

Le lancement du cruncher avec JDCruncher pdf

2022-11-08

Contents

| | |
|--------------------------------------------------------------------------------|----------|
| À propos du cruncher | 1 |
| Création du fichier de paramètres avec <code>create_param_file()</code> | 2 |
| Lancement du cruncher | 3 |

La version actuelle du package JDCruncher (0.2.4) permet de faire deux choses :

- lancer facilement le cruncher de JDemetra+ (JWSACruncher) pour mettre à jour un workspace et en exporter les résultats.
- calculer un *bilan qualité* qui permet de synthétiser l'ensemble des diagnostics liés aux séries désaisonnalisées, afin de repérer rapidement les séries les plus problématiques sur lesquelles il faut concentrer son analyse. Ce bilan est surtout utile dans le cas de la désaisonnalisation d'un grand nombre de séries puisqu'il est dans ce cas impossible de regarder l'ensemble des diagnostics pour chaque série dans un temps raisonnable.

À propos du cruncher

Pour éviter que le présent package (JDCruncher) ne soit trop volumineux, il ne contient pas le cruncher de JDemetra+. Ce dernier peut être téléchargé à l'adresse suivante: <https://github.com/jdemetra/jdemetra-app/releases>. Pour plus d'informations sur l'installation du cruncher et sa configuration avec une version portable de Java, voir le wiki de JDCruncher : <https://github.com/InseeFr/JDCruncher/wiki/Installation-et-configuration-de-JDemetra--et-du-cruncher>.

Pour lancer le cruncher de JDemetra+, il faut trois éléments :

- un fichier contenant les paramètres sur la méthode de rafraîchissement à utiliser pour mettre le workspace à jour ;
- un workspace valide de JDemetra+ ;
- le chemin d'accès au cruncher.

Dans le package JDCruncher, il existe trois fonctions associées au lancement du cruncher :

- `create_param_file()` qui permet de créer le fichier de paramètres ;
- `cruncher()` qui permet de lancer le cruncher sur un workspace à partir d'un fichier de paramètres ;
- `cruncher_and_param()` qui permet de lancer le cruncher tout en créant le fichier de paramètres et de personnaliser certaines sorties du cruncher.

Création du fichier de paramètres avec `create_param_file()`

Les paramètres de la fonction `create_param_file()` sont les mêmes que ceux décrits dans le wiki du cruncher de JDemetra+ (<https://github.com/jdemetra/jwsacruncher/wiki>). Les trois paramètres les plus importants de `create_param_file()` sont :

1. `policy` qui est la méthode de rafraîchissement utilisée (voir tableau ci-dessous).

Table 1: Les différentes politiques de rafraîchissement

| Option sous JDemetra+ | Option du cruncher | Signification |
|-----------------------------------------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Partial concurrent adjustment | | |
| Fixed model | current | Le modèle ARIMA, les outliers et les autres paramètres du modèle de régression ne sont ni ré-identifiés ni ré-estimés. Le schéma de décomposition est inchangé. |
| Estimate regression coefficients | fixedparameters (ou fixed) | Le modèle ARIMA, les outliers et les autres paramètres du modèle regARIMA ne sont pas ré-identifiés. Les coefficients du modèle ARIMA sont fixés et les autres paramètres du modèle de régression sont ré-estimés. Le schéma de décomposition est inchangé. |
| Estimate regression coefficients + Arima parameters | parameters (paramètre par défaut) | Le modèle ARIMA, les outliers et les autres paramètres du modèle de régression ne sont pas ré-identifiés mais sont tous ré-estimés. Le schéma de décomposition est inchangé. |
| Estimate regression coefficients + Last outliers | lastoutliers | Le modèle ARIMA, les outliers (sauf ceux de la dernière année) et les autres paramètres du modèle de régression ne sont pas ré-identifiés mais sont tous ré-estimés. Les outliers de la dernière année sont ré-identifiés. Le schéma de décomposition est inchangé. |
| Estimate regression coefficients + all outliers | outliers | Le modèle ARIMA et les paramètres du modèle regARIMA autres que les outliers ne sont pas ré-identifiés mais ré-estimés. Tous les outliers sont ré-identifiés. Le schéma de décomposition est inchangé. |
| Estimate regression coefficients + Arima model | stochastic | Ré-identification de tous les paramètres du modèle regARIMA hormis les variables calendaires. Le schéma de décomposition est inchangé. |
| Concurrent | | |
| Concurrent | complete ou concurrent | Ré-identification de tout le modèle regARIMA. |

2. `matrix_item` qui est une chaîne de caractères contenant les noms des paramètres à exporter. Par défaut ce sont ceux de l'option `default_matrix_item`. On peut donc au choix modifier l'option `default_matrix_item` ou le paramètre `matrix_item` :

```
library(JDCruncher)
# Pour afficher les paramètres par défaut :
getOption("default_matrix_item")
# Pour modifier les paramètres par défaut pour n'exporter par exemple
# que les critères d'information :
options(default_matrix_item = c("likelihood.aic",
                                "likelihood.aicc",
                                "likelihood.bic",
                                "likelihood.bicc"))
```

3. `tsmatrix_series` qui est une chaîne de caractères contenant les noms des paramètres à exporter. Par

défaut ce sont ceux de l'option `default_tsmatrix_series`. On peut donc au choix modifier l'option `default_tsmatrix_series` ou le paramètre `tsmatrix_series` :

```
# Pour afficher les paramètres par défaut :
getOption("default_tsmatrix_series")
# Pour modifier les paramètres par défaut pour n'exporter par exemple que
# la série désaisonnalisées et ses prévisions :
options(default_tsmatrix_series = c("sa", "sa_f"))
```

Pour voir l'ensemble des paramètres, il suffit d'utiliser sous R la commande `?create_param_file`.

Après cela, il ne reste plus qu'à créer le fichier de paramètres. Ci-dessous quelques exemples.

```
# Un fichier parametres.param sera créé sous D:/ avec la politique de rafraîchissement
# "lastoutliers" et les autres paramètres par défaut
create_param_file(dir_file_param = "D:/",
                  policy = "lastoutliers")

# Si l'on a modifié les options "default_matrix_item" et "default_tsmatrix_series" pour
# n'exporter que les critères d'information, la série désaisonnalisée et ses
# prévisions, la commande précédente est équivalent à :
create_param_file(dir_file_param = "D:/",
                  policy = "lastoutliers",
                  matrix_item = c("likelihood.aic", "likelihood.aicc",
                                  "likelihood.bic", "likelihood.bicc"),
                  tsmatrix_series = c("sa", "sa_f"))
```

Lancement du cruncher

Pour lancer le cruncher avec `cruncher()` ou `cruncher_and_param()`, il faut spécifier le chemin d'accès au dossier contenant le cruncher (paramètre `cruncher_bin_directory`) ainsi que celui du workspace à traiter (paramètre `workspace`).

Par défaut, le chemin d'accès au dossier du cruncher est celui contenu dans le paramètre `cruncher_bin_directory` : il suffit donc de modifier une seule fois cette option afin qu'elle s'applique à toutes les exécutions du cruncher. Le chemin à indiquer est celui du dossier contenant le fichier *jwsacruncher.bat*, situé dans le dossier "Bin" du dossier d'installation du cruncher. Ainsi, s'il a été installé sous `D:\jdemetra-cli-2.2.3`, le fichier *jwsacruncher.bat* sera présent sous `D:\jdemetra-cli-2.2.3\bin`. Il faut donc modifier l'option `cruncher_bin_directory` de la façon suivante :

```
options(cruncher_bin_directory = "D:/jdemetra-cli-2.2.3/bin/")
```

Si aucun chemin de workspace n'est sélectionné, une fenêtre s'ouvre, invitant à sélectionner le workspace sur lequel on souhaite lancer le cruncher.

La fonction `cruncher_and_param()` permet de créer un fichier temporaire de paramètres via la fonction `create_param_file()`, pour ensuite lancer le cruncher via la fonction `cruncher()`. En plus des paramètres disponibles dans ces deux fonctions, `cruncher_and_param()` permet de renommer le dossier contenant les résultats du workspace afin qu'ils soient égaux aux noms des multi-documents affichés dans l'application JDemetra+ grâce au paramètre `rename_multi_documents` (par défaut `rename_multi_documents = TRUE` : les dossiers sont renommés). Ci-dessous quelques exemples d'utilisation.

```
# La commande suivante permet de mettre à jour le workspace "ipi" présent sous
# D:/Campagne_CVS/ avec l'option de rafraîchissement "lastoutliers". Les autres
# options de lancement du cruncher sont ceux par défaut de la fonction create_param_file().
# En particulier, les paramètres exportés sont ceux des options "default_matrix_item"
# et "default_tsmatrix_series", et les résultats sortent sous D:/Campagne_CVS/Output/.
```

```

cruncher_and_param(workspace = "D:/Campagne_CVS/ipi.xml",
                    rename_multi_documents = FALSE,
                    policy = "lastoutliers")

# Utilisation du paramètre "output" pour changer le dossier contenant les résultats :
cruncher_and_param(workspace = "D:/Campagne_CVS/ipi.xml",
                    output = "D:/Resultats campagne/",
                    rename_multi_documents = FALSE,
                    policy = "lastoutliers")

# Pour modifier les noms des dossiers contenant les sorties afin qu'ils soient égaux
# aux noms des multi-documents affichés dans l'application JDemetra+ il suffit
# d'utiliser le paramètre "rename_multi_documents = TRUE" (valeur par défaut).
# Le paramètre "delete_existing_file = TRUE" permet, lui, de supprimer éventuels
# dossiers existants portant le même nom qu'un des multi-documents.
cruncher_and_param(workspace = "D:/Campagne_CVS/ipi.xml",
                    rename_multi_documents = TRUE,
                    delete_existing_file = TRUE,
                    policy = "lastoutliers")

# Pour voir les autres paramètres de la fonction :
?cruncher_and_param

```