rtauargus

A simple way to protect multiple linked tables with R and Tau-Argus

Workshop of the SDC User Group

Julien Jamme
Nathanaël Rastout
Clément Guillo

2022/09/20

Table of Contents

Reminders

Previous presentation

- About the first presentation during the User Group Meeting in February
- But, no English documentation was provided

Reminders

Developing a new package, what for ?

Context : Apply suppressive methods on magnitude and frequency tables

- Preference in using statistical software like R or Python :
  - Keep the code to do a reproducible work ;
  - some complex confidentiality problems require to write algorithm ;
- On R, lack of reliable tools to treat complex demands ;
- Large demand of an R tool within the Institute ;
- τ-Argus is unavoidable to treat complex confidentiality problems.

Reminders

About making a reproducible work with τ-Argus

- Objectives
  - combine reliability of τ-Argus and reproducibility with R
  - make an *easy to handle* package
- Characteristics
  - main purpose : prepare the data with the appropriate formats and write a batch file to run with τ-Argus
  - simple functions that we can combine in custom-made algorithms

Reminders

rtauargus : What it is not !

- a competitor to sdcTable, because rtauargus doesn't handle confidentiality itself *(still true, an alliance could even be found)*
- a solution to deal with any of the issues without effort : algorithms need to be thought separately *(this is not quite true anymore).*

New version

What is really new, since February ?

- Release the functions to proceed with tabular data (not only microdata)
- Documentation of all functions has been translated in English
- Some vignettes in English are provided to show how to use them
- But also, new features to make the protection of tables easier.

$\Longrightarrow$ Introducing these new features is the main topic of this presentation.

- Original github repo : https://github.com/InseeFrLab/rtauargus
- Documentation website : https://inseefrlab.github.io/rtauargus
- sdcTools github repo : https://github.com/sdcTools/rtauargus

Modus Operandi

microdata vs tabular data

- Initially, the package was developed to proceed with microdata
- Microdata management is the usual way (as it seems to be the case around us) to use τ-Argus
- But, we chose to work only with tabular data.
- In `rtauargus`, two kinds of functions :
  - `micro_XXX()` functions to deal with microdata ;
  - `tab_XXX()` functions to deal with tabular data ;
- New features have been implemented only for tabular data.

Modus Operandi

Why choosing tabular data ?

- Tau-Argus can deal with microdata to protect some tables at once, but not when there are many ;
- Microdata file can be very heavy ;
- The primary suppression on microdata with holding and weights is not possible ;
- On the contrary, primary suppression is easy to implement with tabular on R ;
- $\tau-$Argus can deal with *apriori* file
- When many linked tables have to be protected, *ad hoc* algorithms have to be implemented and it's easier with tabular than with microdata.

rtauargus::tab_rtauargus()

All operations in one

`rtauargus::tab_rtauargus()` :

- to protect one table with τ-Argus ;
- creates all the files (tab, rda, hst, arb) required by the software ;
- runs the batch
- imports and returns the result as a dataframe.

https://inseefrlab.github.io/rtauargus/articles/rtauargus.html

RTAUARGUS::TAB_RTAUARGUS()

Example of an input

ACTIVTY	SIZE	VAL	N_OBS	MAX
01	tr1	25	18	5
01	tr2	15	12	5
01	tr3	20	5	10
01	Total	60	35	10
02	tr1	90	15	20
02	tr2	65	3	40
02	tr3	120	2	80
02	Total	275	20	80
Total	tr1	115	33	20
Total	tr2	80	15	40
Total	tr3	140	7	80
Total	Total	335	55	80

Context :

- Protect one table (called *act_size*)
- Crossing two variables (*ACTIVITY* and *SIZE*)
- primary suppression with τ-Argus
- frequency and dominance rules
- Default method for secondary suppression : Modular.

RTAUARGUS::TAB_RTAUARGUS()

How to use it ?

```
ex1 <- tab_rtauargus(
  act_size,
  dir_name = "tauargus_files",
  files_name = "ex1",
  explanatory_vars = c("ACTIVITY","SIZE"),
  safety_rules = "FREQ(3,10)|NK(1,85)",
  value = "VAL",
  freq = "N_OBS",
  maxscore = "MAX",
  totcode = c(ACTIVITY="Total",SIZE="Total")
)
```

Figure – Using `rtauargus::tab_rtauargus()`

Context :

- Protect one table (called *act_size*)
- Crossing two variables (*ACTIVITY* and *SIZE*)
- primary suppression with τ-Argus
- frequency and dominance rules
- Default method for secondary suppression : Modular.

RTAUARGUS::TAB_RTAUARGUS()

Main τ-Argus settings are available

`rtauargus::tab_rtauargus()` displays the main τ-Argus settings

- `safety_rules`, to set the primary suppression rules (or at least the manual safety range)
- `suppress`, to set the method used for secondary suppression
- A vignette lists the options available for both these arguments (in τ-Argus language) : https://inseefrlab.github.io/rtauargus/articles/options_safety_rules.html
- `secret_var`, in case of needing to use an apriori ;
- `cost_var`, in case of needing to use a specific variable for cost.

RTAUARGUS::TAB_MULTI_MANAGER()

The actual improvement provided

`rtauargus::tab_multi_manager()` :

- To protect a set of (linked) tables ;
- Manages all the links between the tables in the set ;
- Just one call to protect the whole set.

RTAUARGUS::TAB_MULTI_MANAGER()

What does the user have to provide ?

- A list of tables ;
  - with values and frequencies to compute secondary suppression ;
  - with a boolean variable containing the status of each cell after the primary suppression.
- A description of the explanatory variables for each table ;
- The hierarchy files if needed.

Example : Two tables with two crossing variables each, including one common variable.

T1

ACT	SIZE
A	tr1
A	tr2
A	Total
B	tr1
B	tr2
B	Total
Total	tr1
Total	tr2
Total	Total

T2

ACT	CJ
A	c1
A	c2
A	Total
B	c1
B	c2
B	Total
Total	c1
Total	c2
Total	Total

RTAUARGUS::TAB_MULTI_MANAGER()

How does it work ?

First, we merge all the tables in one **main table**.

ACT	SIZE	CJ	is_T1	is_T2
A	tr1	Total	TRUE	FALSE
A	tr2	Total	TRUE	FALSE
B	tr1	Total	TRUE	FALSE
B	tr2	Total	TRUE	FALSE
Total	tr1	Total	TRUE	FALSE
Total	tr2	Total	TRUE	FALSE
A	Total	c1	FALSE	TRUE
A	Total	c2	FALSE	TRUE
B	Total	c1	FALSE	TRUE
B	Total	c2	FALSE	TRUE
A	Total	Total	TRUE	TRUE
B	Total	Total	TRUE	TRUE
Total	Total	c1	FALSE	TRUE
Total	Total	c2	FALSE	TRUE
Total	Total	Total	TRUE	TRUE

RTAUARGUS::TAB_MULTI_MANAGER()

How does it work ?

The algorithm handles two lists :

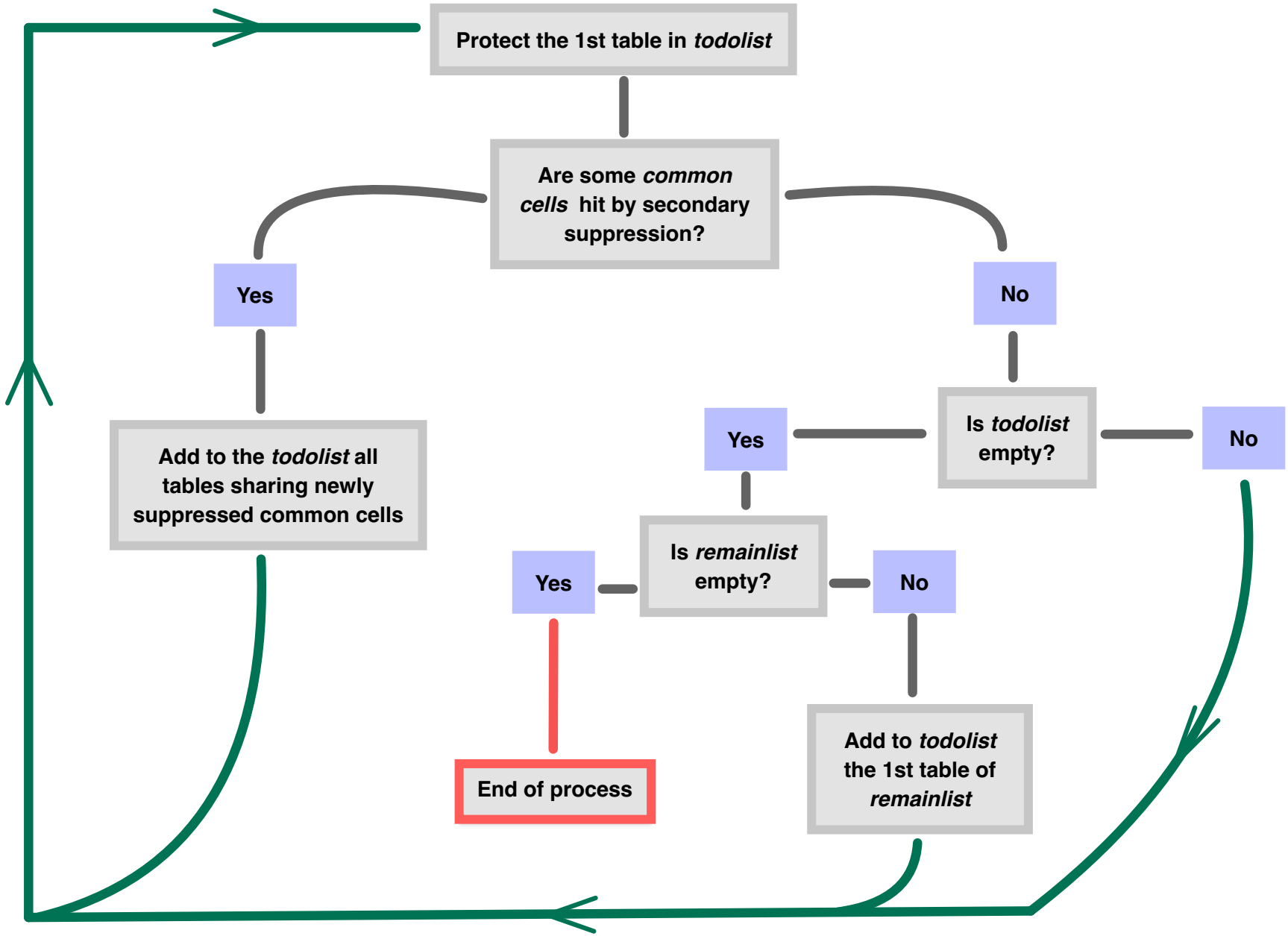
- ***todolist*** = list of tables which have to be protected.
- ***remainlist*** = list of original tables which haven't yet been protected at all.

Initialization of the algorithm :

- ***todolist*** = first table in the original list (eg. T1)
- ***remainlist*** = all the tables of the original list except the first one (eg. T2)

RTAUARGUS::TAB_MULTI_MANAGER()

How does it work ?



RTAUARGUS::TAB_MULTI_MANAGER()

How does it work ?

The protection carries on :

- While the ***remainlist*** is not empty $\Leftrightarrow$ While all the tables of the original list have not been protected at least once.
- While the ***todolist*** is not empty $\Leftrightarrow$ While the ***remainlist*** is not empty or as long as there are **common cells** hit by secondary suppression.

RTAUARGUS::TAB_MULTI_MANAGER()

What do we get ?

The function returns the original list of tables with some other variables :

- All steps of suppression are included in each table ;
- Boolean variables (TRUE if the cell has to be masked, otherwise FALSE).
- Each step takes the previous one into account ;
- The last variable indicates the final step ;
- Final status of cell is easily computable with this last variable and the primary suppression status previously computed by the user.

RTAUARGUS::TAB_MULTI_MANAGER()

What do we get ?

The function writes :

- All τ-Argus files in a directory ;
- At the end, only the last process for each table is available ;
- A **journal** is provided
- with a description of each step :
  - Which table is being protected ;
  - After the protection, how many new common cells have been hit by secondary suppression.
- with also a description of all common cells that have been hit, with the iteration of when they have been hit.

RTAUARGUS::TAB_MULTI_MANAGER()

Example 1

- Provided in the package documentation (data and codes)
- Two linked tables of two dimensions each
- T1 : 414 rows ; T2 : 406 rows
- One hierarchical variable present in both tables (Business sector).

https://inseefrlab.github.io/rtauargus/articles/rtauargus.html

RTAUARGUS::TAB_MULTI_MANAGER()

Example 1

The protection path :

- Intialization :
  - ***todolist*** = T1
  - ***remainlist*** = T2
- Step 1 :
  - Protection of T1
  - 11 common cells hit
  - ***todolist*** = T2
  - ***remainlist*** = $\emptyset$
- Step 2 :
  - Protection of T2
  - 4 common cells hit
  - ***todolist*** = T1
  - ***remainlist*** = $\emptyset$
- Step 3 :
  - Protection of T1
  - 0 common cell hit
  - ***todolist*** = $\emptyset$
  - ***remainlist*** = $\emptyset$

RTAUARGUS::TAB_MULTI_MANAGER()

Example 1

```
list_data_2_tabs <- list(
  act_size = turnover_act_size,
  act_cj = turnover_act_cj
) %>%
  purrr::map(
    function(df){
      df %>%
        mutate(
          is_secret_freq = N_OBS > 0 & N_OBS < 3,
          is_secret_dom = MAX > TOT*0.85,
          is_secret_prim = is_secret_freq | is_secret_dom
        )
    }
  )
```

Figure – Example 1 : About creating the list of tables and applying the primary suppression

RTAUARGUS::TAB_MULTI_MANAGER()

Example 1

```
ex3 <- tab_multi_manager(
  list_tables = list_data_2_tabs,
  list_explanatory_vars = list(
    act_size = c("ACTIVITY", "SIZE"),
    act_cj = c("ACTIVITY", "CJ")
  ),
  hrc = c(ACTIVITY = hrc_file_activity),
  dir_name = "ex3",
  value = "TOT",
  freq = "N_OBS",
  secret_var = "is_secret_prim",
  totcode =  "Total"
)
```

Figure – Example 1 : About running secondary suppression

RTAUARGUS::TAB_MULTI_MANAGER()

Example 2

An example with 4 tables :

- Four tables T1, T2, T3, T4
- T1 : ACTIVITY * SIZE
- T2 : ACTIVITY * CJ
- T3 : NUTS * SIZE
- T4 : NUTS * CJ

RTAUARGUS::TAB_MULTI_MANAGER()

Example 2

- Provided in the package documentation (data and codes)
- Four linked tables of two dimensions each
- T1 : 414 rows ; T2 : 406 rows ; T3 : 460 rows ; T4 : 452 rows
- Two hierarchical variables :
  - Business sector (ACTIVITY), present in T1 and T2 ;
  - Geographical area (NUTS), present in T3 and T4.
- Links :
  - T1 and T2 are linked by ACTIVITY variable
  - T3 and T4 are linked by NUTS variable
  - T1 and T3 are linked by SIZE variable
  - T2 and T4 are linked by CJ variable

RTAUARGUS::TAB_MULTI_MANAGER()

Example 2

The protection path :

- Intialization :
  - ***todolist*** = T1
  - ***remainlist*** = T2,T3,T4
- Step 1 :
  - Protection of T1
  - 11 common cells hit
  - ***todolist*** = T2
  - ***remainlist*** = T3,T4
- Step 2 :
  - Protection of T2
  - 4 common cells hit
  - ***todolist*** = T1
  - ***remainlist*** = T3,T4
- Step 3 :
  - Protection of T1
  - 0 common cell hit
  - ***todolist*** = T3
  - ***remainlist*** = T4
- Step 4 :
  - Protection of T3
  - 5 common cells hit
  - ***todolist*** = T4
  - ***remainlist*** = $\emptyset$
- Step 5 :
  - Protection of T4
  - 0 common cell hit
  - ***todolist*** = $\emptyset$
  - ***remainlist*** = $\emptyset$

Comparison with Tau-Argus microdata

- Tau-Argus can deal with this kind of examples
- The algorithm leads to a slight oversuppression on T1, T3 and T4 (2 cells more over 80/100 suppressed cells in total).
- But the limit number of the set in our function exceeds the limit of tau-argus with microdata approach ;
- The limit depends on the computer power but we are confident to be able to deal with more than 50 tables at a time.
- Tests in progress on ICT survey.

	T1	T2	T3	T4
Argus from microdata	81	90	80	99
tab_multi_manager()	83	90	82	101

Table – Comparison of the results between Tau-Argus microdata approach and tab_multi_manager() on the second example

Further developments

- mainly (only ?) on tabular functions (microdata functions could be deprecated in a next version ;
- fix bugs ;
- try to plug the `rtauargus::tab_multi_manager()` to `sdcTable` to use `simple heuristic` method ;
- enrich the vignettes ;
- yours and others' issues

Collaborate

Writing issues :

- on `sdcTools Usersupport site` : https://github.com/sdcTools/UserSupport/issues
- on `github Insee repository` : https://github.com/InseeFrLab/rtauargus/issues

Reporting uses to us :

- julien.jamme@insee.fr
- nathanael.rastout@insee.fr
- clement.guillo@insee.fr