

# Package ‘EBMAforecast’

February 9, 2016

**Type** Package

**Title** Ensemble BMA Forecasting

**Version** 0.52

**Date** 2016-02-09

**Author** Jacob M. Montgomery, Florian M. Hollenbach, and Michael D. Ward

**Maintainer** Florian M. Hollenbach <fhollenbach@tamu.edu>

**Description** Ensemble BMA for social science data.

**License** GPL (>= 2)

**Depends** separationplot, plyr, methods, Hmisc, abind

**Imports** Rcpp (>= 0.11.3)

**LinkingTo** Rcpp

**RoxygenNote** 5.0.1

## R topics documented:

calibrateEnsemble . . . . .	2
calibrationSample . . . . .	4
compareModels . . . . .	5
EBMAforecast . . . . .	7
EBMAPredict . . . . .	8
ForecastData-class . . . . .	9
makeForecastData . . . . .	11
plot,FDatFitLogit-method . . . . .	13
presidentialForecast . . . . .	14
summary,FDatFitLogit-method . . . . .	15
testSample . . . . .	17
<b>Index</b>	<b>18</b>

calibrateEnsemble

*Calibrate an ensemble Bayesian Model Averaging model***Description**

This function calibrates an EBMA model based on out-of-sample performance in the calibration time period. Given a dependent variable and calibration-sample predictions from multiple component forecast models in the `ForecastData` the `calibrateEnsemble` function fits an ensemble BMA mixture model. The weights assigned to each model are derived from the individual model's performance in the calibration period. Missing observations are allowed in the calibration period, however models with missing observations are penalized. When missing observations are prevalent in the calibration set, the EM algorithm is adjusted and model parameters are estimated by maximizing a renormalized partial expected complete-data log-likelihood (Fraley et al. 2010).

**Usage**

```
calibrateEnsemble(.forecastData = new("ForecastData"), exp = 1,
  tol = sqrt(.Machine$double.eps), maxIter = 1e+06, model = "logit",
  method = "EM", ...)

fitEnsemble(.forecastData, tol = sqrt(.Machine$double.eps), maxIter = 1e+06,
  method = "EM", exp = 1, useModelParams = TRUE,
  predType = "posteriorMedian", const = 0, W = c(), ...)

## S4 method for signature 'ForecastDataNormal'
fitEnsemble(.forecastData,
  tol = sqrt(.Machine$double.eps), maxIter = 1e+06, method = "EM",
  exp = numeric(), useModelParams = TRUE, predType = "posteriorMedian",
  const = 0, W = c())
```

**Arguments**

<code>.forecastData</code>	An object of class 'ForecastData' that will be used to calibrate the model.
<code>exp</code>	The exponential shrinkage term. Forecasts are raised to the $(1/exp)$ power on the logit scale for the purposes of bias reduction. The default value is <code>exp=3</code> .
<code>tol</code>	Tolerance for improvements in the log-likelihood before the EM algorithm will stop optimization. The default is <code>tol= 0.01</code> , which is somewhat high. Researchers may wish to reduce this by an order of magnitude for final model estimation.
<code>maxIter</code>	The maximum number of iterations the EM algorithm will run before stopping automatically. The default is <code>maxIter=10000</code> .
<code>model</code>	The model type that should be used given the type of data that is being predicted (i.e., normal, binary, etc.).
<code>method</code>	The estimation method used. Currently only implements "EM".
<code>...</code>	Not implemented
<code>useModelParams</code>	If "TRUE" individual model predictions are transformed based on logit models. If "FALSE" all models' parameters will be set to 0 and 1.

predType	The prediction type used for the EBMA model under the normal model, user can choose either posteriorMedian or posteriorMean. Posterior median is the default.
const	user provided "wisdom of crowds" parameter, serves as minimum model weight for all models. Default = 0
W	Vector of initial model weights, if unspecified each model will receive weight 1/number of Models

### Value

Returns a data of class 'FdatFitLogit' or FdatFitNormal, a subclass of 'ForecastData', with the following slots

predCalibration	A matrix containing the predictions of all component models and the EBMA model for all observations in the calibration period.
predTest	A matrix containing the predictions of all component models and the EBMA model for all observations in the test period.
outcomeCalibration	A vector containing the true values of the dependent variable for all observations in the calibration period.
outcomeTest	An optional vector containing the true values of the dependent variable for all observations in the test period.
modelName	A character vector containing the names of all component models. If no model names are specified, names will be assigned automatically.
modelWeights	A vector containing the model weights assigned to each model.
modelParams	The parameters for the individual logit models that transform the component models.
useModelParams	Indicator whether model parameters for transformation were estimated or not.
logLik	The final log-likelihood for the calibrated EBMA model.
exp	The exponential shrinkage term.
tol	Tolerance for improvements in the log-likelihood before the EM algorithm will stop optimization.
maxIter	The maximum number of iterations the EM algorithm will run before stopping automatically.
method	The estimation method used.
iter	Number of iterations run in the EM algorithm.
call	The actual call used to create the object.

### Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

## References

- Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.
- Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.
- Raftery, A. E., T. Gneiting, F. Balabdaoui and M. Polakowski. (2005). Using Bayesian Model Averaging to calibrate forecast ensembles. *Monthly Weather Review*. **133**:1155–1174.
- Slughter, J. M., A. E. Raftery, T. Gneiting and C. Fraley. (2007). Probabilistic quantitative precipitation forecasting using Bayesian model averaging. *Monthly Weather Review*. **135**:3209–3220.
- Fraley, C., A. E. Raftery, T. Gneiting. (2010). Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging. *Monthly Weather Review*. **138**:190–202.
- Slughter, J. M., T. Gneiting and A. E. Raftery. (2010). Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. *Journal of the American Statistical Association*. **105**:25–35.
- Fraley, C., A. E. Raftery, and T. Gneiting. (2010). Calibrating multimodel forecast ensembles with exchangeable and missing members using Bayesian model averaging. *Monthly Weather Review*. **138**:190–202.

## Examples

```
## Not run: data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
  .outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
  .outcomeTest=testSample[, "Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

## End(Not run)
```

---

calibrationSample	<i>Calibration sample data</i>
-------------------	--------------------------------

---

## Description

This includes the calibration sample data for the insurgency forecasting example in Montgomery, Hollenbach and Ward (2012). It provides the predictions for the three models included in the Ensemble model, as well as the true values of the dependent variable for insurgency in 29 Asian countries. The calibration sample ranges from January 2008 to December 2009.

## Details

The variables included in the dataset are:

- LMER The calibration sample predictions of the LMER model from the insurgency prediction example in Montgomery et. al. (2012). The LMER model is a generalized linear mixed effects model using the logistic link function. It includes two random effects terms and several other covariates.
- SAE The calibration sample prediction of the SAE model from the insurgency prediction example in Montgomery et. al. (2012). This is a model developed as part of the ICEWS project and was designed by *Strategic Analysis Enterprises*. It is a simple generalized linear model with 27 independent variables.
- GLM The calibration sample prediction of the GLM model from the insurgency prediction example in Montgomery et. al. (2012). This is a crude logistic model with only four independent variables.
- Insurgency The true values of the dependent variable in the calibration sample from the insurgency prediction example in Montgomery et. al. (2012). This is a binary variable indicating the actual occurrence of insurgency for each observation in the calibration sample.

## References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis. Political Analysis.* **20**: 271-291.

---

compareModels	<i>Function for comparing multiple models based on predictive performance</i>
---------------	---

---

## Description

This function produces statistics to compare the predictive performance of the different models component models, as well as for the EBMA model itself, for either the calibration or the test period. It currently calculates the area under the ROC (auc), the brier score, the percent of observations predicted correctly (percCorrect), as well as the proportional reduction in error compared to some baseline model (pre) for binary models. For models with normally distributed outcomes the CompareModels function can be used to calculate the root mean squared error (rmse) as well as the mean absolute error (mae).

## Usage

```
compareModels(.forecastData, .period = "calibration",
  .fitStatistics = c("brier", "auc", "perCorrect", "pre"), .threshold = 0.5,
  .baseModel = 0, ...)
```

## Arguments

- .forecastData An object of class 'ForecastData'.
- .period Can take value of "calibration" or "test" and indicates the period for which the test statistics should be calculated.

<code>.fitStatistics</code>	A vector naming statistics that should be calculated. Possible values include "auc", "brier", "percCorrect", "pre" for logit models and "mae", "rsme" for normal models.
<code>.threshold</code>	The threshold used to calculate when a "positive" prediction is made by the model for binary dependent variables.
<code>.baseModel</code>	Vector containing predictions used to calculate proportional reduction of error ("pre").
<code>...</code>	Not implemented

### Value

A data object of the class 'CompareModels' with the following slots:

<code>fitStatistics</code>	The output of the fit statistics for each model.
<code>period</code>	The period, "calibration" or "test", for which the statistics were calculated.
<code>threshold</code>	The threshold used to calculate when a "positive" prediction is made by the model.
<code>baseModel</code>	Vector containing predictions used to calculate proportional reduction of error ("pre").

### Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

### References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

### See Also

ensembleBMA, other functions

### Examples

```
## Not run: data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
  .outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
  .outcomeTest=testSample[, "Insurgency"], .modelName=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

compareModels(this.ensemble, "calibration")

compareModels(this.ensemble, "test")
```

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

---

EBMAforecast

EBMAforecast

---

## Description

The EBMAforecast package (currently under development) allows users to increase the accuracy of forecasting models by pooling multiple component forecasts to generate ensemble forecasts. It includes functions to fit an ensemble Bayesian model averaging (EBMA) model using in-sample predictions, generate ensemble out-of-sample predictions, and create useful data visualizations. Currently, the package can only handle dichotomous outcomes or those with normally distributed errors, although additional models will be added to the package in the coming months. Missing observations are allowed in the calibration set, but models with many predictions missing are penalized.

## Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

## References

- Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.
- Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.
- Raftery, A. E., T. Gneiting, F. Balabdaoui and M. Polakowski. (2005). Using Bayesian Model Averaging to calibrate forecast ensembles. *Monthly Weather Review*. **133**:1155–1174.
- Slughter, J. M., A. E. Raftery, T. Gneiting and C. Fraley. (2007). Probabilistic quantitative precipitation forecasting using Bayesian model averaging. *Monthly Weather Review*. **135**:3209–3220.
- Fraley, C., A. E. Raftery, T. Gneiting. (2010). Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging. *Monthly Weather Review*. **138**:190–202.
- Slughter, J. M., T. Gneiting and A. E. Raftery. (2010). Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. *Journal of the American Statistical Association*. **105**:25–35.
- Fraley, C., A. E. Raftery, and T. Gneiting. (2010). Calibrating multimodel forecast ensembles with exchangeable and missing members using Bayesian model averaging. *Monthly Weather Review*. **138**:190–202.

## Examples

```
## Not run: demo(EBMAforecast)
demo(presForecast)

## End(Not run)
```

EBMAPredict

*EBMAPredict***Description**

Function allows users to create new predictions given an already estimated EBMA model. This function produces predictions based on EBMA model weights and component model predictions.

**Usage**

```
EBMAPredict(EBMAmodel, Predictions, Outcome = NULL, ...)
```

```
## S4 method for signature 'ForecastData'
EBMAPredict(EBMAmodel, Predictions, Outcome = NULL,
  ...)
```

```
prediction(EBMAmodel, Predictions, Outcome, ...)
```

```
## S4 method for signature 'FdatFitLogit'
prediction(EBMAmodel, Predictions, Outcome, ...)
```

```
## S4 method for signature 'FdatFitNormal'
prediction(EBMAmodel, Predictions, Outcome = c(),
  ...)
```

**Arguments**

EBMAmodel	Output of estimated EBMA model
Predictions	A matrix with a column for each component model's predictions.
Outcome	An optional vector containing the true values of the dependent variable for all observations in the test period.
...	Not implemented

**Value**

Returns a data of class 'FdatFitLogit' or FdatFitNormal, a subclass of 'ForecastData', with the following slots:

predTest	A matrix containing the predictions of all component models and the EBMA model for all observations in the test period.
----------	---

#'

period	The period, "calibration" or "test", for which the statistics were calculated.
outcomeTest	An optional vector containing the true values of the dependent variable for all observations in the test period.
modelNames	A character vector containing the names of all component models. If no model names are specified, names will be assigned automatically.
modelWeights	A vector containing the model weights assigned to each model.



**Author(s)**

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>  
and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

**References**

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

---

ForecastData-class	<i>An ensemble forecasting data object</i>
--------------------	--

---

**Description**

Objects of class ForecastData are used in the calibrateEnsemble function. Datasets should be converted into an object of class ForecastData using the makeForecastData function. Individual slots of the ForecastData object can be accessed and changed using the get and set functions respectively. Missing observations in the prediction calibration set are allowed.

**Usage**

```
setPredCalibration(object) <- value

## S4 replacement method for signature 'ForecastData'
setPredCalibration(object) <- value

setPredTest(object) <- value

## S4 replacement method for signature 'ForecastData'
setPredTest(object) <- value

setOutcomeCalibration(object) <- value

## S4 replacement method for signature 'ForecastData'
setOutcomeCalibration(object) <- value

setOutcomeTest(object) <- value

## S4 replacement method for signature 'ForecastData'
setOutcomeTest(object) <- value

setModelNames(object) <- value

## S4 replacement method for signature 'ForecastData'
setModelNames(object) <- value
```

**Arguments**

object	used for validity checks (internal)
value	used for validity checks (internal) #hack but no idea how to get the warning to go away otherwise

**Details**

A data object of the class 'ForecastData' has the following slots: @slot predCalibration An array containing the predictions of all component models for the observations in the calibration period. @slot predTest An array containing the predictions of all component models for all the observations in the test period. @slot outcomeCalibration A vector containing the true values of the dependent variable for all observations in the calibration period. @slot outcomeTest A vector containing the true values of the dependent variable for all observations in the test period. @slot modelNames A character vector containing the names of all component models.

**Author(s)**

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

**References**

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

**See Also**

ensembleBMA

**Examples**

```
## Not run: data(calibrationSample)

data(testSample)
this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[, "Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

### to acces individual slots in the ForecastData object
getPredCalibration(this.ForecastData)
getOutcomeCalibration(this.ForecastData)
getPredTest(this.ForecastData)
getOutcomeTest(this.ForecastData)
getModelNames(this.ForecastData)

### to assign individual slots, use set functions

setPredCalibration(this.ForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(this.ForecastData)<-calibrationSample[, "Insurgency"]
setPredTest(this.ForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(this.ForecastData)<-testSample[, "Insurgency"]
```

```
setModelNames(this.ForecastData)<-c("LMER", "SAE", "GLM")

## End(Not run)
```

---

makeForecastData	<i>Build a ensemble forecasting data object</i>
------------------	---

---

## Description

This function uses the component model forecasts and dependent variable observations provided by the user to create an object of class `ForecastData`, which can then be used to calibrate and fit the ensemble. Individual slots of the `ForecastData` object can be accessed and changed using the `get` and `set` functions respectively. Missing predictions are allowed in the calibration set.

## Usage

```
makeForecastData(.predCalibration = array(NA, dim = c(0, 0, 0)),
  .predTest = array(NA, dim = c(0, 0, 0)), .outcomeCalibration = numeric(),
  .outcomeTest = numeric(), .modelNames = character(), ...)

## S4 method for signature 'ANY'
makeForecastData(.predCalibration, .predTest,
  .outcomeCalibration, .outcomeTest, .modelNames)

## S4 method for signature 'ForecastData'
print(x, digits = 3, ...)

## S4 method for signature 'ForecastData'
show(object)
```

## Arguments

<code>.predCalibration</code>	A matrix with the number of rows being the number of observations in the calibration period and a column with calibration period predictions for each model.
<code>.predTest</code>	A vector with the number of rows being the number of observations in the test period and a column with test period predictions for each model.
<code>.outcomeCalibration</code>	A vector with the true values of the dependent variable for each observation in the calibration period.
<code>.outcomeTest</code>	A vector with the true values of the dependent variable for each observation in the test period.
<code>.modelNames</code>	A vector of length <code>p</code> with the names of the component models.
<code>...</code>	Additional arguments not implemented
<code>x</code>	A data object of class <code>'ForecastData'</code>
<code>digits</code>	User specified number of digits to be displayed.
<code>object</code>	A data object of class <code>'ForecastData'</code>

Additionally, the functions `show` and `print` can be used to display data objects of class `'ForecastData'`. `show` displays only 1 digit and takes the following parameters:

**Value**

A data object of the class 'ForecastData' with the following slots:

predCalibration	An array containing the predictions of all component models for all observations in the calibration period.
predTest	An array containing the predictions of all component models for all observations in the test period.
outcomeCalibration	A vector containing the true values of the dependent variable for all observations in the calibration period.
outcomeTest	A vector containing the true values of the dependent variable for all observations in the test period.
modelNameNames	A character vector containing the names of all component models. If no model names are specified, names will be assigned automatically.

**Author(s)**

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

**References**

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

**See Also**

ensembleBMA

**Examples**

```
data(calibrationSample)

## Not run: data(testSample)
this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[, "Insurgency"], .modelNameNames=c("LMER", "SAE", "GLM"))

### to acces individual slots in the ForecastData object
getPredCalibration(this.ForecastData)
getOutcomeCalibration(this.ForecastData)
getPredTest(this.ForecastData)
getOutcomeTest(this.ForecastData)
getModelNames(this.ForecastData)

### to assign individual slots, use set functions

setPredCalibration(this.ForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(this.ForecastData)<-calibrationSample[, "Insurgency"]
```

```

setPredTest(this.ForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(this.ForecastData)<-testSample[, "Insurgency"]
setModelNames(this.ForecastData)<-c("LMER", "SAE", "GLM")

## End(Not run)

```

---

plot, FDatFitLogit-method

*Plotting function for ensemble models of the class "FDatFitLogit" or "FDatFitNormal", which are the objects created by the calibrateEnsemble() function.*

---

## Description

Default plotting for objects created by the "calibrateEnsemble" function. See details below.

## Usage

```

## S4 method for signature 'FDatFitLogit'
plot(x, period = "calibration", subset = 1,
     mainLabel = "", xLab = "", yLab = "", cols = 1, ...)

```

## Arguments

x	An object of class "FDatFitLogit" or "FDatFitNormal"
period	Can take value of "calibration" or "test" and indicates the period for which the plots should be produced.
subset	The row names or numbers for the observations the user wishes to plot. Only implemented for the subclass "FDatFitNormal"
mainLabel	A vector strings to appear at the top of each predictive posterior plot. Only implemented for the subclass "FDatFitNormal"
xLab	The label for the x-axis. Only implemented for the subclass "FDatFitNormal"
yLab	The label for the y-axis. Only implemented for the subclass "FDatFitNormal"
cols	A vector containing the color for plotting the predictive pdf of each component model forecast. Only implemented for the subclass "FDatFitNormal"
...	Not implemented

## Details

For objects of the class "FDatFitLogit", this function creates separation plots for each of the fitted models, including the EBMA model. Observations are ordered from left to right with increasing predicted probabilities, which is depicted by the black line. Actual occurrences are displayed by red vertical lines. Plots can be displayed for the test or calibration period. For objects of the class "FDatFitNormal", this function creates a plot of the predictive density distribution containing the EBMA PDF and the PDFs for all component models (scaled by their model weights). It also plots the prediction for the ensemble and the components for the specified observations.

**Author(s)**

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>  
and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

**References**

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

Greenhill, B., M.D. Ward, A. Sacks. (2011). The Separation Plot: A New Visual Method For Evaluating the Fit of Binary Data. *American Journal of Political Science*. **55**: 991–1002.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2015). Calibrating ensemble forecasting models with sparse data in the social sciences. *International Journal of Forecasting*. In Press.

**See Also**

separationplot

**Examples**

```
data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
  .outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
  .outcomeTest=testSample[, "Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

plot(this.ensemble, period="calibration")
plot(this.ensemble, period="test")
```

---

presidentialForecast    *Sample data Presidential Election*

---

**Description**

This includes the data for the presidential election forecasting example in Montgomery, Hollenbach and Ward (2012). The data ranges from 1952 to 2008 and includes predictions for the six different component models included in the Ensemble model. Users may split the sample into calibration and test sample.

## Details

The variables included in the dataset are:

- Campbell Predictions of Campbell's "Trial-Heat and Economy Model" (Campbell 2008).
- Abramowitz Predictions of Abramowitz's "Time for Change Model" (Abramowitz 2008).
- Hibbs Predictions for the "Bread and Peace Model" created by Douglas Hibbs (2008).
- Fair Forecasts from Fair's presidential vote share model (2010).
- Lewis-Beck/Tien Predictions from the "Jobs Model Forecast" by Michael Lewis-Beck and Charles Tien (2008).
- EWT2C2 Predictions from the model in Column 2 in Table 2 by Erickson and Wlezien (2008).
- Actual The true values of the dependent variable, i.e. the incumbent-party voteshare in each presidential election in the sample.

## References

- Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.
- Campbell, James E. 2008. The trial-heat forecast of the 2008 presidential vote: Performance and value considerations in an open-seat election. *PS: Political Science & Politics* **41**:697-701.
- Hibbs, Douglas A. 2000. Bread and peace voting in U.S presidential elections. *Public Choice* **104**:149-180.
- Fair, Ray C. 2010. Presidential and Congressional vote-share equations: November 2010 update. Working Paper. Yale University.
- Lewis-Beck, Michael S. and Charles Tien. 2008. The job of president and the jobs model forecast: Obama for '08? *PS: Political Science & Politics* **41**:687-690.
- Erikson, Robert S. and Christopher Wlezien. 2008. Leading economic indicators, the polls, and the presidential vote. *PS: Political Science & Politics* **41**:703-707.

---

summary,FDatFitLogit-method

*Summary Function*

---

## Description

This function summarizes the Ensemble models that have been fit previously by the user.

## Usage

```
## S4 method for signature 'FDatFitLogit'
summary(object, period = "calibration",
  fitStatistics = c("brier", "auc", "perCorrect", "pre"), threshold = 0.5,
  baseModel = 0, showCoefs = TRUE, ...)
```

**Arguments**

object	An object of the subclass "FDatFitLogit" or "FDatFitNormal"
period	The period for which the summary should be provided, either "calibration" or "test".
fitStatistics	A vector naming statistics that should be calculated. Possible values for objects in the "FDatFitLogit" subclass include "auc", "brier", "percCorrect", "pre". Possible values for objects in the "FDatFitNormal" subclass include "rmse" and "mae." Additional metrics will be made available in a future release of this package.
threshold	The threshold used to calculate when a "positive" prediction is made for a model. Not used for objects of the "FDatFitNormal" subclass.
baseModel	A vector containing predictions used to calculate proportional reduction of error ("pre"). Not used for objects of the "FDatFitNormal" subclass.
showCoefs	A logical indicating whether model coefficients from the ensemble should be shown.
...	Not implemented

**Value**

A data object of the class 'SummaryForecastData' with the following slots:

summaryData	Under the default, the function produces a matrix containing one row for each model plus one row for the EBMA forecast. The first column is always the model weights assigned to the component models. The second and third columns display the model parameters for the transformation of the component models. The remaining columns are the requested fit statistics for all models, as calculated by the copareModels function. If showCoefs=FALSE, then the model parameters will not be shown.
-------------	--

**Author(s)**

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>  
and Florian M. Hollenbach <<florian.hollenbach@tamu.edu>>

**Examples**

```
## Not run: data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
  .outcomeCalibration=calibrationSample[, "Insurgency"], .predTest=testSample[,c("LMER", "SAE", "GLM")],
  .outcomeTest=testSample[, "Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

summary(this.ensemble, period="calibration")

summary(this.ensemble, period="test", showCoefs=FALSE)

## End(Not run)
```



---

testSample	<i>Test sample data</i>
------------	-------------------------

---

### Description

This includes the test sample data for the insurgency forecasting example in Montgomery, Hollenbach and Ward (2012). It provides the predictions for the three models included in the Ensemble model, as well as the true values of the dependent variable for insurgency in 29 Asian countries. The test sample ranges from January 2010 to December 2010.

### Details

The variables included in the dataset are:

- LMER The test sample predictions of the LMER model from the insurgency prediction example in Montgomery et. al. (2012). The LMER model is a generalized linear mixed effects model using the logistic link function. It includes two random effects terms and several other covariates.
- SAE The test sample prediction of the SAE model from the insurgency prediction example in Montgomery et. al. (2012). This is a model developed as part of the ICEWS project and was designed by *Strategic Analysis Enterprises*. It is a simple generalized linear model with 27 independent variables.
- GLM The test sample prediction of the GLM model from the insurgency prediction example in Montgomery et. al. (2012). This is a crude logistic model with only four independent variables.
- Insurgency The true values of the dependent variable in the test sample from the insurgency prediction example in Montgomery et. al. (2012). This is a binary variable indicating the actual occurrence of insurgency for each observation in the test sample.

### References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

# Index

## \*Topic **EBMA**

calibrateEnsemble, [2](#)

## \*Topic **calibrate**

calibrateEnsemble, [2](#)

calibrateEnsemble, [2](#)

calibrateEnsemble, ForecastData-method  
(calibrateEnsemble), [2](#)

calibrationSample, [4](#)

compareModels, [5](#)

compareModels, ForecastData-method  
(compareModels), [5](#)

CompareModels-class (compareModels), [5](#)

EBMAforecast, [7](#)

EBMAforecast-package (EBMAforecast), [7](#)

EBMAPredict, [8](#)

EBMAPredict, ForecastData-method  
(EBMAPredict), [8](#)

FDatFitLogit-class (calibrateEnsemble),  
[2](#)

FDatFitLogit-method (EBMAPredict), [8](#)

FDatFitNormal-class  
(calibrateEnsemble), [2](#)

FDatFitNormal-method (EBMAPredict), [8](#)

fitEnsemble (calibrateEnsemble), [2](#)

fitEnsemble, ForecastData-method  
(calibrateEnsemble), [2](#)

fitEnsemble, ForecastDataLogit-method  
(calibrateEnsemble), [2](#)

fitEnsemble, ForecastDataNormal-method  
(calibrateEnsemble), [2](#)

ForecastData-class, [9](#)

ForecastData-generic  
(makeForecastData), [11](#)

ForecastData-method (makeForecastData),  
[11](#)

ForecastDataLogit-class  
(calibrateEnsemble), [2](#)

ForecastDataLogit-method (EBMAPredict),  
[8](#)

ForecastDataNormal-class  
(calibrateEnsemble), [2](#)

makeForecastData, [11](#)

makeForecastData, ANY-method  
(makeForecastData), [11](#)

makeForecastData-method,  
(makeForecastData), [11](#)

plot, FDatFitLogit-method, [13](#)

plot, FDatFitNormal-method  
(plot, FDatFitLogit-method), [13](#)

prediction (EBMAPredict), [8](#)

prediction, (EBMAPredict), [8](#)

prediction, FDatFitLogit-method  
(EBMAPredict), [8](#)

prediction, FDatFitNormal-method  
(EBMAPredict), [8](#)

prediction, ForecastDataNormal-method  
(EBMAPredict), [8](#)

presidentialForecast, [14](#)

print, ForecastData-method  
(makeForecastData), [11](#)

print, SummaryForecastData-method  
(summary, FDatFitLogit-method),  
[15](#)

print-method, (makeForecastData), [11](#)

setModelNames<- (ForecastData-class), [9](#)

setModelNames<- , (makeForecastData), [11](#)

setModelNames<- , ForecastData-method  
(ForecastData-class), [9](#)

setOutcomeCalibration,  
(makeForecastData), [11](#)

setOutcomeCalibration<-  
(ForecastData-class), [9](#)

setOutcomeCalibration<- , ForecastData-method  
(ForecastData-class), [9](#)

setOutcomeTest<- (ForecastData-class), [9](#)

setOutcomeTest<- , (makeForecastData), [11](#)

setOutcomeTest<- , ForecastData-method  
(ForecastData-class), [9](#)

setPredCalibration, (makeForecastData),  
[11](#)

setPredCalibration<-  
(ForecastData-class), [9](#)

setPredCalibration<-, ForecastData-method  
    (ForecastData-class), [9](#)  
setPredTest, (makeForecastData), [11](#)  
setPredTest<- (ForecastData-class), [9](#)  
setPredTest<-, ForecastData-method  
    (ForecastData-class), [9](#)  
show, ForecastData-method  
    (makeForecastData), [11](#)  
show, SummaryForecastData-method  
    (summary, FDatFitLogit-method),  
    [15](#)  
show-method (makeForecastData), [11](#)  
summary, FDatFitLogit-method, [15](#)  
summary, FDatFitNormal-method  
    (summary, FDatFitLogit-method),  
    [15](#)  
SummaryForecastData-class  
    (summary, FDatFitLogit-method),  
    [15](#)  
testSample, [17](#)