TSACE Webinar, Wednesday December 14th 2022

Using JDemetra+ in R: from version 2 to version 3
Presentation 3: Wrangling workspaces in R

Anna Smyk and Tanguy Barthelemy
With the collaboration of Alain Quartier-la-tente

Contents

Context

- a workspace is a JD+ specific data structure (xml files) which allows to use the GUI and cruncher
- in R we don't need this stucture to run an SA process (cf P2)
- we could think sa processing with JD+ algorithms can happen in two disntict worlds
  - GUI and cruncher using workspaces OR
  - R functions (without workpaces, TS object or numeric vectors for HF)

Still there are benefits in using workspaces and R packages in conjunction

We will highlight them in the remaining of this presentation

Contents

Using workspaces and R packages in conjuction

Two (?) classes of benefits

- instant reading: immersing results in the R world (functions, plots..)
- automating potential manual operations

De manière générale, aucune compatibilité entre v2 et v3

Contents

Instant reading

v2 : `RJDemetra::load_workspace()` -> creation of RJD object (java object pointer) v3 : `rjdemetra3::load_workspace()` -> readable R object (list format)

Contents

Automating operations

On me voit ?

Automating operations on workspaces

Goal: to produce and to reproduce the workspace structure without the JDemetra+ graphical interface but with R and keeping the read, write and refresh properties both by the graphical interface and by the cruncher.

Cruncher and GUI react the same way to a WS : - if the structure is not good : they won't be able to read anything - if the structure is minimal : they will manage to only read and compute the SA-items - if the ws are well-structured and have all the necessairy information : they will succeed to read, compute and refresh with the chosen policy our sa-items.

Outline of the automation

Presentation in v2

operations - creation - export / write - modifications / dynamic update

Creation

`RJDemetra::new_workspace()` : creation of workspace
`RJDemetra::new_multiprocessing()` : creation of multiprocessing

To create a spec, there is several functions : `RJDemetra::x13_spec()`, `RJDemetra::tramoseats_spec()`, `...` and also in v3 with packages `rjd3tramoseats` and `rjd3x13`

To create an SA-item, there is several functions depending on the chosen spec : `RJDemetra::x13()`, `RJDemetra::tramoseats()`, `...` and also in v3 with packages `rjd3tramoseats` and `rjd3x13`

Finally you can use `RJDemetra::add_sa_item()` to add the created SA-item to a multprocessing in your workspace

Export

The function `RJDemetra::save_workspace()` export and create a physical workspace with a structure in folder and xml files. But it only creates what the previous functions constructed.

For example, the function `RJDemetra::x13()`

Dynamic update

Attention, the exported workspace

Automating operations on series

("functions that operate on one workpace") - series object - spec - meta data in v2 RJDemetra in v3 rjdworkpace and other rjd3...

Merging workspaces

("functions that operate two workpaces") context: annual rev

in v2 RJDemetra in v3 rjdworkpace and other rjd3...

Preserving desirable properties ?

two features to preserve - GUI readable - Crunchable

When are the preserved and when not ?

Contents

Summary of functionnalities

- available
- missing

Package's perimeter and ressources

- V2 vs V3: not clear cut yet not many features are available in sheer v3
  rjdworkspace (predating v3) functions will be (partly) integrated into rjdemetra3 package