

# JDemetra+ v3.x : les évolutions du logiciel

## Désaisonnalisation avec JDemetra+

Anna Smyk



# Sommaire I

- ① JDemetra+
- ② Les algorithmes d'ajustement saisonnier
- ③ Nouvelles fonctionnalités pour la production
- ④ Retour sur les spécifications en version 3

## Section 1

JDemetra+

# Le logiciel JDemetra+

JDemetra+ est un logiciel open source pour l'analyse des séries temporelles développé dans le cadre du "Centre d'excellence sur les méthodes et outils statistiques (TSACE)" d'Eurostat par la Banque nationale de Belgique avec le soutien de la Bundesbank et de l'Insee.

Il fournit des algorithmes pour :

- l'Ajustement saisonnier
- l'Estimation des tendances
- le Benchmarking et la désagrégation temporelle
- le Nowcasting
- l'Analyse des révisions

Ces algorithmes (implémentés en Java) sont accessibles via une interface graphique (GUI) et/ou directement dans R (packages).

# Evolution de la version 2 à la version 3

Pourquoi une version 3 ?

- re-ingénierie (rendre le logiciel plus modulaire et facile à maintenir)
- ajouter l'ajustement saisonnier de séries haute fréquence (infra-mensuelles)

Une fois cette refonte commencée, la nouvelle version s'est développée dans plusieurs directions :

- Nouveaux algorithmes pour l'ajustement saisonnier, le Benchmarking, le Nowcasting...
- De nouveaux outils pour l'analyse des séries temporelles (accessibles directement, hors algorithme)
- Plus de périodicités acceptées par les algorithmes d'ajustement saisonnier
- Nouvelles fonctionnalités pour la production (de masse) de séries cvs-cjo

Conséquence: Une interface Graphique et un écosystème R étendus

## Section 2

# Les algorithmes d'ajustement saisonnier

# Les algorithmes d'ajustement saisonnier de JDemetra+

Algorithm	Version 2.x		Version 3.x	
	Access in GUI	Access in R	Access in GUI	Access in R
X-13 Arima	yes	RJDemetra	yes	rjd3x13
Tramo-Seats	yes	RJDemetra	yes	rjd3tramoseats
X12plus			yes	rjd3x11plus
STL			yes	rjd3stl
BSM			yes	rjd3sts
SEATS+			upcoming	upcoming

Deux catégories d'algorithmes :

- coeur historique (main): X-13-Arima et Tramo-Seats (améliorés)
- algorithmes supplémentaires en version 3

# Améliorations des algorithmes historiques

## Améliorations X-13-Arima et Tramo-Seats (JD+ historique)

- correction des outliers prise en compte lors de la sélection du schéma de décomposition
- correction ex ante de l'année bissextile ajoutée à Tramo-Seats (comme dans X13)
- sélection automatique des régresseurs de calendrier



# Nouvelles périodicités acceptables pour l'ajustement saisonnier

Pour les données basse fréquence :  $p$  en 2, 3, 4, 6, 12 est admissible dans tous les algorithmes (principaux et supplémentaires)

Dans la version 2, seul Tramo-Seats permettait de traiter toutes ces fréquences, alors que X13-Arima était limité à  $p$  en 2, 4, 12

Les données haute fréquence (infra-mensuelles) ont souvent des périodicités **multiples** et **non entières**. Il faut adapter les algorithmes classiques mentionnés ci-dessus (avec ou sans arrondir les périodicités, par exemple  $p = 365, 25$ )

Les algorithmes de JDemetra + ont des extensions pour les données à haute fréquence.

## Section 3

# Nouvelles fonctionnalités pour la production

# Processus de production de séries cvs-cjo en version 2

Création d'un workspace : structure de données JD+ spécifique (données brutes, paramètres, diagnostics) permettant de générer des résultats (séries cvs)

- l'estimation peut être réalisée par un module de production (le cruncher) de manière entièrement automatique
- les données peuvent être rafraîchies en appliquant une politique de révision spécifique (estimation sous contrainte)
- lisible par l'interface graphique (expertise manuelle des séries sélectionnées)

Cette mise en place largement utilisée présentait des lacunes :

- portabilité (lourdeur du changement de chemin vers les données brutes)
- définition de spécifications série par série à grande échelle (par exemple un ensemble personnalisé d'outliers pré-spécifiées pour la période de la crise Covid).

# Améliorations en v3.x (1/2)

Nouveaux outils R pour manipuler les workspaces, avec des fonctions pour

- changer le chemin vers les données brutes
- personnaliser les spécifications
- fusionner les workspaces par nom de série, comme deux tables de données

Ces fonctions se trouvent dans les packages `rjd3providers` et `rjd3workspace`.

# Améliorations en v3.x (2/2)

Production entièrement en R, sans structure de workspace

- objets TS et flexibilité totale pour la personnalisation des spécifications
- nouvelles fonctions R permettant d'appliquer des politiques de révision (`rjd3×13::refresh` et `rjd3tramoseats::refresh`), avec encore plus de souplesse

Défaut inhérent : les données ne sont pas lisibles par l'interface graphique, ce qui prive d'une visualisation plus élaborée utile pour une mise au point manuelle.

Solution : de nouvelles fonctions R pour créer à la volée des workspaces dynamiques lisibles par l'interface graphique

Dans la version cible 3.x, des algorithmes supplémentaires (X12+, STL+, STS) seront également utilisables en production avec un workspace et un cruncher (sur les données basse fréquence).

## Section 4

Retour sur les spécifications en version 3

# Retour sur les spécifications en version 3

en v2 deux types de spécification: domain spec et point spec

en v3 trois types : domain spec, estimation spec, point spec

- spécification de référence (GUI) ou domain spec : un ensemble global de contraintes à l'intérieur duquel l'estimation sera effectuée (dans GUI définie "à gauche" ou clic droit sur la série et "reference specification")
- estimation spec: contient tous les choix de paramètres résultant de l'estimation (ou modification dans GUI spec window de la série)
- point spec: résultat complet (avec les coefficients..), permettant de retrouver les series estimées à partir de la série brute

# Passage de la v2 à la v3

## Refresh policies

- concurrent revient à la domain spec en effaçant les paramètres user-defined (différence de comportement par rapport à la v2)

par exemple : refresh concurrent efface les outliers pre-spécifiés

Comment sont copiées les specs d'un WS v2 dans un WS v3 ?

- pas d'estimationSpec en v2
- domainSpec dans domainSpec et estimationSpec
- pointSpec dans pointSpec.

Comment écrire dans la domain spec

- GUI : cloner un spec pré-spécifiée (à gauche) ou modif clic droit
- R tools



# Autres Points spécifiques

## Output X11

- D table
- D final table (avec effets du pré-ajustement)

## Import des données

- si WS en v2 et lien vers fichier texte ou csv : changer “date pattern” car seul “DD” (et non “DD” ou “dd”) est valide en v3
- en v3: clean missing est coché par défaut

Correction des effets de calendrier en v3 : nombreuses nouveautés. (voir séquence calendrier)

# A partir de la version 3.5

- détails des series du pré-ajustement : comme en v2 de nouveau
- corrections de bugs sur les user-defined variables (pas sauvées...)
- Possibilité de “nettoyer” un processing déjà exécuté pour le ré-exécuter

Cas d'utilisation : si on change une user-defined variable/calendar (contenu/nom), on ne peut pas voir le nouveau résultat, sauf en changeant manuellement les spec concernées) (voir “re-run”)

- Réexécuter des SAltems déjà calculés (voir edit > reset processing)