JDemetra+ 3.x: New R tools for (high-frequency) time series analysis

Anna Smyk, Tanguy Barthelemy (Insee)

Keywords: *seasonal adjustment, trend estimation, high-frequency data, trading-day correction*

1 Introduction

JDemetra+ is an open source software for time series analysis and seasonal adjustment in particular.. It has been officially recommended by Eurostat to the European Statistical System members since 2015. Its latest version (3.x), progressively released from end of 2022, fills several critical gaps providing extended features for seasonal adjustment and trend estimation, including high frequency data and production tools. JDemetra+ is unique in its combination of very fast java routines, a graphical user interface and an ecosystem of R packages. The graphical interface offers a structured and visual feedback, suitable for refined analysis and training, whereas R tools allow the user to mix the capabilities of JDemetra+ with the versatility of the R world, be it for mathematical functions or data wrangling. The first package allowing to use JDemetra+ core algorithms `{RJDemetra}` R was created in 2018 [1] and since then algorithms have been extended and tools upgraded, giving way to an R ecosystem [2] covering a wider scope.
Our paper highlights some selected new capabilities of JDemetra+: seasonal adjustment of high frequency data, trend estimation, mass production of seasonally adjusted data fully in R and useful general purpose tools.

2 Seasonal adjustment of High-frequency data

2.1 Specificity of Infra Monthly data

High-frequency data refers to series with a frequency higher than monthly: weekly (traffic casualties), daily (births, credit card payments), hourly (electricity consumption). Such data has become ubiquitous in official statistics due to digital transformations that give access to infra-monthly economic data and also to the covid-19 pandemic which increased the demand for timely indicators. It can be seasonal and hence needs to be seasonally adjusted, but displays some peculiarities such as multiple and non integer periodicities which prevent from using traditional algorithms developed mostly for monthly or quarterly data.

	periodicities (number of observations par cycle)			
data	day	week	month	year
quarterly				4
monthly				12
weekly			4.348125	52.1775
daily		7	30.436875	365.2425
hourly	24	168	730.485	8765.82

Figure 1: Multiple Periodicities

2.2 Tailoring Existing Algorithms

JDemetra+ seasonal adjustment algorithms have been augmented to meet this need, offering an enhanced reg-ARIMA pre-treatment model and extended versions of the ARIMA model-based, STL and X-11 seasonal adjustment approaches able to deal with multiple and non-integer periodicities common in high frequency data, as described in Smyk and Webel (2023) [3]. These extensions are accessible through the `{rjd3highfreq}`, `{rjd3x11plus}` and `{rjd3stl}` R packages as well as through the graphical user-interface.

Some key features like extended X-11 to any (fractional) periodicity or fractional airline model are unique to JD+. Fractional periodicities are tackled using a Taylor approximation for the backshift operator:

$$B^{s+\alpha} \cong (1-\alpha)B^s + \alpha B^{s+1}$$

where α is the decimal part of the periodicity.

2.3 Seasonal adjustment of French daily births series

We consider the series of daily french births from 1968 to 2020. Spectral analysis (Canova-Hansen test, available in JDemetra+) shows that two periodicities $p_1 = 7$ and $p_2 = 365.25$ are present.

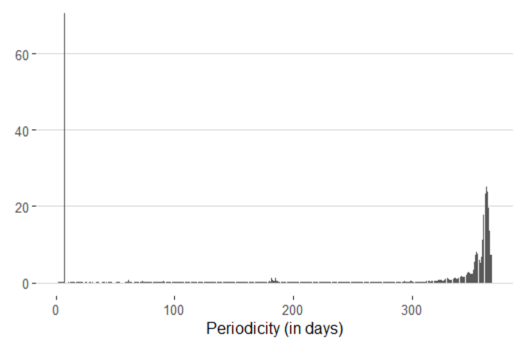

Figure 2: French daily births: estimated seasonal factors , p=7 (black) and p=365.25 (red)

The series is first linearized: outliers are detected and calendar effects removed with the following fractional airline model:

$$(1-B)(1-B^7)(1-B^{365.25})(Y_t - \sum \alpha_i X_{it}) = (1-\theta_1 B)(1-\theta_2 B^7)(1-\theta_3 B^{365.25})\epsilon_t$$

with $1 - B^{365.25} = (1 - 0.75B^{365} - 0.25B^{366})$ and $\epsilon_t \sim \text{NID}(0, \sigma_\epsilon^2)$.

Then a decomposition is performed with extended X-11, using modified filters with the Taylor approximation, which avoids imputing data. Decomposition will be done iteratively periodicity by periodicity starting with the smallest one (highest frequency) as highest frequencies usually display the biggest and most stable variations and cycles of highest frequencies can mix up with lower ones.

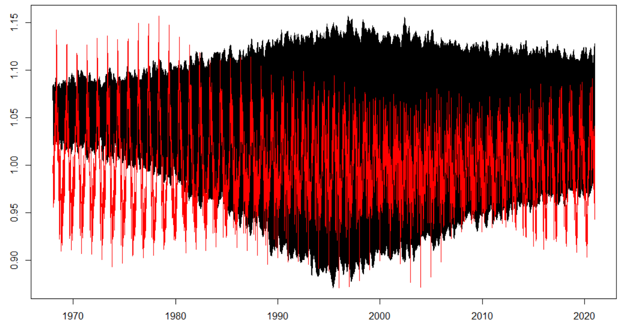

Figure 3: French daily births: estimated seasonal factors , p=7 (black) and p=365.25 (red)

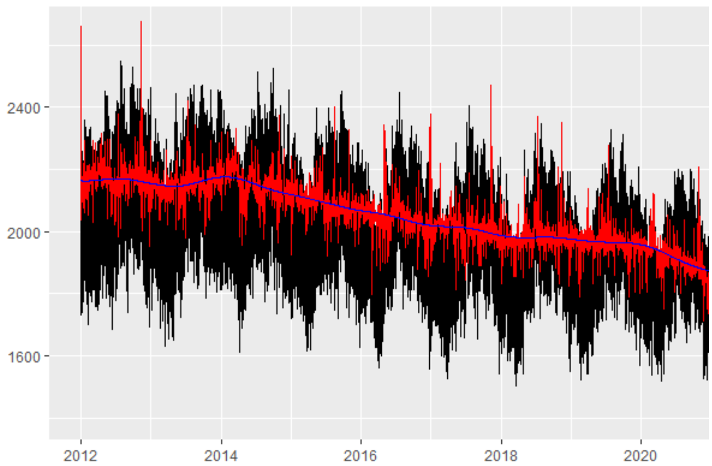

Figure 4: French daily births: raw (black), trend (blue), seasonally adjusted (red)

3 New features for trend estimation

Trend estimation algorithms have been upgraded in the extended x-11 algorithm available in `{rjd3x11plus}` R and as a stand alone feature of the `{rjdfilters}` R package [4].
The refined and final trend-cycle extraction filters of the genuine X-11 method are essentially a set of pre-specified weights for symmetric m-term Henderson filters with $m \in \{3, \ldots, 101\} \cap \mathbb{N}_{\text{odd}}$ and their asymmetric Musgrave surrogates. The extended X-11 approach is founded on applying local polynomial regressions to the input series, as derived in Proietti and Luati (2008) [5].
More algorithms are available via `{rjdfilters}` R such as Reproducing Kernel Hilbert Space (RKHS) filters of Dagum and Bianconcini (2008) [6] with same kernels, FST

filters derived from Grun-Rehomme, Guggemos, and Ladiray (2018) [7] and DFA filters derived from the AST approach of Wildi and McElroy (2019) [8]. This package also provides new functions for building moving average based filters and plotting their properties, which could be used for training purposes.

4 Mass production of seasonally adjusted data in R

4.1 Infra-annual production constraints

R has become ubiquitous in official statistics and the demand for its use in production of seasonally adjusted data is growing fast. JDemetra+ offers the speed and the pre-specified refresh policies recommended by Eurostat Guidelines on SA [9]. A wide range of "partial-concurrent adjustment" options, in which parameters and re-estimated and/ or re-identified gradually have been long available in JDemetra+. But, until now, in the version 2.x family, these options were linked to updating a workspace (specific data structure) via the graphical user interface or more probably via the cruncher (a batch module). It was quite a liability for full production in R as explained in Smyk and Tchang (2021) [10]. Revisions policies are now even more customizable if implemented in R, as time spans on which options are applied can be chosen by the user. Before version 3.0, the user could chose between re-identifying outliers on the whole series span or on the last year of the data (this is the widely applied "partial concurrent last outliers policy"), now the period is customizable, which really makes sense when progressively remodelling the impact of the covid crisis.

4.2 Implementing refresh policies in R

In JD+ v3, the user can re-estimate its current "result_spec" inside a domain of constraints ("estimation_spec"), freeing restrictions on selected parameters, for example :

- Reg-Arima model for pre-adjustment is kept fixed (‘policy="fixed"‘), when new date
- Reg-Arima variables and arima orders are kept identical but outliers are re-identified (‘policy="Outliers"‘)

The code snippet below provides an example using X13-Arima algorithm.

```
library("rjd3x13")

current_result_spec <- sa_x13_v3$result_spec
current_domain_spec <- sa_x13_v3$estimation_spec

# generate NEW spec for refresh
refreshed_spec <- x13_refresh(
    spec = current_result_spec, # point spec to be refreshed
    refspec = current_domain_spec, #domain spec (set of constraints)
    policy = "Outliers",
    period = 12, # monthly series
    start = "2017-01-01",
    end = NULL
)

# apply the new spec on new data : y_new= y_raw + 1 month
sa_x13_v3_refresh <- x13(y_new, refreshed_spec)
```

5 General time series analysis tool

5.1 Generating calendar regressors

The tedious task of generating calendar regressors for quarterly, monthly but also infra-monthly data is now possible using simple R functions of the {rjd3toolkit} R package, for the first time in R.

After defining a calendar, usually composed of national holidays, regressors for calendar correction can be generated as daily dummies (value 1 on a given holiday) or as variables representing the aggregated number of a given type of days for monthly or quarterly frequencies, as shown in the example below. Monthly or quarterly regressors can also be corrected for deterministic seasonality by subtracting a long term mean.

```
library("rjd3toolkit")

# Define a national calendar
frenchCalendar <- national_calendar(
    days = list(
        fixed_day(7, 14), # Bastille Day
        fixed_day(5, 8, validity = list(start = "1982-05-08")), # Victory Day
        special_day('NEWYEAR'),
        special_day('CHRISTMAS'),
        special_day('MAYDAY'),
        special_day('EASTERMONDAY'),
        special_day('ASCENSION'),
        special_day('WHITMONDAY'),
        special_day('ASSUMPTION'),
        special_day('ALLSAINTSDAY'),
        special_day('ARMISTICE'))
)

# Generate daily calendar regressors
q <- holidays(
    calendar = frenchCalendar,
    start = "1968-01-01",
    length = length(my_series),
    type = "All",
    nonworking = 7L
)

# Generate 3 mothly regressors
# group 1: Mondays, Tuesdays, Wednesdays
# group 2: Thusrdays and Fridays
# group 3: Saturdays
# group 0: contrast variable containing Sunadays and holidays
calendar_td(
    calendar = frenchCalendar,
    frequency = 12,
    start = c(1980,1),
    length = 240,
    holiday = 7,
    groups = c(1, 1, 1, 2, 2, 3, 0),
```

```
    contrasts = TRUE
)
```

It is worth noting that JDemetra+ gives access to an extended state-space framework with Basic Structural Models, offering a seasonal adjustment procedure with explicit decomposition and also time varying trading day correction.

5.2 Fast Arima Modeling

JDemetra + provides an Arima estimation function faster than R native 'arima()' function, 17 times faster on the example below.

```
library("rjd3toolkit")
serie <- log(rjd3toolkit::ABS$X0.2.09.10.M)

# JD+
print(system.time(
    for (i in 1:1000) {
        j <- rjd3toolkit::sarima_estimate(
            x = serie,
            order = c(2, 1, 1),
            seasonal = list(order = c(0, 1, 1), period = 12)
        )
    }
))

#     user     system    elapsed    (time in seconds)
#    13.22       0.63       9.84

#R-native
print(system.time(
    for (i in 1:1000) {
        r <- arima(
            x = serie,
            order = c(2, 1, 1),
            seasonal = list(order = c(0, 1, 1), period = 12)
        )
    }
))
#      user     system    elapsed    (time in seconds)
#    225.72       1.72     232.61
```

6 Conclusions

We provided some insight on the capabilities of the latest version of JDemetra+, a software widely used throughout Europe, especially for seasonal adjustment. Currently version (3.x) is still evolving: both the R ecosystem structure (number of packages and their perimeter) as well as more fundamental features undergo changes and enhancements, reported as swiftly as possible in JDemetra+ online documentation [11].

For seasonal adjustment of high-frequency data, further investigations are underway

- Seasonal factor estimation: cubic splines for $p = 365.25$
- Automatic filter selection (X-11, STL)
- Trend-cycle filters: modified I/C ratio? cross validation ? Kernel Parameters ?
- Seasonal filters: Modified I/S ratio? Window length? Spectral approaches?
- Model extensions (pre-treatment and AMB)
    - Arima orders: beyond airline?
    - Fractional periodicities: beyond Taylor approximation ?

References

[1] Jean Palate Alain Quartier-la-Tente Anna Michalek and Raf Baeyens. *RJDemetra: Interface to 'JDemetra+' Seasonal Adjustment Software. R package version 0.1.6.* URL: https://CRAN.R-project.org/package=RJDemetra. (2018).

[2] Jean Palate and al. *R ecosytem for accessing JDemetra+ algorithms in version 3.x.* URL: https://github.com/rjdemetra.

[3] Karsten Webel and Anna Smyk. "Towards Seasonal Adjustment of Infra-Monthly Time Series with JDemetra". In: (2023).

[4] Jean Palate Alain Quartier-la-Tente. *rjdfilters: R package for linear filters.* URL: https://github.com/palatej/rjdfilters.

[5] Tommaso Proietti and Alessandra Luati. "Real time estimation in local polynomial regression, with application to trend-cycle analysis". In: *The Annals of applied statistics* 2.4 (2008), pp. 1523–1553.

[6] Estela Bee Dagum and Silvia Bianconcini. "The Henderson Smoother in Reproducing Kernel Hilbert Space". In: *Journal of Business & Economic Statistics* 26 (2008), pp. 536–545. URL: https://ideas.repec.org/a/bes/jnlbes/v26y2008p536-545.html.

[7] Michel Grun-Rehomme, Fabien Guggemos, and Dominique Ladiray. "Asymmetric Moving Averages Minimizing Phase Shift". In: *Handbook on Seasonal Adjustment* (2018). URL: https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-gq-18-001.

[8] Marc Wildi and Tucker McElroy. "The trilemma between accuracy, timeliness and smoothness in real-time signal extraction". In: *International Journal of Forecasting* 35.3 (2019), pp. 1072–1084. URL: https://EconPapers.repec.org/RePEc:eee:intfor:v:35:y:2019:i:3:p:1072-1084.

[9] Eurostat. *ESS Guidelines on Seasonal Adjustment.* Tech. rep. Eurostat Methodologies and Working Papers, European Commission, 2015. DOI: 10.2785/317290. URL: http://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/KS-GQ-15-001.

[10] A Smyk and A Tchang. *R Tools for Jdemetra+, Seasonal adjustment made easier.* Tech. rep. Institut National de la Statistique et des Etudes Economiques, 2021. URL: https://www.insee.fr/en/statistiques/5019812.

[11] Smyk et al. *JD+ online documentation.* URL: https://jdemetra-new-documentation.netlify.app/. (2023).