The Journal of Open Source Software

piar: Price Index Aggregation in R

Steve Martin [1]

1 Statistics Canada, Ottawa, ON, Canada

DOI: 10.21105/joss.06781

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @schneiderpy
- @realxinzhao

Submitted: 18 April 2024
Published: 06 September 2024

Summary

The systematic inflation of prices for goods and services over time is a ubiquitous feature of modern economies. A price index is the main empirical tool to measure changes in prices over time, and consequently price indexes are a core macroeconomic statistic produced by national statistical agencies to measure and study inflation. Price indexes are often made with a two-step procedure—especially those coming from national statistical agencies—where period-over-period indexes are calculated for a large collection of well-defined goods and services at each point in time and aggregated according to a hierarchical structure (ILO et al., 2004; IMF et al., 2020). These aggregated indexes can then be chained together to form a collection of time series that give the evolution of prices in the economy with respect to a fixed point in time.

`piar` is an R (R Core Team, 2024) package for aggregating price indexes. It contains a collection of functions that revolve around the usual two-step work flow for computing price indexes, making it easy to build large, hierarchical indexes using the methods described in the literature (e.g., Balk, 2008; IMF et al., 2020; von der Lippe, 2007). `piar` is designed to be useful for both researching new sources of data and methods to construct price indexes, and the regular production of price statistics. It is targeted towards economists, statisticians, and data scientists working at national statistical agencies, central banks, financial institutions, and in academia that want to measure and study the evolution of prices over time. `piar` is currently used to produce several price indexes at Statistics Canada (e.g., Mustafa, 2023a, 2023b).

Statement of need

`piar` fills a gap in the open-source ecosystem for measuring inflation by providing a tool to build large, hierarchical price indexes. There are several R and Python packages for accessing price indexes published by national statistical agencies (e.g., von Bergmann & Shkolnik, 2021; Welsh, 2024), but these are not suitable for computing new indexes or researching new methods and data sources to measure inflation.

To compute a price index, the `micEconIndex` (Henningsen, 2022) and `IndexNumbers` (Saavedra-Nieves & Saavedra-Nieves, 2021) packages implement text-book index-number formulas to measure the change in prices over time for a collection of goods and services. These methods, however, are seldom directly used in practice; for example, most text-book index-number formulas use information on both prices and quantities over time, but quantity information is rarely available in most cases. The `IndexNumR` (White, 2023) and `PriceIndices` (Białek, 2024) packages implement a variety of more sophisticated methods to build price indexes with high-frequency retail scanner data (in addition to the text-book methods). Similarly, the `hpiR` (Krause, 2020) and `rsmatrix` (Martin, 2023) packages implement methods that are applicable to housing data. Although these more advanced methods are directly used in practice to make price indexes for certain types of goods and services, they are not suitable for constructing the conventional price indexes that cover a wide range of products or industries.

The Journal of Open Source Software

In contrast to existing tools, piar provides a flexible interface to build large, hierarchical price indexes over time in a way that is suitable for both the data and methods used to make large-scale measures of inflation. Part of this flexible design means that pre-computed indexes can serve as an input to a larger index, alongside other sources of price data (e.g., survey data), and so piar complements existing tools in R by providing a framework for integrating different index-number methods with heterogeneous sources of data.

Example

The goal of this example is to illustrate some of the core features of piar by building a typical industry price index with synthetic data. The built-in ms_prices dataset has random price data for five businesses over four quarters, and the ms_weights dataset contain weights that give the relative importance of these businesses in their respective industries. Note that these data have a fairly realistic pattern of missing data (e.g., there are no prices for businesses B5), and, although small, are emblematic of the kinds of survey data used to measure inflation.

```
head(ms_prices)
```

```
#>   period business product price
#> 1 202001       B1       1  1.14
#> 2 202001       B1       2    NA
#> 3 202001       B1       3  6.09
#> 4 202001       B2       4  6.23
#> 5 202001       B2       5  8.61
#> 6 202001       B2       6  6.40
```

```
ms_weights
```

```
#>   business classification weight
#> 1       B1             11    553
#> 2       B2             11    646
#> 3       B3             11    312
#> 4       B4             12    622
#> 5       B5             12    330
```

The first step to build a price index with these data is to make business-level indexes—so called elemental or elementary indexes—that serve as the building blocks for the industry-level indexes. The elemental_index() function makes elemental indexes using information on the change in price for the products sold by each business (price relatives) in each quarter. By default elemental_index() makes a Jevons index, but any bilateral generalized-mean index is possible. Note that price data here are in levels, not changes, but the price_relative() function can make the necessary conversion and construct a numeric vector of price relatives for each product. Missing price relatives are ignored by setting na.rm = TRUE.

```
elementals <- ms_prices |>
  transform(
    relative = price_relative(price, period = period, product = product)
  ) |>
  elemental_index(relative ~ period + business, na.rm = TRUE)
```

```
elementals
```

```
#> Period-over-period price index for 4 levels over 4 time periods
#>     202001    202002    202003   202004
#> B1       1 0.8949097 0.3342939      NaN
#> B2       1       NaN       NaN 2.770456
#> B3       1 2.0200036 1.6353355 0.537996
#> B4     NaN       NaN       NaN 4.576286
```

With the elemental indexes out of the way, it's time to transform the weights to make a price-index aggregation structure that maps each business to its position in the hierarchical industry structure. Each business has a two-digit industry classification that's first unpacked with the `expand_classification()` function to make the aggregation hierarchy and then combined with the weights to make an aggregation structure.

```
ms_weights[c("level1", "level2")] <-
  expand_classification(ms_weights$classification)

pias <- ms_weights[c("level1", "level2", "business", "weight")]

pias
```

```
#>   level1 level2 business weight
#> 1      1     11       B1    553
#> 2      1     11       B2    646
#> 3      1     11       B3    312
#> 4      1     12       B4    622
#> 5      1     12       B5    330
```

It is now simple to aggregate the business-level indexes according to this aggregation structure with the `aggregate()` function and chain them together to get the evolution of prices over time. Missing elemental indexes are ignored by setting `na.rm = TRUE`.

```
ms_index <- aggregate(elementals, pias, na.rm = TRUE)

chain(ms_index)
```

```
#> Fixed-base price index for 8 levels over 4 time periods
#>    202001    202002    202003    202004
#> 1       1 1.3007239 1.3827662 3.7815355
#> 11      1 1.3007239 1.3827662 2.1771866
#> 12      1 1.3007239 1.3827662 6.3279338
#> B1      1 0.8949097 0.2991629 0.4710366
#> B2      1 1.3007239 1.3827662 3.8308934
#> B3      1 2.0200036 3.3033836 1.7772072
#> B4      1 1.3007239 1.3827662 6.3279338
#> B5      1 1.3007239 1.3827662 6.3279338
```

In addition to calculating the aggregated industry indexes (the first three rows above), missing elemental indexes have been filled in to ensure that the index can be chained over time while remaining consistent in aggregation. There are a variety of methods to manipulate the index objects that come from these computations in order to compose more complex workflows for building price indexes.

Acknowledgements

I am grateful for many useful conversations with the research staff at Statistics Canada, and in particular with Xin Ha, that have improved `piar`.

The views and opinions expressed in this paper are my own and do not necessarily reflect those of the Government of Canada.

References

Balk, B. M. (2008). *Price and quantity index numbers*. Cambridge University Press. https://doi.org/10.1017/CBO9780511720758

Białek, J. (2024). *PriceIndices: Calculating bilateral and multilateral price indexes*. https://doi.org/10.32614/CRAN.package.PriceIndices

Henningsen, A. (2022). *micEconIndex: Price and quantity indices*. https://doi.org/10.32614/CRAN.package.micEconIndex

ILO, IMF, OECD, UNECE, & World Bank. (2004). *Producer price index manual: Theory and practice*. International Monetary Fund. https://doi.org/10.5089/9781589063044.069

IMF, ILO, Eurostat, UNECE, OECD, & World Bank. (2020). *Consumer price index manual: Concepts and methods*. International Monetary Fund. https://doi.org/10.5089/9781484354841.069

Krause, A. (2020). *hpiR: House price indexes*. https://doi.org/10.32614/CRAN.package.hpiR

Martin, S. (2023). *rsmatrix: Matrices for repeat-sales price indexes*. https://doi.org/10.5281/zenodo.10110159

Mustafa, M. (2023a). *Technical guide for the couriers and messengers services price index (CMSPI)*. Prices Analytical Series, Statistics Canada Catalogue no. 62F0014M.

Mustafa, M. (2023b). *Technical guide for the for-hire motor carrier freight services price index (FHMCFSPI)*. Prices Analytical Series, Statistics Canada Catalogue no. 62F0014M.

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Saavedra-Nieves, A., & Saavedra-Nieves, P. (2021). IndexNumber: An R Package for Measuring the Evolution of Magnitudes. *The R Journal*, *13*(1), 253–275. https://doi.org/10.32614/RJ-2021-038

von Bergmann, J., & Shkolnik, D. (2021). *cansim: Accessing Statistics Canada data table and vectors*. https://doi.org/10.32614/CRAN.package.cansim

von der Lippe, P. (2007). *Index theory and price statistics*. Peter Lang. https://doi.org/10.3726/978-3-653-01120-3

Welsh, B. (2024). *cpi*. https://github.com/palewire/cpi

White, G. (2023). *IndexNumR: Index number calculation*. https://doi.org/10.32614/CRAN.package.IndexNumR