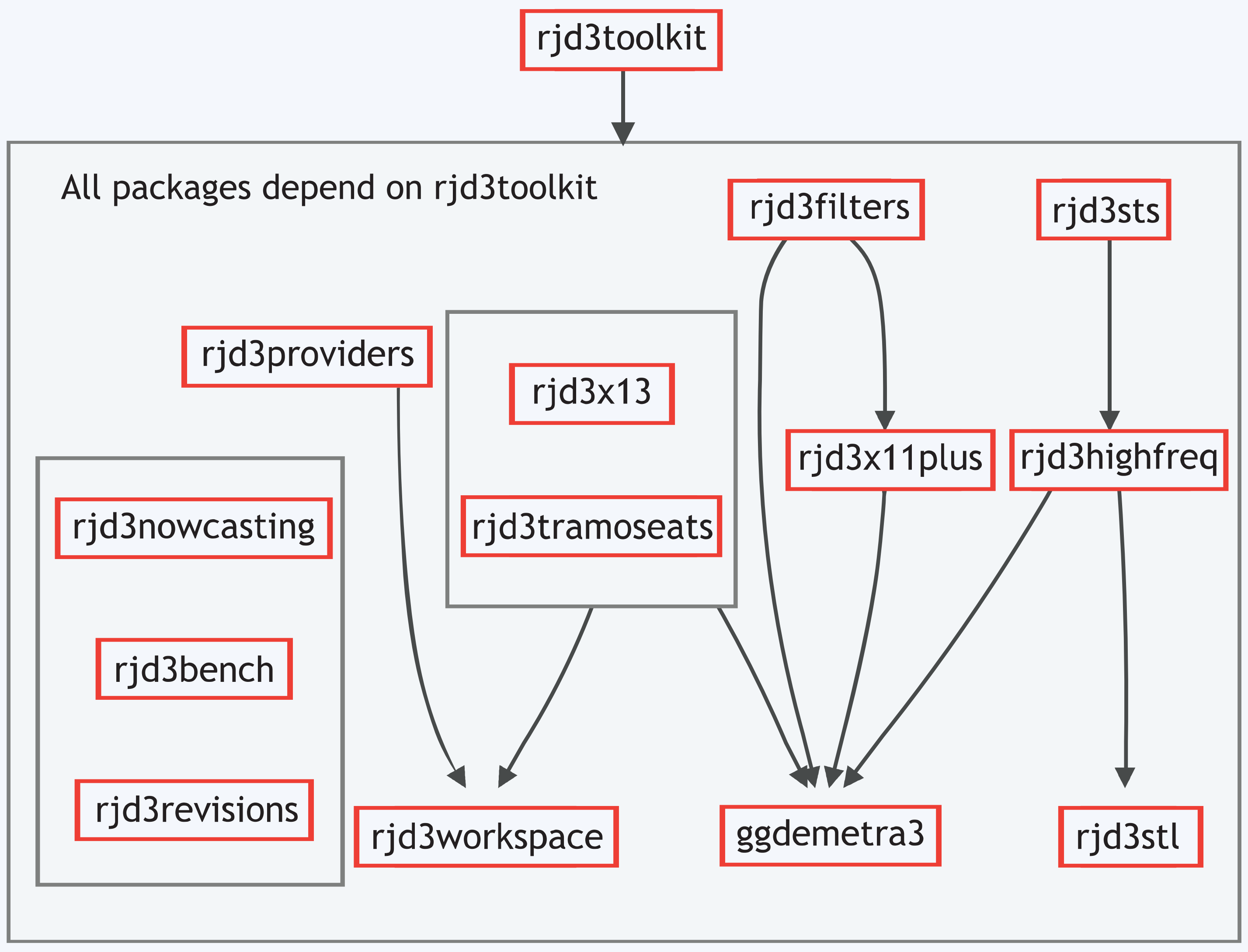
R PACKAGES AROUND JDEMETRA+:
A VERSATILE TOOLBOX FOR TIME SERIES ANALYSIS

Tanguy BARTHELEMY (Insee) TanguyBarthelemy

Anna SMYK (Insee) annasmyk

Time Series Software for Official Statistics

Insee Mesurer pour comprendre



Overview

General purpose tools

{rjd3toolkit}	*SARIMA model modeling and decomposition, tests, regressors generation*
{rjd3sts}	*State-space Framework*

Seasonal adjustment

{rjd3x13}	*Reg-ARIMA preprocessing and X11 decomposition*
{rjd3tramoseats}	*TRAMO preprocessing and SEATS decomposition*
{rjd3stl}	*Local regression decomposition (LOESS)*
{rjd3highfreq}	*Airline modeling and AMB decomposition extended to all periodicities (including non-integer ones)*
{rjd3x11plus}	*Decomposition with X11 extended to all periodicities (including non-integer ones)*

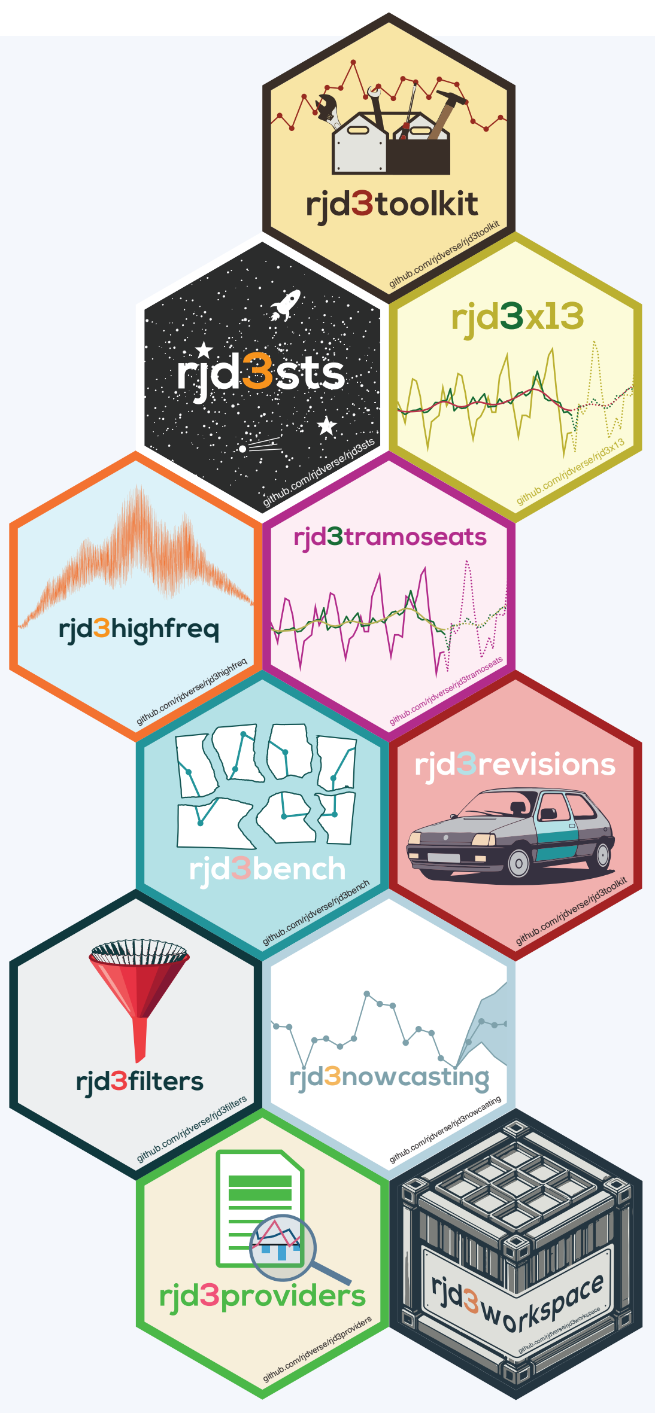
Other algorithms

{rjd3filters}	*Filtering and trend-cycle extraction*
{rjd3bench}	*Benchmarking and temporal disaggregation*
{rjd3revisions}	*Revision analysis*
{rjd3nowcasting}	*Nowcasting*

Tools related to JDemetra+ Graphical User Interface

{rjd3providers}	*Wrangling input data with R*
{rjd3workspace}	*Wrangling workspaces with R*
{rjwsacruncher}	*Automatic launch of seasonal adjustment (production)*

All packages are available on: rOpenSci R-universe   RJD VERSE   rjdverse



{rjd3toolkit}

Creation of a national calendar

```
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")
    )
)
```

Monthly calendar regressors

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

```
##               group_1     group_2
## Jan 1990  0.0000000  2.00000000
## Feb 1990  0.0000000  0.00000000
## Mar 1990 -0.1952313  0.40635829
## Apr 1990  0.1976501 -2.39184549
## May 1990  0.8935244  2.09438426
```

Daily calendar regressors

```
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
##  ...
```

Customizing specifications

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

{rjd3x13} and {rjd3tramoseats}

The {rjd3x13} and {rjd3tramoseats} packages perform seasonal adjustment WITH X13-ARIMA or Tramo-Seats algorithms. We can use a default specification or customize it with user-defined parameters

Specifications

```
regarima_spec()
x11_spec()
x13_spec()

set_x11()

tramo_spec()
tramoseats_spec()
set_seats()
```

Seasonal adjustment

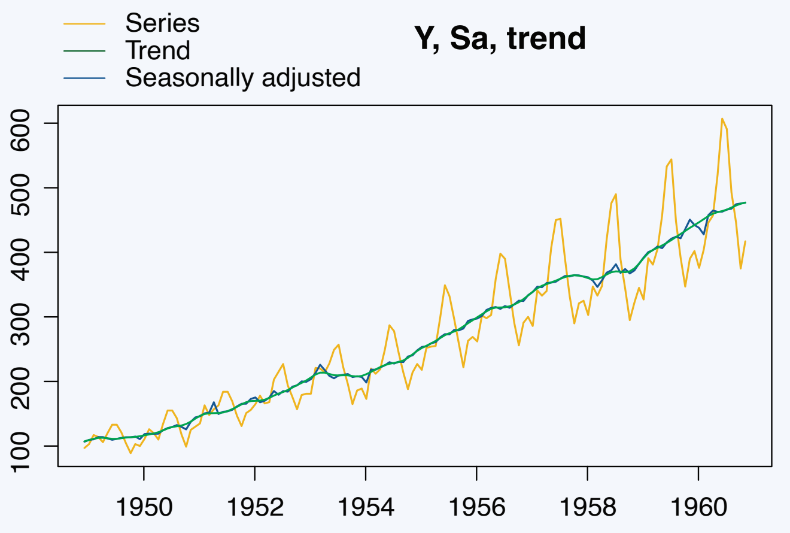
```
regarima()
x11()
x13()

tramo()
tramoseats()
```

Example

```
sa_x13 ← x13(AirPassengers)
sa_tramoseats ← tramoseats(AirPassengers)

plot(sa_x13)
```



Two-stage seasonal adjustment

Notations :

Y Raw series
O_i Outliers
C_j Calendar regressors
Y_{lin} Linearized series

S Seasonal component
T Trend component
I Irregular component

Preprocessing phase: reg-ARIMA modeling

- Estimation of calendar effects and detection of atypical values (outliers)
- $\mathrm{ARIMA}(p, d, q)(P, D, Q)$ modeling

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

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

Decomposition phase

The linearized series (Y_{lin}) is decomposed into 3 unobservable components (S, T et I) to obtain the CVS series.

Additive model: $Y = S + I + T$

Multiplicative model: $Y = S * I * T$

$$\boxed{\begin{aligned} SA = Y - S = T + I \\ SA = Y/S = T * I \end{aligned}}$$

JDemetra+: a user-friendly graphical interface

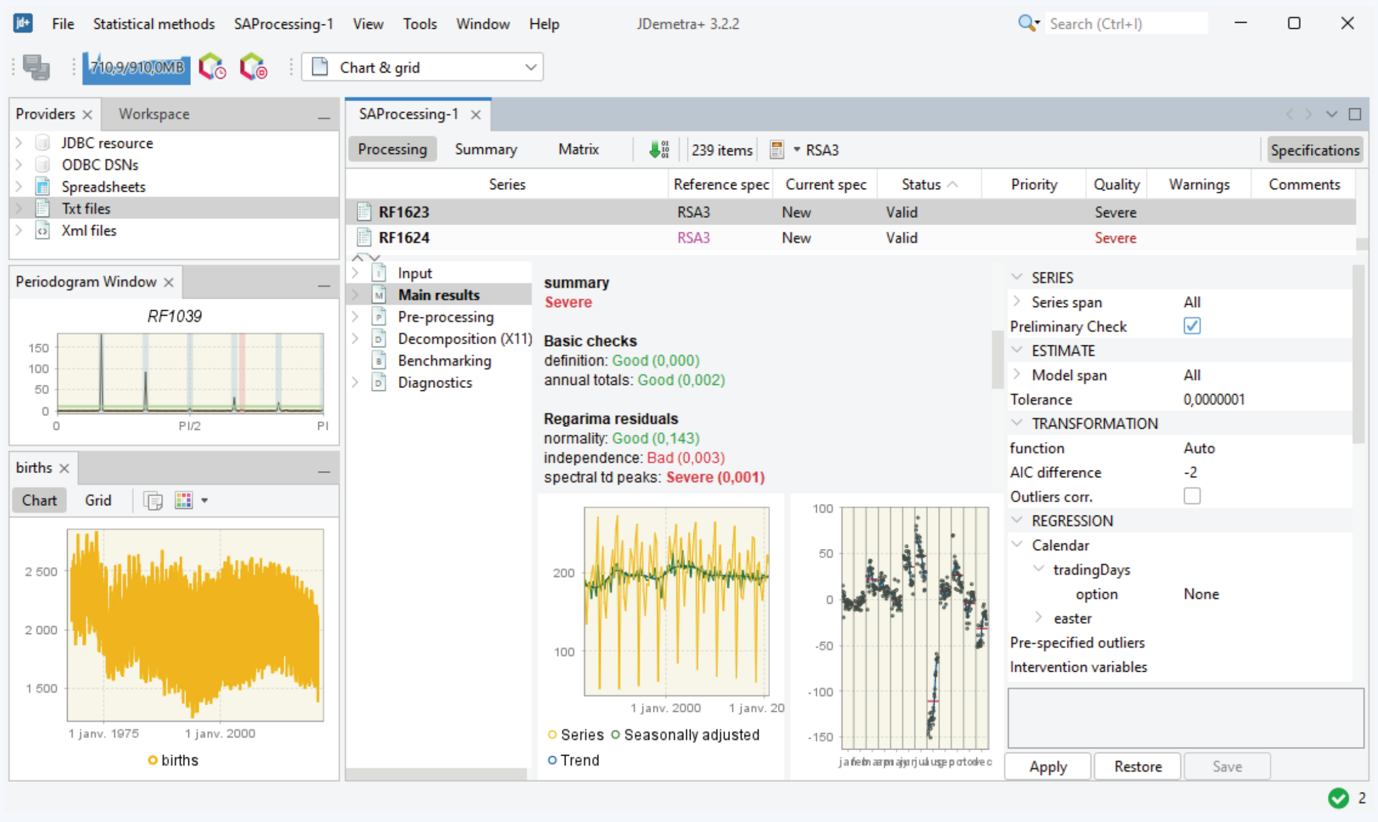
JDemetra+ is a seasonal adjustment and time series analysis tool. It is a free and open source software (FOSS) developed under the EUPL license by the National Bank of Belgium in collaboration with the Deutsche Bundesbank, Insee and Eurostat in accordance with the guidelines of the European Statistical System (ESS).

Since 2015, JDemetra+ has been officially recommended by Eurostat to members of the ESS and the European System of Central Banks for the seasonal and calendar adjustment of official statistics.

Technically, JDemetra+ is a library of algorithms written in Java, easily accessible via a graphical user interface (GUI) and R packages (rjdverse) that overlay the java code (see {RProtoBuf} box).

All java code is available on GitHub in the **jdemetra** organization (*https://github.com/jdemetra*). Here you'll find projects relating to versions 2 and 3 of JDemetra+, as well as plug-ins for extending GUI functionalities.

jdemetra



{rjd3highfreq}

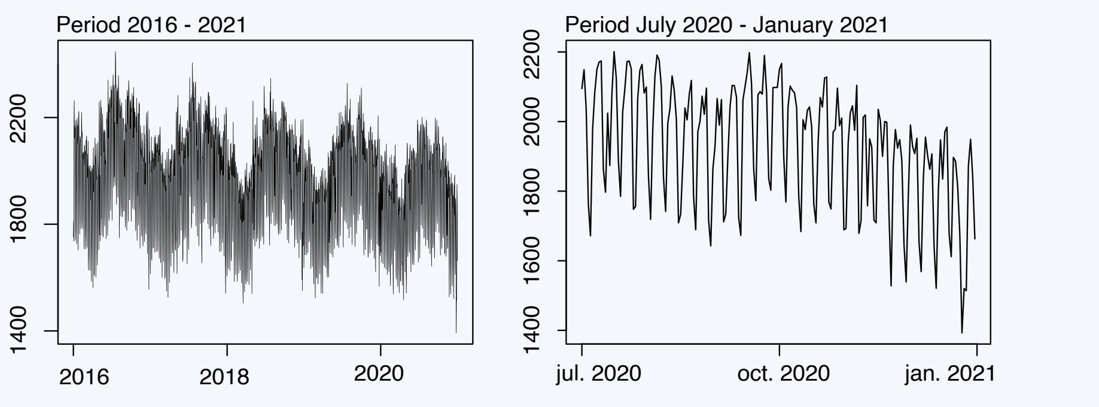
The rjdverse also offers packages dedicated to the seasonal adjustment of high-frequency data (infra-monthly: weekly, daily, hourly...). Such time series may have multiple, non-integer peridocities, which means that conventional algorithms need to be adapted. They can be processed with {rjd3highfreq}, {rjd3x11plus} or {rjd3sts}.

Seasonal adjustment (example)

Preprocessing with a fractional Airline model
AMB decomposition (ARIMA Model Based)

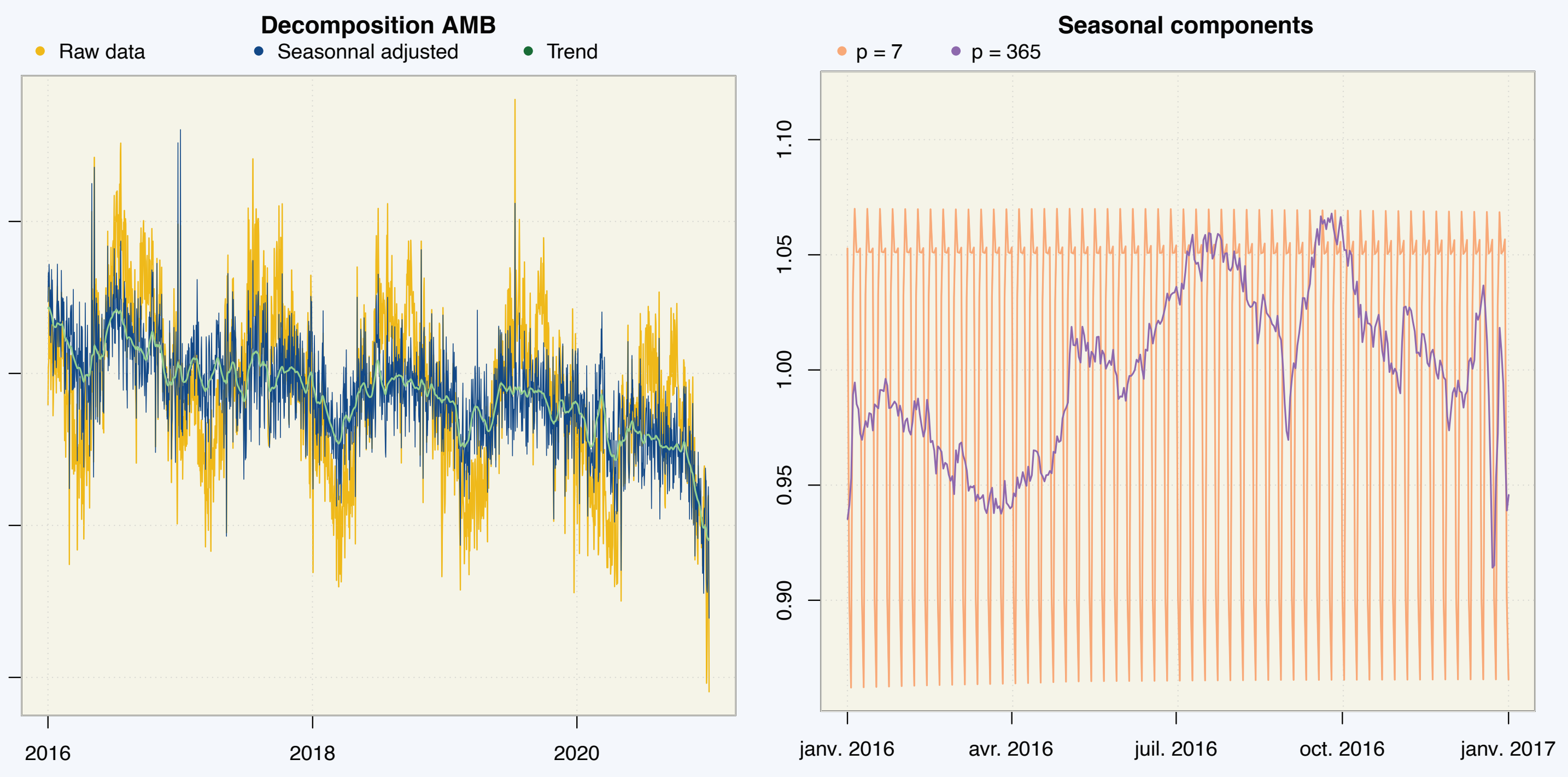
We are working here with the number of daily births in France. We will extract the different seasonal components to compute the SA series:

Number of births in France - raw data



This series has two periodicities (weekly $p = 7$ and annual $p = 365.25$). They will be removed iteratively, starting with the higher frequency.

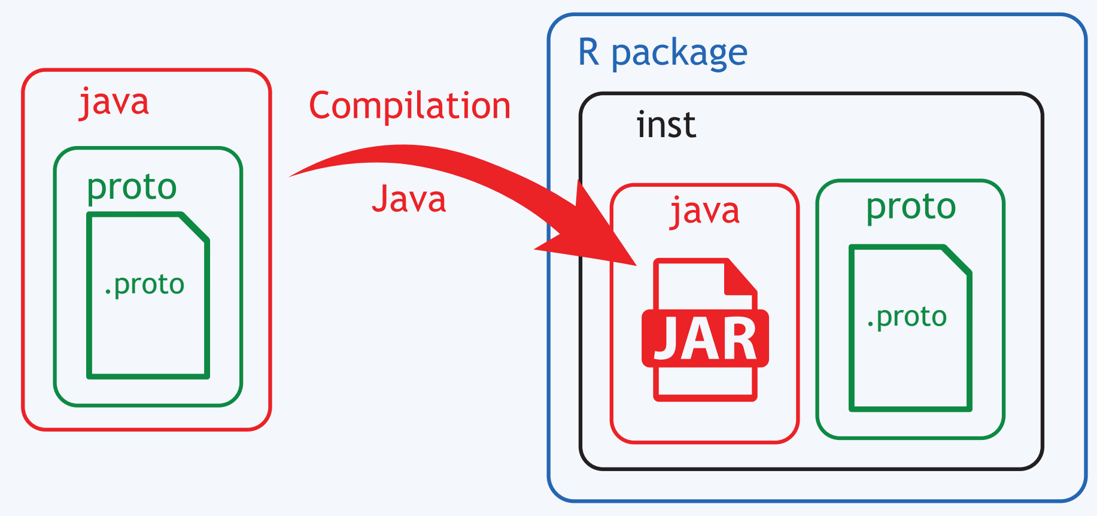
```
pre_ajustement ← fractionalAirlineEstimation(french_births, periods = c(7, 365.2425))
amb_dow ← fractionalAirlineDecomposition(pre_ajustement$model$linearized, period = 7)
amb_doy ← fractionalAirlineDecomposition(amb_dow$decomposition$sa, period = 365.2425)
```



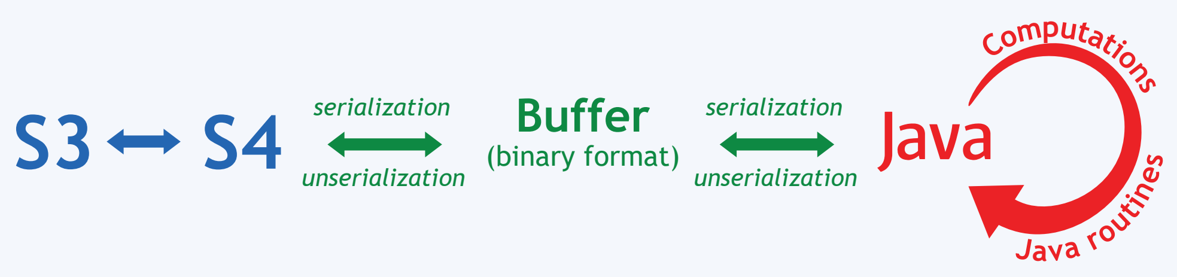
{RProtoBuf}

JDemetra+ version 3 packages use {rJava} and {RProtoBuf} to link Java and R.

Protobuf is a structured data serialization mechanism developed by Google. It is used for communication between services or to store data.



The .proto files define the structure of the objects (in class). In R, {RProtoBuf} converts objects into S4 classes (via autogenerated functions).



References

https://jdemetra-new-documentation.netlify.app/   https://jdemetra-universe-blog.netlify.app/   @TSwithJDemetraandR