Validation and speed

Muhammad Yaseen

2026-11-05

Table of contents

1 Abstract **1**

2 Against version 0.5.0 **1**

3 Against published index numbers **2**

4 Identities **5**

5 Speed **5**

6 References **6**

1 Abstract

This vignette sets out how the results of the package are checked: against version 0.5.0, against the published index numbers of the Consumer Price Index Manual, against the identities the formulas must satisfy, and it ends with timings against version 0.5.0.

2 Against version 0.5.0

The example of version 0.5.0, run with a seed, was saved before the rewrite (`dev/legacy`), and the test `test-legacy.R` compares the new results with it. With `logbase = 10` (the default of 0.5.0) and with `logbase = 2` or e, and for the Hancock method, `DMAI` agrees to 10^{-10}. The one change is the default logarithm: 0.5.0 used base 10, so its default `DMAI` was $1/\log 10 \approx 0.434$ times the growth rate; 1.0.0 uses the natural logarithm, and the ratio of the two is $\log 10 \approx 2.303$:

```
data(money_sim)
new <-
  dmai(
```

```
      .data     = money_sim
    , method    = c("Barnett", "Hancock", "Fisher", "SimpleSum", "CE")
    , logbase   = NULL
    , date      = "Date"
    , x         = NULL
    , r         = NULL
    , user_cost = NULL
    , benchmark = "rb"
    , percent   = TRUE
    , base      = 100
    , as_dt     = TRUE
    )
old <-
  dmai(
      .data     = money_sim
    , method    = c("Barnett", "Hancock", "Fisher", "SimpleSum", "CE")
    , logbase   = 10
    , date      = "Date"
    , x         = NULL
    , r         = NULL
    , user_cost = NULL
    , benchmark = "rb"
    , percent   = TRUE
    , base      = 100
    , as_dt     = TRUE
    )
head(
    data.table::data.table(
        Date    = new$Date
      , natural = new$DMAI
      , base10  = old$DMAI
      , ratio   = new$DMAI / old$DMAI
      )
  )
```

```
         Date    natural     base10    ratio
1: 2005-01-01         NA         NA       NA
2: 2005-02-01  0.8504011  0.3693245 2.302585
3: 2005-03-01  1.7328708  0.7525762 2.302585
4: 2005-04-01 -0.2494176 -0.1083207 2.302585
5: 2005-05-01  0.3359143  0.1458857 2.302585
6: 2005-06-01  0.8085785  0.3511612 2.302585
```

3 Against published index numbers

A Divisia quantity index is the index of Tornqvist with the user costs in the place of prices; the formula is symmetric in prices and quantities, so with the

prices as `x` and the quantities as `user_cost` the package computes a Tornqvist price index. The Consumer Price Index Manual (ILO et al. 2004, Chapter 19) gives the chained Laspeyres, Paasche, Fisher and Tornqvist price indexes of an artificial data set, shipped as `cpi_ch19`:

```
data(cpi_ch19)
cpi_ch19
```

```
  period  p1  p2  p3  p4  p5  p6  q1  q2  q3   q4  q5  q6
1      1 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 2.0  1.0 4.5 0.5
2      2 1.2 3.0 1.3 0.7 1.4 0.8 0.8 0.9 1.9  1.3 4.7 0.6
3      3 1.0 1.0 1.5 0.5 1.7 0.6 1.0 1.1 1.8  3.0 5.0 0.8
4      4 0.8 0.5 1.6 0.3 1.9 0.4 1.2 1.2 1.9  6.0 5.6 1.3
5      5 1.0 1.0 1.6 0.1 2.0 0.2 0.9 1.2 2.0 12.0 6.5 2.5
```

```
tor <-
  dmai(
      .data     = cpi_ch19
    , method    = "Barnett"
    , logbase   = NULL
    , date      = "period"
    , x         = paste0("p", 1:6)
    , r         = NULL
    , user_cost = paste0("q", 1:6)
    , benchmark = NULL
    , percent   = TRUE
    , base      = 1
    , as_dt     = TRUE
    )
fis <-
  dmai(
      .data     = cpi_ch19
    , method    = "Fisher"
    , logbase   = NULL
    , date      = "period"
    , x         = paste0("p", 1:6)
    , r         = NULL
    , user_cost = paste0("q", 1:6)
    , benchmark = NULL
    , percent   = TRUE
    , base      = 1
    , as_dt     = FALSE
    )
data.table::data.table(
    period    = 1:5
  , laspeyres = round(x = cumprod(c(1, fis$fisher$laspeyres[-1])), digits
 ↪  = 4)
  , paasche   = round(x = cumprod(c(1, fis$fisher$paasche[-1])), digits =
 ↪  4)
```

```
  , fisher    = round(x = fis$index$index, digits = 4)
  , tornqvist = round(x = tor$index, digits = 4)
  )
```

```
   period laspeyres paasche fisher tornqvist
1:      1    1.0000  1.0000 1.0000    1.0000
2:      2    1.4200  1.3824 1.4011    1.4052
3:      3    1.3646  1.2740 1.3185    1.3112
4:      4    1.3351  1.2060 1.2689    1.2624
5:      5    1.3306  1.1234 1.2226    1.2224
```

The Manual prints (Tables 19.5 and 19.11): Laspeyres 1.4200, 1.3646, 1.3351, 1.3306; Paasche 1.3823, 1.2740, 1.2060, 1.1234; Fisher 1.4011, 1.3185, 1.2689, 1.2226; Tornqvist 1.4052, 1.3112, 1.2624, 1.2224. Every value agrees to the printed digits except the Paasche index of period 2, 14.10/10.20 = 1.38235, printed as 1.3823. Each value is a row of `inst/extdata/reference.csv`, checked at every `R CMD check`:

```
ref <-
  dmai_reference(
      source   = NULL
    , evaluate = TRUE
    , level    = 1
    )
ref[, .(id, value, computed = round(x = computed, digits = 5), agrees)]
```

```
                   id  value computed agrees
 1: cpi19_tornqvist_t2 1.4052  1.40516   TRUE
 2: cpi19_tornqvist_t3 1.3112  1.31120   TRUE
 3: cpi19_tornqvist_t4 1.2624  1.26244   TRUE
 4: cpi19_tornqvist_t5 1.2224  1.22241   TRUE
 5:    cpi19_fisher_t2 1.4011  1.40105   TRUE
 6:    cpi19_fisher_t3 1.3185  1.31851   TRUE
 7:    cpi19_fisher_t4 1.2689  1.26891   TRUE
 8:    cpi19_fisher_t5 1.2226  1.22261   TRUE
 9: cpi19_laspeyres_t2 1.4200  1.42000   TRUE
10: cpi19_laspeyres_t3 1.3646  1.36461   TRUE
11: cpi19_laspeyres_t4 1.3351  1.33514   TRUE
12: cpi19_laspeyres_t5 1.3306  1.33058   TRUE
13:   cpi19_paasche_t2 1.3823  1.38235   TRUE
14:   cpi19_paasche_t3 1.2740  1.27397   TRUE
15:   cpi19_paasche_t4 1.2060  1.20598   TRUE
16:   cpi19_paasche_t5 1.1234  1.12340   TRUE
```

4 Identities

The tests also check that

- the Divisia, Fisher, Hancock, simple-sum and currency equivalent indexes equal their formulas computed by hand on a small example;
- price times quantity is the expenditure (factor reversal), and the implicit price index of the Fisher quantity index is the Fisher price index of the user costs;
- the shares sum to one at every date, and do not change when the rates are given as decimals instead of percent;
- the Divisia mean of `dmai_moments()` is the Divisia growth, and the variance is never negative;
- the order of the rows and the names of the columns do not change the result.

For example, factor reversal:

```
f <-
  dmai(
      .data     = money_sim
    , method    = "Fisher"
    , logbase   = NULL
    , date      = "Date"
    , x         = NULL
    , r         = NULL
    , user_cost = NULL
    , benchmark = "rb"
    , percent   = TRUE
    , base      = 100
    , as_dt     = FALSE
    )
max(abs(f$price$price * f$index$value - f$price$expenditure))
```

```
[1] 4.547474e-13
```

5 Speed

Version 0.5.0 reshaped the data to a long table with tidyr and computed with grouped dplyr operations; version 1.0.0 works on the matrix of quantities. The timings below were taken by `tools/bench.R` on simulated data (median of 20 runs, milliseconds) and are stored with the package:

```
b <- data.table::fread(system.file("extdata", "bench.csv", package =
 ↪  "dmai"))
w <- data.table::dcast(data = b, formula = case ~ version, value.var =
 ↪  "median_ms")
v <- sort(unique(b$version))
```

```
w[, speedup := round(x = get(v[1]) / get(v[2]), digits = 1)]
w
```

```
             case    0.5.0    1.0.0 speedup
1:  barnett_240x8 30.08232 4.475517     6.7
2:  barnett_36x13 21.14572 3.490219     6.1
3: barnett_600x20 47.86928 9.989501     4.8
4:  hancock_36x13 19.90234 3.449297     5.8
```

The cases are 36 dates of 13 components (the example of version 0.5.0), 240 dates of 8 components and 600 dates of 20 components.

6 References

ILO, IMF, OECD, UNECE, Eurostat and The World Bank (2004). *Consumer Price Index Manual: Theory and Practice*. International Labour Office, Geneva.