Package ‘gamclass’
August 20, 2015

Type Package

Title Functions and Data for a Course on Modern Regression and Classification

Version 0.56

Date 2015-07-29

Author John Maindonald

Maintainer John Maindonald <john.maindonald@anu.edu.au>

LazyData true

Depends R (>= 3.0.0)

Suggests leaps, quantreg, sp, diagram, oz, forecast, SMIR, kernlab, Ecdat, mlbench, DAAGbio, knitr

Imports car, mgcv, DAAG, MASS, rpart, randomForest, lattice, latticeExtra, ape, KernSmooth, methods

VignetteBuilder knitr

Description Functions and data are provided that support a course that emphasizes statistical issues of inference and generalizability. Attention is restricted to a relatively small number of methods, often (misleadingly in my view) referred to as algorithms.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-20 10:39:49

R topics documented:

modregR-package ......................................................... 2
addhlines ................................................................. 3
airAccs ................................................................. 4
bssBYcut ............................................................... 5
compareModels ......................................................... 6
cvalues ................................................................. 7
CVcluster ............................................................... 8
Description

For purposes of this package, modern regression extends to include classification and multivariate exploration.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>modregR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.5</td>
</tr>
<tr>
<td>Date</td>
<td>2011-12-12</td>
</tr>
<tr>
<td>License</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

Functions are mostly designed to facilitate various cross-validation and bootstrap calculations.

Author(s)

John Maindonald
Maintainer: john.maindonald@anu.edu.au

References

addhlines

Add horizontal lines to plot.

Description

This is designed for adding horizontal lines that show predicted values to a plot of observed values versus x-values, in rpart regression. Where predicted values change between two successive x-values lines are extended to the midway point. This reflects the way that predict.rpart handles predictions for new data.

Usage

addhlines(x, y, ...)

Arguments

x Vector of predictor variable values.
y Vector of predicted values.
... Additional graphics parameters, for passing through to the lines() function.

Value

Lines are added to the current graph.

Author(s)

John Maindonald

Examples

x <- c(34, 18, 45, 18, 27, 24, 34, 20, 24, 28, 21, 18)
y <- c(14, 11, 12, 9, 4, 11, 6, 9, 4, 10, 9, 2)
hat <- c(10.5, 7.75, 10.5, 7.75, 7, 7, 10.5, 7.75, 7, 10.5, 7, 7.75)
plot(x, y)
addhlines(x, hat, lwd=2, col=“gray”)

## The function is currently defined as
function(x,y,...){
  ordx <- order(x)
xo <- x[ordx]
yo <- y[ordx]
b breaks <- diff(yo)!=0
xh <- c(xo[1],0.5*(xo[c(FALSE,breaks)]+xo[c(breaks, FALSE)]))
yh <- yo[c(TRUE, breaks)]
y3 <- x3 <- numeric(3*length(xh)-1)
loc1 <- seq(from=1, to=length(x3), by=3)
x3[loc1] <- xh
x3[loc1+1]<- c(xh[-1], max(x))}
Description

Aircraft Crash Data

Usage

data(airAccs)

Format

A data frame with 5666 observations on the following 7 variables.

Date  Date of Accident
location  Location of accident
operator  Aircraft operator
planeType  Aircraft type
Dead  Number of deaths
Aboard  Number aboard
Ground  Deaths on ground

Details

For details of inclusion criteria, see http://www.planecrashinfo.com/database.htm

Source

http://www.planecrashinfo.com/database.htm

References

http://www.planecrashinfo.com/reference.htm

Examples

data(airAccs)
str(airAccs)
bssBYcut

Between group SS for y, for all possible splits on values of x

Description
Each point of separation between successive values of x is used in turn to create two groups of observations. The between group sum of squares for y is calculated for each such split.

Usage
bssBYcut(x, y, data)

Arguments
x Variable (numeric) used to define splits. Observations with x values less than the cut point go into the first group, while those with values >= the cut point go into the second group.
y Variable for which BSS values are to be calculated.
data Data frame with columns x and y.

Value
Data frame with columns:
x0rd Cut points for splits.
comp2 Between groups sum of squares

Author(s)
J H Maindonald

Examples
xy <- bssBYcut(weight, height, women)
with(xy, xy[which.max(bss)])

## The function is currently defined as
function (x, y, data)
{
xnam <- deparse(substitute(x))
ynam <- deparse(substitute(y))
xv <- data[, xnam]
yv <- data[, ynam]
sumss <- function(x, y, cut) {
  av <- mean(y)
  left <- x < cut
  sum(left) * (mean(y[left]) - av)^2 + sum(!left) * (mean(y[!left]) - 
  av)^2
}

...
compareModels

Compare accuracy of alternative classification methods

Description

Compare, between models, probabilities that the models assign to membership in the correct group or class. Probabilites should be estimated from cross-validation or from bootstrap out-of-bag data or preferably for test data that are completely separate from the data used to derive the model.

Usage

```r
compareModels(groups, estprobs = list(lda = NULL, rf = NULL),
              gpnames = NULL, robust = TRUE, print = TRUE)
```

Arguments

- `groups` Factor that specifies the groups
- `estprobs` List whose elements (with names that identify the models) are matrices that give for each observation (row) estimated probabilities of membership for each of the groups (columns).
- `gpnames` Character: names for groups, if different from `levels(groups)`
- `robust` Logical, TRUE or FALSE
- `print` Logical. Should results be printed?

Details

The estimated probabilities are compared directly, under normal distribution assumptions. An effect is fitted for each observation, plus an effect for the method. Comparison on a logit scale may sometimes be preferable. An option to allow this is scheduled for incorporation in a later version.

Value

- `modelAVS` Average accuracies for models
- `modelSE` Approximate average SE for comparing models
- `gpAVS` Average accuracies for groups
- `gpSE` Approximate average SE for comparing groups
- `obsEff` Effects assigned to individual observations
Note

The analysis estimates effects due to model and group (gp), after accounting for differences between observations.

Author(s)

John Maindonald

Examples

library(MASS)
library(DAAG)
library(randomForest)
ldaihat <- lda(species ~ length+breadth, data=cuckoos, CV=TRUE)$posterior
qdahat <- qda(species ~ length+breadth, data=cuckoos, CV=TRUE)$posterior
rfhat <- predict(randomForest(species ~ length+breadth, data=cuckoos), type="prob")
compareModels(groups=cuckoos$species, estprobs=list(lda=ldaihat,
      qda=qdahat, rf=rfhat), robust=FALSE)

cvalues

Historical speed of light measurements

Description

Measurements made between 1675 and 1972

Usage

cvalues

Format

A data frame with 9 observations on the following 3 variables.

<table>
<thead>
<tr>
<th>Year</th>
<th>Year of measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td>estimated speed in meters per second</td>
</tr>
<tr>
<td>error</td>
<td>measurement error, as estimated by experimenter(s)</td>
</tr>
</tbody>
</table>

Source


Examples

data(cvalues)
CVcluster: Cross-validation estimate of predictive accuracy for clustered data

Description
This function adapts cross-validation to work with clustered categorical outcome data. For example, there may be multiple observations on individuals (clusters). It requires a fitting function that accepts a model formula.

Usage
CVcluster(formula, id, data, na.action=na.omit, nfold = 15, FUN = lda, predictFUN=function(x, newdata, ...)predict(x, newdata, ...)$class, printit = TRUE, cvparts = NULL, seed = 29)

Arguments
- formula: Model formula
- id: numeric, identifies clusters
- data: data frame that supplies the data
- na.action: na.fail (default) or na.omit
- nfold: Number of cross-validation folds
- FUN: function that fits the model
- predictFUN: function that gives predicted values
- printit: Should summary information be printed?
- cvparts: Use, if required, to specify the precise folds used for the cross-validation. The comparison between different models will be more accurate if the same folds are used.
- seed: Set seed, if required, so that results are exactly reproducible

Value
- class: Predicted values from cross-validation
- CVaccuracy: Cross-validation estimate of accuracy
- confusion: Confusion matrix

Author(s)
John Maindonald

References
CVgam

Cross-validation estimate of accuracy from GAM model fit

Description

The cross-validation estimate of accuracy is sufficiently independent of the available model fitting criteria (including Generalized Cross-validation) that it provides a useful check on the extent of downward bias in the estimated standard error of residual.

Usage

CVgam(formula, data, nfold = 10, debug.level = 0, method = "GCV.Cp", printit = TRUE, cvparts = NULL, gamma = 1, seed = 29)

Arguments

- **formula**: Model formula, for passing to the gam() function
- **data**: data frame that supplies the data
- **nfold**: Number of cross-validation folds
- **debug.level**: See gam for details
- **method**: Fit method for GAM model. See gam for details
- **printit**: Should summary information be printed?
- **cvparts**: Use, if required, to specify the precise folds used for the cross-validation. The comparison between different models will be more accurate if the same folds are used.
- **gamma**: See gam for details.
- **seed**: Set seed, if required, so that results are exactly reproducible

Value

- **fitted**: fitted values
- **resid**: residuals
- **cvscale**: scale parameter from cross-validation
- **scale.gam**: scale parameter from function gam

The scale parameter from cross-validation is the error mean square

Examples

```r
if(require(mlbench)&require(MASS)){
  data(Vowel)
  acc <- CVcluster(formula=Class ~ ., id = V1, data = Vowel, nfold = 15, FUN = lda, 
      predictFUN=function(x, newdata, ...)predict(x, newdata, ...)$class, 
      printit = TRUE, cvparts = NULL, seed = 29)
}
```
eventCounts

Author(s)
John Maindonald

References

Examples
if(require(sp)){
  library(mgcv)
  data(meuse)
  meuse$ffreq <- factor(meuse$ffreq)
  CVgam(formula=log(zinc)~s(elev) + s(dist) + ffreq + soil,
        data = meuse, nfold = 10, debug.level = 0, method = "GCV.Cp",
        printit = TRUE, cvparts = NULL, gamma = 1, seed = 29)
}

eventCounts(data, dateCol="Date", from = NULL, to = NULL,
            by = "1 month", categoryCol=NULL, takeOnly=NULL, prefix="n_")

Arguments
- data: Data frame that should include any columns whose names appear in other function arguments.
- dateCol: Name of column that holds vector of dates
- from: Starting date. If NULL set to first date given. If supplied, any rows earlier than from will be omitted. Similarly, rows later than any supplied date to will be omitted.
- to: Final date, for which numbers of events are to be tallied. If NULL set to final date given.
- by: Time event to be used; e.g. "1 day", or "1 week", or "4 weeks", or "1 month", or "1 quarter", or "1 year", or "10 years".
- categoryCol: If not NULL create one column of counts for each level (or if not a factor, unique value).

Description
For example, dates may be dates of plane crashes. For purposes of analysis, this function tabulates number of crash events per event of time, for each successive specified event.
takeOnly If not NULL, a charater string that when deparsed and executed will return a vector of logicals.

prefix If categoryCol is not NULL, a prefix for the names of the columns of counts. Otherwise (categoryCol=NULL) a name for the column of counts.

Value
A data frame, with columns Date (the first day of the event for which events are given), and other column(s) that hols counts of events.

Author(s)
John Maindonald

See Also
cut

Examples

```r
crashDate <- as.Date(c("1908-09-17","1912-07-12","1913-08-06",
"1913-09-09","1913-10-17"))
df <- data.frame(date=crashDate)
byYears <- eventCounts(data=df, dateCol="date",
from=as.Date("1908-01-01"),
by="1 year")
```

Description
Data are from the US FARS (Fatality Analysis Recording System) archive that is intended to include every accident in which there was at least one fatality. Data are limited to vehicles where the front seat passenger seat was occupied.

Usage
FARS

Format
A data frame with 153338 observations on the following 17 variables.

caseid a character vector: identifies the vehicle
state a numeric vector. See the FARS website for details
age a numeric vector; 998=not reported; 999=not known
airbag a numeric vector
injury  a numeric vector  
restraint a numeric vector  
sex  1= male, 2= female, 9= unknown  
inimpact  a numeric vector  
modelyr  a numeric vector  
airbagAvail  a factor with levels no yes NA-code  
airbagDeploy  a factor with levels no yes NA-code  
Restraint  a factor with levels no yes NA-code  
D_injury  a numeric vector  
D_airbagAvail  a factor with levels no yes NA-code  
D_airbagDeploy  a factor with levels no yes NA-code  
D_Restraint  a factor with levels no yes NA-code  
year  year of accident  

Details  
Data is for automabiles where the right passenger seat was occupied, with one observation for each such passenger. Observations for vehicles where the most harmful event was a fire or explosion or immersion or gas inhalation, or where someone fell or jumped from the vehicle, are omitted. Data are limited to vehicle body types 1 to 19, 48, 49, 61, or 62. This excludes large trucks, pickup trucks, vans and buses. The 2009 and 2010 data does not include information on whether airbags were installed.

Note  
The papers given as references demonstrate the use of Fatal Accident Recording System data to assess the effectiveness of airbags (even differences between different types of airbags) and seatbelts. Useful results can be obtained by matching driver mortality, with and without airbags, to mortality rates for right front seat passengers in cars without passenger airbags.

Source  

References  
Cummings, P; McKnight, B, 2010. Accounting for vehicle, crash, and occupant characteristics in traffic crash studies. Injury Prevention 16: 363-366

Examples  
data(FARS)
US fatal road accident data, 2007 and 2008

Description

Data are included on variables that may be relevant to assessing airbag and seatbelt effectiveness in preventing fatal injury.

Usage

fars2007
fars2008

Format

A data frame with 72548 observations on the following 24 variables.

Obs.  a numeric vector
state  a numeric vector
casenum  a numeric vector
vnum  a numeric vector
pnum  a numeric vector
lightcond  a numeric vector
numfatal  a numeric vector
vforms  a numeric vector
age  a numeric vector
airbag  a numeric vector
injury  a numeric vector
ptype  a numeric vector
restraint  a numeric vector
seatpos  a numeric vector
sex  a numeric vector
body  a numeric vector
inimpact  a numeric vector
mhevent  a numeric vector
vfatcount  a numeric vector
numocc  a numeric vector
travspd  a numeric vector
make  a numeric vector
model  a numeric vector
modelyr  a numeric vector
Details

Data is for automobiles where a passenger seat was occupied, with one observation for each such passenger.

Source


References


Cummings, P; McKnight, B, 2010. Accounting for vehicle, crash, and occupant characteristics in traffic crash studies. Injury Prevention 16: 363-366


Examples

data(fars2007)
str(fars2007)

---

FARSmiss

Summary information on records omitted from the FARS dataset

Description

Data are a 3-way table, indexed by state, a set of variable names, and years

Usage

FARSmiss

Format

The format is: num [1:51, 1:7, 1:13] 2 0 16 0 75 1 5 0 5 5 ... - attr(*, "dimnames")=List of 3 ..$ : chr [1:51] "1" "2" "3" "4" ... ..$ : chr [1:7] "injury" "age" "airbag" "restraint" ... ..$ : chr [1:13] "1998" "1999" "2000" "2001" ...

Details

These data were generated using the function matchedPairs, using as input data downloaded from the URL given as source. Data for the years 2007 and 2008 are included with this package, and can be used to generate the result of restricting FARS and FARSmiss to those years. The check columns (all values should be zero) nomatch and dups have been omitted from the second dimension of the array.
gamRF

Source


References


See Also

matchedPairs

Examples

data(FARSmiss)
str(FARSmiss)

---

gamRF Random forest fit to residuals from GAM model

Description

Fit model using gam() from mgcv, then use random forest regression with residuals. Check performance of this hybrid model for predictions to newdata, if supplied.

Usage

gamRF(formlist, yvar, data, newdata = NULL, rfVars, method = "GCV.Cp",
printit = TRUE, seed = NULL)

Arguments

- **formlist**: List of right hand sides of formulae for GAM models.
- **yvar**: Character string holding y-variable name.
- **data**: Data
- **newdata**: Optionally, supply test data.
- **rfVars**: Names of explanatory variables for the randomForest model.
- **method**: Smoothing parameter estimation method for use of gam(). See `gam`.
- **printit**: Should a summary of results (error rates) be printed?
- **seed**: Set a seed to make result repeatable.

Value

A vector of test data accuracies for the hybrid models (one for each element of formlist), plus test error mean square and OOB error mean square for the use of randomForest().
Note
The best results are typically obtained when a relatively low degree of freedom GAM model is used. It seems advisable to use those variables for the GAM fit that seem likely to be similar in their effect irrespective of geographic location.

Author(s)
John Maindonald <john.maindonald@anu.edu.au>

References

See Also
CVgam

Examples
if(length(find.package("sp", quiet=TRUE))>0){
  data("meuse", package="sp")
  meuse <- within(meuse, {levels(soil) <- c("1","2","3")
                            ffreq <- as.numeric(ffreq)
                            loglead <- log(lead)})
  form <- ~ dist + elev + ffreq + soil
  rfVars <- c("dist", "elev", "soil", "ffreq", "x", "y")
  ## Select 90 out of 155 rows
  sub <- sample(1:nrow(meuse), 90)
  meuseOut <- meuse[-sub,]
  meuseIn <- meuse[sub,]
  gamRF(formlist=list("lm=form"), yvar="loglead", rfVars=rfVars,
           data=meuseIn, newdata=meuseOut)
}

## The function is currently defined as
function (formlist, yvar, data, newdata = NULL, rfVars = NULL, method = "GCV.Cp",
         printit = TRUE, seed = NULL)
{
  if(is.null(seed)) set.seed(seed)
  errRate <- numeric(length(formlist)+2)
  names(errRate) <- c(names(formlist), "rfTest", "rfOOB")
  ytrain <- data[, yvar]
  xtrain <- data[, rfVars]
  xtest <- newdata[, rfVars]
  ytest = newdata[, yvar]
  res.rf <- randomForest(x = xtrain, y = ytrain,
                         xtest=xtest,
                         ytest=ytest)
  errRate["rfOOB"] <- mean(res.rf$mse)
german $\text{German credit scoring data}$

**Description**

See website for details of data attributes

**Usage**

`german`

**Format**

A data frame with 1000 observations on the following 21 variables.

- **V1**: a factor with levels A11 A12 A13 A14
- **V2**: a numeric vector
- **V3**: a factor with levels A30 A31 A32 A33 A34
- **V4**: a factor with levels A40 A41 A410 A42 A43 A44 A45 A46 A48 A49
- **V5**: a numeric vector
- **V6**: a factor with levels A61 A62 A63 A64 A65
- **V7**: a factor with levels A71 A72 A73 A74 A75
- **V8**: a numeric vector
- **V9**: a factor with levels A91 A92 A93 A94

```r
errRate["rfTest"] <- mean(res.rf$test$ mse)
GAMhat <- numeric(nrow(data))
for (nam in names(formlist)){
  form <- as.formula(paste(c(yvar, paste(formlist[[nam]])), collapse = " "))
  train.gam <- gam(form, data = data, method = method)
  res <- resid(train.gam)
  cvGAMms <- sum(res^2)/length(res)
  if (!all(rfVars %in% names(newdata))){
    missNam <- rfVars[!rfVars %in% names(newdata)]
    stop(paste("The following were not found in 'newdata': ",
               paste(missNam, collapse = ", ")))
  }
  GAMtesthat <- predict(train.gam, newdata = newdata)
  GAMtestres <- ytest - GAMtesthat
  Gres.rf <- randomForest(x = xtrain, y = res, xtest = xtest,
                          ytest = GAMtestres)
  errRate[nam] <- mean(Gres.rf$test$ mse)
} if (printit)
  print(round(errRate, 4))
  invisible(errRate)
```
V10  a factor with levels A101 A102 A103
V11  a numeric vector
V12  a factor with levels A121 A122 A123 A124
V13  a numeric vector
V14  a factor with levels A141 A142 A143
V15  a factor with levels A151 A152 A153
V16  a numeric vector
V17  a factor with levels A171 A172 A173 A174
V18  a factor with levels good bad
V19  a factor with levels A191 A192
V20  a factor with levels A201 A202
V21  a numeric vector

Source


Examples

data(german)

loti

Global temperature anomalies

Description

Anomalies, for the years 1880 to 2010, from the 1951 - 1980 average. These are the GISS (Goddard Institute for Space Studies) Land-Ocean Temperature Index (LOTI) data

Usage

loti

Format

A data frame with 131 observations on the following 19 variables.

Jan  a numeric vector
Feb  a numeric vector
Mar  a numeric vector
Apr  a numeric vector
May  a numeric vector
Jun  a numeric vector
matchedPairs

July a numeric vector
August a numeric vector
September a numeric vector
October a numeric vector
November a numeric vector
December a numeric vector
J.D Jan-Dec averages
D.N Dec-Nov averages
DJF Dec-Jan-Feb averages
MAM Mar-Apr-May
JJA Jun-Jul-Aug
SON Sept-Oct-Nov
Year a numeric vector

Source

http://data.giss.nasa.gov/gistemp/tabledata/GLB.Ts+dSST.txt

Examples

data(loti)

matchedPairs

Collect together driver and right seat passenger information, for the
specified year

Description

This function collates the information needed for a matched pairs analysis. Driver mortalities, with
and without airbags, are matched to passenger mortalities for right front seat passengers in cars
without passenger airbags. It was used to generate the FARS and FARSmiss datasets

Usage

matchedPairs(years = 2007:2008, prefix = "fars", compareByVar = c("airbagAvail",
"airbagDeploy", "Restraint"),
bycat = list(airbagAvail = list(yes = c(1:9, 20, 28:29, 31:32), no = 30,
leaveout = c(0, 98, 99)),
airbagDeploy = list(yes = c(1:9),
no = c(20, 28, 30:32), leaveout = c(0, 29, 98, 99)),
Restraint = list(yes = c(1:4, 8, 10:12, 97), no = c(0, 5, 6, 7, 13:17),
leaveout = (98:99)),
restrict = "body%in%c(1:19,48,49,61,62)&!(mhevent%in%c(2:5))",
restrictvars = c("body", "mhevent", "seatpos", "injury"),
retain = c("state", "age", "airbag", "injury", "restraint",
"sex", "inimpact", "modelyr"), progress = TRUE)
Arguments

- **years**: Years for which data is required
- **prefix**: Prefix for file name.
- **compareBYvar**: Variables to be included in output, selected from `airbagAvail`, `airbagDeploy` and `Restraint`), for which deaths are to be compared between drivers (w/wo
- **bycat**: Maps airbag and restraint codes to yes, no or leaveout
- **restrict**: Allows restriction of data to specified variable subsets.
- **restrictvars**: character vector: names of variables that appear in the `restrict` argument
- **retain**: Retain these columns in the output data
- **progress**: Print year by year details of the progress of calculations.

Details

This function is designed for processing data obtained from the FARS url noted under references. This function was used to generate the data in the `FARS` data frame. Two of the datasets from which the FARS dataset was generated are included with this package – these are `fars2007` and `fars2008`

Value

- **data**: Data frame, with driver information matched against passenger information for the same vehicle
- **miss**: 3-way table holding missing data information. The table is has margins state, a set of variable names, and years

Author(s)

John Maindonald

References


See Also

- plotFars, FARS, fars2007

Examples

```r
farsMatch0708 <- matchedPairs(years=2007:2008)
```
plotFars

Extract from FARS data set the ratio of ratios estimate of safety device effectiveness, and return trellis graphics object

Description
Safety devices may be airbags or seatbelts. For airbags, alternatives are to use ‘airbag installed’ or ‘airbag deployed’ as the criterion. Ratio of driver deaths to passenger deaths are calculated for driver with device and for driver without device, in both cases for passenger without device.

Usage
plotFars(restrict = 
"age>=16&age<998&inimpact%in%c(11,12,1)", fatal = 4, statistics = 
c("airbagAvail", "airbagDeploy", "Restraint"))

Arguments
\begin{itemize}
\item \texttt{restrict} text: an expression that restricts observations considered
\item \texttt{fatal} numeric: 4 for fatal injury, or \texttt{c(3, 4)} for incapacitating or fatal injury
\item \texttt{statistics} Vector of character: ratio of ratios variables that will be plotted
\end{itemize}

Details
Note that the ‘airbag deployed’ statistic is not a useful measure of airbag effectiveness. At its most effective, the airbag will deploy only when the accident is sufficiently serious that deployment will reduce the risk of serious injury and/or accident. The with/without deployment comparison compares, in part, serious accidents with less serious accidents.

Value
A graphics object is returned

Author(s)
John Maindonald

See Also
matchedPairs

Examples
\begin{verbatim}
## Not run:
gphFars <- plotFars()
## End(Not run)
\end{verbatim}
Random forests estimate of predictive accuracy for clustered data

Description

This function adapts random forests to work (albeit clumsily and inefficiently) with clustered categorical outcome data. For example, there may be multiple observations on individuals (clusters). Predictions are made for the OOB (out of bag) clusters.

Usage

RFcluster(formula, id, data, nfold = 15, ntree = 500, progress = TRUE, printit = TRUE, seed = 29)

Arguments

- formula: Model formula
- id: numeric, identifies clusters
- data: data frame that supplies the data
- nfold: numeric, number of folds
- ntree: numeric, number of trees (number of bootstrap samples)
- progress: Print information on progress of calculations
- printit: Print summary information on accuracy
- seed: Set seed, if required, so that results are exactly reproducible

Details

Bootstrap samples are taken of observations in the in-bag clusters. Predictions are made for all observations in the OOB clusters.

Value

- class: Predicted values from cross-validation
- OOBaccuracy: Cross-validation estimate of accuracy
- confusion: Confusion matrix

Author(s)

John Maindonald

References

Examples

```r
## Not run:
library(mlbench)
library(randomForest)
data(Vowel)
RFcluster(formula=Class ~ ., id = V1, data = Vowel, nfold = 15,
          ntree=500, progress=TRUE, printit = TRUE, seed = 29)
```

## End(Not run)

---

**simreg**

*Simulate (repeated) regression calculations*

**Description**

Derive parameter estimates and standard errors by simulation, or by bootstrap resampling.

**Usage**

```r
simreg(formula, data, nsim = 1000)
bootreg(formula, data, nboot = 1000)
```

**Arguments**

- `formula`: Model formula
- `data`: Data frame from which names in formula can be taken
- `nsim`: Number of repeats of the simulation (`simreg`)
- `nboot`: Number of bootstrap resamples (`bootreg`)

**Value**

Matrix of coefficients from repeated simulations, or from bootstrap resamples. For `simreg` there is one row for each repeat of the simulation. For `bootreg` there is one row for each resample.

**Note**

Note that `bootreg` uses the simplest possible form of bootstrap. For any except very large datasets, standard errors may be substantial under-estimates.

**Author(s)**

John Maindonald

**References**

Examples

```r
xy <- data.frame(x=rnorm(100), y=rnorm(100))
simcoef <- simreg(formula = y~x, data = xy, nsim = 100)
bootcoef <- bootreg(formula = y~x, data = xy, nboot = 100)
```

---

### Description

Safety devices may be airbags or seatbelts. For airbags, alternatives are to use `‘airbag installed’` or `‘airbag deployed’` as the criterion. Ratio of driver deaths to passenger deaths are calculated for driver with device and for driver without device, in both cases for passenger without device.

### Usage

```r
tabFarsDead(restrict = "age>=16&age<998&inimpact%in%c(11,12,1)", fatal = 4, statistics = c("airbagAvail", "airbagDeploy", "RestRAINT"))
```

### Arguments

- **restrict**
  - text: an expression that restricts observations considered
- **fatal**
  - numeric: 4 for fatal injury, or c(3,4) for incapacitating or fatal injury
- **statistics**
  - Vector of character: ratio of ratios variables that will be plotted

### Details

Note that the `‘airbag deployed’` statistic is not a useful measure of airbag effectiveness. At its most effective, the airbag will deploy only when the accident is sufficiently serious that deployment will reduce the risk of serious injury and/or accident. The with/without deployment comparison compares, in part, serious accidents with less serious accidents.

### Value

A list with elements

- **airbagAvail**
  - a multiway table with margins yrs, airbagAvail, and a third margin with levels P_injury, D_injury, tot, and prop
- **airbagDeploy**
  - a multiway table with margins yrs, airbagDeploy, and a third margin with levels P_injury, D_injury, tot, and prop
- **RestRAINT**
  - a multiway table with margins yrs, RestRAINT, and a third margin injury with levels P_injury, D_injury, tot, and prop
Author(s)

John Maindonald

See Also

matchedPairs

Examples

tabDeaths <- tabFarsDead()
Index

*Topic chron
  eventCounts, 10

*Topic datasets
  airAccs, 4
  cvalues, 7
  FARS, 11
  fars2007, 13
  FARSmiss, 14
  german, 17
  loti, 18

*Topic graphics
  addhlines, 3
  plotFars, 21

*Topic manip
  bssBYcut, 5
  eventCounts, 10
  matchedPairs, 19
  tabFarsDead, 24

*Topic models
  CVcluster, 8
  CVgam, 9
  gamRF, 15
  RFcluster, 22
  simreg, 23

*Topic multivariate
  compareModels, 6

*Topic package
  modregR-package, 2

*Topic regression
  CVcluster, 8
  CVgam, 9
  gamRF, 15
  RFcluster, 22
  simreg, 23

*Topic statistics
  compareModels, 6

addhlines, 3
airAccs, 4

bootreg(simreg), 23
bssBYcut, 5
compareModels, 6
cut, 11
cvalues, 7
CVcluster, 8
CVgam, 9, 16
eventCounts, 10
FARS, 11, 14, 19, 20
fars2007, 13
fars2008(fars2007), 13
FARSmiss, 14, 19
gam, 9, 15
gamRF, 15
german, 17
loti, 18
matchedPairs, 14, 15, 19, 21, 25
modregR(modregR-package), 2
modregR-package, 2
plotFars, 21
predict.rpart, 3
RFcluster, 22
rpart, 3
simreg, 23
tabFarsDead, 24