Getting started with dmai

Muhammad Yaseen

2026-11-05

Table of contents

1 Abstract **1**

2 The data **1**

3 The Divisia aggregate **2**

4 Reading the result **4**

5 The other methods **5**

6 User costs and second moments **6**

7 Your own data **7**

8 References **7**

1 Abstract

This vignette shows the package at work: the data it expects, the Divisia monetary aggregate of Barnett (1980) with its user costs and weights, the other methods, and the plain-language reading of a result. The example data are **simulated**; the vignette "Theory" gives the formulas and "Validation" shows that they reproduce published values.

2 The data

`dmai()` takes a data frame with one row per date: a date column, the quantities of the components and their rates of return. By default the columns are named `Date`, `x1`, `x2`, … and `r1`, `r2`, …, as in version 0.5.0; `date`, `x` and `r` name other columns. `money_sim` holds simulated monthly data from 2005 to 2024: eight components (currency, demand deposits, savings deposits, four maturities

of time deposits and foreign currency deposits), their rates in percent and a benchmark rate `rb`, a long bond yield.

```
data(money_sim)
head(money_sim[, c("Date", "x1", "x2", "x3", "r1", "r2", "r3", "rb")])
```

```
        Date      x1     x2      x3 r1 r2    r3    rb
1 2005-01-01 1235.89 902.17 1528.41  0  0 3.500 9.500
2 2005-02-01 1227.77 929.38 1549.25  0  0 3.510 9.553
3 2005-03-01 1248.90 947.40 1585.89  0  0 3.165 9.028
4 2005-04-01 1243.45 947.19 1584.36  0  0 3.729 9.267
5 2005-05-01 1231.78 945.22 1603.69  0  0 3.070 9.127
6 2005-06-01 1221.64 963.80 1627.26  0  0 3.216 9.058
```

3 The Divisia aggregate

The services of money are priced by the user cost: the interest given up by holding an asset instead of a benchmark asset that yields no monetary services. Currency pays no interest, so its user cost is the whole benchmark rate; a time deposit that pays nearly the benchmark rate costs little to hold and is held mostly as a store of value.

```
fit <-
  dmai(
      .data     = money_sim
    , method    = "Barnett"
    , logbase   = NULL
    , date      = "Date"
    , x         = NULL
    , r         = NULL
    , user_cost = NULL
    , benchmark = "rb"
    , percent   = TRUE
    , base      = 100
    , as_dt     = FALSE
    )
fit
```

```
Divisia (Barnett) index of 8 components, 240 dates
==================================================
Dates: 2005-01-01 to 2024-12-01 (240); components: x1, x2, x3, x4, x5, x6, x7, x8
Benchmark rate: column rb (upper envelope with the component rates)
Growth (DMAI, log, percent): mean 0.8896, sd 0.5902, last 0.6266
Index: 100 at the first date, 838.3 at the last; simple sum: 1,106

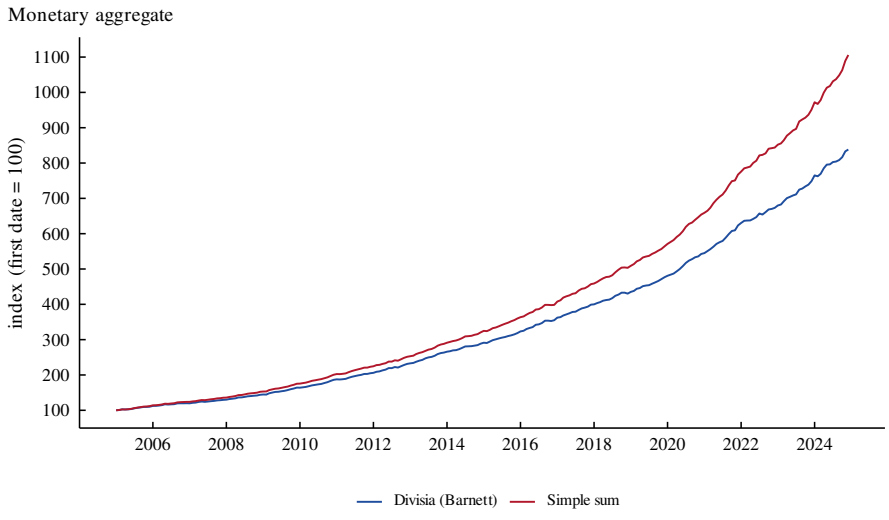
           Date    DMAI index value
  1: 2005-01-01      NA 100.0  5413
```

```
  2: 2005-02-01  0.8504 100.9  5459
  3: 2005-03-01  1.7329 102.6  5555
  4: 2005-04-01 -0.2494 102.4  5541
  5: 2005-05-01  0.3359 102.7  5560
 ---
236: 2024-08-01  0.1823 804.2 43533
237: 2024-09-01  0.5280 808.5 43764
238: 2024-10-01  0.9856 816.5 44197
239: 2024-11-01  2.0063 833.0 45093
240: 2024-12-01  0.6266 838.3 45376
```

```
Sources: Barnett (1978); Barnett (1980); Tornqvist (1936); Theil (1967). Full
references: dmai_cite(x).
```

`DMAI` is the growth from the previous date in percent (the log difference times 100), `index` the aggregate with 100 at the first date and `value` the aggregate in the units of the components, equal to the simple sum at the first date. With `as_dt = TRUE` (the default) `dmai()` returns only that table, as version 0.5.0 did.

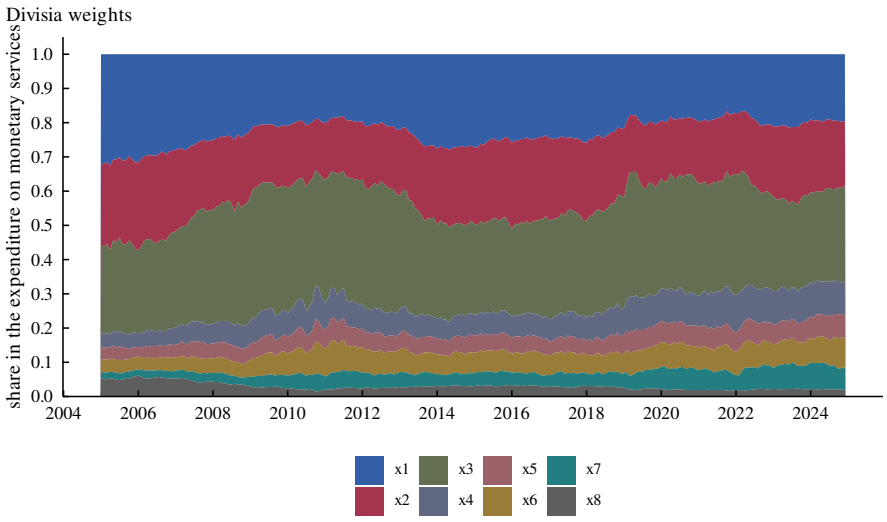
```
plot(
    x     = fit
  , which = "index"
  )
```



The Divisia aggregate grows more slowly than the simple sum here because the interest-bearing deposits grow fastest and they carry small weights. The weights over time:

```
plot(
```

```
    x     = fit
  , which = "shares"
  )
```



4 Reading the result

```
dmai_interpret(
    x   = fit
  , top = 3
  )
```

```
Divisia (Barnett) monetary aggregate

From 2005-01-01 to 2024-12-01 the Divisia (Barnett) aggregate grew 11.3% a
year, against 12.8% for the simple sum (1.6 points less).

Findings:
- Largest growth at 2009-03-01, 2021-12-01, 2024-01-01 (2.49, 2.24, 2.17);
  smallest at 2007-06-01, 2012-09-01, 2018-12-01 (-0.71, -0.67, -0.61).
- At the last date the largest weights are x3 (28.1%), x1 (19.5%), x2
  (18.8%).
- The weights moved most for x1 (32.3 to 19.5%), x4 (4.3 to 9.9%), x6 (3.7 to
  8.5%).
- The simple sum overstates money growth: it gives the same weight to
  interest-bearing deposits, which are held largely as savings, as to
  currency and demand deposits.
```

```
Recommendations:
- Compare the methods with dmai_compare(); check how the result depends on
  the benchmark rate.
- dmai_moments() shows when the components grow at very different rates.

Sources:
- Barnett, W. A. (1978). The user cost of money. Economics Letters, 1(2),
  145-149. doi:10.1016/0165-1765(78)90051-4
- Barnett, W. A. (1980). Economic monetary aggregates: an application of
  index number and aggregation theory. Journal of Econometrics, 14(1), 11-48.
  doi:10.1016/0304-4076(80)90070-6
- Tornqvist, L. (1936). The Bank of Finland's consumption price index. Bank
  of Finland Monthly Bulletin, 10, 1-8.
- Theil, H. (1967). Economics and Information Theory. North-Holland,
  Amsterdam.
```

5 The other methods

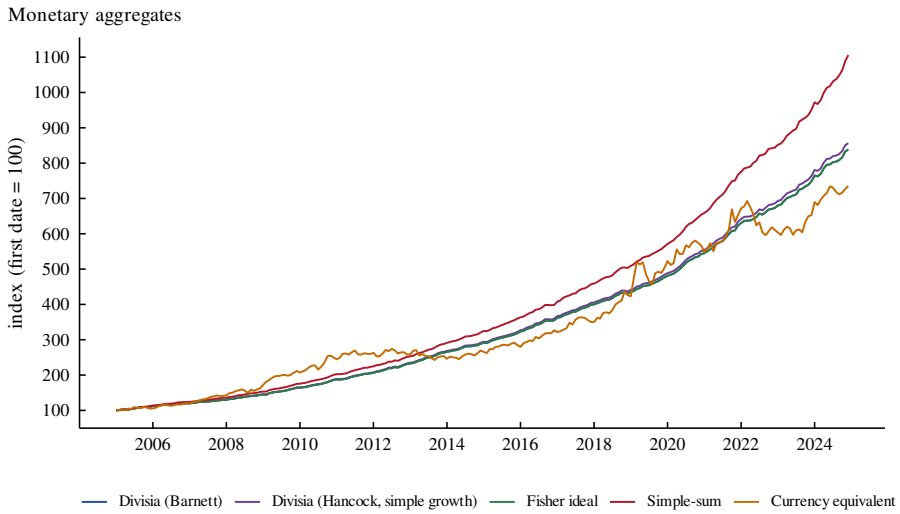
`dmai_compare()` computes every method on the same data:

```
cmp <-
  dmai_compare(
      .data     = money_sim
    , methods   = c("Barnett", "Hancock", "Fisher", "SimpleSum", "CE")
    , logbase   = NULL
    , date      = "Date"
    , x         = NULL
    , r         = NULL
    , user_cost = NULL
    , benchmark = "rb"
    , percent   = TRUE
    , base      = 100
    )
cmp
```

```
5 monetary aggregates of 8 components, 240 dates
================================================
      method mean_growth annual_growth sd_growth last_index
1:   Barnett      0.8896         11.27    0.5902      838.3
2:   Hancock      0.9044         11.39    0.5954      856.4
3:    Fisher      0.8896         11.27    0.5902      838.2
4: SimpleSum      1.0055         12.82    0.5726     1105.9
5:        CE      0.8347         10.54    2.6930      735.2

Sources: Barnett (1978); Barnett (1980); Hancock (1985); Fisher (1922); Diewert
(1976); Rotemberg et al. (1995). Full references: dmai_cite(x).
```

```
plot(
    x     = cmp
  , which = "index"
  )
```



The Divisia and Fisher indexes nearly coincide: both are superlative indexes (Diewert 1976). The Hancock variant uses simple instead of log growth rates and drifts above them; the currency equivalent index moves with the rates.

6 User costs and second moments

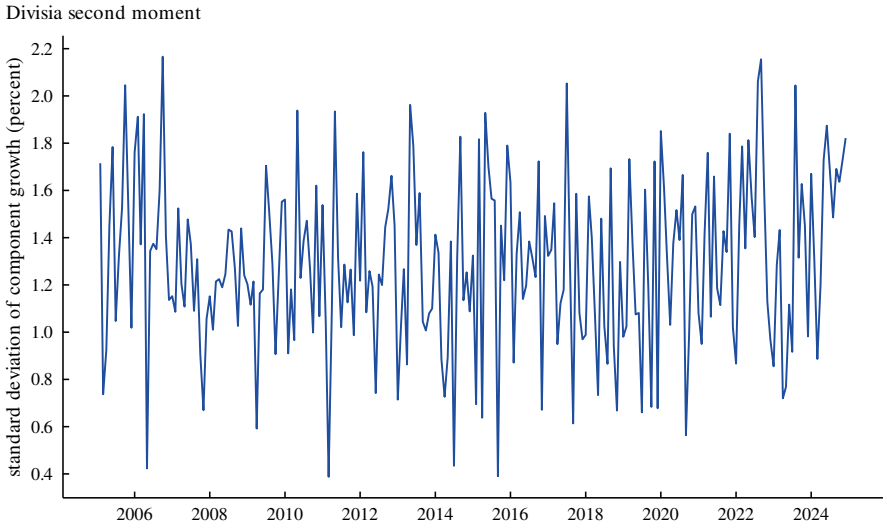
```
dmai_user_cost(
    .data     = money_sim
  , date      = "Date"
  , x         = NULL
  , r         = NULL
  , benchmark = "rb"
  , percent   = TRUE
  )
```

```
User costs of 8 components, 240 dates
=====================================
Averages over the dates:
   component mean_rate mean_user_cost mean_share last_share
1:        x1     0.000        0.08124    0.22914    0.19497
2:        x2     0.000        0.08124    0.20366    0.18827
3:        x3     3.050        0.05360    0.30779    0.28085
```

```
4:         x4     4.903         0.03662     0.07486     0.09851
5:         x5     5.414         0.03193     0.05076     0.06799
6:         x6     5.921         0.02727     0.06106     0.08527
7:         x7     6.647         0.02059     0.04325     0.06416
8:         x8     1.316         0.06916     0.02949     0.01998
```

```
Sources: Barnett (1978). Full references: dmai_cite(x).
```

```r
m <-
  dmai_moments(
      .data = fit
    )
plot(x = m)
```



7 Your own data

A CSV file with one row per date can be read with `data.table::fread()`; give the column names with `date`, `x` and `r`, and the benchmark with `benchmark` (a column, a number or one value per date). Without a benchmark the largest component rate is used at each date, as in version 0.5.0; that component then has a user cost of zero and no weight, so a benchmark above every component rate is the better choice. `dmai_app()` does all of this without code.

8 References

Barnett, W. A. (1978). The user cost of money. *Economics Letters*, 1(2), 145–149.

Barnett, W. A. (1980). Economic monetary aggregates: an application of index number and aggregation theory. *Journal of Econometrics*, 14(1), 11–48.

Diewert, W. E. (1976). Exact and superlative index numbers. *Journal of Econometrics*, 4(2), 115–145.