Seasonal Adjustment of Infra-Monthly Time Series

Empirical Comparison of (some) Algorithms and Tools

Anna Smyk - Insee

uRos, Bucharest, November 24-26th 2025

Insee
Mesurer pour comprendre

Time Series Software for Official Statistics

Table of Contents I

Motivation

High-frequency or infra-monthly data (weekly, daily, hourly...)

- becomes ubiquitous in official statistics (digital transformations of data collection give access to infra-monthly economic data and covid-19 pandemic outbreak was a demand accelerator)
- can be seasonal and hence needs to be seasonally adjusted, but intrinsic specificities requiring tailored algorithms
- therefore numerous specific algorithms have been developed

Goal of this presentation : compare some of them in the key task for seasonal adjustment : decomposition into unobservable components

..and see how JDemetra+ compares to the state-of-the-art

Infra-yearly periodicities : multiple and non integer (1/2)

High-frequency data can display **multiple** and **non integer** periodicities

	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

Infra-yearly periodicities : multiple and non integer (2/2)

A daily series daily might display 3 periodicities

- weekly ($p = 7$): Mondays are alike and different from Sundays (DOW)
- intra-monthly ($p = 30.44$): the last days of each month are different from the fist ones (DOM), much less common than the previous one
- yearly periodicity ($p = 365.25$) : from on year to another the 15th of June are alike, summer days are alike (DOY)

Decomposition into Unobservable Components

Usual decomposition for seasonal adjustment

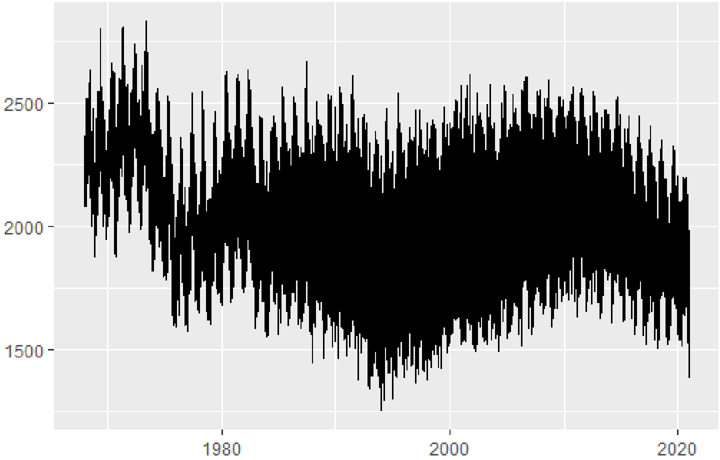
$$Y_t = T_t \circ S_t \circ I_t$$

Modification for a daily series: (iterative) estimation of multiple seasonal factors

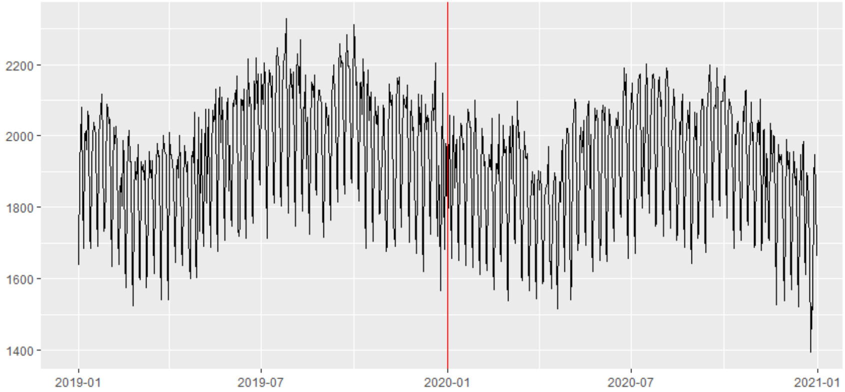
$$S_t = S_{t,7} \circ S_{t,30.44} \circ S_{t,365.25}$$

If decomposition is Additive ($\circ = +$), if multiplicative ($\circ = \times$)

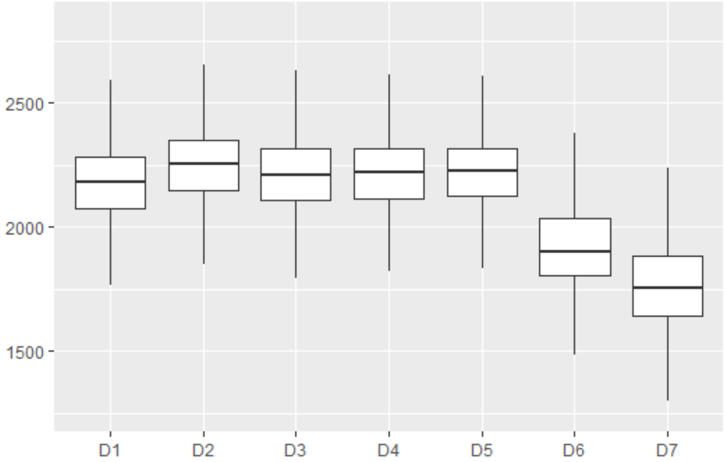
Example: Daily births in France 1968-2020



Example: Daily births in France zoom 2019-2020

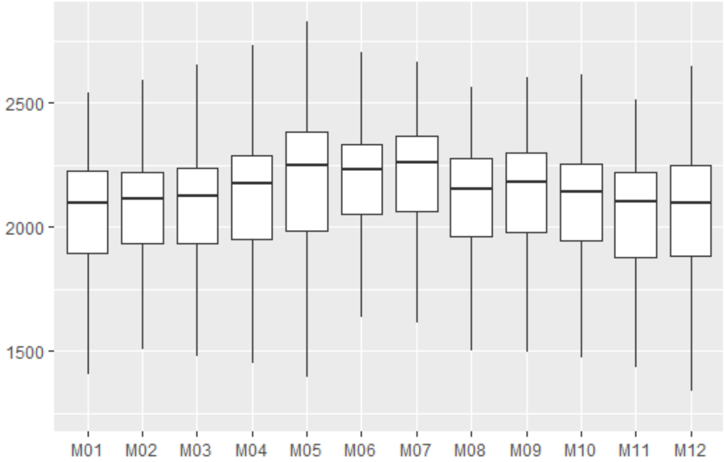


Daily births in France broken down by day of week (1968-2020)



Highlighting weekly periodicity ($p = 7$)

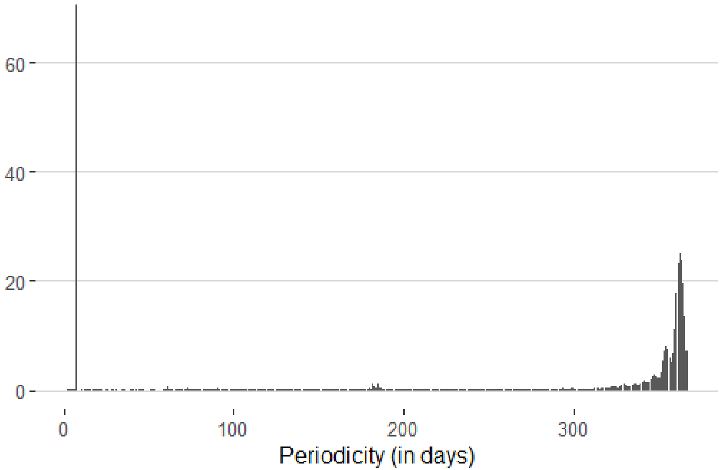
Daily births in France broken down by month (1986-2020)



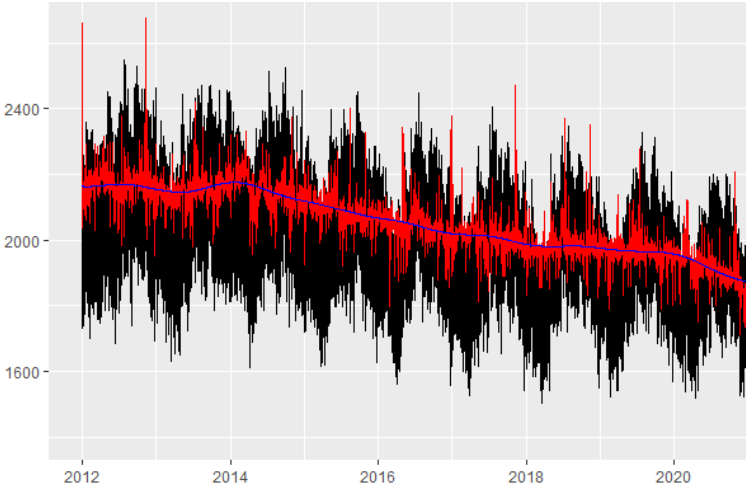
Highlighting yearly periodicity ($p = 365.25$)

Identification of seasonal patterns

Canova-Hansen test allows to identify multiple seasonal patterns



Seasonal Adjustment (SA)



Calendar effects

Structural calendar effects, like in low frequency data, but usually more impactful

- disturb the comparison between two similar periods
- need to be removed before estimating S_7 and $S_{365.25}$, for example
- this is will be modelled as a deterministic effect and corrected by regression in the pre-adjustment phase

Tailoring classic algorithms

Classic seasonal adjustment algorithms, designed for monthly or quarterly data, cannot tackle multiple and fractional periodicities

Several classes of solutions for fractional periodicities :

- use a Taylor approximation for fractional powers of the backshift operators $(B^{s+\alpha} \approx (1-\alpha)B^s + \alpha B^{s+1})$
- use Fourier harmonics
- round periodicities

Decomposition might be done iteratively periodicity by periodicity starting with the smallest one (highest frequency) as:

- highest frequencies usually display the biggest and most stable variations
- cycles of highest frequencies can mix up with lower ones

Seasonal Adjustment Algorithms for High-Frequency data (1/2)

Extended "Tramo-Seats"

- Arima Model Based Decomposition (and Pre-adjustment with outlier detection and calendar correction)
- uses extended Airline Model with fractional powers (backshift operator), iterative or simultaneous decomposition
- available in `rjd3highfreq` R package

Extended X-11

- Moving Average based sequential trend-cycle and seasonal extraction
- fractional powers (backshift operator), kernel-based trend-cycle filters, iterations on multiple seasonal patterns
- available in `rjd3x11plus` R package

Extended STL

- Loess filters based sequential trend-cycle and seasonal extraction
- rounding down fractional periodicities, iterations on multiple seasonal patterns
- two implementations available in `rjd3stl` and in `forecast` R packages

Seasonal Adjustment Algorithms for High-Frequency data (2/2)

TBATS (**T**rigonometric Seasonal representation, **B**ox-Cox transformation, **A**rma disturbances, **T**rend and **S**easonal components)

- trigonometric representation of seasonality is similar to the classical STS model
- available in the R package `forecast`

Prophet: (META) forecasting tool based on a Bayesian modelling approach

- seasonal components estimation relies on stable trigonometric patterns
- can tackle changepoints in trend (piecewise linear)
- has built-in features for calendar correction with dummy regressors

JDemetra+ software overview

Many algorithms (corresponding to `rdj3..` packages) described above have been implemented as extensions of JDemetra+

JDemetra+ is an open source software for time series analysis in official statistics developed in the framework of a Eurostat grant by the National Bank of Belgium with the support of the Bundesbank and Insee.

It provides algorithms for:

- Seasonal Adjustment
- Trend estimation
- Benchmarking and temporal disaggregation
- Nowcasting
- Revision analysis

These algorithms (implemented in Java) can be accessed with a graphical user-interface (GUI) and/or in R packages.

Empirical comparison: decomposition efficiency

- Aforementioned algorithms compared on simulated data
- Criterion: RMSE by component, averaged over the whole data set for a given simulation scenario

$$RMSE = \sqrt{(\frac{1}{n}\sum_{i=1}^{n}(X_t - \hat{X}_t)^2)}$$

- No specific parameter optimization, automatic selection left where possible

Data Simulation Scenarios

We simulate daily time series with additive decomposition pattern (12 years long)

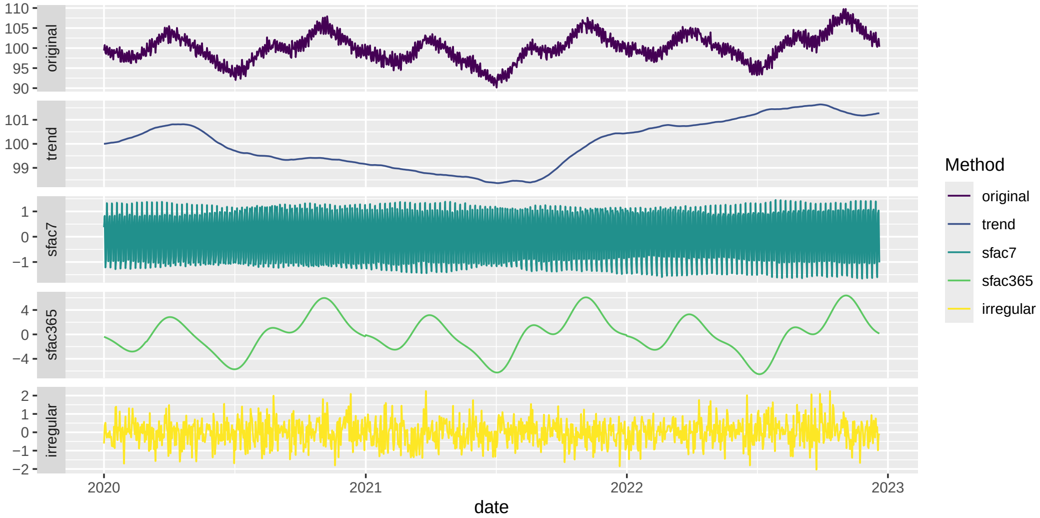
$$X_t = T_t + S_t^W + S_t^Y + \gamma R_t$$

with

- deterministic and stochastic DGP (time varying seasonal components)
- $\gamma \in (0.2, 0.4, 0.6)$, for each DGP
- stochastic strength of seasonality is controlled by $\sigma \in \{0.025, 0.05, 0.075\}$ parameter
- 50 series for each case (300 series in total)

Reference : Bandara and al. (2021), reproduced in `tssim` R package (CRAN)

Simulated data



Results: deterministic process

Average RMSE by component on the whole dataset

γ	Algorithm	Trend	S7	S365	Irregular	SA
0.6	X11	0.047	0.185	0.291	0.351	0.348
0.6	STL	0.063	0.544	0.107	0.559	0.557
0.6	AMB	0.073	0.032	0.137	0.155	0.141
0.6	MAMB	0.053	0.057	0.249	0.272	0.340
0.6	TBATS	0.081	0.023	0.064	0.100	0.069
0.6	PROPHET	0.032	0.022	0.062	0.073	0.066
0.6	MSTL	0.030	0.199	0.199	0.288	0.286

Results: stochastic process

Average RMSE by component on the whole dataset

γ	σ^2	Algorithm	Trend	S7	S365	Irregular	SA
0.6	0.075	X11	0.080	0.191	0.350	0.393	0.400
0.6	0.075	STL	0.172	0.544	0.347	0.618	0.651
0.6	0.075	AMB	0.424	0.119	0.443	0.211	0.460
0.6	0.075	MAMB	0.076	0.153	0.539	0.561	0.576
0.6	0.075	TBATS	0.553	0.269	0.512	0.343	0.609
0.6	0.075	PROPHET	0.089	0.505	0.463	0.689	0.697
0.6	0.075	MSTL	0.060	0.203	0.411	0.460	0.463

Decomposition efficiency

- overall good performance, for SA series less than 1% error
- JDemetra+ extensions results are not far from MSTL, TBATS and Prophet
- model based tools much better at deterministic process, worse with stochastic DGP
- rjd3stl and MSTL perform differently though based on the same algorithm
- from a sheer SA point of view : X11 equal performance in deterministic and stochastic

Computational Efficiency

Average computation time by series

method	Deterministic DGP	Stochastic DGP
X11	0.234	0.235
STL	0.149	0.150
AMB	3.649	3.291
MAMB	3.206	5.819
TBATS	21.547	32.250
PROPHET	2.235	3.310
MSTL	0.036	0.038

- model based approaches (especially with Fourier harmonics) are slower, Prophet is the most efficient among them.
- Model-based times double between deterministic and stochastic processes (convergence times)
- TBATS would be the only algorithm not fit for "mass" production

User-friendliness of the SA process

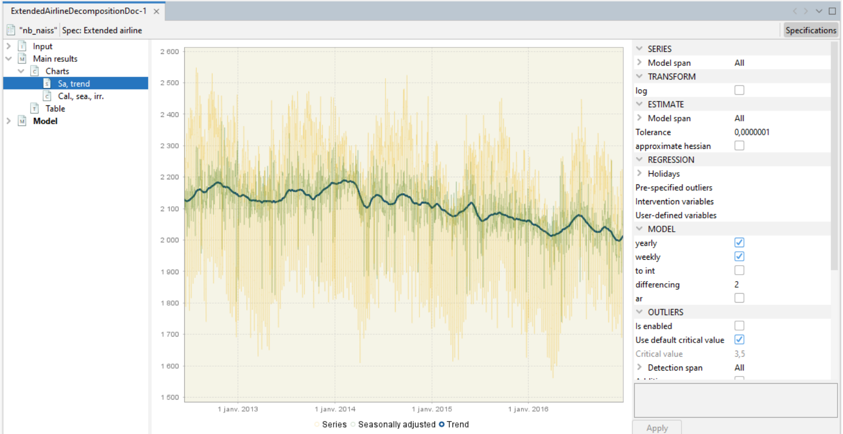
As a spillover of decomposition efficiency comparison, we could assess the user friendliness of the tools: for the seasonal adjustment practitioner what is the distance between a highly integrated process for low frequency data vs high-frequency data

- seasonality identification (test):
  - JDemetra+ extensisons, TBATS et MSTL : to be specified by the user
  - Prophet : $p = 7$ et $p = 365.25$ are tested (if not imposed)
  - $p = 30.44$ can be added everywhere but not in Prophet
- linearization (removing outliers and calendar effects)
  - extended airline model in JDemetra+: all integrated (except log level test)
  - Prophet: integrated calendar correction, but no outlier detection
  - TBATS : all up to the user with external tools
  - decomposition: all integrated automatic parameter selection or default parameter (optimization to be improved for filter selection)
  - testing for residual seasonality: all up to the user with external tool

JDemetra+ Graphical user interface

In JDemetra+, we try to make the overall SA process as automated and user friendly as for monthly or quarterly data.

Results of Extended airline linearization and Decomposition (SEATS) in the Graphical User Interface



Conclusion

Main challenges when seasonally adjusting high-frequency data: multiple and non integer periodicities

Classic algorithms have been (partly) tailored to this purpose, and perform well overall as far as decomposition is concerned

On going investigations around JDemetra+ SA algorithms

- Seasonal factor estimation: cubic splines for $p = 365.25$
- Automatic filter selection (X-11, STL) (now just default values)
  - Trend-cycle filters: modified I/C ratio? cross validation ? Kernel Parameters ?
  - Seasonal filters: Modified I/S ratio? Window length? Spectral approaches?

Limitations: R world, Prophet available in Python

Other algorithms available in R: STR, STD, Ecce Signum...

Other criteria of comparison to investigate: linearization capabilities, revision analysis, forecasting

References and Links

- MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns, International Journal of Operational Research, 2025, Bandara Kasun, Hyndman Rob J and Bergmeir Christoph
- Seasonal Adjustment of Infra-Monthly Time Series with JDemetra+, Journal of Official Statistics vol 40, 2024, Karsten Webel and Anna Smyk
- R Packages giving access to JDemetra+ v3.x (not on CRAN yet): on GitHub https://github.com/rjdverse and on r-universe https://rjdverse.r-universe.dev/packages
- JDemetra+ documentation: https://jdemetra-new-documentation.netlify.app/

- Contact: `anna.smyk@insee.fr` https://github.com/annasmyk