

3 - Production de séries CVS-CJO avec JDemetra+ et R

Series Temporelles avec R (Avancé)

Anna Smyk, Tanguy Barthelemy

Insee - Département des Methodes Statistiques



Table of Contents I

- 1 Introduction
- 2 Selection de regresseurs CJO
- 3 New R Tools for wrangling workspaces: complements

Section 1

Introduction

Section 2

Selection de regresseurs CJO

Selection de regressseurs CJO

- plus de details (vs partie 1)
- garder des outliers ?

Section 3

New R Tools for wrangling workspaces: complements

Présentation

Objectif : profiter des atouts de la GUI et automatiser les opérations manuelles à forte intensité de clics avec les outils R (pas seulement le processus d'estimation et les outputs du cruncher)

Les packages **{rjd3providers}** et **{rjd3workspace}** fournissent des fonctions pour :

- La création, la lecture des workspaces, la mise à jour du lien entre les données brutes et le workspace
- La personnalisation des spécifications à grande échelle, comme la pre-spécification d'outliers série par série ou l'attribution de régresseurs de calendriers
- La fusion de workspace par nom de série, comme vous le feriez avec un tableau de données.

Cela peut s'avérer utile lors de la copie de spécifications et de modèles sélectionnées à partir d'une ancienne version du processus. Si le processus précédent n'est pas JDemetra+, des convertisseurs de spécifications sont disponibles dans X13 et Tramo-Seats. Les nouvelles spécifications peuvent également être stockées sous forme d'objets XML.

Voyons quelques exemples.

Chargement des packages R

```
library("rjd3toolkit")  
library("rjd3x13")  
library("rjd3tramoseats")  
library("rjd3workspace")
```


Création de workspaces

- Création automatisée de workspaces

rjd3workspace dispose d'une collection de fonctions permettant de créer un workspace et des multiprocessus (équivalent d'une feuille dans un fichier tableur) :

Deux étapes :

- Créer un workspace *virtuel* : `jws_new()`
- Créer un multiprocessus *virtuel* : `jws_sap_new()`

i Vocabulaire

Objet **virtuel** = objet existant uniquement dans la session **R** et non écrit dans un dossier sur votre ordinateur

```
mon_ws <- jws_new
mon_sap <- jws_sap_new(jws = mon_ws, name = "sap1")
```

Lecture de workspace

Pour lire (ouvrir) un workspace écrit sur votre ordinateur, on utilise la fonction `load_workspace()` du package `{rjd3workspace}`.

```
mon_ws <- load_workspace(file = "chemin/menant/à/mon/ws.xml")
```

Enregistrement de workspaces

La fonction `save_workspace()` exporte dans un nouvel emplacement et crée un *vrai* workspace (avec dossier et fichiers XML).

```
save_workspace(jws = mon_ws, file = "où/je/veux/enregistrer/mon/ws.xml")
```

Ajouter des séries aux workspaces (1/3)

Vous pouvez ajouter un nouvel SA-Item (données, paramètres, diagnostics liés à une série) à un workspace de différentes manières.

Avec un SA-Item créé par R

On peut utiliser les fonctions `rjd3x13::x13` et `rjd3tramoseats::tramoseats` pour créer directement notre SA-Item et l'ajouter au WS

```
# add a SA-Item created with R
sa_x13 <- x13(rjd3toolkit::ABS[, 1])
sa_ts <- tramoseats(rjd3toolkit::ABS[, 2])

add_sa_item(jsap = new_jsap, name = "ABS_1", x = sa_x13)
add_sa_item(jsap = new_jsap, name = "ABS_2", x = sa_ts)
```

Ajouter des séries aux workspaces (2/3)

Avec une specification et des données brutes

```
# add a raw series with a spec created in R
add_sa_item(
  jsap = new_jsap,
  name = "ABS_3",
  x = rjd3toolkit::ABS[, 3],
  spec = rjd3x13::x13_spec(name = "RSA5c")
)
add_sa_item(
  jsap = new_jsap,
  name = "ABS_4",
  x = rjd3toolkit::ABS[, 4],
  spec = rjd3tramoseats::tramoseats_spec(name = "rsafull")
)
```

Ajouter des séries aux workspaces (3/3)

A partir d'un autre workspace

```
# add a SA_Item from another workspace  
add_sa_item(jsap = new_jsap, name = "ABS_4", x = jsai_from)
```

Remplacer des séries

La fonction `rjd3workspace::replace_sa_item` permet à l'utilisateur de remplacer un SA-Item existant par un nouveau.

```
# On remplace ici le premier SA-Item du SA-Processing jsap_object par jsai.  
replace_sa_item(  
  jsap = jsap_object,  
  jsai = jsai_from,  
  idx = 1L  
)
```

Suppression de series

Il est aussi possible de supprimer des SA-Item d'un workspace avec les fonctions `rjd3workspace::remove_sa_item()` and `rjd3workspace::remove_all_sa_item()`.

```
# On supprime le SA-Item n° 5 et 6 du SA-Processing jsap_object
remove_sa_item(jsap = jsap_object, idx = 5L)
remove_sa_item(jsap = jsap_object, idx = 6L)

# On supprime tout les SA-Item du SA-Processing jsap_object
remove_all_sa_item(jsap = jsap_object)
```


Mise à jour du chemin vers les données brutes I

Pour être pleinement fonctionnel, un workspace doit lier un SA-Item au chemin vers les données brutes et ce chemin doit être valide. Ce lien peut-être rompu si les données ont été déplacées, supprimées ou renommées.

Les fonctions `XXX_update_path()` (du package `{rjd3providers}`) mettent à jour le chemin menant aux données en fonction de l'extension du fichier (XLS pour un tableur ou TXT pour un fichier plat).

Arguments :

- `jws` : le pointeur Java vers notre WS
- `new_path` : le nouveau chemin vers les données brutes.
- `idx_sap` : le numéro du SAP contenant la série à mettre à jour. (Optionnel)
- `idx_sai` : le numéro du SA-Item de la série à mettre à jour. (Optionnel)

Mise à jour du chemin vers les données brutes II

Exemple

```
txt_update_path(  
  jws = jws_object,  
  new_path = path_csv,  
  idx_sap = 1L  
)  
spreadsheet_update_path(  
  jws = jws_object,  
  new_path = path_xlsx,  
  idx_sap = 2L  
)
```

Fusion de 2 workspaces

Exemple de cas d'utilisation : campagne annuelle

- L'estimation actuelle et la nouvelle estimation sont comparées à l'aide d'un score
- Les séries ayant un score inférieur seront remplacées dans le workspace actuel par leur nouvelle estimation et leurs nouveaux paramètres.

On utilise la fonction `transfer_series()` dans `{rjd3workspace}`

Exemple

```
transfer_series(  
  jsap_from = sap_auto,  
  jsap_to = sap_travail,  
  selected_sa_items = c("RF0610", "RF0620")  
)
```

Personnalisation de spécifications I

Cas d'utilisation : les spécifications sont connues mais peuvent être différentes pour chaque série, dont le nombre est important.

Deux cas très courants :

- Recopie d'outliers (à partir d'un ancien workspace) série par série
- Ajout ou modification de calendrier ou autres régresseurs externes

On peut utiliser toutes les fonctions de modification de **{rjd3toolkit}**, **{rjd3x13}** et **{rjd3tramoseats}**.

Personnalisation de spécifications II

Exemple

```
sai <- read_sai(my_jsai)
my_spec <- sai$estimationSpec

my_new_spec <- my_spec |>
  add_outlier(type = "A0", date = "2022-01-01") |>
  set_x11(
    mode = "LogAdditive",
    seasonal.filter = "S3X9",
    henderson.filter = 7
  )
```

Rafraîchir un workspace en R

Sans lancer le Cruncher, on peut rafraîchir des modèles avec les fonctions `jsap_refresh()` et `jws_refresh()` de **{rjd3workspace}**.

Cette étape permet l'estimation et le rafraîchissement d'un WS ou d'un SA-Processing (SAP) et propose une mise à jour plus granulaire que le cruncher (qui tourne sur tout un workspace).

```
jws_refresh(mon_ws, policy = "Complete")
```