# How-To

Maël-Luc Buron, Peter-Paul de Wolf, Bernhard Meindl, Tobias Enderle

2021-06-01

## protoTestCensus

Prototype Testing For Census (Target Record Swapping and Cell Key Method)

### How to use the open-source versions of the recommended census protection methods

(This text is also available in **pdf-format**.)

In the previous Specific Grant Agreement called *Harmonised protection of CENSUS data in the ESS* methods have been developed and tested. In deliverables 3.3 and 3.4 recommendations were given on how to protect the hypercubes and the grid data in the next Census. The two methods (record swapping and the cell key method) can be applied independently from each other.

However, with the SAS codes tested in this previous SGA, it was more straightforward to implement the cell key method. That is why part of the work in a new SGA called "Open source tools for perturbative confidentiality methods" is to implement these methods in the open-source software R and Argus. This was done in the last eight months, some tests were carried out by the SGA members and now we would like to invite census experts to try the new implementations. They can be downloaded on / installed from github: **https://github.com/sdcTools/protoTestCensus**.

Part 1 is related to targeted record swapping while part 2 focuses on the cell-key method. The methods can be used independently: it is either possible to compute unperturbed hypercubes with swapped microdata (part 2 can be ignored), perturbed hypercubes with original microdata (part 1 can be skipped), or perturbed hypercubes with swapped microdata (part 1 then part 2).

#### 1 Targeted record swapping

To apply record swapping to original microdata it is possible to use either the R package recordSwapping (paragraph 1.1) or muArgus (paragraph 1.2).

Both implementation use the same underlying new C++ code which is optimized to be incredibly fast. This new implementation of the targeted record swapping method is intended to be very close to the SAS codes tested in the previous SGA based on the ONS codes. More information regarding the small differences can be found in the vignette of the recordSwapping package.

All software for the prototype testing of the Targeted Record Swapping can be found **here**.

**1.1 package recordSwapping** The recordSwapping package needs to be installed from github using the devtools package with the following line of codes:

```
rm(list=ls())
install.packages("devtools")
devtools::install_github("sdcTools/recordSwapping", build_vignette=TRUE)
```

The vignette explains how the original microdata must be provided to the recordSwap function and how to use the different arguments. It also provides a use case of the recordSwap function based on a dummy dataset. This use case makes it possible to quickly understand the different arguments of the function, and makes it easier to start testing and trying different parameters.

**1.2 MuARGUS** The muArgus version that includes the test release implementation of record swapping can be downloaded from **here**. A quick reference documentation with explanations and screenshots can also be downloaded.

A small dataset is available in the test version of muArgus. It enables users to first replicate the walk through described in the quick reference documentation on this small dataset, before putting their own microdata in muArgus. The walk throught is described step by step. It starts by opening the microdata and specifying the metadata and describes the new Targeted Record Swapping window: how to select and order the microdata variables that will be used as argument of the C++ function, where to set the parameters of the function.

## 2 Cell-key method

The cell-key method can be applied either with the R package cellKey (paragraph 2.2) or tauArgus (paragraph 2.3). Both implementation use the ptable package (paragraph 2.1) but on the one hand it is included in the cellKey package (ptable still need to be installed but it is not necessary to use its functions because cellKey calls it directly), while on the other hand you need to use the ptable package to first compute a perturbation table, to export it, and use it in tauArgus.

All software for the prototype testing of the cell-key method can be found **here**.

**2.1 package ptable** The ptable package needs to be installed from github using the devtools package with the following line of codes:

```r
rm(list=ls())
update.packages(ask=FALSE)
install.packages("devtools")
library(devtools)
install_github("sdcTools/ptable", ref="v0.2.0_prototype",
  build_opts="--build-vignettes", dependencies=c("Imports","Depends"))
```

The package includes a vignette and a graphical user interface and it can draw plots that describe the pertubations that will be applied to the hypercubes based on a specific pTable with the cellKey package, tauArgus or SAS.

**2.2 package cellKey** The cellKey package also needs to be installed from github using the devtools package with the following line of codes. Note that these instructions automatically install all required dependencies, including package `ptable`. It is therefore not required to install the `ptable` package separately as shown in the previous section.

```r
rm(list=ls())
update.packages(ask=FALSE)
install.packages("devtools")
remotes::install_github(
  repo = "sdcTools/cellKey",
  ref = "v0.16.1",
  dependencies = TRUE,
  build_opts = "--no-resave-data",
  force = TRUE
)
```

The package includes a vignette with an example on how to describe and create hierarchies and use the different functions of the package. Users have to be aware that to keep the consistency between different hypercubes (i.e. that if the same cell appears in different hypercubes, it is perturbed in the same way), the same pTable need to be used. Including the ptable package in cellKey package makes it more straightforward to use, but this mistake of using different pTables for different hypercubes need to be avoided to keep the consistency of the method. However, if prefered, users can use the ptable package directly and use its output as input to the cellKey package.

**2.3 TauARGUS** The tauArgus version that includes the test release implementation of the cell-key method can be downloaded from **here**. A quick reference documentation with explanations and screenshots can also be downloaded.

The cell-key method is based on record keys that are randomly assigned to each record in the microdata. In tauArgus, it is necessary to add these record keys in the microdata before loading the microdata in tauArgus. It is possible to do so with these lines of R code:

```r
# Record keys (direct approach)
set.seed(123)
microData$recordkey <- format(runif(dim(microData)[1]), scientific=FALSE, nsmall=15)
```

Then in the metadata window of tauArgus there is a new type of variable “record key” to select. Once the data is loaded in tauArgus and described by the metadata, the cell-key method is applied from the main screen using the new option “Cell Key Method” in the suppress panel. A visual feedback with graded colors cell based on the perturbation applied to each can be selected using a “Colored view” check-box. A new CKM-format enables users to save the perturbed table, and if needed to add the original value, the difference, and the cell-key.

More information on the procedure is available in the quick reference documentation.