SORT 50 (1) January-June 2026, 1-22 DOI:

An algorithm for reconciling indicators across multiple dimensions: weighted iterative proportional fitting

Jose M. Pavía[1], Josep Lledó[2] and Priscila Espinosa[3]

Abstract

Reconciling multidimensional count data across multiple sources is a common challenge in social and economic research. Iterative proportional fitting is widely used for this purpose, but aligning indices under weighted sum-convex constraints calls for a more flexible approach. We introduce the weighted iterative proportional fitting algorithm, which incorporates sum-weighted constraints to adjust indicators—such as death-risk indices by wealth, habitat, and climate—while preserving marginal consistency. Weighted iterative proportional fitting has been implemented in an R package of the same name, enabling scholars, statisticians, and policymakers' advisors, among others, to apply it easily to multidimensional data.

MSC: 62P20, 62H10, 15A23.

Keywords: WIPF, biproportional fitting, RAS, raking, matrix scaling, socioeconomic indices, official statistics.

1. Introduction

The iterative proportional fitting (IPF) algorithm is a well-established procedure for reconciling count data coming from two or more sources of information. Typically, the process involves adjusting the (non-negative) inner-entries of a matrix (or a two-way contin-

[1] Universitat de Valencia, Valencia (Spain), email: pavia@uv.es ORCID: https://orcid.org/0000-0002-0129-726X. Corresponding author.

[2] Universitat de Valencia, Valencia (Spain), email: josep.lledo@uv.es ORCID: https://orcid.org/0000-0002-7475-8549

[3] Universitat de Valencia, Valencia (Spain), email: priscila.espinosa@uv.es ORCID: https://orcid.org/0000-0002-3870-6194

Received: January 2026
Accepted: March 2026

gency table) of estimates—or entries derived from a less reliable source of information—to match some known aggregate margins (the row and column totals) coming from a more reliable data source.

Beyond this ad hoc, data-reconciliation interpretation, IPF is also closely connected to maximum likelihood estimation and constrained optimization (Zaloznik, 2011). It can be viewed both as a procedure for adjusting a multidimensional table to match known marginals while preserving the initial cell structure (i.e., prior information) as much as possible, and as a method for estimating maximum likelihood parameters in theoretical or model-based contexts; particularly within log-linear models. In the first case, IPF yields the distribution that is closest to the initial distribution in terms of relative entropy—that is, the distribution that minimizes the Kullback-Leibler divergence—subject to the marginal constraints. In the second case, however, despite the widespread belief that IPF always converges to the maximum likelihood (ML) estimate consistent with the marginal constraints, the property only holds when the initial estimates are consistent with the model structure.

The procedure, which can be traced back to Kruithof (1937) and Deming and Stephan (1940), is known by various names across different disciplines in the literature. These include biproportional fitting in statistics (Pavía, Cabrer and Sala, 2009), RAS in economics (Wiebe and Lenzen, 2016), raking in survey research (Deville, Särndal and Sautory, 1993), and matrix scaling in computer science (Kalantari and Khachiyan, 1996). Furthermore, IPF is also referred to as iterative proportional scaling (IPS), particularly in the context of ML estimation, as used in machine learning, image processing, or graphical models (Coons, Langer and Ruddy, 2024).

Despite the relevance of the approach, as demonstrated by its repeated rediscovery by different scientific communities (Allen-Zhu et al., 2017), its broad use in a multitude of problems (Idel, 2016), and its many extensions (Klimova and Rudas, 2015; Suesse et al., 2017; Fournier Gabela, 2020; von Lindheim and Steidl, 2023), a generalization that involves making adjustments after imposing constraints on weighted (convex) sums of inner values across dimensions is still missing. The aim of this paper is to address this gap by generalizing the IPF algorithm in that direction. This gives rise to a new procedure: the weighted iterative proportional fitting (WIPF) algorithm, available in an R-package of the same name (Pavía, 2026).

Our motivation for developing this new algorithm is practical rather than theoretical. During the process of estimating socio-economic death-risk indices as a function of contextual wealth, habitat size, and climate area, we experienced smaller sampling sizes (exposed-to-risk) and increasing levels of uncertainty and lack of reliability in the estimates as the population was progressively split by a larger number of factors. Given the convex aggregation relationships linking inner and outer levels of risk-indices, we found that three- and two-interaction estimates could be significantly improved by making the initial estimates congruent with the two- and one-factor estimates, and that this could be easily performed by generalizing IPF. This paper describes the new algorithm, presents the `WIPF` package, and exemplifies its use by applying it to the problem of making congruent death-risk indices of different levels. Obviously, the usefulness of this new algorithm extends beyond this example, as it can be employed to reconcile any set of indicators that are linearly related with positive coefficients—for instance, harmonizing life expectancies jointly estimated by educational attainment and sex with their corresponding marginal life expectancies.

2. A motivating application

Understanding mortality patterns is fundamental for assessing social well-being and inequality (Lagravinese, Liberati and Resce, 2020). Differences in mortality levels across population groups provide valuable information about the distribution of health, living conditions, and social opportunity within a society (Pavía, Lledó and Roig, 2026). Assessing and quantifying these differences is essential for monitoring social progress and designing evidence-based public policies, being also of value for guaranteeing fairness in the insurance industry.

Beyond individual characteristics, contextual factors such as neighborhood wealth, habitat size, and climatic conditions exert a strong influence on mortality patterns. Populations with higher economic resources tend to enjoy better access to healthcare and healthier living environments (McMaughan, Oloruntoba and Smith, 2020; Dwyer-Lindgren et al., 2024), while rural or sparsely populated areas often face limited access to essential services (Cohen et al., 2023). Likewise, growing evidence links climatological conditions and climate change–related phenomena (e.g., extreme heat, pollution, or drought episodes) with increased mortality risks, particularly among vulnerable populations (Bell, O'Neill and Zanobetti, 2024; García-León et al., 2024; Masselot et al., 2025). Measuring and monitoring mortality through these contextual dimensions is therefore crucial to identify and address emerging inequalities driven by environmental and socioeconomic transformations.

In this context, drawing on more than four billion individual demographic records from the Spanish population (Pavía et al., 2026) and adapting the methodology proposed in Pavía and Lledó (2022), we derived a set of mortality indices by age and sex that capture how death risks vary with contextual factors such as neighborhood wealth, habitat size, and climatological conditions. These indices provide a multidimensional picture of inequality in mortality, reflecting how the social and environmental context in which individuals live is systematically related to their survival prospects.

The resulting indicators can be readily integrated into social statistics and monitoring systems to complement traditional demographic and economic measures. They can support the design and evaluation of regional and social policies aimed at reducing health inequalities, enhancing the capacity of policymakers and researchers to identify vulnerable areas and population groups.

Mortality risks also constitute the keystone of the life insurance business, and following the entry into force of the Test-Achats ruling (European Commission, 2012), EU insurers are interested in introducing easily accessible additional variables, other than mortality and survival risks—so as to better assess and segment their portfolios (Lledó and Pavía, 2026). These indices can be easily introduced into actuarial processes: by simply knowing the postal addresses of the insured, the baseline company death rate can be multiplied by the relevant index to adjust the initially assumed probabilities of death or survival, thus incorporating contextual risks.

However, estimating mortality indices that account simultaneously for several contextual dimensions poses a methodological challenge, since these factors interact and are not independent. When multiple factors are combined, small sample sizes arise, producing unstable or volatile estimates. Fortunately, indices at different aggregation levels are linearly related: outer-level indices can be expressed as convex combinations of inner risk indices. This property opens the door to more efficient estimation procedures, which is precisely where the methodological contribution of this paper lies.

In particular, following with this example, (i) symbolizing age by x and sex by s, (ii) designating by w, h and c as the generic levels of the factors contextual wealth, habitat size and climatological area, respectively (with $w = 1,\dots;W$, $h = 1,\dots;H$, $c = 1,\dots,C$), and (iii) defining by $E^{(\cdot)}_{x,s}$ as the exposed-to-risk population and $I^{(\cdot)}_{x,s}$ as the corresponding index—for instance, $E^{w,h}_{x,s}$ $\left(= \sum_{c=1}^{C} E^{w,h,c}_{x,s}\right)$ denotes the size of population exposed-to-risk with age x and sex s in areas characterized by contextual wealth w and habitat size h, while $I^{w,h,c}_{x,s}$ represents the index corresponding to age x and sex s for population living in areas with contextual wealth w, habitat size h, and climatological condition c—the following theoretical relationships hold:

$$1 = \sum_{w=1}^{W} \frac{E^{w}_{x,s}}{E_{x,s}} I^{w}_{x,s}, \qquad 1 = \sum_{h=1}^{H} \frac{E^{h}_{x,s}}{E_{x,s}} I^{h}_{x,s}, \qquad 1 = \sum_{c=1}^{C} \frac{E^{c}_{x,s}}{E_{x,s}} I^{c}_{x,s},$$

$$I^{w}_{x,s} = \sum_{h=1}^{H} \frac{E^{w,h}_{x,s}}{E^{w}_{x,s}} I^{w,h}_{x,s} = \sum_{c=1}^{C} \frac{E^{w,c}_{x,s}}{E^{w}_{x,s}} I^{w,c}_{x,s},$$
$$I^{h}_{x,s} = \sum_{w=1}^{W} \frac{E^{w,h}_{x,s}}{E^{h}_{x,s}} I^{w,h}_{x,s} = \sum_{c=1}^{C} \frac{E^{h,c}_{x,s}}{E^{h}_{x,s}} I^{h,c}_{x,s},$$
$$I^{c}_{x,s} = \sum_{w=1}^{W} \frac{E^{w,c}_{x,s}}{E^{c}_{x,s}} I^{w,c}_{x,s} = \sum_{h=1}^{H} \frac{E^{h,c}_{x,s}}{E^{c}_{x,s}} I^{h,c}_{x,s},$$

$$I^{w,h}_{x,s} = \sum_{c=1}^{C} \frac{E^{w,h,c}_{x,s}}{E^{w,h}_{x,s}} I^{w,h,c}_{x,s}, \qquad I^{w,c}_{x,s} = \sum_{h=1}^{H} \frac{E^{w,h,c}_{x,s}}{E^{w,c}_{x,s}} I^{w,h,c}_{x,s}, \qquad I^{h,c}_{x,s} = \sum_{w=1}^{W} \frac{E^{w,h,c}_{x,s}}{E^{h,c}_{x,s}} I^{w,h,c}_{x,s}.$$

Unfortunately, these theoretical relationships are not preserved when assessed using estimated indices, with discrepancies tending to increase as the number of involved factors grows. To address this issue, we propose a generalization of the iterative proportional fitting procedure that incorporates weights. This leads to a new algorithm, referred to as weighted iterative proportional fitting (WIPF).

3. The weighted iterative proportional fitting algorithm

3.1. Problem setting

In this section, we detail WIPF for the case at hand, with three dimensions. For analogy with IPF, we adopt a contingency table representation and, without loss of generality, set the one-dimensional weighted sums to 1. The approach and notation can be straightforwardly generalized to any number of dimensions.

Let us assume three categorical variables X, Y and Z with R, C and L levels (from rows, columns and layers), respectively, and two 3-way arrays: an initial 3-way array, referred to as the seed, of non-negative initial indices/indicators I^0_{rcl}, $S = \left[I^0_{rcl}\right]$, and a 3-way contingency table of non-negative weights, $W = \left[w_{rcl}\right]$, where $r \in \{1,\dots,R\}$, $c \in \{1,\dots,C\}$ and $l \in \{1,\dots,L\}$ correspond to the levels of the first, second and third variable, respectively. In addition, let us suppose that these initial indices correspond to estimates of the true underlying indices, I_{rcl}, which satisfy a set of weighted sum-convex constraints across their marginal dimensions:

$$\sum_{r=1}^{R} \frac{w_{r++}}{w_{+++}} I_{r**} = \sum_{c=1}^{C} \frac{w_{+c+}}{w_{+++}} I_{*c*} = \sum_{l=1}^{L} \frac{w_{++l}}{w_{+++}} I_{**l} = 1,$$

$$\begin{aligned} I_{r**} &= \sum_{c=1}^{C} \frac{w_{rc+}}{w_{r++}} I_{rc*} = \sum_{l=1}^{L} \frac{w_{r+l}}{w_{r++}} I_{r*l}, \\ I_{*c*} &= \sum_{r=1}^{R} \frac{w_{rc+}}{w_{+c+}} I_{rc*} = \sum_{l=1}^{L} \frac{w_{+cl}}{w_{+c+}} I_{*cl}, \\ I_{**l} &= \sum_{r=1}^{R} \frac{w_{r+l}}{w_{++l}} I_{r*l} = \sum_{c=1}^{C} \frac{w_{+cl}}{w_{++l}} I_{*cl}, \end{aligned} \tag{1}$$

$$I_{rc*} = \sum_{l=1}^{L} \frac{w_{rcl}}{w_{rc+}} I_{rcl}, \qquad I_{r*l} = \sum_{c=1}^{C} \frac{w_{rcl}}{w_{r+l}} I_{rcl}, \qquad I_{*cl} = \sum_{r=1}^{R} \frac{w_{rcl}}{w_{+cl}} I_{rcl},$$

where $\mathcal{M} = \left\{I_{r**}, I_{*c*}, I_{**l}, I_{rc*}, I_{r*l}, I_{*cl}\right\}$ represents the set of (known) margin indices—for which consistency across the margins is also assumed—and sub-index $+$ refers to the standard notation of summation over the corresponding sub-index, e.g., $w_{r+l} = \sum_{c=1}^{C} w_{rcl}$. Note that in this case the margins I_{r**}, I_{*c*} and I_{**l} are vectors, and the margins I_{rc*}, I_{r*l} and I_{*cl} are matrices.

To facilitate interpretation of the notation, we present a small illustrative example with three categorical variables X, Y, and Z, each taking two levels. Table 1 shows the individual indices in a long-table format, with the last column displaying the corresponding 1D marginal weighted sums. Table 2 presents the 2D marginal weighted sums, arranged in three blocks corresponding to I_{rc*}, I_{r*l}, and I_{*cl}, where each block shows the weighted averages over the remaining dimension. The denominators in these expressions represent the appropriate sums of weights used in the marginal calculations. Together, these two tables provide a concrete illustration of how the 3-way array of indices relates to its marginal sums through the weights.

Table 1. *Illustrative 2×2×2 array of indices with 1D marginal weighted sums.*

X	Y	Z	I_{rcl}	1D Marginals
X_1	Y_1	Z_1	I_{111}	$I_{1**} = \frac{w_{111}I_{111}+w_{121}I_{121}+w_{112}I_{112}+w_{122}I_{122}}{w_{111}+w_{121}+w_{112}+w_{122}} = \frac{w_{111}I_{111}+w_{112}I_{112}+w_{121}I_{121}+w_{122}I_{122}}{w_{111}+w_{112}+w_{121}+w_{122}}$
X_1	Y_2	Z_1	I_{121}	$I_{2**} = \frac{w_{211}I_{211}+w_{221}I_{221}+w_{212}I_{212}+w_{222}I_{222}}{w_{211}+w_{221}+w_{212}+w_{222}} = \frac{w_{211}I_{211}+w_{212}I_{212}+w_{221}I_{221}+w_{222}I_{222}}{w_{211}+w_{212}+w_{221}+w_{222}}$
X_1	Y_1	Z_2	I_{112}	
X_1	Y_2	Z_2	I_{122}	$I_{*1*} = \frac{w_{111}I_{111}+w_{211}I_{211}+w_{112}I_{112}+w_{212}I_{212}}{w_{111}+w_{211}+w_{112}+w_{212}} = \frac{w_{111}I_{111}+w_{112}I_{112}+w_{211}I_{211}+w_{212}I_{212}}{w_{111}+w_{112}+w_{211}+w_{212}}$
X_2	Y_1	Z_1	I_{211}	$I_{*2*} = \frac{w_{121}I_{121}+w_{221}I_{221}+w_{122}I_{122}+w_{222}I_{222}}{w_{121}+w_{221}+w_{122}+w_{222}} = \frac{w_{121}I_{121}+w_{122}I_{122}+w_{221}I_{221}+w_{222}I_{222}}{w_{121}+w_{122}+w_{221}+w_{222}}$
X_2	Y_2	Z_1	I_{221}	
X_2	Y_1	Z_2	I_{212}	$I_{**1} = \frac{w_{111}I_{111}+w_{211}I_{211}+w_{121}I_{121}+w_{221}I_{221}}{w_{111}+w_{211}+w_{121}+w_{221}} = \frac{w_{111}I_{111}+w_{121}I_{121}+w_{211}I_{211}+w_{221}I_{221}}{w_{111}+w_{121}+w_{211}+w_{221}}$
X_2	Y_2	Z_2	I_{222}	$I_{**2} = \frac{w_{112}I_{112}+w_{212}I_{212}+w_{122}I_{122}+w_{222}I_{222}}{w_{112}+w_{212}+w_{122}+w_{222}} = \frac{w_{112}I_{112}+w_{122}I_{122}+w_{212}I_{212}+w_{222}I_{222}}{w_{112}+w_{122}+w_{212}+w_{222}}$

Table 2. *Illustrative 2×2×2 array: 2D marginal weighted sums.*

I_{rc*}		I_{r*l}		I_{*cl}	
$I_{11*} = \frac{w_{111}I_{111}+w_{112}I_{112}}{w_{111}+w_{112}}$	$I_{12*} = \frac{w_{121}I_{121}+w_{122}I_{122}}{w_{121}+w_{122}}$	$I_{1*1} = \frac{w_{111}I_{111}+w_{121}I_{121}}{w_{111}+w_{121}}$	$I_{1*2} = \frac{w_{112}I_{112}+w_{122}I_{122}}{w_{112}+w_{122}}$	$I_{*11} = \frac{w_{111}I_{111}+w_{211}I_{211}}{w_{111}+w_{211}}$	$I_{*12} = \frac{w_{112}I_{112}+w_{212}I_{212}}{w_{112}+w_{212}}$
$I_{21*} = \frac{w_{211}I_{211}+w_{212}I_{212}}{w_{211}+w_{212}}$	$I_{22*} = \frac{w_{221}I_{221}+w_{222}I_{222}}{w_{221}+w_{222}}$	$I_{2*1} = \frac{w_{211}I_{211}+w_{221}I_{221}}{w_{211}+w_{221}}$	$I_{2*2} = \frac{w_{212}I_{212}+w_{222}I_{222}}{w_{212}+w_{222}}$	$I_{*21} = \frac{w_{121}I_{121}+w_{221}I_{221}}{w_{121}+w_{221}}$	$I_{*22} = \frac{w_{122}I_{122}+w_{222}I_{222}}{w_{122}+w_{222}}$

The convex-weights are calculated by normalizing W in a margin-dependent manner, ensuring that when aggregated across the missing dimensions the normalized weights sum to one. In the `WIPF` package, this is controlled by the `normalize` argument. Alternatively, the relationships can be formulated without normalizing the weights. In this case, the requirement that all zero-dimensional target margins equal 1 is replaced by the weaker assumption that all one-dimensional weighted sums coincide. Moreover, in the special case when the weights are unitary, the problem collapses to the one corresponding to the standard IPF procedure.

Importantly, the normalized and non-normalized formulations are not, in general, equivalent up to a simple rescaling of the solution. While both formulations preserve the relative structure induced by the weights, the resulting fitted arrays tend to differ in absolute magnitude unless the weights are proportional within each marginal dimension. A small numerical illustration showing that the two formulations can lead to non-proportional solutions is provided in Appendix C. For the remainder of this exposition, we focus on the more complex case of weighted sum-convex constraints.

3.2. The algorithm

WIPF is the procedure that, given a non-empty subset $\mathcal{C} \subseteq \mathcal{M}$ of target margin constraints, iteratively updates the seed array based on the targets and weights. Although the full set of margins may not be available in a given problem, for completeness we detail below the full set of updating adjustments (steps). In particular, $\forall r, c, l$, steps at iteration $i \geq 1$ may be computed by the equations:

$$I_{rcl}^{(1)} = I_{rcl}^{i-1} \frac{I_{r**}}{I_{r**}^{i-1}}, \qquad I_{rcl}^{(2)} = I_{rcl}^{(1)} \frac{I_{*c*}}{I_{*c*}^{(1)}},$$

$$I_{rcl}^{(3)} = I_{rcl}^{(2)} \frac{I_{**l}}{I_{**l}^{(2)}}, \qquad I_{rcl}^{(4)} = I_{rcl}^{(3)} \frac{I_{rc*}}{I_{rc*}^{(3)}},$$

$$I_{rcl}^{(5)} = I_{rcl}^{(4)} \frac{I_{r*l}}{I_{r*l}^{(4)}}, \qquad I_{rcl}^{i} = I_{rcl}^{(5)} \frac{I_{*cl}}{I_{*cl}^{(5)}}.$$

where the temporal margins are calculated using the expressions defined in (1) applied to the temporarily updated indices.

Although these steps describe an updating process that first cycles through individual dimensions and then through pairs of dimensions, alternative cycling sequences are equally valid. If only a subset of the target margins is available, i.e., $\mathcal{C} \subset \mathcal{M}$, adjustments are restricted to those margins at each iteration. In other words, each iteration has as many steps as there are marginal configurations to adjust to. These iterations are repeated until the maximum difference between two consecutive iterations—whether in the indices or the constraints—does not exceed a pre-specified, sufficiently small threshold $\varepsilon > 0$. The values at the last iteration comprise the WIPF solution.

It should be noted that the WIPF solution cannot be obtained through the classical IPF procedure. Although one might attempt to recast the WIPF problem as an IPF problem—by defining the seed as the elementwise product of the initial seed array $S = [I_{rcl}^0]$ and the weight matrix $W = [w_{rcl}]$, and setting the target margins as the product of the intended constraints (for $\mathcal{C} \subseteq \mathcal{M}$) and the appropriate partial sums of W—this alternative formulation does not converge to the WIPF solution. While this can be easily illustrated with a numerical example (see Appendix A), the core reason lies in the treatment of the weights: in WIPF, weights remain fixed throughout the iterations, whereas in the IPF-based formulation, they are implicitly altered at each step. As a result, the indices produced by the latter are not weight-consistent.

Despite the differences between WIPF and classical IPF, as with classical IPF, if the initial table contains some $I_{rcl}^0 = 0$, the corresponding value will remain zero throughout the iterations. This property makes it straightforward to fit tables with structural zeros. Furthermore, in the same vein as IPF—which requires the target marginal totals to be internally consistent (i.e., in two dimensions, row and column sums must match) for convergence—WIPF requires the target marginal (convex) sums to be internally consistent. When this condition is met, convergence is typically achieved in very few iterations.

4. On the theoretical and computational properties of WIPF

In this section, we discuss the theoretical foundations of the Weighted Iterative Proportional Fitting (WIPF) algorithm. While Section 3 describes the algorithmic procedure and draws analogies with classical IPF, it is useful to formalize the conditions under which the algorithm is guaranteed to produce a well-defined solution, to clarify its uniqueness, and to relate it to standard convex optimization principles. We organize the discussion from three complementary perspectives, to subsequently end the section with computational considerations.

4.1. Existence, uniqueness, and convergence

As WIPF retains the typical multiplicative structure of IPF, it inherits the fundamental theoretical properties of IPF under standard regularity conditions (Bishop, Fienberg and Holland, 2007). Specifically, if the seed array is strictly positive and the weighted marginal constraints are mutually compatible, then a feasible solution satisfying all weighted sum constraints exists, is unique within the multiplicative family generated from the seed, and the iterative procedure converges to this solution. The justification follows from the fact that each WIPF step is a multiplicative scaling operation analogous to the updating steps of classical IPF, with margins computed as convex combinations determined by the weights. These properties hold for any set of non-negative weights, provided that the target margins are consistent and compatible.

4.2. Formulation as a weighted Kullback–Leibler minimization problem

WIPF can equivalently be expressed as the solution of a convex optimization problem that minimizes the weighted Kullback–Leibler (KL) divergence between the fitted array and the initial seed, subject to the weighted sum constraints. Formally, in the three dimensional case, if $S = \left[I^0_{ijk}\right]$, denotes the seed array and $W = \left[w_{ijk}\right]$ the array of non-negative weights, the optimization problem can be written as

$$\arg\min_{I_{ijk}>0} \sum_{i,j,k} w_{ijk} I_{ijk} \log\left(\frac{I_{ijk}}{I^0_{ijk}}\right) \quad \text{subject to the weighted marginal constraints.}$$

The strict convexity of the weighted KL objective and the convexity of the feasible set ensure the existence of a unique minimizer whenever the constraints are compatible. Note that the classical KL representation of IPF is recovered as the special case in which all weights are equal to one.

This equivalence provides an alternative perspective for understanding WIPF and facilitates connections with other convex optimization techniques. A simple example illustrating the equivalence between the solutions of this weighted KL optimization problem and the normalized WIPF solution is provided in Appendix B.

4.3. Bregman projection interpretation

Finally, WIPF can be also interpreted as a sequence of Bregman projections onto convex sets defined by the weighted marginal constraints, using the corresponding generalized weighted Kullback–Leibler divergence (Kurras, Greenewald and Grosse, 2015) as the generating function:

$$D^{w}_{\mathrm{KL}}(I \,\|\, I^0) = \sum_{i,j,k} w_{ijk} \left[I_{ijk} \log \frac{I_{ijk}}{I^0_{ijk}} - I_{ijk} + I^0_{ijk} \right].$$

Each iterative adjustment of a particular margin corresponds to projecting the current array onto the associated constraint set in the space of positive arrays, ensuring consistency while maintaining multiplicative relationships with the seed. From this perspective, WIPF can be seen as an alternating projection method, where the weighted KL divergence defines the geometry of the space. Convergence follows from general results on cyclic Bregman projections onto convex sets (Censor and Zenios, 1997), which apply because the weighted marginal sets are convex and compatible. This interpretation provides both a geometric intuition for convergence and a theoretical framework linking WIPF with a broader class of iterative projection methods in convex analysis.

4.4. Computational considerations

From an algorithmic standpoint, WIPF performs multiplicative updates across the entries of the array in a manner analogous to classical IPF. Consequently, the computational complexity of WIPF can be assessed drawing on the existing IPF literature. Each iteration requires $\mathcal{O}(T \cdot M)$ operations, where T is the total number of entries in the array and M is the number of marginal constraints applied per iteration (Bishop et al., 2007). Although the calculation of weighted marginal sums introduces additional per-cell multiplications compared to classical IPF, this only affects the constant factor and does not change the asymptotic complexity.

The number of iterations needed for convergence depends on the initial seed, the weight distribution, and the chosen tolerance, but is typically modest when the margins are compatible. These considerations suggest that WIPF is computationally tractable for most practical multiway arrays encountered in applied settings, with solutions typically obtained within a few seconds on a standard laptop, even for problems involving thousands of entries and dozens of constraints.

5. The `WIPF` package

The `WIPF` package provides a general and flexible implementation of the weighted iterative proportional fitting (WIPF) algorithm for arrays of arbitrary dimension, together with a set of auxiliary functions intended to facilitate its practical use. Its functions operate on an initial seed array and an array of non-negative weights of the same dimension. Given a set of target margins—possibly of different orders and with some margins missing—the algorithm iteratively updates the seed so that the weighted sums over the specified dimension index subsets match the supplied margins, whenever these are compatible with the weights. The implementation supports arbitrary combinations of one-dimensional, two-dimensional, and higher-order margins, making it suitable for high-dimensional applications.

The core functionality of the package is the `WIPF` function, which applies the algorithm to arrays of dimension $N \geq 2$. The one-dimensional case is not handled by `WIPF`, as it reduces to a trivial rescaling problem; this setting is instead covered by the function `WIPF1`. For convenience and computational efficiency, dedicated shortcuts are also provided for the two- and three-dimensional cases through the functions `WIPF2` and `WIPF3`, respectively. These functions follow the same logic as the general routine but exploit the specific structure of low-dimensional arrays, requiring simpler inputs.

When the supplied margins are mutually incompatible given the weights, the package includes procedures to restore compatibility prior to the main fitting step. Such adjustments are not performed silently: the user is informed through a warning message whenever they occur, and the full output of the functions report the magnitude of the corresponding margin corrections. In such cases, lower-dimensional WIPF routines are applied recursively to adjust the margins themselves, following a well-defined updating order based on the dimensionality and ordering of the associated index sets. Lower-dimensional margins are made compatible first, with row-margins taking preference over column-margins, which in turn take precedence over layer-margins, and so on. This strategy ensures that the algorithm remains well-defined and convergent even when the initial margin specifications cannot be jointly satisfied. Convergence is ensured because each pre-adjustment applies `WIPF` to a lower-dimensional problem, which inherits the same theoretical properties described in Section 4.

As one of the intended uses of WIPF is the reconciliation of statistical indices, it is not uncommon to encounter slightly incompatible margins due to statistical uncertainty or, as illustrated in the example presented in the next section, to rounding in both the weights and the initial estimated indices. In this context, the availability of automated compatibility adjustments is particularly valuable. Furthermore, when the weighted sum of one-dimensional marginal indices does not equal one, `WIPF1` can be used to enforce this condition, since margins of dimension zero are not allowed in `WIPF`. The functions also incorporate utilities to check dimensional consistency across inputs and to assess margin deviations on convergence.

To support data manipulation, the package includes auxiliary functions to convert between array- and tabular-based representations. In particular, `array2df` transforms an N-dimensional array into a long-format data frame, while `df2array` performs the inverse operation, reconstructing an array from a data frame with factor columns and a column of values. These utilities are especially useful when seed arrays or margins are obtained from external data sources in tabular form, or when `WIPF` outputs need to be exported to other environments in tabular form.

The inputs required by the `WIPF` function are summarized in Table 3. These include an initial seed array, the associated weights, a list of target margins, and a set of indices identifying the dimensions to which each margin corresponds. Additional arguments control normalization, convergence tolerance, and the maximum number of iterations.

Table 3. *Inputs of the `WIPF` function.*

Argument	**Description**
`seed`	An *N*-dimensional array of non-negative values providing the initial configuration.
`weights`	An *N*-dimensional array of non-negative weights associated with the entries of `seed`.
`margins`	A list of lower-dimensional arrays containing the target weighted margins.
`indices`	A list of index vectors identifying the dimensions corresponding to each element of `margins`.
`normalize`	Logical indicator specifying whether weights are normalized before computing weighted sums.
`tol`	Convergence tolerance for the iterative algorithm.
`maxit`	Maximum number of allowed iterations.
`full`	Logical indicator specifying whether the more complete output should be saved.

Source: compiled by the authors.

Table 4. *Outputs of the `WIPF` function.*

Output	**Description**
`sol`	An array with the same dimension as `seed`, containing the solution at convergence (or at the final iteration).
`iter`	Number of iterations performed by the algorithm.
`margins`	A list of arrays with the margins effectively used to reach the solution; margins that are compatible with the weights coincide with the original inputs.
`dev.margins`	A list of arrays, structured as `margins`, containing the maximum absolute deviations between target margins and the corresponding weighted sums of `sol`.
`dev.congruence`	A list of arrays, structured as `margins`, containing the differences between the original target margins and the adjusted margins actually used in the fitting procedure.
`inputs`	A list collecting all input objects used in the call to `WIPF`.

Source: compiled by the authors.

The full output of the function (see Table 4) includes the fitted array, the number of iterations required for convergence, and detailed information on the final deviations: between the original and (if necessary) adjusted target margins and between the margins finally used as constraints and corresponding weighted sums implied by the fitted array.

In short, the `WIPF` functions are designed to work directly with multidimensional arrays and collections of lower-dimensional margins, following the theoretical framework described in the previous sections. Taken together, these components make the `WIPF` package a flexible and transparent tool for applying weighted proportional fitting methods in multidimensional settings.

6. `WIPF` in action: adjusting death-risk indices

To illustrate the procedure and the use of the package, we adjust raw initial estimates of risk indices for Spanish males aged 70, cross-classified by four levels of habitat size (`H1`, `H2`, `H3`, `H4`) and four levels of contextual wealth (`W1`, `W2`, `W3`, `W4`). Table 5 reports the initial and WIPF-adjusted estimates in the left and right panels, respectively, rounded to four decimal places. The central panel provides the weights used in the adjustment process.

Box 1 presents the reproducible code, which starts by installing the package (line 1) and loading it in the active R session (line 2). The `WIPF` package is available under the General Public License (GPL ≥ 2) on the Comprehensive R Archive Network (CRAN) at `<https://CRAN.R-project.org/package=WIPF>`.

Since, due to rounding, the initial row and column marginal death-risk estimates are incompatible with the weights as their weighted sums do not match unity exactly, WIPF in one dimension (`WIPF1`) is first applied to adjust the margins to unitary weighted sums (see lines 7 and 8 in Box 1). Subsequently, two-dimensional WIPF (`WIPF2`) is used to obtain fully compatible two-factor death-risk indices (line 9).

More generally, in practical applications we recommend performing the adjustments hierarchically, as implemented by default in the `WIPF` functions: first ensuring that the weighted sum of first-order estimates equals the dimension-one margins; next, adjusting second-order estimates using the corrected first-order values; and proceeding analogously for higher-order terms.

After this process, final estimates of the marginal and joint risk indices for habitat size and contextual wealth are obtained (see Table 5). For instance, the final estimates indicate that among men aged 70, and relative to the average, the wealthiest men living in rural areas exhibit a mortality risk that is almost 18% lower, whereas the poorest men in urban areas face a risk that is more than 22% higher.

To quantify the magnitude of the adjustment introduced by WIPF in this example, we compare the raw and adjusted indices. The relative changes are generally small, with an average absolute relative change of about 1.15%. The largest adjustment corresponds to a relative increase of approximately 3.53%, while most entries change by less than 1.30%. These results indicate that the procedure preserves the overall structure of the original indices while ensuring compatibility with the imposed margins. Naturally, the magnitude of the adjustments is case-dependent and reflects how compatible the original indices are with the marginal constraints given the weights.

Code example

```
install.packages("WIPF")
library(WIPF)
# Initial row margins
row.margin0 <- c(0.9687, 1.0286, 1.0097, 0.9921)
# Initial column margins
col.margin0 <- c(1.1388, 1.0366, 0.9702, 0.8549)
# Initial inner indices:  seed
cross0 <- matrix(c(1.0487, 0.9835, 0.9175, 0.8297,
                       1.1123, 1.0653, 0.9949, 0.8733,
                       1.1986, 1.0461, 0.9773, 0.8517,
                       1.1812, 1.0594, 1.0023, 0.8578),
                       nrow = 4, byrow = TRUE)
# Weights
weights <- matrix(c(140378, 149253, 112978, 54399,
                        142328, 149483, 114749, 80757,
                        133018, 120618, 133842, 150811,
                        67276, 74263, 91499, 213701),
                        nrow = 4, byrow = TRUE)
# Adjustment of margins, to unitary weighted sum
row.margin <- WIPF1(seed = row.margin0,
                        weights = rowSums(weights),
                        margin = 1)
col.margin <- WIPF1(seed = col.margin0,
                        weights = colSums(weights),
                        margin = 1)
# Adjustment of inner indices
cross <- WIPF2(seed = cross0, weights = weights,
                  margin1 = row.margin,
                  margin2 = col.margin)
```

Box 1: Reproducible code for the example developed in Table 5.

For clarity of exposition, the numerical example presented here focuses on the two-dimensional case. Although the motivating application is inherently multidimensional, displaying higher-dimensional contingency structures within the constraints of a two-dimensional page would require large and cumbersome tables that may obscure the main ideas. Hence, interested readers are referred to the accompanying documentation of the functions `WIPF` and `WIPF3` in the package (Pavía, 2026) for illustrative examples of three-dimensional applications of the algorithm.

Table 5. *Example of adjusted death-risk indices using WIPF for Spanish males aged 70.*

	Raw initial estimates					Weights				Final adjusted estimates				
	W1	**W2**	**W3**	**W4**		**W1**	**W2**	**W3**	**W4**	**W1**	**W2**	**W3**	**W4**	
H1	1.0487	0.9835	0.9175	0.8297	0.9687	140378	149253	112978	54399	1.0563	0.9818	0.9115	0.8201	0.9680
H2	1.1123	1.0653	0.9949	0.8733	1.0286	142328	149483	114749	80757	1.1189	1.0621	0.9871	0.8621	1.0279
H3	1.1986	1.0461	0.9773	0.8517	1.0097	133018	120618	133842	150811	1.2066	1.0437	0.9703	0.8414	1.0090
H4	1.1812	1.0594	1.0023	0.8578	0.9921	67276	74263	91499	213701	1.2229	1.0871	1.0235	0.8715	0.9914
	1.1388	1.0366	0.9702	0.8549						1.1393	1.0371	0.9707	0.8553	

Source: compiled by the authors using data from Spain between 2010 and 2019.

Note: The final solution reflects adjustments to both the marginal and inner indices. The indices were adjusted hierarchically: first, the initial row and column marginal death-risk estimates were scaled to yield unitary weighted sums; then, the inner estimates were adjusted conditionally on these corrected margins.

7. Conclusions

The iterative proportional fitting (IPF) procedure is a widely-established algorithm for adjusting an initial cross-classification of counts to align with specified target marginal distributions while preserving the original association structure. Despite numerous generalizations of IPF proposed over time, a method that allows differential weighting of cells has been lacking. This paper addresses this gap by introducing the Weighted Iterative Proportional Fitting (WIPF) algorithm and implementing it in an R package of the same name. This novel approach can be applied to adjust an initial set of indices/indicators subject to weighted sum-convex constraints across marginal dimensions.

The WIPF algorithm can be applied to adjust multidimensional sets of indices or indicators while preserving marginal consistency. For example, it can be used to reconcile economic indicators defined across multiple dimensions, such as region and demographic group, or to harmonize poverty measures disaggregated simultaneously by region and age group.

Another possible application arises in consumer choice modeling (Qu et al., 2025), where data can be arranged in a two-dimensional array representing quantities of products purchased across different consumer groups. This array may need to be adjusted to match a pre-specified marginal consisting of total product sales. In this context, weights can represent product prices, and the balancing procedure ensures consistency between observed quantities, prices, and aggregate spending patterns.

An important avenue for future research concerns the systematic study of the statistical properties of the WIPF procedure. In many practical applications, the initial seed arrays used as inputs are themselves estimated from data and therefore subject to sampling variability. Understanding how such uncertainty propagates through the WIPF adjustment, and how it affects the resulting reconciled indices, deserves further investigation. In particular, simulation-based studies could explore how the performance of WIPF varies with the number of dimensions, the number of categories across margins, the structure of the weights, and the magnitude of sampling variability in the initial indices. Such analyses would provide valuable insights into the bias, variance, and robustness properties of the method in different empirical settings.

Another promising direction concerns the extension of the method to settings with missing or unreliable cells in the initial arrays. In many empirical contexts, some cells may be unobserved or estimated with different levels of reliability. Incorporating cell-specific measures of uncertainty or confidence into the adjustment procedure could allow the algorithm to regulate the magnitude of changes across cells according to the quality of the underlying information. Developing such reliability-aware reconciliation procedures represents a natural extension of the WIPF framework.

Appendix A. Numerical counterexample on the non-equivalence between recast IPF and WIPF

A natural question is whether the weighted iterative proportional fitting (WIPF) procedure proposed in this paper can be reproduced by applying the classical IPF algorithm to a suitably transformed problem. In particular, one might conjecture that the weighted problem can be converted into a standard IPF problem by redefining the seed as the elementwise product of the original seed and the weights, and redefining the margins as the product of the initial target margins and the corresponding partial sums of weights.

To illustrate the non-equivalence between the two approaches, consider the following simple numerical example. Let the seed matrix and the weights be

$$S = \begin{pmatrix} 0.8 & 1.3 \\ 1.6 & 0.9 \end{pmatrix}, \qquad W = \begin{pmatrix} 500 & 9000 \\ 8000 & 1000 \end{pmatrix}.$$

Suppose that the weighted row and column constraints are both equal to the vector $(1,1)$. Then the solutions obtained using the recast IPF formulation and the WIPF procedure (rounded to four decimals) are

$$\hat{S}_{\mathrm{rIPF}} = \begin{pmatrix} 0.0133 & 0.5002 \\ 0.4462 & 0.0403 \end{pmatrix}, \qquad \hat{S}_{\mathrm{WIPF}} = \begin{pmatrix} 0.4922 & 1.0282 \\ 1.0317 & 0.7461 \end{pmatrix}.$$

The two fitted matrices are clearly different. This simple example therefore shows that WIPF cannot, in general, be reproduced by applying classical IPF to a reweighted seed and correspondingly transformed margins.

The results can be easily verified using the reproducible code provided in Box 2.

Reproducible code Appendix A

```
# Data
seed <- matrix(c(0.8, 1.3, 1.6, 0.9),
                    ncol = 2, byrow = TRUE)
weights <- matrix(c(500, 9000, 8000, 1000),
                      ncol = 2, byrow = TRUE)
margin.c <- margin.r <- c(1, 1)
weights <- weights/sum(weights)

# IPF recast solution
install.packages("mipfp")
library(mipfp)
sol.rIPF <- Ipfp(seed * weights,
                    target.list = list(1, 2),
                    target.data = list(margin.r *
                         rowSums(weights),
                         margin.c * colSums(weights)) )
round(sol.rIPF[[1]], 4)

# WIPF solution
install.packages("WIPF")
library(WIPF)
sol.WIPF <- WIPF2(seed, weights, margin.r,
                      margin.c)
round(sol.WIPF, 4)
```

Box 2: Reproducible code for the example developed in Appendix A.

Appendix B. Numerical illustration: Weighted KL formulation of WIPF

This appendix illustrates, using the example introduced in Appendix A, that the weighted iterative proportional fitting (WIPF) solution can be obtained, when no null indices are involved, as the solution of a weighted Kullback–Leibler (KL) minimization problem whose objective function is given by

$$\min_{I_{ij}>0} \sum_{i,j} w_{ij} I_{ij} \log \frac{I_{ij}}{I_{ij}^0},$$

subject to the corresponding weighted marginal constraints. In our example, these constraints are $(I_{11}w_{11}+I_{12}w_{12})/(w_{11}+w_{12}) = 1$, $(I_{21}w_{21}+I_{22}w_{22})/(w_{21}+w_{22}) = 1$, $(I_{11}w_{11}+I_{21}w_{21})/(w_{11}+w_{21}) = 1$, and $(I_{12}w_{12}+I_{22}w_{22})/(w_{12}+w_{22}) = 1$.

This result can be easily illustrated using the reproducible code provided in Box 3, which solves the corresponding KL-based optimization problem in R using the `Rsolnp` package.

Reproducible code Appendix B

```
# Package
install.packages("Rsolnp")
library(Rsolnp)

# Data
I0v <- c(0.8, 1.3, 1.6, 0.9)
wv <- c(500, 9000, 8000, 1000)
wv <- wv / sum(wv)

# Objective function
fobj <- function(x) sum(wv * x * log(x/I0v))

# Constraints
heq <- function(x){
 c( (wv[1]*x[1] + wv[2]*x[2]) / sum(wv[1:2]) - 1,
    (wv[3]*x[3] + wv[4]*x[4]) / sum(wv[3:4]) - 1,
    (wv[1]*x[1] + wv[3]*x[3]) / sum(wv[c(1,3)]) - 1) }

# Solution
sol <- Rsolnp::solnp(
 pars = c(0.5,0.5,0.5,0.5),
 fun = fobj,
 eqfun = heq,
 eqB = rep(0,3),
 LB = rep(1e-12,4) }

matrix(round(sol[[1]], 4), 2, 2)
```

Box 3: Reproducible code for the example formulated as a weighted KL optimization problem.

Appendix C. Effect of weight normalization on WIPF solutions

In the WIPF algorithm, the choice of normalized versus non-normalized weights systematically affects the resulting solution. Normalized weights ensure that, for each marginal dimension, the sum of the weights across the missing dimensions equals one. In this formulation, the target margins are preserved exactly in the scale of the seed array.

Non-normalized weights retain the original magnitudes of the weights. In this case, the resulting solution generally differs in scale from the normalized version: while the relative distribution of values within each marginal remains consistent with the weight ratios, the absolute magnitudes of the fitted entries may change. Equivalence up to a simple rescaling does not generally hold unless all weights along a given marginal are proportional. Therefore, practitioners should be aware that the `normalize` argument in the R implementation can produce genuinely different solutions.

In practice, the normalized formulation is recommended when the goal is to maintain comparability with classical IPF and to preserve the interpretability of the fitted array in the same units as the seed. Conversely, the non-normalized formulation may be preferable when absolute magnitudes need to reflect the raw weights.

To illustrate the non-equivalence up to a simple rescaling, we consider the same small numerical example analyzed in the previous appendixes and compare the WIPF solutions obtained with normalized and non-normalized weights.

As shown by the solutions of the example, the two solutions are not proportional, highlighting the practical impact of the normalization choice.

$$\hat{S}_{\text{nWIPF}} = \begin{pmatrix} 1.7293 & 1.9595 \\ 2.2044 & 0.8647 \end{pmatrix}, \qquad \hat{S}_{\text{WIPF}} = \begin{pmatrix} 0.4922 & 1.0282 \\ 1.0317 & 0.7461 \end{pmatrix}.$$

The corresponding R code to reproduce both solutions (normalized and non-normalized) is provided in Box 4.

Reproducible code Appendix C

```
# Data
seed <- matrix(c(0.8, 1.3, 1.6, 0.9),
                  ncol = 2, byrow = TRUE)
weights <- matrix(c(500, 9000, 8000, 1000),
                      ncol = 2, byrow = TRUE)
margin.c <- margin.r <- c(1, 1)
weights <- weights/sum(weights)


# WIPF non-normalized solution
install.packages("WIPF")
library(WIPF)
sol.nWIPF <- WIPF2(seed, weights, margin.r,
                      margin.c, normalize = FALSE)
round(sol.nWIPF, 4)


# WIPF normalized solution
sol.WIPF <- WIPF2(seed, weights, margin.r,
                      margin.c, normalize = TRUE)
round(sol.WIPF, 4)
```

Box 4: Reproducible code for the example developed in Appendix C.

Acknowledgements

The authors wish to thank two anonymous reviewers and the editor for valuable comments and suggestions, and Marie Hodkinson for revising the English of the paper. The usual disclaimers apply.

Funding

This work was supported by Generalitat Valenciana (Conselleria de Educación, Cultura y Universidades) under Grant CIACIO/2023/031; and Fundación MAPFRE under Grant "Modelización espacial e intra-anual de la mortalidad en España. Una herramienta automática para el cálculo de productos de vida".

Disclosure statement

The authors report there are no competing interests to declare.

References

Allen-Zhu, Z., Li, Y., Oliveira, R. and Wigderson, A. (2017). Much faster algorithms for matrix scaling. *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, 890–901.

Bell, M. L., O'Neill, M. S. and Zanobetti, A. (2024). Climate change, extreme heat, and health. *New England Journal of Medicine* 390, 1793–1801.

Bishop, Y., Fienberg, S. E. and Holland, P. W. (2007). *Discrete Multivariate Analysis: Theory and Practice*. Springer.

Censor, Y. and Zenios S. A. (1997). *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press.

Cohen, S. A., Nash, C. C., Byrne, E. N. and Greaney, M. L. (2023). Income and rural-urban status moderate the association between income inequality and life expectancy in US census tracts. *Journal of Health, Population and Nutrition* 42, 24.

Coons, J. Y., Langer, C. and Ruddy, M. (2024). Classical iterative proportional scaling of log-linear models with rational maximum likelihood estimator. *International Journal of Approximate Reasoning* 164, 109043.

Deming, W. E. and Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics* 11, 427–444.

Deville, J. C., Särndal, C.-E. and Sautory, O. (1993). Generalized raking procedures in survey sampling. *Journal of the American Statistical Association* 88, 1013–1020.

Dwyer-Lindgren, L., Baumann, M. M., Li, Z., Kelly, Y. O., Schmidt, C., Searchinger, C., La Motte-Kerr, W., Bollyky, T. J., Mokdad, A. H. and Murray, C. J. (2024). Ten

Americas: A systematic analysis of life expectancy disparities in the USA. *The Lancet* 404(10469), 2299–2313.

European Commission (2012). Guidelines on the application of Council Directive 2004/113/EC to insurance, in the light of the judgment of the Court of Justice of the European Union in Case C-236/09 (Test-Achats). *Official Journal of the European Union* C, 1–11.

Fournier Gabela, J. G. (2020). On the accuracy of gravity-RAS approaches used for inter-regional trade estimation: evidence using the 2005 inter-regional input–output table of Japan. *Economic Systems Research* 32, 521–539.

García-León, D., Masselot, P., Mistry, M. N., Gasparrini, A., Motta, C., Feyen, L. and Ciscar, J. C. (2024). Temperature-related mortality burden and projected changes under climate change scenarios. *The Lancet Public Health* 9, e644–e653.

Idel, M. (2016). A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. *arXiv preprint* arXiv:1609.06349.

Kalantari, B. and Khachiyan, L. (1996). On the complexity of nonnegative-matrix scaling. *Linear Algebra and its Applications* 240, 87–103.

Kurras, P., Greenewald, K. and Grosse, R. (2015). A simple and provably good projection update for iterative proportional fitting. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 38, 570–578. PMLR.

Klimova, A. and Rudas, T. (2015). Iterative proportional scaling for curved exponential families. *Scandinavian Journal of Statistics* 42, 832–847.

Kruithof, J. (1937). Telefoonverkeersrekening. *De Ingenieur* 52, E15–E25.

Lagravinese, R., Liberati, P. and Resce, G. (2020). Measuring health inequality in US: A composite index approach. *Social Indicators Research* 147, 921–946.

Lledó, J. and Pavía, J. M. (2026). Integrating income-indexes into life insurance and financial products. *Financial Innovation* 12, 94.

Masselot, P., Mistry, M. N., Rao, S., Huber, V., Monteiro, A., Samoli, E., Stafoggia, M., de'Donato, F., Garcia-Leon, D., Ciscar, J. C., Feyen, L., Schneider, A., Katsouyanni, K., Vicedo-Cabrera, A. M., Aunan, K. and Gasparrini, A. (2025). Estimating future heat-related and cold-related mortality under climate change, demographic and adaptation scenarios in 854 European cities. *Nature Medicine* 31, 1294–1302.

McMaughan, D. J., Oloruntoba, O. and Smith, M. L. (2020). Socioeconomic status and access to healthcare: Interrelated drivers for healthy aging. *Frontiers in Public Health* 8, 231.

Pavía, J. M. (2026). *WIPF: Weighted Iterative Proportional Fitting*. R-package, version 0.1.0-4. `https://CRAN.R-project.org/package=WIPF`.

Pavía, J. M., Cabrer, B. and Sala, R. (2009). Updating Input-Output matrices: Assessing alternatives through simulation. *Journal of Statistical Computation and Simulation* 79, 1467–1498.

Pavía, J. M. and Lledó, J. (2022). Estimation of the combined effects of ageing and seasonality on mortality risk: An application to Spain. *Journal of the Royal Statistical Society Series A: Statistics in Society* 185(2), 471–497.

Pavía, J. M., Lledó, J., Espinosa, P. and Pérez, V. (2026). *Modelización espacial e intra-anual de la mortalidad en España. Una herramienta automática para el cálculo de productos de vida*. Madrid: Fundación Mapfre.

Pavía, J. M., Lledó, J. and Roig, R. (2026). The cost of inequality: Income and life expectancy at birth in Spain. *Health Policy* 169, 105616.

Qu, Z., Galichon, A., Gao, W. and Ugander, J. (2025). On Sinkhorn's algorithm and choice modeling. *Operations Research*, online available. 10.1287/opre.2023.0596.

Suesse, T., Namazi-Rad, M. R., Mokhtarian, P. and Barthélemy, J. (2017). Estimating cross-classified population counts of multidimensional tables: An application to regional Australia to obtain pseudo-census counts. *Journal of Official Statistics* 33, 1021–1050.

von Lindheim, J. and Steidl, G. (2023). Generalized iterative scaling for regularized optimal transport with affine constraints: Application examples. *arXiv preprint* arXiv:2305.07071v1.

Wiebe, K. S. and Lenzen, M. (2016). To RAS or not to RAS? What is the difference in outcomes in multi-regional input–output models? *Economic Systems Research* 28, 383–402.

Zaloznik, M. (2011). *Iterative Proportional Fitting. Theoretical Synthesis and Practical Limitations*. PhD dissertation, University of Oxford.