Package 'ensembleBMA'

July 22, 2025

Version 5.1.8

Date 2022-09-01

Title Probabilistic Forecasting using Ensembles and Bayesian Model Averaging

Author Chris Fraley, Adrian E. Raftery, J. McLean Sloughter, Tilmann Gneiting, University of Washington.

Maintainer Chris Fraley <fraley@u.washington.edu>

Depends R (>= 2.10), chron

Suggests fields, maps

Description Bayesian Model Averaging to create probabilistic forecasts from ensemble forecasts and weather observations <https://stat.uw.edu/sites/default/files/files/reports/2007/tr516.pdf>.

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2022-09-02 07:20:05 UTC

Contents

`brierScore` *Brier Scores*

Description

Computes Brier Scores for climatology, raw ensemble, and ensemble forecasting models given observation thresholds.

Usage

```
brierScore( fit, ensembleData, thresholds, dates = NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data.
`ensembleData`	An `ensembleData` object including ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. \ This need not be the data used for the model `fit`, although it must include the same ensemble members.
`thresholds`	One or more threshold values for the Brier score computations.

dates	The dates for which the Brier score will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.
...	Included for generic function compatibility.

Value

A data frame giving the Brier Scores for climatology (empirical distribution of the verifying observations), ensemble (voting), and ensemble foreacsting models for the specified thresholds.
A logistic Brier score is also given for the *BMAgamma0* model.

References

G. W. Brier, Verification of forecasts expressed in terms of probability, *Monthly Weather Review, 78:1–3, 1950*.

T. Gneiting and A. E. Raftery, Strictly proper scoring rules, prediction and estimation, *Journal of the American Statistical Association 102:359–378, 2007.*

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleBMA

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("PCP24","obs", sep = ".")
  ens <- paste("PCP24", ensMemNames, sep = ".")

  prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  prcpTestFit <- ensembleBMAgamma0( prcpTestData, trainingDays = 30)

## End(Not run)

  hist(prcpTestData$obs)

  brierScore(prcpTestFit, prcpTestData,
```

```
thresholds = seq(from = 0, to = .5, by = .1))
```

---

cdf *Cummulative Distribution Function for ensemble forcasting models*

Description

Computes the cumulative distribution function (CDF) of an ensemble forecasting model at observation locations.

Usage

```
cdf( fit, ensembleData, values, dates = NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data.
`ensembleData`	An `ensembleData` object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. \ This need not be the data used for the model `fit`, although it must include the same ensemble members.
`values`	The vector of desired values at which the CDF of the ensemble forecasting model is to be evaluated.
`dates`	The dates for which the CDF will be computed. These dates must be consistent with `fit` and `ensembleData`. The default is to use all of the dates in `fit`. The dates are ignored if `fit` originates from `fitBMA`, which also ignores date information.
`...`	Included for generic function compatibility.

Details

This method is generic, and can be applied to any ensemble forecasting model.
Note the model may have been applied to a power transformation of the data, but that information is included in the input `fit`, and the output is transformed appropriately.

Value

A vector of probabilities corresponding to the CDF at the desired values. Useful for determining propability of freezing, precipitation, etc.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

`ensembleBMA`, `fitBMA`, `quantileForecast`

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")


## Not run: # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

# for quick run only; use more training days for forecasting
  tempTestFit <- ensembleBMAnormal( tempTestData[1:20,], trainingDays = 8)

  tempTestForc <- quantileForecast( tempTestFit, tempTestData)
  range(tempTestForc)

  tempTestCDF <- cdf( tempTestFit, tempTestData,
                      values = seq(from=277, to=282, by = 1))

  tempTestCDF
```

combine *Combine Compatible BMA Models*

Description

Combines BMA models having the same characteristics for different dates.

Usage

```
combine( x, y, ...)
```

Arguments

x	An ensembleBMA model.
y	An ensembleBMA model having the same characteristics as x except for dates.
...	Other ensembleBMA models compatible with x and y.

Details

Input models are checked for compatibility, and entries from different inputs having the same dates are eliminated. Dates are ordered chronologically and ensemble members are ordered in the order in which they occur in inout x.

Value

An ensembleBMA model that merges the models from each input into a single model for the common dates.

References

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleBMA

Examples

```
data(ensBMAtest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
```

```
  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit12 <- ensembleBMAnormal( tempTestData, trainingDays = 30,
                   dates = c("2008010100","2008010200"))

  tempTestFit34 <- ensembleBMAnormal( tempTestData, trainingDays = 30,
                   dates = c("2008010300","2008010400"))

## End(Not run)

# for quick run only; use more training days for forecasting
  tempTestFit12 <- ensembleBMAnormal( tempTestData, trainingDays = 8,
                   dates = c("2008010100","2008010200"))

  tempTestFit34 <- ensembleBMAnormal( tempTestData, trainingDays = 8.,
                   dates = c("2008010300","2008010400"))

  tempTestFit <- combine( tempTestFit12, tempTestFit34)
```

`controlBMAgamma` *Control parameters for BMA wind speed modeling*

Description

Specifies a list of values controling the Bayesian Model Averaging fit of a mixture of gammas to ensemble forecasts for wind speed.

Usage

```
controlBMAgamma(maxIter, tol, power = 1, startupSpeed = NULL, init,
           optim.control = list(ndeps = rep( sqrt(.Machine$double.eps), 2)))
```

Arguments

`maxIter`	An integer specifying an upper limit on the number of iterations‘ for fitting the BMA mixture via EM. The default is `Inf`, which sets no upper limit on the number of iterations, so that the convergence criterion based on `eps` is used.
`tol`	A numeric convergence tolerance. The EM fit for the mixture of gammas is terminated when the relative error in successive objective values in the M-step falls below `tol`. The default is `sqrt(.Machine$double.eps)`, which is approximately `1.e-8` on IEEE compliant machines.

power	A scalar value giving the power by which the data will be transformed to fit the model for mean of the observations. The default is not to transform the data. The untransformed forecast is used to fit the variance model.
startupSpeed	A scalar value giving a global value for the anemometer startup speed, or the threshold below which a value of 0 is recorded. As this can vary from station to station and network to network, it may be preferable to include startupSpeed as part of the ensembleData object.
init	An optional list of initial values for variance coefficients and weights. The default is to start with the variance coefficients equal to 1, and with equal weights for each member of the ensemble.
optim.control	Control parameters for the optim function used in the M-step of EM. The default here is list(ndeps = rep(sqrt(.Machine$double.eps), 2)), which assigns a smaller finite-difference step size than the optim default of 1.e-3. To use the default control parameters for optim, set optim.control=NULL.

Value

A list whose components are the input arguments and their assigned values.

References

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010.*

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Ensemble Forecasting using Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleBMAgamma, fitBMAgamma

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("MAXWSP10","obs", sep = ".")
  ens <- paste("MAXWSP10", ensMemNames, sep = ".")

  winsTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
   winsTestFit1 <- ensembleBMAgamma(winsTestData, trainingDays = 30,
       control = controlBMAgamma(maxIter = 100, tol = 1.e-6,
```

```
   startupSpeed =1))

## End(Not run)

# for quick run only; use more training days for forecasting
   winsTestFit1 <- ensembleBMAgamma(winsTestData[1:14,], trainingDays = 5,
      control = controlBMAgamma(maxIter = 100, tol = 1.e-6, startupSpeed = 1))
```

`controlBMAgamma0` *Control parameters for BMA precipitation modeling*

Description

Specifies a list of values controling the Bayesian Model Averaging fit of a mixture of gammas with a point mass at 0 to ensemble forecasts for precipitation.

Usage

```
controlBMAgamma0(maxIter = Inf, tol = sqrt(.Machine$double.eps),
                 power = (1/3), rainobs = 10,
                 init = list(varCoefs = NULL, weights = NULL),
             optim.control = list(ndeps = rep( sqrt(.Machine$double.eps), 2)))
```

Arguments

`maxIter`	An integer specifying an upper limit on the number of iterations for fitting the BMA mixture via EM. The default is `Inf`, which sets no upper limit on the number of iterations, so that the convergence criterion based on `eps` is used.
`tol`	A numeric convergence tolerance. The EM fit for the mixture of gammas is terminated when the relative error in successive objective values in the M-step falls below `tol`. The default is `sqrt(.Machine$double.eps)`, which is approximately `1.e-8` on IEEE compliant machines.
`power`	A scalar value giving the power by which the data will be transformed to fit the models for the point mass at 0 and mean of nonzero observations. The default is to use the 1/3 power of the data. The untransformed forecast is used to fit the variance model.
`rainobs`	An integer specifying a minimum number of observations with nonzero precipitation in the training data. When necessary and possible, the training period will be extended backward in increments of days to meet the minimum requirement. It is not possible to fit the BMA model for precipitation without sufficient nonzero observations. The default minimum number is 10. It many instances fewer nonzero observations may suffice, but it could also be that more are needed to model precipitation in some datasets.
`init`	An optional list of initial values for variance coefficients and weights. The default is to start with the variance coefficients equal to 1, and with equal weights for each member of the ensemble.

`optim.control` Control parameters for the optim function used in the M-step of EM. The default here is list(ndeps = rep(sqrt(.Machine$double.eps), 2)), which assigns a smaller finite-difference step size than the `optim` default of 1.e-3. To use the default control parameters for `optim`, set `optim.control=NULL`.

Value

A list whose components are the input arguments and their assigned values.

References

J. M. Sloughter, A. E. Raftery, T Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Ensemble Forecasting using Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

`ensembleBMAgamma0`, `fitBMAgamma0`

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("PCP24","obs", sep = ".")
  ens <- paste("PCP24", ensMemNames, sep = ".")

  prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  prcpTestFit1 <- ensembleBMAgamma0( prcpTestData, trainingDays = 30,
       control = controlBMAgamma0(power = (1/4)))

## End(Not run)

# for quick run only; use more training days for forecasting
  prcpTestFit1 <- ensembleBMAgamma0( prcpTestData[1:14,], trainingDays = 6,
       control = controlBMAgamma0(power = (1/4)))
```

controlBMAnormal *Control parameters for BMA mixtures of normals*

Description

Specifies a list of values controling the Bayesian Model Averaging fit of a mixture of normals to ensemble forecasts.

Usage

```
controlBMAnormal(maxIter, tol, equalVariance, biasCorrection, init)
```

Arguments

maxIter	An integer specifying an upper limit on the number of iterations for fitting the BMA mixture via EM. The default is Inf, which sets no upper limit on the number of iterations, so that the convergence criterion based on eps is used.
tol	A numeric convergence tolerance. The EM fit for the mixture model is terminated when the relative error in successive objective values in the M-step falls below tol. The default is sqrt(.Machine$double.eps), which is approximately 1.e-8 on IEEE compliant machines.
equalVariance	A logical value indicating whether or not the variances for the mixture components should to be equal. The default is to constrain them to be equal.
biasCorrection	A character string describing the type of bias correction to be used.
	"regression" The bias correction term is formed by regression on the forecast values (including an intercept).
	"additive" The mean of the difference between observations and forecasts is used for bias correction.
	"none" No bias correction.
init	An optional list of initial values for standard deviations and weights. The default is to start with all standard deviations equal to 1, and with equal weights for each member of the ensemble.

Value

A list whose components are the input arguments and their assigned values.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleBMAnormal, fitBMAnormal

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit1 <- ensembleBMAnormal(tempTestData, trainingDays = 30,
    control = controlBMAnormal(maxIter = 100, biasCorrection = "additive"))

## End(Not run)

# for quick run only; use more training days for forecasting
  tempTestFit1 <- ensembleBMAnormal(tempTestData[1:20,], trainingDays = 5,
    control = controlBMAnormal(maxIter = 100, biasCorrection = "additive"))
```

---

crps *Continuous Ranked Probability Score*

Description

Computes the continuous ranked probability score (CRPS) for univariate ensemble forecasting models.

Usage

```
crps( fit, ensembleData, dates=NULL, nSamples=NULL, seed=NULL, ...)
CRPS( fit, ensembleData, dates=NULL, nSamples=NULL, seed=NULL, ...)
```

Arguments

fit	A model fit to ensemble forecasting data.

ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
nSamples	The number of simulation samples for CRPS via simulation. For the normal model, the default is analytic computation of the CRPS. For the gamma model with a point mass at 0 (precipitation), the CRPS is always computed by simulation, with default nSamples = 10000.
seed	Argument to set.seed for random number generation in simulation.
dates	The dates for which the CRPS will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.
...	Included for generic function compatibility.

Details

These methods are generic, and can be applied to all ensemble forecasting models.
For gamma0 model for precipitation and the gamma model for wind speed the CRPS is only available through simulation. The default number of simulation samples is 10,000.
Note that the gamma0 model for precipitation and the gamma model for wind speed may have been applied to a power transformation of the data.
For normal models for temperature and pressure, analytic computation of the CRPS is the default. CRPS will be computed via simulation for normal models only if nSamples is set to a positive value.
For the bivariate normal model for wind speed and direction, the CRPS is computed for the marginal wind speed distribution.

Value

crps is a matrix giving the CRPS for each instance in the data for both the raw ensemble and the median probabilistic forecast.
CRPS is a vector giving the mean of the CRPS over all instances for the raw ensemble and the median probabilistic forecast.

References

T. Gneiting and A. E. Raftery, Strictly proper scoring rules, prediction and estimation, *Journal of the American Statistical Association 102:359–378, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleBMA, fitBMA

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run: # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

# for quick run only; use more training days for forecasting
  tempTestFit <- ensembleBMAnormal( tempTestData[1:20,], trainingDays = 8)

  crpsValues <- crps( tempTestFit, tempTestData)
  colMeans(crpsValues)

  CRPS( tempTestFit, tempTestData)
```

---

dateCheck *Checks date format.*

Description

Checks that the character form of a vector of dates conforms to YYYYMMDDHH or YYYYMMDD.

Usage

```
dateCheck(YYYYMMDDHH)
```

Arguments

YYYYMMDDHH	A character vector (or its factor equivalent) of dates which should be in the form YYYYMMDDHH or YYYYMMDD, in which YYYY specifies the year, MM the month, DD the day, and (optionally) HH the hour.

Details

If both YYYYMMDDHH and YYYYMMDD are present, the YYYYMMDD dates are assumed to be in error even if HH == 00 for all of the longer dates.
Requires the chron library.

Value

A logical vector indicating whether or not each element of YYYYMMDDHH has the correct format.

See Also

ymdhTOjul, julTOymdh

Examples

```
dateCheck(c("2008043000", "20080431", "20080501"))
```

---

ensBMAtest *Ensemble BMA Test Data Set*

Description

This data set gives 48-hour forecasts for 2-m temperature, precipitation accumulated over the last 24 hours, and maximum wind speed at SeaTac (KSEA) and Portland (PDX) ariports in 2007/2008 initialized at 00 hours UTC using a 12km grid. The forecasts are based on an 8 member version of the University of Washington mesoscale ensemble (Grimit and Mass 2002; Eckel and Mass 2005).

Format

A data frame with 66 rows and 34 columns:
idate the initialization date of each forecast/observation, format YYYYMMDDHH (categorical).
vdate the validation date of each forecast/observation, format YYYYMMDDHH (categorical).
latitude the latitude of each forecast/observation (numeric).
longitude the longitude of each forecast/observation (numeric).
longitude the elevation (in meters) above sea level (numeric).
station weather station identifier (categorical).
network weather network identifier (categorical). *.gfs,*.cmcg,*.eta,*.gasp,*.jma,*.ngps,*.tcwb forecasts from the 8 members of the ensemble (numeric). *.obs observed values for the weather parameters. The prefix * is one of T2 for temperature, PCP24 for precipitation, MAXWSP10 for wind speed.

Details

Temperature is given in Kelvin.
Precipitation amounts are quantized to hundredths of an inch.
Maximum wind speed is defined as the maximum of the hourly 'instantaneous' wind speeds over the previous 18 hours, where an hourly 'instantaneous' wind speed is a 2-minute average from the period of two minutes before the hour to on the hour.
The wind speed observations are measured at 10-m above the ground and discretized when recorded by rounding to the nearest meter per second.
This is a small dataset provided for the purposes of testing. Typically forecasting would be performed on much larger datasets.

References

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

Examples

```
## Not run: # R check

  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

#-------------------------------------------------------------------------

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

  MAE( tempFit, tempTestData)
  CRPS( tempFit, tempTestData)

#-------------------------------------------------------------------------

  obs <- paste("PCP24","obs", sep = ".")
  ens <- paste("PCP24", ensMemNames, sep = ".")

  prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
```

```
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

  prcpTestFit <- ensembleBMAgamma0( prcpTestData, trainingDays = 30)

  MAE( prcpTestFit, prcpTestData)
  CRPS( prcpTestFit, prcpTestData)

#------------------------------------------------------------------------

  obs <- paste("MAXWSP10","obs", sep = ".")
  ens <- paste("MAXWSP10", ensMemNames, sep = ".")

  winsTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

   winsTestFit <- ensembleBMAgamma(winsTestData, trainingDays = 30)

   MAE( winsTestFit, winsTestData)
   CRPS( winsTestFit, winsTestData)

## End(Not run)
```

---

ensembleBMA *BMA mixture model fit*

Description

Fits a BMA mixture model to ensemble forecasts. Allows specification of a model, training rule, and forecasting dates.

Usage

```
ensembleBMA( ensembleData, trainingDays, dates = NULL, control = NULL,
             model = NULL, exchangeable = NULL, minCRPS = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by NA) are allowed.
dates	The dates for which BMA forecasting models are desired. By default, this will be all dates in ensembleData for which modeling is allowed given training rule.

trainingDays	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
control	A list of control values for the fitting functions. The default is controlBMAnormal() for normal models and controlBMAgamma0() for gamma models with a point mass at 0.
model	A character string describing the BMA model to be fit. Current choices are "normal", typically used for temperature or pressure data, and "gamma0", typically used for precipitation data.
exchangeable	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. The default determines exchangeability from ensembleData.
minCRPS	A logical variable indicating whether or not to add a postprocessing step after a normal BMA fit to choose the standard deviation so as to minimize the CRPS for the training data. This argument is used only for normal models, and the default is to not do the CRPS minimization for those models because it may require consderably more computation time, expecially when there are many ensemble members.

Details

If dates are specified in dates that cannot be forecast with the training rule, the corresponding BMA model parameter outputs will be missing (NA) but not NULL.
The training rule uses the number of days corresponding to its length regardless of whether or not the dates are consecutive.
The following methods are available for the output of ensembleBMA: cdf, quantileForecast, modelParameters, brierScore, crps, CRPS and MAE.

Value

A list with the following output components:

dateTable	The table of observations corresponding to the dates in x in chronological order.
trainingDays	The number of days in the training period as specified in input.
...	One or more components corresponding to fitted coefficients for the model.
weights	The fitted BMA weights for the mixture components for each ensemble member at each date.
power	A scalar value giving the power (if any) by which the data was transformed for modeling. The untransformed forecast is used to fit the variance model. This is input as part of control, and applies only to certain models.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian Model Averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

ensembleData, ensembleBMAnormal, ensembleBMAgamma0, ensembleBMAgamma, cdf, quantileForecast, modelParameters, brierScore, crps, MAE, controlBMAnormal, controlBMAgamma0, controlBMAgamma

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit <- ensembleBMA( tempTestData, trainingDays = 30,
                              model = "normal")

## equivalent to
##    tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

# for quick run only; use more training days for forecasting
  tempTestFit <- ensembleBMA( tempTestData[1:20,], trainingDays = 8,
                              model = "normal")

  set.seed(0); exch <- sample(1:length(ens),replace=TRUE)

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                exchangeable = exch,
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
```

```
                                initializationTime = "00")

# for quick run only; use more training days for forecasting
  tempTestFit <- ensembleBMA( tempTestData[1:20,], trainingDays = 8,
                              model = "normal")
```

---

ensembleBMAgamma *BMA wind speed modeling*

Description

Fits a Bayesian Model Averaging mixture of gammas to ensemble forecasts. Intended for predicting wind speed. Allows specification of a training period and forecasting dates.

Usage

```
ensembleBMAgamma( ensembleData, trainingDays, dates = NULL,
                  control = controlBMAgamma(), exchangeable = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by NA) are allowed.
trainingDays	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
dates	The dates for which forecasting models are desired. By default, this will be all dates in ensembleBMA for which modeling is allowed given the training rule.
control	A list of control values for the fitting functions. The defaults are given by the function controlBMAgamma0.
exchangeable	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The models fit will have equal weights and parameters within each group. The default determines exchangeability from ensembleData.

Details

The output is for all of the dates in ensembleBMA, so there will be missing entries denoted by NA for dates that are too recent to be forecast with the training rule.
The following methods are available for ensembleBMAgamma0 objects: cdf, quantileForecast, modelParameters, brierScore, crps, CRPS and MAE.

Value

A list with the following output components:

`training`	A list containing information on the training length and lag and the number of instances used for training for each modeling day.
`prob0coefs`	The fitted coefficients in the model for the point mass at 0 (probability of zero precipitaion) for each member of the ensemble at each date.
`biasCoefs`	The fitted coefficients in the model for the mean of the gamma components for each member of the ensemble at each date (bias correction).
`varCoefs`	The fitted coefficients for the model for the variance of gamma components for each date. The coefficients are the same for all members of the ensemble.
`weights`	The fitted BMA weights for the gamma components for each ensemble member at each date.
`power`	A scalar value giving to the power by which the data was transformed to fit the models for the point mass at 0 and the bias model. The untransformed forecast is used to fit the variance model. This is input as part of `control`.

References

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010.*

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

`ensembleData`, `controlBMAgamma`, `fitBMAgamma`, `cdf`, `quantileForecast`, `modelParameters`, `brierScore`, `crps`, `MAE`

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("MAXWSP10","obs", sep = ".")
  ens <- paste("MAXWSP10", ensMemNames, sep = ".")

  winsTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  winsTestFit <- ensembleBMAgamma(winsTestData, trainingDays = 30,
                                  control = controlBMAgamma(startupSpeed = 1))
```

```
## equivalent to
##     winsTestFit <- ensembleBMA(winsTestData, trainingDays = 30,
##                                model = "gamma")

## End(Not run)

# for quick run only; use more training days for forecasting
   winsTestFit <- ensembleBMAgamma(winsTestData[1:14,], trainingDays = 5,
                            control = controlBMAgamma(startupSpeed = 1))
```

`ensembleBMAgamma0` *BMA precipitation modeling*

Description

Fits a Bayesian Model Averaging mixture of gammas with a point mass at 0 to ensemble forecasts. Intended for predicting precipitation. Allows specification of a training rule and forecasting dates.

Usage

```
ensembleBMAgamma0( ensembleData, trainingDays, dates = NULL,
                   control = controlBMAgamma0(), exchangeable = NULL)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by `NA`) are allowed.
`trainingDays`	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
`dates`	The dates for which forecasting models are desired. By default, this will be all dates in `ensembleData` for which modeling is allowed given the training rule.
`control`	A list of control values for the fitting functions. The defaults are given by the function `controlBMAgamma0`.
`exchangeable`	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The models fit will have equal weights and parameters within each group. The default determines exchangeability from `ensembleData`.

Details

The output is for all of the `dates` in `ensembleBMA`, so there will be missing entries denoted by `NA` for dates that are too recent to be forecast with the training rule.
The following methods are available for `ensembleBMAgamma0` objects: `cdf`, `quantileForecast`, `modelParameters`, `brierScore`, `crps`, `CRPS` and `MAE`.

Value

A list with the following output components:

`training`	A list containing information on the training length and lag and the number of instances used for training for each modeling day.
`prob0coefs`	The fitted coefficients in the model for the point mass at 0 (probability of zero precipitaion) for each member of the ensemble at each date.
`biasCoefs`	The fitted coefficients in the model for the mean of the gamma components for each member of the ensemble at each date (bias correction).
`varCoefs`	The fitted coefficients for the model for the variance of gamma components for each date. The coefficients are the same for all members of the ensemble.
`weights`	The fitted BMA weights for the gamma components for each ensemble member at each date.
`power`	A scalar value giving to the power by which the data was transformed to fit the models for the point mass at 0 and the bias model. The untransformed forecast is used to fit the variance model. This is input as part of `control`.

References

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

See Also

`ensembleData`, `controlBMAgamma0`, `fitBMAgamma0`, `cdf`, `quantileForecast`, `modelParameters`, `brierScore`, `crps`, `MAE`

Examples

```
data(ensBMAtest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")

prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                              dates = ensBMAtest[,"vdate"],
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")
```

```
## Not run:  # R check
  prcpTestFit <- ensembleBMAgamma0( prcpTestData, trainingDays = 30)

## equivalent to
##    prcpTestFit <- ensembleBMA( prcpTestData, trainingDays = 30,
##                                model = "gamma0")

## End(Not run)

# for quick run only; use more training days for forecasting
  prcpTestFit <- ensembleBMAgamma0( prcpTestData[3:16,], trainingDays = 6)
```

`ensembleBMAnormal` *BMA mixture of normals modeling*

Description

Fits a Bayesian Model Averaging mixture of normals to ensemble forecasts. Allows specification of a training rule and forecasting dates.

Usage

```
ensembleBMAnormal(ensembleData, trainingDays, dates = NULL,
                  control = controlBMAnormal(), exchangeable = NULL,
                  minCRPS = FALSE)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by `NA`) are allowed.
`trainingDays`	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
`dates`	The dates for which BMA forecasting models are desired. By default, this will be all dates in `ensembleData` for which modeling is allowed given the training rule.
`control`	A list of control values for the fitting functions. The defaults are given by the function `controlBMAnormal`.
`exchangeable`	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The modeling will have equal weights and parameters within each group. The default determines exchangeability from `ensembleData`.
`minCRPS`	A logical variable indicating whether or not to add a postprocessing step after the BMA fit to choose the standard deviation so as to minimize the CRPS for the training data. The default is not to do the CRPS minimization, because it can add considerable extra cost to the computation, particularly when there are many ensemble members.

Details

The output is for all of the dates in ensembleData, so there will be missing entries denoted by NA for dates that are too recent to be forecast with the training rule.
The following methods are available for ensembleBMAnormal objects: cdf, quantileForecast, modelParameters, brierScore, crps, CRPS and MAE.

Value

A list with the following output components:

training	A list containing information on the training length and lag and the number of instances used for training for each modeling day.
biasCoefs	The fitted bias-correction coefficients for each ensemble member at each date.
sd	The fitted standard deviations for the mixture of normals model at each date.
weights	The fitted BMA weights for the normal components for each ensemble member at each date.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155-1174, 2005*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

See Also

ensembleData, controlBMAnormal, fitBMAnormal, cdf, quantileForecast, modelParameters, brierScore, crps, MAE

Examples

```
data(ensBMAtest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")

tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                              dates = ensBMAtest[,"vdate"],
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")
```

```
## Not run:  #  R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## equivalent to
##   tempTestFit <- ensembleBMA( tempTestData, trainingDays = 30,
##                               model = "normal")

## End(Not run)

# for quick run only; use more training days for forecasting
 tempTestFit <- ensembleBMAnormal( tempTestData[1:20,], trainingDays = 8)
```

`ensembleData` *Create an ensembleData object*

Description

Creates an `ensembleData` object including ensemble forecasts along with dates and (optionally) observations. Other descriptive information such as latitude, longitude, and station type may be included as well.

Usage

```
ensembleData( forecasts, dates = NULL, observations = NULL, ...,
              forecastHour, initializationTime,
              startupSpeed = NULL, exchangeable = NULL)
```

Arguments

`forecasts`	A matrix or array (for vector quantities) with columns corresponding to forecasts from individual members of an ensemble and rows corresponding to forecasts for the same date.
`dates`	A numeric or character vector or factor specifying the valid dates for the forecasts. If numeric, it is interpreted as a Julian date if it has an `origin` attribute specifying the month, day, and year, e.g. `c(month = 1, day = 1, year = 2000)`. Otherwise the character form of each date must be a string with format format YYYYMMDDHH or YYYYMMDD, where YYYY is the year, MM the monthn, DD the day, and (optionally) HH the hour.
`observations`	Optional vector (or matrix for vector quantities) of observed weather conditions corresponding to the forecasts. Must be supplied if the data is to be used for BMA modeling.
`...`	A named list of additional attributes such as latitude, longitude, and startupSpeed for wind speed.
`forecastHour`	A number giving the *forecast hour*, the time interval between the initialization and forecast times, in units of hours.

`initializationTime`
: A number or character string giving the initialization time.

`startupSpeed`
: A numeric value specifying a value below which the anemometer readings for wind speed will be recorded as zero. This value is used for all stations when the startup speed is not explicity specified as part of the data.

`exchangeable`
: A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The models fit will have equal weights and parameters within each group. The same names/labels should be used as for the forecasts. The default assumes that none of the ensemble members are exhangeable.

Details

For use with batch processing modeling functions (`ensembleBMA` etc), instances `ensembleData` object are assumed the same forecast hour and initialization time, which should be specified as part of the object.
Methods for `ensembleData` objects include `ensembleSize`, `ensembleForecasts`, `ensembleValidDates`. Subsetting is possible, but in the case of columns it applies only to the ensemble forecasts.
For vector wind computations, the velocity should be in the first column and the direction in the second.

Value

An `ensembleData` object, incorporating forecasts and (optionally) observations with the associated valid dates.

References

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

`ensembleBMA`, `ensembleBMAgamma`, `ensembleBMAgamma0`, `ensembleBMAnormal`

Examples

```
data(ensBMAtest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")


tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                              dates = ensBMAtest["vdate"],
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest["station"],
```

```
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check

  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)


## End(Not run)

  obs <- paste("PCP24","obs", sep = ".")
  ens <- paste("PCP24", ensMemNames, sep = ".")

  prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")
## Not run:  # R check

  prcpTestFit <- ensembleBMAgamma0( prcpTestData, trainingDays = 30)


## End(Not run)

  obs <- paste("MAXWSP10","obs", sep = ".")
  ens <- paste("MAXWSP10", ensMemNames, sep = ".")

  winsTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check

 winsTestFit <- ensembleBMAgamma(winsTestData, trainingDays = 30)


## End(Not run)
```

`fitBMA` *BMA model fit to a training set*

Description

Fits a Bayesian Modeling Averaging mixture model to a given training set.

Usage

```
fitBMA( ensembleData, control = NULL, model = NULL, exchangeable = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts and verification observations. Missing values (indicated by NA) are allowed. Dates are ignored if they are included. This is the training set for the model.
control	A list of control values for the fitting functions. The default is controlBMAnormal() for normal models and controlBMAgamma0() for gamma models with a point mass at 0.
model	A character string describing the BMA model to be fit. Current choices are "normal" for temperature or pressure data, and "gamma0" for precipitation data.
exchangeable	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. The default determines exchangeability from ensembleData.

Details

This function fits a BMA model to a training data set.
Methods available for fitBMA objects (the output of fitBMA) include: cdf, quantileForecast, and modelParameters.

Value

A list with the following output components:

...	One or more components corresponding to the coeffcients of the model.
weights	The fitted BMA weights for the mixture components for each ensemble member.
nIter	The number of EM iterations.
power	A scalar value giving the power (if any) by which the data was transformed for modeling. The untransformed forecast is used to fit the variance model. This is input as part of control, and applies only to certain models.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

ensembleData, ensembleBMA, fitBMAgamma, fitBMAgamma0, fitBMAnormal, cdf, quantileForecast, modelParameters, controlBMAgamma, controlBMAgamma0, controlBMAnormal

Examples

```
  data(ensBMAtest)

  ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                dates = ensBMAtest[,"vdate"],
                                forecastHour = 48,
                                initializationTime = "00")

  tempTrain <- trainingData( tempTestData, trainingDays = 30,
                             date  = "2008010100")

  tempTrainFit <- fitBMA( tempTrain, model = "normal")

## equivalent to
##    tempTrainFit <- fitBMAnormal( tempTrain)

  set.seed(0); exch <- sample(1:length(ens),replace=TRUE)

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                exchangeable = exch,
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                dates = ensBMAtest[,"vdate"],
                                forecastHour = 48,
                                initializationTime = "00")
```

---

fitBMAgamma *BMA wind speed model fit to a training set*

Description

Fits a Bayesian Modeling Averaging mixture of gammas. Intended for wind speed forecasts.

Usage

```
fitBMAgamma( ensembleData, control = controlBMAgamma(), exchangeable = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts and verification observations. Missing values (indicated by NA) are allowed. Dates are ignored if they are included. This is the training set for the model.
control	A list of control values for the fitting functions. The defaults are given by the function controlBMAgamma.
exchangeable	An optional numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. If supplied, this argument will override any specification of exchangeability in ensembleData.

Details

This function fits a BMA model to a training data set.
It is called by ensembleBMAgamma, which can produce a sequence of fits over a larger precipitation data set.
Methods available for the output of fitBMA include: cdf, quantileForecast, and modelParameters.

Value

A list with the following output components:

biasCoefs	The fitted coefficients in the model for the mean of nonzero observations for each member of the ensemble (used for bias correction).
varCoefs	The fitted coefficients for the model for the variance of nonzero observations (these are the same for all members of the ensemble).
weights	The fitted BMA weights for the gamma components for each ensemble member.
nIter	The number of EM iterations.
power	A scalar value giving to the power by which the data was transformed to fit the models for the point mass at 0 and the bias model. The untransformed forecast is used to fit the variance model. This is input as part of control.

References

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

ensembleData, controlBMAgamma, ensembleBMAgamma, cdf, quantileForecast, modelParameters

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("MAXWSP10","obs", sep = ".")
  ens <- paste("MAXWSP10", ensMemNames, sep = ".")

  winsTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                startupSpeed = 1,
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  winsTrain <- trainingData( winsTestData, trainingDays = 30,
                             date = "2008010100")

## End(Not run)

# for quick run only; use more training days for forecasting
  winsTrain <- trainingData( winsTestData, trainingDays = 10,
                             date = "2008010100")

  winsTrainFit <- fitBMAgamma( winsTrain)

## equivalent to
##    winsTrainFit <- fitBMA( winsTrain, model = "gamma")
```

`fitBMAgamma0` *BMA precipitation model fit to a training set*

Description

Fits a Bayesian Modeling Averaging mixture of gammas with a point mass at 0 to a given training set. Intended for precipitation forecasts.

Usage

```
fitBMAgamma0( ensembleData, control = controlBMAgamma0(),
              exchangeable = NULL)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts and verification observations. Missing values (indicated by `NA`) are allowed. Dates are ignored if they are included. This is the training set for the model.

control	A list of control values for the fitting functions. The defaults are given by the function controlBMAgamma0.
exchangeable	An optional numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. If supplied, this argument will override any specification of exchangeability in ensembleData.

Details

This function fits a BMA model to a training data set.
It is called by ensembleBMAgamma0, which can produce a sequence of fits over a larger precipitation data set.
Methods available for the output of fitBMA include: cdf, quantileForecast, and modelParameters.

Value

A list with the following output components:

prob0coefs	The fitted coefficients in the model for the point mass at 0 (probability of zero precipitation) for each member of the ensemble.
biasCoefs	The fitted coefficients in the model for the mean of nonzero observations for each member of the ensemble (used for bias correction).
varCoefs	The fitted coefficients for the model for the variance of nonzero observations (these are the same for all members of the ensemble).
weights	The fitted BMA weights for the gamma components for each ensemble member.
nIter	The number of EM iterations.
power	A scalar value giving to the power by which the data was transformed to fit the models for the point mass at 0 and the bias model. The untransformed forecast is used to fit the variance model. This is input as part of control.

References

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

See Also

ensembleData, controlBMAgamma0, ensembleBMAgamma0, cdf, quantileForecast, modelParameters

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("PCP24","obs", sep = ".")
  ens <- paste("PCP24", ensMemNames, sep = ".")

  prcpTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  prcpTrain <- trainingData( prcpTestData, trainingDays = 30,
                             date = "2008010100")

## End(Not run)

# quick run only; use more training days for forecasting
  prcpTrain <- trainingData( prcpTestData, trainingDays = 10,
                             date = "2008010100")

  prcpTrainFit <- fitBMAgamma0( prcpTrain)

## equivalent to
##    prcpTrainFit <- fitBMA( prcpTrain, model = "gamma0")
```

`fitBMAnormal` *BMA mixture of normals fit to a training set*

Description

Fits a Bayesian Model Averaging mixture of normals to a given training set.

Usage

```
fitBMAnormal( ensembleData, control = controlBMAnormal(),
              exchangeable = NULL)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts and verification observations. Missing values (indicated by `NA`) are allowed. Dates are ignored if they are included. This is the training set for the model.

control	A list of control values for the fitting functions. The defaults are given by the function controlBMAnormal.
exchangeable	An optional numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The models have equal weights and parameters within each group. If supplied, this argument will override any specification of exchangeability in ensembleData.

Details

This function fits a BMA model to a training data set.
It is called by ensembleBMAnormal, which can produce a sequence of fits over a larger data set.
Methods available for the output of fitBMAnormal include: cdf, quantileForecast, and modelParameters.

Value

A list with the following output components:

biasCoefs	The fitted bias-correction coefficients.
sd	The fitted standard deviations for the mixture of normals model (equal or varying across components according to the equalVariance setting in the control input).
weights	The fitted BMA weights for the normal components for each ensemble member.
nIter	The number of EM iterations.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian Model Averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

See Also

ensembleData, controlBMAnormal, ensembleBMAnormal, cdf, quantileForecast, modelParameters

Examples

```
  data(ensBMAtest)

  ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
```

```
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                dates = ensBMAtest[,"vdate"],
                                forecastHour = 48,
                                initializationTime = "00")

tempTrain <- trainingData( tempTestData, trainingDays = 30,
                           date  = "2008010100")

tempTrainFit <- fitBMAnormal( tempTrain)
```

`julTOymdh` *Convert Julian dates to character format.*

Description

Converts Julian dates to YYYYMMDDHH or YYYYMMDD character format.

Usage

```
julTOymdh( julianDates, origin = NULL, dropHour = NULL)
```

Arguments

`julianDates`	A numeric vector specifying Julian dates.
`origin`	A named vector specifying the month, day, and year for the origin of the Julian dates. The default is `c(month = 1, day = 1, year = 2000)`. The default is `attr(julianDates, "origin")` if it exists.
`dropHour`	A logical value indicating whether of not the hour information should be drop from the specifiation of the dates if none of the Julian dates are fractional. The default is `attr(julianDates, "dropHour")` if it exists.

Details

Requires the `chron` library.

Value

A character vector or numeric equivalent of dates in the form YYYYMMDDHH or YYYYMMDD, in which YYYY specifies the year, MM the month, DD the day, and (optionally) HH the hour corresponding to the Julian input.

See Also

`dateCheck`, `ymdhTOjul`

Examples

```
data(ensBMAtest)

julianIdates <- ymdhTOjul(ensBMAtest$idate)

all.equal( julTOymdh(julianIdates), as.character(ensBMAtest$idate))

all.equal( ymdhTOjul(ensBMAtest$vdate), julianIdates+2)
```

MAE *Mean Absolute Error*

Description

Computes the mean absolute error (MAE) for ensemble forecasting models.

Usage

```
MAE( fit, ensembleData, dates=NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data.
`ensembleData`	An `ensembleData` object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. \ This need not be the data used for the model `fit`, although it must include the same ensemble members.
`dates`	The dates for which the CRPS and MAE will be computed. These dates must be consistent with `fit` and `ensembleData`. The default is to use all of the dates in `fit`.
`...`	Included for generic function compatibility.

Details

This method is generic, and can be applied to all ensemble forecasting models.
Note the model may have been applied to a power transformation of the data, but that information is included in the input `fit`, and the output is transformed appropriately.

Value

A vector giving the MAE for the deterministic forecasts associated with the raw ensemble and for the ensemble forecasting model. This is the mean absolute difference of the raw ensemble medians and the observations, and the mean absolute difference of the median forecast and the observations (as in Sloughter et al. 2007). \ Note that Raftery et al. 2005 uses the mean absolute difference of the raw ensemble means and the observations, and the mean absolute difference of the BMA predictive mean and the observations.

References

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised in 2010).

See Also

ensembleBMA

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")
## Not run:  # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

  MAE( tempTestFit, tempTestData)
```

---

modelParameters *Extract model parameters*

Description

Extracts model parameters for ensemble forecasting models.

Usage

```
modelParameters( fit, ...)
```

Arguments

fit	A model fit to ensemble forecasting data.
...	For ensemble fits involving dates, there is an additional dates argument, giving a character representation of the dates for which model parameters are desired. In this case dates must correspond to the models in the fit and the default is to give the model parameters for all available dates.

Value

A list of parameters (including weights) corresponding to the ensemble forecasting model for the specified dates. The list may also include a power by which the forecasts were transformed to obtain the model parameters.

See Also

ensembleBMAgamma, ensembleBMAgamma0, ensembleBMAnormal, fitBMAgamma, fitBMAgamma0, fitBMAnormal

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

  modelParameters( tempTestFit, date = "2008010100")

  tempTrain <- trainingData( tempTestData, date = "2008010100",
                             trainingDays = tempTestFit$training$days)

  tempTrainFit <- fitBMAnormal( tempTrain)

  modelParameters( tempTrainFit)
```

---

pit *Probability Integral Transform for ensemble forcasting models*

Description

Computes the probabilty integral transform (PIT) of a BMA ensemble forecasting model at observation locations.

Usage

```
pit( fit, ensembleData, dates = NULL, randomizeATzero=FALSE, ...)
```

Arguments

fit	A model fit to ensemble forecasting data.
ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
dates	The dates for which the CDF will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.
randomizeATzero	For the gamma0 model for precipitation, observations should be randomized at 0 for assessing the calibration. Has no effect for the other models.
...	Included for generic function compatibility.

Details

Most often used for computing PIT histograms to assess calibration of forecasts, in which case the observations in ensembleData would be those used in modeling fit.
Instances in ensembleData without verifying observations are ignored.
Note the model may have been applied to a power transformation of the data, but that information is included in the input fit, and the output is transformed appropriately.
The PIT is a continuous analog of the verification rank.

Value

The value of the BMA cumulative distribution function CDF corresponding to the fit at the observed values in ensembleData.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

T. Gneiting, F. Balabdaoui and A. Raftery, Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society, Series B 69:243–268, 2007*.

J. M. Sloughter, A. E. Raftery, T Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

pitHist, verifRankHist, ensembleBMA, fitBMA, quantileForecast

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

  tempTestForc <- quantileForecast( tempTestFit, tempTestData)
  range(tempTestForc)

  tempTestPIT <- pit( tempTestFit, tempTestData)
```

---

pitHist *PIT Histogram*

Description

Computes the probability integral transform of the obervations relative to the BMA forecast, and plots its histogram.

Usage

```
pitHist( fit, ensembleData, dates=NULL)
```

Arguments

fit	A model fit to ensemble forecasting data.
ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
dates	The dates for which the CDF will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.

Details

PIT histograms are used to assess calibration of forecasts, in which case the observations in ensembleData would be those used in modeling fit.
Instances in ensembleData without verifying observations are ignored.
In the case of the gamma0 model for precipitation, observations of zero precipitation are randomized within their probabilistics range to avoid a false impression of bias.
Note the model may have been applied to a power transformation of the data, but that information is included in the input fit, and the output is transformed appropriately.
The PIT is a continuous analog of the verification rank.

Value

The value of the BMA cumulative distribution function CDF corresponding to the fit at the observed values in ensembleData. The corresponding histogram is also plotted.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155-1174, 2005*.

T. Gneiting, F. Balabdaoui and A. Raftery, Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society, Series B 69:243–268, 2007*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

ensembleData, pit, verifRankHist.

Examples

```
  data(srft)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData( forecasts = srft[ ,labels],
                            dates = srft$date,
                            observations = srft$obs,
                            latitude = srft$lat,
                            longitude = srft$lon,
                            forecastHour = 48,
                            initializationTime = "00")


## Not run:
# this takes time
# the PIT should be evaluated over relatively long periods

  srftFITall <- ensembleBMA( srftData, model = "normal", trainingDays = 25)

  srftPIT <- pitHist( srftFITall, srftData)

## End(Not run)
```

`plot.ensembleBMA` *Plot the Predictive Distribution Function for ensemble forcasting models*

Description

Plots the Predictive Distribution Function (PDF) of an ensemble forecasting model.

Usage

```
## S3 method for class 'ensembleBMAgamma'
plot( x, ensembleData, dates=NULL, ask=TRUE, ...)
## S3 method for class 'ensembleBMAgamma0'
plot( x, ensembleData, dates=NULL, ask=TRUE, ...)
## S3 method for class 'ensembleBMAnormal'
plot( x, ensembleData, dates=NULL, ask=TRUE, ...)
## S3 method for class 'fitBMAgamma'
plot( x, ensembleData, dates=NULL, ...)
## S3 method for class 'fitBMAgamma0'
plot( x, ensembleData, dates=NULL, ...)
## S3 method for class 'fitBMAnormal'
plot( x, ensembleData, dates=NULL, ...)
```

Arguments

x	A model fit to ensemble forecasting data.
ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
dates	The dates for which the PDF will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.
ask	A logical value indicating whether or not the user should be prompted for the next plot.
...	Included for generic function compatibility.

Details

This method is generic, and can be applied to any ensemble forecasting model.
The colored curves are the weighted PDFs of the ensemble members, and the bold curve is the overall PDF. The vertical black line represents the median forecast, and the dotted back lines represent the .1 and .9 quartiles. The vertical orange line is the verifying observation (if any).
Exchangeable members are represented in the plots by the weighted group sum rather than by the indivdual weighted PDFs of each member.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190-202, 2010*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

Examples

```
data(ensBMAtest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
```

```
  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run: # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)
  plot(tempTestFit, tempTestData)

## End(Not run)
```

---

plotProbcast *Surface plots for forecast information.*

Description

Produces contour, image, or perspective plot of a forecast using loess prediction on a grid.

Usage

```
plotProbcast( forecast, longitude, latitude, nGrid = 65,
                 type = c("image", "contour", "persp"), ...,
                 interpolate = FALSE, span = 0.75, maps = NULL)
```

Arguments

forecast	Numeric vector of forecasts.
longitude	Numeric vector giving the longitude of each forecast location.
latitude	Numeric vector giving the latitude of each forecast location.
nGrid	Number of grid points for loess interpolation. (Binning and interpolation are done on an nGrid by nGrid grid).
type	A character string indicating the desired plot type. Should be one of either "contour", "image", or "persp".
...	Additional arguments to be passed to the plotting method.
interpolate	A logical variable indicating whether or not a loess fit should be used to interpolate the data to points on a grid. The default is to determine grid values by binning, rather than interpolation.
span	Smoothing parameter for loess (used only when interpolate = TRUE). The default value is 0.75, which is the default for loess.
maps	A logical value indicating whether or not to include a map outline. The default is to include an outline if type = "image" and the fields library is loaded.

Details

If the `fields` library is loaded, a legend (and optionally a map outline) will be included in image plots.

Value

An image, contour, or perspective plot of the forecast.

References

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

See Also

`quantileForecast`

Examples

```
  data(srft)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData( forecasts = srft[,labels],
                            dates = srft$date, observations = srft$obs,
                            latitude = srft$lat, longitude = srft$lon,
                            forecastHour = 48, initializationTime = "00")

## Not run:  # R check

  bmaFit <- ensembleBMA( srftData, date = "2004012900", trainingDays = 25,
                         model = "normal")

  bmaForc <- quantileForecast( bmaFit, srftData, date = "2004012900",
                                  quantiles = c(.1, .5, .9))

  obs <- srftData$date == "2004012900"
  lat <- srftData$latitude[obs]
  lon <- srftData$longitude[obs]

  plotProbcast( bmaForc[,"0.5"], lat, lon,
                  type = "contour", interpolate = TRUE)
  title("Median Forecast")

  plotProbcast( srftData$obs[obs], lat, lon,
                type = "contour", interpolate = TRUE)
  title("Observed Surface Temperature")

  data(srftGrid)
```

```
memberLabels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

srftGridData <- ensembleData(forecasts = srftGrid[,memberLabels],
    latitude = srftGrid[,"latitude"], longitude = srftGrid[,"longitude"],
                          forecastHour = 48, initializationTime = "00")

gridForc <- quantileForecast( bmaFit, srftGridData,
                  date = "2004021400", quantiles = c( .1, .5, .9))

library(fields)

plotProbcast(gridForc[,"0.5"],lon=srftGridData$lon,
   lat=srftGridData$lat,type="image",col=rev(rainbow(100,start=0,end=0.85)))
title("Median Grid Forecast for Surface Temperature", cex = 0.5)

probFreeze <- cdf( bmaFit, srftGridData,  date = "2004021400",
                           value = 273.15)

plotProbcast(probFreeze, lon=srftGridData$lon, lat=srftGridData$lat,
                 type="image",col=gray((32:0)/32))
title("Probability of Freezing", cex = 0.5)


## End(Not run)
```

`prcpDJdata` *Precipitation Data*

Description

A subset of daily 48 hour forecasts of 24 hour accumulated precipitation over the US Pacific Northwest region from December 2002 to January 2003 based on a 9 member version of the University of Washington mesoscale ensemble (Grimit and Mass 2002; Eckel and Mass 2005). Precipitation amounts are quantized to hundredths of an inch.
Note that forecasts are not available for some of the interim dates.

Format

A data frame with 175 rows and 15 columns:
`CENT`,`AVN`,`CMCG`,`ETA`,`GASP`,`JMA`,`NGAPS`,`TCWB`,`UKMO` forecasts from the 9 members of the ensemble (numeric).
`observation` the observed accumulated precipitation (numeric).
`date` the date of each forecast/observation, format YYYYMMDDHH (categorical).
`station` weather station identifier (categorical).
`latitude` the latitude of each weather station (numeric).
`longitude` the longitude of each weather station (numeric).
`elevation` the elevation of each weather station (numeric).

Details

This dataset is a small subset of the data used in Sloughter et al. (2006), provided for the purposes of testing. Typically forecasting would be performed on much larger datasets.

References

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3309–3320, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

Examples

```
## Not run:  # R check

 data(prcpDJdata)
 data(prcpFit)

 prcpForc <- quantileForecast( prcpFit, prcpDJdata, date = "20030113",
                               quantiles = c( .1, .5, .9))

## End(Not run)
```

`prcpFit` *BMA Model Fit to Precipitation Data*

Description

The `ensembleBMAgamma0` model fit with a 30 day training period to the precipitation data set from *http://www.stat.washington.edu/MURI*, which gives daily daily 48 hour forecasts of 24 hour accumulated precipitation over the US Pacific Northwest region from December 12, 2002 through March 31, 2005 on a 9 member version of the University of Washington mesoscale ensemble (Grimit and Mass 2002; Eckel and Mass 2005). Precipitation amounts are quantized to hundredths of an inch.

Format

A list with the following arguments:

`dateTable` A named vector in which the names are the dates and the entries are the number of observations for each date.

`trainingRule` The training rule used to compute the model fits.

`prob0coefs` The coefficients in the logistic regression for probability of zero precipitation.

`biasCoefs` The coefficients in the linear regression for bias correction.

`varCoefs` The variance coefficients of the models.

`weights` The BMA weights for the models.

`power` An scalar value giving the power by which the forecasts are transformed for the BMA fitting.

References

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3309–3320, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

Examples

```
## Not run:  # R check

  data(prcpFit)

  modelParameters(prcpFit, date = "20030113")

  data(prcpGrid)

  prcpGridData <- ensembleData(forecasts = prcpGrid[,1:9],
                               latitude = prcpGrid[,"latitude"],
                               longitude = prcpGrid[,"longitude"],
                               forecsatHour = 48,
                               initializationTime = "00")

# probability of precipitation
  1 - cdf( prcpFit, prcpGridData, value = 0)

# probability of precipitation above 0.25 in
  1 - cdf( prcpFit, prcpGridData, date = "20030115", value = 25)


## End(Not run)
```

prcpGrid *Gridded Ensemble Forecasts of Precipitation*

Description

This data set gives 48-hour forecasts of 24 hour accumulated precipitation on a grid of locations in the US Pacific Northwest initialized on January 13, 2003 OOZ and valid on January 15, 2003 OOZ. The ensemble forecasts come from a nine member version of the University of Washington Mesoscale Ensemble (Grimit and Mass 2002; Eckel and Mass 2005). Precipitation amounts are quantized to hundredths of an inch.

Format

A data frame with 8188 rows and 11 columns:
`avn/gfs,cent,cmcg,eta,gasp,jma,ngps,tcwb,ukmo` forecasts from the 9 members of the ensemble (numeric).
`latitude` the latitude of each forecast (numeric).
`longitude` the longitude of each forecast (numeric).

References

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Ensemble Forecasting using Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2009.

Examples

```
## Not run:  # R check

  data(prcpGrid)

  prcpGridData <- ensembleData(forecasts = prcpGrid[,1:9],
                               latitude = prcpGrid[,"latitude"],
                               longitude = prcpGrid[,"longitude"],
                               forecastHour = 48,
                               initilaizationTime = "00")

  data(prcpFit)

# median forecast for Jan 15, 2003 at the grid points
```

```
  quantileForecast( prcpFit, prcpGridData, date = "20030115")

## End(Not run)
```

`quantileForecast` *Quantile forecasts at observation locations*

Description

Computes quantiles for the probability distribution function (PDF) for ensemble forecasting models.

Usage

```
quantileForecast( fit, ensembleData, quantiles = 0.5, dates=NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data.
`ensembleData`	An `ensembleData` object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. \ This need not be the data used for the model `fit`, although it must include the same ensemble members.
`quantiles`	The vector of desired quantiles for the PDF of the BMA mixture model.
`dates`	The dates for which the quantile forecasts will be computed. These dates must be consistent with `fit` and `ensembleData`. The default is to use all of the dates in `fit`. If `ensembleData` does not include dates, they will be inferred from `fit` and `dates`.
`...`	Included for generic function compatibility.

Details

This method is generic, and can be applied to any ensemble forecasting model.
Note the model may have been applied to a power transformation of the data, but that information is included in the input `fit`, and the output is transformed appropriately.
This can be used to compute prediction intervals for the PDF.
For the bivariate normal model for wind speed and direction, the CRPS is computed for the marginal wind speed distribution.

Value

A vector of forecasts corresponding to the desired quantiles.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155–1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

`ensembleBMA`, `fitBMA`, `cdf`

Examples

```
  data(ensBMAtest)

  ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

  obs <- paste("T2","obs", sep = ".")
  ens <- paste("T2", ensMemNames, sep = ".")

  tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                                dates = ensBMAtest[,"vdate"],
                                observations = ensBMAtest[,obs],
                                station = ensBMAtest[,"station"],
                                forecastHour = 48,
                                initializationTime = "00")

## Not run:  # R check
  tempTestFit <- ensembleBMAnormal( tempTestData, trainingDays = 30)

## End(Not run)

  tempTestForc <- quantileForecast( tempTestFit, tempTestData)


## Not run:  # R check

  data(srft)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData( forecasts = srft[ ,labels],
                            dates = srft$date,
```

```
                              observations = srft$obs,
                              latitude = srft$lat,
                              longitude = srft$lon,
                              forecastHour = 48,
                              initializationTime = "00")

  srftFit <- ensembleBMAnormal(srftData, date = "2004013100",
                               trainingDays = 25)

  data(srftGrid)

  srftGridData <- ensembleData(forecasts = srftGrid[ ,labels],
                               latitude = srftGrid$lat,
                               longitude = srftGrid$lon,
                               forecastHour = 48,
                               initializationTime = "00")

  srftGridForc <- quantileForecast( srftFit, srftGridData,
                    date = "2004013100")

## End(Not run)
```

`srft` *Surface Temperature Ensemble Forecasts and Observations*

Description

This data set gives 48-hour forecasts of 2-m surface temperature and the associated observations for the US Pacific Northwest from January 1, 2004 to February 28, 2004. The ensemble forecasts come from an eight-member version of the University of Washington Mesoscale Ensemble (Grimit and Mass 2002; Eckel and Mass 2005). Temperatures are measured in kelvins.
Note that forecasts are not available for some of the interim dates.

Format

A data frame with 36826 rows and 15 columns:
`CMCG,ETA,GASP,GFS,JMA,NGAPS,TCWB,UKMO` forecasts from the 8 members of the ensemble (numeric).
`observation` the observed surface temperature (numeric).
`date` the date of each forecast/observation set, in the format YYYYMMDDHH (categorical).
`latitude` the latitude of each forecast (numeric).
`longitude` the longitude of each forecast (numeric).
`station` weather station identifier (categorical).
`type` weather station type (categorical).

References

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

V. J. Berrocal, A. E. Raftery and T. Gneiting, Combining spatial and ensemble information in probabilistic weather forecasts, *Monthly Weather Review 133:1386–1402, 2007*.

V. J. Berrocal, A. E. Raftery, T. Gneiting and R. C. Steed, Probabilistic Weather Forecasting for Winter Road Maintenance, *Journal of the American Statistical Association, 2010 (to appear)*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

Examples

```
## Not run:  # R check

  data(srft)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData( forecasts = srft[ ,labels],
                            dates = srft$date,
                            observations = srft$obs,
                            latitude = srft$lat,
                            longitude = srft$lon,
                            forecastHour = 48,
                            initializationTime = "00")

  srftFit <- ensembleBMAnormal( srftData, date = "2004013100",
                                trainingDays = 25)

## End(Not run)
```

---

srftGrid — *Gridded Surface Temperature Ensemble Forecasts*

Description

This data set gives 48-hour forecasts of 2-m surface temperature on a grid of locations in the US Pacific Northwest initialized on January 29, 2004 00UTC and valid on January 31, 2004 00UTC. The ensemble forecasts come from an eight member version of the University of Washington Mesoscale Ensemble (Grimit and Mass 2002; Eckel and Mass 2005). Temperatures are measured in kelvins. Note that forecasts are not available for some of the interim dates.

Format

A data frame with 10098 rows and 10 columns:
CMCG,ETA,GASP,GFS,JMA,NGAPS,TCWB,UKMO forecasts from the 8 members of the ensemble (numeric).
latitude the latitude of each forecast (numeric).
longitude the longitude of each forecast (numeric).

References

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting 20:328–350, 2005*.

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting 17:192–205, 2002*.

V. J. Berrocal, A. E. Raftery and T. Gneiting, Combining spatial and ensemble information in probabilistic weather forecasts, *Monthly Weather Review 133:1386–1402, 2007*.

V. J. Berrocal, A. E. Raftery, T. Gneiting and R. C. Steed, Probabilistic Weather Forecasting for Winter Road Maintenance, *Journal of the American Statistical Association, 2010 (to appear)*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

Examples

```
## Not run:  # R check
  data(srft)
  data(srftGrid)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData( forecasts = srft[ ,labels],
                            dates = srft$date,
                            observations = srft$obs,
                            latitude = srft$lat,
                            longitude = srft$lon,
                            forecastHour = 48,
                            initializationTime = "00")

  srftFit <- ensembleBMAnormal( srftData, date = "2004013100",
                                trainingDays = 25)


  srftGridData <- ensembleData( forecasts = srftGrid[ ,labels],
                                latitude = srftGrid$lat,
                                longitude = srftGrid$lon,
                                forecastHour = 48,
                                initializationTime = "00")

   CRPS( srtGridData, srftFit)
```

```
## End(Not run)
```

`trainingData` *Extract Training Data*

Description

Extracts a subset of an `ensembleData` object corresponding to a given date and number of training days.

Usage

```
trainingData( ensembleData, trainingDays, date)
```

Arguments

`ensembleData`	An `ensembleData` object that includes, ensemble forecasts, observations and dates.
`trainingDays`	An integer specifying the number of days in the training period.
`date`	The date for which the training data is desired.

Details

The most recent days are used for training regardless of whether or not they are consecutive.

Value

An `ensembleData` object corresponding to the training data for the given date relative to `ensembleData`.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155-1174, 2005*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3309–3320, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

ensembleBMA, fitBMA

Examples

```
data(ensBMAtest)

ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

obs <- paste("T2","obs", sep = ".")
ens <- paste("T2", ensNames, sep = ".")


tempTestData <- ensembleData( forecasts = ensBMAtest[,ens],
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              dates = ensBMAtest[,"vdate"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTrain <- trainingData( tempTestData, trainingDays = 30,
                           date  = "2008010100")

tempTrainFit <- fitBMAnormal( tempTrain)
```

---

verifPlot — *Plot observations along with median, 10th and 90th percentile forecasts.*

Description

Computes the median, 10th and 90th percentile forecasts, and plots the corresponding observations.

Usage

```
verifPlot( fit, ensembleData, dates = NULL)
```

Arguments

fit	A model fit to ensemble forecasting data.
ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
dates	The dates for which the CDF will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. The dates are ignored if fit originates from fitBMA, which also ignores date information.

Value

A matrix giving the median, 10th and 90th percentile forecasts for the ensemble data at the specified dates. If observations are available, they are plotted along with the forecasts in order of increasing 90th percentile forecast.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155-1174, 2005*.

T. Gneiting, F. Balabdaoui and A. Raftery, Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society, Series B 69:243–268, 2007*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

`ensembleData`, `pit`

Examples

```
data(prcpFit)
data(prcpDJdata)

forc <- verifPlot( prcpFit, prcpDJdata, date = "20030113")
```

`verifRankHist` *Verification Rank and Histogram*

Description

Computes the rank of verifying observations relative to the corresponding ensemble forecasts and plots its histogram.

Usage

```
verifRankHist( forecasts, observations)
```

Arguments

forecasts	A matrix of ensemble forecasts, in which the rows corresponds to locations and times and the columns correspond to the individual ensemble members.
observations	A vector of observations corresponding to the locations and times of the forecasts.

Details

The verification rank is used to assess calibration of a forecast ensemble. A more uniform verification rank histogram indicates better calibartion.

Value

A vector giving the rank of verifying observations relative to the corresponding ensemble forecasts. The verification rank historgram is plotted.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review 133:1155-1174, 2005*.

T. Gneiting, F. Balabdaoui and A. Raftery, Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society, Series B 69:243–268, 2007*.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review 135:3209–3220, 2007*.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, ensembleBMA: An R Package for Probabilistic Forecasting using Ensemble and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, 2007 (revised 2010).

C. Fraley, A. E. Raftery, T. Gneiting, Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging, *Monthly Weather Review 138:190–202, 2010*.

J. M. Sloughter, T. Gneiting and A. E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, *Journal of the American Statistical Association, 105:25–35, 2010*.

See Also

ensembleData, pit

Examples

```
data(srft)

labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

srftData <- ensembleData( forecasts = srft[ ,labels],
                          dates = srft$date,
                          observations = srft$obs,
                          latitude = srft$lat,
                          longitude = srft$lon,
```

```
                          forecastHour = 48,
                          initializationTime = "00")

use <- ensembleValidDates(srftData) >= "2004013000"

verifRankHist( ensembleForecasts(srftData[use,]),
               dataVerifObs(srftData[use,]))
```

---

ymdhTOjul *Convert to Julian dates.*

Description

Converts YYYYMMDDHH or YYYYMMDD dates to Julian dates.

Usage

```
ymdhTOjul( YYYYMMDDHH, origin = c(month = 1, day = 1, year = 2000))
```

Arguments

YYYYMMDDHH	A character vector (or its factor equivalent) of dates in the form YYYYMMDDHH or YYYYMMDD, in which YYYY specifies the year, MM the month, DD the day, and (optionally) HH the hour.
origin	A named vector specifying the month, day, and year for the origin of the Julian dates. The default is `c(month = 1, day = 1, year = 2000)`.

Details

Requires the chron library.

Value

A vector of Julian dates corresponding to YYYYMMDDHH. The vector has `"origin"` and `"dropHour"` attributes which give the origin for the Julian output and indicate whether or not the original format included the hour.

See Also

dateCheck, julTOymdh

Examples

```
data(ensBMAtest)

julianVdates <- ymdhTOjul(ensBMAtest$vdate)
all.equal( julTOymdh(julianVdates), as.character(ensBMAtest$vdate))

all.equal( ymdhTOjul(ensBMAtest$idate), julianVdates-2)
```

Index