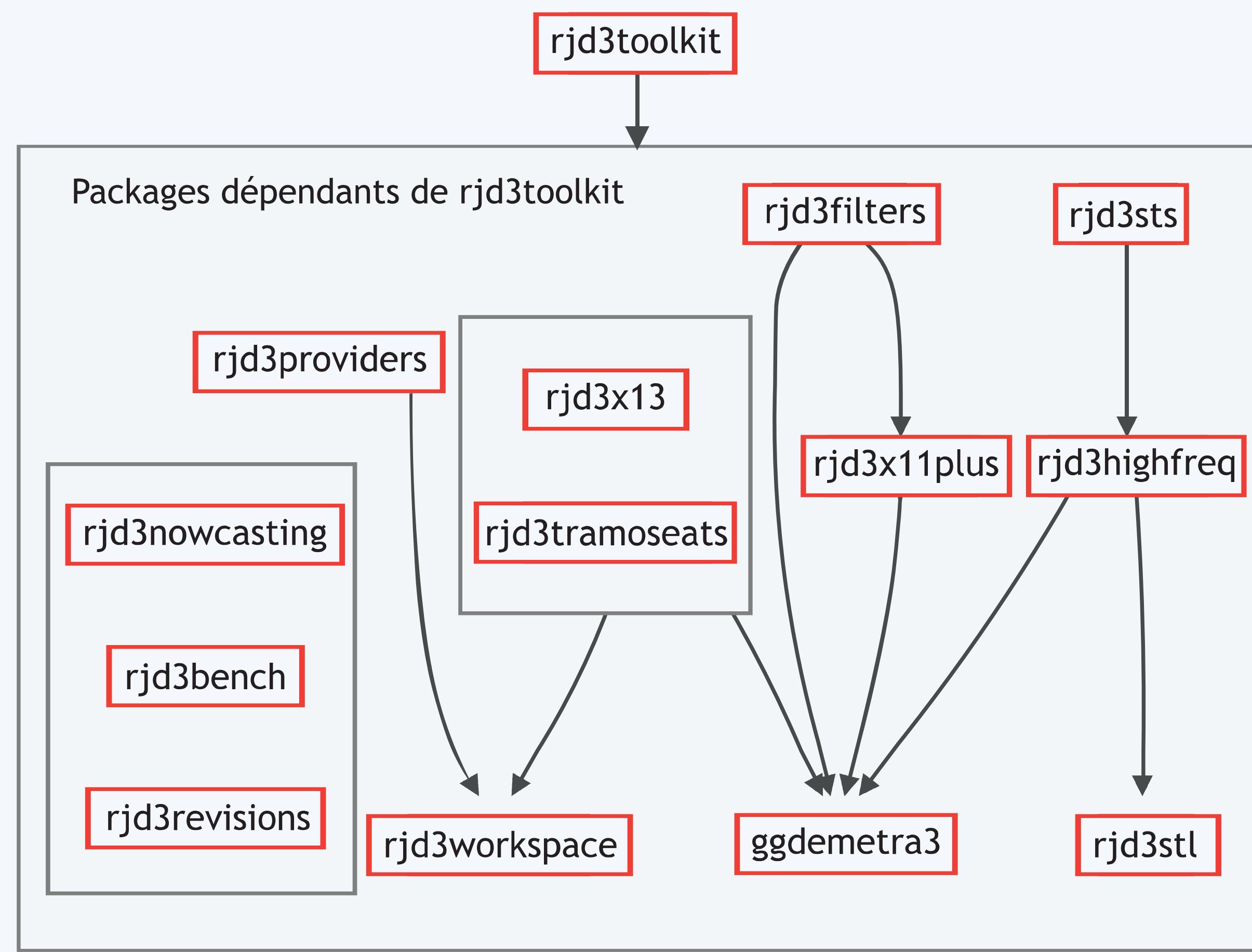
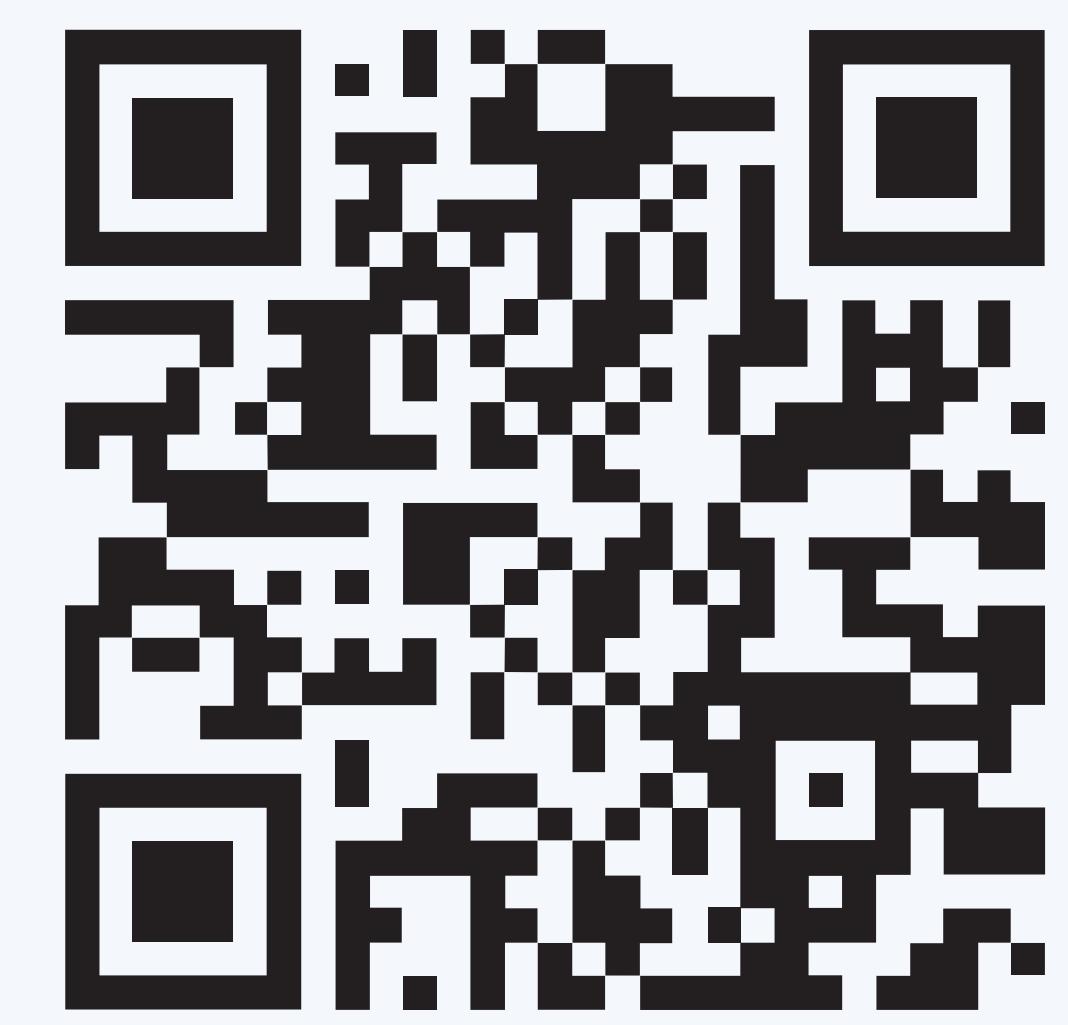




LES PACKAGES R AUTOUR DE JDEMETRA+ : UNE BOÎTE À OUTILS COMPLÈTE POUR L'ANALYSE DES SÉRIES TEMPORELLES

Tanguy BARTHELEMY (Insee)
 TanguyBarthelemy

Anna SMYK (Insee)
 annasmyk



Présentation

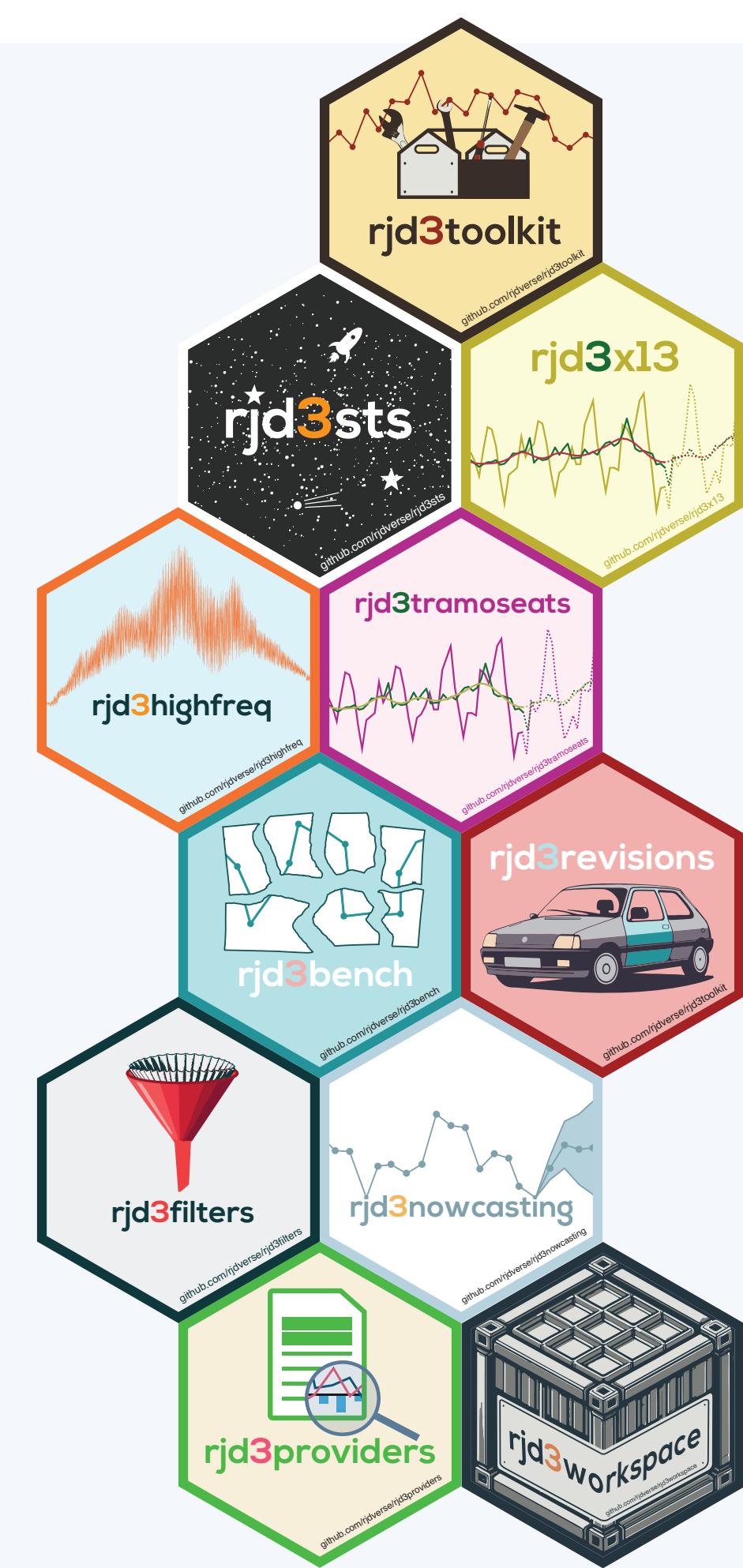
Outils généraux d'analyse de séries temporelles
 {rjd3toolkit} Modélisation et décomposition de modèles SARIMA, tests, génération de regresseurs
 {rjd3sts} Modèles espace-état

Ajustement saisonnier
 {rjd3x13} Pré-ajustement Reg-ARIMA et décomposition X11
 {rjd3tramoseats} Pré-ajustement TRAMO et décomposition SEATS
 {rjd3stl} Décomposition par régression locale (LOESS)
 {rjd3highfreq} Modélisation Airline et décomposition AMB étendues à toutes les périodicités (y.c non entières).
 {rjd3x11plus} Décomposition avec X11 étendue à toutes les périodicités (y.c non entières)

Autres algorithmes
 {rjd3filters} Filtrage et extraction de la tendance-cycle
 {rjd3bench} Benchmarking et désagrégation temporelle
 {rjd3revisions} Analyse des révisions
 {rjd3nowcasting} Nowcasting

Outils liés à l'interface graphique de JDmetra+
 {rjd3providers} Manipulation des données en entrée avec R
 {rjd3workspace} Manipulation de workspaces avec R
 {rjwsacruncher} Lancement automatique de l'ajustement saisonnier (production)

Tous ces packages sont disponibles sur :



{rjd3toolkit}

Création d'un calendrier national



```

french_calendar <- national_calendar()
days = list(
  Bastille_day = fixed_day(
    month = 7,
    day = 14
  ),
  Victory_day = fixed_day(
    month = 5,
    day = 8,
    validity = list(start = "1982-05-08")
  ),
  NEWYEAR = special_day("NEWYEAR"),
  CHRISTMAS = special_day("CHRISTMAS"),
  MAYDAY = special_day("MAYDAY"),
  EASTERMONDAY = special_day("EASTERMONDAY"),
  ASCENSION = special_day("ASCENSION"),
  WHITMONDAY = special_day("WHITMONDAY"),
  ASSUMPTION = special_day("ASSUMPTION"),
  ALLSAINTSDAY = special_day("ALLSAINTSDAY"),
  ARMISTICE = special_day("ARMISTICE")
)

```

Régresseurs mensuels

```

calendar_td(
  calendar = french_calendar,
  frequency = 12L,
  start = c(1990L, 1L),
  length = 480L,
  groups = c(1, 2, 2, 2, 2, 0, 0),
  contrasts = TRUE
)

```

Régresseurs journaliers

```

holidays(
  calendar = french_calendar,
  start = "1968-01-01",
  length = 19359L,
  type = "All",
  nonworking = 7L
)
## NEWYEAR Victory_day MAYDAY ASCENSION ...
## 2018-05-08 0 1 0 0
## 2018-05-09 0 0 0 0
## 2018-05-10 0 0 0 1
## ...

```

Personnalisation des spécifications

```

set_basic()
set_estimate()
set_arima()
set_automodel()
set_tradingdays()
set_outlier()
set_easter()
set_benchmarking()
add_outlier()
add_usrdefvar()
modelling_context()

```

{rjd3x13} et {rjd3tramoseats}

Les packages {rjd3x13} et {rjd3tramoseats} réalisent des désaisonnalisations basées sur les algorithmes x13 (reg-ARIMA et décomposition x11) et TRAMOSEATS. Pour cela on peut utiliser une spécification par défaut ou la personnaliser avec des paramètres préédéfinis.

Spécifications

```

regarima_spec()
x11_spec()
x13_spec()

set_x11()

tramo_spec()
tramoSeats_spec()
set_seats()

```

Ajustement saisonnier

```

regarima()
x11()
x13()

tramo()
tramoSeats()

```

Exemple

```

sa_x13 <- x13(AirPassengers)
sa_tramoSeats <- tramoseats(AirPassengers)

plot(sa_x13)

Y, Sa, trend
Series Trend Seasonally adjusted
```

Désaisonnalisation en deux étapes

Notations :

Y Série brute
 O_i Outliers
 C_j Regresseurs de calendrier
 Y_{lin} Série linéarisée

S Composante saisonnière
 T Composante de tendance
 I Composante irrégulière

Phase de pré-ajustement : modélisation reg-ARIMA

- Estimation des effets de calendrier et détection de valeurs atypiques (outliers)
- modélisation ARIMA(p, d, q)(P, D, Q)

$$Y_{lin} = Y - \sum \hat{\alpha}_i O_i - \sum \hat{\beta}_j C_j$$

$$\phi(B)\phi_s(B)(I - B)^d(I - B^s)^D X_t = \theta(B)\theta_s(B)\epsilon_t$$

Phase de décomposition

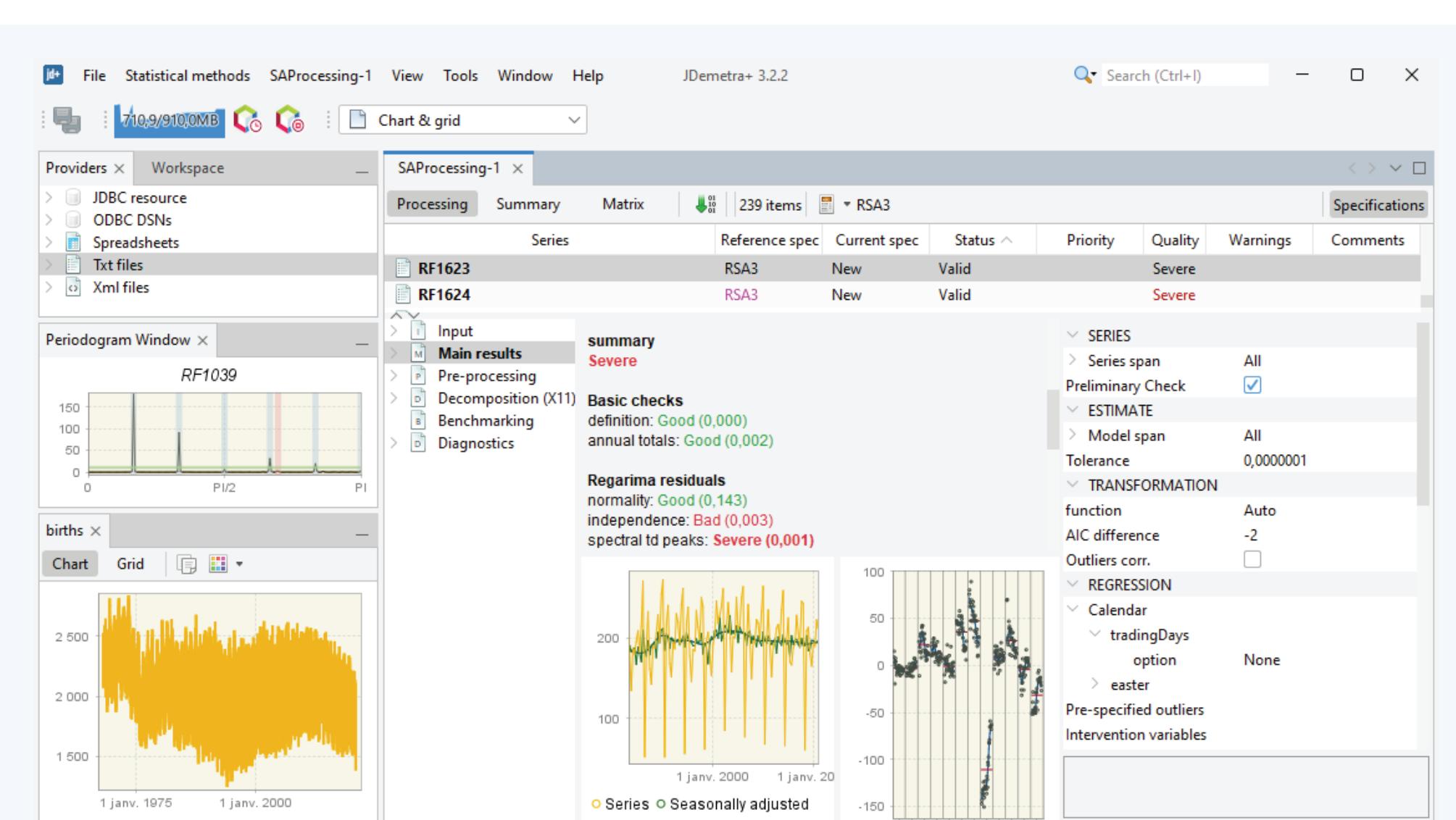
On décompose la série linéarisée (Y_{lin}) en 3 composantes inobservables (S, T et I) pour obtenir la série CVS.

Modèle additif : $Y = S + I + T$

Modèle multiplicatif : $Y = S * I * T$

$$CVS = Y - S = T + I$$

$$CVS = Y/S = T * I$$



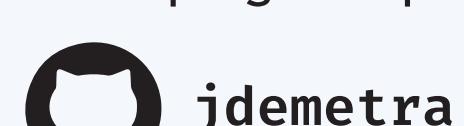
JDmetra+ : une interface graphique polyvalente

JDmetra+ est un outil de désaisonnalisation et d'analyse des séries temporelles. C'est un logiciel libre et open source (FOSS) développé sous la licence EUPL par la Banque nationale de Belgique en collaboration avec la Deutsche Bundesbank, l'Insee et Eurostat conformément aux guidelines du Système statistique européen (SSE).

Depuis 2015, JDmetra+ est officiellement recommandé par Eurostat aux membres du SSE et du Système européen de banques centrales pour la correction des variations saisonnières et calendaires des statistiques officielles.

Techniquement, JDmetra+ est une librairie d'algorithmes écrits en Java, facilement accessibles via une interface graphique (GUI) et des packages R (rjdverse) qui sont une surcouche du code java (voir encadré {RProtoBuf}).

L'ensemble du code java est disponible sous GitHub dans l'organisation jdemetra (<https://github.com/jdemetra>). Vous pouvez y retrouver les projets relatifs à la version 2 et 3 de JDmetra+ ainsi que les plug-ins permettant d'étendre les fonctionnalités de l'interface graphique.



{rjd3highfreq}

Le rjdverse propose aussi des packages dédiés à la désaisonnalisation de données haute-fréquence (infra-mensuelles: hebdomadaires, journalières, horaires). Celles-ci peuvent présenter des périodicités multiples et non entières ce qui exige d'adapter les algorithmes classiques. Elles pourront être traitées avec {rjd3highfreq}, {rjd3x11plus} ou {rjd3sts}.

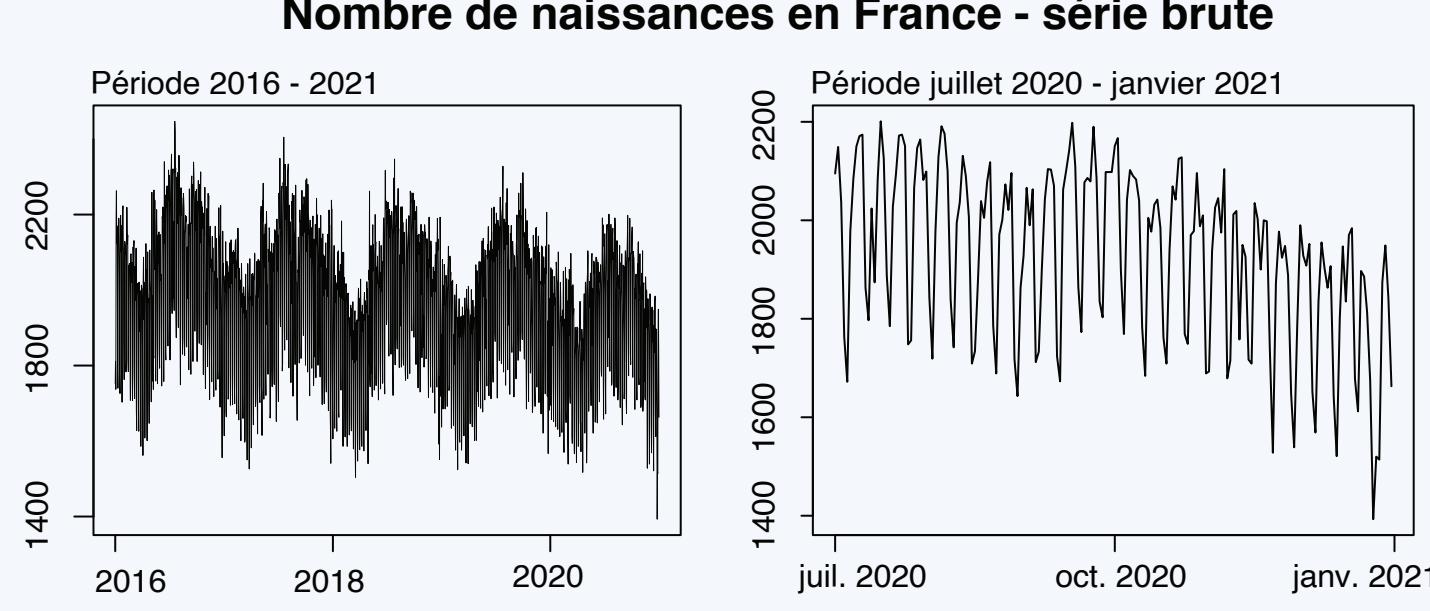
Ajustement saisonnier

Pré-ajustement avec un fractional Airline modèle (ARIMA model Based)

Exemple

Nous travaillons ici avec le nombre de naissances journalières en France. Nous allons extraire les différentes composantes saisonnières pour calculer la série CVS :

Nombre de naissances en France - série brute

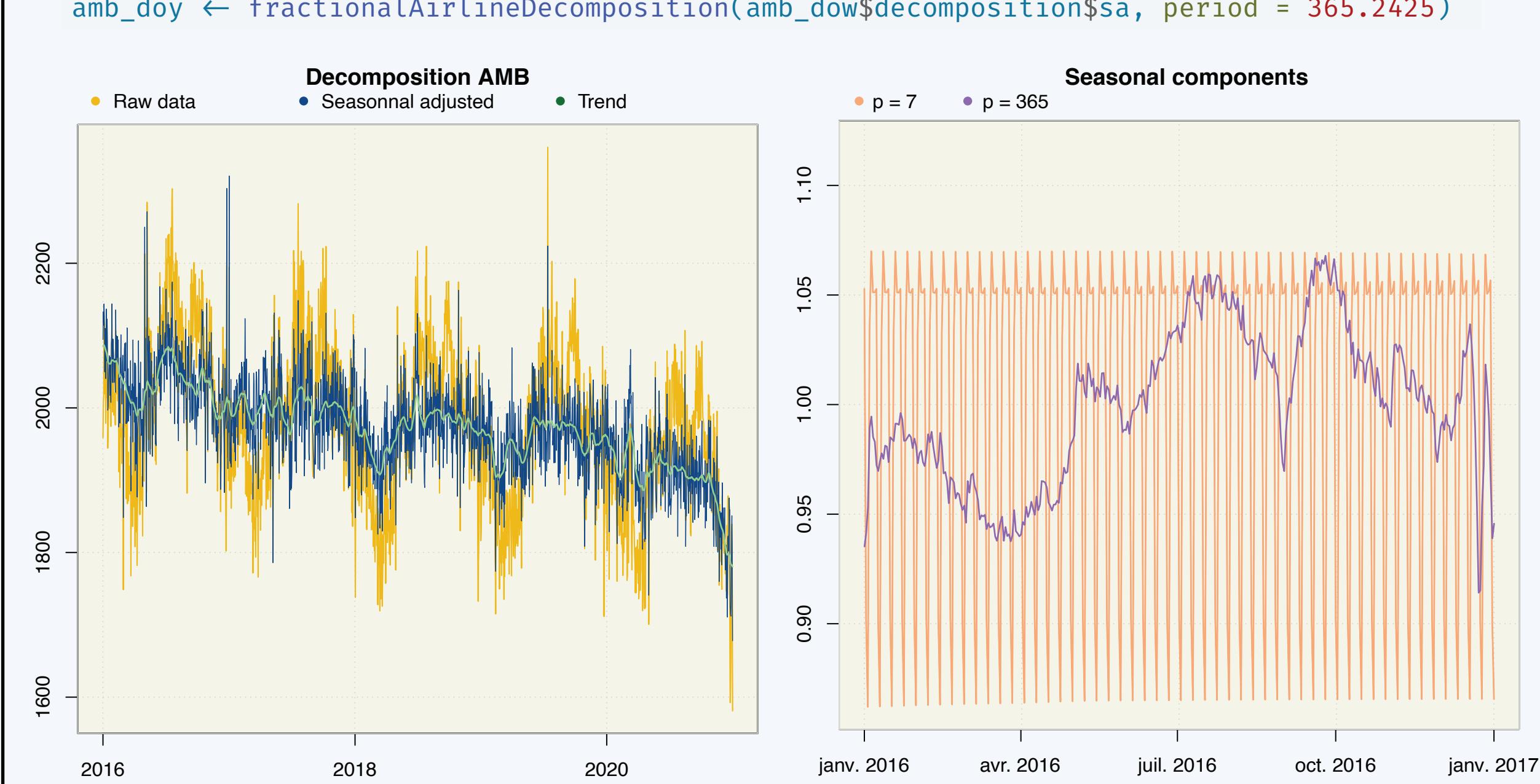


Cette série présente deux périodicités (hebdomadaire $p = 7$ et annuelle $p = 365.25$). Elles seront retirées itérativement en commençant par la plus haute fréquence.

```

pre_ajustement <- fractionalAirlineEstimation(french_births, periods = c(7, 365.2425))
amb_dow <- fractionalAirlineDecomposition(pre_ajustement$model$linearized, period = 7)
amb_doy <- fractionalAirlineDecomposition(amb_dow$decomposition$sa, period = 365.2425)

```



References



<https://jdemetra-new-documentation.netlify.app/>



<https://jdemetra-universe-blog.netlify.app/>



@TSwithJDmetraandR

