# Benchmarking, Temporal Disaggregation and Reconciliation using R: the `td_R` Package*

Enrique M. Quilis[†‡]

April 16, 2024

**Abstract**

The `td_R` package contains a complete set of R functions designed to perform benchmarking, temporal disaggregation and reconciliation of economic time series using a variety of techniques: methods without indicators; methods with indicators using different approaches: quadratic optimization, static models, dynamic models, ARIMA models; and multivariate methods with indicators and transversal constraints. The package also contains functions for transversal reconciliation: proportional, bi-proportional and restricted least squares.

**Keywords**: Benchmarking, Temporal Aggregation, Temporal Disaggregation, Reconciliation, Statistical Software, Economic Time Series, Time Series Models.

**JEL Codes**: C10, C32, C51, C52, C53, C87.

---

*I am grateful to Ana Abad, Fátima Cardoso, José Ramón Cancelo, Alex Costa, Ángel Cuevas, Tommaso Di Fonzo, Antoni Espasa, Rafael Frutos, Miguel Jerez, François Libeau, Agustín Maravall, Francisco Melis, Leandro Navarro, Julián Pérez, Tommaso Proietti, José Miguel Ramos, Silvia Relloso and Eduardo Salazar for their input at different stages of this project.

[†]Research and Statistics Department, Spanish Tax Agency.

[‡]Email: equilis.work@gmail.com

# Contents

# 1 Introduction

Far away from the idealized textbook setting, statistical practitioners face a plethora of alternative sources of information to perform their tasks. These sources differ in their coverage, compilation procedures, sampling frequency and timeliness, among many other features. Taking into account this environment, it comes as unsurprising the rise of benchmarking as a tool to tame this multiplicity, extracting from each source its best features and combining all of them in a new and improved synthesis. In this realm, two types of benchmarking, temporal disaggregation and transversal reconciliation of economic time series, are especially relevant, due to their key role in the compilation of the Quarterly National Accounts (Eurostat, 2018; IMF, 2018).

As it is the case in other quantitative fields, the availability of appropriate computer software has been instrumental for the spread of benchmarking techniques among practitioners. In this way, the `td_R` package is intended for its use in production mode, easing the tasks of regular data compilation and short-term monitoring, and also for its use in research mode, allowing an in-depth exploration of the estimation results and its internal mechanics.

The `td_R` package contains a complete set of R functions designed to perform temporal disaggregation of economic time series using a variety of techniques: methods without indicators (Boot et al., 1967; Stram and Wei, 1986 and low-pass interpolation inspired by Sims, 1974); methods with indicators using different approaches: quadratic optimization (Denton, 1971), static models (Chow and Lin, 1971; Fernández, 1981 and Litterman, 1983), dynamic models (Santos-Cardoso, 2001 and Proietti, 2006), ARIMA models (Guerrero, 1990); and multivariate methods with indicators and transversal constraints (Denton, 1971; Rossi, 1982 and Di Fonzo, 1990). The package also contains functions for reconciliation using alternative procedures: proportional, RAS bi-proportional (Bacharach, 1965) and constrained least squares (van der Ploeg, 1982, 1985).

Apart from the specific papers above mentioned, the general theoretical background of the methods can be found in Di Fonzo (1987), Cardoso (1999) and Dagum and Cholette (2006), among others. A comprehensive and updated analysis of temporal disaggregation, benchmarking and balancing can be found in Chen et al. (2018a) and the papers cited therein. Due to its close relationship with the procedures included in this package, we should mention Abad and Quilis (2005), Bisio and Moauro (2018), Chen et al. (2018b), Daalmans (2018), Guerrero and Corona (2018), Quilis (2018) and Temursho

(2018).

The `td_R` package draws heavily on `td`, its Matlab ancestor (Quilis, 2019). The migration has been a nice opportunity to think again about benchmarking and all that jazz, learning on the way some new tricks. `td_R` is mostly self-contained, using only some functions provided by the packages `pracma` (Borchers et al., 2023) and `signal` (signal developers, 2023) for certain specific computations (e.g. generalized Moore-Penrose inverse, ARMA impulse response function). Alternative software implementations of temporal disaggregation methods can be found in Barcellan and Mazzi (1995), Barcellan et al. (2003), Doan (2008), Sax and Steiner (2013), Grudkowska (2015) and Casals et al. (2016), among others.

# 2 Temporal Aggregation

The best way to understand temporal disaggregation is understanding temporal aggregation in the first place. As we will see, temporal aggregation is a wolf in sheep's clothing, making temporal disaggregation not its direct inverse problem, but a risky (albeit fascinating!) endeavor, fraught with informational gaps and identification problems (Tiao, 1972; Wei, 1990).

We will review temporal aggregation using three different approaches. First, we will present it in matrix form. Then, we will explore temporal aggregation as the combination of a low-pass filter (the moving sum filter) and a systematic sampler. Finally, we will consider temporal aggregation as a time-varying accumulator.

## 2.1 Temporal Aggregation in Matrix Form

Let $y : nx1$ be a high frequency time series (e.g., quarterly). The corresponding temporal aggregate can be defined as:

$$Y = Cy, \tag{1}$$

where $C$ is the temporal aggregation–extrapolation matrix defined as:

$$C = \left[ \begin{array}{c|c} I_N \otimes c & 0_{N,n-sN} \end{array} \right], \tag{2}$$

where $N$ is the number of low frequency observations and $\otimes$ stands for the Kronecker product. $c$ is a row vector of size $s$, which defines the type of temporal aggregation, and $s$ is the number of high frequency data points for each low frequency data point (frequency conversion ratio). Depending on $c$, three cases may arise as follows:

- Aggregation: $c = [1, 1, \ldots, 1]$ (e.g., flow).
- Averaging: $c = [1/s, 1/s, \ldots, 1/s]$, (e.g., index).
- Interpolation: $c = [0, 0, \ldots, 1]$, (e.g., final stock) or $c = [1, 0, \ldots, 0]$, (e.g., initial stock).

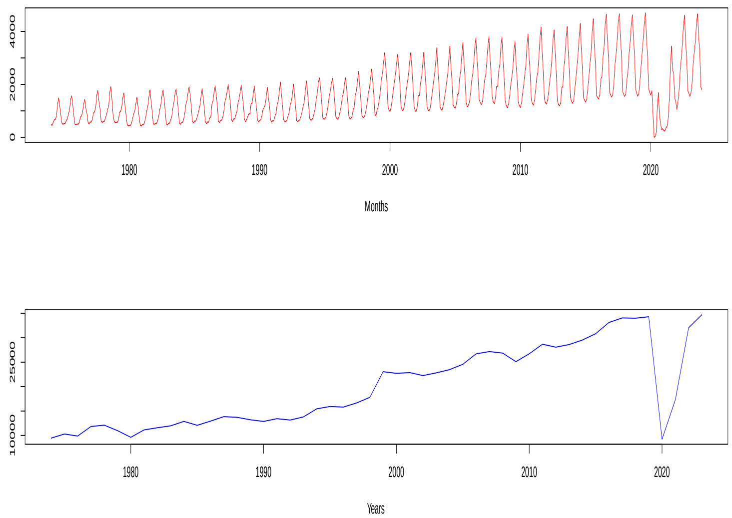
The mechanics of (1) and (2) imply an irreversible compression since $N < n$ and the $C$ matrix is not invertible. Hence, we cannot simply invert it to recover $y$ from $Y$.

We define extrapolation as the case in which the size of the indicators' sample is not equal to the one directly implied by the low frequency series ($n > sN$). In this case, the constraint (2) is not binding for the last $(n - sN)$ elements of $y$. The next R script shows an example of temporal aggregation, being $z$ the monthly *Number of overnight stays in hotels in Spain*, 1974:1 - 2023:12.

```
# Temporal aggregation:  sum
ta <- 1
# Frequency conversion:  quarter -> year
sc <- 12
# Calling function
Z <- temporal_agg(z, ta, sc)
```

The next figure shows the original monthly indicator and its annual counterpart:

Figure 1: *Number of overnight stays in hotels in Spain: temporal aggregation*




## 2.2 Temporal Aggregation as the Combination of a Low-pass Filter and a Systematic Sampler

Temporal aggregation is applied continuously in a routine fashion, because it only requires simple operations (sums or averages). But hidden among these innocent operations is a low-pass filter (smoother) combined with a comb band-pass filter (deseasonalizer). We can see this clearly when considering temporal aggregation as the result of applying systematic sampling to a moving average of the high frequency time series (Melis, 1990).

$$y_t^{(sc)} = y_t + y_{t-1} + \ldots + y_{t-sc-1} = (1 + B + \ldots B^{sc-1})y_t = U_{sc-1}(B)y_t, \quad (3)$$

Applying the aggregation matrix (2) in interpolation mode ($ta = 3$) to $y^s$ we get the corresponding temporal aggregate:

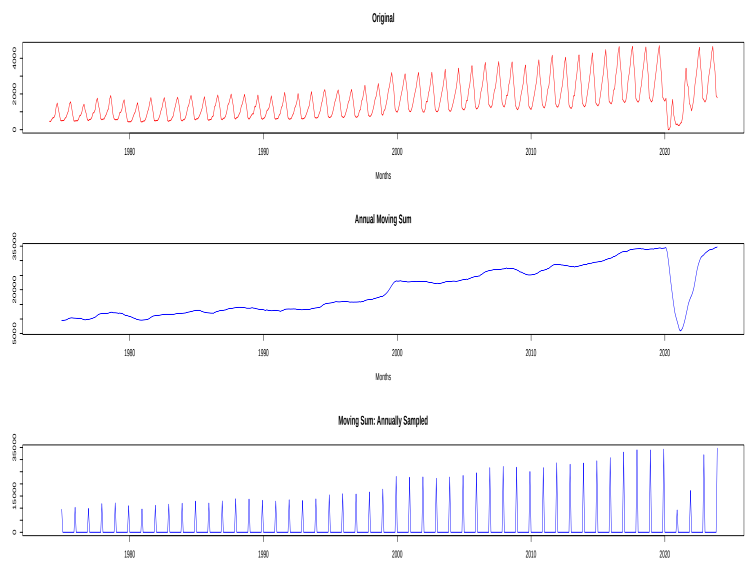
$$Y = Cy^{(sc)}. \quad (4)$$

The combination of (3) and (4) is presented in the next R script:

```
# Temporal aggregation:  sum
ta <- 1
# Frequency conversion:  quarter -> year
sc <- 12
# Calling functions
zs <- moving_sum(z, ta, sc)
zss <- systematic_sampler(zs, sc)
```

Note that the function `systematic_sampler()` is equivalent to (4) but preserves the input dimension, $n$. The high frequency time series, its annual moving sum and the sampled time series at a rate $sc$ are presented in the next graph:

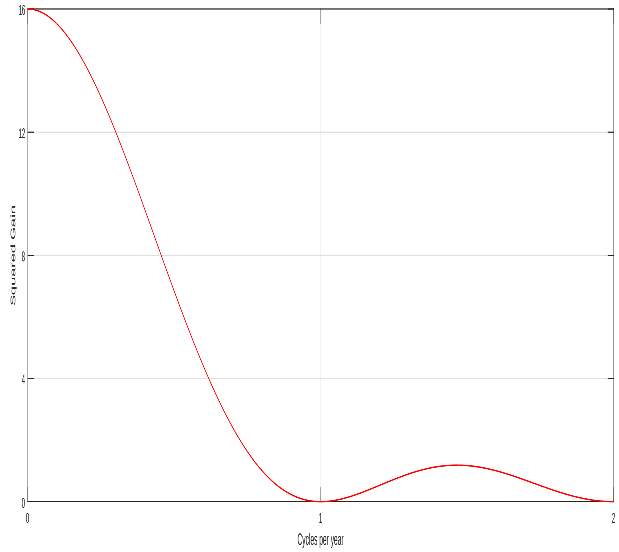
Figure 2: *Number of overnight stays in hotels in Spain: smoothing and systematic sampling*




The $U_{sc-1}(B)$ filter operates as the combination of a low-pass filtering (smoother) and a comb band-pass filter (deseasonalizer), as its gain function shows clearly[1]:

[1]We consider here a quarter to annual conversion rate: $sc = 4$. The reasoning is the same for other cases, such as $sc = 12$.

Figure 3: *$U_3(B)$: Gain function*




Note the amplification of the gain around $w = 0$ (trend) and the zeroes at the seasonal frequencies $w = \pi/2$ (4 quarters=1 cycle per year) and $w = \pi$ (2 quarters=2 cycle per year).

## 2.3 Temporal Aggregation by Means of a Time-varying Accumulator

Finally, an alternative way to consider temporal aggregation is by means of an accumulator loop that collapses according to the sample conversion parameter $sc$:

$$y_t^c = \rho_{t-1} y_{t-1}^c + y_t \qquad\qquad t = 1..n, \tag{5}$$

being $\rho_t$ a binary index that indicates the collapse of the accumulator:

$$\rho_t = \begin{cases} 0 & t = 1 + sc \cdot (T-1) \qquad T = 1 \ldots N \\ 1 & \qquad\qquad otherwise \end{cases} \tag{6}$$

The corresponding R script is as follows:

```
# Temporal aggregation:  sum
ta <- 1
# Frequency conversion:  month -> year
sc <- 12
# Calling functions
zc <- temporal_acc(z, ta,sc)
zcc <- temporal_acc_loop(z,ta,sc)
```

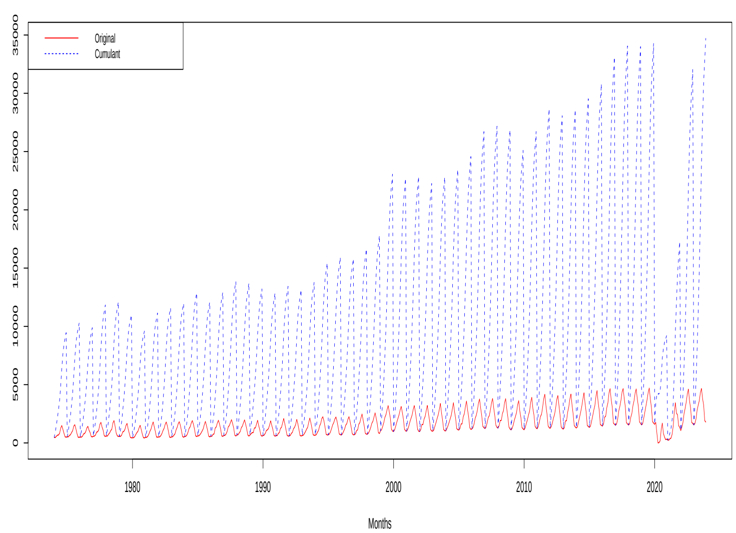
Note that the function `temporal_acc()` performs the same computations as `temporal_acc_loop()`, but using a matrix form to represent (5) and (6):

$$AA = [I_N \otimes A]\,, \tag{7}$$

being $A$ a lower triangular matrix of ones with dimension *sc*.

The high frequency indicator and its accumulated version are presented in the next graph:

Figure 4: *Number of overnight stays in hotels in Spain: temporal accumulation*




The next table provides a list of the main functions related to temporal aggregation.

Table 1: Main functions for temporal aggregation

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| acc | A | ta, n, sc | Temporal accumulation matrix |
| aggreg | C | ta, N, sc | Temporal aggregation matrix |
| moving_sum | zs | z, ta, sc | Moving sum of *sc* periods |
| moving_sum_matrix | S | h, n | Moving sum: basic matrix |
| systematic_sampler | zs | z, sc, codeNA | Systematic sampling |
| temporal_acc | zc | z, ta, sc | Cumulative time series |
| temporal_acc_loop | zc | z, ta, sc | Cumulative time series, looped version |
| temporal_agg | Z | z, ta, sc | Temporal aggregation |

# 3 Temporal Disaggregation

Temporal disaggregation is a special case of benchmarking in which two sources of information, $Y$ and $x$, differing in their sampling frequencies, are combined in a hierarchical way. On the one hand, the low frequency input $Y$ is the absolute benchmark, to which the combined output must adapt: $Y = Cy$. On the other hand, the high frequency input $x$ acts as an interpolation basis, transferring its sampling frequency and its high frequency features (e.g. seasonality, irregularity) to the combined output, $y$. The high frequency input $x$ operates also as an extrapolation basis for the combined output, $y$.

The `td_R` package considers several alternative methods to perform temporal disaggregation. Most of them can be considered as special cases of a general equation that relates the unobservable high frequency interpoland $y$ with the observable high frequency interpolator $x$:

$$(1 - \phi B) y_t = \alpha + (\beta + \gamma B) x_t + \frac{1}{(1 - \rho B)(1 - \mu B)} u_t. \tag{8}$$

The next table defines the role of each parameter and its theoretical range.

Table 2: Parameters of the high frequency model

| y | x | | | $u^2$ | | |
|---|---|---|---|---|---|---|
| Own dynamics | Intercept | Static | Dynamic | First order | Second order | Variance |
| $\phi$ | $\alpha$ | $\beta$ | $\gamma$ | $\rho$ | $\mu$ | $v_u$ |
| [0, 1) | $(-\infty, \infty)$ | | | (-1, 1] | (-1, 1) | $(0, \infty)$ |

The model is completed by means of a temporal aggregation constraint that links $y$ with its observed, low frequency counterpart $Y$. Of course, this constraint is precisely [1]-[2]:

$$Y = Cy = \left[ \begin{array}{c|c} I_N \otimes c & 0_{N,n-sN} \end{array} \right] y. \tag{9}$$

## 3.1 Quadratic Optimization Methods without Indicator

When the information set is composed only by the low frequency benchmark $Y$, we have several methods to perform temporal disaggregation in a univariate way. The method proposed by Boot et al. (1967) -BFL- is a natural starting point due to its well-defined structure and its ease of implementation. The BFL method is derived as the solution for a Quadratic Optimization (QO) program. The BFL method has also a model-based interpretation, as a very restricted case of (8):

Table 3: High frequency model: BFL

| y | x | | | u | |
|---|---|---|---|---|---|
| Own dynamics | Intercept | Static | Dynamic | First order | Second order |
| $\phi = 0$ | $\alpha$=0 | $\beta = 0$ | $\gamma = 0$ | $\rho = 1$ | $\mu = 0$ |

The corresponding high frequency model can be generalized by considering $d \geq 1$ instead of just $d = 1$:

$$y_t = \frac{1}{(1-B)^d} u_t. \tag{10}$$

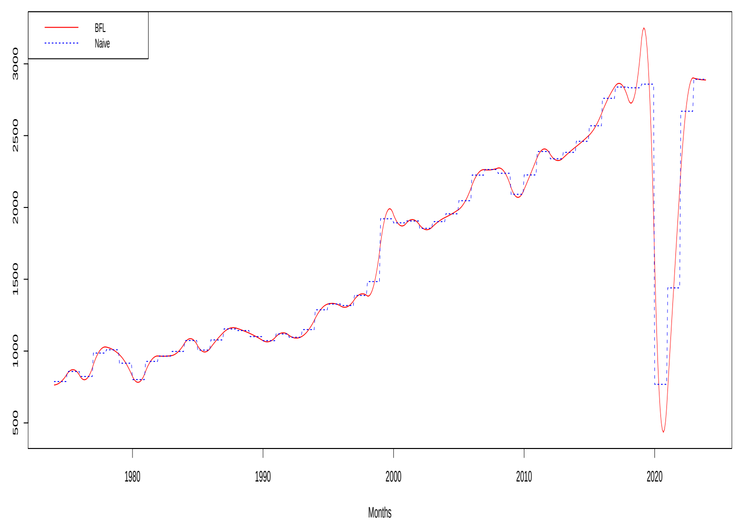
The BFL method can be implemented using `td_R` as follows, using the annual aggregate of the number of overnight stays in hotels in Spain, previously analyzed.

```
# Calling the function
res <- bfl(Z,
ta = 1, # Type of aggregation
d = 1, # Number of unit roots (high frequency time series)
sc = 12)# Frequency conversion
```

[2]In many practical applications, the range is restricted to [0,1) for the first ($\rho$) and second order ($\mu$) parameters of the innovation.

The next graph compares the naive estimate (obtained simply dividing by 12) with the one provided by BFL.

Figure 5: *Boot-Feibes-Lisman method: high frequency estimate*




Note how the BFL estimate preserves the level of the annual benchmark, avoiding spurious breaks when the year changes and the absence of any seasonal pattern in the monthly estimate, thus confirming the irreversible nature of temporal aggregation.

The BFL method assumes that the underlying, unobservable high frequency time series $y$ evolves according to an $I(d)$ model, as represented in (10). Stram and Wei (1986) consider a more general ARIMA(p,d,q) model, thus extending the BFL approach:

$$y_t = \frac{\theta_q(B)}{\phi_p(B)(1-B)^d} u_t. \tag{11}$$

The price to pay for this generalization is that we have to feed the Stram-Wei function with an estimate of the stationary variance-covariance (VCV) matrix of the (unobserved) high frequency time series. This VCV matrix cannot be estimated directly, so a process through its low frequency counterpart is necessary, see Stram and Wei (1986b). Once $v$ has been determined, the `td_R` package allows the application of this method by means of the `stram_wei()`function. In fact, this function is invoked by the `bfl()` function to operate.

Finally, temporal disaggregation without high frequency trackers can be considered as a sort of interpolation applied to a moving sum. This function is reminiscent of Sims (1974), combining non-informative interpolation with low-pass filtering, see also Wei (1990). This approach is encapsulated in the `low_pass_interpolation()` function. The procedure has three steps:

- Raw interpolation: padding the low frequency benchmark with zeros and scaling it.
- Low-pass smoothing by means of the Hodrick-Prescott filter[3].
- Enforcing consistency with the low frequency counterpart by means of benchmarking, using the Denton procedure (additive variant)[4].

The next table provides a list of the main functions related to temporal disaggregation in an univariate context.

Table 4: Main functions for temporal disaggregation without indicators

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| bfl | res | Z, ta, sc | Temporal disaggregation using BFL |
| stram_wei | res | Z, ta, sc, v | Temporal disaggregation using Stram-Wei |
| copy_low | z | Z, ta, sc | Adapts a low frequency time series to a high frequency format |
| low_pass_interpolation | z | Z, ta, sc | Adapts a low frequency time series to a high frequency format |

In most of the cases, the results are stored in a list called res, whose internal structure is reported in Appendix A.

## 3.2 Quadratic Optimization Methods with Indicator

The QO approach followed by BFL and Stram-Wei can be easily extended to the case where the information set also includes a high frequency tracker $x$. This is precisely the approach followed by Denton (1971), see also Cholette (1984) and Di Fonzo and Marini (2012).

[3] Or any other low-pass Butterworth filter.

[4] See the next subsection.

Table 5: High frequency model: Denton

| y | x | | | u | |
|---|---|---|---|---|---|
| Own dynamics | Intercept | Static | Dynamic | First order | Second order |
| $\phi = 0$ | $\alpha$=0 | $\beta = 1$ | $\gamma = 0$ | $\rho = 1$ | $\mu = 0$ |

Again, note that the model considered by Denton can be enlarged in the same way as BFL, by considering $d \geq 1$ instead of just $d = 1$:

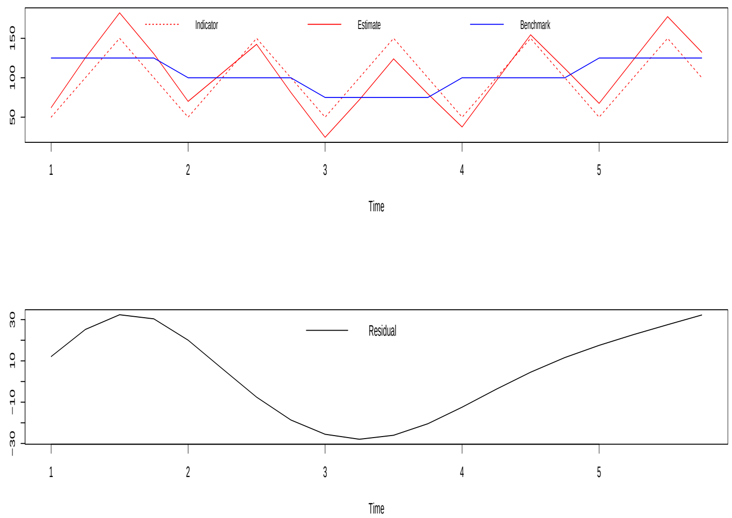
$$y_t = x_t + \frac{1}{(1-B)^d} u_t. \tag{12}$$

Of course, the BFL method can be considered as the limiting case of the Denton method when $x = 0$. The next box shows how to apply Denton's method, using the same data as in his paper.

```
res <- denton(Y,
x, # High frequency tracker
ta = 1, # Type of aggregation
d = 2, # Minimizing the volatility of the d-differenced series
sc = 4, # Frequency conversion
op1 = 1 # Variant:  1=additive, 2=proportional
)
```

The output from this code is presented in the next graph:

Figure 6: *Denton method: estimates, benchmark and residuals*




The next table provides a description of the `denton()` function:

Table 6: Functions for temporal disaggregation with indicators: dynamic

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| denton | res | Y, x, ta, sc, opl | Denton method |

The method of Denton has a proportional variant that circumvents the (implicit) constraint $\beta = 1$. This variant provides a lot of flexibility to the method, thus explaining the central role that it has in some official guidelines (IMF, 2018).

## 3.3 Best Linear Unbiased Esimation with Indicators: Static Models

The package `td_R` includes a set of functions that implement a model-based approach to temporal disaggregation, using as inputs one or several high frequency trackers. All of them confine the dynamics in the innovation term $u$: AR(1) (Chow and Lin, 1971), I(1) (Fernández, 1981) and ARI(1,1) (Litterman, 1983). The `chow_lin()` and `litterman()` functions consider both the Maximum Likelihood (ML) approach suggested by Bournay and Laroque (1979) and the Weighted Least Squares (WLS) proposal of Barbone et al. (1981). The high frequency equation for these methods is presented in the next table.

Table 7: High frequency model: Chow-Lin, Fernández and Litterman

<table>
<tr><td>y</td><td>y</td><td colspan="3">x</td><td colspan="2">u</td></tr>
<tr><td>Method</td><td>Own dynamics</td><td>Intercept</td><td>Static</td><td>Dynamic</td><td>First order</td><td>Second order</td></tr>
<tr><td>Chow-Lin</td><td rowspan="3">$\phi = 0$</td><td rowspan="3">$\alpha$</td><td rowspan="3">$\beta$</td><td rowspan="3">$\gamma = 0$</td><td>$\rho \in [0, 1)$</td><td rowspan="2">$\mu = 0$</td></tr>
<tr><td>Fernández</td><td>$\rho = 1$</td></tr>
<tr><td>Litterman</td><td>$\rho = 1$</td><td>$\mu \in (0, 1)$</td></tr>
</table>

The Fernández method can be considered as the limiting case of the Litterman method when $\mu = 0$ or the limiting case of Chow-Lin when $\rho = 1$. In this way, the Fernández method plays an intermediate role between the methods of Chow-Lin and Litterman. Reinforcing the central role of the Fernández method, note that the Denton procedure is a special case when $\alpha = 0$ and $\rho = \beta = 1$ in equation (13).

The following table provides a description of the main functions.

Table 8: Functions for temporal disaggregation with indicators: static

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| chow_lin | res | Y, x, ta, sc, type_estim, opC, rl | Chow-Lin method |
| fernandez | res | Y, x, ta, sc, opC | Fernández method |
| litterman | res | Y, x, ta, sc, type_estim, opC, rl | Litterman method |

The Chow-Lin relies on the following equation for the high frequency model:

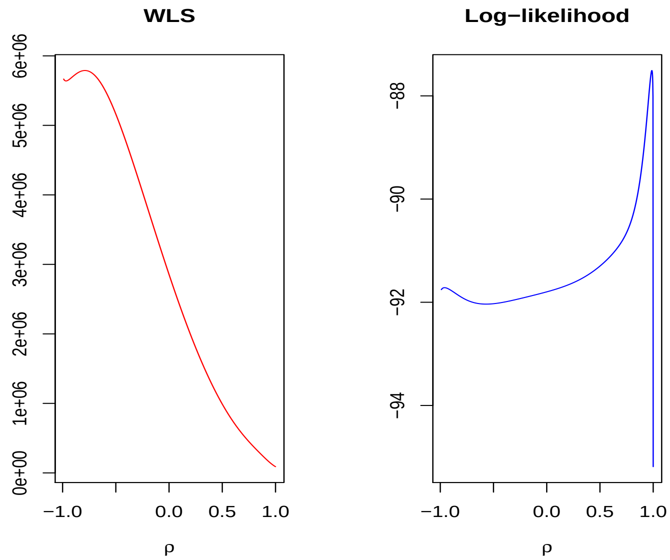
$$y_t = \alpha + \beta x_t + \frac{1}{(1 - \rho B)} u_t. \tag{13}$$

It can be implemented as follows:

```
# Type of aggregation
ta <- 1
# Frequency conversion
sc <- 4
# Method of estimation
type_estim <- 1
# Intercept
opC <- 1
# Interval of grid search for rho
# rl <- NULL # Default:  search on [0.05 0.99] with 100 grid points
# rl <- 0.75 # Fixed value
rl <- c(-.99, 0.999999999, 1000) # Specific range:  min, max and
number of grid points
# Calling the function
res <- chow_lin(Y, x, ta, sc, type_estim, opC, rl)
```

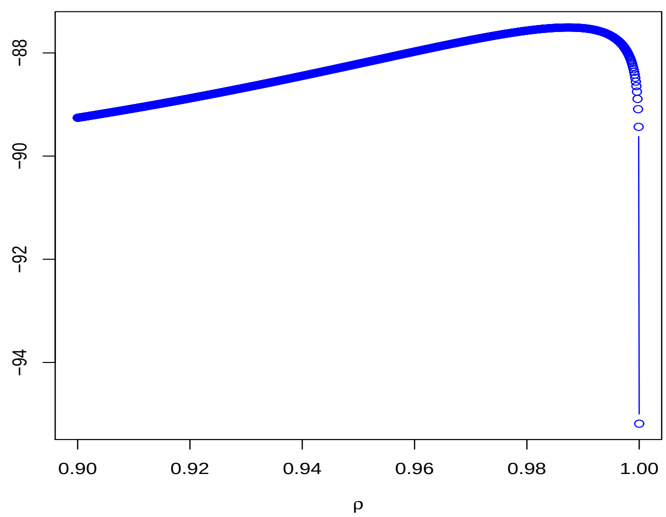
The output of two alternative objective functions, the sum of Weighted Least Squares (WLS) and the Log-Likelihood, are drawn in the range of search, using the same data as Bournay and Laroque (1979): $Y$ = Output. Textile industries at 1956 prices. Unit: FF, $x$ = Index of Industrial Production. Textile industries. Unit: base 1952=100. The sample runs from 1949:Q1 to 1960:Q4, with 4 extrapolations.

Figure 7: *Chow-Lin method: Objective functions for estimation*



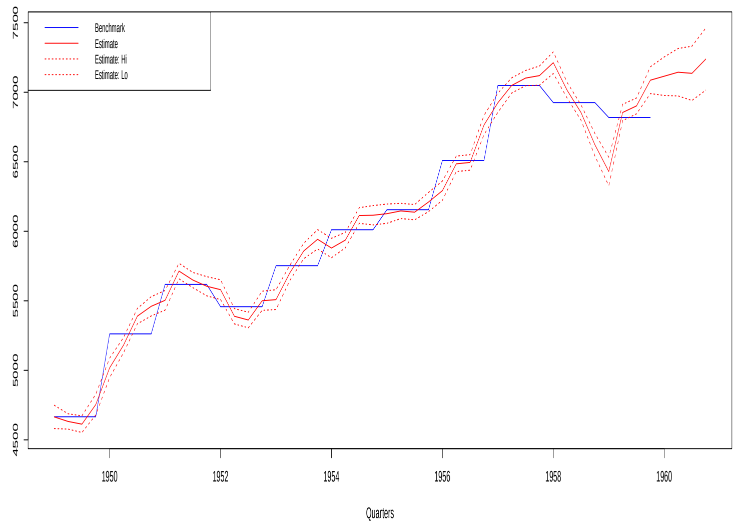

Refining the search over the restricted range $[0.90, 0.999999999]$, with 1000 grid points, we get a more clear view of the "cliff" of the log-likelihood that starts at $\rho = 0.9874$. This value can be considered as the optimal $\rho$ to set the basis for the estimation.

Figure 8: *Chow-Lin method: Log-likelihood function*




The $y$ estimates with their $\pm\sigma$ intervals are represented in the next graph, including the $Y$ benchmark.

Figure 9: *Chow-Lin method: estimates vs annual benchmark*




### 3.4 Best Linear Unbiased Esimation with indicators: dynamic models

The `td_R` package also includes two functions that implement an explicitly dynamic approach: Santos-Cardoso (2001) and Proietti (2006), see Di Fonzo (2002)[5]. Both methods share the same equation, restricted in the case of Santos-Cardoso by imposing $\gamma = 0$ in equation (14).

$$(1 - \phi B)y_t = \alpha + (\beta + \gamma B)x_t + u_t. \tag{14}$$

The corresponding functions are described in the next table:

Table 9: Functions for temporal disaggregation with indicators: dynamic

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| santos_cardoso | res | Y, x, ta, sc, type_estim, opC, rl | Santos-Cardoso method |
| proietti | res | Y, x, ta, sc, type_estim, opC, rl | Proietti method |

[5]For alternative dynamic models, see Gregoir (1994), Salazar et al. (1994) and Guay and Maurin (2015).

Note that the Proietti model implies a special type of transfer function, the so-called first-order Autoregressive Dynamic Linear model or ADL(1,1):

$$y_t = \alpha^* + \frac{(\beta + \gamma B)}{(1 - \phi B)} x_t + \frac{1}{(1 - \phi B)} u_t. \tag{15}$$

In this model, the signal provided by the input $x_t$ is transferred to the output $y_t$ via a balanced linear filter with a long-run gain $\frac{\beta+\gamma}{1-\phi}$. The innovations are processed by means of an AR(1) filter, controlled by the same $\phi$ parameter that appears in the ADL(1,1) transfer operator.

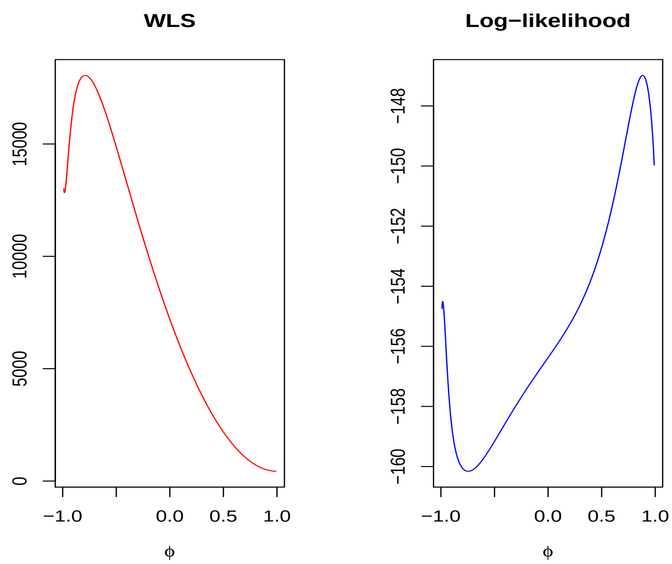
Although both methods share a similar transition equation, they are estimated in rather different ways. Santos-Cardoso adapt the Bournay-Laroque ML approach to deal with the initial condition $y_0$, while Proietti casts the model in state space form and then runs the Kalman filter to compute its likelihood, which is maximized in order to estimate the parameters. The `proietti()` function of `td_R` uses a shortcut, adapting the `santos_cardoso()` function to deal with an additional regressor, $x_{t-1}$, suitably completed with an ARIMA backcast for $x_0$. This backast is estimated by means of the `auto.arima()` function of the `forecast` package (Hyndman and Khandakar, 2008).

Hence, the script for running Proietti is quite similar to the one used for running Santos-Cardoso, as can be seen in the next box:

```
# Type of aggregation
ta <- 1
# Frequency conversion
sc <- 4
# Method of estimation
type_estim <- 1
# Intercept
opC <- 1
# Interval of grid search for rho
# rl <- NULL
# Default:  search on [0.05 0.99] with 100 grid points
# rl <- 0.75
# Fixed value
rl <- c(-0.99, 0.99, 1000) # Specific range:  min, max and number of
grid points
# Calling the function
res <- proietti(Y, x, ta, sc, type_estim, opC, rl)
```

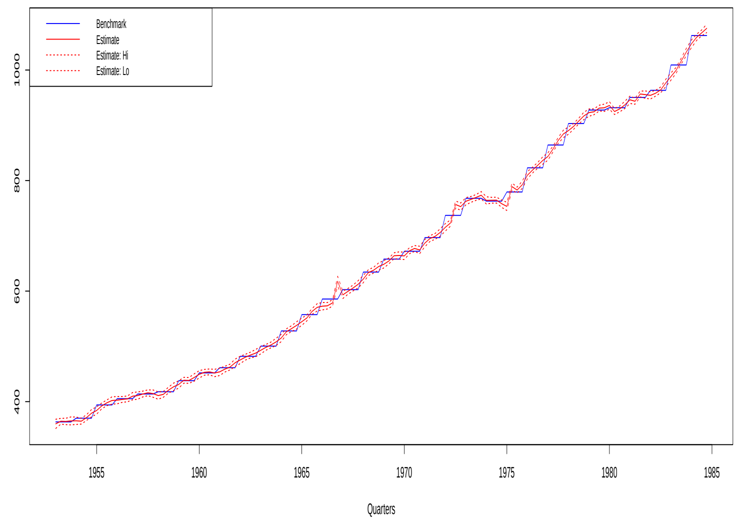
The output of two alternative objective functions, the sum Weighted Least Squares (WLS) and the Log-Likelihood, are drawn in the range of search, using the same data as Santos and Cardoso (2001): $Y$ = US annual personal consumption, $x$ = US quarterly personal personal disposable income, seasonally adjusted. The sample runs from 1953:Q1 to 1984:Q4, without extrapolations.

Figure 10: *Proietti method: Objective functions for estimation*




The search on the log-likelihood yields $\phi = 0.8830$ as the optimal value, forming the basis for the complete estimation of the model. The corresponding $y$ estimates with their $\pm 2\sigma$ intervals are represented in the next graph, including the $Y$ benchmark.

Figure 11: *Proietti method: Estimates vs annual benchmark*




## 3.5 Best Linear Unbiased Esimation with Indicators: ARIMA-based Approach

The starting point of the method proposed by Guerrero (1990) is the assumption that the unobservable high frequency counterparty $y$ of a low frequency benchmark $Y$ can be represented by means of a general multiplicative ARIMA model. Guerrero's method solves the benchmarking problem using also the information available in a set of $k$ high frequency indicators using also a BLUE approach. The method can be stated using the following algorithm[6]:

- Estimation of the scaled indicator, by means of OLS on the low frequency model.
- Preliminary estimator, based on the information provided by the ARIMA model for the scaled indicator.

[6]See also Martínez and Guerrero (1995) and Quilis (2004) for additional information about this method.

- Final estimator, adding the information provided by the model for the high frequency discrepancy.

The inputs and outputs of the Guerrero (1990) function are described in the next script.

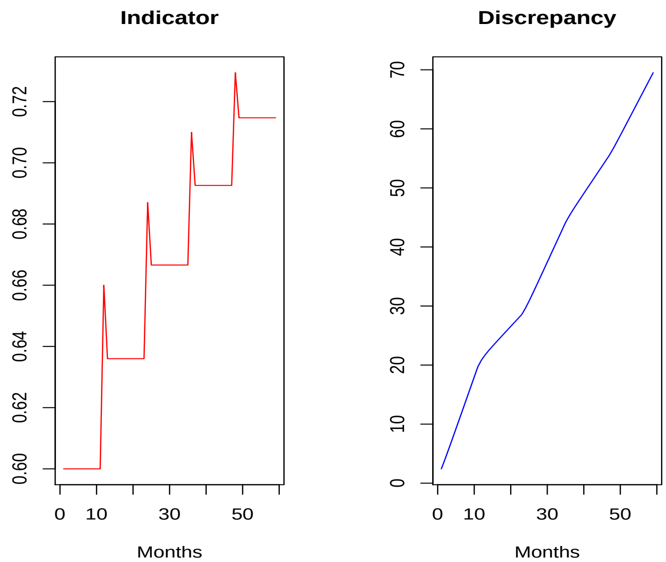
```
# Type of aggregation
ta <- 1
# Frequency conversion
sc <- 12
# Intercept
opC <- -1
# Model for w:  (0,1,1)(1,0,1) [Scaled indicator]
rexw <- list( ar_reg = c(1),
d = 1,
ma_reg = c(1, -0.40),
ar_sea = c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.85),
bd = 0,
ma_sea = c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.79),
sigma = (4968.716)**2)
# Model for the discrepancy:  (1,2,0)(1,0,0)
rexd <- list( ar_reg = c(1, -0.43),
d = 2,
ma_reg = c(1),
ar_sea = c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.62),
bd = 0,
ma_sea = c(1),
sigma = (76.95)**2)
# Calling the function
res <- guerrero(Y, x, ta, sc, rexw, rexd, opC)
```

The main difference with respect to other BLUE methods is the need to specify completely an ARIMA model for the (scaled) high frequency indicator, which is stored in the list `rexw`. Depending on the dynamic nature of the discrepancy between a preliminary high frequency estimate and the scaled indicator, an additional ARIMA model is also required, which is stored in the list `rexd`.

The $\psi-$weights of the models stored in `rexw` and `rexd` are represented in the next figure. They are derived from the data used in Guerrero (1990): $Y =$ México's annual Real Gross Domestic Product (RGDP), unit: Millions of

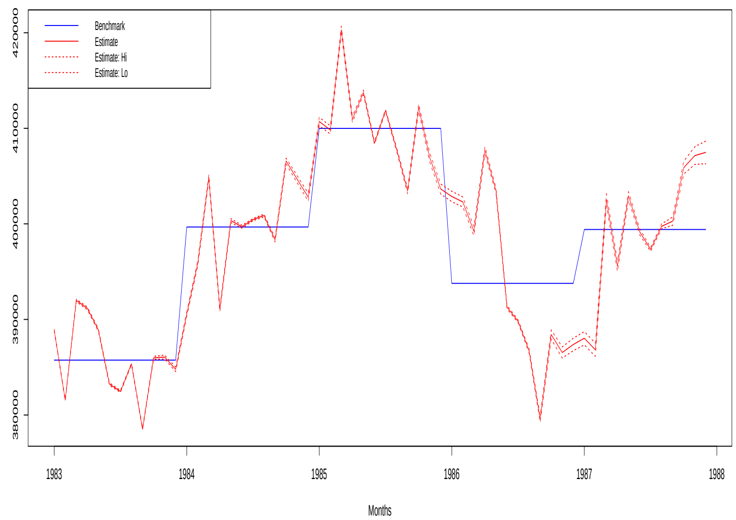
Pesos at the 1980 value, $x =$ México's monthly Index of Volume of Industrial Production (IVIP), unit: Index 1980=100. The sample runs from 1983:1 to 1987:12, without extrapolations.

Figure 12: *Guerrero method: $\psi$-weights*




The corresponding $y$ estimates with their $\pm 2\sigma$ intervals are represented in the next graph, including the $Y$ benchmark.

Figure 13: *Guerrero method: Estimates vs annual benchmark*




# 4 Multivariate Temporal Disaggregation

The evolution of temporal disaggregation is clearly related to its multivariate extension, including the incorporation of richer transversal constraints, see Rossi (1982), Di Fonzo (1990, 1994), Guerrero and Nieto (1999), Di Fonzo and Marini (2003, 2011) and Proietti (2011a) among others. In this package we have included alternative methods that consider explicitly several low frequency benchmarks and one high frequency transversal constraint. They are:

Table 10: Multivariate Temporal Disaggregation

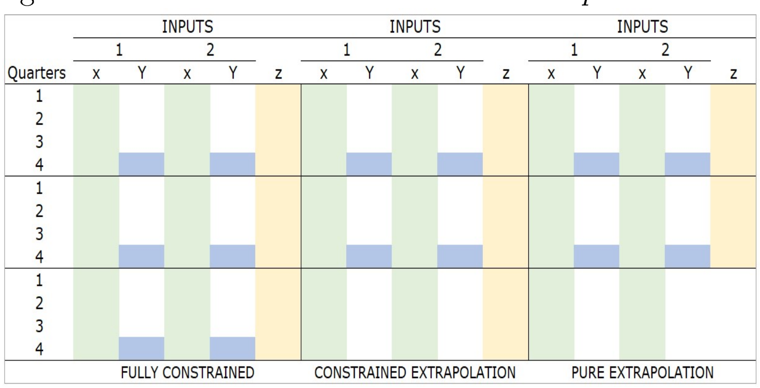
| Name | Output | Inputs | Explanation |
|---|---|---|---|
| denton_multi | res | Y, x, z, ta, sc, d, op1 | Solution based on Denton's quadratic optimization |
| di_fonzo | res | Y, x, z, ta, sc, op1, f | Di Fonzo's model-based approach |
| rossi_d | Script that implements Rossi's method | | |

To illustrate the methods, we have used a simplified example based on regional data for Spain, see Cuevas et al. (2015) for additional details.

The multivariate version of the Denton procedure can be considered as a straightforward extension of its univariate version. Rossi (1982) proposed a two-step approach that combines a first-step (preliminary) estimation by means of a model-based procedure (e.g., Chow-Lin, Fernández or Litterman) and a second-step that incorporates the transversal constraint while preserving the temporal consistency achieved in the first step. This approach can be considered as an intermediate case between the multivariate Denton (which is close to a reconciliation procedure) and the Di Fonzo's method (which is a model-based approach).

Di Fonzo (1990) proposed a model-based method to perform multivariate temporal disaggregation with a transversal constraint. The method extends the framework provided by Chow-Lin to the multivariate case and assumes that the innovations are driven by a vector white noise or by a vector random walk. This procedure can handle doubly constrained estimation, transversally constrained estimation and free estimation. The next table illustrates the different cases[7]:

Figure 14: *Di Fonzo method: alternative input structures*




As can be seen, the method allows for fully constrained estimation (the temporal constraint is binding as well as the cross-sectional constraint), constrained extrapolation (only the cross-sectional constraint is binding) and pure extrapolation (without any binding constraint). An additional source of flexibility is provided by the fact that the aggregates may have a differ-

[7]Here we consider $M = 2$ annual benchmarks ($Y$), with their corresponsing quarterly trackers ($x$) and one quarterly constraint ($z$) that (eventually) binds the quarterly estimates ($y$).

ent number of trackers. In addition to this flexibility, the method provides standard errors of the estimates that are useful to gauge its uncertainty.

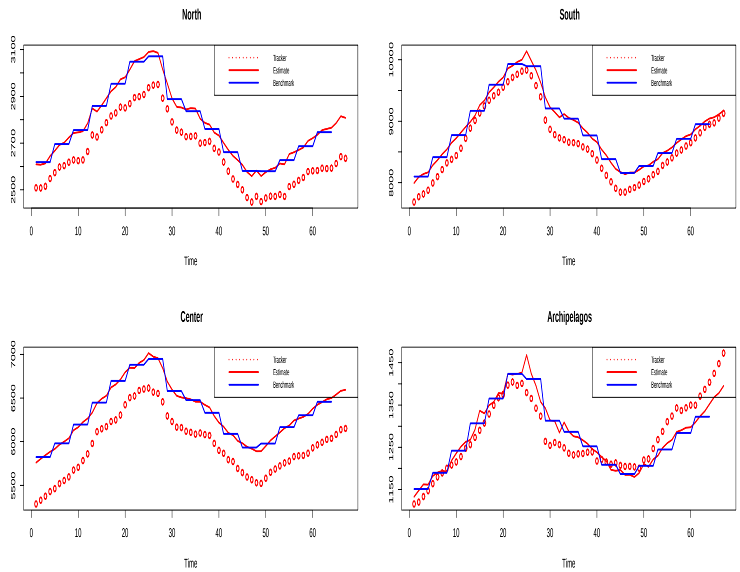
The corresponding function can be implemented as follows[8]:

```
# Type of aggregation
ta <- 2
# Frequency conversion
sc <- 4
# Model for the innovations:  white noise (0), random walk (1)
op1 <- 1
# Number of high frequency indicators linked to each low frequency
aggregate (one tracker)
M <- ncol(Y)
f <- matrix(1, 1, M)
# Multivariate temporal disaggregation
res <- di_fonzo(Y, x, z, ta, sc, op1, f)
```

The next figure shows the graphical output:

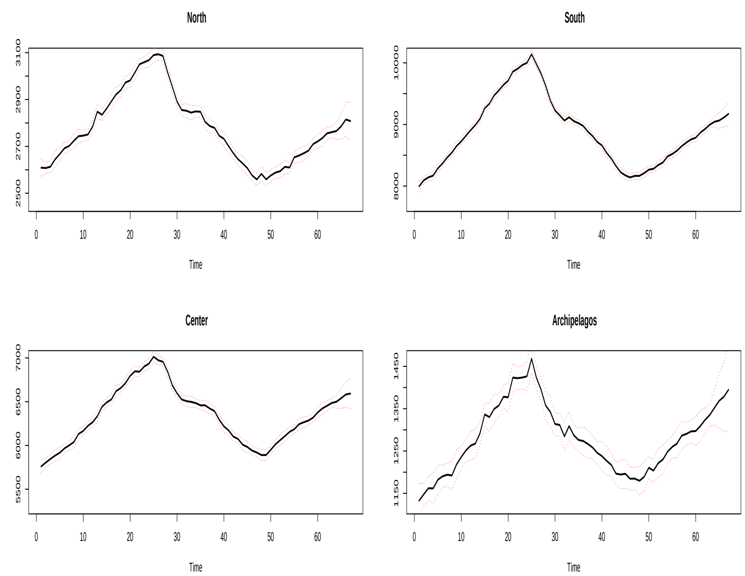
[8] To illustrate the method, we have used a simplified version of the regional data set used by Cuevas et al. (2015). The simplification consists of considering 4 mega-regions using a simple geographic criterion, instead of the actual 17 Spanish regions and only one high frequency tracker instead of several trackers.

Figure 15: *Di Fonzo method: benchmarks, trackers and estimates*




The next figure shows the estimates for each region and a measure of its reliability, as reported by their $\pm 2\sigma$ intervals.

Figure 16: *Di Fonzo method: estimates and $\pm 2\sigma$ intervals*




# 5 Transversal Reconciliation

Up to a certain point, reconciliation can be considered as a special case of multivariate temporal disaggregation with a transversal constraint, by means of removing from the latter the temporal consistency requirement.

Dropping this requirement, we can consider a simple method (proportional balancing) or adding a layer of complexity, if we consider several transversal constraints (as in the case of the bi-proportional RAS method) or the incorporation of uncertainty in the estimation process (as in the van der Ploeg method). A detailed analysis of these issues can be found in Chen (2012), Bikker et al. (2010) and Chen et al. (2018b), among others.

The list of available procedures is:

Table 11: Functions for transversal reconciliation

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| bal | res | Y, z | Proportional reconciliation of Y given z |
| ras | F1 | F0, x0, x1, v, u, opG | Bi-proportional reconciliation of 2x2 tables |
| van_der_ploeg | res | y, S, A, a | Reconciliation by means of QL optimization |

The so-called RAS or bi-proportional method provides a bidimensional extension of the proportional method, e.g. funcion `bal()`, allowing its application to more complex data structures like Input-Outout (IO) tables, see Bacharach (1965). The `ras()` function can be applied as follows:

```
# RAS reconciliation
F1 <- ras(F0, x0, x1, v, u, opG=1)
```

The inputs and the output of the function `ras()` are summarized in the next figure:

Figure 17: *Bi-proportional reconciliation of 2x2 tables: RAS algorithm*

**BENCHMARK (T=0)**

| | Product | | | | | | |
|---|---|---|---|---|---|---|---|
| Product | A | B | C | Total | Discrepancy | Final Demand | Total Output |
| A | 50 | 100 | 0 | **150** | 0 | 50 | **200** |
| B | 30 | 50 | 20 | **100** | 0 | 200 | **300** |
| C | 20 | 50 | 30 | **100** | 0 | 100 | **200** |
| Total | **100** | **200** | **50** | | **0** | 350 | 700 |
| Discrepancy | 0 | 0 | 0 | | | | |
| Value Added | 100 | 100 | 150 | | **F0** | | |
| Total Output | **200** | **300** | **200** | 700 | **x0** | | |

**UPDATE (T=1)**

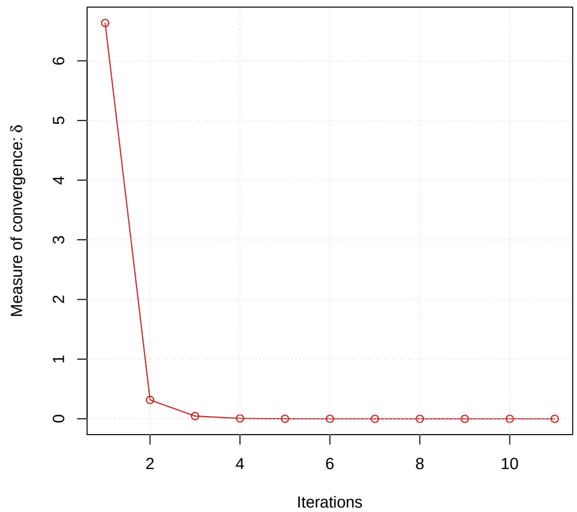
| | Product | | | | | | |
|---|---|---|---|---|---|---|---|
| Product | A | B | C | Total | Discrepancy | Final Demand | Total Output |
| A | | | | **160** | | 40 | **200** |
| B | | | | **150** | | 250 | **400** |
| C | | | | **120** | | 180 | **300** |
| Total | **100** | **250** | **80** | | | 470 | 900 |
| Discrepancy | | | | | | | |
| Value Added | 100 | 150 | 220 | | **u** | | |
| Total Output | **200** | **400** | **300** | 900 | **v** | | |

**BALANCED UPDATE (T=1)**

| | Product | | | | | | |
|---|---|---|---|---|---|---|---|
| Product | A | B | C | Total | Discrepancy | Final Demand | Total Output |
| A | **45.25** | **114.75** | **0.00** | **160** | 0 | 40 | **200** |
| B | **36.23** | **76.56** | **37.21** | **150** | 0 | 250 | **400** |
| C | **18.52** | **58.69** | **42.79** | **120** | 0 | 180 | **300** |
| Total | **100** | **250** | **80** | | **0** | 470 | 900 |
| Discrepancy | 0 | 0 | 0 | | | | |
| Value Added | 100 | 150 | 220 | | **F1** | | |
| Total Output | **200** | **400** | **300** | 900 | | | |

The function also provides information about the convergence speed of the algorithm:

Figure 18: *RAS algorithm: convergence*




The method proposed by van der Ploeg (1985, 1986) combines a solid reconciliation procedure, based on solving a quadratic optimization program, with the possibility of including a priori information about the reliability of the estimates. This combination yields an extremely flexible procedure that can handle several constraints as well as different scenarios regarding the uncertainty of the inputs, see Abad et al. (2006) for a full-fledged application of this method to the Spanish Quarterly Non-Financial Accounts for the Institutional Sectors.

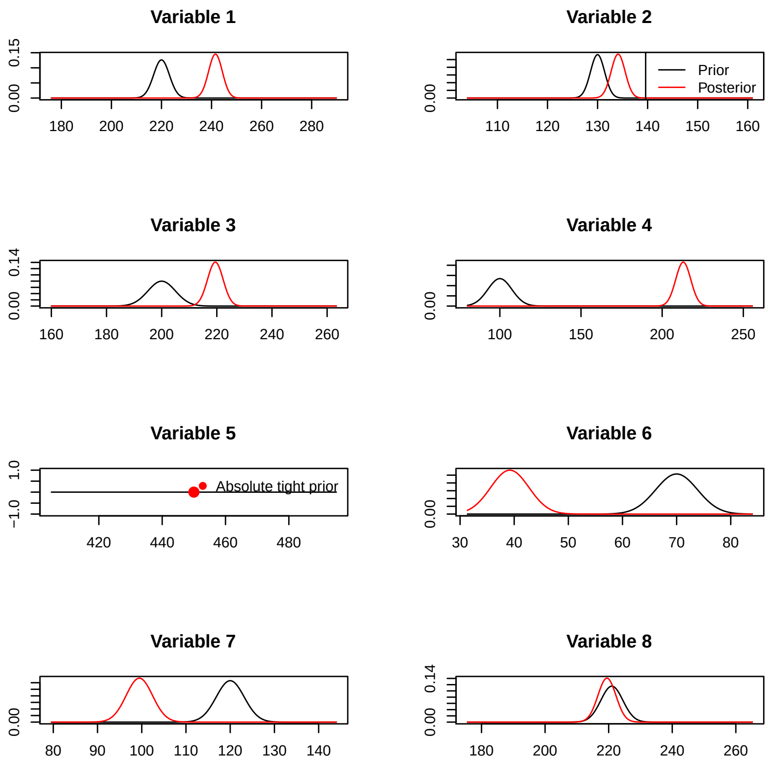
The method proposed by van der Ploeg (1985, 1986) combines a solid reconciliation procedure, based on a quadratic optimization problem, with the possibility of including prior information about the reliability of the estimates. This combination yields a very flexible procedure that can handle several constraints as well as different scenarios regarding the uncertainty of the inputs, see Abad et al. (2006) for a full-fledged application to the Spanish Quarterly Non-Financial Accounts for the Institutional Sectors.

The next script implements the van der Ploeg method in a case in which eight estimates must satisfy two transversal constraints. The inputs include an initial (unbalanced) estimate of the variables ($y$) as well as a measure of it's a priori uncertainty ($S$).

```
# Unbalanced cross-section vector
y <- c(220.00, 130.00, 200.00, 100.00, 450.00, 70.00, 120.00,
221.00)
y <- as.matrix(y)
k <- length(y)
# Linear constraints
A <- matrix(c(1, 0, 1, 0, 1, 1, 1, 0, -1, 0, -1, 0, -1, 0, -1, -1),
ncol = 2, byrow = TRUE)
# VCV matrix of estimates
s <- c(10, 2, 25, 55, 0, 15, 10, 12) #Variable 5 does not require
adjustment: absolutely tight prior
Aux1 <- diag(sqrt(s))
# Correlation matrix
C <- matrix(0, nrow = k, ncol = k)
C[1, 3] <- C[3, 1] <- 0.5
C <- C + diag(rep(1, k))
# Final S matrix
S <- Aux1 %*% C %*% Aux1
# van der Ploeg balancing
res <- van_der_ploeg(y, S, A)
```

The corresponding output of the function `van_der_ploeg()` is depicted in the next figure, including a comparison of the prior distribution (unbalanced) with the posterior distribution (balanced) under the assumption of Gaussianity.

Figure 19: *Constrained least squares reconciliation: van der Ploeg method*




# A Appendix. Structure of the Output: the `res` List

Most of the functions referenced in this paper generate as output an object belonging to the list class. This list contains the detailed results from the corresponding function and its typical structure is as follows:

```
res <- list(
meth ='Chow-Lin',
ta = type of disaggregation,
type_estima = estimation method,
opC = option related to the intercept,
N = number of observations of the low frequency input (Y)
n = number of observations of the high frequency input (x)
npred = number of extrapolations,
sc = frequency conversion between the low and the high frequency
inputs,
p = number of regressors (including the intercept, if it exists),
Y = low frequency data,
x = high frequency indicators,
y = high frequency estimate,
y_se = high frequency estimate:  standard error,
y_lo = high frequency estimate:  se - sigma,
y_hi = high frequency estimate:  se + sigma,
u = high frequency residuals,
U = low frequency residuals,
beta = estimated model parameters,
beta_se = estimated model parameters:  standard error,
beta_t = estimated model parameters:  t ratios,
rho = innovational parameter,
sigma_a = variance of shocks,
aic = Akaike's Information Criterion,
bic = Bayesian Information Criterion,
val = Objective function used by the estimation method,
wls = Weighted Least Squares as a function of rho,
loglik = Log likelihood as a function of rho,
r = grid of innovational parameters used for estimation
)
```

Depending on the specific function the output may include more or less items (e.g. the methods of Santos-Cardoso and Proietti include information about the estimate for the remainder, Guerrero's procedure include information about a statistic to check the conformity between $Y$ and $x$, etc.).

# B Appendix: Auxiliary Functions

`td_R` has some additional auxiliary functions that help other functions to perform their main task but can also be used independently. They are presented in the next table:

Table 12: Auxiliary functions

| Name | Output | Inputs | Explanation |
|---|---|---|---|
| dif | D | d, n | Difference operator $(1-B)^d$ in matrix form |
| vec | a | A | Vectorizing (column stacking) of matrix $A$ |
| desvec | A | a, m | Forming a matrix with $m$ columns from vector $a$ |
| hp | res | z, lambda | Hodrick-Prescott filter (two-sided) |

# References


[1] Abad, A. and E.M. Quilis (2005) "Software to Perform Temporal Disaggregation", Eurostat, Working Paper and Studies, ISSN 1725-4825.

[2] Abad, A., A. Cuevas and E.M. Quilis (2006) "Proyección del cuadro macroeconómico y de las cuentas de los sectores institucionales mediante un modelo de equilibrio", Instituto de Estudios Fiscales, Papeles de Trabajo n. 27.06.

[3] Bacharach, M. (1965) "Estimating Non-Negative Matrices from Marginal Data", International Economic Review, 6(3), 294-310.

[4] Barcellan, R. and G.L. Mazzi (1995) "ECOTRIM a Program for Temporal Disaggregation of Time Series", Eurostat, Working Paper.

[5] Barcellan, R., T. Di Fonzo, D. Raffaele, V. Staplehurst and D. Buono(2003) "ECOTRIM: A Program for Temporal Disaggregation of Time Series", Eurostat, Working Paper.

[6] Barbone, L., G. Bodo and I. Visco (1981) "Costi e profitti nell'industria in senso stretto: Un'analisi su serie trimestrali, 1970–1980", Bolletino della Banca d'Italia, p. 467–510.

[7] Bikker, R., J. Daalmans and N. Mushkudiani (2010) "A Multivariate Denton Method for Benchmarking Large Data Sets", Statistics Netherlands, Discussion Paper n. 10002.

[8] Bisio, L. and F. Moauro (2018) "Temporal Disaggregation by Dynamic Regressions: Recent Developments in Italian Quarterly National Accounts", Statistica Neerlandica, 72, 471-494.

[9] Boot, J.C.G., W. Feibes and J.H.C. Lisman (1967) "Further Methods of Derivation of Quarterly Figures from Annual Data", Applied Statistics, 16, 65-75.

[10] Bournay, J. and G. Laroque (1979) "Réflexions sur la méthode d'elaboration des comptes trimestriels", Annales de l'INSEE, 36, 3-30.

[11] Borchers, H.W., R. Hankin and S. Sokol (2023) "`pracma`: Practical Numerical Math Functions", R package version 2.4.4, CRAN Repository.

[12] Cardoso, F. (1999) "Métodos econométricos de desagregação de series temporais", Bolsa de Derivados do Porto, Porto.

[13] Casals, J., A. García-Hiernaux, M. Jerez, S. Sotoca and A.A. Trindade (2016) "State Space Methods for Time Series Analysis: Theory, Applications and Software", Chapman and Hall/CRC.

[14] Chen, B. (2012) "A Balanced System of U.S. Industry Accounts and Distribution of the Aggregate Statistical Discrepancy by Industry", Journal of Business and Economic Statistics, 30, 202-211.

[15] Chen, B., T. Di Fonzo and N. Mushkudiani (2018a), "Benchmarking, Temporal Disaggregation, and Reconciliation of Systems of Time Series", Statistica Neerlandica, 72, 402-405.

[16] Chen, B., T. Di Fonzo and N. Mushkudiani (2018b), "The Statistical Reconciliation of Time Series of Accounts Between Two Benchmark Revisions", Statistica Neerlandica, 72, 533-552.

[17] Cholette, P. A. (1984) "Adjusting Sub-Annual Series to Yearly Benchmarks," Survey Methodology, 10, 35–49.

[18] Chow, G. and A.L. Lin (1971) "Best Linear Unbiased Distribution and Extrapolation of Economic Time Series by Related Series", Review of Economic and Statistics, 53, 372-375. 47.

[19] Cuevas, A., E.M. Quilis and A. Espasa (2015) "Quarterly Regional GDP Flash Estimates by Means of Benchmarking and Chain Linking", Journal of Official Statistics, 31, 627–647.

[20] Daalmans, J. (2018) "On the Sequential Benchmarking of Subannual Series to Annual Totals", Statistica Neerlandica, 72, 406-420.

[21] Dagum, E.B. and P.A. Cholette (2006) Benchmarking, Temporal Distribution, and Reconciliation Methods for Time Series, Springer.

[22] Denton, F.T. (1971) "Adjustment of Monthly or Quarterly Series to Annual Totals: An Approach Based on Quadratic Minimization", Journal of the American Statistical Society, 66, 99-102.

[23] Di Fonzo, T. (1987) La stima indiretta di serie economiche trimestrali, Cleup Editore.

[24] Di Fonzo, T. (1990) "The Estimation of m Disaggregate Time Series when Contemporaneous and Temporal Aggregates are Known", Review of Economics and statistics, 72, 178-182.

[25] Di Fonzo, T. (1994) "Temporal Disaggregation of a System of Time Series when the Aggregate Is Known: Optimal vs. Adjustment Methods", INSEE-Eurostat Workshop on National Accounts, Paris.

[26] Di Fonzo, T. (2002) "Temporal Disaggregation of Economic Time Series: Towards a Dynamic Extension", Dipartimento di Scienze Statistiche, Università di Padova, Working Paper n. 2002-17.

[27] Di Fonzo, T. and M. Marini (2003) "Benchmarking Systems of Seasonally Adjusted Time Series According to Denton's Movement Preservation Principle", Dipartimento di Scienze Statistiche, Università di Padova, Working Paper no. 2003–09.

[28] Di Fonzo, T. and M. Marini (2011) "Simultaneous and Two-step Reconciliation of Systems of Time Series: Methodological and Practical Issues", Journal of the Royal Statistical Society, Series C, 60, 143–164.

[29] Di Fonzo, T. and M. Marini (2012) "On the Extrapolation with the Denton Proportional Benchmarking Method", International Monetary Fund, WP 12/169.

[30] Doan, T. (2008) "Disaggregate: A General Procedure for Interpolation", RATS-Estima.

[31] Eurostat (2018) "ESS Guidelines on Temporal Disaggregation, Benchmarking and Reconciliation", Eurostat.

[32] Fernández, R.B. (1981) "Methodological Note on the Estimation of Time Series", Review of Economic and Statistics, 63, 471-478.

[33] Gregoir, S. (1994) "Propositions pour une désagrégation temporelle basée sur des modèles dynamiques simples", Eurostat (Ed.) Workshop on Quarterly National Accounts, Luxembourg.

[34] Grudkowska, S. (2015) "JDemetra+ User Guide", Narodowy Bank Polski, Working Paper.

[35] Guay, A., and A. Maurin (2015) "Disaggregation methods based on MIDAS regression", Economic Modelling, 50, 123–129.

[36] Guerrero, V. (1990) "Temporal Disaggregation of Time Series: an ARIMA-Based Approach", International Statistical Review, 58, 29-46.

[37] Guerrero, V. and F.H. Nieto (1999) "Temporal and Contemporaneous Disaggregation of Multiple Time Series", Test, 8, 459-489.

[38] Guerrero, V. and F. Corona (2018) "Retropolating Some Relevant Series of Mexico's System of National Accounts at Constant Prices: The Case of Mexico City's GDP", Statistica Neerlandica, 72, 495-519.

[39] Hyndman, R.J. and Y. Khandakar. 2008 "Automatic time series forecasting: the forecast package for R", Journal of Statistical Software, 27(3).

[40] IMF (2018) "Quartlery Accounts Manual", Real Sector Division of the Statistics Department, International Monetary Fund.

[41] Litterman, R.B. (1983) “A Random Walk, Markov Model for the Distribution of Time Series”, Journal of Business and Economic Statistics, 1, 169-173.

[42] Martínez, J. and V. Guerrero (1995) “A Recursive ARIMA-Based Procedure for Disaggregating a Time Series Variable Using Concurrent Data”, Test, 4(2), 359-76.

[43] Melis, F. (1992) “Agregación temporal y solapamiento o ’aliasing’ ”, Estadística Española, 130, 309-346.

[44] Proietti, T. (2006) “Temporal Disaggregation by State Space Methods: Dynamic Regression Methods Revisited”, Econometrics Journal, 9, 357–372.

[45] Proietti, T. (2011a) “Multivariate Temporal Disaggregation with Cross-sectional Constraints”, Journal of Applied Statistics, 38, 1455–1466.

[46] Proietti, T. (2011b) “Estimation of Common Factors under Cross-sectional and Temporal Aggregation Constraints”, International Statistical Review, 79, 455–476.

[47] Quilis, E.M. (2004) “Sobre el método de desagregación temporal de Guerrero”, Instituto Nacional de Estadística, Boletín Trimestral de Coyuntura 91, 163-189.

[48] Quilis, E.M. (2018) “Temporal Disaggregation of Economic Time Series: The View from the Trenches”, Statistica Neerlandica, 72, 447-470.

[49] Quilis, E.M. (2019) “Temporal Disaggregation Of Time Series: A Matlab Library”, MATLAB Central File Exchange.

[50] Rossi, N. (1982) “A Note on the Estimation of Disaggregate Time Series when the Aggregate is Known”, Review of Economics and Statistics, 64, 695-696.

[51] Salazar, E., R. Smith, S. Wright and M. Weale (1994) “Indicators of Monthly National Accounts”, Eurostat (Ed.) Workshop on Quarterly National Accounts, Luxembourg.

[52] Santos, J.M.C. and F. Cardoso (2001) “The Chow-Lin Method Using Dynamic Models”, Economic Modelling, 18, 269-280.

[53] Sax, C., and P. Steiner (2013) ”Temporal Disaggregation of Time Series”, The R Journal, 5, 80–87.

[54] `signal` developers (2023) “signal: Signal Processing”, R package version 1.8.0, CRAN Repository.

[55] Sims, C.A. (1974) “Seasonality in Regression”, Journal of the American Statistical Association, 69: 618-626.

[56] Stram, D.O. and W.W.S. Wei (1986a) “A Methodological Note on the Disaggregation of Time Series Totals”, Journal of Time Series Analysis, 7, 293-302.

[57] Stram, D.O. and W.W.S. Wei (1986b) “Temporal Aggregation in the ARIMA Process”, Journal of Time Series Analysis, 7(4), 279-292.

[58] Temursho, U. (2018) “Entropy-Based Benchmarking Methods”, Statistica Neerlandica, 72, 421-446.

[59] Tiao, G.C. (1972) “Asymptotic behavior temporal aggregates of time series”, Biometrika, 59, 525-31.

[60] van der Ploeg, F. (1982) “Reliability and the Adjustment of Large Economic Accounting Matrices”, Journal of the Royal Statistical Society, series A, 145, 169-194.

[61] van der Ploeg, F. (1985) “Econometrics and Inconsistencies in the National Accounts”, Economic Modelling, 2, 8-16.

[62] Wei, W.W.S. (1990) Time Series Analysis, Addison Wesley.