Package ‘EBMAforecast’

February 10, 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 stats, graphics, separationplot, plyr, methods, Hmisc, abind
Imports Rcpp (>= 0.11.3)
LinkingTo Rcpp
RoxygenNote 5.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-02-10 09:11:07

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calibrateEnsemble</td>
<td>2</td>
</tr>
<tr>
<td>calibrationSample</td>
<td>4</td>
</tr>
<tr>
<td>compareModels</td>
<td>5</td>
</tr>
<tr>
<td>EBMAforecast</td>
<td>7</td>
</tr>
<tr>
<td>EBMApredict</td>
<td>8</td>
</tr>
<tr>
<td>ForecastData-class</td>
<td>9</td>
</tr>
<tr>
<td>makeForecastData</td>
<td>11</td>
</tr>
<tr>
<td>plot,FDatFitLogit-method</td>
<td>14</td>
</tr>
<tr>
<td>presidentialForecast</td>
<td>15</td>
</tr>
<tr>
<td>summary,FDatFitLogit-method</td>
<td>16</td>
</tr>
<tr>
<td>testSample</td>
<td>18</td>
</tr>
</tbody>
</table>

Index 19
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

```r
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'

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

Arguments

- `.forecastData` An object of class `ForecastData` that will be used to calibrate the model.
- `exp` 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 `exp = 3`.
- `tol` Tolerance for improvements in the log-likelihood before the EM algorithm will stop optimization. The default is `tol = 0.01`, which is somewhat high. Researchers may wish to reduce this by an order of magnitude for final model estimation.
- `maxIter` The maximum number of iterations the EM algorithm will run before stopping automatically. The default is `maxIter = 10000`.
- `model` The model type that should be used given the type of data that is being predicted (i.e., normal, binary, etc.).
- `method` The estimation method used. Currently only implements "EM".
- `...` Not implemented
**calibrateEnsemble**

**useModelParams**  
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


Examples

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

data(testSample)

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

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

# End(Not run)
```

calibrationSample Calibration sample data

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


compareModels

Function for comparing multiple models based on predictive performance

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 (perCorrect), 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

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

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

Value

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

- `fitStatistics` The output of the fit statistics for each model.
- `period` The period, "calibration" or "test", for which the statistics were calculated.
- `threshold` The threshold used to calculate when a "positive" prediction is made by the model.
- `baseModel` 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


See Also

ensembleBMA, other functions
### Examples

```r
## 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)
```

### 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 observation 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


**Examples**

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

### 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

```r
EBMApredict(EBMAmodel, Predictions, Outcome = NULL, ...)
```

## S4 method for signature 'ForecastData'

```r
EBMApredict(EBMAmodel, Predictions, Outcome = NULL, ...)
prediction(EBMAmodel, Predictions, Outcome, ...)
```

## S4 method for signature 'FDatFitLogit'

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

## S4 method for signature 'FDatFitNormal'

```r
prediction(EBMAmodel, Predictions, Outcome = c(), ...)
```
ForecastData-class

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.
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.

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


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

```r
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

See Also
ensembleBMA

Examples
## Not run: data(calibrationSample)

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

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

### to assign individual slots, use set functions
setPredCalibration(thisForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(thisForecastData)<-calibrationSample[,"Insurgency"]
setPredTest(thisForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(thisForecastData)<-testSample[,"Insurgency"]
setModelNames(thisForecastData)<-c("LMER", "SAE", "GLM")

## End(Not run)
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

.predCalibration
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.

.predTest
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.

.outcomeCalibration
A vector with the true values of the dependent variable for each observation in the calibration period.

.outcomeTest
A vector with the true values of the dependent variable for each observation in the test period.

.modelNames
A vector of length p with the names of the component models.

...  
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:

x  
A data object of class 'ForecastData'

print let's the use specify the number of digits printed and takes the arguments:

digits
User specified number of digits to be displayed.

object
A data object of class 'ForecastData'

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.
makeForecastData

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.

modelNames

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


See Also

ensembleBMA

Examples

data(calibrationSample)

## Not run: data(testSample)
thisForecastData <- 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 access individual slots in the ForecastData object
getPredCalibration(thisForecastData)
getOutcomeCalibration(thisForecastData)
getPredTest(thisForecastData)
getOutcomeTest(thisForecastData)
getModelNames(thisForecastData)

### to assign individual slots, use set functions
setPredCalibration(thisForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(thisForecastData)<-calibrationSample[,"Insurgency"]
setPredTest(thisForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(thisForecastData)<-testSample[,"Insurgency"]
setModelNames(thisForecastData)<-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

```r
## 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


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).
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".

References


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 compareModels 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

```r
# 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)

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

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

```
# End(Not run)
```
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
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), 16
show-method (makeForecastData), 11
summary,FDatFitLogit-method, 16
summary,FDatFitNormal-method
(summary,FDatFitLogit-method), 16
SummaryForecastData-class
(summary,FDatFitLogit-method), 16

testSample, 18