Package ‘FoReco’

April 15, 2026

Type Package

Title Forecast Reconciliation

Version 1.2.1

Description Classical (bottom-up and top-down), optimal combination and heuristic point (Di Fonzo and Girolimetto, 2023 <doi:10.1016/j.ijforecast.2021.08.004>) and probabilistic (Girolimetto et al. 2024 <doi:10.1016/j.ijforecast.2023.10.003>) forecast reconciliation procedures for linearly constrained time series (e.g., hierarchical or grouped time series) in cross-sectional, temporal, or cross-temporal frameworks.

License GPL-3

URL https://github.com/danigiro/FoReco, https://danigiro.github.io/FoReco/

BugReports https://github.com/danigiro/FoReco/issues

Depends R (>= 3.4), Matrix

Imports methods, osqp, stats, cli, distributional, lifecycle

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

NeedsCompilation no

RoxygenNote 7.3.3

Author Daniele Girolimetto [aut, cre] (ORCID: <https://orcid.org/0000-0001-9387-1232>), Tommaso Di Fonzo [aut] (ORCID: <https://orcid.org/0000-0003-3388-7827>), Yangzhuoran Fin Yang [ctb] (ORCID: <https://orcid.org/0000-0002-1232-8017>)

Maintainer Daniele Girolimetto <daniele.girolimetto@unipd.it>

Repository CRAN

Date/Publication 2026-04-15 06:31:00 UTC

Contents

`FoReco-package` *FoReco: Forecast Reconciliation*

Description

Classical (bottom-up and top-down), optimal combination and heuristic point (Di Fonzo and Girolimetto, 2023 doi:10.1016/j.ijforecast.2021.08.004) and probabilistic (Girolimetto et al. 2024 doi:10.1016/j.ijforecast.2023.10.003) forecast reconciliation procedures for linearly constrained time series (e.g., hierarchical or grouped time series) in cross-sectional, temporal, or cross-temporal frameworks.

Author(s)

Maintainer: Daniele Girolimetto `<daniele.girolimetto@unipd.it>` (ORCID)

Authors:

- Tommaso Di Fonzo (ORCID)

Other contributors:

- Yangzhuoran Fin Yang (ORCID) [contributor]

See Also

Useful links:

- `https://github.com/danigiro/FoReco`
- `https://danigiro.github.io/FoReco/`
- Report bugs at `https://github.com/danigiro/FoReco/issues`

aggts *Non-overlapping temporal aggregation of a time series*

Description

Non-overlapping temporal aggregation of a time series according to a specific aggregation order.

Usage

```
aggts(y, agg_order, tew = "sum", align = "end", rm_na = FALSE)
```

Arguments

y	Univariate or multivariate time series: a vector/matrix or a ts object.
agg_order	A numeric vector with the aggregation orders to consider.
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
align	A string or a vector specifying the alignment of y. Options include: "end" (end of the series, *default*), "start" (start of the series), an integer (or a vector of integers) indicating the starting period of the temporally aggregated series.
rm_na	If TRUE the missing values are removed.

Value

A list of vectors or ts objects.

See Also

Utilities: FoReco2matrix(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
# Monthly time series (input vector)
y <- ts(rnorm(24), start = 2020, frequency = 12)
# Quarterly time series
x1 <- aggts(y, 3)
# Monthly, quarterly and annual time series
x2 <- aggts(y, c(1, 3, 12))
# All temporally aggregated time series
x3 <- aggts(y)

# Ragged data
y2 <- ts(rnorm(11), start = c(2020, 3), frequency = 4)
# Annual time series: start in 2021
```

```
x4 <- aggts(y2, 4, align = 3)
# Semi-annual (start in 2nd semester of 2020) and annual (start in 2021)
# time series
x5 <- aggts(y2, c(2, 4), align = c(1, 3))
```

---

as_ctmatrix *Convert between horizon-stacked and cross-temporal layouts*

Description

These functions convert matrix between the two canonical layouts used in cross-temporal reconciliation. Let m be the maximum temporal aggregation order and k^* the sum of a subset of the $(p-1)$ proper factors of m (excluding m); let h be the forecast horizon for the lowest frequency series (e.g., most aggregated temporal forecast horizon) and n the number of variables:

- *Horizon-stacked layout (cross-temporal version)*: a $h \times n(k^* + m)$ matrix where rows are the most aggregated temporal forecast horizons, and the values in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency grouped by variable.
- *Cross-temporal layout*: a $n \times h(k^* + m)$ matrix where rows are variables, and horizons for each temporal block appear consecutively. rows are variables, and the values in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.

Then, as_ctmatrix converts a $(h \times n(k^* + m))$ horizon-stacked to a $(n \times h(k^* + m))$ cross-temporal matrix; as_hstack_ctlayout performs the inverse transform.

Usage

```
as_ctmatrix(hmat, agg_order, n, row_names = NULL)

as_hstack_ctlayout(ctmat, agg_order)
```

Arguments

`hmat`	A $h \times n(k^* + m)$ numeric matrix in *horizon-stacked* layout (cross-temporal version).
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`n`	Cross-sectional number of variables.
`row_names`	Optional character vector of length `n` with row names for the *cross-temporal* output of `as_ctmatrix()`. If `NULL` (*default*) no custom names are assigned.
`ctmat`	A $n \times h(k^* + m)$ numeric matrix in *cross-temporal* layout.

Value

as_ctmatrix returns a $n \times h(k^* + m)$ numeric matrix in *cross-temporal* layout.

as_hstack_ctlayout returns a $h \times n(k^* + m)$ numeric matrix in *horizon-stacked* layout (cross-temporal version).

See Also

Utilities: FoReco2matrix(), aggts(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
h <- 2   # horizons
n <- 3   # variables
m <- 4   # temporal aggregation order
kt <- tetools(m)$dim["kt"]

# Build a horizon-stacked matrix: h rows, n * k_t columns
input_ct <- matrix(seq_len(h * n * kt), nrow = n, byrow = TRUE)

hmat <- as_hstack_ctlayout(input_ct, agg_order = m)
ctmat <- as_ctmatrix(hmat, agg_order = m, n = n)
# all.equal(ctmat, input_ct, check.attributes = FALSE)
```

---

as_tevector — *Convert between horizon-stacked and temporal layouts*

Description

These functions convert matrix between the two canonical layouts used in temporal reconciliation. Let m be the maximum temporal aggregation order and k^* the sum of a subset of the $(p-1)$ proper factors of m (excluding m); let h be the forecast horizon for the lowest frequency series (e.g., most aggregated temporal forecast horizon):

- *Horizon-stacked layout (temporal version)*: a $h \times (k^* + m)$ matrix where rows are the most aggregated temporal forecast horizons, and the values in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.
- *Temporal layout*: a $(h(k^* + m) \times 1)$ numeric vector where values are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.

Then, as_tevector converts a $(h \times (k^* + m))$ horizon-stacked matrix to a $(h(k^* + m) \times 1)$ temporal vector; as_hstack_telayout performs the inverse transform.

Usage

```
as_tevector(hmat, agg_order)

as_hstack_telayout(tevec, agg_order)
```

Arguments

hmat	A $h \times (k^* + m)$ numeric matrix in *horizon-stacked* layout (temporal version).
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
tevec	A $(h(k^* + m) \times 1)$ numeric vector in *temporal* layout.

Value

as_tevector returns a $(h(k^* + m) \times 1)$ numeric vector in *temporal* layout.

as_hstack_telayout returns a $h \times (k^* + m)$ numeric matrix in *horizon-stacked* layout (temporal version).

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
h <- 2   # horizons
m <- 4   # temporal aggregation order
kt <- tetools(m)$dim["kt"]

# Build a horizon-stacked matrix: h rows, n * k_t columns
input_te <- seq_len(h * kt)

hmat <- as_hstack_telayout(input_te, agg_order = m)
tevec <- as_tevector(hmat, agg_order = m)
# all.equal(tevec, input_te, check.attributes = FALSE)
```

---

balance_hierarchy *Aggregation matrix of a (possibly) unbalanced hierarchy in balanced form*

Description

A hierarchy with L upper levels is said to be balanced if each variable at level l has at least one child at level $l+1$. When this doesn't hold, the hierarchy is unbalanced. This function transforms an aggregation matrix of an unbalanced hierarchy into an aggregation matrix of a balanced one. This function is used to reconcile forecasts with cslcc, which operates exclusively with balanced hierarchies.

Usage

```
balance_hierarchy(agg_mat, nodes = "auto", sparse = TRUE)
```

Arguments

agg_mat
: A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

nodes
: A $(L \times 1)$ numeric vector indicating the number of variables in each of the upper L levels of the hierarchy. The *default* value is the string "auto" which calculates the number of variables in each level.

sparse
: Option to return sparse matrices (*default* is TRUE).

Value

A list containing four elements:

bam
: The balanced aggregation matrix.

agg_mat
: The input matrix.

nodes
: A $(L \times 1)$ numeric vector indicating the number of variables in each of the L upper levels of the balanced hierarchy.

id
: The identification number of each variable in the balanced hierarchy. It may contains duplicated values.

See Also

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
#    Unbalanced     ->      Balanced
#        T                     T
#    |-------|             |-------|
#    A       |             A       B
#  |---|     |           |---|     |
# AA   AB    B          AA   AB    BA
A <- matrix(c(1, 1, 1,
              1, 1, 0), 2, byrow = TRUE)
obj <- balance_hierarchy(agg_mat = A, nodes = c(1, 1))
obj$bam
```

---

commat *Commutation matrix*

Description

This function returns the $(rc \times rc)$ commutation matrix $\mathbf{P}$ such that $\mathbf{P}\text{vec}(\mathbf{Y}) = \text{vec}(\mathbf{Y}')$, where $\mathbf{Y}$ is a $(r \times c)$ matrix (Magnus and Neudecker, 2019).

Usage

```
# Commutation matrix
commat(r, c)

# Vector of indexes for the rows of commutation matrix
commat_index(r, c)
```

Arguments

`r`	Number of rows of $\mathbf{Y}$.
`c`	Number of columns of $\mathbf{Y}$.

Value

A sparse $(rc \times rc)$ matrix $\mathbf{P}$ (commat), or the vector of indexes for the rows of commutation matrix $\mathbf{P}$ (commat_index)

References

Magnus, J.R. and Neudecker, H. (2019), Matrix Differential Calculus with Applications in Statistics and Econometrics, third edition, New York, Wiley, pp. 54-55.

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
Y <- matrix(rnorm(30), 5, 6)
P <- commat(5, 6)
P %*% as.vector(Y) == as.vector(t(Y)) # check
```

`csboot` *Cross-sectional joint block bootstrap*

Description

Joint block bootstrap for generating probabilistic base forecasts that take into account the correlation between different time series (Panagiotelis et al. 2023).

Usage

```
csboot(model_list, boot_size, block_size, seed = NULL, xreg = NULL, ...)
```

Arguments

`model_list`	A list of all the n base forecasts models. A `simulate()` function for each model has to be available and implemented according to the package **forecast**, with the following mandatory parameters: *object*, *innov*, *future*, and *nsim*.
`boot_size`	The number of bootstrap replicates.
`block_size`	Block size of the bootstrap, which is typically equivalent to the forecast horizon.
`seed`	An integer seed.
`xreg`	An optional 3-d numeric array of dimensions (block_size $\times n \times N_{xreg}$) containing the new values of `xreg` to be used for forecasting. It can contain `NA`s.
`...`	Additional arguments for the `simulate()` function.

Value

A list with two elements: the seed used to sample the errors and a 3-d array (block_size $\times$ n $\times$ boot_size).

References

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G. and Hyndman, R.J. (2023), Probabilistic forecast reconciliation: Properties, evaluation and score optimisation, *European Journal of Operational Research*, 306(2), 693–706. doi:10.1016/j.ejor.2022.07.040

See Also

Bootstrap samples: `ctboot()`, `teboot()`

Cross-sectional framework: `csbu()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstd()`, `cstools()`

---

csbu — *Cross-sectional bottom-up reconciliation*

Description

This function computes the cross-sectional bottom-up reconciled forecasts (Dunn et al., 1976) for all series by appropriate summation of the bottom base forecasts $\widehat{\mathbf{b}}$:

$$\widetilde{\mathbf{y}} = \mathbf{S}_{cs}\widehat{\mathbf{b}},$$

where $\mathbf{S}_{cs}$ is the cross-sectional structural matrix.

Usage

```
csbu(base, agg_mat, sntz = FALSE, round = FALSE)
```

Arguments

- `base` — A $(h \times n_b)$ numeric matrix or multivariate time series (`mts` class) containing bottom base forecasts; h is the forecast horizon, and n_b is the total number of bottom variables.
- `agg_mat` — A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
- `sntz` — Logical. If `TRUE`, the negative base forecasts are set to zero (Di Fonzo and Girolimetto, 2023) before applying bottom-up. *Default* is `FALSE`.
- `round` — Logical. If `TRUE`, base forecasts are rounded before applying the bottom-up reconciliation. *Default* is `FALSE`.

Value

A $(h \times n)$ numeric matrix of cross-sectional reconciled forecasts.

References

Dunn, D. M., Williams, W. H. and Dechaine, T. L. (1976), Aggregate versus subaggregate models in local area forecasting, *Journal of the American Statistical Association* 71(353), 68–71. doi:10.1080/01621459.1976.10481478

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

See Also

Bottom-up reconciliation: `ctbu()`, `tebu()`

Cross-sectional framework: `csboot()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstd()`, `cstools()`

Examples

```
set.seed(123)
# (3 x 2) bottom base forecasts matrix (simulated), Z = X + Y
bts <- matrix(rnorm(6, mean = c(10, 10)), 3, byrow = TRUE)

# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
reco <- csbu(base = bts, agg_mat = A)

# Non negative reconciliation
bts[2,2] <- -bts[2,2] # Making negative one of the base forecasts for Y
nnreco <- csbu(base = bts, agg_mat = A, sntz = TRUE)
```

cscov *Cross-sectional covariance matrix approximation*

Description

This function provides an approximation of the cross-sectional base forecasts errors covariance matrix using different reconciliation methods (see Wickramasuriya et al., 2019 and Di Fonzo and Girolimetto, 2023).

Usage

```
cscov(comb = "ols", agg_mat = NULL, res = NULL, n = NULL, mse = TRUE,
      shrink_fun = shrink_estim, ...)
```

Arguments

`comb` A string specifying the covariance approximation method.

- For ordinary least squares reconciliation:
  - `"ols"` (*default*) - identity error covariance matrix.
- For weighted least squares reconciliation:
  - `"str"` - structural variances.
  - `"wls"` - series variances (uses `res`).
- For generalized least squares (uses `res`) reconciliation:
  - `"shr"` - shrunk covariance (Wickramasuriya et al., 2019).
  - `"oasd"` - oracle shrunk covariance (Ando and Xiao, 2023).
  - `"sam"` - sample covariance.
- Others (no for reconciliation):
  - `"bu"` - bottom-up covariance.

`agg_mat` A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

`res` An $(N \times n)$ optional numeric matrix containing the in-sample residuals or validation errors. This matrix is used to compute some covariance matrices.

`n` Number of variables $(n = n_a + n_b)$.

`mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.

`shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

`...` Not used.

Value

A $(n \times n)$ symmetric positive (semi-)definite matrix.

References

Ando, S., and Xiao, M. (2023), High-dimensional covariance matrix estimation: shrinkage toward a diagonal target. *IMF Working Papers*, 2023(257), A001.

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Wickramasuriya, S.L., Athanasopoulos, G. and Hyndman, R.J. (2019), Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, *Journal of the American Statistical Association*, 114, 526, 804-819. doi:10.1080/01621459.2018.1448825

See Also

Cross-sectional framework: `csboot()`, `csbu()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstd()`, `cstools()`

Examples

```
# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
# (10 x 3) in-sample residuals matrix (simulated)
res <- t(matrix(rnorm(n = 30), nrow = 3))

cov1 <- cscov("ols", n = 3)          # OLS
cov2 <- cscov("str", agg_mat = A)    # STR
cov3 <- cscov("wls", res = res)      # WLS
cov4 <- cscov("shr", res = res)      # SHR
cov5 <- cscov("sam", res = res)      # SAM

# Custom covariance matrix
cscov.ols2 <- function(comb, x) diag(x)
cscov(comb = "ols2", x = 3) # == cscov("ols", n = 3)
```

---

cslcc — *Level conditional coherent reconciliation for genuine hierarchical/grouped time series*

Description

This function implements the cross-sectional forecast reconciliation procedure that extends the original proposal by Hollyman et al. (2021). Level conditional coherent reconciled forecasts are conditional on (i.e., constrained by) the base forecasts of a specific upper level in the hierarchy (exogenous constraints). It also allows handling the linear constraints linking the variables endogenously (Di Fonzo and Girolimetto, 2022). The function can calculate Combined Conditional Coherent (CCC) forecasts as simple averages of Level-Conditional Coherent (LCC) and bottom-up reconciled forecasts, with either endogenous or exogenous constraints.

Usage

```
cslcc(base, agg_mat, comb = "ols", res = NULL, approach = "proj", nn = NULL,
      settings = NULL, CCC = TRUE, const = "exogenous", bts = NULL, ...)
```

Arguments

base A $(h \times n)$ numeric matrix or multivariate time series (mts class) containing the base forecasts to be reconciled; h is the forecast horizon, and n is the total number of time series ($n = n_a + n_b$).

agg_mat A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

comb A string specifying the reconciliation method. For a complete list, see cscov.

res An $(N \times n)$ optional numeric matrix containing the in-sample residuals or validation errors. This matrix is used to compute some covariance matrices.

approach A string specifying the approach used to compute the reconciled forecasts. Options include:

- "proj" (*default*): Projection approach according to Byron (1978, 1979).
- "strc": Structural approach as proposed by Hyndman et al. (2011).
- "proj_osqp": Numerical solution using **osqp** for projection approach.
- "strc_osqp": Numerical solution using **osqp** for structural approach.

nn A string specifying the algorithm to compute non-negative forecasts:

- "osqp": quadratic programming optimization (**osqp** solver, Girolimetto 2025).
- "bpv": block principal pivoting algorithm (Wickramasuriya et al., 2020).
- "nfca": negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
- "nnic": iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
- "sntz": heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).

settings A list of control parameters.

- nn = "osqp" An object of class osqpSettings specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
- nn = "bpv"
  - ptype = "fixed": permutation method: "random" or "fixed"
  - par = 10: the number of full exchange rules that may be attempted
  - tol = sqrt(.Machine$double.eps): the tolerance criteria
  - gtol = sqrt(.Machine$double.eps): the gradient tolerance criteria
  - itmax = 100: the maximum number of algorithm iterations
- nn = "nfca" and nn = "nnic"
  - tol = sqrt(.Machine$double.eps): the tolerance criteria
  - itmax = 100: the maximum number of algorithm iterations

- nn = "sntz"
  - type = "bu": the type of set-negative-to-zero heuristic: "bu" for bottom-up, "tdp" for top-down proportional, "tdsp" for top-down square proportional, "tdvw" for top-down variance weighted (the res param is used). See Girolimetto (2025) for details.
  - tol = sqrt(.Machine$double.eps): the tolerance identification of negative values

CCC
: A logical value indicating whether the Combined Conditional Coherent reconciled forecasts reconciliation should include bottom-up forecasts (TRUE, *default*), or not.

const
: A string specifying the reconciliation constraints:
  - "exogenous" (*default*): Fixes the top level of each sub-hierarchy.
  - "endogenous": Coherently revises both the top and bottom levels.

bts
: A $(h \times n_b)$ numeric matrix or multivariate time series (mts class) containing bottom base forecasts defined by the user (e.g., seasonal averages, as in Hollyman et al., 2021). This parameter can be omitted if only base forecasts are used (see Di Fonzo and Girolimetto, 2024).

...
: Arguments passed on to cscov

  mse If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected.

  shrink_fun Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A $(h \times n)$ numeric matrix of cross-sectional reconciled forecasts.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2024), Forecast combination-based forecast reconciliation: Insights and extensions, *International Journal of Forecasting*, 40(2), 490–514. doi:10.1016/j.ijforecast.2022.07.001

Di Fonzo, T. and Girolimetto, D. (2023b) Spatio-temporal reconciliation of solar forecasts. *Solar Energy* 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Hollyman, R., Petropoulos, F. and Tipping, M.E. (2021), Understanding forecast reconciliation. *European Journal of Operational Research*, 294, 149–160. doi:10.1016/j.ejor.2021.01.017

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

See Also

Level conditional coherent reconciliation: `ctlcc()`, `telcc()`

Cross-sectional framework: `csboot()`, `csbu()`, `cscov()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstd()`, `cstools()`

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y, X = XX + XY and Y = YX + YY
A <- matrix(c(1,1,1,1,1,1,0,0,0,0,1,1), 3, byrow = TRUE)
# (2 x 7) base forecasts matrix (simulated)
base <- matrix(rnorm(7*2, mean = c(40, 20, 20, 10, 10, 10, 10)), 2, byrow = TRUE)
# (10 x 7) in-sample residuals matrix (simulated)
res <- matrix(rnorm(n = 7*10), ncol = 7)
# (2 x 7) Naive bottom base forecasts matrix: all forecasts are set equal to 10
naive <- matrix(10, 2, 4)

## EXOGENOUS CONSTRAINTS (Hollyman et al., 2021)
# Level Conditional Coherent (LCC) reconciled forecasts
exo_LC <- cslcc(base = base, agg_mat = A, comb = "wls", bts = naive,
                res = res, CCC = FALSE)

# Combined Conditional Coherent (CCC) reconciled forecasts
exo_CCC <- cslcc(base = base, agg_mat = A, comb = "wls", bts = naive,
                 res = res, CCC = TRUE)

# Results detailed by level:
# L-1: Level 1 immutable reconciled forecasts for the whole hierarchy
# L-2: Middle-Out reconciled forecasts
# L-3: Bottom-Up reconciled forecasts
info_exo <- recoinfo(exo_CCC, verbose = FALSE)
info_exo$lcc

## ENDOGENOUS CONSTRAINTS (Di Fonzo and Girolimetto, 2024)
# Level Conditional Coherent (LCC) reconciled forecasts
endo_LC <- cslcc(base = base, agg_mat = A, comb = "wls",
                 res = res, CCC = FALSE,
                 const = "endogenous")

# Combined Conditional Coherent (CCC) reconciled forecasts
endo_CCC <- cslcc(base = base, agg_mat = A, comb = "wls",
                  res = res, CCC = TRUE,
                  const = "endogenous")
```

```
# Results detailed by level:
# L-1: Level 1 reconciled forecasts for L1 + L3 (bottom level)
# L-2: Level 2 reconciled forecasts for L2 + L3 (bottom level)
# L-3: Bottom-Up reconciled forecasts
info_endo <- recoinfo(endo_CCC, verbose = FALSE)
info_endo$lcc
```

---

csmo *Cross-sectional middle-out reconciliation*

Description

The middle-out forecast reconciliation (Athanasopoulos et al., 2009) combines top-down (cstd) and bottom-up (csbu) for genuine hierarchical/grouped time series. Given the base forecasts of variables at an intermediate level l, it performs

- a top-down approach for the levels $< l$;
- a bottom-up approach for the levels $> l$.

Usage

```
csmo(base, agg_mat, weights, id_rows = 1, normalize = TRUE)
```

Arguments

`base`	A $(h \times n_l)$ numeric matrix containing the l-level base forecast; n_l is the number of variables at level l, and h is the forecast horizon.
`agg_mat`	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
`weights`	A $(h \times n_b)$ numeric matrix containing the proportions for the bottom time series; h is the forecast horizon, and n_b is the total number of bottom variables.
`id_rows`	A numeric vector indicating the l-level rows of `agg_mat`.
`normalize`	If `TRUE` (*default*), the `weights` will sum to 1.

Value

A $(h \times n)$ numeric matrix of cross-sectional reconciled forecasts.

References

Athanasopoulos, G., Ahmed, R. A. and Hyndman, R.J. (2009) Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting* 25(1), 146–166. doi:10.1016/j.ijforecast.2008.07.004

See Also

Middle-out reconciliation: ctmo(), temo()

Cross-sectional framework: csboot(), csbu(), cscov(), cslcc(), csmvn(), csrec(), cssmp(), cstd(), cstools()

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y, X = XX + XY and Y = YX + YY
A <- matrix(c(1,1,1,1,1,1,0,0,0,0,1,1), 3, byrow = TRUE)
# (3 x 2) top base forecasts vector (simulated), forecast horizon = 3
baseL2 <- matrix(rnorm(2*3, 5), 3, 2)
# Same weights for different forecast horizons
fix_weights <- runif(4)
reco <- csmo(base = baseL2, agg_mat = A, id_rows = 2:3, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- matrix(runif(4*3), 3, 4)
recoh <- csmo(base = baseL2, agg_mat = A, id_rows = 2:3, weights = h_weights)
```

---

csmvn — *Cross-sectional Gaussian probabilistic reconciliation*

Description

This function performs cross-sectional probabilistic forecast reconciliation assuming a multivariate normal base forecast distribution (Panagiotelis et al., 2023; Girolimetto et al., 2024; Wickramasuriya, 2024) for linearly constrained (e.g. hierarchical or grouped) multiple time series.

Usage

```
csmvn(base, agg_mat, cons_mat, comb = "ols", res = NULL,
      approach = "proj", comb_base = comb, reduce_form = FALSE, ...)
```

Arguments

base	A $(h \times n)$ numeric matrix or multivariate time series (mts class) containing the base forecasts to be reconciled; h is the forecast horizon, and n is the total number of time series ($n = n_a + n_b$).
agg_mat	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
cons_mat	A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.

comb	A string specifying the reconciliation method. For a complete list, see cscov.
res	An $(N \times n)$ optional numeric matrix containing the in-sample residuals or validation errors. This matrix is used to compute some covariance matrices.
approach	A string specifying the approach used to compute the reconciled mean and covariance matrix. Options include: • "proj" (*default*): Projection approach according to Byron (1978, 1979). • "strc": Structural approach as proposed by Hyndman et al. (2011).
comb_base	A string specifying the base covariance matrix approach. For a complete list, see cscov. Default is the equal to comb.
reduce_form	A logical parameter indicating whether the function should return the full distribution (FALSE, *default*) or only the distribution corresponding to the bottom-level time series (TRUE).
...	Arguments passed on to cscov mse If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected. shrink_fun Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A distributional::dist_multivariate_normal object.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G. and Hyndman, R.J. (2023), Probabilistic forecast reconciliation: Properties, evaluation and score optimisation, *European Journal of Operational Research* 306(2), 693–706. doi:10.1016/j.ejor.2022.07.040

Wickramasuriya, S. L. (2024). Probabilistic Forecast Reconciliation under the Gaussian Framework. *Journal of Business & Economic Statistics*, 42(1), 272–285. doi:10.1080/07350015.2023.2181176

See Also

Probabilistic reconciliation: cssmp(), ctmvn(), ctsmp(), temvn(), tesmp()

Cross-sectional framework: csboot(), csbu(), cscov(), cslcc(), csmo(), csrec(), cssmp(), cstd(), cstools()

Examples

```
set.seed(123)
# (2 x 3) base forecasts matrix (simulated), Z = X + Y
base <- matrix(rnorm(6, mean = c(20, 10, 10)), 2, byrow = TRUE)
# (10 x 3) in-sample residuals matrix (simulated)
res <- t(matrix(rnorm(n = 30), nrow = 3))

# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
reco_dist <- csmvn(base = base, agg_mat = A, comb = "shr", res = res)
```

---

csprojmat — *Projection matrix for optimal combination cross-sectional reconciliation*

Description

This function computes the projection or the mapping matrix $\mathbf{M}$ and $\mathbf{G}$, respectively, such that $\widetilde{\mathbf{y}} = \mathbf{M}\widehat{\mathbf{y}} = \mathbf{S}_{cs}\mathbf{G}\widehat{\mathbf{y}}$, where $\widetilde{\mathbf{y}}$ is the vector of the reconciled forecasts, $\widehat{\mathbf{y}}$ is the vector of the base forecasts, $\mathbf{S}_{cs}$ is the cross-sectional structural matrix, and $\mathbf{M} = \mathbf{S}_{cs}\mathbf{G}$. For further information regarding on the structure of these matrices, refer to Girolimetto et al. (2023).

Usage

```
csprojmat(agg_mat, cons_mat, comb = "ols", res = NULL, mat = "M", ...)
```

Arguments

- `agg_mat` — A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
- `cons_mat` — A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.
- `comb` — A string specifying the reconciliation method. For a complete list, see cscov.
- `res` — An $(N \times n)$ optional numeric matrix containing the in-sample residuals or validation errors. This matrix is used to compute some covariance matrices.
- `mat` — A string specifying which matrix to return: "M" (*default*) for $\mathbf{M}$ and "G" for $\mathbf{G}$.
- `...` — Arguments passed on to `cscov`
  - `mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.
  - `shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

The projection matrix $\mathbf{M}$ (mat = "M") or the mapping matrix $\mathbf{G}$ (mat = "G").

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
# Cross-sectional framework
A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
Mcs <- csprojmat(agg_mat = A, comb = "ols")
Gcs <- csprojmat(agg_mat = A, comb = "ols", mat = "G")
```

---

csrec *Optimal combination cross-sectional reconciliation*

Description

This function performs optimal (in least squares sense) combination cross-sectional forecast reconciliation for a linearly constrained (e.g., hierarchical/grouped) multiple time series (Wickramasuriya et al., 2019, Panagiotelis et al., 2022, Girolimetto and Di Fonzo, 2023). The reconciled forecasts are calculated using either a projection approach (Byron, 1978, 1979) or the equivalent structural approach by Hyndman et al. (2011). Non-negative (Di Fonzo and Girolimetto, 2023) and immutable including Zhang et al., 2023) reconciled forecasts can be considered.

Usage

```
csrec(base, agg_mat, cons_mat, comb = "ols", res = NULL, approach = "proj",
      nn = NULL, settings = NULL, bounds = NULL, immutable = NULL, ...)
```

Arguments

base	A $(h \times n)$ numeric matrix or multivariate time series (mts class) containing the base forecasts to be reconciled; h is the forecast horizon, and n is the total number of time series ($n = n_a + n_b$).
agg_mat	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

`cons_mat` A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.

`comb` A string specifying the reconciliation method. For a complete list, see cscov.

`res` An $(N \times n)$ optional numeric matrix containing the in-sample residuals or validation errors. This matrix is used to compute some covariance matrices.

`approach` A string specifying the approach used to compute the reconciled forecasts. Options include:

- "`proj`" (*default*): Projection approach according to Byron (1978, 1979).
- "`strc`": Structural approach as proposed by Hyndman et al. (2011).
- "`proj_osqp`": Numerical solution using **osqp** for projection approach.
- "`strc_osqp`": Numerical solution using **osqp** for structural approach.

`nn` A string specifying the algorithm to compute non-negative forecasts:

- "`osqp`": quadratic programming optimization (**osqp** solver, Girolimetto 2025).
- "`bpv`": block principal pivoting algorithm (Wickramasuriya et al., 2020).
- "`nfca`": negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
- "`nnic`": iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
- "`sntz`": heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).

`settings` A list of control parameters.

- `nn = "osqp"` An object of class `osqpSettings` specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
- `nn = "bpv"`
  - `ptype = "fixed"`: permutation method: "`random`" or "`fixed`"
  - `par = 10`: the number of full exchange rules that may be attempted
  - `tol = sqrt(.Machine$double.eps)`: the tolerance criteria
  - `gtol = sqrt(.Machine$double.eps)`: the gradient tolerance criteria
  - `itmax = 100`: the maximum number of algorithm iterations
- `nn = "nfca"` and `nn = "nnic"`
  - `tol = sqrt(.Machine$double.eps)`: the tolerance criteria
  - `itmax = 100`: the maximum number of algorithm iterations
- `nn = "sntz"`
  - `type = "bu"`: the type of set-negative-to-zero heuristic: "`bu`" for bottom-up, "`tdp`" for top-down proportional, "`tdsp`" for top-down square proportional, "`tdvw`" for top-down variance weighted (the `res` param is used). See Girolimetto (2025) for details.
  - `tol = sqrt(.Machine$double.eps)`: the tolerance identification of negative values

`bounds` A matrix (see set_bounds) with 3 columns ($i, lower, upper$), such that

- Column 1 represents the cross-sectional series ($i = 1, \dots, n$).
- Columns 2 and 3 indicates the *lower* and *upper* bounds, respectively.

`immutable` A numeric vector containing the column indices of the base forecasts (`base` parameter) that should be fixed.

`...` Arguments passed on to `cscov`

`mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.

`shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A $(h \times n)$ numeric matrix of cross-sectional reconciled forecasts.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Girolimetto, D. and Di Fonzo, T. (2023), Point and probabilistic forecast reconciliation for general linearly constrained multiple time series, *Statistical Methods & Applications*, 33, 581-607. doi:10.1007/s10260023007386.

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Panagiotelis, A., Athanasopoulos, G., Gamakumara, P. and Hyndman, R.J. (2021), Forecast reconciliation: A geometric view with new insights on bias correction, *International Journal of Forecasting*, 37, 1, 343–359. doi:10.1016/j.ijforecast.2020.06.004

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

Wickramasuriya, S.L., Athanasopoulos, G. and Hyndman, R.J. (2019), Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, *Journal of the American Statistical Association*, 114, 526, 804-819. doi:10.1080/01621459.2018.1448825

Wickramasuriya, S. L., Turlach, B. A., and Hyndman, R. J. (2020). Optimal non-negative forecast reconciliation. *Statistics and Computing*, 30(5), 1167–1182. doi:10.1007/s11222020099300

Zhang, B., Kang, Y., Panagiotelis, A. and Li, F. (2023), Optimal reconciliation with immutable forecasts, *European Journal of Operational Research*, 308(2), 650–660. doi:10.1016/j.ejor.2022.11.035

See Also

Regression-based reconciliation: `ctrec()`, `terec()`

Cross-sectional framework: `csboot()`, `csbu()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `cssmp()`, `cstd()`, `cstools()`

Examples

```
set.seed(123)
# (2 x 3) base forecasts matrix (simulated), Z = X + Y
base <- matrix(rnorm(6, mean = c(20, 10, 10)), 2, byrow = TRUE)
# (10 x 3) in-sample residuals matrix (simulated)
res <- t(matrix(rnorm(n = 30), nrow = 3))

# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
reco <- csrec(base = base, agg_mat = A, comb = "wls", res = res)

# Zero constraints matrix for Z - X - Y = 0
C <- t(c(1, -1, -1))
reco <- csrec(base = base, cons_mat = C, comb = "wls", res = res)

# Non negative reconciliation
# Making negative one of the base forecasts for variable Y
base[1,3] <- -base[1,3]
nnreco <- csrec(base = base, agg_mat = A, comb = "wls", res = res,
                nn = "osqp")
recoinfo(nnreco, verbose = FALSE)$info
```

---

cssmp — *Cross-sectional probabilistic reconciliation (sample approach)*

Description

This function performs cross-sectional probabilistic forecast reconciliation using a sample-based approach (Panagiotelis et al., 2023, Girolimetto et al., 2024) for linearly constrained (e.g., hierarchical or grouped) multiple time series. Given an array of L simulated base forecast draws, `cssmp()` applies a chosen **FoReco** reconciliation independently to each draw, producing a coherent sample distribution of reconciled forecasts. Typical choices for the reconciliation include optimal combination (csrec) as well as top-down (cstd), middle-out (csmo), bottom-up (csbu), and level-conditional (cslcc) approaches.

Usage

```
cssmp(sample, fun = csrec, ...)
```

Arguments

`sample`	A $(h \times n \times L)$ numeric array containing the base forecasts samples to be reconciled; h is the forecast horizon, n is the total number of time series ($n = n_a + n_b$), and L is the sample size.
`fun`	A string specifying the reconciliation function to be used, as implemented in **FoReco**.
`...`	Arguments passed on to `fun`

Value

A distributional::dist_sample object.

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G. and Hyndman, R.J. (2023), Probabilistic forecast reconciliation: Properties, evaluation and score optimisation, *European Journal of Operational Research* 306(2), 693–706. doi:10.1016/j.ejor.2022.07.040

See Also

Probabilistic reconciliation: `csmvn()`, `ctmvn()`, `ctsmp()`, `temvn()`, `tesmp()`

Cross-sectional framework: `csboot()`, `csbu()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cstd()`, `cstools()`

Examples

```
set.seed(123)
A <- matrix(c(1,1,1,1,  # Z = X + Y
              1,1,0,0,  # X = XX + XY
              0,0,1,1), # Y = YX + YY
              nrow = 3, byrow = TRUE)
rownames(A) <- c("Z", "X", "Y")
colnames(A) <- c("XX", "XY", "YX", "YY")

# (100 x 7) base forecasts sample (simulated) for h = 1
base_h1 <- matrix(rnorm(100*7, mean = c(20, rep(10, 2), rep(5, 4))),
                  100, byrow = TRUE)
# (100 x 7) base forecasts sample (simulated) for h = 2
base_h2 <- matrix(rnorm(100*7, mean = c(20, rep(10, 2), rep(5, 4))),
                  100, byrow = TRUE)
# (2 x 7 x 100) base forecasts sample array with
# 2 forecast horizons, 7 time series and 100 sample
base_sample <- aperm(simplify2array(list(base_h1, base_h2)), c(3,2,1))

# Top-down probabilistic reconciliation
reco_dist_td <- cssmp(base_sample[, 1, , drop = FALSE], agg_mat = A,
```

```
                      fun = cstd, weights = c(0.3, 0.2, 0.1, 0.4))

# Middle-out probabilistic reconciliation
reco_dist_mo <- cssmp(base_sample[, c(2,3), , drop = FALSE], agg_mat = A,
                      fun = csmo, weights = c(0.3, 0.7, 0.8, 0.2),
                      id_rows = 2:3)

# Bottom-up probabilistic reconciliation
reco_dist_bu <- cssmp(base_sample[,-c(1:3),], agg_mat = A, fun = csbu)

# Level conditional coherent probabilistic reconciliation
reco_dist_lcc <- cssmp(base_sample, agg_mat = A, fun = cslcc)

# Optimal cross-sectional probabilistic reconciliation
reco_dist_opt <- cssmp(base_sample, agg_mat = A)
```

---

cstd *Cross-sectional top-down reconciliation*

Description

Top-down forecast reconciliation for genuine hierarchical/grouped time series (Gross and Sohl, 1990), where the forecast of a 'Total' (top-level series, expected to be positive) is disaggregated according to a proportional scheme (weights). Besides fulfilling any aggregation constraint, the top-down reconciled forecasts should respect two main properties:

- the top-level value remains unchanged;
- all the bottom time series reconciled forecasts are non-negative.

Usage

```
cstd(base, agg_mat, weights, normalize = TRUE)
```

Arguments

`base`	A $(h \times 1)$ numeric vector containing the top-level base forecast; h is the forecast horizon.
`agg_mat`	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
`weights`	A $(h \times n_b)$ numeric matrix containing the proportions for the bottom time series; h is the forecast horizon, and n_b is the total number of bottom variables.
`normalize`	If `TRUE` (*default*), the `weights` will sum to 1.

Value

A $(h \times n)$ numeric matrix of cross-sectional reconciled forecasts.

References

Gross, C.W. and Sohl, J.E. (1990), Disaggregation methods to expedite product line forecasting. *Journal of Forecasting* 9(3), 233–254. doi:10.1002/for.3980090304

See Also

Top-down reconciliation: `cttd()`, `tetd()`

Cross-sectional framework: `csboot()`, `csbu()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstools()`

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y, X = XX + XY and Y = YX + YY
A <- matrix(c(1,1,1,1,1,1,0,0,0,0,1,1), 3, byrow = TRUE)
# (3 x 1) top base forecasts vector (simulated), forecast horizon = 3
topf <- rnorm(3, 10)
# Same weights for different forecast horizons
fix_weights <- runif(4)
reco <- cstd(base = topf, agg_mat = A, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- matrix(runif(4*3), 3, 4)
recoh <- cstd(base = topf, agg_mat = A, weights = h_weights)
```

---

cstools — *Cross-sectional reconciliation tools*

Description

Some useful tools for the cross-sectional forecast reconciliation of a linearly constrained (e.g., hierarchical/grouped) multiple time series.

Usage

```
cstools(agg_mat, cons_mat, sparse = TRUE)
```

Arguments

`agg_mat`	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
`cons_mat`	A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.
`sparse`	Option to return sparse matrices (*default* is `TRUE`).

Value

A list with four elements:

`dim`	A vector containing information about the number of series for the complete system (n), for upper levels (na) and bottom level (nb).
`agg_mat`	The cross-sectional aggregation matrix.
`strc_mat`	The cross-sectional structural matrix.
`cons_mat`	The cross-sectional zero constraints matrix.

See Also

Cross-sectional framework: `csboot()`, `csbu()`, `cscov()`, `cslcc()`, `csmo()`, `csmvn()`, `csrec()`, `cssmp()`, `cstd()`

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
# Cross-sectional framework
# One level hierarchy A = [1 1]
A <- matrix(1, 1, 2)
obj <- cstools(agg_mat = A)
```

---

ctboot *Cross-temporal joint block bootstrap*

Description

Joint block bootstrap for generating probabilistic base forecasts that take into account the correlation between variables at different temporal aggregation orders (Girolimetto et al. 2023).

Usage

```
ctboot(model_list, boot_size, agg_order, block_size = 1, seed = NULL,
       xreg = NULL, ...)
```

Arguments

`model_list`	A list of p elements, one for each temporal level ordered from the lowest frequency (most temporally aggregated) to the highest frequency. Each elements is a list with the n base forecasts models for each cross-sectional series. A `simulate()` function for each model has to be available and implemented according to the package **forecast**, with the following mandatory parameters: *object*, *innov*, *future*, and *nsim*.

`boot_size`	The number of bootstrap replicates.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`block_size`	Block size of the bootstrap, which is typically equivalent to the forecast horizon for the most temporally aggregated series.
`seed`	An integer seed.
`xreg`	An optional 3-d numeric array of dimensions $(n \times \text{boot\_size}(k^* + m) \times N_{xreg})$ containing the new values of `xreg` to be used for forecasting ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). It can contain `NA`s.
`...`	Additional arguments for the `simulate()` function.

Value

A list with two elements: the seed used to sample the errors and a list with boot_size matrix of size $(n \times (k^* + m)$block_size) matrix.

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Bootstrap samples: `csboot()`, `teboot()`

Cross-temporal framework: `ctbu()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctrec()`, `ctsmp()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

`ctbu` *Cross-temporal bottom-up reconciliation*

Description

Cross-temporal bottom-up reconciled forecasts for all series at any temporal aggregation level are computed by appropriate summation of the high-frequency bottom base forecasts $\widehat{\mathbf{B}}^{[1]}$:

$$\widetilde{\mathbf{X}} = \mathbf{S}_{cs}\widehat{\mathbf{B}}^{[1]}\mathbf{S}'_{te},$$

where $\mathbf{S}_{cs}$ and $\mathbf{S}_{te}$ are the cross-sectional and temporal structural matrices, respectively.

Usage

```
ctbu(base, agg_mat, agg_order, tew = "sum", sntz = FALSE, round = FALSE)
```

Arguments

- `base` — A $(n_b \times hm)$ numeric matrix containing high-frequency bottom base forecasts; n_b is the total number of high-frequency bottom variables, m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
- `agg_mat` — A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
- `agg_order` — Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
- `tew` — A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
- `sntz` — Logical. If `TRUE`, the negative base forecasts are set to zero (Di Fonzo and Girolimetto, 2023) before applying bottom-up. *Default* is `FALSE`.
- `round` — Logical. If `TRUE`, base forecasts are rounded before applying the bottom-up reconciliation. *Default* is `FALSE`.

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

References

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

See Also

Bottom-up reconciliation: `csbu()`, `tebu()`

Cross-temporal framework: `ctboot()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctrec()`, `ctsmp()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
# (2 x 4) high frequency bottom base forecasts matrix (simulated),
# agg_order = 4 (annual-quarterly)
hfbts <- matrix(rnorm(4*2, 2.5), 2, 4)

reco <- ctbu(base = hfbts, agg_mat = A, agg_order = 4)

# Non negative reconciliation
hfbts[1,4] <- -hfbts[1,4] # Making negative one of the quarterly values for X
nnreco <- ctbu(base = hfbts, agg_mat = A, agg_order = 4, sntz = TRUE)
```

ctcov *Cross-temporal covariance matrix approximation*

Description

This function provides an approximation of the cross-temporal base forecasts errors covariance matrix using different reconciliation methods (Di Fonzo and Girolimetto, 2023, and Girolimetto et al., 2023).

Usage

```
ctcov(comb = "ols", agg_mat = NULL, agg_order = NULL, tew = "sum",
      res = NULL, n = NULL, mse = TRUE, shrink_fun = shrink_estim, ...)
```

Arguments

comb
: A string specifying the reconciliation method.

- For ordinary least squares reconciliation:
  - "ols" (*default*) - identity error covariance.
- For weighted least squares reconciliation:
  - "str" - structural variances.
  - "csstr" - cross-sectional structural variances.
  - "testr" - temporal structural variances.
  - "wlsh" - hierarchy variances (uses res).
  - "wlsv" - series variances (uses res).
- For generalized least squares (uses res) reconciliation:
  - "acov" - series auto-covariance.
  - "bdshr"/"bdsam" - shrunk/sample block diagonal cross-sectional covariance.
  - "Sshr"/"Ssam" - series shrunk/sample covariance.
  - "shr"/"sam" - shrunk/sample covariance.
  - "hbshr"/"hbsam" - shrunk/sample high frequency bottom time series covariance.
  - "bshr"/"bsam" - shrunk/sample bottom time series covariance.
  - "hshr"/"hsam" - shrunk/sample high frequency covariance.
- Others (no for reconciliation):
  - "bu" - bottom-up covariance.

agg_mat
: A ($n_a \times n_b$) numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

agg_order
: Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.

`tew`	A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
`res`	A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.
`n`	Cross-sectional number of variables.
`mse`	If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.
`shrink_fun`	Shrinkage function of the covariance matrix, shrink_estim (*default*).
`...`	Not used.

Value

A $(n(k^* + m) \times n(k^* + m))$ symmetric matrix.

References

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Cross-temporal framework: `ctboot()`, `ctbu()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctrec()`, `ctsmp()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y
A <- t(c(1,1))
# (3 x 70) in-sample residuals matrix (simulated),
# agg_order = 4 (annual-quarterly)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))

cov1 <- ctcov("ols", n = 3, agg_order = 4)                  # OLS
cov2 <- ctcov("str", agg_mat = A, agg_order = 4)            # STR
cov3 <- ctcov("csstr", agg_mat = A, agg_order = 4)          # CSSTR
cov4 <- ctcov("testr", n = 3, agg_order = 4)                # TESTR
cov5 <- ctcov("wlsv", agg_order = 4, res = res)             # WLSv
cov6 <- ctcov("wlsh", agg_order = 4, res = res)             # WLSh
cov7 <- ctcov("shr", agg_order = 4, res = res)              # SHR
cov8 <- ctcov("sam", agg_order = 4, res = res)              # SAM
cov9 <- ctcov("acov", agg_order = 4, res = res)             # ACOV
```

```
cov10 <- ctcov("Sshr", agg_order = 4, res = res)               # Sshr
cov11 <- ctcov("Ssam", agg_order = 4, res = res)               # Ssam
cov12 <- ctcov("hshr", agg_order = 4, res = res)               # Hshr
cov13 <- ctcov("hsam", agg_order = 4, res = res)               # Hsam
cov14 <- ctcov("hbshr", agg_mat = A, agg_order = 4, res = res) # HBshr
cov15 <- ctcov("hbsam", agg_mat = A, agg_order = 4, res = res) # HBsam
cov16 <- ctcov("bshr", agg_mat = A, agg_order = 4, res = res)  # Bshr
cov17 <- ctcov("bsam", agg_mat = A, agg_order = 4, res = res)  # Bsam
cov18 <- ctcov("bdshr", agg_order = 4, res = res)              # BDshr
cov19 <- ctcov("bdsam", agg_order = 4, res = res)              # BDsam

# Custom covariance matrix
ctcov.ols2 <- function(comb, x) diag(x)
cov20 <- ctcov(comb = "ols2", x = 21) # == ctcov("ols", n = 3, agg_order = 4)
```

ctlcc — *Level conditional coherent reconciliation for cross-temporal hierarchies*

Description

This function implements a forecast reconciliation procedure inspired by the original proposal by Hollyman et al. (2021) for cross-temporal hierarchies. Level conditional coherent reconciled forecasts are conditional on (i.e., constrained by) the base forecasts of a specific upper level in the hierarchy (exogenous constraints). It also allows handling the linear constraints linking the variables endogenously (Di Fonzo and Girolimetto, 2022). The function can calculate Combined Conditional Coherent (CCC) forecasts as simple averages of Level-Conditional Coherent (LCC) and bottom-up reconciled forecasts, with either endogenous or exogenous constraints.

Usage

```
ctlcc(base, agg_mat, agg_order, tew = "sum", comb = "ols", res = NULL,
      approach = "proj", nn = NULL, settings = NULL, CCC = TRUE,
      const = "exogenous", hfbts = NULL, ...)
```

Arguments

`base` A $(n \times h(k^*+m))$ numeric matrix containing the base forecasts to be reconciled; n is the total number of variables, m is the maximum aggregation order, and k^* is the sum of a chosen subset of the $p-1$ factors of m (excluding m itself), and h is the forecast horizon for the lowest frequency time series. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.

`agg_mat` A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

agg_order
: Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.

tew
: A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).

comb
: A string specifying the reconciliation method. For a complete list, see ctcov.

res
: A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.

approach
: A string specifying the approach used to compute the reconciled forecasts. Options include:

- "proj" (*default*): Projection approach according to Byron (1978, 1979).
- "strc": Structural approach as proposed by Hyndman et al. (2011).
- "proj_osqp": Numerical solution using **osqp** for projection approach.
- "strc_osqp": Numerical solution using **osqp** for structural approach.

nn
: A string specifying the algorithm to compute non-negative forecasts:

- "osqp": quadratic programming optimization (**osqp** solver, Girolimetto 2025).
- "bpv": block principal pivoting algorithm (Wickramasuriya et al., 2020).
- "nfca": negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
- "nnic": iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
- "sntz": heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).

settings
: A list of control parameters.

- nn = "osqp" An object of class osqpSettings specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
- nn = "bpv"
  - ptype = "fixed": permutation method: "random" or "fixed"
  - par = 10: the number of full exchange rules that may be attempted
  - tol = sqrt(.Machine$double.eps): the tolerance criteria
  - gtol = sqrt(.Machine$double.eps): the gradient tolerance criteria
  - itmax = 100: the maximum number of algorithm iterations
- nn = "nfca" and nn = "nnic"
  - tol = sqrt(.Machine$double.eps): the tolerance criteria
  - itmax = 100: the maximum number of algorithm iterations
- nn = "sntz"
  - type = "bu": the type of set-negative-to-zero heuristic: "bu" for bottom-up, "tdp" for top-down proportional, "tdsp" for top-down square proportional, "tdvw" for top-down variance weighted (the res param is used). See Girolimetto (2025) for details.

- `tol = sqrt(.Machine$double.eps)`: the tolerance identification of negative values

`CCC` A logical value indicating whether the Combined Conditional Coherent reconciled forecasts reconciliation should include bottom-up forecasts (`TRUE`, *default*), or not.

`const` A string specifying the reconciliation constraints:

- `"exogenous"` (*default*): Fixes the top level of each sub-hierarchy.
- `"endogenous"`: Coherently revises both the top and bottom levels.

`hfbts` A $(n \times mh)$ numeric matrix containing high frequency bottom base forecasts defined by the user. This parameter can be omitted if only base forecasts are used (see Di Fonzo and Girolimetto, 2024).

`...` Arguments passed on to `ctcov`

`mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.

`shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2024), Forecast combination-based forecast reconciliation: Insights and extensions, *International Journal of Forecasting*, 40(2), 490–514. doi:10.1016/j.ijforecast.2022.07.001

Di Fonzo, T. and Girolimetto, D. (2023b) Spatio-temporal reconciliation of solar forecasts. *Solar Energy* 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Hollyman, R., Petropoulos, F. and Tipping, M.E. (2021), Understanding forecast reconciliation. *European Journal of Operational Research*, 294, 149–160. doi:10.1016/j.ejor.2021.01.017

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

See Also

Level conditional coherent reconciliation: cslcc(), telcc()

Cross-temporal framework: ctboot(), ctbu(), ctcov(), ctmo(), ctmvn(), ctrec(), ctsmp(), cttd(), cttools(), iterec(), tcsrec()

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y, X = XX + XY and Y = YX + YY
A <- matrix(c(1,1,1,1,1,1,0,0,0,0,1,1), 3, byrow = TRUE)
# (7 x 7) base forecasts matrix (simulated), agg_order = 4
base <- rbind(rnorm(7, rep(c(40, 20, 10), c(1, 2, 4))),
              rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))
# (7 x 70) in-sample residuals matrix (simulated)
res <- matrix(rnorm(70*7), nrow = 7)
# (4 x 4) Naive high frequency bottom base forecasts vector:
# all forecasts are set equal to 2.5
naive <- matrix(2.5, 4, 4)

## EXOGENOUS CONSTRAINTS (Hollyman et al., 2021)
# Level Conditional Coherent (LCC) reconciled forecasts
exo_LC <- ctlcc(base = base, agg_mat = A, agg_order = 4, comb = "wlsh", nn = "osqp",
                hfbts = naive, res = res, CCC = FALSE)

# Combined Conditional Coherent (CCC) reconciled forecasts
exo_CCC <- ctlcc(base = base, agg_mat = A, agg_order = 4, comb = "wlsh",
                hfbts = naive, res = res, CCC = TRUE)

# Results detailed by level:
info_exo <- recoinfo(exo_CCC, verbose = FALSE)
# info_exo$lcc

## ENDOGENOUS CONSTRAINTS (Di Fonzo and Girolimetto, 2024)
# Level Conditional Coherent (LCC) reconciled forecasts
endo_LC <- ctlcc(base = base, agg_mat = A, agg_order = 4, comb = "wlsh",
                 res = res, CCC = FALSE, const = "endogenous")

# Combined Conditional Coherent (CCC) reconciled forecasts
endo_CCC <- ctlcc(base = base, agg_mat = A, agg_order = 4, comb = "wlsh",
                  res = res, CCC = TRUE, const = "endogenous")

# Results detailed by level:
info_endo <- recoinfo(endo_CCC, verbose = FALSE)
# info_endo$lcc
```

ctmo *Cross-temporal middle-out reconciliation*

Description

The cross-temporal middle-out forecast reconciliation combines top-down (cttd) and bottom-up (ctbu) methods in the cross-temporal framework for genuine hierarchical/grouped time series. Given the base forecasts of an intermediate cross-sectional level l and aggregation order k, it performs

- a top-down approach for the aggregation orders $\geq k$ and cross-sectional levels $\geq l$;
- a bottom-up approach, otherwise.

Usage

```
ctmo(base, agg_mat, agg_order, weights, id_rows = 1, order = max(agg_order),
     tew = "sum", normalize = TRUE)
```

Arguments

`base`	A $(n_l \times hk)$ numeric matrix containing the l-level base forecasts of temporal aggregation order k; n_l is the number of variables at level l, k is an aggregation order (a factor of m, and $1 < k < m$), m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`agg_mat`	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`weights`	A $(n_b \times hm)$ numeric matrix containing the proportions for each high-frequency bottom time series; n_b is the total number of high-frequency bottom variables, m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`id_rows`	A numeric vector indicating the l-level rows of `agg_mat`.
`order`	The intermediate fixed aggregation order k.
`tew`	A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
`normalize`	If `TRUE` (*default*), the `weights` will sum to 1.

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

See Also

Middle-out reconciliation: csmo(), temo()

Cross-temporal framework: ctboot(), ctbu(), ctcov(), ctlcc(), ctmvn(), ctrec(), ctsmp(), cttd(), cttools(), iterec(), tcsrec()

Examples

```
set.seed(123)
# Aggregation matrix for Z = X + Y, X = XX + XY and Y = YX + YY
A <- matrix(c(1,1,1,1,1,1,0,0,0,0,1,1), 3, byrow = TRUE)
# (2 x 6) base forecasts matrix (simulated), forecast horizon = 3
# and intermediate aggregation order k = 2 (max agg order = 4)
baseL2k2 <- rbind(rnorm(3*2, 5), rnorm(3*2, 5))

# Same weights for different forecast horizons, agg_order = 4
fix_weights <- matrix(runif(4*4), 4, 4)
reco <- ctmo(base = baseL2k2, id_rows = 2:3, agg_mat = A,
             order = 2, agg_order = 4, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- matrix(runif(4*4*3), 4, 3*4)
recoh <- ctmo(base = baseL2k2, id_rows = 2:3, agg_mat = A,
             order = 2, agg_order = 4, weights = h_weights)
```

---

ctmvn *Cross-temporal Gaussian probabilistic reconciliation*

Description

This function performs cross-temporal probabilistic forecast reconciliation assuming a multivariate normal base forecast distribution (Girolimetto et al., 2024) for linearly constrained multiple time series observed across both cross-sectional and temporal dimensions (Di Fonzo and Girolimetto, 2023).

Usage

```
ctmvn(base, agg_mat, cons_mat, agg_order, tew = "sum", comb = "ols",
      res = NULL, approach = "proj", comb_base = comb,
      reduce_form = FALSE, ...)
```

Arguments

base A $(n \times h(k^* + m))$ numeric matrix containing the base forecasts to be reconciled; n is the total number of variables, m is the maximum aggregation order, and k^* is the sum of a chosen subset of the $p-1$ factors of m (excluding m itself), and h is the forecast horizon for the lowest frequency time series. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.

agg_mat	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
cons_mat	A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
comb	A string specifying the reconciliation method. For a complete list, see ctcov.
res	A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.
approach	A string specifying the approach used to compute the reconciled forecasts. Options include: • "proj" (*default*): Projection approach according to Byron (1978, 1979). • "strc": Structural approach as proposed by Hyndman et al. (2011). • "proj_osqp": Numerical solution using **osqp** for projection approach. • "strc_osqp": Numerical solution using **osqp** for structural approach.
comb_base	A string specifying the base covariance matrix approach. For a complete list, see ctcov. Default is the equal to comb.
reduce_form	A logical parameter indicating whether the function should return the full distribution (FALSE, *default*) or only the distribution corresponding to the high-frequency bottom time series (TRUE).
...	Arguments passed on to ctcov mse If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected. shrink_fun Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A distributional::dist_multivariate_normal object.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G. and Hyndman, R.J. (2023), Probabilistic forecast reconciliation: Properties, evaluation and score optimisation, *European Journal of Operational Research* 306(2), 693–706. doi:10.1016/j.ejor.2022.07.040

See Also

Probabilistic reconciliation: `csmvn()`, `cssmp()`, `ctsmp()`, `temvn()`, `tesmp()`

Cross-temporal framework: `ctboot()`, `ctbu()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctrec()`, `ctsmp()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

Examples

```
set.seed(123)
# (3 x 7) base forecasts matrix (simulated), Z = X + Y and m = 4
base <- rbind(rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))
# (3 x 70) in-sample residuals matrix (simulated)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))
A <- t(c(1,1))
reco_dist <- ctmvn(base = base, res = res, agg_mat = A, agg_order = 4)
```

---

ctprojmat — *Projection matrix for optimal combination cross-temporal reconciliation*

Description

This function computes the projection or the mapping matrix $\mathbf{M}$ and $\mathbf{G}$, respectively, such that $\widetilde{\mathbf{y}} = \mathbf{M}\widehat{\mathbf{y}} = \mathbf{S}_{ct}\mathbf{G}\widehat{\mathbf{y}}$, where $\widetilde{\mathbf{y}}$ is the vector of the reconciled forecasts, $\widehat{\mathbf{y}}$ is the vector of the base forecasts, $\mathbf{S}_{ct}$ is the cross-temporal structural matrix, and $\mathbf{M} = \mathbf{S}_{ct}\mathbf{G}$. For further information regarding on the structure of these matrices, refer to Girolimetto et al. (2023).

Usage

```
ctprojmat(agg_mat, cons_mat, agg_order, comb = "ols", res = NULL,
          mat = "M", tew = "sum", ...)
```

Arguments

`agg_mat` A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.

`cons_mat` A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.

`agg_order` Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.

`comb` A string specifying the reconciliation method. For a complete list, see ctcov.

`res` A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.

`mat` A string specifying which matrix to return: "M" (*default*) for $\mathbf{M}$ and "G" for $\mathbf{G}$.

`tew` A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).

`...` Arguments passed on to `ctcov`

- `mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.
- `shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

The projection matrix $\mathbf{M}$ (`mat = "M"`) or the mapping matrix $\mathbf{G}$ (`mat = "G"`).

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
# Cross-temporal framework (Z = X + Y, annual-quarterly)
A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
```

```
Mct <- ctprojmat(agg_mat = A, agg_order = 4, comb = "ols")
Gct <- ctprojmat(agg_mat = A, agg_order = 4, comb = "ols", mat = "G")
```

ctrec *Optimal combination cross-temporal reconciliation*

Description

This function performs optimal (in least squares sense) combination cross-temporal forecast reconciliation (Di Fonzo and Girolimetto 2023a, Girolimetto et al. 2023). The reconciled forecasts are calculated using either a projection approach (Byron, 1978, 1979) or the equivalent structural approach by Hyndman et al. (2011). Non-negative (Di Fonzo and Girolimetto, 2023) and immutable reconciled forecasts can be considered.

Usage

```
ctrec(base, agg_mat, cons_mat, agg_order, tew = "sum", comb = "ols",
      res = NULL, approach = "proj", nn = NULL, settings = NULL,
      bounds = NULL, immutable = NULL, ...)
```

Arguments

base	A $(n \times h(k^* + m))$ numeric matrix containing the base forecasts to be reconciled; n is the total number of variables, m is the maximum aggregation order, and k^* is the sum of a chosen subset of the $p-1$ factors of m (excluding m itself), and h is the forecast horizon for the lowest frequency time series. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.
agg_mat	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
cons_mat	A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
comb	A string specifying the reconciliation method. For a complete list, see ctcov.
res	A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.

`approach` A string specifying the approach used to compute the reconciled forecasts. Options include:

- "`proj`" (*default*): Projection approach according to Byron (1978, 1979).
- "`strc`": Structural approach as proposed by Hyndman et al. (2011).
- "`proj_osqp`": Numerical solution using **osqp** for projection approach.
- "`strc_osqp`": Numerical solution using **osqp** for structural approach.

`nn` A string specifying the algorithm to compute non-negative forecasts:

- "`osqp`": quadratic programming optimization (**osqp** solver, Girolimetto 2025).
- "`bpv`": block principal pivoting algorithm (Wickramasuriya et al., 2020).
- "`nfca`": negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
- "`nnic`": iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
- "`sntz`": heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).

`settings` A list of control parameters.

- `nn = "osqp"` An object of class `osqpSettings` specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
- `nn = "bpv"`
  - `ptype = "fixed"`: permutation method: "`random`" or "`fixed`"
  - `par = 10`: the number of full exchange rules that may be attempted
  - `tol = sqrt(.Machine$double.eps)`: the tolerance criteria
  - `gtol = sqrt(.Machine$double.eps)`: the gradient tolerance criteria
  - `itmax = 100`: the maximum number of algorithm iterations
- `nn = "nfca"` and `nn = "nnic"`
  - `tol = sqrt(.Machine$double.eps)`: the tolerance criteria
  - `itmax = 100`: the maximum number of algorithm iterations
- `nn = "sntz"`
  - `type = "bu"`: the type of set-negative-to-zero heuristic: "`bu`" for bottom-up, "`tdp`" for top-down proportional, "`tdsp`" for top-down square proportional, "`tdvw`" for top-down variance weighted (the `res` param is used). See Girolimetto (2025) for details.
  - `tol = sqrt(.Machine$double.eps)`: the tolerance identification of negative values

`bounds` A matrix (see set_bounds) with 5 columns $(i, k, j, lower, upper)$, such that

- Column 1 represents the cross-sectional series ($i = 1, \dots, n$).
- Column 2 represents the temporal aggregation order ($k = m, \dots, 1$).
- Column 3 represents the temporal forecast horizon ($j = 1, \dots, m/k$).
- Columns 4 and 5 indicates the *lower* and *upper* bounds, respectively.

`immutable` A matrix with three columns (i, k, j), such that

- Column 1 represents the cross-sectional series ($i = 1, \dots, n$).

- Column 2 represents the temporal aggregation order ($k = m, \dots, 1$).
- Column 3 represents the temporal forecast horizon ($j = 1, \dots, m/k$).

For example, when working with a quarterly multivariate time series ($n = 3$):

- `t(c(1, 4, 1))` - Fix the one step ahead annual forecast of the first time series.
- `t(c(2, 1, 2))` - Fix the two step ahead quarterly forecast of the second time series.

`...` Arguments passed on to `ctcov`

`mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.

`shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*).

Value

A ($n \times h(k^* + m)$) numeric matrix of cross-temporal reconciled forecasts.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

Wickramasuriya, S. L., Turlach, B. A., and Hyndman, R. J. (2020). Optimal non-negative forecast reconciliation. *Statistics and Computing*, 30(5), 1167–1182. doi:10.1007/s11222020099300

See Also

Regression-based reconciliation: `csrec()`, `terec()`

Cross-temporal framework: `ctboot()`, `ctbu()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctsmp()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

Examples

```
set.seed(123)
# (3 x 7) base forecasts matrix (simulated), Z = X + Y and m = 4
base <- rbind(rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))
# (3 x 70) in-sample residuals matrix (simulated)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))

A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
m <- 4 # from quarterly to annual temporal aggregation
reco <- ctrec(base = base, agg_mat = A, agg_order = m,
              comb = "wlsv", res = res)

C <- t(c(1, -1, -1)) # Zero constraints matrix for Z - X - Y = 0
reco <- ctrec(base = base, cons_mat = C, agg_order = m,
              comb = "wlsv", res = res)

# Immutable reconciled forecasts
# Fix all the quarterly forecasts of the second variable.
imm_mat <- expand.grid(i = 2, k = 1, j = 1:4)
immreco <- ctrec(base = base, cons_mat = C, agg_order = m, comb = "wlsv",
                 res = res, immutable = imm_mat)

# Non negative reconciliation
# Making negative one of the quarterly base forecasts for variable X
base[2,7] <- -2*base[2,7]
nnreco <- ctrec(base = base, cons_mat = C, agg_order = m, comb = "wlsv",
                res = res, nn = "osqp")
recoinfo(nnreco, verbose = FALSE)$info
```

`ctsmp` *Cross-temporal probabilistic reconciliation (sample approach)*

Description

This function performs cross-temporal probabilistic forecast reconciliation using a sample-based approach (Girolimetto et al., 2024) for linearly constrained multiple time series observed across both cross-sectional and temporal dimensions (Di Fonzo and Girolimetto, 2023). Given a $(n \times h(k^* + m) \times L)$ array of simulated base forecast draws, `ctsmp()` applies a chosen **FoReco** cross-temporal reconciliation independently to each draw, enforcing coherence simultaneously over aggregation levels and across series. Typical choices for the reconciliation include optimal combination (ctrec) as well as top-down (cttd), middle-out (ctmo), bottom-up (ctbu), and level-conditional (ctlcc) approaches.

Usage

```
ctsmp(sample, agg_order, fun = ctrec, ...)
```

Arguments

sample	A $(n \times h(k^* + m) \times L)$ numeric array containing the base forecasts samples to be reconciled; n is the total number of variables, m is the max. order of temporal aggregation, k^* is the sum of (a subset of) $(p-1)$ factors of m, excluding m, h is the forecast horizon for the lowest frequency time series, and L is the sample size. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
fun	A string specifying the reconciliation function to be used, as implemented in **FoReco**.
...	Arguments passed on to `fun`

Value

A distributional::dist_sample object.

References

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Probabilistic reconciliation: `csmvn()`, `cssmp()`, `ctmvn()`, `temvn()`, `tesmp()`

Cross-temporal framework: `ctboot()`, `ctbu()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctrec()`, `cttd()`, `cttools()`, `iterec()`, `tcsrec()`

Examples

```
set.seed(123)
A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
m <- 4 # from quarterly to annual temporal aggregation

# (100 x 14) base forecasts sample matrix (simulated), m = 4, h = 2, n = 3
sample <- simplify2array(lapply(1:100, function(x){
```

```
  rbind(rnorm(14, rep(c(20, 10, 5), 2*c(1, 2, 4))),
        rnorm(14, rep(c(10, 5, 2.5), 2*c(1, 2, 4))),
        rnorm(14, rep(c(10, 5, 2.5), 2*c(1, 2, 4))))
}))
# (3 x 70) in-sample residuals matrix (simulated)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))

# Top-down probabilistic reconciliation
reco_dist_td <- ctsmp(sample[1, 1:2, , drop = FALSE], agg_order = m,
                      agg_mat = A, fun = cttd, weights = matrix(runif(8), 2))

# Middle-out probabilistic reconciliation
reco_dist_mo <- ctsmp(sample[1, 3:6, , drop = FALSE], agg_order = m,
                      agg_mat = A, fun = ctmo, weights = matrix(runif(8), 2),
                      id_rows = 1, order = 2)
# Bottom-up probabilistic reconciliation
reco_dist_bu <- ctsmp(sample[-1,-c(1:6), ], agg_order = m, agg_mat = A,
                      fun = ctbu)

# Level conditional coherent probabilistic reconciliation
reco_dist_lcc <- ctsmp(sample, agg_order = m, agg_mat = A, fun = ctlcc)

# Optimal cross-sectional probabilistic reconciliation
reco_dist_opt <- ctsmp(sample, agg_order = m, agg_mat = A, res = res,
                       comb = "bdshr")
```

---

cttd *Cross-temporal top-down reconciliation*

Description

Top-down forecast reconciliation for cross-temporal hierarchical/grouped time series, where the forecast of a 'Total' (top-level series, expected to be positive) is disaggregated according to a proportional scheme (weights). Besides fulfilling any aggregation constraint, the top-down reconciled forecasts should respect two main properties:

- the top-level value remains unchanged;
- all the bottom time series reconciled forecasts are non-negative.

Usage

```
cttd(base, agg_mat, agg_order, weights, tew = "sum", normalize = TRUE)
```

Arguments

base — A $(hm \times 1)$ numeric vector containing top- and m temporal aggregated level base forecasts; m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.

agg_mat	A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
weights	A $(n_b \times hm)$ numeric matrix containing the proportions for each high-frequency bottom time series; n_b is the total number of high-frequency bottom variables, m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
normalize	If TRUE (*default*), the weights will sum to 1.

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

See Also

Top-down reconciliation: cstd(), tetd()

Cross-temporal framework: ctboot(), ctbu(), ctcov(), ctlcc(), ctmo(), ctmvn(), ctrec(), ctsmp(), cttools(), iterec(), tcsrec()

Examples

```
set.seed(123)
# (3 x 1) top base forecasts vector (simulated), forecast horizon = 3
topf <- rnorm(3, 10)
A <- t(c(1,1)) # Aggregation matrix for Z = X + Y

# Same weights for different forecast horizons, agg_order = 4
fix_weights <- matrix(runif(4*2), 2, 4)
reco <- cttd(base = topf, agg_mat = A, agg_order = 4, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- matrix(runif(4*2*3), 2, 3*4)
recoh <- cttd(base = topf, agg_mat = A, agg_order = 4, weights = h_weights)
```

---

cttools — *Cross-temporal reconciliation tools*

Description

Some useful tools for the cross-temporal forecast reconciliation of a linearly constrained (e.g., hierarchical/grouped) multiple time series.

Usage

```
cttools(agg_mat, cons_mat, agg_order, tew = "sum", fh = 1, sparse = TRUE)
```

Arguments

- `agg_mat` A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
- `cons_mat` A $(n_a \times n)$ numeric matrix representing the cross-sectional zero constraints: each row represents a constraint equation, and each column represents a variable. The matrix can be of full rank, meaning the rows are linearly independent, but this is not a strict requirement, as the function allows for redundancy in the constraints.
- `agg_order` Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
- `tew` A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
- `fh` Forecast horizon for the lowest frequency (most temporally aggregated) time series (*default* is 1).
- `sparse` Option to return sparse matrices (*default* is TRUE).

Value

A list with four elements:

- `dim` A vector containing information about the number of series for the complete system (n), for upper levels (na) and bottom level (nb), the maximum aggregation order (m), the number of factor (p), the partial (ks) and total sum (kt) of factors.
- `set` The vector of the temporal aggregation orders (in decreasing order).
- `agg_mat` The cross-temporal aggregation matrix.
- `strc_mat` The cross-temporal structural matrix.
- `cons_mat` The cross-temporal zero constraints matrix.

See Also

Cross-temporal framework: ctboot(), ctbu(), ctcov(), ctlcc(), ctmo(), ctmvn(), ctrec(), ctsmp(), cttd(), iterec(), tcsrec()

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
# Cross-temporal framework
A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
m <- 4 # from quarterly to annual temporal aggregation
cttools(agg_mat = A, agg_order = m)
```

---

df2aggmat *Cross-sectional aggregation matrix of a dataframe*

Description

This function allows the user to easily build the $(n_a \times n_b)$ cross-sectional aggregation matrix starting from a data frame.

Usage

```
df2aggmat(formula, data, sep = "_", sparse = TRUE, top_label = "Total",
          verbose = TRUE)
```

Arguments

formula	Specification of the hierarchical structure: grouped hierarchies are specified using ~ g1 * g2 and nested hierarchies are specified using ~ parent / child. Mixtures of the two formulations are also possible, like ~ g1 * (grandparent / parent / child).
data	A dataset in which each column contains the values of the variables in the formula and each row identifies a bottom level time series.
sep	Character to separate the names of the aggregated series, (*default* is "_").
sparse	Option to return sparse matrices (*default* is TRUE).
top_label	Label of the top level variable (*default* is "Total").
verbose	If TRUE (*default*), hierarchy informations are printed.

Value

A (na x nb) matrix.

See Also

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
## Balanced hierarchy
#         T
#    |--------|
#    A        B
#  |---|   |--|--|
# AA   AB  BA BB BC
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "B", "B", "B"),
                       X2 = c("A", "B", "A", "B", "C"),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
agg_mat <- df2aggmat(~ X1 / X2, data_bts, sep = "", verbose = FALSE)

## Unbalanced hierarchy
#                 T
#       |---------|---------|
#       A         B         C
#     |---|     |---|     |---|
#    AA   AB   BA   BB   CA   CB
#  |----|         |----|
# AAA  AAB       BBA  BBB
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "A", "B", "B", "B", "C", "C"),
                       X2 = c("A", "A", "B", "A", "B", "B", "A", "B"),
                       X3 = c("A", "B", NA, NA, "A", "B", NA, NA),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
agg_mat <- df2aggmat(~ X1 / X2 / X3, data_bts, sep = "", verbose = FALSE)

## Group of two hierarchies
#     T          T         T | A  | B  | C
#  |--|--|  X  |---|  ->  ---+----+----+----
#  A  B  C     M   F       M | AM | BM | CM
#                          F | AF | BF | CF
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "B", "B", "C", "C"),
                       Y1 = c("M", "F", "M", "F", "M", "F"),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
agg_mat <- df2aggmat(~ Y1 * X1, data_bts, sep = "", verbose = FALSE)
```

`FoReco2matrix` *Reconciled forecasts to matrix/vector*

Description

This function splits the temporal vectors and the cross-temporal matrices in a list according to the temporal aggregation order

Usage

```
FoReco2matrix(x, agg_order, keep_names = FALSE, temporal_names = NULL)
```

Arguments

x	An output from any reconciliation function implemented by **FoReco**.
agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
keep_names	If FALSE (*default*), the rownames names of the output matrices are removed.
temporal_names	A character vector containing the names of the temporal aggregation levels.

Value

A list of matrices or vectors distinct by temporal aggregation order.

See Also

Utilities: aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), tetools(), unbalance_hierarchy()

Examples

```
set.seed(123)
# (3 x 7) base forecasts matrix (simulated), Z = X + Y and m = 4
base <- rbind(rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))

reco <- ctrec(base = base, agg_mat = t(c(1,1)), agg_order = 4, comb = "ols")
matrix_list <- FoReco2matrix(reco)

# With temporal names
temporal_names <- c("Annual", "Semi-annual", "Quarterly")
matrix_list <- FoReco2matrix(reco, temporal_names = temporal_names)
```

---

itagdp *Italian Quarterly National Accounts*

Description

A subset of the data used by Girolimetto et al. (2023) from the Italian Quarterly National Accounts (output, income and expenditure sides) spanning the period 2000:Q1-2019:Q4.

Usage

```
# 21 time series of the Italian Quarterly National Accounts
itagdp

# 'agg_mat' and 'cons_mat' for the output side
outside

# 'agg_mat' and 'cons_mat' for the expenditure side
expside

# 'agg_mat' and 'cons_mat' for the income side
incside

# zero constraints matrix encompassing output, expenditure and income sides
gdpconsmat
```

Format

`itagdp` is a (80×21) `ts` object, corresponding to 21 time series of the Italian Quarterly National Accounts (2000:Q1-2019:Q4).

`outside`, `income` and `expenditure` are lists with two elements:

- `agg_mat` contains the (1×2), (2×4), or (6×8) aggregation matrix according to output, income or expenditure side, respectively.
- `cons_mat` contains the (1×3), (2×6), or (6×14) zero constraints matrix according to output, income or expenditure side, respectively.

`gdpconsmat` is the complete (9×21) zero constraints matrix encompassing output, expenditure and income sides.

Source

https://ec.europa.eu/eurostat/web/national-accounts/

References

Girolimetto, D. and Di Fonzo, T. (2023), Point and probabilistic forecast reconciliation for general linearly constrained multiple time series, *Statistical Methods & Applications*, 33, 581-607. doi:10.1007/s10260023007386.

`iterec` *Iterative cross-temporal reconciliation*

Description

This function performs the iterative procedure described in Di Fonzo and Girolimetto (2023), which produces cross-temporally reconciled forecasts by alternating forecast reconciliation along one single dimension (either cross-sectional or temporal) at each iteration step.

Usage

```
iterec(base, cslist, telist, res = NULL, itmax = 100, tol = 1e-5,
       type = "tcs", norm = "inf", verbose = TRUE)
```

Arguments

- `base` — A $(n \times h(k^* + m))$ numeric matrix containing the base forecasts to be reconciled; n is the total number of variables, m is the maximum aggregation order, and k^* is the sum of a chosen subset of the $p-1$ factors of m (excluding m itself), and h is the forecast horizon for the lowest frequency time series. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.
- `cslist` — A list of elements for the cross-sectional reconciliation. See csrec for a complete list (excluded `base` and `res`).
- `telist` — A list of elements for the temporal reconciliation. See terec for a complete list (excluded `base` and `res`).
- `res` — A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.
- `itmax` — Max number of iteration (`100`, *default*).
- `tol` — Convergence tolerance (`1e-5`, *default*).
- `type` — A string specifying the uni-dimensional reconciliation order: temporal and then cross-sectional (`"tcs"`) or cross-sectional and then temporal (`"cst"`).
- `norm` — Norm used to calculate the temporal and the cross-sectional incoherence: infinity norm (`"inf"`, *default*), one norm (`"one"`), and 2-norm (`"two"`).
- `verbose` — If `TRUE`, reconciliation information are printed.

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

References

Di Fonzo, T. and Girolimetto, D. (2023), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

See Also

Cross-temporal framework: `ctboot()`, `ctbu()`, `ctcov()`, `ctlcc()`, `ctmo()`, `ctmvn()`, `ctrec()`, `ctsmp()`, `cttd()`, `cttools()`, `tcsrec()`

Examples

```
set.seed(123)
# (3 x 7) base forecasts matrix (simulated), Z = X + Y and m = 4
base <- rbind(rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))
# (3 x 70) in-sample residuals matrix (simulated)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))

A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
m <- 4 # from quarterly to annual temporal aggregation

rite <- iterec(base = base,
               cslist = list(agg_mat = A, comb = "shr"),
               telist = list(agg_order = m, comb = "wlsv"),
               res = res)
```

`lcmat` *Linear combination (aggregation) matrix for a general linearly constrained multiple time series*

Description

This function transforms a general (possibly redundant) zero constraints matrix into a linear combination (aggregation) matrix $\mathbf{A}_{cs}$. When working with a general linearly constrained multiple (n-variate) time series, getting a linear combination matrix $\mathbf{A}_{cs}$ is a critical step to obtain a structural-like representation such that

$$\mathbf{C}_{cs} = [\mathbf{I} \quad -\mathbf{A}],$$

where $\mathbf{C}_{cs}$ is the full rank zero constraints matrix (Girolimetto and Di Fonzo, 2023).

Usage

```
lcmat(cons_mat, method = "rref", tol = sqrt(.Machine$double.eps),
      verbose = FALSE, sparse = TRUE)
```

Arguments

`cons_mat`	A $(r \times n)$ numeric matrix representing the cross-sectional zero constraints.
`method`	Method to use: "rref" for the Reduced Row Echelon Form through Gauss-Jordan elimination (*default*), or "qr" for the (pivoting) QR decomposition (Strang, 2019).
`tol`	Tolerance for the "rref" or "qr" method.
`verbose`	If `TRUE`, intermediate steps are printed (*default* is `FALSE`).
`sparse`	Option to return a sparse matrix (*default* is `TRUE`).

Value

A list with two elements: (i) the linear combination (aggregation) matrix (`agg_mat`) and (ii) the vector of the column permutations (`pivot`).

References

Girolimetto, D. and Di Fonzo, T. (2023), Point and probabilistic forecast reconciliation for general linearly constrained multiple time series, *Statistical Methods & Applications*, 33, 581-607. doi:10.1007/s10260023007386.

Strang, G. (2019), *Linear algebra and learning from data*, Wellesley, Cambridge Press.

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
## Two hierarchy sharing the same top-level variable, but not sharing the bottom variables
#        X            X
#    |-------|    |-------|
#    A       B    C       D
#  |---|
# A1   A2
# 1) X = C + D,
# 2) X = A + B,
# 3) A = A1 + A2.
cons_mat <- matrix(c(1,-1,-1,0,0,0,0,
               1,0,0,-1,-1,0,0,
               0,0,0,1,0,-1,-1), nrow = 3, byrow = TRUE)
obj <- lcmat(cons_mat = cons_mat, verbose = TRUE)
agg_mat <- obj$agg_mat # linear combination matrix
pivot <- obj$pivot # Pivot vector
```

`new_foreco_info` *Low-level construction for reconcilied forecasts attribute foreco_info class*

Description

`new_foreca_info()` is the contructor for the `foreca_info` class, which acompany the output from the reconciliation functions in the attribute `FoReco`. This is exported for extension purposes and for expert use only.

Usage

```
new_foreco_info(x = list())
```

Arguments

x — A list of information related to the reconcilied forecasts.

Examples

```
new_foreco_info(list(
  framework = "Cross-sectional",
  forecast_horizon = 1,
  comb = "shr",
  cs_n = 3,
  rfun = "csrec"
))
```

---

recoinfo — *Informations on the reconciliation process*

Description

This function extracts reconciliation information from the output of any reconciled function implemented by **FoReco**.

Usage

```
recoinfo(x, verbose = TRUE)
```

Arguments

`x`	An output from any reconciliation function implemented by **FoReco**.
`verbose`	If `TRUE` (*defaults*), reconciliation information are printed.

Value

A list containing the following reconciliation process information:

`rfun`	the reconciliation function.
`cs_n`	the cross-sectional number of variables.
`te_set`	the set of temporal aggregation orders.
`forecast_horizon`	the forecast horizon (in temporal and cross-temporal frameworks, for the most temporally aggregated series).
`framework`	the reconciliation framework (cross-sectional, temporal or cross-temporal).
`info`	non-negative reconciled forecast convergence information.
`lcc`	list of level conditional reconciled forecasts (+ BU) for cslcc, telcc and ctlcc.
`nn`	if `TRUE`, all the forecasts are not negative.
`comb`	the covariance approximation.

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

`res2matrix` *One-step and multi-step residuals*

Description

These functions can be used to arrange residuals to reconcile temporal or cross-temporal forecasts.

res2matrix **[Deprecated]** takes as input a set of temporal and cross-temporal residuals and reorganizes them into a matrix where the rows correspond to different forecast horizons, capturing the temporal dimension. Meanwhile, the columns are ordered based on the specific arrangement as described in Di Fonzo and Girolimetto (2023). Please see as_hstack_ctlayout and as_hstack_telayout.

arrange_hres takes as input a list of multi-step residuals and is designed to organize them in accordance with their time order (Girolimetto et al. 2023). When applied, this function ensures that the sequence of multi-step residuals aligns with the chronological order in which they occurred.

Usage

```
res2matrix(res, agg_order)

arrange_hres(list_res)
```

Arguments

`res`	A $(n \times N(k^* + m))$ numeric matrix (cross-temporal framework) or an $(N(k^* + m) \times 1)$ numeric vector (temporal framework) representing the in-sample residuals or validation errors at all the temporal frequencies ordered from the lowest frequency to the highest frequency (columns) for each variable (rows).
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`list_res`	A list of H multi-step residuals. Each element in the list can be either a $(T \times 1)$ vector (temporal framework) or a $(T \times n)$ matrix (cross-temporal framework).

Details

Let Z_t, $t = 1, \dots, T$, be a univariate time series. We can define the multi-step residuals such us

$$\widehat{\varepsilon}_{h,t} = Z_{t+h} - \widehat{Z}_{t+h|t} \qquad h \leq t \leq T - h$$

where $\widehat{Z}_{t+h|t}$ is the h-step fitted value, calculated as the h-step ahead forecast condition to the information up to time t. Given the list of errors at different steps

$$\left([\widehat{\varepsilon}_{1,1}, \; \dots, \; \widehat{\varepsilon}_{1,T}], \dots, [\widehat{\varepsilon}_{H,1}, \; \dots, \; \widehat{\varepsilon}_{H,T}]\right),$$

arrange_hres returns a T-vector with the residuals, organized in the following way:

$$[\varepsilon_{1,1} \; \varepsilon_{2,2} \; \dots \; \varepsilon_{H,H} \; \varepsilon_{1,H+1} \; \dots \; \varepsilon_{H,T-H}]'$$

A similar organisation can be apply to a multivariate time series.

Value

res2matrix returns a $(N \times n(k^* + m))$ matrix, where $n = 1$ for the temporal framework.

arrange_hres returns a $(N(k^* + m) \times 1)$ vector (temporal framework) or a $(n \times N(k^* + m))$ matrix (cross-temporal framework) of multi-step residuals.

References

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
h <- 10
agg_order <- 4
tmp <- tetools(agg_order)
kt <- tmp$dim["kt"]

# Simulate vector (temporal case)
vec <- rnorm(kt*h)
out <- res2matrix(vec, agg_order) # matrix h x kt

# Simulate (n x kt) matrix (cross-temporal case) with n = 3
mat <- rbind(rnorm(kt*h), rnorm(kt*h), rnorm(kt*h))
out <- res2matrix(mat, agg_order) # matrix h x (3*kt)

# Input: 4 (forecast horizons) vectors with 4*10 elements
input <-  list(rnorm(4*10), rnorm(4*10), rnorm(4*10), rnorm(4*10))
# Output: 1 vector with 4*10 elements
out <- arrange_hres(input)

# Matrix version
input <-  list(matrix(rnorm(4*10*3), 4*10), matrix(rnorm(4*10*3), 4*10),
               matrix(rnorm(4*10*3), 4*10), matrix(rnorm(4*10*3), 4*10))
out <- arrange_hres(input)
```

`set_bounds` *Set bounds for bounded forecast reconciliation*

Description

This function defines the bounds matrix considering cross-sectional, temporal, or cross-temporal frameworks. The output matrix can be used as input for the `bounds` parameter in functions such as csrec, terec, or ctrec, to perform bounded reconciliations.

Usage

```
set_bounds(n, k, h, lb = -Inf, ub = Inf, approach = "osqp", bounds = NULL)
```

Arguments

`n`	A $(b \times 1)$ vector representing the ith cross-sectional series $(i = 1, \dots, n)$, where b is the number of bounds to be set.
`k`	A $(b \times 1)$ vector specifying the temporal aggregation orders $(k = m, \dots, 1)$.
`h`	A $(b \times 1)$ vector representing the forecast horizons $(j = 1, \dots, m/k)$.
`lb, ub`	A $(b \times 1)$ vector of lower and upper bounds.
`approach`	A string specifying the algorithm to compute bounded reconciled forecasts: • `"osqp"`: quadratic programming optimization (**osqp** solver). • `"sftb"`: heuristic "set-forecasts-to-bounds", which adjusts the reconciled forecasts to be within specified bounds without further optimization.
`bounds`	A matrix of previous bounds to be added. If not specified, new bounds will be computed.

Value

A numeric matrix representing the computed bounds, which can be:

- Cross-sectional $(b \times 3)$ matrix for cross-sectional reconciliation (csrec).
- Temporal $(b \times 4)$ matrix for temporal reconciliation (terec).
- Cross-temporal $(b \times 5)$ matrix for cross-temporal reconciliation (ctrec).

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

Examples

```
# Example 1
# Two cross-sectional series (i = 2,3),
# with each series required to be between 0 and 1.
n <- c(2, 3)
lb <- c(0, 0)
ub <- c(1,1)
bounds_mat <- set_bounds(n = c(2, 3),
                         lb = rep(0, 2), # or lb = 0
                         ub = rep(1, 2)) # or ub = 1

# Example 2
# All the monthly values are between 0 and 1.
bounds_mat <- set_bounds(k = rep(1, 12),  # or k = 1
                         h = 1:12,
                         lb = rep(0, 12), # or lb = 0
                         ub = rep(1, 12)) # or ub = 1

# Example 3
# For two cross-sectional series (i = 2,3),
# all the monthly values are between 0 and 1.
bounds_mat <- set_bounds(n = rep(c(2, 3), each = 12),
                         k = 1,
                         h = rep(1:12, 2),
                         lb = 0, # or lb = 0
                         ub = 1) # or ub = 1
```

`shrink_estim` *Shrinkage of the covariance matrix*

Description

Shrinkage of the covariance matrix according to Schäfer and Strimmer (2005).

Usage

```
shrink_estim(x, mse = TRUE)
```

Arguments

x	A numeric matrix containing the in-sample residuals or validation errors.
mse	If TRUE (*default*), the residuals used to compute the covariance matrix are not mean-corrected.

Value

A shrunk covariance matrix.

References

Schäfer, J.L. and Strimmer, K. (2005), A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1. doi:10.2202/15446115.1175

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_oasd()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

---

shrink_oasd — *Shrinkage of the covariance matrix using the Oracle approximation*

Description

Shrinkage of the covariance matrix according to the Oracle Approximating Shrinkage (OAS) of Chen et al. (2009) and Ando and Xiao (2023).

Usage

```
shrink_oasd(x, mse = TRUE)
```

Arguments

`x`	A numeric matrix containing the in-sample residuals or validation errors.
`mse`	If `TRUE` (*default*), the residuals used to compute the covariance matrix are not mean-corrected.

Value

A shrunk covariance matrix.

References

Ando, S., and Xiao, M. (2023), High-dimensional covariance matrix estimation: shrinkage toward a diagonal target. *IMF Working Papers*, 2023(257), A001.

Chen, Y., Wiesel, A., and Hero, A. O. (2009), Shrinkage estimation of high dimensional covariance matrices, *2009 IEEE international conference on acoustics, speech and signal processing*, 2937–2940. IEEE.

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `teprojmat()`, `tetools()`, `unbalance_hierarchy()`

`tcsrec` *Heuristic cross-temporal reconciliation*

Description

tcsrec replicates the procedure by Kourentzes and Athanasopoulos (2019): (i) for each time series the forecasts at any temporal aggregation order are reconciled using temporal hierarchies; (ii) time-by-time cross-sectional reconciliation is performed; and (iii) the projection matrices obtained at step (ii) are then averaged and used to cross-sectionally reconcile the forecasts obtained at step (i). In cstrec, the order of application of the two reconciliation steps (temporal first, then cross-sectional), is inverted compared to tcsrec (Di Fonzo and Girolimetto, 2023).

Usage

```
# First-temporal-then-cross-sectional forecast reconciliation
tcsrec(base, cslist, telist, res = NULL, avg = "KA")

# First-cross-sectional-then-temporal forecast reconciliation
cstrec(base, cslist, telist, res = NULL)
```

Arguments

`base`	A $(n \times h(k^*+m))$ numeric matrix containing the base forecasts to be reconciled; n is the total number of variables, m is the maximum aggregation order, and k^* is the sum of a chosen subset of the $p-1$ factors of m (excluding m itself), and h is the forecast horizon for the lowest frequency time series. The row identifies a time series, and the forecasts in each row are ordered from the lowest frequency (most temporally aggregated) to the highest frequency.
`cslist`	A list of elements for the cross-sectional reconciliation. See csrec for a complete list (excluded `base` and `res`).
`telist`	A list of elements for the temporal reconciliation. See terec for a complete list (excluded `base` and `res`).
`res`	A $(n \times N(k^* + m))$ optional numeric matrix containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency (columns) for each variable (rows). This matrix is used to compute some covariance matrices.
`avg`	If `avg = "KA"` (*default*), the final projection matrix $\mathbf{M}$ is the one proposed by Kourentzes and Athanasopoulos (2019), otherwise it is calculated as simple average of all the involved projection matrices at step 2 of the procedure (see Di Fonzo and Girolimetto, 2023).

Value

A $(n \times h(k^* + m))$ numeric matrix of cross-temporal reconciled forecasts.

Warning

The two-step heuristic reconciliation allows considering non negativity constraints only in the first step. This means that non-negativity is not guaranteed in the final reconciled values.

References

Di Fonzo, T. and Girolimetto, D. (2023), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

Kourentzes, N. and Athanasopoulos, G. (2019), Cross-temporal coherent forecasts for Australian tourism, *Annals of Tourism Research*, 75, 393-409. doi:10.1016/j.annals.2019.02.001

See Also

Cross-temporal framework: `ctboot`(), `ctbu`(), `ctcov`(), `ctlcc`(), `ctmo`(), `ctmvn`(), `ctrec`(), `ctsmp`(), `cttd`(), `cttools`(), `iterec`()

Examples

```
set.seed(123)
# (3 x 7) base forecasts matrix (simulated), Z = X + Y and m = 4
base <- rbind(rnorm(7, rep(c(20, 10, 5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))),
              rnorm(7, rep(c(10, 5, 2.5), c(1, 2, 4))))
# (3 x 70) in-sample residuals matrix (simulated)
res <- rbind(rnorm(70), rnorm(70), rnorm(70))

A <- t(c(1,1)) # Aggregation matrix for Z = X + Y
m <- 4 # from quarterly to annual temporal aggregation

rtcs <- tcsrec(base = base,
               cslist = list(agg_mat = A, comb = "shr"),
               telist = list(agg_order = m, comb = "wlsv"),
               res = res)

rcst <- tcsrec(base = base,
               cslist = list(agg_mat = A, comb = "shr"),
               telist = list(agg_order = m, comb = "wlsv"),
               res = res)
```

`teboot` *Temporal joint block bootstrap*

Description

Joint block bootstrap for generating probabilistic base forecasts that take into account the correlation between different temporal aggregation orders (Girolimetto et al. 2023).

Usage

```
teboot(model_list, boot_size, agg_order, block_size = 1, seed = NULL,
       xreg = NULL, ...)
```

Arguments

`model_list`	A list of all the p base forecasts models ordered from the lowest frequency (most temporally aggregated) to the highest frequency. A `simulate()` function for each model has to be available and implemented according to the package **forecast**, with the following mandatory parameters: *object*, *innov*, *future*, and *nsim*.
`boot_size`	The number of bootstrap replicates.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`block_size`	Block size of the bootstrap, which is typically equivalent to the forecast horizon for the most temporally aggregated series.
`seed`	An integer seed.
`xreg`	A (block_size$(k^* + m) \times N_{xreg}$) optional numeric matrix containing the new values of `xreg` to be used for forecasting ordered from the lowest frequency to the highest frequency. It can contains `NA`s.
`...`	Additional arguments for the `simulate()` function.

Value

A list with two elements: the seed used to sample the errors and a (boot_size $\times (k^* + m)$block_size) matrix.

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Bootstrap samples: `csboot()`, `ctboot()`

Temporal framework: `tebu()`, `tecov()`, `telcc()`, `temo()`, `temvn()`, `terec()`, `tesmp()`, `tetd()`, `tetools()`

tebu — *Temporal bottom-up reconciliation*

Description

Temporal bottom-up reconciled forecasts at any temporal aggregation level are computed by appropriate aggregation of the high-frequency base forecasts, $\widehat{\mathbf{x}}^{[1]}$:

$$\widetilde{\mathbf{x}} = \mathbf{S}_{te}\widehat{\mathbf{x}}^{[1]},$$

where $\mathbf{S}_{te}$ is the temporal structural matrix.

Usage

```
tebu(base, agg_order, tew = "sum", sntz = FALSE, round = FALSE)
```

Arguments

`base`	A $(hm \times 1)$ numeric vector containing the high-frequency base forecasts; m is the max. temporal aggregation order, and h is the forecast horizon for the lowest frequency time series.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`tew`	A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
`sntz`	Logical. If `TRUE`, the negative base forecasts are set to zero (Di Fonzo and Girolimetto, 2023) before applying bottom-up. *Default* is `FALSE`.
`round`	Logical. If `TRUE`, base forecasts are rounded before applying the bottom-up reconciliation. *Default* is `FALSE`.

Value

A $(h(k^* + m) \times 1)$ numeric vector of temporal reconciled forecasts.

References

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

See Also

Bottom-up reconciliation: `csbu()`, `ctbu()`

Temporal framework: `teboot()`, `tecov()`, `telcc()`, `temo()`, `temvn()`, `terec()`, `tesmp()`, `tetd()`, `tetools()`

Examples

```
set.seed(123)
# (4 x 1) high frequency base forecasts vector (simulated),
# agg_order = 4 (annual-quarterly)
hfts <- rnorm(4, 5)

reco <- tebu(base = hfts, agg_order = 4)

# Non negative reconciliation
hfts[4] <- -hfts[4] # Making negative one of the quarterly base forecasts
nnreco <- tebu(base = hfts, agg_order = 4, sntz = TRUE)
```

---

tecov *Temporal covariance matrix approximation*

Description

This function provides an approximation of the temporal base forecasts errors covariance matrix using different reconciliation methods (see Di Fonzo and Girolimetto, 2023).

Usage

```
tecov(comb, agg_order = NULL, tew = "sum", res = NULL, mse = TRUE,
      shrink_fun = shrink_estim, ...)
```

Arguments

`comb` A string specifying the covariance approximation method.

- For ordinary least squares reconciliation:
  - "ols" (*default*) - identity error covariance.
- For weighted least squares reconciliation:
  - "str" - structural variances.
  - "wlsh" - hierarchy variances (uses res).
  - "wlsv" - series variances (uses res).
- For generalized least squares (uses res) reconciliation:
  - "acov" - series auto-covariance.
  - "strar1" - structural Markov covariance.
  - "sar1" - series Markov covariance.
  - "har1" - hierarchy Markov covariance.
  - "shr"/"sam" - shrunk/sample covariance.
- Others (no for reconciliation):
  - "bu" - bottom-up covariance.

`agg_order` Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.

tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
res	A $(N(k^* + m) \times 1)$ optional numeric vector containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency. This vector is used to compute come covariance matrices.
mse	If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected.
shrink_fun	Shrinkage function of the covariance matrix, shrink_estim (*default*)
...	Not used.

Value

A $((k^* + m) \times (k^* + m))$ symmetric matrix.

References

Di Fonzo, T. and Girolimetto, D. (2023a), Cross-temporal forecast reconciliation: Optimal combination method and heuristic alternatives, *International Journal of Forecasting*, 39, 1, 39-57. doi:10.1016/j.ijforecast.2021.08.004

See Also

Temporal framework: teboot(), tebu(), telcc(), temo(), temvn(), terec(), tesmp(), tetd(), tetools()

Examples

```
# (7 x 70) in-sample residuals matrix (simulated), agg_order = 4
res <- rnorm(70)

cov1 <- tecov("ols", agg_order = 4)                  # OLS
cov2 <- tecov("str", agg_order = 4)                  # STRC
cov3 <- tecov("wlsv", agg_order = 4, res = res)      # WLSv
cov4 <- tecov("wlsh", agg_order = 4, res = res)      # WLSh
cov5 <- tecov("acov", agg_order = 4, res = res)      # ACOV
cov6 <- tecov("strar1", agg_order = 4, res = res)    # STRAR1
cov7 <- tecov("har1", agg_order = 4, res = res)      # HAR1
cov8 <- tecov("sar1", agg_order = 4, res = res)      # SAR1
cov9 <- tecov("shr", agg_order = 4, res = res)       # SHR
cov10 <- tecov("sam", agg_order = 4, res = res)      # SAM

# Custom covariance matrix
tecov.ols2 <- function(comb, x) diag(x)
tecov(comb = "ols2", x = 7) # == tecov("ols", agg_order = 4)
```

telcc *Level conditional coherent reconciliation for temporal hierarchies*

Description

This function implements a forecast reconciliation procedure inspired by the original proposal by Hollyman et al. (2021) for temporal hierarchies. Level conditional coherent reconciled forecasts are conditional on (i.e., constrained by) the base forecasts of a specific upper level in the hierarchy (exogenous constraints). It also allows handling the linear constraints linking the variables endogenously (Di Fonzo and Girolimetto, 2022). The function can calculate Combined Conditional Coherent (CCC) forecasts as simple averages of Level-Conditional Coherent (LCC) and bottom-up reconciled forecasts, with either endogenous or exogenous constraints.

Usage

```
telcc(base, agg_order, tew = "sum", comb = "ols", res = NULL,
      approach = "proj", nn = NULL, settings = NULL, CCC = TRUE,
      const = "exogenous", hfts = NULL, ...)
```

Arguments

`base` A $(h(k^* + m) \times 1)$ numeric vector containing the base forecasts to be reconciled, ordered from lowest to highest frequency; m is the maximum aggregation order, k^* is the sum of a chosen subset of the $p - 1$ factors of m (excluding m itself) and h is the forecast horizon for the lowest frequency time series.

`agg_order` Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.

`tew` A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).

`comb` A string specifying the reconciliation method. For a complete list, see tecov.

`res` A $(N(k^* + m) \times 1)$ optional numeric vector containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency. This vector is used to compute come covariance matrices.

`approach` A string specifying the approach used to compute the reconciled forecasts. Options include:

- `"proj"` (*default*): Projection approach according to Byron (1978, 1979).
- `"strc"`: Structural approach as proposed by Hyndman et al. (2011).
- `"proj_osqp"`: Numerical solution using **osqp** for projection approach.
- `"strc_osqp"`: Numerical solution using **osqp** for structural approach.

`nn` A string specifying the algorithm to compute non-negative forecasts:

- `"osqp"`: quadratic programming optimization (**osqp** solver, Girolimetto 2025).
- `"bpv"`: block principal pivoting algorithm (Wickramasuriya et al., 2020).

- "nfca": negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
- "nnic": iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
- "sntz": heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).

settings
: A list of control parameters.
  - nn = "osqp" An object of class osqpSettings specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
  - nn = "bpv"
    - ptype = "fixed": permutation method: "random" or "fixed"
    - par = 10: the number of full exchange rules that may be attempted
    - tol = sqrt(.Machine$double.eps): the tolerance criteria
    - gtol = sqrt(.Machine$double.eps): the gradient tolerance criteria
    - itmax = 100: the maximum number of algorithm iterations
  - nn = "nfca" and nn = "nnic"
    - tol = sqrt(.Machine$double.eps): the tolerance criteria
    - itmax = 100: the maximum number of algorithm iterations
  - nn = "sntz"
    - type = "bu": the type of set-negative-to-zero heuristic: "bu" for bottom-up, "tdp" for top-down proportional, "tdsp" for top-down square proportional, "tdvw" for top-down variance weighted (the res param is used). See Girolimetto (2025) for details.
    - tol = sqrt(.Machine$double.eps): the tolerance identification of negative values

CCC
: A logical value indicating whether the Combined Conditional Coherent reconciled forecasts reconciliation should include bottom-up forecasts (TRUE, *default*), or not.

const
: A string specifying the reconciliation constraints:
  - "exogenous" (*default*): Fixes the top level of each sub-hierarchy.
  - "endogenous": Coherently revises both the top and bottom levels.

hfts
: A $(mh \times 1)$ numeric vector containing high frequency base forecasts defined by the user. This parameter can be omitted if only base forecasts in base are used (see Di Fonzo and Girolimetto, 2024).

...
: Arguments passed on to tecov
  - mse If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected.
  - shrink_fun Shrinkage function of the covariance matrix, shrink_estim (*default*)

Value

A $(h(k^* + m) \times 1)$ numeric vector of temporal reconciled forecasts.

References

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2024), Forecast combination-based forecast reconciliation: Insights and extensions, *International Journal of Forecasting*, 40(2), 490–514. doi:10.1016/j.ijforecast.2022.07.001

Di Fonzo, T. and Girolimetto, D. (2023b) Spatio-temporal reconciliation of solar forecasts. *Solar Energy* 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Hollyman, R., Petropoulos, F. and Tipping, M.E. (2021), Understanding forecast reconciliation. *European Journal of Operational Research*, 294, 149–160. doi:10.1016/j.ejor.2021.01.017

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

See Also

Level conditional coherent reconciliation: `cslcc()`, `ctlcc()`

Temporal framework: `teboot()`, `tebu()`, `tecov()`, `temo()`, `temvn()`, `terec()`, `tesmp()`, `tetd()`, `tetools()`

Examples

```
set.seed(123)
# (7 x 1) base forecasts vector (simulated), agg_order = 4
base <- rnorm(7, rep(c(20, 10, 5), c(1, 2, 4)))
# (70 x 1) in-sample residuals vector (simulated)
res <- rnorm(70)
# (4 x 1) Naive high frequency base forecasts vector: all forecasts are set equal to 2.5
naive <- rep(2.5, 4)

## EXOGENOUS CONSTRAINTS
# Level Conditional Coherent (LCC) reconciled forecasts
exo_LC <- telcc(base = base, agg_order = 4, comb = "wlsh", hfts = naive,
                res = res, CCC = FALSE)

# Combined Conditional Coherent (CCC) reconciled forecasts
exo_CCC <- telcc(base = base, agg_order = 4, comb = "wlsh", hfts = naive,
                 res = res, CCC = TRUE)
```

```
# Results detailed by level:
info_exo <- recoinfo(exo_CCC, verbose = FALSE)
# info_exo$lcc

## ENDOGENOUS CONSTRAINTS
# Level Conditional Coherent (LCC) reconciled forecasts
endo_LC <- telcc(base = base, agg_order = 4, comb = "wlsh", res = res,
                 CCC = FALSE, const = "endogenous")

# Combined Conditional Coherent (CCC) reconciled forecasts
endo_CCC <- telcc(base = base, agg_order = 4, comb = "wlsh", res = res,
                  CCC = TRUE, const = "endogenous")

# Results detailed by level:
info_endo <- recoinfo(endo_CCC, verbose = FALSE)
# info_endo$lcc
```

---

temo *Temporal middle-out reconciliation*

Description

The middle-out forecast reconciliation for temporal hierarchies combines top-down (tetd) and bottom-up (tebu) methods. Given the base forecasts of an intermediate temporal aggregation order k, it performs

- a top-down approach for the aggregation orders $< k$;
- a bottom-up approach for the aggregation orders $> k$.

Usage

```
temo(base, agg_order, weights, order = max(agg_order), tew = "sum",
     normalize = TRUE)
```

Arguments

`base`	A $(hk \times 1)$ numeric vector containing the temporal aggregated base forecasts of order k; k is an aggregation order (a factor of m, and $1 < k < m$), m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`weights`	A $(hm \times 1)$ numeric vector containing the proportions for the high-frequency time series; m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`order`	The intermediate fixed aggregation order k.

`tew`	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
`normalize`	If TRUE (*default*), the weights will sum to 1.

Value

A $(h(k^* + m) \times 1)$ numeric vector of temporal reconciled forecasts.

See Also

Middle-out reconciliation: csmo(), ctmo()

Temporal framework: teboot(), tebu(), tecov(), telcc(), temvn(), terec(), tesmp(), tetd(), tetools()

Examples

```
set.seed(123)
# (6 x 1) base forecasts vector (simulated), forecast horizon = 3
# and intermediate aggregation order k = 2 (max agg order = 4)
basek2 <- rnorm(3*2, 5)
# Same weights for different forecast horizons
fix_weights <- runif(4)
reco <- temo(base = basek2, order = 2, agg_order = 4, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- runif(4*3)
recoh <- temo(base = basek2, order = 2, agg_order = 4, weights = h_weights)
```

`temvn` *Temporal Gaussian probabilistic reconciliation*

Description

This function performs temporal probabilistic forecast reconciliation assuming a multivariate normal base forecast distribution (Girolimetto et al., 2024) for a single time series using temporal hierarchies (Athanasopoulos et al., 2017).

Usage

```
temvn(base, agg_order, tew = "sum", comb = "ols", res = NULL,
      approach = "proj", comb_base = comb, reduce_form = FALSE, ...)
```

Arguments

- `base` — A $(h(k^* + m) \times 1)$ numeric vector containing the base forecasts to be reconciled, ordered from lowest to highest frequency; m is the maximum aggregation order, k^* is the sum of a chosen subset of the $p - 1$ factors of m (excluding m itself) and h is the forecast horizon for the lowest frequency time series.
- `agg_order` — Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
- `tew` — A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
- `comb` — A string specifying the reconciliation method. For a complete list, see tecov.
- `res` — A $(N(k^* + m) \times 1)$ optional numeric vector containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency. This vector is used to compute come covariance matrices.
- `approach` — A string specifying the approach used to compute the reconciled forecasts. Options include:
  - `"proj"` (*default*): Projection approach according to Byron (1978, 1979).
  - `"strc"`: Structural approach as proposed by Hyndman et al. (2011).
  - `"proj_osqp"`: Numerical solution using **osqp** for projection approach.
  - `"strc_osqp"`: Numerical solution using **osqp** for structural approach.
- `comb_base` — A string specifying the base covariance matrix approach. For a complete list, see tecov. Default is the equal to `comb`.
- `reduce_form` — A logical parameter indicating whether the function should return the full distribution (`FALSE`, *default*) or only the distribution corresponding to the high-frequency time series (`TRUE`).
- `...` — Arguments passed on to `tecov`
  - `mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.
  - `shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*)

Value

A distributional::dist_multivariate_normal object.

References

Athanasopoulos, G., Hyndman, R.J., Kourentzes, N. and Petropoulos, F. (2017), Forecasting with Temporal Hierarchies, *European Journal of Operational Research*, 262, 1, 60-74. doi:10.1016/j.ejor.2017.02.046

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

See Also

Probabilistic reconciliation: `csmvn()`, `cssmp()`, `ctmvn()`, `ctsmp()`, `tesmp()`

Temporal framework: `teboot()`, `tebu()`, `tecov()`, `telcc()`, `temo()`, `terec()`, `tesmp()`, `tetd()`, `tetools()`

Examples

```
set.seed(123)
# (7 x 1) base forecasts vector (simulated), m = 4
base <- rnorm(7*2, rep(c(20, 10, 5), 2*c(1, 2, 4)))
# (70 x 1) in-sample residuals vector (simulated)
res <- rnorm(70)

m <- 4 # from quarterly to annual temporal aggregation
reco_dist <- terec(base = base, agg_order = m, comb = "wlsv", res = res)
```

`teprojmat` *Projection matrix for optimal combination temporal reconciliation*

Description

This function computes the projection or the mapping matrix $\mathbf{M}$ and $\mathbf{G}$, respectively, such that $\widetilde{\mathbf{y}} = \mathbf{M}\widehat{\mathbf{y}} = \mathbf{S}_{te}\mathbf{G}\widehat{\mathbf{y}}$, where $\widetilde{\mathbf{y}}$ is the vector of the reconciled forecasts, $\widehat{\mathbf{y}}$ is the vector of the base forecasts, $\mathbf{S}_{te}$ is the temporal structural matrix, and $\mathbf{M} = \mathbf{S}_{te}\mathbf{G}$. For further information regarding on the structure of these matrices, refer to Girolimetto et al. (2023).

Usage

```
teprojmat(agg_order, comb = "ols", res = NULL, mat = "M", tew = "sum", ...)
```

Arguments

`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`comb`	A string specifying the reconciliation method. For a complete list, see tecov.
`res`	A $(N(k^* + m) \times 1)$ optional numeric vector containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency. This vector is used to compute come covariance matrices.

mat	A string specifying which matrix to return: "M" (*default*) for $\mathbf{M}$ and "G" for $\mathbf{G}$.
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
...	Arguments passed on to tecov mse If TRUE (*default*) the errors used to compute the covariance matrix are not mean-corrected. shrink_fun Shrinkage function of the covariance matrix, shrink_estim (*default*)

Value

The projection matrix $\mathbf{M}$ (mat = "M") or the mapping matrix $\mathbf{G}$ (mat = "G").

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), tetools(), unbalance_hierarchy()

Examples

```
# Temporal framework (annual-quarterly)
Mte <- teprojmat(agg_order = 4, comb = "ols")
Gte <- teprojmat(agg_order = 4, comb = "ols", mat = "G")
```

---

terec *Optimal combination temporal reconciliation*

Description

This function performs forecast reconciliation for a single time series using temporal hierarchies (Athanasopoulos et al., 2017, Nystrup et al., 2020). The reconciled forecasts can be computed using either a projection approach (Byron, 1978, 1979) or the equivalent structural approach by Hyndman et al. (2011). Non-negative (Di Fonzo and Girolimetto, 2023) and immutable reconciled forecasts can be considered.

Usage

```
terec(base, agg_order, tew = "sum", comb = "ols", res = NULL,
      approach = "proj", nn = NULL, settings = NULL, bounds = NULL,
      immutable = NULL, ...)
```

Arguments

- `base` — A $(h(k^* + m) \times 1)$ numeric vector containing the base forecasts to be reconciled, ordered from lowest to highest frequency; m is the maximum aggregation order, k^* is the sum of a chosen subset of the $p - 1$ factors of m (excluding m itself) and h is the forecast horizon for the lowest frequency time series.
- `agg_order` — Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
- `tew` — A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
- `comb` — A string specifying the reconciliation method. For a complete list, see tecov.
- `res` — A $(N(k^* + m) \times 1)$ optional numeric vector containing the in-sample residuals or validation errors ordered from the lowest frequency to the highest frequency. This vector is used to compute come covariance matrices.
- `approach` — A string specifying the approach used to compute the reconciled forecasts. Options include:
  - `"proj"` (*default*): Projection approach according to Byron (1978, 1979).
  - `"strc"`: Structural approach as proposed by Hyndman et al. (2011).
  - `"proj_osqp"`: Numerical solution using **osqp** for projection approach.
  - `"strc_osqp"`: Numerical solution using **osqp** for structural approach.
- `nn` — A string specifying the algorithm to compute non-negative forecasts:
  - `"osqp"`: quadratic programming optimization (**osqp** solver, Girolimetto 2025).
  - `"bpv"`: block principal pivoting algorithm (Wickramasuriya et al., 2020).
  - `"nfca"`: negative forecasts correction algorithm (Kourentzes and Athanasopoulos, 2021; Girolimetto 2025).
  - `"nnic"`: iterative non-negative reconciliation with immutable constraints (Girolimetto 2025).
  - `"sntz"`: heuristic "set-negative-to-zero" (Di Fonzo and Girolimetto, 2023; Girolimetto 2025).
- `settings` — A list of control parameters.
  - `nn = "osqp"` An object of class `osqpSettings` specifying settings for the **osqp** solver. For details, refer to the **osqp** documentation (Stellato et al., 2020)
  - `nn = "bpv"`
    - `ptype = "fixed"`: permutation method: `"random"` or `"fixed"`
    - `par = 10`: the number of full exchange rules that may be attempted
    - `tol = sqrt(.Machine$double.eps)`: the tolerance criteria

- `gtol = sqrt(.Machine$double.eps)`: the gradient tolerance criteria
- `itmax = 100`: the maximum number of algorithm iterations

• `nn = "nfca"` and `nn = "nnic"`

- `tol = sqrt(.Machine$double.eps)`: the tolerance criteria
- `itmax = 100`: the maximum number of algorithm iterations

• `nn = "sntz"`

- `type = "bu"`: the type of set-negative-to-zero heuristic: "bu" for bottom-up, "tdp" for top-down proportional, "tdsp" for top-down square proportional, "tdvw" for top-down variance weighted (the `res` param is used). See Girolimetto (2025) for details.
- `tol = sqrt(.Machine$double.eps)`: the tolerance identification of negative values

`bounds` A matrix (see set_bounds) with 4 columns ($k, j, lower, upper$), such that

- Column 1 represents the temporal aggregation order ($k = m, \dots, 1$).
- Column 2 represents the temporal forecast horizon ($j = 1, \dots, m/k$).
- Columns 3 and 4 indicates the *lower* and *upper* bounds, respectively.

`immutable` A matrix with 2 columns (k, j), such that

- Column 1 represents the temporal aggregation order ($k = m, \dots, 1$).
- Column 2 represents the temporal forecast horizon ($j = 1, \dots, m/k$).

For example, when working with a quarterly time series:

- `t(c(4, 1))` - Fix the one step ahead annual forecast.
- `t(c(1, 2))` - Fix the two step ahead quarterly forecast.

`...` Arguments passed on to `tecov`

`mse` If `TRUE` (*default*) the errors used to compute the covariance matrix are not mean-corrected.

`shrink_fun` Shrinkage function of the covariance matrix, shrink_estim (*default*)

Value

A $(h(k^* + m) \times 1)$ numeric vector of temporal reconciled forecasts.

References

Athanasopoulos, G., Hyndman, R.J., Kourentzes, N. and Petropoulos, F. (2017), Forecasting with Temporal Hierarchies, *European Journal of Operational Research*, 262, 1, 60-74. doi:10.1016/j.ejor.2017.02.046

Byron, R.P. (1978), The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 141, 3, 359-367. doi:10.2307/2344807

Byron, R.P. (1979), Corrigenda: The estimation of large social account matrices, *Journal of the Royal Statistical Society, Series A*, 142(3), 405. doi:10.2307/2982515

Di Fonzo, T. and Girolimetto, D. (2023), Spatio-temporal reconciliation of solar forecasts, *Solar Energy*, 251, 13–29. doi:10.1016/j.solener.2023.01.003

Girolimetto, D. (2025), Non-negative forecast reconciliation: Optimal methods and operational solutions. *Forecasting*, 7(4), 64; doi:10.3390/forecast7040064

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G. and Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589. doi:10.1016/j.csda.2011.03.006

Kourentzes, N. and Athanasopoulos, G. (2021) Elucidate structure in intermittent demand series. *European Journal of Operational Research*, 288, 141-152. doi:10.1016/j.ejor.2020.05.046

Nystrup, P., Lindström, E., Pinson, P. and Madsen, H. (2020), Temporal hierarchies with autocorrelation for load forecasting, *European Journal of Operational Research*, 280, 1, 876-888. doi:10.1016/j.ejor.2019.07.061

Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S. (2020), OSQP: An Operator Splitting solver for Quadratic Programs, *Mathematical Programming Computation*, 12, 4, 637-672. doi:10.1007/s12532020001792

Wickramasuriya, S. L., Turlach, B. A., and Hyndman, R. J. (2020). Optimal non-negative forecast reconciliation. *Statistics and Computing*, 30(5), 1167–1182. doi:10.1007/s11222020099300

See Also

Regression-based reconciliation: `csrec()`, `ctrec()`

Temporal framework: `teboot()`, `tebu()`, `tecov()`, `telcc()`, `temo()`, `temvn()`, `tesmp()`, `tetd()`, `tetools()`

Examples

```
set.seed(123)
# (7 x 1) base forecasts vector (simulated), m = 4
base <- rnorm(7, rep(c(20, 10, 5), c(1, 2, 4)))
# (70 x 1) in-sample residuals vector (simulated)
res <- rnorm(70)

m <- 4 # from quarterly to annual temporal aggregation
reco <- terec(base = base, agg_order = m, comb = "wlsv", res = res)

# Immutable reconciled forecast
# E.g. fix all the quarterly forecasts
imm_q <- expand.grid(k = 1, j = 1:4)
immreco <- terec(base = base, agg_order = m, comb = "wlsv",
                 res = res, immutable = imm_q)

# Non negative reconciliation
base[7] <- -base[7] # Making negative one of the quarterly base forecasts
nnreco <- terec(base = base, agg_order = m, comb = "wlsv",
                res = res, nn = "osqp")
recoinfo(nnreco, verbose = FALSE)$info
```

tesmp — *Temporal probabilistic reconciliation (sample approach)*

Description

This function performs temporal probabilistic forecast reconciliation using a sample-based approach (Girolimetto et al., 2024) for a single time series using temporal hierarchies (Athanasopoulos et al., 2017). Given a $(L \times h(k^* + m))$ matrix of simulated base forecast draws, `tesmp()` applies a chosen **FoReco** temporal reconciliation independently to each draw, producing a coherent sample distribution of reconciled forecasts across the temporal hierarchy. Typical choices for the reconciliation include optimal combination (terec) as well as top-down (tetd), middle-out (temo), bottom-up (tebu), and level-conditional (telcc) approaches.

Usage

```
tesmp(sample, agg_order, fun = terec, ...)
```

Arguments

`sample`	A $(L \times h(k^* + m))$ numeric matrix containing the base forecasts samples to be reconciled; m is the max aggregation order, k^* is the sum of (a subset of) $(p-1)$ factors of m, excluding m, h is the forecast horizon for the lowest frequency time series, and L is the sample size.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`fun`	A string specifying the reconciliation function to be used, as implemented in **FoReco**.
`...`	Arguments passed on to `fun`

Value

A distributional::dist_sample object.

References

Athanasopoulos, G., Hyndman, R.J., Kourentzes, N. and Petropoulos, F. (2017), Forecasting with Temporal Hierarchies, *European Journal of Operational Research*, 262, 1, 60-74. doi:10.1016/j.ejor.2017.02.046

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

See Also

Probabilistic reconciliation: `csmvn()`, `cssmp()`, `ctmvn()`, `ctsmp()`, `temvn()`

Temporal framework: `teboot()`, `tebu()`, `tecov()`, `telcc()`, `temo()`, `temvn()`, `terec()`, `tetd()`, `tetools()`

Examples

```
set.seed(123)
m <- 4 # from quarterly to annual temporal aggregation

# (100 x 14) base forecasts sample matrix (simulated), m = 4, h = 2
sample <- t(sapply(1:100, function(x) {
  rnorm(14, rep(c(20, 10, 5), 2 * c(1, 2, 4)))
}))
# (70 x 1) in-sample residuals vector (simulated)
res <- rnorm(70)

# Top-down probabilistic reconciliation
reco_dist_td <- tesmp(sample[,c(1:2), drop = FALSE], agg_order = m,
                      fun = tetd, weights = c(0.2, 0.5, 0.3, 0.3))

# Middle-out probabilistic reconciliation
reco_dist_mo <- tesmp(sample[,c(3:6), drop = FALSE], agg_order = m,
                      fun = temo, weights = c(0.2, 0.5, 0.3, 0.3), order = 2)

# Bottom-up probabilistic reconciliation
reco_dist_bu <- tesmp(sample[,-c(1:6)], agg_order = m, fun = tebu)

# Level conditional coherent probabilistic reconciliation
reco_dist_lcc <- tesmp(sample, agg_order = m, fun = telcc)

# Optimal cross-sectional probabilistic reconciliation
reco_dist_opt <- tesmp(sample, agg_order = m, res = res, comb = "wlsv")
```

`tetd` *Temporal top-down reconciliation*

Description

Top-down forecast reconciliation for a univariate time series, where the forecast of the most aggregated temporal level is disaggregated according to a proportional scheme (weights). Besides fulfilling any aggregation constraint, the top-down reconciled forecasts should respect two main properties:

- the top-level value remains unchanged;
- all the bottom time series reconciled forecasts are non-negative.

Usage

```
tetd(base, agg_order, weights, tew = "sum", normalize = TRUE)
```

Arguments

`base`	A $(hm \times 1)$ numeric vector containing the temporal aggregated base forecasts of order m; m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`agg_order`	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
`weights`	A $(hm \times 1)$ numeric vector containing the proportions for the high-frequency time series; m is the max aggregation order, and h is the forecast horizon for the lowest frequency time series.
`tew`	A string specifying the type of temporal aggregation. Options include: `"sum"` (simple summation, *default*), `"avg"` (average), `"first"` (first value of the period), and `"last"` (last value of the period).
`normalize`	If `TRUE` (*default*), the `weights` will sum to 1.

Value

A $(h(k^* + m) \times 1)$ numeric vector of temporal reconciled forecasts.

See Also

Top-down reconciliation: `cstd()`, `cttd()`

Temporal framework: `teboot()`, `tebu()`, `tecov()`, `telcc()`, `temo()`, `temvn()`, `terec()`, `tesmp()`, `tetools()`

Examples

```
set.seed(123)
# (2 x 1) top base forecasts vector (simulated), forecast horizon = 2
topf <- rnorm(2, 10)
# Same weights for different forecast horizons
fix_weights <- runif(4)
reco <- tetd(base = topf, agg_order = 4, weights = fix_weights)

# Different weights for different forecast horizons
h_weights <- runif(4*2)
recoh <- tetd(base = topf, agg_order = 4, weights = h_weights)
```

`tetools` *Temporal reconciliation tools*

Description

Some useful tools for forecast reconciliation through temporal hierarchies.

Usage

```
tetools(agg_order, fh = 1, tew = "sum", sparse = TRUE)
```

Arguments

agg_order	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation, m), or a vector representing a subset of p factors of m.
fh	Forecast horizon for the lowest frequency (most temporally aggregated) time series (*default* is 1).
tew	A string specifying the type of temporal aggregation. Options include: "sum" (simple summation, *default*), "avg" (average), "first" (first value of the period), and "last" (last value of the period).
sparse	Option to return sparse matrices (*default* is TRUE).

Value

A list with five elements:

dim	A vector containing information about the maximum aggregation order (m), the number of factor (p), the partial (ks) and total sum (kt) of factors.
set	The vector of the temporal aggregation orders (in decreasing order).
agg_mat	The temporal linear combination or aggregation matrix.
strc_mat	The temporal structural matrix.
cons_mat	The temporal zero constraints matrix.

See Also

Temporal framework: teboot(), tebu(), tecov(), telcc(), temo(), temvn(), terec(), tesmp(), tetd()

Utilities: FoReco2matrix(), aggts(), as_ctmatrix(), as_tevector(), balance_hierarchy(), commat(), csprojmat(), cstools(), ctprojmat(), cttools(), df2aggmat(), lcmat(), recoinfo(), res2matrix(), set_bounds(), shrink_estim(), shrink_oasd(), teprojmat(), unbalance_hierarchy()

Examples

```
# Temporal framework (quarterly data)
obj <- tetools(agg_order = 4, sparse = FALSE)
```

unbalance_hierarchy *Aggregation matrix of a balanced hierarchy in (possibly) unbalanced form*

Description

A hierarchy with L upper levels is said to be balanced if each variable at level l has at least one child at level $l + 1$. When this doesn't hold, the hierarchy is unbalanced. This function transforms an aggregation matrix of a balanced hierarchy into an aggregation matrix of an unbalanced one, by removing possible duplicated series.

Usage

```
unbalance_hierarchy(agg_mat, more_info = FALSE, sparse = TRUE)
```

Arguments

- `agg_mat` A $(n_a \times n_b)$ numeric matrix representing the cross-sectional aggregation matrix. It maps the n_b bottom-level (free) variables into the n_a upper (constrained) variables.
- `more_info` If `TRUE`, it returns only the aggregation matrix of the unbalanced hierarchy. *Default* is `FALSE`.
- `sparse` Option to return sparse matrices (*default* is `TRUE`).

Value

A list containing four elements (`more_info` = `TRUE`):

- `ubm` The aggregation matrix of the unbalanced hierarchy.
- `agg_mat` The input matrix.
- `idrm` The identification number of the duplicated variables (row numbers of the aggregation matrix `agg_mat`).
- `id` The identification number of each variable in the balanced hierarchy. It may contains duplicated values.

See Also

Utilities: `FoReco2matrix()`, `aggts()`, `as_ctmatrix()`, `as_tevector()`, `balance_hierarchy()`, `commat()`, `csprojmat()`, `cstools()`, `ctprojmat()`, `cttools()`, `df2aggmat()`, `lcmat()`, `recoinfo()`, `res2matrix()`, `set_bounds()`, `shrink_estim()`, `shrink_oasd()`, `teprojmat()`, `tetools()`

Examples

```
#     Balanced     ->     Unbalanced
#        T                    T
#    |-------|            |-------|
#    A       B            A       |
#  |---|     |          |---|     |
# AA   AB    BA         AA   AB    BA
A <- matrix(c(1, 1, 1,
              1, 1, 0,
              0, 0, 1), 3, byrow = TRUE)
obj <- unbalance_hierarchy(agg_mat = A)
obj
```

`vndata` *Australian Tourism Demand dataset*

Description

The Australian Tourism Demand dataset (Wickramasuriya et al. 2019) measures the number of nights Australians spent away from home. It includes 228 monthly observations of Visitor Nights (VNs) from January 1998 to December 2016, and has a cross-sectional grouped structure based on a geographic hierarchy crossed by purpose of travel. The geographic hierarchy comprises 7 states, 27 zones, and 76 regions, for a total of 111 nested geographic divisions. Six of these zones are each formed by a single region, resulting in 105 unique nodes in the hierarchy. The purpose of travel comprises four categories: holiday, visiting friends and relatives, business, and other. To avoid redundancies (Girolimetto et al. 2023), 24 nodes (6 zones are formed by a single region) are not considered, resulting in an unbalanced hierarchy of 525 (304 bottom and 221 upper time series) unique nodes instead of the theoretical 555 with duplicated nodes.

Usage

```
# 525 time series of the Australian Tourism Demand dataset
vndata

# aggregation matrix
vnaggmat
```

Format

`vndata` is a (228×525) `ts` object, corresponding to 525 time series of the Australian Tourism Demand dataset (1998:01-2016:12).

`vnaggmat` is the (221×304) aggregation matrix.

Source

https://robjhyndman.com/publications/mint/

References

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R.J. (2024), Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues. *International Journal of Forecasting*, 40, 3, 1134-1151. doi:10.1016/j.ijforecast.2023.10.003

Wickramasuriya, S.L., Athanasopoulos, G. and Hyndman, R.J. (2019), Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, *Journal of the American Statistical Association*, 114, 526, 804-819. doi:10.1080/01621459.2018.1448825

Index