

Package ‘logistf’

July 2, 2014

Type Package

Title Firth's bias reduced logistic regression

Version 1.21

Date 2013-06-26

Author Georg Heinze <georg.heinze@meduniwien.ac.at>, Meinhard Ploner, Daniela Dunkler (former versions), Harry Southworth (former versions)

Maintainer Georg Heinze <georg.heinze@meduniwien.ac.at>

Depends R (>= 3.0.0), mice, mgcv

Description Firth's bias reduced logistic regression approach with penalized profile likelihood based confidence intervals for parameter estimates.

License GPL

URL <http://cemsiiis.meduniwien.ac.at/en/kb/science-research/software/statistical-software/fllogistf/>

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-06-26 18:31:49

R topics documented:

logistf-package	2
add1.logistf	5
anova.logistf	6
backward	7
CLIP.confint	9
CLIP.profile	11

is.logistf	13
logistf	14
logistf.control	17
logistftest	18
logistpl.control	20
plot.logistf.profile	21
pool.RR	23
print.logistf	25
print.logistftest	26
profile.logistf	27
PVR.confint	28
sex2	30
summary.logistf	31
vcov.logistf	31

Index	33
--------------	-----------

logistf-package	<i>An R package for Firth-type bias-reduced logistic regression</i>
-----------------	---

Description

Implements Firth's penalized likelihood logistic regression

Details

Package:	logistf
Type:	Package
Version:	1.20
Date:	2013-05-15
License:	GPL
LazyLoad:	yes

The package `logistf` provides a comprehensive tool to facilitate the application of Firth's modified score procedure in logistic regression analysis. It was written on a PC with S-PLUS 4.0, later translated to S-PLUS 6.0, and to R.

Version 1.10 improves on previous versions by the possibility to include case weights and offsets, and better control of the iterative fitting algorithm.

Version 1.20 provides a major update in many respects:

- (1) Many `S3Methods` have been defined for objects of type `logistf`, including `add1`, `drop1` and `anova` methods
- (2) New forward and backward functions allow for automated variable selection using penalized likelihood ratio tests
- (3) The core routines have been transferred to C code, and many improvements for speed have been done

(4) Handling of multiple imputed data sets: the 'combination of likelihood profiles' (CLIP) method has been implemented, which builds on datasets that were imputed by the package mice, but can also handle any imputed data.

The call of the main function of the library follows the structure of the standard functions as `lm` or `glm`, requiring a `data.frame` and a formula for the model specification. The resulting object belongs to the new class `logistf`, which includes penalized maximum likelihood ('Firth-Logistic'- or 'FL'-type) logistic regression parameters, standard errors, confidence limits, p-values, the value of the maximized penalized log likelihood, the linear predictors, the number of iterations needed to arrive at the maximum and much more. Furthermore, specific methods for the resulting object are supplied. Additionally, a function to plot profiles of the penalized likelihood function and a function to perform penalized likelihood ratio tests have been included.

In explaining the details of the estimation process we follow mainly the description in Heinze & Ploner (2003). In general, maximum likelihood estimates are often prone to small sample bias. To reduce this bias, Firth (1993) suggested to maximize the penalized log likelihood $\log L(\beta)^* = \log L(\beta) + 1/2 \log |I(\beta)|$, where $I(\beta)$ is the Fisher information matrix, i. e. minus the second derivative of the log likelihood. Applying this idea to logistic regression, the score function $U(\beta)$ is replaced by the modified score function $U(\beta)^* = U(\beta) + a$, where a has r th entry $a_r = 0.5 \text{tr} I(\beta)^{-1} [dI(\beta)/d\beta_r]$, $r = 1, \dots, k$. Heinze and Schemper (2002) give the explicit formulae for $I(\beta)$ and $I(\beta)/d\beta_r$.

In our programs estimation of β is based on a Newton-Raphson algorithm. Parameter values are initialized usually with 0, but in general the user can specify arbitrary starting values.

With a starting value of $\beta^{(0)}$, the penalized maximum likelihood estimate β is obtained iteratively:

$$\beta^{(s+1)} = \beta^{(s)} + I(\beta^{(s)})^{-1} U(\beta^{(s)})^*$$

If the penalized log likelihood evaluated at $\beta^{(s+1)}$ is less than that evaluated at $\beta^{(s)}$, then $\beta^{(s+1)}$ is recomputed by step-halving. For each entry r of β with $r = 1, \dots, k$ the absolute step size $|\beta_r^{(s+1)} - \beta_r^s|$ is restricted to a maximal allowed value `maxstep`. These two means should avoid numerical problems during estimation. The iterative process is continued until the parameter estimates converge, i. e., until three criteria are met: the change in log likelihood is less than `lconv`, the maximum absolute element of the score vector is less than `gconv`, the maximum absolute change in beta is less than `xconv`. `lconv`, `gconv`, `xconv` can be controlled by `control=logistf.control(lconv=..., gconv=..., xconv=...)`.

Computation of profile penalized likelihood confidence intervals for parameters (`logistpl`) follows the algorithm of Venzon and Moolgavkar (1988). For testing the hypothesis of $\gamma = \gamma_0$, let the likelihood ratio statistic

$$LR = 2[\log L(\gamma, \delta) - \log L(\gamma_0, \delta_{\gamma_0})^*]$$

where (γ, δ) is the joint penalized maximum likelihood estimate of $\beta = (\gamma, \delta)$, and δ_{γ_0} is the penalized maximum likelihood estimate of δ when $\gamma = \gamma_0$. The profile penalized likelihood confidence interval is the continuous set of values γ_0 for which LR does not exceed the $(1 - \alpha)100$ th percentile of the χ_1^2 -distribution. The confidence limits can therefore be found iteratively by approximating the penalized log likelihood function in a neighborhood of β by the quadratic function

$$l(\beta + \delta) = l(\beta) + \delta' U^* - 0.5 \delta' I \delta$$

where $U^* = U(\beta)^*$ and $-I = -I(\beta)$.

In some situations computation of profile penalized likelihood confidence intervals may be time consuming since the iterative procedure outlined above has to be repeated for the lower and for the upper confidence limits of each of the k parameters. In other problems one may not be interested in interval estimation, anyway. In such cases, the user can request computation of Wald confidence intervals and P-values, which are based on the normal approximation of the parameter estimates and do not need any iterative estimation process. Standard errors $\sigma_r, r = 1, \dots, k$, of the parameter estimates are computed as the roots of the diagonal elements of the variance matrix $V(\beta) = I(\beta)^{-1}$. A $100(1 - \alpha)$ per cent Wald confidence interval for parameter β_r is then defined as $[\beta_r + \Psi_{\alpha/2}\sigma_r, \beta_r + \Psi_{1-\alpha/2}\sigma_r]$ where Ψ_α denotes the α -quantile of the standard normal distribution function. The adequacy of Wald confidence intervals for parameter estimates should be verified by plotting the profile penalized log likelihood (PPL) function. A symmetric shape of the PPL function allows use of Wald intervals, while an asymmetric shape demands profile penalized likelihood intervals (Heinze & Schemper (2002)). Further documentation can be found in Heinze & Ploner (2004).

The latest version now also includes functions to work with multiply imputed data sets, such as generated by the `mice` package. Results on individual fits can be pooled to obtain point and interval estimates, as well as profile likelihood confidence intervals and likelihood profiles in general (Heinze, Ploner and Beyea, 2013).

Author(s)

Georg Heinze <georg.heinze@meduniwien.ac.at> and Meinhard Ploner

References

- Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80, 27–38.
- Heinze G, Schemper M (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine* 21: 2409-2419.
- Heinze G, Ploner M (2003). Fixing the nonconvergence bug in logistic regression with SPLUS and SAS. *Computer Methods and Programs in Biomedicine* 71: 181-187.
- Heinze G, Ploner M (2004). Technical Report 2/2004: A SAS-macro, S-PLUS library and R package to perform logistic regression without convergence problems. Section of Clinical Biometrics, Department of Medical Computer Sciences, Medical University of Vienna, Vienna, Austria. http://www.meduniwien.ac.at/user/georg.heinze/techreps/tr2_2004.pdf
- Heinze G (2006). A comparative investigation of methods for logistic regression with separated or nearly separated data. *Statistics in Medicine* 25: 4216-4226.
- Heinze G, Ploner M, Beyea J (2013). Confidence intervals after multiple imputation: combining profile likelihood information from logistic regressions. *Statistics in Medicine*, to appear.
- Venzon DJ, Moolgavkar AH (1988). A method for computing profile-likelihood based confidence intervals. *Applied Statistics* 37:87-94.

 add1.logistf

Add or Drop All Possible Single Terms to a logistf Model

Description

Compute all the single terms in the scope argument that can be added to or dropped from the model, fit those models and compute a table of the changes in fit.

Usage

```
## S3 method for class 'logistf'
add1(object, scope, test = "PLR", ...)
## S3 method for class 'logistf'
drop1(object, scope, test = "PLR", ...)
```

Arguments

object	a fitted logistf object
scope	The scope of variables considered for adding or dropping. Should be a vector of variable names. Can be left missing; the method will then use all variables in the object's data slot which are not identified as the response variable.
test	The type of test statistic. Currently, only the PLR test (penalized likelihood ratio test) is allowed for logistf fits.
...	further arguments passed to or from other methods.

Details

drop1 and add1 generate a table where for each variable the penalized likelihood ratio chi-squared, the degrees of freedom, and the p-value for dropping/adding this variable are given.

Value

A matrix with nvar rows and 3 columns (Chisquared, degrees of freedom, p-value).

Author(s)

Georg Heinze

Examples

```
data(sex2)
fit<-logistf(data=sex2, case~1, pl=FALSE)
add1(fit)

fit2<-logistf(data=sex2, case~age+oc+dia+vic+vicl+vis)
drop1(fit2)
```

`anova.logistf`*Analysis of penalized deviance for logistf models*

Description

This method compares hierarchical and non-hierarchical logistf models using penalized likelihood ratio tests. It replaces the function `logistftest` of former versions of `logistf`.

Usage

```
## S3 method for class 'logistf'  
anova(object, fit2, formula, method = "nested", ...)
```

Arguments

<code>object</code>	a fitted logistf model object
<code>fit2</code>	another fitted logistf model object, to be compared with <code>object</code>
<code>formula</code>	alternatively to <code>fit2</code> , a formula which specifies terms to omit from the object model fit.
<code>method</code>	One of <code>c("nested", "PLR")</code> . <code>nested</code> is the default for hierarchically nested models, and will compare the penalized likelihood ratio statistics (minus twice the difference between maximized penalized log likelihood and null penalized log likelihood), where the null penalized log likelihood is computed from the same, hierarchically superior model. Note that unlike in maximum likelihood analysis, the null penalized likelihood depends on the penalty (Jeffreys prior) which itself depends on the scope of variables of the hierarchically superior model. <code>PLR</code> compares the difference in penalized likelihood ratio between the two models, where for each model the null penalized likelihood is computed within the scope of variables in that model. For <code>PLR</code> , the models need not be hierarchically nested.
<code>...</code>	Further arguments passed to the method.

Details

Comparing models fitted by penalized methods, one must consider that the penalized likelihoods are not directly comparable, since a penalty is involved. Or in other words, inserting zero for some regression coefficients will not lead to the same penalized likelihood as if the corresponding variables are simply "unknown" to a model. The `anova` method takes care that the same penalty is used for two hierarchically nested models, and if the models are not hierarchically nested, it will first relate each penalized likelihood to its null penalized likelihood, and only compare the resulting penalized likelihood ratio statistics. The chi-squared approximation for this latter method (`PLR`) is considered less accurate than that of the `nested` method. Nevertheless, it is the only way to go for comparison of non-nested models.

Value

An object of class `anova.logistf` with items

<code>chisq</code>	the chisquared statistic for the model comparison
<code>df</code>	the degrees of freedom
<code>pval</code>	the p-value
<code>call</code>	the function call
<code>method</code>	the method of comparison (input)
<code>model1</code>	the first model
<code>model2</code>	the second model which was compared to the first model
<code>PLR1</code>	the PLR statistic of the first model
<code>PLR2</code>	the PLR statistic of the second model; for the nested method, this will be the drop in chi-squared due to setting the coefficients to zero

Author(s)

Georg Heinze

Examples

```
data(sex2)
fit<-logistf(data=sex2, case~age+oc+dia+vic+vicl+vis)

#simultaneous test of variables vic, vicl, vis:
anova(fit, formula=~vic+vicl+vis)

#test versus a simpler model
fit2<-logistf(data=sex2, case~age+oc+dia)
# or: fit2<-update(fit, case~age+oc+dia)
anova(fit,fit2)

# comparison of non-nested models (with different df):
fit3<-logistf(data=sex2, case~age+vic+vicl+vis)
anova(fit2,fit3, method="PLR")
```

backward

Backward elimination of model terms in logistf models

Description

These functions provide simple backward elimination/forward selection procedures for logistf models

Usage

```
backward(object, scope, steps = 1000, slstay = 0.05, trace = TRUE,
  printwork = FALSE, ...)
forward(object, scope, steps = 1000, slentry = 0.05, trace = TRUE,
  printwork = FALSE, pl = TRUE, ...)
```

Arguments

object	A fitted logistf model object. To start with an empty model, create a model fit with a formula= <code><y>~1</code> , <code>pl=FALSE</code> . (Replace <code><y></code> by your response variable.)
scope	The scope of variables to add/drop from the model. If left blank, backward will use the terms of the object fit, and forward will use all variables in <code>object\$data</code> not identified as the response variable. Alternatively, an arbitrary vector of variable names can be given, to allow that only some of the variables will be competitively selected or dropped.
steps	The number of forward selection/backward elimination steps.
slstay	For backward, the significance level to stay in the model.
slentry	For forward, the significance level to enter the model.
trace	If TRUE, protocols selection steps.
printwork	If TRUE, prints each working model that is visited by the selection procedure.
pl	For forward, computes profile likelihood confidence intervals for the final model if TRUE.
...	Further arguments to be passed to methods.

Details

The variable selection is simply performed by repeatedly calling `add1` or `drop1` methods for `logistf`, and is based on penalized likelihood ratio test. It can also properly handle variables that were defined as factors in the original data set.

Value

An updated `logistf` fit with the finally selected model.

Author(s)

Georg Heinze

Examples

```
data(sex2)
fit<-logistf(data=sex2, case~1, pl=FALSE)
fitf<-forward(fit)

fit2<-logistf(data=sex2, case~age+oc+vic+vicl+vis+dia)
fitb<-backward(fit2)
```

CLIP.confint	<i>Confidence intervals after multiple imputation: Combination of Likelihood Profiles</i>
--------------	---

Description

This function implements the new combination of likelihood profiles (CLIP) method described in Heinze, Ploner and Beyea (2013). This method is useful for computing confidence intervals for parameters after multiple imputation of data sets, if the normality assumption on parameter estimates and consequently the validity of applying Rubin's rules (pooling of variances) is in doubt. It consists of combining the profile likelihoods into a posterior. The function `CLIP.confint` searches for those values of a regression coefficient, at which the cumulative distribution function of the posterior is equal to the values specified in the argument `ci.level` (usually 0.025 and 0.975). The search is performed using R's `optimize` function.

Usage

```
CLIP.confint(obj = NULL, variable = NULL, data, firth = TRUE, weightvar = NULL,
             control = logistf.control(), ci.level = c(0.025, 0.975), pvalue = TRUE,
             offset = NULL, bound.lo = NULL, bound.up = NULL, legacy = FALSE)
```

Arguments

<code>obj</code>	Either a list of <code>logistf</code> fits (on multiple imputed data sets), or the result of analysis of a <code>mice</code> (multiply imputed) object using <code>with.mids</code> .
<code>variable</code>	The variable of interest, for which confidence intervals should be computed. If missing, confidence intervals for all variables will be computed.
<code>data</code>	A list of data set corresponding to the model fits. Can be left blank if <code>obj</code> was obtained with the <code>dataout=TRUE</code> option or if <code>obj</code> was obtained by <code>mice</code> .
<code>firth</code>	If <code>TRUE</code> , applies the Firth correction. Should correspond to the entry in <code>obj</code> .
<code>weightvar</code>	An optional weighting variable for each observation.
<code>control</code>	control parameters for <code>logistf</code> , usually obtained by <code>logistf.control()</code>
<code>ci.level</code>	The two confidence levels for each tail of the posterior distribution.
<code>pvalue</code>	If <code>TRUE</code> , will also compute a P-value from the posterior.
<code>offset</code>	An optional offset variable
<code>bound.lo</code>	Bounds (vector of length 2) for the lower limit. Can be left blank. Use only if problems are encountered.
<code>bound.up</code>	Bounds (vector of length 2) for the upper limit. Can be left blank. Use only if problems are encountered.
<code>legacy</code>	If <code>TRUE</code> , will use pure R code for all model fitting. Can be slow. Not recommended.

Details

For each confidence limit, this function performs a binary search to evaluate the combined posterior, which is obtained by first transforming the imputed-data likelihood profiles into cumulative distribution functions (CDFs), and then averaging the CDFs to obtain the CDF of the posterior. Usually, the binary search manages to find the confidence intervals very quickly. The number of iterations (mean and maximum) will be supplied in the output object. Further details on the method can be found in Heinze, Ploner and Beyea (2013).

Value

An object of class `CLIP.confint`, with items

<code>variable</code>	the variable(s) which were analyzed
<code>estimate</code>	the pooled estimate (average over imputations)
<code>ci</code>	the confidence interval(s)
<code>pvalue</code>	the pvalue(s)
<code>imputations</code>	the number of imputed data sets
<code>ci.level</code>	the confidence level (input)
<code>bound.lo</code>	The bounds used for finding the lower confidence limit; usually not of interest. May be useful for error-tracing.
<code>bound.up</code>	The bounds used for finding the upper confidence limit.
<code>iter</code>	The number of iterations (for each variable and each tail (lower or upper)).
<code>call</code>	the call to <code>CLIP.confint</code>

Author(s)

Georg Heinze and Meinhard Ploner

References

Heinze G, Ploner M, Beyea J (2013). Confidence intervals after multiple imputation: combining profile likelihood information from logistic regressions. *Statistics in Medicine*, to appear.

Examples

```
#generate data set with NAs
freq=c(5,2,2,7,5,4)
y<-c(rep(1,freq[1]+freq[2]), rep(0,freq[3]+freq[4]), rep(1,freq[5]), rep(0,freq[6]))
x<-c(rep(1,freq[1]), rep(0,freq[2]), rep(1,freq[3]), rep(0,freq[4]), rep(NA,freq[5]),
      rep(NA,freq[6]))
toy<-data.frame(x=x,y=y)

# impute data set 5 times
set.seed(169)
toymi<-list(0)
for(i in 1:5){
```

```

  toymi[[i]]<-toy
  y1<-toymi[[i]]$y==1 & is.na(toymi[[i]]$x)
  y0<-toymi[[i]]$y==0 & is.na(toymi[[i]]$x)
  xnew1<-rbinom(sum(y1),1,freq[1]/(freq[1]+freq[2]))
  xnew0<-rbinom(sum(y0),1,freq[3]/(freq[3]+freq[4]))
  toymi[[i]]$x[y1==TRUE]<-xnew1
  toymi[[i]]$x[y0==TRUE]<-xnew0
}

# logistf analyses of each imputed data set
fit.list<-lapply(1:5, function(X) logistf(data=toymi[[X]], y~x, pl=TRUE, dataout=TRUE))

# CLIP confidence limits
CLIP.confint(obj=fit.list)

```

CLIP.profile

Combine profile likelihoods from imputed-data model fits

Description

This function uses CLIP (combination of likelihood profiles) to compute the pooled profile of the posterior after multiple imputation.

Usage

```

CLIP.profile(obj = NULL, variable, data, which, firth = TRUE, weightvar,
  control = logistf.control(), offset = NULL, from = NULL, to = NULL,
  steps = 101, legacy = FALSE, keep = FALSE)

```

Arguments

obj	Either a list of logistf fits (on multiple imputed data sets), or the result of analysis of a mice (multiply imputed) object using <code>with.mids</code> .
variable	The variable of interest, for which confidence intervals should be computed. If missing, confidence intervals for all variables will be computed.
data	A list of data set corresponding to the model fits. Can be left blank if obj was obtained with the <code>dataout=TRUE</code> option or if obj was obtained by mice.
which	Alternatively to <code>variable</code> , the argument which allows to specify the variable to compute the profile for as righthand formula, e.g. <code>which=~X</code> .
firth	If TRUE, applies the Firth correction. Should correspond to the entry in obj.
weightvar	An optional weighting variable for each observation.
control	control parameters for logistf, usually obtained by <code>logistf.control()</code>
offset	An optional offset variable.
from	Lowest value for the sequence of values for the regression coefficients for which the profile will be computed. Can be left blank.

to	Highest value for the sequence of values for the regression coefficients for which the profile will be computed. Can be left blank.
steps	Number of steps for the sequence of values for the regression coefficients for which the profile will be computed.
legacy	If TRUE, only R code will be used. Avoid.
keep	If TRUE, keeps the profiles for each imputed data sets in the output object.

Details

While CLIP.confint iterates to find those values at which the CDF of the pooled posterior equals the confidence levels, CLIP.profile will evaluate the whole profile, which enables plotting and evaluating the skewness of the combined and the completed-data profiles. The combined and completed-data profiles are available as cumulative distribution function (CDF) or in the scaling of relative profile likelihood (minus twice the likelihood ratio statistic compared to the maximum). Using a plot method, the pooled posterior can also be displayed as a density.

Value

An object of class CLIP.profile with items

beta	the values of the regression coefficient
cdf	the cumulative distribution function of the posterior
profile	the profile of the posterior
cdf.matrix	An imputations x steps matrix with the values of the completed-data CDFs for each beta
profile.matrix	An imputations x steps matrix with the values of the completed-data profiles for each beta
call	the function call

Author(s)

Georg Heinze and Meinhard Ploner

References

Heinze G, Ploner M, Beyea J (2013). Confidence intervals after multiple imputation: combining profile likelihood information from logistic regressions. *Statistics in Medicine*, to appear.

Examples

```
#generate data set with NAs
freq=c(5,2,2,7,5,4)
y<-c(rep(1,freq[1]+freq[2]), rep(0,freq[3]+freq[4]), rep(1,freq[5]), rep(0,freq[6]))
x<-c(rep(1,freq[1]), rep(0,freq[2]), rep(1,freq[3]), rep(0,freq[4]), rep(NA,freq[5]),
     rep(NA,freq[6]))
toy<-data.frame(x=x,y=y)
```

```

# impute data set 5 times
set.seed(169)
toymi<-list(0)
for(i in 1:5){
  toymi[[i]]<-toy
  y1<-toymi[[i]]$y==1 & is.na(toymi[[i]]$x)
  y0<-toymi[[i]]$y==0 & is.na(toymi[[i]]$x)
  xnew1<-rbinom(sum(y1),1,freq[1]/(freq[1]+freq[2]))
  xnew0<-rbinom(sum(y0),1,freq[3]/(freq[3]+freq[4]))
  toymi[[i]]$x[y1==TRUE]<-xnew1
  toymi[[i]]$x[y0==TRUE]<-xnew0
}

# logistf analyses of each imputed data set
fit.list<-lapply(1:5, function(X) logistf(data=toymi[[X]], y~x, pl=TRUE, dataout=TRUE))

# CLIP profile
xprof<-CLIP.profile(obj=fit.list, variable="x", keep=TRUE)
plot(xprof)

#plot as CDF
plot(xprof, "cdf")

#plot as density
plot(xprof, "density")

```

is.logistf

logistf objects

Description

Checks if an object is of class `logistf`

Usage

```
is.logistf(object)
```

Arguments

`object` An object.

Value

A logical value

Author(s)

Georg Heinze

Examples

```
data(sex2)
fit<-logistf(data=sex2, case~1, pl=FALSE)
is.logistf(fit)
```

logistf

Bias-reduced logistic regression

Description

Implements Firth's penalized-likelihood logistic regression

Usage

```
logistf(formula = attr(data, "formula"), data = sys.parent(), pl = TRUE,
        alpha = 0.05, control, plcontrol, firth = TRUE, init, weights,
        plconf = NULL, dataout = TRUE, ...)
```

Arguments

formula	a formula object, with the response on the left of the operator, and the model terms on the right. The response must be a vector with 0 and 1 or FALSE and TRUE for the outcome, where the higher value (1 or TRUE) is modeled. It is possible to include contrasts, interactions, nested effects, cubic or polynomial splines and all S features as well, e.g. $Y \sim X1 * X2 + ns(X3, df=4)$. From version 1.10, you may also include <code>offset()</code> terms.
data	a data.frame where the variables named in the formula can be found, i. e. the variables containing the binary response and the covariates.
pl	specifies if confidence intervals and tests should be based on the profile penalized log likelihood (<code>pl=TRUE</code> , the default) or on the Wald method (<code>pl=FALSE</code>).
alpha	the significance level ($1-\alpha$ the confidence level, 0.05 as default).
control	Controls Newton-Raphson iteration. Default is <code>control=logistf.control(maxstep, maxit, maxhs, lconv, gconv, xconv)</code>
plcontrol	Controls Newton-Raphson iteration for the estimation of the profile likelihood confidence intervals. Default is <code>plcontrol=logistpl.control(maxstep, maxit, maxhs, lconv, xconv, ortho, pr)</code>
firth	use of Firth's penalized maximum likelihood (<code>firth=TRUE</code> , default) or the standard maximum likelihood method (<code>firth=FALSE</code>) for the logistic regression. Note that by specifying <code>pl=TRUE</code> and <code>firth=FALSE</code> (and probably a lower number of iterations) one obtains profile likelihood confidence intervals for maximum likelihood logistic regression parameters.
init	specifies the initial values of the coefficients for the fitting algorithm.
weights	specifies case weights. Each line of the input data set is multiplied by the corresponding element of weights.

plconf	specifies the variables (as vector of their indices) for which profile likelihood confidence intervals should be computed. Default is to compute for all variables.
dataout	If TRUE, copies the data set to the output object.
...	Further arguments to be passed to logistf.

Details

logistf is the main function of the package. It fits a logistic regression model applying Firth's correction to the likelihood. The following generic methods are available for logistf's output object: print, summary, coef, vcov, confint, anova, extractAIC, add1, drop1, profile, terms, nobs. Furthermore, forward and backward functions perform convenient variable selection. Note that anova, extractAIC, add1, drop1, forward and backward are based on penalized likelihood ratios.

Value

The object returned is of the class logistf and has the following attributes:

coefficients	the coefficients of the parameter in the fitted model.
alpha	the significance level (1- the confidence level) as specified in the input.
terms	the column names of the design matrix
var	the variance-covariance-matrix of the parameters.
df	the number of degrees of freedom in the model.
loglik	a vector of the (penalized) log-likelihood of the full and the restricted models.
iter	the number of iterations needed in the fitting process.
n	the number of observations.
y	the response-vector, i. e. 1 for successes (events) and 0 for failures.
formula	the formula object.
call	the call object.
terms	the model terms (column names of design matrix).
linear.predictors	a vector with the linear predictor of each observation.
predict	a vector with the predicted probability of each observation.
hat.diag	a vector with the diagonal elements of the Hat Matrix.
conv	the convergence status at last iteration: a vector of length 3 with elements: last change in log likelihood, max(abs(score vector)), max change in beta at last iteration.
method	depending on the fitting method 'Penalized ML' or 'Standard ML'.
method.ci	the method in calculating the confidence intervals, i.e. 'profile likelihood' or 'Wald', depending on the argument pl.
ci.lower	the lower confidence limits of the parameter.
ci.upper	the upper confidence limits of the parameter.
prob	the p-values of the specific parameters.

pl.iter	only if pl==TRUE: the number of iterations needed for each confidence limit.
betahist	only if pl==TRUE: the complete history of beta estimates for each confidence limit.
pl.conv	only if pl==TRUE: the convergence status (deviation of log likelihood from target value, last maximum change in beta) for each confidence limit.

If dataout=TRUE, additionally:

data	a copy of the input data set
weights	the weights variable (if applicable)

Author(s)

Georg Heinze and Meinhard Ploner

References

- Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80, 27–38.
- Heinze G, Schemper M (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine* 21: 2409-2419.
- Heinze G, Ploner M (2003). Fixing the nonconvergence bug in logistic regression with SPLUS and SAS. *Computer Methods and Programs in Biomedicine* 71: 181-187.
- Heinze G, Ploner M (2004). Technical Report 2/2004: A SAS-macro, S-PLUS library and R package to perform logistic regression without convergence problems. Section of Clinical Biometrics, Department of Medical Computer Sciences, Medical University of Vienna, Vienna, Austria. http://www.meduniwien.ac.at/user/georg.heinze/techreps/tr2_2004.pdf
- Heinze G (2006). A comparative investigation of methods for logistic regression with separated or nearly separated data. *Statistics in Medicine* 25: 4216-4226.
- Venzon DJ, Moolgavkar AH (1988). A method for computing profile-likelihood based confidence intervals. *Applied Statistics* 37:87-94.

See Also

[drop1.logistf](#) [add1.logistf](#) [anova.logistf](#)

Examples

```
data(sex2)
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)
summary(fit)
nobs(fit)
drop1(fit)
plot(profile(fit,variable="dia"))

extractAIC(fit)

fit1<-update(fit, case ~ age+oc+vic+vicl+vis)
extractAIC(fit1)
```



```

anova(fit,fit1)

data(sexagg)
fit2<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sexagg, weights=COUNT)
summary(fit2)

# simulated SNP example
# not run
set.seed(72341)
snpdata<-rbind(
  matrix(rbinom(2000,2,runif(2000)*0.3),100,20),
  matrix(rbinom(2000,2,runif(2000)*0.5),100,20))
colnames(snpdata)<-paste("SNP",1:20,"_",sep="")
snpdata<-as.data.frame(snpdata)
for(i in 1:20) snpdata[,i]<-as.factor(snpdata[,i])
snpdata$case<-c(rep(0,100),rep(1,100))

fitsnp<-logistf(data=snpdata, formula=case~1, pl=FALSE)
add1(fitsnp)
fitf<-forward(fitsnp)
fitf

```

logistf.control	<i>Control parameters for logistf</i>
-----------------	---------------------------------------

Description

Sets parameters for Newton-Raphson iteration in Firth's penalized-likelihood logistic regression

Usage

```
logistf.control(maxit = 25, maxhs = 5, maxstep = 5, lconv = 1e-05, gconv = 1e-05,
  xconv = 1e-05, collapse=TRUE)
```

Arguments

maxit	the maximum number of iterations
maxhs	the maximum number of step-halvings in one iteration. The increment of the beta vector within one iteration is divided by 2 if the new beta leads to a decrease in log likelihood.
maxstep	specifies the maximum step size in the beta vector within one iteration.
lconv	specifies the convergence criterion for the log likelihood.
gconv	specifies the convergence criterion for the first derivative of the log likelihood (the score vector).

xconv	specifies the convergence criterion for the parameter estimates.
collapse	if TRUE, evaluates all unique combinations of x and y and collapses data set. This may save computing time with large data sets with only categorical (binary) covariates.

Details

logistf.control() is used by logistf and logistftest to set control parameters to default values. Different values can be specified, e. g., by logistf(..., control=logistf.control(maxstep=1)).

Value

maxit	the maximum number of iterations
maxhs	the maximum number of step-halvings in one iteration. The increment of the beta vector within one iteration is divided by 2 if the new beta leads to a decrease in log likelihood.
maxstep	specifies the maximum step size in the beta vector within one iteration.
lconv	specifies the convergence criterion for the log likelihood.
gconv	specifies the convergence criterion for the first derivative of the log likelihood (the score vector).
xconv	specifies the convergence criterion for the parameter estimates.
collapse	if TRUE, evaluates all unique combinations of x and y and collapses data set.

Author(s)

Georg Heinze

Examples

```
data(sexagg)
fit2<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sexagg, weights=COUNT,
  control=logistf.control(maxstep=1))
summary(fit2)
```

logistftest	<i>Penalized likelihood ratio test</i>
-------------	--

Description

This function performs a penalized likelihood ratio test on some (or all) selected factors. The resulting object is of the class logistftest and includes the information printed by the proper print method.

Usage

```
logistftest(object, test, values, firth = TRUE, beta0, weights, control)
```

Arguments

object	a fitted logistf object
test	righthand formula of parameters to test (e.g. $\sim B + D - 1$). As default all parameter apart from the intercept are tested. If the formula includes -1, the intercept is omitted from testing. As alternative to the formula one can give the indexes of the ordered effects to test (a vector of integers). To test only the intercept specify test = $\sim - .$ or test = 1.
values	null hypothesis values, default values are 0. For testing the specific hypothesis $B1=1, B4=2, B5=0$ we specify test= $\sim B1+B4+B5-1$ and values=c(1, 2, 0).
firth	use of Firth's (1993) penalized maximum likelihood (firth=TRUE, default) or the standard maximum likelihood method (firth=FALSE) for the logistic regression. Note that by specifying pl=TRUE and firth=FALSE (and probably a lower number of iterations) one obtains profile likelihood confidence intervals for maximum likelihood logistic regression parameters.
beta0	specifies the initial values of the coefficients for the fitting algorithm.
weights	case weights
control	control parameters for iterative fitting

Details

This function performs a penalized likelihood ratio test on some (or all) selected factors. The resulting object is of the class logistftest and includes the information printed by the proper print method. Further documentation can be found in Heinze & Ploner (2004).

In most cases, the functionality of the logistftest function is replaced by anova.logistf, which is a more standard way to perform likelihood ratio tests. However, as shown in the example below, logistftest provides some specials such as testing against non-zero values. (By the way, anova.logistf calls logistftest.)

A print method is available.

Value

The object returned is of the class logistf and has the following attributes:

testcov	a vector of the fixed values of each covariate; NA stands for a parameter which is not tested.
loglik	a vector of the (penalized) log-likelihood of the full and the restricted models. If the argument beta0 not missing, the full model isn't evaluated.
df	the number of degrees of freedom in the model.
prob	the p-value of the test.
call	the call object
method	depending on the fitting method 'Penalized ML' or 'Standard ML'.
beta	the coefficients on the restricted solution.

Author(s)

Georg Heinze

References

Firth D (1993). Bias reduction of maximum likelihood estimates. *Biometrika* 80, 27–38.

Heinze G, Ploner M (2004). Technical Report 2/2004: A SAS-macro, S-PLUS library and R package to perform logistic regression without convergence problems. Section of Clinical Biometrics, Department of Medical Computer Sciences, Medical University of Vienna, Vienna, Austria. http://www.meduniwien.ac.at/user/georg.heinze/techreps/tr2_2004.pdf

Heinze G (2006). A comparative investigation of methods for logistic regression with separated or nearly separated data. *Statistics in Medicine* 25: 4216-4226.

See Also

[anova.logistf](#)

Examples

```
data(sex2)
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)

logistftest(fit, test = ~ vic + vicl - 1, values = c(2, 0))
```

logistpl.control	<i>Control parameters for logistf profile likelihood confidence interval estimation</i>
------------------	---

Description

Sets parameters for modified Newton-Raphson iteration for finding profile likelihood confidence intervals in Firth's penalized likelihood logistic regression

Usage

```
logistpl.control(maxit=100, maxhs=5, maxstep=5, lconv=0.00001, xconv=0.00001,
  ortho=FALSE, pr=FALSE)
```

Arguments

maxit	the maximum number of iterations
maxhs	the maximum number of step-halvings in one iteration. The increment of the beta vector within one iteration is divided by 2 if the new beta leads to a decrease in log likelihood.
maxstep	specifies the maximum step size in the beta vector within one iteration.

lconv	specifies the convergence criterion for the log likelihood.
xconv	specifies the convergence criterion for the parameter estimates.
ortho	requests orthogonalization of variable for which confidence intervals are computed with respect to other covariates.
pr	request rotation of the matrix spanned by the covariates

Details

logistpl.control() is used by logistf to set control parameters to default values when computing profile likelihood confidence intervals. Different values can be specified, e. g., by logistf(..., control=logistf.control(maxstep=1)).

Value

maxit	the maximum number of iterations
maxhs	the maximum number of step-halvings in one iteration. The increment of the beta vector within one iteration is divided by 2 if the new beta leads to a decrease in log likelihood.
maxstep	specifies the maximum step size in the beta vector within one iteration.
lconv	specifies the convergence criterion for the log likelihood.
xconv	specifies the convergence criterion for the parameter estimates.
ortho	specifies if orthogonalization is requested.
pr	specifies if rotation is requested

Author(s)

Georg Heinze

Examples

```
data(sexagg)
fit2<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sexagg, weights=COUNT,
  plcontrol=logistpl.control(maxstep=1))
summary(fit2)
```

plot.logistf.profile *plot method for logistf likelihood profiles*

Description

provides the plot method for objects created by [profile.logistf](#) or [CLIP.profile](#)

Usage

```
## S3 method for class 'logistf.profile'
plot(x, type = "profile", max1 = TRUE, colmain = "black",
  colimp = "gray", plotmain = T, ylim = NULL, ...)
```

Arguments

x	A profile.logistf object
type	Type of plot: one of c("profile", "cdf", "density")
max1	if type="density", normalizes density to maximum 1
colmain	color for main profile line
colimp	color for completed-data profile lines (for logistf.profile objects that also carry the CLIP.profile class attribute)
plotmain	if FALSE, suppresses the main profile line (for logistf.profile objects that also carry the CLIP.profile class attribute)
ylim	limits for the y-axis
...	further arguments to be passed to plot

Details

The plot method provides three types of plots (profile, CDF, and density representation of a profile likelihood). For objects generated by CLIP.profile, it also allows to show the completed-data profiles along with the pooled profile.

Value

The function is called for its side effects

Author(s)

Georg Heinze and Meinhard Ploner

References

Heinze G, Ploner M, Beyea J (2013). Confidence intervals after multiple imputation: combining profile likelihood information from logistic regressions. *Statistics in Medicine*, to appear.

See Also

[profile.logistf](#), [CLIP.profile](#)

Examples

```
data(sex2)
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)
plot(profile(fit,variable="dia"))
plot(profile(fit,variable="dia"), "cdf")
plot(profile(fit,variable="dia"), "density")

#generate data set with NAs
freq=c(5,2,2,7,5,4)
y<-c(rep(1,freq[1]+freq[2]), rep(0,freq[3]+freq[4]), rep(1,freq[5]), rep(0,freq[6]))
x<-c(rep(1,freq[1]), rep(0,freq[2]), rep(1,freq[3]), rep(0,freq[4]), rep(NA,freq[5]),
     rep(NA,freq[6]))
```

```

toy<-data.frame(x=x,y=y)

# impute data set 5 times
set.seed(169)
toymi<-list(0)
for(i in 1:5){
  toymi[[i]]<-toy
  y1<-toymi[[i]]$y==1 & is.na(toymi[[i]]$x)
  y0<-toymi[[i]]$y==0 & is.na(toymi[[i]]$x)
  xnew1<-rbinom(sum(y1),1,freq[1]/(freq[1]+freq[2]))
  xnew0<-rbinom(sum(y0),1,freq[3]/(freq[3]+freq[4]))
  toymi[[i]]$x[y1==TRUE]<-xnew1
  toymi[[i]]$x[y0==TRUE]<-xnew0
}

# logistf analyses of each imputed data set
fit.list<-lapply(1:5, function(X) logistf(data=toymi[[X]], y~x, pl=TRUE, dataout=TRUE))

# CLIP profile
xprof<-CLIP.profile(obj=fit.list, variable="x", keep=TRUE)
plot(xprof)

#plot as CDF
plot(xprof, "cdf")

#plot as density
plot(xprof, "density")

```

pool.RR	<i>Compute pooled normal confidence intervals (following Rubin's rules) after multiple imputation</i>
---------	---

Description

Computes pooled normal (symmetric) confidence intervals after multiple imputation, following Rubin's rules.

Usage

```
pool.RR(object, method = "plain")
```

Arguments

object A list of fits from logistf.

method A string describing the method to compute the degrees of freedom, "plain" produces the conventional degrees of freedom as in Rubin (1987), and this is required for fits from logistf.

Details

This is a slightly modified version of the `pool()` function of the `mice` package, for use with a list of `logistf`-fits. If `logistf` fits on multiple imputed versions of a data set are on hand in a list object, use this function to pool results using Rubin's rules. If `logistf` fits on multiple imputed versions of a data set are on hand in a `mira` object, use `mice`'s `pool(..., method="plain")` function.

Value

An object of class `mipo`, which stands for 'multiple imputation pooled outcome'. The object is a list containing the following items:

<code>call</code>	The call to the pool function.
<code>call1</code>	The original call how the <code>mira</code> object was calculated.
<code>call2</code>	The original call to the <code>mice</code> function to calculate the underlying <code>midsobject</code> . (NA)
<code>formula</code>	The formula that is used in <code>call1</code> .
<code>nmis</code>	The number of missing entries for each variable. (NA)
<code>m</code>	The number of imputations
<code>qhat</code>	A matrix, containing the estimated coefficients of the <code>m</code> repeated complete data analyses
<code>u</code>	The corresponding <code>m</code> variancematrices of the estimates in an three dimensional array.
<code>qbar</code>	The pooled estimate, formula (3.1.2) Rubin (1987).
<code>ubar</code>	The mean of the variances, formula (3.1.3), Rubin (1987).
<code>b</code>	The within imputation variance, formula (3.1.4), Rubin (1987).
<code>t</code>	Total variance of the pooled estimates, formula (3.1.5), Rubin (1987).
<code>r</code>	Relative increase in variance due to nonresponse, formula (3.1.7), Rubin (1987).
<code>df</code>	Degrees of freedom for <code>t</code> reference distribution, calculated according to the article of Barnard and Rubin (1999).
<code>f</code>	Fraction missing information due to nonresponse, formula (3.1.10), Rubin (1987).

Author(s)

adapted from `mice` package for use with `logistf` objects by Georg Heinze

References

- Barnard, J. and Rubin, D.B. (1999). Small sample degrees of freedom with multiple imputation. *Biometrika*, 86, 948-955.
- Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley and Sons.
- Van Buuren, S., Groothuis-Oudshoorn, K. (2011) mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, Volume 45, Issue 3.
- Pinheiro, J.C. and Bates, D.M. (2000). *Mixed-Effects Models in S and S-PLUS*. Berlin: Springer.

See Also

[mice](#), [CLIP.confint](#), [PVR.confint](#)

Examples

```
#generate data set with NAs
freq=c(5,2,2,7,5,4)
y<-c(rep(1,freq[1]+freq[2]), rep(0,freq[3]+freq[4]), rep(1,freq[5]), rep(0,freq[6]))
x<-c(rep(1,freq[1]), rep(0,freq[2]), rep(1,freq[3]), rep(0,freq[4]), rep(NA,freq[5]),
     rep(NA,freq[6]))
toy<-data.frame(x=x,y=y)

# impute data set 5 times
set.seed(169)
toymi<-list(0)
for(i in 1:5){
  toymi[[i]]<-toy
  y1<-toymi[[i]]$y==1 & is.na(toymi[[i]]$x)
  y0<-toymi[[i]]$y==0 & is.na(toymi[[i]]$x)
  xnew1<-rbinom(sum(y1),1,freq[1]/(freq[1]+freq[2]))
  xnew0<-rbinom(sum(y0),1,freq[3]/(freq[3]+freq[4]))
  toymi[[i]]$x[y1==TRUE]<-xnew1
  toymi[[i]]$x[y0==TRUE]<-xnew0
}

# logistf analyses of each imputed data set
fit.list<-lapply(1:5, function(X) logistf(data=toymi[[X]], y~x, pl=TRUE, dataout=TRUE))
summary(pool.RR(fit.list))
```

print.logistf

print method for logistf objects

Description

provides the print method for logistf objects

Usage

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

Arguments

x	a logistf object
...	additional parameters

Value

A formatted table with results from logistf analysis.

Author(s)

Georg Heinze

<code>print.logistftest</code>	<i>print method for logistftest objects</i>
--------------------------------	---

Description

provides the print method for logistftest objects

Usage

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

Arguments

x	a logistftest object
...	additional parameters

Value

Side effect: prints results from a penalized likelihood ratio test performed by logistftest.

Author(s)

Georg Heinze

profile.logistf	<i>Profile penalized likelihood</i>
-----------------	-------------------------------------

Description

Evaluates the profile penalized likelihood of a variable based on a logistf model fit.

Usage

```
## S3 method for class 'logistf'
profile(fitted, which, variable, steps = 100, pitch = 0.05, limits,
       alpha = 0.05, firth = TRUE, legends = TRUE, control, plcontrol, plot = FALSE,
       ...)
```

Arguments

fitted	An object fitted by logistf
which	A righthand formula to specify the variable for which the profile should be evaluated, e.g., which=~X).
variable	Alternatively to which, a variable name can be given, e.g., variable="X"
steps	number of steps in evaluating the profile likelihood
pitch	alternatively to steps, one may specify the step width in multiples of standard errors
limits	lower and upper limits of parameter values at which profile likelihood is to be evaluated
alpha	the significance level ($1-\alpha$ the confidence level, 0.05 as default).
firth	use of Firth's penalized maximum likelihood (firth=TRUE, default) or the standard maximum likelihood method (firth=FALSE) for the logistic regression.
legends	legends to be included in the optional plot
control	Controls Newton-Raphson iteration. Default is control=logistf.control(maxstep, maxit, maxhs, lconv, gconv, xconv)
plcontrol	Controls Newton-Raphson iteration for the estimation of the profile likelihood confidence intervals. Default is plcontrol=logistpl.control(maxstep, maxit, maxhs, lconv, xconv, ortho, pr)
plot	If TRUE, profile likelihood is plotted. This parameter becomes obsolete as a generic plot function is now provided.
...	Further arguments to be passed.

Value

An object of class logistf.profile with the following items:

beta	parameter values at which likelihood was evaluated
------	--

stdbeta	parameter values divided by standard error
profile	profile likelihood, standardized to 0 at maximum of likelihood. The values in profile are given as minus χ^2 .
loglike	unstandardized profile likelihood
signed.root	signed root (z) of χ^2 values (negative for values below the maximum likelihood estimate, positive for values above the maximum likelihood estimate)
cdf	profile likelihood expressed as cumulative distribution function, obtained as $\Phi(z)$, where Φ denotes the standard normal distribution function.

Author(s)

Georg Heinze and Meinhard Ploner

See Also

[plot.profile.logistf](#)

Examples

```
data(sex2)
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)
plot(profile(fit,variable="dia"))
plot(profile(fit,variable="dia"), "cdf")
plot(profile(fit,variable="dia"), "density")
```

PVR.confint

Pseudo-variance modification of Rubin's rules

Description

The pseudo-variance modification proposed by Heinze, Ploner and Beyea (2013) provides a quick way to adapt Rubin's rules to situations of a non-normal distribution of a regression coefficient. However, the approximation is less accurate than that of the CLIP method.

Usage

```
PVR.confint(obj, variable, skewbeta = FALSE)
```

Arguments

obj	A list of objects fitted by <code>logistf</code> .
variable	The variable(s) to compute the PVR confidence intervals, either provided as names or as numbers.
skewbeta	If TRUE, incorporates information on the skewness of the parameter estimates across the imputed data sets.

Details

The pseudo-variance modification computes a lower and an upper pseudo-variance, which are based on the distance between profile likelihood limits and the parameter estimates. These are then plugged into the usual Rubin's rules method of variance combination.

Value

An object of class `PVR.confint` with items

<code>estimate</code>	the pooled parameter estimate(s) (the average across completed-data estimates)
<code>ci</code>	the confidence intervals based on the PVR method
<code>lower.var</code>	the lower pseudo-variance(s)
<code>upper.var</code>	the upper pseudo-variance(s)
<code>conflev</code>	the confidence level: this is determined by the confidence level (1-alpha) used in the input fit objects
<code>call</code>	the function call
<code>variable</code>	the variable(s) for which confidence intervals were computed

Author(s)

Georg Heinze

References

Heinze G, Ploner M, Beyea J (2013). Confidence intervals after multiple imputation: combining profile likelihood information from logistic regressions. *Statistics in Medicine*, to appear.

Examples

```
#generate data set with NAs
freq=c(5,2,2,7,5,4)
y<-c(rep(1,freq[1]+freq[2]), rep(0,freq[3]+freq[4]), rep(1,freq[5]), rep(0,freq[6]))
x<-c(rep(1,freq[1]), rep(0,freq[2]), rep(1,freq[3]), rep(0,freq[4]), rep(NA,freq[5]),
     rep(NA,freq[6]))
toy<-data.frame(x=x,y=y)

# impute data set 5 times
set.seed(169)
toymi<-list(0)
for(i in 1:5){
  toymi[[i]]<-toy
  y1<-toymi[[i]]$y==1 & is.na(toymi[[i]]$x)
  y0<-toymi[[i]]$y==0 & is.na(toymi[[i]]$x)
  xnew1<-rbinom(sum(y1),1,freq[1]/(freq[1]+freq[2]))
  xnew0<-rbinom(sum(y0),1,freq[3]/(freq[3]+freq[4]))
  toymi[[i]]$x[y1==TRUE]<-xnew1
  toymi[[i]]$x[y0==TRUE]<-xnew0
}
```

```
# logistf analyses of each imputed data set
fit.list<-lapply(1:5, function(X) logistf(data=toymi[[X]], y~x, pl=TRUE, dataout=TRUE))

# CLIP confidence limits
PVR.confint(obj=fit.list)
```

sex2

Urinary tract infection in American college students

Description

This data set deals with urinary tract infection in sexually active college women, along with covariate information on age and contraceptive use. The variables are all binary and coded in 1 (condition is present) and 0 (condition is absent): case (urinary tract infection, the study outcome variable), age (≥ 24 years), dia (use of diaphragm), oc (use of oral contraceptive), vic (use of condom), vicl (use of lubricated condom), and vis (use of spermicide).

Usage

```
sex2
sexagg
```

Format

sex2: a data.frame containing 239 observations; sexagg: an aggregated data.frame containing 31 observations with case weights (COUNT).

Source

www.cytel.com, the full data was available for download at their webpage, and is distributed along with their LogXact software. Here a reduced version of the original data is reproduced.

References

Cytel Inc., (2010) *LogXact 9 user manual*, Cambridge, MA:Cytel Inc.

summary.logistf	summary method for logistf objects
-----------------	------------------------------------

Description

provides a summary method for logistf objects

Usage

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

Arguments

object	a logistf object
...	additional parameters

Value

Side effect: a summary table with results from logistf analysis.

Author(s)

Georg Heinze

Examples

```
data(sex2)  
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)  
summary(fit)
```

vcov.logistf	vcov method for logistf objects
--------------	---------------------------------

Description

provides the vcov method for logistf objects

Usage

```
## S3 method for class 'logistf'  
vcov(object, ...)
```

Arguments

object	a logistf object
...	further arguments

Value

the variance-covariance matrix of a `logistf` fit

Author(s)

Georg Heinze

See Also

[logistf](#)

Examples

```
data(sex2)
fit<-logistf(case ~ age+oc+vic+vicl+vis+dia, data=sex2)
vcov(fit)
```


Index

- *Topic **\textasciitildekwd1**
 - PVR.confint, 28
 - vcov.logistf, 31
- *Topic **\textasciitildekwd2**
 - PVR.confint, 28
 - vcov.logistf, 31
- *Topic **datasets**
 - sex2, 30
- *Topic **models**
 - add1.logistf, 5
 - anova.logistf, 6
 - backward, 7
 - CLIP.confint, 9
 - CLIP.profile, 11
 - is.logistf, 13
 - logistf, 14
 - logistf-package, 2
 - logistf.control, 17
 - logistftest, 18
 - logistpl.control, 20
 - plot.logistf.profile, 21
 - pool.RR, 23
 - print.logistf, 25
 - print.logistftest, 26
 - profile.logistf, 27
 - summary.logistf, 31
- *Topic **regression**
 - add1.logistf, 5
 - anova.logistf, 6
 - backward, 7
 - CLIP.confint, 9
 - CLIP.profile, 11
 - is.logistf, 13
 - logistf, 14
 - logistf-package, 2
 - logistf.control, 17
 - logistftest, 18
 - logistpl.control, 20
 - plot.logistf.profile, 21
 - pool.RR, 23
 - print.logistf, 25
 - print.logistftest, 26
 - profile.logistf, 21, 22, 27
 - PVR.confint, 25, 28
 - sex2, 30
 - sexagg (sex2), 30
 - summary.logistf, 31
 - vcov.logistf, 31
- pool.RR, 23
- print.logistf, 25
- print.logistftest, 26
- profile.logistf, 27
- summary.logistf, 31
- add1.logistf, 5, 16
- anova.logistf, 6, 16, 20
- backward, 7
- CLIP.confint, 9, 25
- CLIP.profile, 11, 21, 22
- drop1.logistf, 16
- drop1.logistf (add1.logistf), 5
- forward (backward), 7
- is.logistf, 13
- logistf, 14, 32
- logistf-package, 2
- logistf.control, 17
- logistftest, 18
- logistpl.control, 20
- mice, 4, 24, 25
- plot.logistf.profile, 21
- plot.profile.logistf, 28
- plot.profile.logistf (profile.logistf), 27
- pool.RR, 23
- print.logistf, 25
- print.logistftest, 26
- profile.logistf, 21, 22, 27
- PVR.confint, 25, 28
- sex2, 30
- sexagg (sex2), 30
- summary.logistf, 31
- vcov.logistf, 31