Package 'MAPA'

November 7, 2026

Type Package

Title Multiple Aggregation Prediction Algorithm

Version 2.1.0

Description Functions and wrappers for using the Multiple Aggregation Prediction Algorithm (MAPA) for time series forecasting. MAPA models and forecasts time series at multiple temporal aggregation levels, thus strengthening and attenuating the various time series components for better holistic estimation of its structure. For details see Kourentzes et al. (2014) <doi:10.1016/j.ijforecast.2013.09.006>.

LazyData yes

License GPL (>= 2)

Depends R (>= 4.1.0)

Imports forecast (>= 5.3), parallel, RColorBrewer, smooth (>= 4.0.0), grDevices, graphics, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

URL https://kourentzes.com/forecasting/2014/04/19/multiple-aggregation-prediction-algorithm-mapa/

BugReports https://github.com/trnnick/mapa/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Nikolaos Kourentzes [aut, cre], Fotios Petropoulos [aut]

Maintainer Nikolaos Kourentzes <nikolaos@kourentzes.com>

Repository CRAN

Date/Publication 2026-11-07 18:10:20 UTC

Contents

admissions . . . 2
mapa . . . 3
mapacalc . . . 6
mapaest . . . 8
mapafit.old . . . 12
mapafor . . . 13
mapasimple . . . 16
plot.mapa.calc . . . 19
plot.mapa.fit . . . 20
plot.mapa.for . . . 21
plotmapa . . . 23
print.mapa.fit . . . 24
tsaggr . . . 25

Index **27**

`admissions` *Total Non-elective G&A Admissions (FFCEs)*

Description

UK NHS Total Non-elective G&A Admissions (FFCEs). April 2008 – June 2013.

Usage

```
admissions
```

Format

Time series data

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

Source

UK Department of Health: Monthly Activity Return

Examples

```
forecast::tsdisplay(admissions)
```

mapa *Multiple Aggregation Prediction Algorithm (Wrapper)*

Description

Wrapper to estimate and produce MAPA in- and out-of-sample forecasts. Uses mapaest and mapafor.

Usage

```
mapa(
  y,
  ppy = NULL,
  fh = NULL,
  ifh = 1,
  minimumAL = 1,
  maximumAL = ppy,
  comb = c("w.mean", "w.median", "mean", "median", "wght"),
  paral = c(0, 1, 2),
  display = c(0, 1),
  outplot = c(0, 1),
  hybrid = c(TRUE, FALSE),
  model = "ZZZ",
  type = c("ets", "es"),
  conf.lvl = NULL,
  xreg = NULL,
  pr.comp = 0,
  cl = NULL,
  ...
)
```

Arguments

y	In sample observations of a time series (vector). If y == "paper" then it prints paper reference.
ppy	Periods in a season of the time series at the sampled frequency. If insample is a ts object then this is taken from its frequency, unless overriden.
fh	Forecast horizon. Default = ppy. If fh = 0 then only in-sample forecasts are produced.
ifh	In-sample forecast horizon. Default = 1.
minimumAL	Lowest aggregation level to use. Default = 1.
maximumAL	Highest aggregation level to use. Default = ppy, maximumAL>1.
comb	Combination operator. This can be: "mean"; "median"; "wght" - where each aggregation level is weighted inversly to aggregation; "w.mean" - level and trend components are averaged, but seasonal and xreg follow the wght combination;

	"w.median" - as w.mean, but with median. It is suggested that for data with high sampling frequency to use one of the "w.mean" and "w.median".
paral	Use parallel processing. 0 = no; 1 = yes (requires initialised cluster); 2 = yes and initialise cluster. Default is 0. Parallel processing is used for both the estimation and the in-sample forecasts.
display	Display calculation progress in console. 0 = no; 1 = yes. Default is 0.
outplot	Provide output plot. 0 = no; 1 = yes. Default is 0.
hybrid	Provide hybrid forecasts, as in Kourentzes et al. paper. If minimumAL > 1 then the minimumAL ETS forecasts are used. Default is TRUE.
model	Allow only that type of ETS at each aggregation level. This follows similar coding to the ets function. The first letter refers to the error type ("A", "M", or "Z"); the second letter refers to the trend type ("N","A","Ad","M","Md", "Z", "X", or "Y"); and the third letter refers to the season type ("N","A","M", "Z", "X", or "Y"). The letters mean: "N"=none, "A"=additive, "M"=multiplicative, "Z"=automatically selected, "X"=automatically select between none and additive, and "Y"=automatically select between none and multiplicative. A "d" for trend implies damped. "X" and "Y" supported only by type=="es". If used with type=="ets" a warning will be given and they will default to "Z". By default model="ZZZ". If due to sample limitation ETS cannot be calculated at an aggregation level for the selected model, then no estimation is done for that specific level.
type	What type of exponential smoothing implementation to use. "es" = use from the smooth package; "ets" = use from the forecast package. Default is "es"
conf.lvl	Vector of confidence level for prediction intervals. Values must be (0,1). If conf.lvl == NULL then no intervals are calculated. For example to get the intervals for 80% and 95% use conf.lvl=c(0.8,0.95).
xreg	Vector or matrix of exogenous variables to be included in the MAPA. If matrix then rows are observations and columns are variables. Must be at least as long as in-sample. Additional observations are unused. Note that including xreg will force type="es".
pr.comp	MAPAx can use principal component analysis to preprocess xreg. When comp is -1 then the number of retained components is chosen automatically. When comp=0 then no pre-processing is performed and the original xreg is used. Any other value represents the number of principal components retained.
cl	Cluster to use when paral=1, created with `parallel::makeCluster`. If NULL, a cluster named `cl` in the global environment is used, for compatibility with earlier versions.
...	Pass additional arguments to es or ets.

Details

This function calls *mapaest* and *mapafor* internally.

Use `type="es"` for the `es` function of the **smooth** package, instead of the default `ets` function of the **forecast** package. See `mapaest` for details on the differences between the two and on the use of xreg.

Value

infor	In-sample forecasts.
outfor	Out-of-sample forecasts.
PI	Prediction intervals for given confidence levels.
MSE	In-sample MSE error.
MAE	In-sample MAE error.

The output is of class mapa.for, which also stores the time series and the settings used. See mapafor and plot.mapa.for.

Note

The calculation of the prediction intervals is based on the empirical multiple step ahead MSE. To speed up calculations set conf.lvl=NULL. If very long forecast horizons are requested then once no more t+h MSE can be calculated the following approximation is used: $\sqrt{MSE_{t+1}} * \sqrt{h}$ for the error.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

mapaest, mapafor, mapasimple.

Examples

```
# MAPA with ets from the forecast package (default), forecasting 18 steps
# ahead, with in-sample forecasts up to 3 steps ahead and prediction intervals
out <- mapa(admissions, fh=18, ifh=3, conf.lvl=c(0.8,0.95))
print(out)
summary(out)
plot(out)

# MAPAx with es from the smooth package. xreg must contain the in-sample
# observations and the fh future values
set.seed(1)
fh <- 12
x <- matrix(rnorm((length(admissions)+fh)*2), ncol=2)
out <- mapa(admissions, fh=fh, type="es", xreg=x)
```

```
# es with a different combination operator and without hybrid forecasts
out <- mapa(admissions, type="es", comb="median", hybrid=FALSE)

# Non-seasonal models only, at aggregation levels 1 to 6
out <- mapa(admissions, model="ZZN", maximumAL=6)

# MAPAx with xreg preprocessed by principal components, selecting the number
# of components automatically, and plotting the output
out <- mapa(admissions, fh=fh, type="es", xreg=x, pr.comp=-1, conf.lvl=0.9, outplot=1)

# Only in-sample forecasts
out <- mapa(admissions, fh=0, ifh=6)
```

`mapacalc` *MAPA Internal Calculation of Forecasts*

Description

The following function produces a simple MAPA forecast from a given origin. This is meant to be an internal function. Use mapafor instead.

Usage

```
mapacalc(
  y,
  mapafit,
  fh = NULL,
  comb = c("w.mean", "w.median", "mean", "median", "wght"),
  outplot = c(0, 1, 2),
  hybrid = c(TRUE, FALSE),
  xreg = NULL
)
```

Arguments

`y`	In sample observations of a time series (vector).
`mapafit`	Fitted MAPA model (from mapaest).
`fh`	Forecast horizon. Default = ppy. For compatibility with earlier versions, fh = 0 also gives the default.
`comb`	Combination operator. This can be: "mean"; "median"; "wght" - where each aggregation level is weighted inversly to aggregation; "w.mean" - level and trend components are averaged, but seasonal and xreg follow the wght combination; "w.median" - as w.mean, but with median. It is suggested that for data with high sampling frequency to use one of the "w.mean" and "w.median".

outplot	Provide output plot. 0 = no; 1 = time series and forecast only; 2 = time series, forecasts, and components. For the components the spectral colouring scheme is used. Red is the lowest aggregation level and blue the highest. Default is 0.
hybrid	Provide hybrid forecasts, as in Kourentzes et al. paper. If minimumAL > 1 then the minimumAL ETS forecasts are used. Default is TRUE.
xreg	Vector or matrix of exogenous variables to be included in the MAPA. If matrix then rows are observations and columns are variables. Must be at least as long as in-sample plus fh. Additional observations are unused.

Details

If mapafit was estimated with xreg, then xreg must also be provided here, with the same variables and at least length(y) + fh observations, so that the values for the forecast horizon are available. See mapaest for details on the use of xreg.

mapafit can also have the structure of earlier versions (<= 2.0.6), for example a fitted MAPA saved with an earlier version, or produced by mapafit.old. Note that mapafit.old is provided only for compatibility and will be removed in a later version of the package.

Value

forecast	Vector with forecasts. If y is a ts object, then this is also a ts object.
components	Array with MAPA components.
combined	Combined components across aggregation levels.

The output is of class mapa.calc, which also stores the time series and the settings used. It has print, summary, and plot methods, see plot.mapa.calc.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

mapafor, mapa.

Examples

```
mapafit <- mapaest(admissions)

# Forecast a full season ahead (default)
out <- mapacalc(admissions, mapafit)
print(out)

# Forecast 18 steps ahead, and plot the forecasts and the components
out <- mapacalc(admissions, mapafit, fh=18, outplot=2)
summary(out)
plot(out, components=c("level","season"), main="Admissions")

# Different combination operators, with and without the hybrid forecasts
out.mean <- mapacalc(admissions, mapafit, comb="mean")
out.wght <- mapacalc(admissions, mapafit, comb="wght", hybrid=FALSE)

# The same works for MAPA estimated with es from the smooth package
mapafit.es <- mapaest(admissions, type="es")
out.es <- mapacalc(admissions, mapafit.es, fh=18)

# MAPAx: xreg is needed both in estimation and forecasting. For forecasting
# it must contain the in-sample observations and the fh future values.
set.seed(1)
fh <- 12
x <- rnorm(length(admissions)+fh)
mapafit.x <- mapaest(admissions, type="es", xreg=x)
out.x <- mapacalc(admissions, mapafit.x, fh=fh, xreg=x)
plot(out.x, components=TRUE)
```

`mapaest` *MAPA Estimation*

Description

The following function estimates MAPA and saves all fitted model.

Usage

```
mapaest(
  y,
  ppy = NULL,
  minimumAL = 1,
  maximumAL = ppy,
  paral = c(0, 1, 2),
  display = c(0, 1),
  outplot = c(0, 1, 2),
  model = "ZZZ",
  type = c("ets", "es"),
  xreg = NULL,
```

```
  pr.comp = 0,
  cl = NULL,
  ...
)
```

Arguments

y	In sample observations of a time series (vector).
ppy	Periods in a season of the time series at the sampled frequency. If insample is a ts object then this is taken from its frequency, unless overriden.
minimumAL	Lowest aggregation level to use. Default = 1, maximumAL>1.
maximumAL	Highest aggregation level to use. Default = ppy.
paral	Use parallel processing. 0 = no; 1 = yes (requires initialised cluster); 2 = yes and initialise cluster. Default is 0.
display	Display calculation progress in console. 0 = no; 1 = yes. Default is 0.
outplot	Provide output plot. 0 = no; 1 or 2 = yes. Default is 0.
model	Allow only that type of ETS at each aggregation level. This follows similar coding to the ets function. The first letter refers to the error type ("A", "M", or "Z"); the second letter refers to the trend type ("N","A","Ad","M","Md", "Z", "X", or "Y"); and the third letter refers to the season type ("N","A","M", "Z", "X", or "Y"). The letters mean: "N"=none, "A"=additive, "M"=multiplicative, "Z"=automatically selected, "X"=automatically select between none and additive, and "Y"=automatically select between none and multiplicative. A "d" for trend implies damped. "X" and "Y" supported only by type=="es". If used with type=="ets" a warning will be given and they will default to "Z". By default model="ZZZ". If due to sample limitation ETS cannot be calculated at an aggregation level for the selected model, then no estimation is done for that specific level. For aggregation levels that seasonality becomes 1 then a non-seasonal model is estimated.
type	What type of exponential smoothing implementation to use. "es" = use from the smooth package; "ets" = use from the forecast package. Default is "es"
xreg	Vector or matrix of exogenous variables to be included in the MAPA. If matrix then rows are observations and columns are variables. Must be at least as long as in-sample. Additional observations are unused. Note that including xreg will force type="es".
pr.comp	MAPAx can use principal component analysis to preprocess xreg. When comp is -1 then the number of retained components is chosen automatically. When comp=0 then no pre-processing is performed and the original xreg is used. Any other value represents the number of principal components retained.
cl	Cluster to use when paral=1, created with `parallel::makeCluster`. If NULL, a cluster named `cl` in the global environment is used, for compatibility with earlier versions.
...	Pass additional arguments to es or ets.

Details

MAPA can use two implementations of exponential smoothing at each aggregation level. With `type="ets"` the `ets` function of the **forecast** package is used, which supports seasonal periods up to 24. With `type="es"` the `es` function of the **smooth** package is used, which supports any seasonal period, the model codes "X" and "Y", and exogenous variables (xreg). Additional arguments are passed to `ets` or `es` through `...`, for example the information criterion used for model selection (`ic="bic"` for `ets`, or `ic="BIC"` for `es`).

MAPAx: exogenous variables are provided with `xreg`, as a vector or as a matrix with one column per variable. These are aggregated in the same way as y at each aggregation level. Including xreg forces `type="es"`. For estimation xreg must have at least as many observations as y, and any additional observations are ignored. For forecasting, with `mapacalc`, `mapafor`, `mapa`, and `mapasimple`, xreg must also contain the values for the forecast horizon, so it must have at least length(y) + fh observations. The same xreg variables must be used in estimation and forecasting. When `pr.comp` is not 0, xreg is preprocessed with principal components. These are calculated in estimation and stored in the fitted MAPA, so that the same components are used in forecasting.

Since version 2.1.0 the fitted MAPA is a list, as described in Value. Earlier versions (<= 2.0.6) returned a list-matrix with one row per aggregation level. Code that depends on the earlier structure, for example indexing as `mapafit[,"use"]` or `mapafit[[1,14]]`, can transform the fitted MAPA to the earlier structure with `mapafit.old`. This is provided only for compatibility and will be removed in a later version of the package. Such code should be updated to use the current structure, for example `mapafit$use` or `mapafit$models[[1]]$components`.

Value

Estimated MAPA model structure, of class `mapa.fit`. This is a list containing:

`models`	List with the fitted model at each aggregation level, as returned by `ets` or `es`. NULL for levels that could not be estimated.
`AL`	Aggregation levels.
`use`	TRUE for the aggregation levels that were estimated.
`q`	Sample size at each aggregation level.
`ppyA`	Seasonal periods at each aggregation level.
`ppy`, `type`, `model`, `observations`	Settings used in the estimation.
`xreg.n`	Number of xreg variables.
`pr.comp`	Information for the preprocessing of xreg.

Earlier versions (<= 2.0.6) stored the fitted models in a list-matrix. Such objects are still accepted by `mapacalc`, `mapafor`, and the methods for `mapa.fit`.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

mapa, mapafor.

Examples

```
# Estimate MAPA using ets from the forecast package (default)
mapafit <- mapaest(admissions)
print(mapafit)
summary(mapafit)
plot(mapafit)

# Use es from the smooth package instead
mapafit.es <- mapaest(admissions, type="es")

# Restrict the model at each aggregation level. With type="es" the codes "X"
# and "Y" can be used, here to select between no and additive trend and
# seasonality, with additive errors
mapafit.es <- mapaest(admissions, type="es", model="AXX")

# Use only non-seasonal models, at aggregation levels 2 to 6
mapafit <- mapaest(admissions, model="ZZN", minimumAL=2, maximumAL=6)

# If y is not a ts object, then ppy must be provided
mapafit <- mapaest(as.numeric(admissions), ppy=12)

# MAPAx: estimate with exogenous variables, here 3 simulated ones. Only
# type="es" accepts xreg. Observations beyond the length of y are ignored
# in estimation, so the same xreg can be used later for forecasting
set.seed(1)
x <- matrix(rnorm((length(admissions)+12)*3), ncol=3)
mapafit.x <- mapaest(admissions, type="es", xreg=x)
summary(mapafit.x)
plot(mapafit.x)


# Preprocess xreg with principal components, retaining 2 components, or
# selecting the number of components automatically
mapafit.pc <- mapaest(admissions, type="es", xreg=x, pr.comp=2)
mapafit.pc <- mapaest(admissions, type="es", xreg=x, pr.comp=-1)
summary(mapafit.pc)

# Pass additional arguments to ets or es, here the information criterion
```

```
mapafit <- mapaest(admissions, ic="bic")
mapafit.es <- mapaest(admissions, type="es", ic="BIC")

# Parallel processing with an existing cluster
library(parallel)
cl <- makeCluster(2)
mapafit <- mapaest(admissions, paral=1, cl=cl)
stopCluster(cl)


# Code that depends on the structure of earlier versions can use mapafit.old.
# This will be removed in a later version of the package.
mapafit.prev <- suppressWarnings(mapafit.old(mapafit))
mapafit.prev[,"use"]
```

`mapafit.old` *Transform MAPA fit to the structure of earlier versions*

Description

Since version 2.1.0 the fitted MAPA model returned by `mapaest` is a list that keeps the complete fitted model at each aggregation level. Earlier versions (<= 2.0.6) stored it as a list-matrix, with one row per aggregation level. This function transforms the current structure to the earlier one, so that code that depends on the old indexing of the fitted MAPA keeps working.

This function is provided only for compatibility and will be removed in a later version of the package. A warning is given every time it is used.

Usage

```
mapafit.old(mapafit)
```

Arguments

`mapafit`	Fitted MAPA model (from mapaest).

Value

Fitted MAPA model as a list-matrix of class `mapa.fit`, with one row per aggregation level, as in MAPA <= 2.0.6. For `type="ets"` the columns are the 18 elements of the `ets` fit that were stored previously, followed by `use`, `AL`, `original.ppy`, `etstype`, and `pr.comp`. For `type="es"` the columns are the elements of the `es` fit that were stored previously, followed by `smoothLength`, `use`, `AL`, `original.ppy`, `etstype`, and `pr.comp`. The output is still accepted by `mapacalc`, `mapafor`, and the methods of `mapa.fit`.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

See Also

mapaest.

Examples

```
mapafit <- mapaest(admissions)

# Transform to the structure of earlier versions. A warning is given, as this
# function will be removed in a later version of the package.
mapafit.prev <- suppressWarnings(mapafit.old(mapafit))

# Old style indexing
mapafit.prev[,"use"]
mapafit.prev[[1,"method"]]

# The equivalent with the current structure
mapafit$use
mapafit$models[[1]]$method

# The earlier structure is still accepted for forecasting
mapacalc(admissions, mapafit.prev)
```

---

mapafor *MAPA Forecast*

Description

The following function produces in- and out-of-sample MAPA forecasts, for multiple steps ahead. This is the recommended function to use in forecasting with MAPA.

Usage

```
mapafor(
  y,
  mapafit,
  fh = NULL,
  ifh = 1,
  comb = c("w.mean", "w.median", "mean", "median", "wght"),
  outplot = c(0, 1),
  hybrid = c(TRUE, FALSE),
  conf.lvl = NULL,
  xreg = NULL,
  paral = c(0, 1, 2),
  cl = NULL
)
```

Arguments

y	In sample observations of a time series (vector).
mapafit	Fitted MAPA model (from mapaest).
fh	Forecast horizon. Default = ppy. If fh = 0 then only in-sample forecasts are produced. For compatibility with earlier versions, fh = -1 also gives the default.
ifh	In-sample forecast horizon. Default = 1.
comb	Combination operator. This can be: "mean"; "median"; "wght" - where each aggregation level is weighted inversly to aggregation; "w.mean" - level and trend components are averaged, but seasonal and xreg follow the wght combination; "w.median" - as w.mean, but with median. It is suggested that for data with high sampling frequency to use one of the "w.mean" and "w.median".
outplot	Provide output plot. 0 = no; 1 = yes. Default is 0.
hybrid	Provide hybrid forecasts, as in Kourentzes et al. paper. If minimumAL > 1 then the minimumAL ETS forecasts are used. Default is TRUE.
conf.lvl	Vector of confidence level for prediction intervals. Values must be (0,1). If conf.lvl == NULL then no intervals are calculated. For example to get the intervals for 80% and 95% use conf.lvl=c(0.8,0.95).
xreg	Vector or matrix of exogenous variables to be included in the MAPA. If matrix then rows are observations and columns are variables. Must be at least as long as in-sample plus fh. Additional observations are unused.
paral	Use parallel processing for the in-sample forecasts. 0 = no; 1 = yes (requires initialised cluster); 2 = yes and initialise cluster. Default is 0.
cl	Cluster to use when paral=1, created with `parallel::makeCluster`. If NULL, a cluster named `cl` in the global environment is used, for compatibility with earlier versions.

Details

If `mapafit` was estimated with xreg, then xreg must also be provided here, with the same variables and at least length(y) + fh observations, so that the values for the forecast horizon are available. See `mapaest` for details on the use of xreg.

`mapafit` can also have the structure of earlier versions (<= 2.0.6), for example a fitted MAPA saved with an earlier version, or produced by `mapafit.old`. Note that `mapafit.old` is provided only for compatibility and will be removed in a later version of the package.

Value

infor	In-sample forecasts.
outfor	Out-of-sample forecasts.
PI	Prediction intervals for given confidence levels.
MSE	In-sample MSE error.
MAE	In-sample MAE error.

If y is a ts object, then outfor is also a ts object. The output is of class `mapa.for`, which also stores the time series and the settings used. It has `print`, `summary`, and `plot` methods, see `plot.mapa.for`. The summary reports in-sample errors for each horizon.

Note

The calculation of the prediction intervals is based on the empirical multiple step ahead MSE. To speed up calculations set `conf.lvl=NULL`. If very long forecast horizons are requested then once no more t+h MSE can be calculated the following approximation is used: $\sqrt{MSE_{t+1}} * \sqrt{h}$ for the error.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

`mapa`, `mapaest`, `mapacalc`.

Examples

```
mapafit <- mapaest(admissions)

# Out-of-sample forecasts a full season ahead and in-sample 1-step ahead forecasts
out <- mapafor(admissions, mapafit)

# Forecast 18 steps ahead, with in-sample forecasts up to 3 steps ahead and
# 80% and 95% prediction intervals
out <- mapafor(admissions, mapafit, fh=18, ifh=3, conf.lvl=c(0.8,0.95))
print(out)
summary(out)
plot(out, main="Admissions", col.PI="lightblue")

# Only in-sample forecasts, for example to evaluate the fit up to 12 steps ahead
out <- mapafor(admissions, mapafit, fh=0, ifh=12)
summary(out)


# With es from the smooth package and the w.median combination
mapafit.es <- mapaest(admissions, type="es")
out.es <- mapafor(admissions, mapafit.es, comb="w.median")

# MAPAx: xreg must contain the in-sample observations and the fh future values
set.seed(1)
fh <- 12
x <- rnorm(length(admissions)+fh)
```

```
mapafit.x <- mapaest(admissions, type="es", xreg=x)
out.x <- mapafor(admissions, mapafit.x, fh=fh, xreg=x, conf.lvl=0.9)
plot(out.x)

# Parallel processing for the in-sample forecasts, with an existing cluster
library(parallel)
cl <- makeCluster(2)
out <- mapafor(admissions, mapafit, ifh=3, paral=1, cl=cl)
stopCluster(cl)
```

`mapasimple` *MAPA Single Origin Forecast (Wrapper)*

Description

The following function estimates MAPA and produces out-of-sample forecasts. It is not suggested, unless only simple single origin forecasts are required.

Usage

```
mapasimple(
  y,
  ppy = NULL,
  fh = NULL,
  minimumAL = 1,
  maximumAL = ppy,
  comb = c("w.mean", "w.median", "mean", "median", "wght"),
  paral = c(0, 1, 2),
  display = c(0, 1),
  outplot = c(0, 1, 2),
  hybrid = c(TRUE, FALSE),
  model = "ZZZ",
  type = c("ets", "es"),
  xreg = NULL,
  pr.comp = 0,
  cl = NULL,
  ...
)
```

Arguments

`y`	In sample observations of a time series (vector). If y == "paper" then it prints paper reference.
`ppy`	Periods in a season of the time series at the sampled frequency. If insample is a ts object then this is taken from its frequency, unless overriden.
`fh`	Forecast horizon. Default = ppy.

`minimumAL` Lowest aggregation level to use. Default = 1, maximumAL>1.

`maximumAL` Highest aggregation level to use. Default = ppy.

`comb` Combination operator. This can be: "mean"; "median"; "wght" - where each aggregation level is weighted inversly to aggregation; "w.mean" - level and trend components are averaged, but seasonal and xreg follow the wght combination; "w.median" - as w.mean, but with median. It is suggested that for data with high sampling frequency to use one of the "w.mean" and "w.median".

`paral` Use parallel processing. 0 = no; 1 = yes (requires initialised cluster); 2 = yes and initialise cluster. Default is 0.

`display` Display calculation progress in console. 0 = no; 1 = yes. Default is 0.

`outplot` Provide output plot. 0 = no; 1 = time series and forecast only; 2 = time series, forecasts, and components. For the components the spectral colouring scheme is used. Red is the lowest aggregation level and blue the highest. Default is 0.

`hybrid` Provide hybrid forecasts, as in Kourentzes et al. paper. If minimumAL > 1 then the minimumAL ETS forecasts are used. Default is TRUE.

`model` Allow only that type of ETS at each aggregation level. This follows similar coding to the ets function. The first letter refers to the error type ("A", "M", or "Z"); the second letter refers to the trend type ("N","A","Ad","M","Md", "Z", "X", or "Y"); and the third letter refers to the season type ("N","A","M", "Z", "X", or "Y"). The letters mean: "N"=none, "A"=additive, "M"=multiplicative, "Z"=automatically selected, "X"=automatically select between none and additive, and "Y"=automatically select between none and multiplicative. A "d" for trend implies damped. "X" and "Y" supported only by type=="es". If used with type=="ets" a warning will be given and they will default to "Z". By default model="ZZZ". If due to sample limitation ETS cannot be calculated at an aggregation level for the selected model, then no estimation is done for that specific level. For aggregation levels that seasonality becomes 1 then a non-seasonal model is estimated.

`type` What type of exponential smoothing implementation to use. "es" = use from the smooth package; "ets" = use from the forecast package. Default is "es"

`xreg` Vector or matrix of exogenous variables to be included in the MAPA. If matrix then rows are observations and columns are variables. Must be at least as long as in-sample. Additional observations are unused. Note that including xreg will force type="es".

`pr.comp` MAPAx can use principal component analysis to preprocess xreg. When comp is -1 then the number of retained components is chosen automatically. When comp=0 then no pre-processing is performed and the original xreg is used. Any other value represents the number of principal components retained.

`cl` Cluster to use when paral=1, created with `parallel::makeCluster`. If NULL, a cluster named `cl` in the global environment is used, for compatibility with earlier versions.

`...` Pass additional arguments to es or ets.

Details

This function calls *mapaest* and *mapacalc* internally.

Use `type="es"` for the `es` function of the **smooth** package, instead of the default `ets` function of the **forecast** package. See `mapaest` for details on the differences between the two and on the use of xreg.

Value

`forecast`	Vector with forecasts. If y is a ts object, then this is also a ts object.
`components`	Array with MAPA components.
`combined`	Combined components across aggregation levels.

The output is of class `mapa.calc`. See `mapacalc` and `plot.mapa.calc`.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

`mapa`.

Examples

```
# MAPA forecasts with ets from the forecast package (default)
out <- mapasimple(admissions)
print(out)

# With es from the smooth package, 24 steps ahead, plotting forecasts and components
out <- mapasimple(admissions, fh=24, type="es", outplot=2)

# Weighted combination, without the hybrid forecasts
out <- mapasimple(admissions, comb="wght", hybrid=FALSE)

# MAPAx: xreg must contain the in-sample observations and the fh future values
set.seed(1)
x <- rnorm(length(admissions)+12)
out <- mapasimple(admissions, fh=12, type="es", xreg=x)
summary(out)
```

`plot.mapa.calc` *Plot MAPA forecasts and components*

Description

The following function plots the forecasts produced by mapacalc or mapasimple, and optionally the forecasted components at each aggregation level. The black line in each component panel is the combination of the component across aggregation levels, as used for the forecasts. The plot is the same as the one that can be produced optionally by mapacalc.

Usage

```
## S3 method for class 'mapa.calc'
plot(x, components = c(FALSE, TRUE), col = NULL, col.AL = NULL, ...)
```

Arguments

`x`	MAPA forecasts (from mapacalc or mapasimple).
`components`	Plot the components. FALSE = no; TRUE = all available components. Alternatively a vector with any of "level", "trend", "season", and "xreg". Components that are not available are not plotted.
`col`	Vector of 2 colours for the time series and the forecasts. If NULL the default colours are used.
`col.AL`	Vector of colours for the components at each aggregation level. If NULL the spectral colouring scheme is used, with red the lowest and blue the highest aggregation level.
`...`	Additional arguments passed to `plot` for the forecast panel, for example `main`, `xlab`, `ylab`, `xlim`, `ylim`, and `lwd`.

Note

If y is a ts object, its time index is used for the horizontal axis. When only the forecasts are plotted the current graphical layout is respected, so the plot can be placed in a user defined grid. When components are plotted the layout is restored on exit.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- You can find more information about MAPA at Nikos' blog.

See Also

mapacalc, mapasimple.

Examples

```
mapafit <- mapaest(admissions)
out <- mapacalc(admissions, mapafit, fh=18)
plot(out)
plot(out, components=TRUE)
plot(out, components="season", main="Admissions", col=c("black","red"))

# Custom colours for the aggregation levels
plot(out, components=c("level","trend"), col.AL=rainbow(12))

# Place the forecast plot in a user defined layout
par(mfrow=c(1,2))
plot(out, xlim=c(2012,2015), main="Zoomed")
plot(mapafit)
par(mfrow=c(1,1))
```

---

plot.mapa.fit *Produce estimated MAPA fit plot*

Description

The following function plots the fitted components at each aggregation level of an estimated MAPA. The plot is the same as the one that can be produced optionally by mapaest. Aggregation levels for which seasonality cannot be estimated are in grey and levels that were not estimated due to insufficient sample are marked with '-'.

Usage

```
## S3 method for class 'mapa.fit'
plot(x, xreg.plot = c(TRUE, FALSE), col = NULL, ...)
```

Arguments

x	Fitted MAPA model (from mapaest).
xreg.plot	Add infromation about xreg in the figure.
col	Vector of 5 colours for: none, additive, additive damped, multiplicative, and multiplicative damped. If NULL the default colours are used.
...	Additional arguments passed to image, for example main, xlab, and ylab.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

mapaest, mapa.

Examples

```
mapafit <- mapaest(admissions)
plot(mapafit)
plot(mapafit, main="Admissions")

# With es and xreg, the number of xreg variables is shown at each level
set.seed(1)
x <- rnorm(length(admissions))
mapafit.x <- mapaest(admissions, type="es", xreg=x)
plot(mapafit.x)
plot(mapafit.x, xreg.plot=FALSE, col=heat.colors(5))
```

---

plot.mapa.for *Plot MAPA in- and out-of-sample forecasts*

Description

The following function plots the in- and out-of-sample forecasts and prediction intervals produced by mapafor or mapa. The plot is the same as the one that can be produced optionally by mapafor.

Usage

```
## S3 method for class 'mapa.for'
plot(
  x,
  insample = c(TRUE, FALSE),
  PI = c(TRUE, FALSE),
  col = NULL,
  col.PI = NULL,
  ...
)
```

Arguments

x	MAPA forecasts (from mapafor or mapa).
insample	Plot the in-sample forecasts, if available.
PI	Plot the prediction intervals, if available.
col	Vector of 2 colours for the time series and the forecasts. If NULL the default colours are used.
col.PI	Colour for the prediction intervals. When multiple intervals are available these are shaded from this colour. If NULL, a lighter shade of the forecast colour is used, so the intervals follow any colour set with col.
...	Additional arguments passed to plot, for example main, xlab, ylab, xlim, ylim, and lwd.

Note

If y is a ts object, its time index is used for the horizontal axis. The current graphical layout is respected, so the plot can be placed in a user defined grid.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- You can find more information about MAPA at Nikos' blog.

See Also

mapafor, mapa.

Examples

```
out <- mapa(admissions, ifh=3, conf.lvl=c(0.8,0.95))
plot(out)
plot(out, insample=FALSE, main="Admissions", ylab="Patients", col.PI="lightblue")
plot(out, PI=FALSE, lwd=2, col=c("grey50","blue"))

# The prediction intervals are shaded from the forecast colour, unless col.PI is given
plot(out, col=c("grey30","darkgreen"))
```

`plotmapa` *Produce estimated MAPA fit plot (OBSOLETE!)*

Description

The following function plots the fitted components at each aggregation level of an estimated MAPA. The plot is the same as the one that can be produced optionally by mapaest.

Usage

```
plotmapa(mapafit)
```

Arguments

`mapafit`	Fitted MAPA model (from mapaest).

Note

The function 'plotmapa' is deprecated. Use plot() instead.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>; Fotios Petropoulos.

References

- Kourentzes N., Petropoulos F., Trapero J.R. (2014) Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, **30**(**2**), 291–302.
- Kourentzes N., Petropoulos F. (2015) Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*.
- You can find more information about MAPA at Nikos' blog.

See Also

`mapaest`, `mapa`.

Examples

```
mapafit <- mapaest(admissions,outplot=0)
# plotmapa is deprecated, use plot instead
plot(mapafit)
```

print.mapa.fit *Summaries of MAPA estimation and forecasts*

Description

Print and summary methods for the outputs of mapaest, mapacalc, mapasimple, mapafor, and mapa.

Usage

```
## S3 method for class 'mapa.fit'
print(x, ...)

## S3 method for class 'mapa.fit'
summary(object, ...)

## S3 method for class 'mapa.calc'
print(x, ...)

## S3 method for class 'mapa.calc'
summary(object, ...)

## S3 method for class 'mapa.for'
print(x, ...)

## S3 method for class 'mapa.for'
summary(object, ...)
```

Arguments

...	Additional arguments, currently unused.
object, x	Output of mapaest (class mapa.fit), mapacalc or mapasimple (class mapa.calc), or mapafor or mapa (class mapa.for).

Details

The summary of mapa.fit reports for each aggregation level: the aggregated sample size, the seasonal periods, the fitted model, the smoothing parameters, and the number of xreg variables. Levels that could not be estimated due to insufficient sample are marked.

The summary of mapa.calc reports the combined level, trend, season, and xreg components for each horizon, alongside the forecasts.

The summary of mapa.for reports the in-sample errors (ME, MAE, and RMSE) for each in-sample forecast horizon, and the out-of-sample forecasts with any prediction intervals.

The summaries are returned as objects, so their contents can be used directly, for example summary(fit)$table.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

See Also

mapaest, mapacalc, mapafor.

Examples

```
mapafit <- mapaest(admissions)
summary(mapafit)
summary(mapafit)$table

# Summary of a MAPAx with principal component preprocessing of xreg
set.seed(1)
x <- matrix(rnorm(length(admissions)*3), ncol=3)
summary(mapaest(admissions, type="es", xreg=x, pr.comp=-1))

# Summaries of forecasts
summary(mapacalc(admissions, mapafit))
out <- mapafor(admissions, mapafit, ifh=3, conf.lvl=0.9)
summary(out)
summary(out)$errors
```

---

tsaggr *Non-overlapping temporal aggregation*

Description

Non-overlapping temporal aggregation.

Usage

```
tsaggr(
  y,
  fout,
  fmean = c(FALSE, TRUE),
  outplot = c(FALSE, TRUE),
  col.AL = NULL,
  ...
)
```

Arguments

y	Time series vector (can be ts object).
fout	Vector containing desirable aggregation levels. Must be positive and integer. If larger than length(y) then it is ignored.
fmean	If TRUE the aggregated is done using mean, otherwise sum is used.
outplot	If TRUE a plot of the original series and the aggregated ones is produced.
col.AL	Vector of colours for the aggregated series in the plot. If NULL the spectral colouring scheme is used, with red the lowest and blue the highest aggregation level, as in the component plots of plot.mapa.calc.
...	Additional arguments passed to plot, for example main, xlab, and ylab.

Value

out	List of temporally aggregated series. If y was a ts object, then 'out' has ts objects with appropriate frequencies. Any non-integer frequency is set equal to 1. Series are named ALx, where x is the aggregation level.
all	An array containing all aggregated series in the original frequency. Series are named ALx, where x is the aggregation level.
idx	List of indices used to produce 'out' from 'all': y.out[[i]] <- y.all[y.idx[[i]],i]. Series are named ALx, where x is the aggregation level.

Author(s)

Nikolaos Kourentzes, <nikolaos@kourentzes.com>.

Examples

```
# Aggregate to quarterly and annual series, using sums
out <- tsaggr(admissions, fout=c(3,12))
out$out$AL12

# All aggregation levels from 2 to 12, using means, with a plot
out <- tsaggr(admissions, fout=2:12, fmean=TRUE, outplot=TRUE, main="Admissions")

# User defined colours for the aggregated series
out <- tsaggr(admissions, fout=c(3,12), fmean=TRUE, outplot=TRUE, col.AL=c("blue","red"))
```

Index

∗ **~mapacalc**
  mapafor, 13
∗ **~mapaest**
  mapa, 3
  mapafit.old, 12
  mapafor, 13
∗ **~mapafor**
  mapa, 3
  mapacalc, 6
  mapaest, 8
∗ **~mapasimple**
  mapa, 3
∗ **~mapa**
  mapaest, 8
  mapafor, 13
  mapasimple, 16
∗ **~plotmapa**
  plot.mapa.fit, 20
  plotmapa, 23
∗ **~plot**
  plot.mapa.calc, 19
  plot.mapa.fit, 20
  plot.mapa.for, 21
∗ **~summary**
  print.mapa.fit, 24
∗ **~tsaggr**
  tsaggr, 25
∗ **datasets**
  admissions, 2

admissions, 2

mapa, 3, *7*, *11*, *15*, *18*, *21–23*
mapacalc, 6, *15*, *18*, *20*, *25*
mapaest, *4, 5*, *7*, 8, *13–15*, *18*, *21*, *23*, *25*
mapafit.old, *7*, *10*, 12, *14*
mapafor, *5*, *7*, *11*, 13, *22*, *25*
mapasimple, *5*, 16, *20*

plot.mapa.calc, *7*, *18*, 19, *25*
plot.mapa.fit, 20
plot.mapa.for, *5*, *14*, 21
plotmapa, 23
print.mapa.calc *(*print.mapa.fit*)*, 24
print.mapa.fit, 24
print.mapa.for *(*print.mapa.fit*)*, 24

summary.mapa.calc *(*print.mapa.fit*)*, 24
summary.mapa.fit *(*print.mapa.fit*)*, 24
summary.mapa.for *(*print.mapa.fit*)*, 24

tsaggr, 25