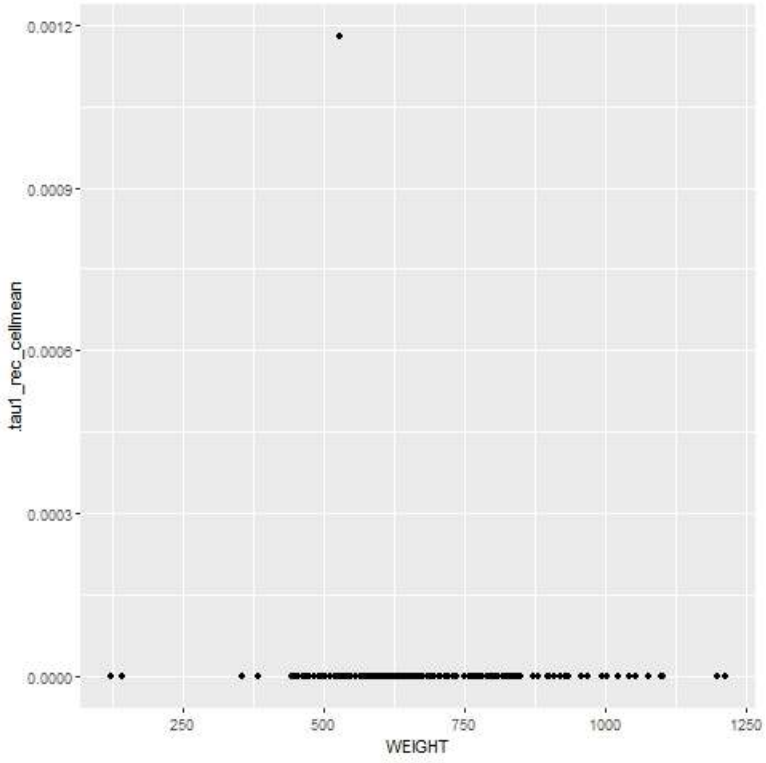
box–tau2_cellmean–BORNUSA.jpeg



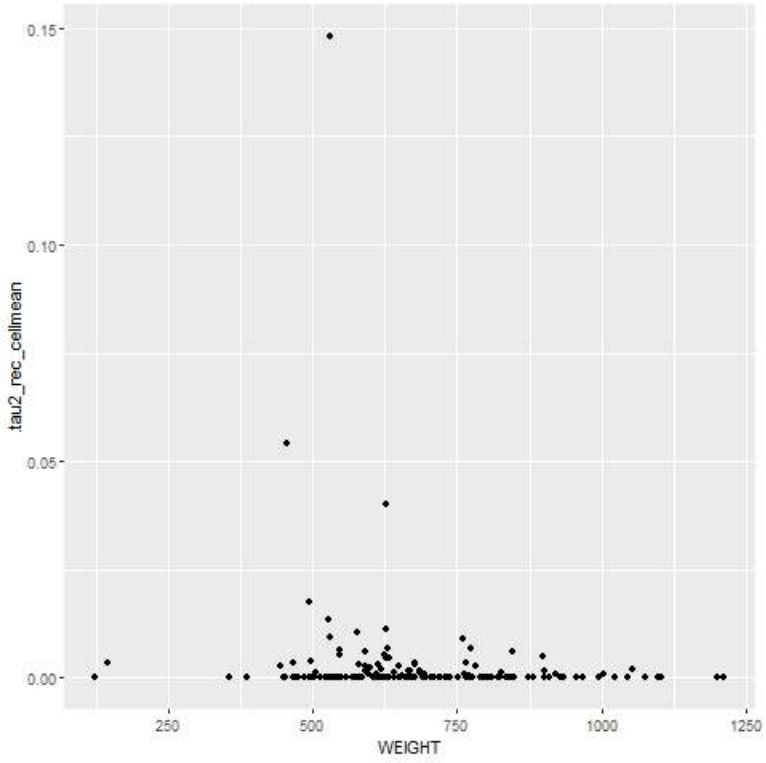
scatter–tau1_overallmean–WEIGHT.jpeg



scatter–tau2_overallmean–WEIGHT.jpeg



scatter–tau1_cellmean–WEIGHT.jpeg



scatter–tau2_cellmean–WEIGHT.jpeg

Reference

El Emam, K. (2011). Methods for the de-identification of electronic health records for genomic research. *Genome Medicine*, 3, 1-9. (Appendix: http://genomemedicine.com/content/supplementary/gm239-s1.pdf.

Polettini, S. (2003). Some remarks on the individual risk methodology. In: Proceedings of the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality. Luxembourg.

Skinner, C.J. and Shlomo, N. (2008). Assessing Identification Risk in Survey Microdata Using Log-linear Models. Journal of American Statistical Association, 103, 989– 1001.